Universidad del Azuay

Facultad de Ciencia y Tecnología
Maestría en Telemática

Banca electrónica usando dispositivos inalámbricos con tecnología SMS

Trabajo de graduación previo a la obtención del título de Magíster en Telemática

Autor: Ing. Juan Diego Landy Andrade
Director: Ing. Pablo Pintado, MBA

Cuenca, Ecuador
2007

DEDICATORIA

Dedico este trabajo de graduación a mi hijo Juan Bernardo y a mi esposa Lourdes, quienes fueron los grandes sacrificados en mis largas horas de ausencia y estudio, su apoyo incondicional ha sido mi gran fortaleza para la obtención de este título.

AGRADECIMIENTOS

A Dios, por ser tan bondadoso conmigo y permitirme culminar con éxito esta alta distinción académica con salud y sabiduría junto a mi familia.

A mis padres por enseñarme que en la vida los objetivos fundamentales siempre son conseguidos con esfuerzo y dedicación.

A mi familia quienes asumieron el bienestar de mi hogar y estuvieron presentes durante todo el tiempo que demandó estudiar esta Maestría.

Quiero expresar mi profundo agradecimiento al Ing. Pablo Pintado quien puso toda su experiencia profesional para una acertada dirección en este trabajo de graduación.

RESUMEN

El presente trabajo de graduación utiliza la tecnología celular SMS – Short Message Service, para procesar transacciones bancarias a través del uso de una pasarela SMS, de tal manera que los clientes tengan acceso a los servicios bancarios enviando solicitudes o peticiones a través de su celular y el banco responda a estas transacciones electrónicas con la información solicitada de manera fácil, rápida y segura sin necesidad de que el cliente se acerque a los agencias bancarias, y por otra parte que el banco pueda utilizar esta tecnología para el envío de notificaciones o mensajes de advertencia de eventos que esten por suceder o que han sucedido sobre los productos que tenga contratado el cliente con la institución.

El objetivo de este presente trabajo de tesis es la implementación de un sistema denominado "Banca Electrónica usando dispositivos inalámbricos con tecnología SMS" utilizando para este desarrollo de software la metodología de Rational Unified Process (RUP) con UML como lenguaje de modelamiento.

Este estudio genera un sistema flexible, robusto, que permite a la institución bancaria ir implementando nuevas transacciones bancarias dentro de este esquema con el mínimo de desarrollo. Todas las herramientas de software utilizadas son tecnología Microsoft y complementa muy bien con una metodología de análisis y diseño orientado a objetos RUP debido a que es una de las más maduras y eficientes que existen en la actualidad.

El alcance del proyecto define e implementa un conjunto de transacciones bancarias específicas, en donde se desarrollo todas las fases de la metodología RUP.

Finalmente se presenta en este trabajo, un resumen de las principales conclusiones y recomendaciones obtenidas en cada una de las fases de la metodología.

ABSTRACT

The present graduation thesis uses SMS (Short Message Service) cellular technology to process banking transactions through SMS footbridge, of such way clients have access to the banking services sending requests through their cellular and the bank responds those electronic transactions with the requested information in an easy, fast and safe way without going to the banking agencies. In other hand, the bank can use this technology to send notifications or messages of warnings events that can happen or have happened in products that clients have contracted with the bank institution.

The objective of this thesis is the implementation of a system called “Electronic Bank using wireless devices with SMS technology” using, for this software development, the methodology of Rational Unified Process (RUP) with UML.

This study generates a flexible and robust system that allows to the banking institution implements new banking transactions in this scheme with the minimum of development. All the used software tools are Microsoft technology and can complement very well with an Oriented Object RUP analysis and design methodology because is one of most mature and efficient that exists at the present time.

The reach of this project defines and implements a set of specific banking transactions, using all the phases of RUP methodology.

Finally this work presents a summary of the main conclusions and recommendations obtained in each phase of the methodology.

INDICE GENERAL

CAPITULO 1. PLAN DE DESARROLLO DE SOFTWARE 1

1.1 INTRODUCCIÓN 1
1.1.1 Propósito 1
1.1.2 Alcance 2
1.1.3 Resumen 2
1.2 VISTA GENERAL DEL PROYECTO 3
1.2.1 Propósito, Alcance y Objetivos 3
1.2.2 Suposiciones y Restricciones 6
1.2.3 Entregables del proyecto 7
1.2.4 Evolución del Plan de Desarrollo del Software 10
1.3 ORGANIZACIÓN DEL PROYECTO 10
1.3.1 Participantes en el Proyecto 10
1.3.2 Interfaces Externas 11
1.3.3 Roles y Responsabilidades 11
1.4 GESTIÓN DEL PROCESO 12
1.4.1 Estimaciones del Proyecto 12
1.4.2 Plan del Proyecto 13
1.4.3 Seguimiento y Control del Proyecto 22
1.5 CONCLUSIONES 22
1.6 RECOMENDACIONES 23

CAPITULO 2. REQUERIMIENTOS DEL SISTEMA 24

2.1 INTRODUCCIÓN 24
2.2 VISIÓN 24
2.2.1 Aspectos generales 24
2.2.2 Posicionamiento 26
2.3 ESPECIFICACION DE REQUISITOS DE SOFTWARE – ERS 30
2.3.1 Introducción 30
2.3.2 DESCRIPCION GENERAL 31
2.3.3 Requisitos específicos 38
2.3.4 Requisitos de interfaces externas 50
2.4 CONCLUSIONES 52
2.5 RECOMENDACIONES 53

CAPITULO 3. ANÁLISIS ORIENTADO A OBJETOS 55

3.1 INTRODUCCIÓN 55
3.2 CASO EXPANDIDO DE USOS: ESENCIALES 55
3.2.1 Gestión de administración del sistema de Banca Celular 56
3.2.2 Gestión BackOffice del producto Banca Celular 58
3.2.3 Caso de uso gestión de suscripción de clientes 71
3.2.4 Caso de uso gestión de suscripción de un servicio adicional 75
3.2.5 Gestión de adquiriencia de transacciones tipo PULL 80
3.2.6 Gestión de generación de transacciones tipo PUSH ONLINE con suscripción 92
3.2.7 Gestión de generación de transacciones tipo PUSH ONLINE sin suscripción 103
3.2.8 Gestión de generación de transacciones tipo PUSH BATCH con suscripción 107
3.2.9 Gestión de generación de transacciones tipo PUSH BATCH sin suscripción 109
3.3 MODELO CONCEPTUAL 113
3.4 DIAGRAMAS DE SECUENCIA 114
3.5 CONTRATOS DE OPERACIÓN 114
3.6 DIAGRAMAS DE ESTADO 122
3.7 CLASIFICACIÓN Y PROGRAMACIÓN DE CASOS DE USO. VERSIÓN 1. 122
3.8 CONCLUSIONES 123
3.9 RECOMENDACIONES 124

CAPITULO 4. DISEÑO ORIENTADO A OBJETOS 126

4.1 INTRODUCCIÓN 126
4.2 DISEÑO DE INTERFAZ DE USUARIO 127
4.2.1 Diseño de mensajes SMS para las operaciones PULL. 127
4.2.2 Diseño de interfaces de usuario para el BackOffice. 130
4.3 DIAGRAMAS DE INTERACCIÓN 137
4.4 DIAGRAMAS DE CLASES 138
4.4.1 Arquitecturas Multicapas 138
4.4.2 Diseño de Componentes 139
4.4.3 Clase ServiciosElectronicos 141
4.4.4 Clase ServiciosOperacion 183
4.5 DIAGRAMAS DE ARQUITECTURA. 239
4.5.1 Vista lógica. 239
4.5.2 Vista física 241
4.5.3 Vista de implementación 244
4.6 DIAGRAMA DE BASE DE DATOS DBSERVICIOSE 245
4.6.1 Diseño conceptual 245
4.6.2 Diseño lógico 246
4.6.3 Diseño físico 246
4.6.4 Diccionario de base de datos 247
4.7 CONCLUSIONES 248
4.8 RECOMENDACIONES 250

CAPITULO 5. IMPLEMENTACION Y PRUEBAS DEL SISTEMA 252

5.1 INTRODUCCIÓN 252
5.2 CÓDIGO FUENTE 253
5.2.1 Código fuente ServiciosElectronicos 253
5.2.2 Código fuente ServiciosOperacion 273
5.2.3 Página SuscripcionSMS.asp 281
5.3 PRUEBAS ORIENTADAS A OBJETOS 289
5.3.1 Modelos de Prueba AOO (Análisis Orientado a Objetos) y DOO (Diseño Orientado a Objetos). 289
5.3.2 Definición de estrategias para pruebas Orientadas a objetos. 292
5.3.3 Pruebas de validación. Definición de casos de pruebas y resultados 295
5.3.3.1 Operaciones PULL 295
5.3.4 Pruebas de esfuerzo o estrés. 335
5.3.5 Pruebas de rendimiento reales 336
5.4 IMPLEMENTACIÓN – ESQUEMA TECNOLÓGICO 337
5.4.2 Infraestructura física 338
5.4.3 Infraestructura lógica 338
5.5 CONCLUSIONES. 342
5.6 RECOMENDACIONES 343

CONCLUSIONES 345

RECOMENDACIONES 349

GLOSARIO 353

BIBLIOGRAFIA 360

ANEXOS 362

INDICE DE TABLAS

Tabla 1.1 Roles de los participantes de este proyecto 12

Tabla 1.2 Fases del proyecto 15

Tabla 1.3 Notas – Fase de Planeación y Elaboración 17

Tabla 1.4 Actividades de la fase de Planeación y Elaboración 18

Tabla 1.5 Notas – Fase de Análisis 19

Tabla 1.6 Actividades de la Fase de Análisis 20

Tabla 1.7 Notas – Fase de Diseño 20

Tabla 1.8 Actividades de la Fase de Diseño 20

Tabla 2.1 Sentencia que define el problema 28

Tabla 2.2 Sentencia que define la posición del producto 29

Tabla 2.3 Categoría de las funciones 32

Tabla 2.4 Funciones del sistema banca celular 32

Tabla 2.5 Atributos del sistema banca celular 34

Tabla 2.6 Relación funciones del sistema con sus atributos 35

Tabla 4.1 Diseño de comandos SMS para operaciones tipo PULL banca celular 129

Tabla 4.2 Estructura de diccionario de datos 248

Tabla 5.1 Resultado de pruebas de operación PULL. 296

Tabla 5.2 Resultado de pruebas de operación PASSW. 299

Tabla 5.3 Resultado de pruebas de operación SALDO. 302

Tabla 5.4 Resultado de pruebas de operación CONCH. 304

Tabla 5.5 Resultado de pruebas de operación SUSSA. 306

Tabla 5.6 Resultado de pruebas de operación SALTC. 309

Tabla 5.7 Resultado de pruebas de operación CUPTC. 311

Tabla 5.8 Resultado de pruebas de operación NOTON. 312

Tabla 5.9 Resultado de pruebas de operación NOLSU. 314

Tabla 5.10 Resultado de pruebas de operación NOTBA. 318

Tabla 5.11 Resultado de pruebas de procesos automáticos - PUSH Batch sin suscripción. 320

Tabla 5.12 Resultado de pruebas de las transacciones del BACKOFFICE banca celular. 322

INDICE DE FIGURAS

Figura 1.1 Pasos en macro nivel en un proceso de desarrollo RUP 13

Figura 1.2 Ciclos iterativos de desarrollo 14

Figura 1.3 Relación entre los artefactos generados en una fase de planeación y elaboración 18

Figura 1.4 Dependencias de los artefactos durante la fase de construcción 21

Figura 2.1 Representación grafica de un caso de uso y un actor 39

Figura 2.2 Diagrama de casos de uso del sistema de banca celular 40

Figura 4.1 Diseño de pantalla de seguridad 130

Figura 4.2 Diseño pantalla de seguridad para supervisor 131

Figura 4.3 Diseño de pantalla de suscripción 131

Figura 4.4 Diseño de pantalla de modificación de teléfono 132

Figura 4.5 Diseño de pantalla para consulta de suscripciones 132

Figura 4.6 Diseño de pantalla para consulta de servicios adicionales de un cliente 133

Figura 4.7 Diseño de pantalla para la suscripción de un servicio adicional 133

Figura 4.8 Diseño de pantalla para inactivación del servicio banca celular 134

Figura 4.9 Diseño de pantalla para inactivación de un servicio adicional 134

Figura 4.10 Diseño de pantalla para el cambio de clave del servicio banca celular 135

Figura 4.11 Diseño de pantalla para la consulta de transacciones banca celular 135

Figura 4.12 Diseño de pantalla de consulta de clave de servicio banca celular 136

Figura 4.13 Diseño de pantalla para el reverso de una suscripción banca celular 136

Figura 4.14 Diseño de pantalla para el reverso de una suscripción de un servicio adiciona 137

Figura 4.15 Diagrama de sistema 'multi-tiered' con objetos interfaz de usuario separados de los objetos 'business', que representa la arquitectura multicapas 139

Figura 4.16 Notaciones para mostrar una interfaz de una clase de manera gráfica 140

Figura 4.17 Diagrama de componentes del sistema Banca Celular 141

Figura 4.18 Diagrama de arquitectura. Vista de diseño conceptual 240

Figura 4.19 Patrón de diseño en capas. 241

Figura 4.20 Diagrama de arquitectura. Vista de diseño físico del BackOffice. 242

Figura 4.21 Diagrama de arquitectura. Vista de diseño físico de transacciones PULL. 243

Figura 4.22 Diagrama de arquitectura. Vista de diseño físico de transacciones PUSH. 244

Figura 4.23 Diagrama de arquitectura. Vista de implementación. 245

Figura 4.24 Diseño físico de la tabla TS-Suscripcion 247

Figura 5.1 Estrategia de pruebas orientadas a objetos 293

Figura 5.2 Configuración de mensajes Push batch sin suscripción 319

Figura 5.3 Módulo seguridad. Ingreso de usuario incorrecto. 322

Figura 5.4 Módulo seguridad. Ingreso de password incorrecto. 323

Figura 5.5 Módulo menú BackOffice. Opciones al usuario previa autentificación. 324

Figura 5.6 Módulo suscripción. Tarjeta incorrecta. 324

Figura 5.7 Módulo suscripción. Suscripción correcta. 325

Figura 5.8 Módulo consulta de suscripciones. 326

Figura 5.9 Módulo modificación de teléfono. 326

Figura 5.10 Módulo suscripción a un servicio adicional. 327

Figura 5.11 Módulo suscripción a un servicio adicional. 328

Figura 5.12 Módulo suscripción de un servicio adicional ya asociado previamente. 328

Figura 5.13 Módulo consulta de servicios adicionales. 329

Figura 5.14 Módulo consulta de servicios adicionales. 330

Figura 5.15 Módulo in activación de suscripción principal banca celular. 330

Figura 5.16 Módulo cambio de clave banca celular. 331

Figura 5.17 Módulo consulta de clave banca celular. 332

Figura 5.18 Módulo consulta de clave banca celular. 333

Figura 5.19 Módulo reverso de un servicio adicional. 334

Figura 5.20 Módulo reverso de un servicio adicional. 335

Figura 5.21 Esquema tecnológico a implementar. 337

Figura 5.22 Base de datos dbServiciosE en SQLServer 2000 Enterprise Edition. 340

Figura 5.23 Servidor de componentes, el gráfico muestra los paquetes ServiciosElectronicos y ServiciosOperacion. 341

Figura 5.24 Directorio virtual SMS en donde apunta a las páginas ASP en el servidor IIS..342

CAPITULO 1. PLAN DE DESARROLLO DE SOFTWARE

1.1 Introducción

Este Plan de Desarrollo del Software es una versión preliminar preparada para ser incluida en la propuesta elaborada como respuesta a la tesis "Banca electrónica usando dispositivos inalámbricos con tecnología SMS". Este documento provee una visión global del enfoque de desarrollo propuesto.

El proyecto ha sido presentado por Juan Diego Landy Andrade basado en una metodología de Rational Unified Process en la que únicamente se procederá a cumplir con las tres primeras fases que marca la metodología, constando únicamente en la tercera fase de una iteración. Es importante destacar esto puesto que utilizaremos la terminología RUP en este documento. Se incluirá el detalle para las fases de Inicio y Elaboración y adicionalmente se esbozarán las fases posteriores de Construcción y Transición para dar una visión global de todo proceso.

El enfoque desarrollo propuesto constituye una configuración del proceso RUP de acuerdo a las características del proyecto, seleccionando los roles de los participantes, las actividades a realizar y los artefactos (entregables) que serán generados. Este documento es a su vez uno de los artefactos de RUP.

1.1.1 *Propósito*

El propósito del Plan de Desarrollo de Software es proporcionar la información necesaria para controlar el proyecto. En él se describe el enfoque de desarrollo del software.

Los usuarios del Plan de Desarrollo del Software son:

- El director de la tesis para controlar los avances de las tesis y hacer un seguimiento los documentos o artefactos que se entregan al finalizar cada una de las fases de RUP.
- El ejecutor de la tesis lo utiliza para organizar la agenda y necesidades de recursos, así mismo ayuda a entender lo qué se debe hacer, cuándo se debe hacer y qué otras actividades dependen de ello.

1.1.2 *Alcance*

El Plan de Desarrollo del Software describe el plan global usado para el desarrollo de la tesis "Banca celular usando dispositivos inalámbricos con tecnología SMS". El detalle de las iteraciones individuales se describe en los planes de cada iteración, documentos que se aportan en forma separada. Durante el proceso de desarrollo en el artefacto "Visión" se definen las características del producto a desarrollar, lo cual constituye la base para la planificación de las iteraciones. Para la versión 1.0 del Plan de Desarrollo del Software, nos hemos basado en la captura de requisitos por medio del stakeholder representante de la empresa para hacer una estimación aproximada, una vez comenzado el proyecto y durante la fase de Inicio se generará la primera versión del artefacto "Visión", el cual se utilizará para refinar este documento. Posteriormente, el avance del proyecto y el seguimiento en cada una de las iteraciones ocasionará el ajuste de este documento produciendo nuevas versiones actualizadas.

1.1.3 *Resumen*

Después de esta introducción, el resto del documento está organizado en las siguientes secciones:

Vista General del Proyecto — proporciona una descripción del propósito, alcance y objetivos del proyecto, estableciendo los artefactos que serán producidos y utilizados durante el proyecto.

Organización del Proyecto — describe la estructura organizacional del equipo de desarrollo.

Gestión del Proceso — explica los costos y planificación estimada, define las fases e hitos del proyecto y describe cómo se realizará su seguimiento.

Planes y Guías de aplicación — proporciona una vista global del proceso de desarrollo de software, incluyendo métodos, herramientas y técnicas que serán utilizadas.

1.2 Vista General del Proyecto

1.2.1 *Propósito, Alcance y Objetivos*

La información que a continuación se incluye ha sido extraída de las diferentes reuniones que se han celebrado con el stakeholder de la empresa desde el inicio del proyecto, gerencia de proyectos.

El presente planteamiento hace uso de la tecnología SMS aplicadas en la banca celular, que nos permita cambiar el paradigma de negocio que implica que el cliente se acerque a la institución bancaria para realizar una transacción.

La idea es que con el uso de una pasarela SMS los clientes tengan acceso a los servicios bancarios enviando solicitudes o peticiones a través de su celular y el banco responda a estas transacciones electrónicas con la información solicitada de manera fácil, rápida y segura sin necesidad de que el cliente se acerque a los agencias bancarias, soportando las molestosas filas y esperas de turnos de atención, así como también la institución financiera necesita enviar notificaciones o avisos a sus clientes a través de su celular de eventos que ocurren sobre acciones relacionadas entre el banco y el cliente, cuyo resultado es el mejoramiento de la calidad del servicio bancario

Objetivo General

Desarrollar un sistema de Banca celular que permita mediante un dispositivo inalámbrico celular acceder a los servicios financieros de la banca utilizando tecnología SMS.

Objetivos específicos

El proyecto debe proporcionar una propuesta para el desarrollo de todos los subsistemas implicados en la gestión de transacciones electrónicas bancarias a través de un dispositivo celular usando tecnología SMS".

Estos subsistemas se pueden diferenciar en siete grandes bloques:

a) Gestión de autentificación de clientes y celulares, incluyendo:

- Procedimiento de validación de un cliente de la institución bancaria a través de su tarjeta de débito o crédito.
- Procedimiento de suscripción al servicio vía celular.
- Procedimiento de suscripción al servicio usando otros canales - Intranet.
- Procedimiento de suscripción de un cliente a un servicio complementario o adicional.

b) Gestión de BackOffice del servicio Banca celular , incluyendo:

- Gestión para el cambio o modificación del teléfono original de suscripción
- Consulta de suscripciones realizadas por un cliente

- Gestión para la consulta servicios adicionales
- Inactivación de una suscripción
- Inactivación de un servicio adicional
- Gestión para el cambio de clave del servicio banca celular
- Gestión de consulta de transacciones realizadas por un cliente
- Consulta de clave servicio electrónico Banca celular
- Procedimiento de reverso de las suscripciones de un cliente y sus comisiones
- Reversa servicios adicionales banca celular y comisión de cobro
- Estado de Cuenta

c) Gestión de adquiriencia de transacciones tipo PULL (Usuario iniciar una transacción desde su celular solicitando un servicio financiero), incluyendo:

- Verificación de suscripción del servicio
- Verificación de clave en transacciones delicadas
- Invocación de objetos del servicio
- Gestión de débito por comisión
- Registro en bitácora transaccional de la transacción

d) Gestión de generación notificaciones tipo PUSH-Online (Institución bancaria envía alertas en forma asincrónica de ciertos eventos realizados sobre sus cuentas), incluyendo.

- Verificación de suscripción del servicio y servicio adicional
- Gestión de débito por comisión
- Registro en bitácora transaccional de la transacción
- Gestión de avisos Push-Online sin suscripción

e) Gestión de transacciones de tipo PUSH-Batch (Sistemas de la institución bancaria generen y envíen a través de este subsistema notificaciones alertas relacionadas con el cliente)

- Verificación de suscripción del servicio y servicio adicional
- Gestión de débito por comisión
- Gestión de avisos Push-Batch sin suscripción

f) Gestión de cobros del servicio, incluyendo:

- Cobro de comisiones on-line.
- Cobro de comisiones en proceso batch.

1.2.2 *Suposiciones y Restricciones*

Las suposiciones y restricciones respecto del sistema, y que se derivan directamente de las entrevistas con el stakeholder de la empresa son:

a) Arrendamiento de la pasarela SMS como switch entre las operadoras.

b) Debe contemplarse las implicaciones de los siguientes puntos críticos:

- Sistemas seguros: protección de información, seguridad en las trasmisiones de datos (PKI), etc.
- Gestión de flujos de trabajo, seguridad de transacciones e intercambio de información
- Adaptación a la normativa de Protección de Datos

c) El subsistema “Gestión de BackOffice del servicio Banca celular” debe diseñarse como módulo independiente para ser utilizado a nivel nacional a través de la Intranet.

Como es natural, la lista de suposiciones y restricciones se incrementará durante el desarrollo del proyecto, particularmente una vez establecido el artefacto “Visión”.

1.2.3 *Entregables del proyecto*

A continuación se indican y describen cada uno de los artefactos que serán generados y utilizados por el proyecto y que constituyen los entregables. Esta lista constituye la configuración de RUP desde la perspectiva de artefactos, y que proponemos para este proyecto.

Es preciso destacar que de acuerdo a la filosofía de RUP (y de todo proceso iterativo e incremental), todos los artefactos son objeto de modificaciones a lo largo del proceso de desarrollo, con lo cual, sólo al término del proceso podríamos tener una versión definitiva y completa de cada uno de ellos. Sin embargo, el resultado de cada iteración y los hitos del proyecto están enfocados a conseguir un cierto grado de completitud y estabilidad de los artefactos. Esto será indicado más adelante cuando se presenten los objetivos de cada iteración.

1) Plan de Desarrollo del Software

Es el presente documento.

2) Modelo de Casos de Uso del Negocio

Es un modelo de las funciones de negocio vistas desde la perspectiva de los actores externos (Agentes de registro, solicitantes finales, otros sistemas etc.) permite situar al sistema en el contexto organizacional haciendo énfasis en los objetivos en este ámbito. Este modelo se representa con un Diagrama de Casos de Uso usando estereotipos específicos para este modelo.

3) Glosario

Es un documento que define los principales términos usados en el proyecto. Permite establecer una terminología consensuada. .

4) Modelo de Casos de Uso

El modelo de Casos de Uso presenta las funciones del sistema y los actores que hacen uso de ellas. Se representa mediante Diagramas de Casos de Uso.

5) Visión

Este documento define la visión del producto desde la perspectiva del cliente, especificando las necesidades y características del producto. Constituye una base de acuerdo en cuanto a los requisitos del sistema.

6) Especificaciones de Casos de Uso

Para los casos de uso que lo requieran (cuya funcionalidad no sea evidente o que no baste con una simple descripción narrativa) se realiza una descripción detallada utilizando una plantilla de documento, donde se incluyen: precondiciones, post-condiciones, flujo de eventos, requisitos no-funcionales asociados. También, para casos de uso cuyo flujo de eventos sea complejo podrá adjuntarse una representación gráfica mediante un Diagrama de Actividad.

7) Prototipos de Interfaces de Usuario

Se trata de prototipos que permiten al usuario hacerse una idea más o menos precisa de las interfaces que proveerá el sistema y así, conseguir retroalimentación de su parte respecto a los requisitos del sistema. Estos prototipos se realizarán como: dibujos a mano en papel, dibujos con alguna herramienta gráfica o prototipos ejecutables interactivos, siguiendo ese orden de acuerdo al avance del proyecto. Sólo los de este último tipo serán entregados al final de la fase de Elaboración, los otros serán desechados. Asimismo, este artefacto, será desechado en la fase de Construcción en la medida que el resultado de las iteraciones vayan desarrollando el producto final.

8) Modelo de Análisis y Diseño

Este modelo establece la realización de los casos de uso en clases y pasando desde una representación en términos de análisis (sin incluir aspectos de implementación) hacia una de diseño (incluyendo una orientación hacia el entorno de implementación), de acuerdo al avance del proyecto.

9) Modelo de Datos

Previendo que la persistencia de la información del sistema será soportada por una base de datos relacional, este modelo describe la representación lógica de los datos persistentes, de acuerdo con el enfoque para modelado relacional de datos. Para expresar este modelo se utiliza un Diagrama de Clases (donde se utiliza un profile UML para Modelado de Datos, para conseguir la representación de tablas, claves, etc.).

10) Modelo de Implementación

Este modelo es una colección de componentes y los subsistemas que los contienen. Estos componentes incluyen: ficheros ejecutables, ficheros de código fuente, y todo otro tipo de ficheros necesarios para la implantación y despliegue del sistema. (Este modelo es sólo una versión preliminar al final de la fase de Elaboración, posteriormente tiene bastante refinamiento).

11) Modelo de Despliegue

Este modelo muestra el despliegue la configuración de tipos de nodos del sistema, en los cuales se hará el despliegue de los componentes.

12) Casos de Prueba

Cada prueba es especificada mediante un documento que establece las condiciones de ejecución, las entradas de la prueba, y los resultados esperados. Estos casos de prueba son aplicados como pruebas de regresión en cada iteración. Cada caso de prueba llevará asociado un procedimiento de prueba con las instrucciones para realizar la prueba, y dependiendo del tipo de prueba dicho

procedimiento podrá ser automatizado mediante un script de prueba.

13) Plan de Iteración

Es un conjunto de actividades y tareas ordenadas temporalmente, con recursos asignados, dependencias entre ellas. Se realiza para cada iteración, y para todas las fases.

14) Matriz de Riesgos

Este documento incluye una lista de los riesgos conocidos y vigentes en el proyecto, ordenados en orden decreciente de importancia y con acciones específicas de contingencia o para su mitigación.

1.2.4 *Evolución del Plan de Desarrollo del Software*

El Plan de Desarrollo del Software se revisará semanalmente y se refinará antes del comienzo de cada iteración.

1.3 Organización del Proyecto

1.3.1 *Participantes en el Proyecto*

Debido a que esta tesis es desarrollada para una institución financiera se incluye el cargo de gerente de proyectos como responsable del proyecto quien proporcionara los requisitos y validar el sistema y al director de tesis como responsable del control y seguimiento.

El presente trabajo de graduación a sido desarrollado por el aspirante a la Maestría en Telemática de la UDA en las fases de Inicio, Elaboración y una iteración de la fase de Construcción, sin embargo considero que se debe hacer mención de los diferentes perfiles que se pueden asignar en proyectos de esta magnitud:

Gerente de proyecto, función a cargo del gerente de proyectos de la institución el perfil establecido es profesional en informática con experiencia en metodologías de desarrollo y notaciones.

Director de tesis, función a cargo del Ing. Pablo Pintado, docente de la Universidad del Azuay y con gran experiencia en metodologías y técnicas de la ingeniería de software quien realiza labores de dirección, control y seguimiento en el desarrollo del presente trabajo de graduación.

Jefe de Proyecto, labor de Juan Diego Landy Andrade, alumno de la Maestría en Telemática de la Universidad del Azuay., con experiencia en notación UML y el proceso de desarrollo RUP y lenguajes de programación.

1.3.2 *Interfaces Externas*

La institución financiera definirá los participantes del proyecto que proporcionarán los requisitos del sistema, y entre ellos quiénes serán los encargados de evaluar los artefactos de acuerdo a cada subsistema y según el plan establecido.

El equipo de desarrollo (Jefe del Proyecto) interactuará activamente con los participantes de la institución financiera para especificación y validación de los artefactos generados.

1.3.3 *Roles y Responsabilidades*

A continuación se describen las principales responsabilidades de cada uno de los puestos en el equipo de desarrollo durante las fases de Inicio y Elaboración, de acuerdo con los roles que desempeñan en RUP.

Puesto	Responsabilidad
Gerente del Proyecto	El jefe de proyecto asigna los recursos, gestiona las prioridades, coordina las interacciones con los clientes y usuarios, y mantiene al equipo del proyecto enfocado en los objetivos. Captura, especificación y validación de requisitos, interactuando con el cliente y los usuarios mediante entrevistas.
Director de Tesis	El director del proyecto también establece un conjunto de prácticas que aseguran la integridad y calidad de los artefactos del proyecto. Además, el director de la tesis se encargará de supervisar el establecimiento de la arquitectura del sistema. Valida los riesgos. Planificación y control del proyecto.
Jefe de Proyecto	Gestión de requisitos, gestión de configuración y cambios, elaboración del modelo de datos, preparación de las pruebas funcionales, elaboración de la documentación. Elaboración del Modelo de Análisis y Diseño. Colaboración en la elaboración de las pruebas funcionales y el modelo de datos. Elabora modelos de implementación y despliegue. Construcción de prototipos. Colaboración en la elaboración de las pruebas funcionales, modelo de datos y en las validaciones con el usuario

Tabla 1.1 Roles de los participantes de este proyecto

1.4 Gestión del Proceso

1.4.1 *Estimaciones del Proyecto*

En el Anexo 1.1 se presenta el análisis financiero del proyecto, que incluye estimaciones de costo del proyecto, rentabilidad, punto de equilibrio.

1.4.2 *Plan del Proyecto*

En esta sección se presenta la organización en fases e iteraciones y el calendario del proyecto. Para este trabajo de graduación solo hemos realizado un ciclo de iteración, por la complejidad de generar una gran cantidad de artefactos y todo este trabajo recae en una sola persona.

1.4.2.1 **Plan de las Fases**

En macro nivel las fases principales en la elaboración del sistema son las siguientes:

1. ***Planeación y elaboración:*** planear, definir los requerimientos, construir prototipos, etc.
2. ***Construcción:*** la creación del sistema.
3. ***Aplicación:*** la transición de la implementación del sistema a su uso.

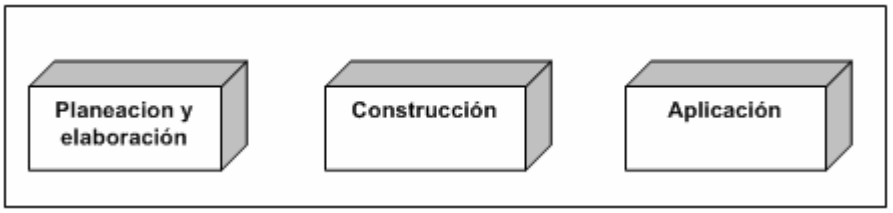


Figura 1.1 Pasos en macro nivel en un proceso de desarrollo RUP

Se plantea realizar un desarrollo iterativo, un ciclo de vida iterativo se basa en el agrandamiento y perfeccionamiento secuencial de un sistema a través de múltiples ciclos de desarrollo de análisis, diseño, implementación y pruebas.

El sistema va creciendo al incorporar nuevas funciones en cada ciclo de desarrollo. Tras la fase preliminar de planeación y especificación, el desarrollo pasa a la fase de construcción a través de una serie de ciclos de desarrollo.

En cada ciclo se aborda un conjunto relativamente pequeño de requerimientos, pasando por el análisis, diseño, construcción y pruebas. La siguiente figura muestra gráficamente como se procede en desarrollo iterativo.

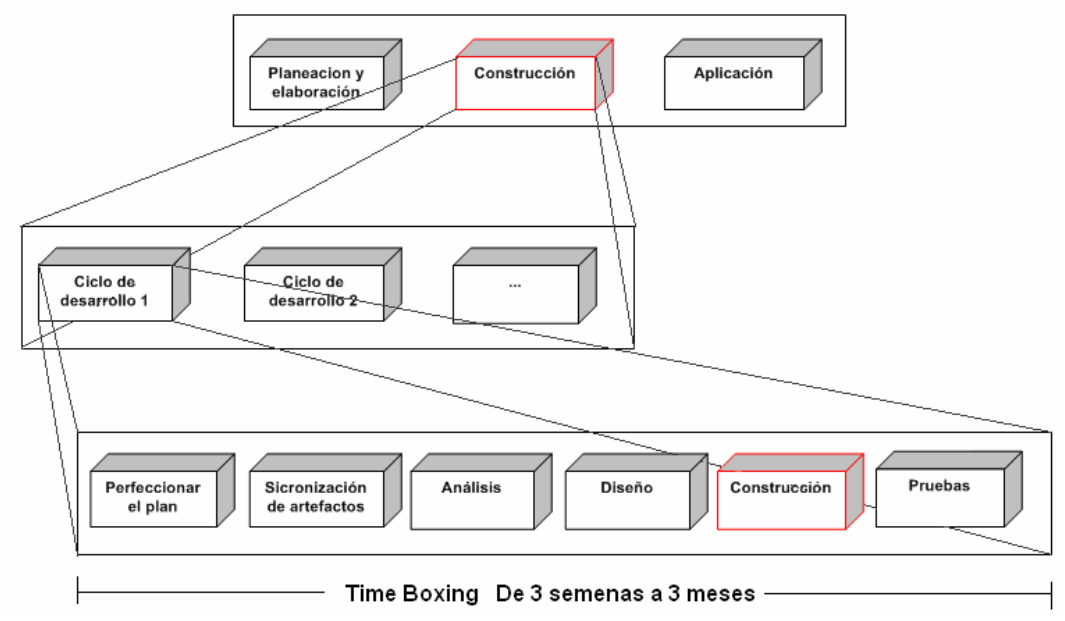


Figura 1.2 Ciclos iterativos de desarrollo

El ***Time boxing*** definido para el desarrollo de este sistema es de 3 meses. Los ciclos iterativos se van organizar a partir de los requerimientos del casos de uso.

Los hitos que marcan el final de cada fase se describen en la siguiente tabla.

FASE	DESCRIPCION	HITOS
INICIO	Su meta principal es lograr el consenso de todos los involucrados Acerca de los objetivos del ciclo de vida del proyecto. Es muy importante especialmente en proyectos nuevos en que existen riesgos significativos en el negocio o la implementación de los Requisitos, y deben ser solucionados para que el proyecto proceda. Para los proyectos que se enfocan en mejorar sistemas existentes, esta fase es más breve, pero aún así centrada en asegurar que el proyecto vale la pena y se puede realizar.	Establecer el ámbito del producto, la identificación de los principales Riesgos y la viabilidad del proyecto.

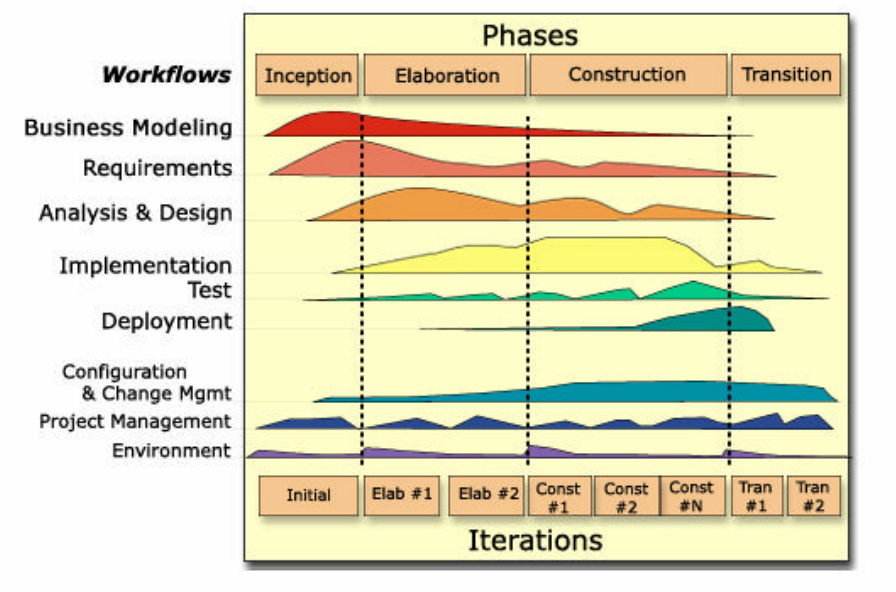
ELABORACION	El propósito de la etapa de Elaboración es crear la línea base de la arquitectura para así disponer de unos cimientos sólidos sobre los que se basará el grueso del esfuerzo de diseño e implementación durante la fase de Construcción. La arquitectura evoluciona de los Requisitos más significativos considerados (aquellos que tienen un fuerte impacto en la arquitectura del sistema) y la evaluación de riesgos. La estabilidad de la arquitectura se evalúa mediante el uso de prototipos de arquitectura.	Obtener una línea base de la arquitectura del sistema, capturar la mayoría de los requisitos y reducir los riesgos principales así como permitir la escalabilidad del equipo del proyecto durante la fase de Construcción
CONSTRUCCION	En la fase de Construcción se deben aclarar los requisitos restantes y completar el desarrollo del sistema basándose en la arquitectura que se ha sido añadida a la línea base. Puede ser vista como un proceso de fabricación donde se hace énfasis en la administración de los recursos y el control de operaciones para optimizar costes, planificaciones y calidad. En este sentido la administración experimenta una transición del desarrollo de Propiedad intelectual durante las fases de Inicio y Elaboración al Desarrollo de productos instalables durante la Construcción y Transición.	Desarrollo del sistema con calidad de producción y prepararse para la entrega al equipo de transición. Toda la funcionalidad debe haber sido implementada y las pruebas para el estado beta de la aplicación completadas. Si el proyecto no logra alcanzar este hito, entonces la transición deberá posponerse una iteración.
TRANSICION	La atención se enfoca en asegurar que el software está disponible para los usuarios finales. Incluye las pruebas del producto como parte de su preparación para ser entregado, y la realización de ajustes menores en respuesta a la retroalimentación recibida de los usuarios. En este punto del ciclo de vida la retroalimentación de los usuarios debe enfocarse fundamentalmente en ajustes específicos y de corto alcance al producto junto a otros temas como configuración, instalación, y usabilidad. Referencias a otros ajustes estructurales mayores debieron haber sido solucionados anteriormente en el ciclo de vida y deberán documentarse para futuras generaciones del software	Consiste en decidir si los objetivos se cumplieron y si debe comenzarse otro ciclo de desarrollo. Es el resultado de la revisión y aceptación por parte del cliente de los artefactos que le han sido entregados.

Tabla 1.2 Fases del proyecto

1.4.2.2 Calendario del Proyecto

A continuación se presenta un calendario de las principales tareas del proyecto incluyendo sólo las fases de Inicio y Elaboración. Como se ha comentado, el proceso iterativo e incremental de RUP está caracterizado por la realización en paralelo de todas las disciplinas de desarrollo a lo largo del proyecto, con lo cual la mayoría de los artefactos son generados muy tempranamente en el proyecto pero

van desarrollándose en mayor o menor grado de acuerdo a la fase e iteración del proyecto. La siguiente figura ilustra este enfoque, en ella lo ensombrecido marca el énfasis de cada disciplina (workflow) en un momento determinado del desarrollo.



Planificación del proyecto

En el Anexo 1.2, se presenta el cronograma de actividades planificadas para el desarrollo de cada una de las fases planeadas para la implementación de este trabajo de graduación. Sin embargo a continuación se presenta por cada fase una plantilla de las actividades posibles a desarrollar y los artefactos que se deben generar como resultado de cada actividad.

Fase de Planeación y Elaboración

Para este proyecto se ha establecido el siguiente cronograma de actividades. La fecha de aprobación indica cuándo el artefacto en cuestión tiene un estado de completitud suficiente para someterse a revisión y aprobación, pero esto no quita la posibilidad de su posterior refinamiento y cambios. A continuación se presenta una plantilla de actividades que se pueden elaborar en una fase de planeación y elaboración, sin embargo existen artefactos que no se han de generar debido a que son opcionales dependiendo de la complejidad del sistema a diseñar. Para el caso de este trabajo de graduación en esta fase no se generan los artefactos Prototipos, Diagramas de casos de uso esenciales expandidos, Modelo conceptual preliminar,

debido que se aplaza su desarrollo para las fases posteriores.

A continuación se definen unas notas importantes acerca del cumplimiento de cada actividad de la fase de planeación y elaboración.

Notas	
Tipo	**Descripción**
A	Constante
B	Opcional
C	Aplazable
D	Orden variable

Tabla 1.3 Notas – Fase de Planeación y Elaboración

FASE DE PLANEACION Y ELABORACION			
Nº	**Actividades**	**Notas**	**Artefactos generados**
1	Definir el plan preliminar		*Plan*, Programa, recursos, presupuesto, etc.
2	Elaborar el informe preliminar de investigación		*Informe preliminar de investigación*, Motivos, alternativas, necesidades de la empresa
3	Definir los requerimientos		*Especificación de requerimientos*, declaración de los requerimientos
4	Registrar los términos en el glosario	a	*Glosario*, Diccionario, restricciones, reglas
5	Implementar el prototipo	b,d	*Prototipo*, sistema de prototipos que ayudan a facilitar la comprensión de problema, los problemas de alto riesgo y los requerimientos.
6	Definir los casos de uso (Alto nivel y esenciales)		*Casos de uso*, descripciones narrativas de los procesos del dominio
			Diagramas de los casos de uso, descripción grafica de todos los casos de uso y sus relaciones
7	Definir el modelo conceptual preliminar	c	*Bosquejo del modelo conceptual*, es preliminar su finalidad es facilitar el conocimiento del vocabulario del dominio, especialmente en su relación

			con los casos de uso y con las especificaciones de los requerimientos.
8	Definir la arquitectura preliminar del sistema	a,c,d	
9	Perfeccionar el plan		

Tabla 1.4 Actividades de la fase de Planeación y Elaboración

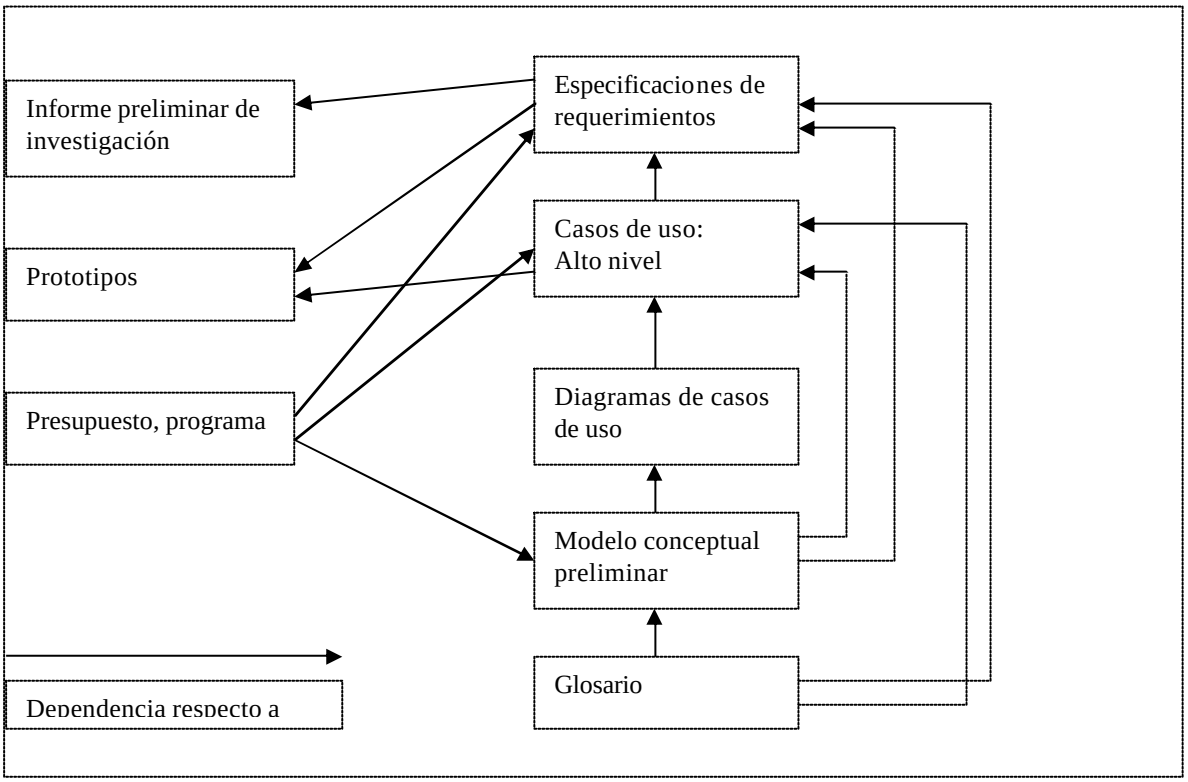


Figura 1.3 Relación entre los artefactos generados en una fase de planeación y elaboración

Fase de construcción – Iteración 1

A continuación se presentan unas notas importantes de las actividades de la fase de análisis.

Notas	
Tipo	**Descripción**
a	Si todavía no se realiza
b	Actual
c	Opcional

Tabla 1.5 Notas – Fase de Análisis

FASE DE CONSTRUCCION - Iteración 1			
Fase de Análisis			
Nº	**Actividades**	**Notas**	**Artefactos generados**
1	Definir los casos de uso esenciales	a	*Casos de uso de alto nivel y esenciales*
2	Perfeccionar los diagramas de casos de uso		*Diagramas de casos de uso*
3	Perfeccionar el modelo conceptual		*Diagramas de estructura estática,* para los conceptos del dominio
4	Perfeccionar el glosario		*Glosario*, Diccionario, restricciones, reglas
5	Definir los diagramas de secuencia de los sistemas		*Diagramas de secuencia del sistema*
6	Definir los contratos de operaciones		*Contratos para operaciones del sistema*
7	Definir los diagramas de estados	c	*Diagramas de estados*, para

			conceptos y casos de uso

Tabla 1.6 Actividades de la Fase de Análisis

Dentro de la misma macro fase de construcción tenemos la fase de diseño en donde se especifican las siguientes actividades y artefactos generados.

Notas	
Tipo	**Descripción**
a	En paralelo con los diagramas de interacción
b	Orden variable

Tabla 1.7 Notas – Fase de Diseño

Fase de Diseño			
Nº	**Actividades**	**Notas**	**Artefactos generados**
1	Definir los casos reales de uso		*Casos de uso reales*
2	Definir los reportes, la interfaz de usuario del usuario.		*Ventanas y reportes*
3	Perfeccionar la arquitectura del sistema	B	*Diagramas de paquete*
4	Definir los diagramas de interacción		*Métodos*
5	Definir los diagramas de clases de diseño	A	*Diagramas de clase y de interfaz*
6	Definir el esquema de la base de datos		*Diagrama de base de datos*

Tabla 1.8 Actividades de la Fase de Diseño

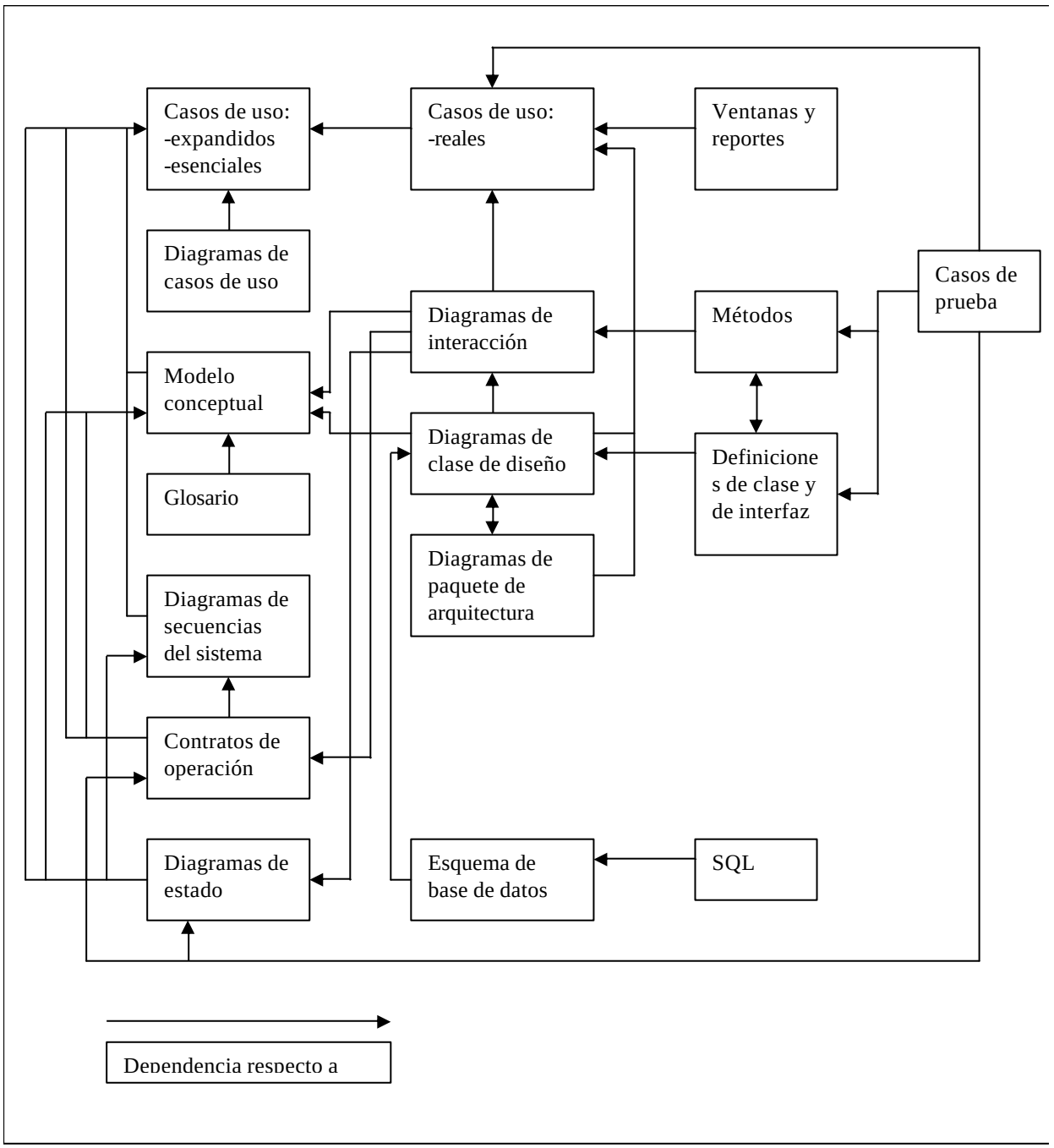


Figura 1.4 Dependencias de los artefactos durante la fase de construcción

1.4.3 *Seguimiento y Control del Proyecto*

Gestión de Requisitos

Los requisitos del sistema son especificados en el artefacto Visión. Cada requisito tendrá una serie de atributos tales como importancia, estado, iteración donde se implementa, etc. Estos atributos permitirán realizar un efectivo seguimiento de cada requisito. Los cambios en los requisitos serán gestionados mediante una Solicitud de Cambio, las cuales serán evaluadas y distribuidas para asegurar la integridad del sistema y el correcto proceso de gestión de configuración y cambios.

Control de Plazos

El calendario del proyecto tendrá un seguimiento y evaluación quincenal por el director de la tesis.

Control de Calidad

Los defectos detectados en las revisiones y formalizados también en una solicitud de cambio tendrán un seguimiento para asegurar la conformidad respecto de la solución de dichas deficiencias Para la revisión de cada artefacto y su correspondiente garantía de calidad se utilizarán las guías de revisión y checklist (listas de verificación) incluidas en RUP.

Gestión de Riesgos

A partir de la fase de Inicio se mantendrá una lista de riesgos asociados al proyecto y de las acciones establecidas como estrategia para mitigarlos o acciones de contingencia. Refiérase al anexo 1.3 para ver la matriz de riesgos elaborada para este proyecto.

1.5 Conclusiones

- Para fabricar un producto de software de alta calidad, que satisfaga las demandas de los usuarios finales con una planificación y un presupuesto

predecibles, es necesario establecer un modelo de proceso que contemple la asignación de tareas y responsabilidades de forma disciplinada dentro de la organización, garantizando con ello que la calidad esté presente en cada etapa del ciclo de vida del proyecto.

- Para ello, se ha apostado por implantar el Proceso Unificado de Rational o RUP (Rational Unified Process), adaptándolo a las necesidades de los clientes en conjunción con una serie de normas y estándares internacionales para el diseño, codificación y pruebas del software.

- El resultado es un proceso de fabricación con un desarrollo incremental por iteraciones dirigidas por el riesgo, y con un control continuo de la calidad. El cambio se acepta de manera natural y es gestionado adecuadamente beneficiando con ello el resultado del proyecto.

- Dentro del proceso unificado de desarrollo de software se denomina artefacto a todo producto o subproducto del proceso de fabricación. Dentro de esto se encuentra desde el propio código fuente hasta la documentación aportada por el cliente y la entregada al completarse cada hito dentro del proyecto.

1.6 Recomendaciones

- Antes del inicio de un proyecto de cualquier tipo, se recomienda hacer el plan de desarrollo de software. Este documento da una visión global a los directivos de la empresa del alcance de la solución, de sus riesgos y responsabilidades, estimaciones de costo y tiempo. En función de estos parámetros podrán tomar la decisión de poner en marcha el proyecto o simplemente descartarlo.

CAPITULO 2. REQUERIMIENTOS DEL SISTEMA

2.1 Introducción

En este capitulo se va a desarrollar el levantamiento de los requerimientos del sistema que son una descripción de las necesidades o deseos del producto de software que se va a generar. El levantamiento de los requisitos del producto se lo hace desde la perspectiva del usuario, los cuales serán establecidos en el artefacto Visión. Los principales casos de uso serán identificados utilizando la especificación de requisitos de software (ERS) según el estándar de la IEEE 830-1998. El objetivo de ERS es plasmar los requisitos del usuario en las plantillas previamente definidas en el estándar y nos ayuda para realizar un refinamiento del plan de desarrollo del proyecto. El objetivo de este capitulo es la aceptación de la institución bancaria (cliente/usuario) del artefacto Visión y el Plan de Desarrollo elaborado en el capítulo 1 que marcan el final de la fase de Inicio dentro de la metodología RUP.

2.2 Visión

2.2.1 *Aspectos generales*

2.2.1.1 Propósito

El propósito de éste documento es recoger, analizar y definir las necesidades de alto nivel y las características del sistema de “Banca Celular usando dispositivos inalámbricos con tecnología SMS”. La visión se centra en la funcionalidad requerida por los participantes del proyecto y los usuarios finales.

Esta funcionalidad se basa principalmente en los servicios de consultas demandados por un cliente en una institución bancaria (Ejemplo Consulta de Saldos) y el envío de alertas y notificaciones que una institución financiera necesita enviar al cliente enmarcados en aspectos de seguridad y confidencialidad de la información.

Los detalles de cómo el sistema cubre los requerimientos se pueden observar en la especificación de los casos de uso y otros documentos adicionales.

2.2.1.2 *Alcance*

El documento Visión se ocupa, como ya se ha apuntado, del sistema de “Banca Celular usando dispositivos inalámbricos con tecnología SMS”. Dicho sistema será desarrollado como trabajo de tesis previa la obtención del título de Magíster en Telemática.

El desarrollo de este sistema se ocupara principalmente del intercambio de mensajes de texto entre la institución bancaria y sus clientes a través de un celular utilizando tecnología SMS - Short Message Service o Servicio de Mensajes Cortos bajo el esquema de servicio basado en operaciones que se explican a continuación:

a) Operación PULL, cuando un usuario inicia una transacción desde su celular.

b) Operación PUSH-Online, cuando un sistema automático genera un evento en línea y envía un mensaje SMS al celular de un usuario. Los eventos a capturar por el sistema son:

Clientes suscritos al servicio de Banca Celular:

- Retiros o avances de efectivo en cajeros automáticos propios.
- Compras en red de POS originadas en red propia.
- Autorizaciones de compras en POS vía Terminal.
- Compras en POS originadas en red VISA.
- Retiros o avances en efectivos en cajeros automáticos originadas en red financiera local, Banred, VISA, MATERCARD

Clientes no suscritos al servicio de Banca Celular:

- Notificación de llegada de una remesa desde el exterior.

c) Operación PUSH-Batch cuando un sistema automático procesa y envía notificaciones o avisos a los celulares pero no son generados por eventos en línea y son alimentados normalmente por archivos.

Clientes no suscritos al servicio de Banca Celular:

- Al corte del estado de cuenta
- El día tope de pago
- Dos días después del día tope de pago
- Antes del siguiente corte
- El día que abona al pago mínimo o total

Adicionalmente este sistema permitirá a los encargados de la institución bancaria controlar todo lo relativo con las suscripciones de los clientes a este servicio, es decir el back office del producto para su respectivo soporte por parte de las áreas de atención al cliente de la institución financiera.

2.2.2 *Posicionamiento*

2.2.2.1 *Oportunidad de Negocio*

La capacidad para cualquier cliente que tenga un teléfono celular y una cuanta bancaria para consultar información financiera y recibir notificaciones y alertas de los productos financieros que están relacionados directa o indirectamente con el cliente usando mensaje de texto (tecnología SMS)

Este sistema permitirá a una institución bancaria informatizar el acceso a los diferentes servicios financieros que brinda (consulta de saldos, consulta de cheques, bloqueo de tarjetas, etc.), lo cual supondrá un acceso rápido y sencillo a la información bancaria, gracias a la tecnología SMS a través de un dispositivo celular. Adicionalmente la institución financiera podrá emitir o enviar mensajes de notificaciones a sus clientes relacionados con avisos importantes y eventos que sucedan sobre sus cuentas u otros servicios (Avisos de retiros de dinero de una

cuenta, notificaciones de vencimiento de créditos, notificación de llegada de una remesa del exterior, etc.) El sistema permite a los ejecutivos del banco a través de la Intranet dar soporte a los clientes para receptar las suscripciones a este servicio que a la final representara un ingreso adicional que se cobrara al cliente por el servicio

2.2.2.2 *Oportunidad de Mercado*

Banca a través de acceso remoto vino para quedarse. Banca por Internet es ya el 40% de las transacciones bancarias en USA.

Debido a la baja penetración del Internet en Sur América vs. la alta penetración de la telefonía celular, la opción lógica y natural es usar la plataforma móvil.

En El Salvador menos del 5% de los consumidores están conectados al Internet

En Ecuador 8 millones de celulares con capacidad de SMS o mas del 70% de los hogares del país.

2.2.2.3 *Sentencia que define el problema*

El problema de:	Consume mucho tiempo para le cliente ir a una agencia del banco a realizar una consulta de información bancaria. Gran cantidad de gente en áreas de atención al cliente. Dificultades experimentadas por los proveedores de productos o servicios de llegar al cliente. No disponer de la información bancaria en el momento preciso. Necesidad de la institución bancaria de generar avisos en línea.
Afecta a:	Operaciones. Áreas de atención al cliente.

	Clientes. Área o departamento de la entidad financiera que requiera notificar eventos a sus clientes como crédito, cartera, tarjetas de crédito, etc. Sistema de alertas bancarias que notifiquen en línea diferentes tipos de avisos después de generase algún tipo de evento como un retiro de cajero automático.
El impacto asociado es:	Dificultades experimentadas por el consumidor para acceder a servicios bancarios básicos y de mucha utilidad. Las notificaciones que se realizan a los clientes por vía tradicional como el teléfono, genera costos demasiado altos (CALL CENTER) y la tasa de éxito en la notificación es baja debido a que en la mayoría de los casos no se puede ubicar al cliente en los lugares de trabajo o domicilio en las horas precisas o simplemente la información de teléfonos fijos esta desactualizada.
Una solución adecuada seria:	Implementar un sistema de Banca Celular usando al dispositivo celular como dispositivo de contacto entre el Banco y el cliente y que se implemente servicios tipo PULL para que el cliente pueda realizar operaciones bancarias sobre sus cuentas en el momento que desee y se implementen servicios Push-Batch para que los funcionarios del banco alerten de manera preventiva acciones a tomar sobre sus cuentas o servicios relacionados con el cliente y servicios Push-Online para que se envíen notificaciones en línea cuando se realice algún evento sobre la cuenta de un cliente. .

Tabla 2.1 Sentencia que define el problema

2.2.2.4 *Sentencia que define la posición del producto*

Para	Cliente Áreas de atención al cliente Operaciones Departamento de crédito Dispositivos electrónicos Departamento que requiera notificar algún evento al celular
Quienes	Acceden a los servicios financieros Dan el soporte del producto al cliente Realizan operaciones de control Notifican eventos crediticios sobre las cuentas del cliente Que generan los eventos automáticos cuando se produce alguna acción sobre una cuenta Notificaciones a los clientes de cierto servicio o producto que se brinde al cliente
El nombre del producto	Banca Móvil
Que	Gestiona toda la información de Banca Celular vía SMS
No como	Call center
Nuestro producto	Permite gestionar las distintas operaciones PULL, PUSH-Batch, Push-Online para que la institución financiera mediante interfaces sencillas y amigables brinde este nuevo servicio a sus clientes. Para no clientes se brindara el servicio de notificaciones. Además proporciona un acceso rápido y actualizado a la información financiera desde cualquier punto geográfico en donde exista cobertura celular y a través de un dispositivo móvil el cliente del banco se beneficie de con la implementación de este nuevo sistema.

Tabla 2.2 Sentencia que define la posición del producto

2.3 ESPECIFICACION DE REQUISITOS DE SOFTWARE – ERS

2.3.1 *Introducción*

Este documento es una Especificación de Requisitos de Software para el "SISTEMA DE BANCA ELECTRONICA USANDO DISPOSITIVOS INALÁMBRICOS CON TECNOLOGÍA SMS". Todo su contenido será elaborado con la colaboración de los usuarios, directivos y desarrolladores de la institución financiera.

2.3.1.1 **Propósito**

El objetivo de la especificación es definir de manera clara y precisa las funcionalidades y condicionantes técnicos del sistema que se desea desarrollar. Este documento va dirigido a los directivos y usuarios finales del sistema, será el medio de comunicación entre las partes involucradas en el desarrollo del sistema de banca móvil.

Esta especificación está sujeta a revisiones por el grupo de usuarios a las diversas versiones que se generen.

2.3.1.2 **Ámbito del sistema**

Este sistema en el futuro se denominara comercialmente como Banca celular.

El objetivo del proyecto es desarrollar un sistema de Banca Celular que permita mediante un dispositivo inalámbrico celular acceder a los servicios financieros de la banca utilizando tecnología SMS.

Permite gestionar las distintas operaciones PULL, PUSH Batch, Push Online para que la institución financiera mediante interfaces sencillas y amigables brinde este nuevo servicio a sus clientes. Para no clientes se brindara el servicio de notificaciones. Además proporciona un acceso rápido y actualizado a la información financiera desde cualquier punto geográfico en donde exista cobertura celular y a través de un dispositivo móvil el cliente del banco se beneficie de con la implementación de este nuevo sistema.

2.3.1.3 Definiciones, acrónimos y abreviaturas

Refiérase a la sección glosario.

2.3.1.4 Visión general del documento ERS

Este documento consta de tres secciones. Esta sección es la Introducción y proporciona una visión general de la ERS. En la Sección 2 se da una descripción general del sistema, con el fin de conocer las principales funciones que debe realizar, los datos asociados y los factores, requisitos técnicos, supuestos y dependencias que afectan al desarrollo, sin entrar en detalles. En la sección 3 se definen detalladamente los requisitos que debe satisfacer el sistema.

2.3.2 *DESCRIPCION GENERAL*

Esta sección presenta una descripción de alto nivel del sistema. Se presentarán las principales áreas de negocio a las cuales el sistema debe dar soporte, las funciones que el sistema debe realizar, la información utilizada, los condicionantes técnicos y otros factores que afecten al desarrollo del mismo.

2.3.2.1 Funciones del sistema

Las funciones han de clasificarse a fin de establecer prioridades entre ellas e identificar las que de lo contrario pasarían inadvertidas (pero que consumen tiempo y otros recursos). Las categorías son:

Categoría de la función	***Significado***
Evidente	Debe realizarse, y el usuario debería saber que se ha realizado
Oculta	Debe realizarse, aunque no es visible para los usuarios. Esto se aplica a

	muchos servicios técnicos subyacentes, como guardar información en un mecanismo persistente de almacenamiento. Las funciones ocultas a menudo se omiten erróneamente durante el proceso de obtención de los requerimientos.
Superflua	Opcionales, su inclusión no repercute significativamente en el costo ni en otras funciones

Tabla 2.3 Categoría de las funciones

El sistema de "Banca Celular usando dispositivos inalámbricos con tecnología SMS" tendrá la siguiente funcionalidad.

Ref #	**Función**	**Categoría**
R1.1	Administración del sistema de Banca Celular	Oculta
R1.2	BackOffice del producto banca celular	Evidente
R1.3	Suscripción al servicio	Evidente
R1.4	Adquiriencia de transacciones tipo PULL	Evidente
R1.5	Envío de notificaciones tipo PUSH-Online con suscripción al servicio	Evidente
R1.6	Envío de notificaciones tipo PUSH-Online sin suscripción al servicio	Evidente
R1.7	Envío de notificaciones de tipo PUSH-Batch con suscripción al servicio	Evidente
R1.8	Envío de notificaciones de tipo PUSH-Batch sin suscripción al servicio	Evidente

Tabla 2.4 **Funciones del sistema banca celular**

2.3.2.2 Atributos del sistema

Los atributos del sistema son sus características o dimensiones: no son funciones.

Algunos atributos del sistema, pueden tener restricciones de frontera del atributo, que son condiciones obligatorias de frontera, generalmente expuestos en un rango numérico de valores.

Atributo	Detalles y restricciones de frontera
Tiempo de respuesta	(Restricción de frontera), Cuando se recepte un petición de una transacción tipo Pull, el sistema de formatear la respuesta a la pasarela en un tiempo no mayor a dos segundos.
Metáfora de interfaz Web	(detalle) Pantallas basadas en aplicación WEB.
Metáfora de interfaz sms	(detalle) Los mensajes sms enviados por el cliente, debe ser lógicos y cortos.
Bitácora de errores	(Restricción de frontera), Se grabe todos los eventos de error en una bitácora en caso de presentarse un error y mapear un error entendible al usuario, esto con el objeto de atender futuros reclamos por parte del cliente de una operación no atendida.
Automático persistente asíncrono	El sistema es alimentado por un generador de eventos asíncrono que dispara o envía mensajes después de que ocurre una acción sobre una cuenta de un cliente.
Automático persistente	Proceso automático persistente que obtiene los mensajes a enviar en intervalos de tiempo determinados desde una base de datos con

	información previamente generada por otros sistemas.
Plataforma del sistema operativo	Windows 2003 Server, XP, 98

Tabla 2.5 **Atributos del sistema banca celular**

Conviene relacionar claramente los atributos del sistema con la funciones. Además los detalles de los atributos y las restricciones de frontera pueden catalogarse como obligatorios u opcionales. En la tabla 2.6 se presenta las funciones del sistema y las relaciones con los atributos.

Ref #	Función	Cat	Atributo	Detalles y restricciones	Cat
R1.1	Administración del sistema de Banca Celular	Oculta	Metáfora de interfaz Web	Pantallas basadas en paginas Web	Obligatorio
R1.2	BackOffice del producto banca celular	Evidente	Metáfora de interfaz sms	Pantallas basadas en paginas Web	Obligatorio
R1.3	Suscripción al servicio	Evidente	Tiempo de respuesta	2 segundos como máximo	Obligatorio Obligatorio
R1.4	Adquiriencia de transacciones tipo PULL	Evidente	Tiempo de respuesta	2 segundos como máximo	Obligatorio Obligatorio
R1.5	Envío de notificaciones tipo PUSH-Online con suscripción al servicio	Evidente	Automático persistente asíncrono		Obligatorio
R1.6	Envío de notificaciones tipo PUSH-Online sin suscripción al	Evidente	Automático persistente asíncrono		Obligatorio

	servicio				
R1.7	Envío de notificaciones de tipo PUSH-Batch con suscripción al servicio	Evidente	Automático persistente	10 minutos	Obligatorio
R1.8	Envío de notificaciones de tipo PUSH-Batch sin suscripción al servicio	Evidente	Automático persistente	10 minutos	Obligatorio

Tabla 2.6 **Relación funciones del sistema con sus atributos**

Existen atributos del sistema que están relacionados con todas las funciones del sistema como por ejemplo la plataforma del sistema operativo y bitácora de errores.

2.3.2.3 Características de los usuarios

Los usuarios y clientes actualmente no disponen de un sistema de banca por celular en la institución bancaria en donde se implementara en producción este trabajo de tesis.

Es necesario capacitar a los usuarios de la institución bancaria en el uso de las interfaces Web, y a los clientes en la manipulación de los mensajes de texto SMS para el ingreso de las operaciones PULL para que puedan acceder a los servicios de consulta de información financiera construidas en el sistema.

2.3.2.4 Condicionantes externos al sistema

El sistema implementará la política de reutilización de componentes existentes para acceso a la información vigente actualmente en la institución bancaria en donde se

implementara el sistema, y también se comprometerá a desarrollar nuevos componentes con sus respectivas interfaces en caso de así requerir el desarrollo de esta tesis. También es necesario dejar abierta la posibilidad de futuros cambios en los modos de trabajo y políticas que impactarán directamente en el sistema.

La institución financiera para el que se desarrolla esta tesis se compromete a implementar las comunicaciones con las operadoras celulares siguientes los estándares de seguridad de la institución bancaria y las operadoras.

El servicio brindado para la entrega de los mensajes a los clientes va a depender de distintos factores:

- Estabilidad del servicio de los componentes facilitados por la institución bancaria para el acceso a la información.

- Estabilidad de las comunicaciones entre las operadoras celulares y la institución bancaria.

- Estabilidad de la pasarela SMS.

- Estabilidad del servicio de mensajería SMS de la operadora celular.

- El servicio de entrega del mensaje va a depender de que el usuario, este con su celular encendido, este bajo la cobertura de servicio de la operadora.

- Es responsabilidad de la operadora en cualquier caso de no entrega del mensaje al usuario celular, el intento reenvío con éxito en un lapso de 48 horas después de la recepción del mensaje por parte de la institución bancaria.

- La pasarela SMS no envía una retroalimentación acerca del estatus de entrega del mensaje al cliente, por lo que un mensaje SMS enviado a la operadora celular, para la institución bancaria es un mensaje entregado con éxito.

Existen condicionantes de software que se utilizara, tales como el SQL Server 2000

para la base de datos, Visual Studio 6.0 y Visual Studio .NET 2005 por tratarse de las herramientas de desarrollo que tiene licenciadas la institución bancaria en donde se implementara este nuevo sistema.

2.3.2.5 Suposiciones y dependencias

Se asume que los requisitos descritos en este documento no van a variar luego de que hayan sido aprobados por los líderes del proyecto.

Para el desarrollo se este sistema existe la dependencia de la contratación por parte de la institución bancaria de una pasarela SMS que normalmente son implementaciones propias de empresas externas que interconectan a las operadores celulares de un país utilizando el protocolo smpp (Short Message peer-to-peer protocol) sobre IP (Internet Protocol) y sirven de intermediario para brindar el servicio de mensajería SMS a terceras instituciones, en este caso a la Banca.

Esta empresa que proporciona la pasarela SMS debe brindar el servicio sujeto a lo que establece la Superintendencia de Comunicaciones del Ecuador en cuento a la regulación de los servicios adicionales de la telefonía móvil.

De cualquier manera, la pasarela de mensajería sólo trata de servir de intermediario y facilitar la labor de desarrollo de las extensiones móviles para una aplicación dada.

La pasarela de mensajería SMS/IP trata de ser como un servidor Web, realiza sus tareas de cambio de formato y ajuste del mensaje, pero precisa de contenidos que realmente le den una utilidad, aunque en este caso los contenidos son pequeñas o grandes aplicaciones que permiten interactuar al usuario móvil con otro programa.

El banco compromete a prestar su infraestructura de hardware para la instalación de la pasarela SMS y el software implementado en este nuevo sistema, así como también incluir en sus bitácoras de operadores y monitoreo el control y soporte de los componentes de software generados para brindar un servicio de calidad.

El banco se compromete a brindar el ambiente de desarrollo adecuado para las pruebas del sistema y su certificación por parte de las áreas involucradas antes del ingreso a producción.

Para que un cliente pueda acceder a todas las operaciones PULL, Push-Batch o Push-Online es necesario que el cliente realice la suscripción al servicio de Banca Celular, para esto es necesario que el usuario sea cliente de la institución financiera o cliente de otras instituciones financieras que sean participantes en la red de servicios financieros que el Banco para el que se desarrolla esta tesis es dueño y administrador. Por lo tanto todo cliente debe tener una tarjeta de débito o una tarjeta de crédito con estatus activa. Por la necesidad del Banco de autentificar a sus clientes, el usuario debe proporcionar al sistema el pin o clave asignada a su tarjeta d débito o crédito.

Dependiendo de la tarjeta con la que haya realizado la suscripción el sistema habilitara los servicios relacionados con el producto tarjeta de débito, y utilizara las cuentas asociadas a la tarjeta para proveer del servicio. Si el cliente se ha suscrito con una tarjeta de crédito el sistema habilitara servicios relacionados con este producto.

Un cliente puede tener sobre un mismo celular únicamente una suscripción con la tarjeta de débito y una suscripción con tarjeta de crédito.

Un cliente puede tener una tarjeta de débito o crédito suscrito a varios celulares.

La gestión de cobro va a ser en línea por transacción, y los valores relacionados con los cobros deben ser parametritos para cada transacción de tipo PULL, PUSH.

Toda la administración del sistema, así como las operaciones Pull, Push-Batch y Push Online deben tener soporte para que sean invocados por varias canales sean estos Intranet, SMS, etc.

2.3.3 *Requisitos específicos*

Esta es la sección más larga e importante del documento por cuanto aquí se especifican detalladamente los requisitos. Esto permite que los diseñadores

realicen un sistema que satisfaga completamente los requisitos de los usuarios y que se planifiquen las pruebas de aceptación para la satisfacción de los directivos y usuarios.

2.3.3.1 Requisitos funcionales

Los requisitos funcionales serán agrupados en casos de uso. El ***caso de uso*** es un documento narrativo que describe la secuencia de eventos de un actor (agente externo) que utiliza un sistema para completar un proceso [Jacobson92]. Un error común en la identificación de los casos de uso consiste en representar los pasos, las operaciones o las transacciones individuales como casos[1].

El ***Actor*** es una entidad externa del sistema que de alguna manera participa en la historia del caso de uso. Generalmente estimula el sistema con eventos de entrada o recibe algo de él. Los actores están representados por el papel que desempeñan en el caso. Conviene escribir su nombre con mayúscula en la narrativa del caso para facilitar su identificación.

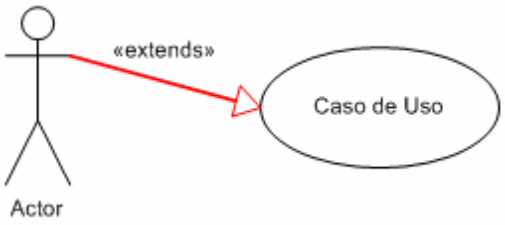


Figura 2.1 Representación grafica de un caso de uso y un actor

Los casos de uso dan una perspectiva del sistema desde el punto de vista de los actores, permiten dar una visión general, los participantes, y el alcance del sistema. La figura 2.2 muestra los casos de uso de los requerimientos del sistema.

[1] Un caso de uso es una descripción de un proceso de principio a fin relativamente amplia, descripción que suele abarcar muchos pasos o transacciones; normalmente no es un paso ni una actividad individual del proceso.

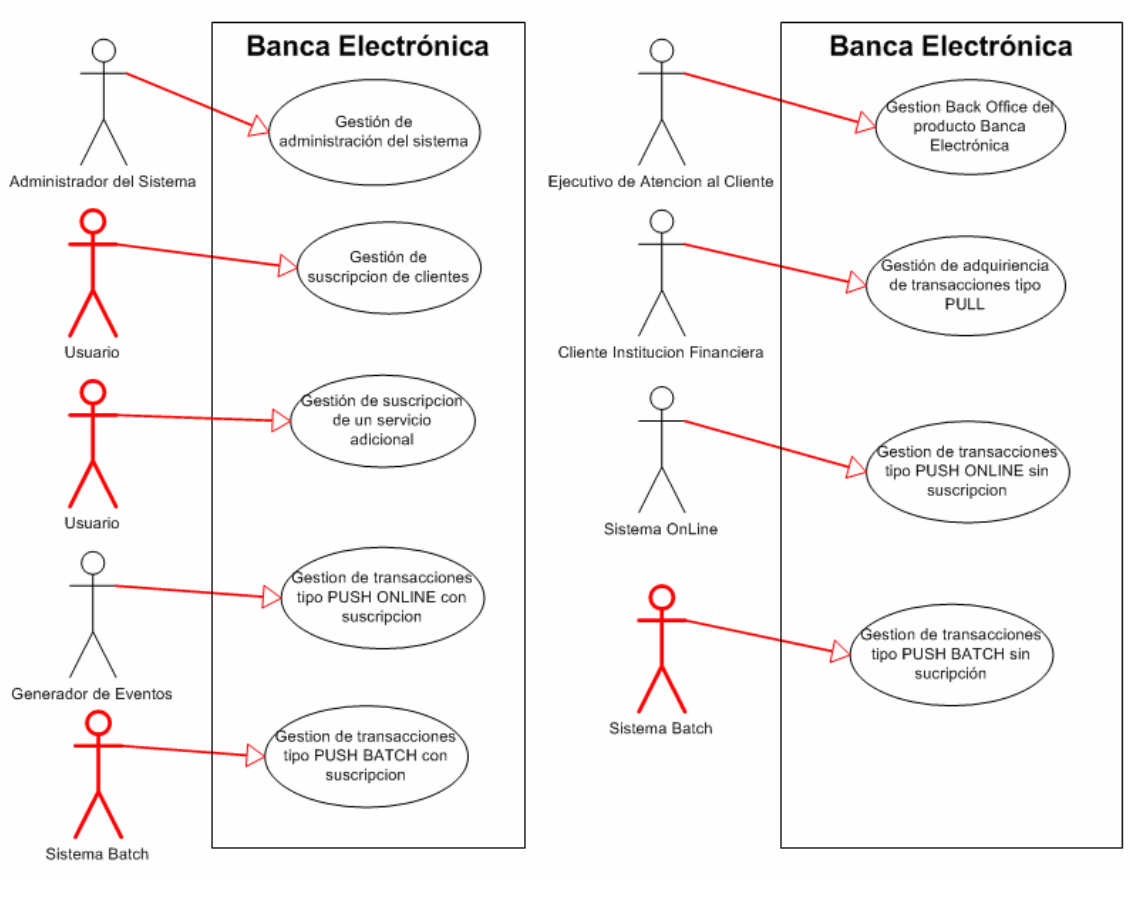


Figura 2.2 Diagrama de casos de uso del sistema de banca celular

2.3.3.2 Descripción de actores

Administrador: Administrador del sistema

Ejecutivo de atención al cliente: Ejecutivo que pertenece al área de servicios bancarios, atención al cliente o atención de reclamos.

Usuario: Puede ser un cliente de la institución financiera o un ejecutivo de atención al cliente.

Cliente de institución financiera: Cliente de la institución financiera participante de la red del servicios financieros del banco para el que se desarrolla la tesis.

Sistema Online: Sistema de la institución financiera que envía una notificación o mensaje a un usuario celular luego de ocurrido un evento específico.

Sistema Batch: Sistema de la institución financiera que envía una notificación o mensaje a un usuario celular, y cuyo proceso es planificado como un proceso batch.

Cliente de suscripción: Puede ser un cliente de la institución financiera que requiere suscribirse al servicio desde un celular o un funcionario de atención al

cliente que ayude a suscribir a un cliente desde la intranet.

2.3.3.3 Descripción de casos de uso

La prioridad de automatización puede ser: opcional, deseable, necesario u obligatorio.

Caso de uso 1	**Gestión de administración del sistema de Banca Celular**
Actor:	Administrador del sistema
***Descripción*:**	Se definen parámetros de configuración para el funcionamiento adecuado del sistema.
Prioridad:	Necesario
REQUISITOS ASOCIADOS	

☐ R.1.1 El sistema gestionara los diferentes mantenimientos de los parámetros de configuración

☐ R.1.2 El sistema permitirá ingresar las operaciones tipo PULL, PUSH-Batch, PUSH-Online, y las operaciones de BackOffice. Para cada transacción se debe ingresar los parámetros de funcionamiento:

- *Tiene costo*
- *Requiere suscripción adicional*
- *Requiere clave*
- *Estatus de activación*
- *Parámetros de valores por comisiones*, para el adquiriente de la transacción y para el autorizador de la información. Estos valores deben ser categorizados para la institución adquiriente de la transacción y para la institución participante. En cada caso los parámetros del cobro dependen del estatus del cliente que puede ser empleado, normal, vip, vip plus.

☐ R.1.3 El sistema permitirá definir los bines de las tarjetas de débito y crédito soportados por el sistema, así como también las transacciones que puede soportar cada bin.

☐ R.1.3 El sistema permitirá definir los bines de las tarjetas de débito y crédito soportados por el sistema, así como también las transacciones que puede soportar cada bin.

Caso de uso 2	**Gestión BackOffice del producto Banca Celular**
Actor:	Ejecutivo de atención al cliente
***Descripción*:**	El actor da todo el soporte de mantenimiento del producto al cliente. El actor debe buscar la suscripción del cliente y posteriormente realizar la operación de mantenimiento solicitada por el cliente.
Prioridad:	Necesario
REQUISITOS ASOCIADOS	

☐ R.2.1 El sistema permitirá el cambio o modificación del teléfono celular de suscripción. ☐ R.2.2 El sistema para cada operación de mantenimiento permitirá buscar la suscripción realizada por un cliente ingresando el número celular, la tarjeta de débito o crédito y el rango de fechas en donde se realizo la suscripción inicial. ☐ R.2.3 El sistema permitirá cambiar le numero de celular de la suscripción. ☐ R.2.4 El sistema permitirá consultar los servicios adicionales al que esta suscrito un cliente ☐ R.2.5 El sistema ofrecerá inactivar una suscripción. ☐ R.2.6 El sistema permitirá inactivar un servicio adicional. ☐ R.2.7 El sistema permitirá el cambio de clave del servicio banca celular. ☐ R.2.8 El sistema permitirá la consulta de clave servicio electrónico Banca Celular. ☐ R.2.9 El sistema permitirá las consultas de transacciones realizadas por un cliente. ☐ R.2.10 El sistema permite realizar el reverso de la suscripción no deseada. ☐ R.2.11 El sistema permitirá realizar el reverso de la suscripción a un servicio adicional no deseado.

Caso de uso 3	**Gestión de suscripción de clientes al producto**
Actor:	Cliente de suscripción
***Descripción*:**	Es una persona que puede ser el cliente de la institución financiera. Si es un cliente de la entidad bancaria realizara la suscripción desde el celular. Si es un ejecutivo de atención al cliente, realizara la suscripción del cliente desde la intranet.
Prioridad:	Obligatoria
REQUISITOS ASOCIADOS	

☐ R.3.1 El cliente de la institución financiera participante en la red de servicios del banco adquiriente puede suscribirse por varios canales, ya sea Intranet o celular, ingresando el numero de la tarjeta de débito o crédito y su clave de la tarjeta. ☐ R.3.2 El sistema deberá autentificar al cliente y celular, verificando el estatus de la tarjeta, verificando la clave. ☐ R.3.2 El sistema siempre debe verificar si la suscripción existe, en caso que no exista lo registra como una suscripción nueva. ☐ R.3.3 En caso de que el cliente haya desactivado la suscripción por algún motivo y quiere volver activarla con el mismo numero de tarjeta y celular, el sistema cambiara simplemente el estatus de la suscripción a activa. ☐ R.3.4 En caso de que el cliente haya desactivado la suscripción por algún motivo y quiere volver activarla, con otro numero de celular, el sistema lo registra como una nueva suscripción. ☐ R.3.5 El sistema en caso de registrar correctamente la suscripción enviara un mensaje de Bienvenida al servicio, caso contrario enviara un mensaje especificando el error cometido al actor.

Caso de uso 4	Gestión de suscripción de operaciones adicionales al producto
Actor:	Cliente de suscripción
***Descripción*:**	Existen servicios u operaciones financieras que no están disponibles únicamente con la suscripción del cliente a banca celular por considerarse de tratamiento delicado. Estas transacciones necesitan de una autorización adicional por parte del cliente, por lo tanto debe solicitar el acceso al servicio deseado vía celular o a través de un funcionario de atención al cliente desde la intranet de la institución financiera. Para poder acceder a este tipo de transacciones el cliente necesita realizar una suscripción adicional al servicio deseado. En este caso de uso esencial se presenta la interacción entre el actor que puede ser el cliente o un funcionario de atención al cliente y el sistema

Prioridad:	Obligatoria
REQUISITOS ASOCIADOS	
☐ R.4.1 El cliente de la institución financiera participante en la red de servicios del banco adquiriente puede suscribir un servicio adicional por varios canales, ya sea Intranet o celular, ingresando el numero de la tarjeta de débito o crédito, su clave de banca celular y el código de operación del servicio adicional deseado. ☐ R.4.2 Verifica si el cliente este suscrito al servicio de banca por celular, si es el caso entonces procede a registrar el servicio adicional. Si el cliente no tiene suscripción, se procede primero a registrar la suscripción al servicio banca por celular y luego se registra el servicio adicional deseado. ☐ R.4.2 El sistema siempre debe verificar si el servicio adicional existe, en caso que no exista lo registra como un registro de servicio adicional nuevo. ☐ R.4.3 En caso de que el cliente haya desactivado la suscripción al servicio adicional por algún motivo y quiere volver activarla con el mismo numero de tarjeta, celular y código de operación, el sistema cambiara simplemente el estatus del registro del servicio adicional. ☐ R.4.4 En caso de que el cliente haya desactivado el servicio adicional algún motivo y quiere volver activarla, con otro numero de celular, el sistema lo registra como un nuevo registro de servicio adicional. ☐ R.4.5 El sistema en caso de registrar correctamente el servicio adicional enviara un mensaje de al cliente al celular indicándole la activación del servicio. Si el registro no tuvo éxito se enviara un mensaje especificando el error al actor.	

Caso de uso 5	**Gestión de adquiriencia de transacciones tipo PULL**
Actor:	Cliente de institución financiera
***Descripción*:**	Cliente de la institución financiera que tiene una tarjeta de débito o una tarjeta de crédito y que iniciar una transacción desde su celular solicitando un servicio financiero.
Prioridad:	Obligatoria

REQUISITOS ASOCIADOS	
R.5.1 Definición de la interfaz para la recepción de las operaciones PULL con la pasarela SMS. R.5.1 El sistema verificara el estatus de la suscripción del cliente al servicio banca celular, si tiene algún problema envía el mensaje de error al celular del usuario que origino la transacción. R.5.2 El sistema consultara por los parámetros de configuración de la transacción solicitada. R.5.3 El sistema verificara, los permisos de acceso de la transacción solicitada de acuerdo al bin de la tarjeta al que tiene asignada la suscripción y si esta activa. R.5.4 El sistema realizara la verificación de la clave, en caso de que este parámetro este configurada en la transacción como obligatorio. R.5.5 En caso de que el cliente este invocando una transacción que requiera suscripción a un servicio adicional el sistema verificara si el cliente esta suscrito al servicio adicional respectivo. R.5.6 El sistema luego revisar todos los controles relacionados con la configuración de la transacción que el cliente esta solicitando debe invocar al objeto que proporciona el servicio. R.5.7 gestión de débito por comisión. R.5.8 Registro en bitácora transaccional de la transacción.	

Caso de uso 6	**Gestión de generación de transacciones tipo PUSH ONLINE con suscripción**
Actor:	Generador de Eventos
***Descripción*:**	Es un sistema que captura diferentes tipos de eventos cuando se producen acciones sobre las cuentas de débito o crédito.
Prioridad:	Necesario
REQUISITOS ASOCIADOS	

☐ R.6.1 Definición de la interfaz para la recepción de las operaciones PUSH-Online con la pasarela SMS. ☐ R.6.2 El sistema verificara el estatus de la suscripción del cliente al servicio banca celular, si tiene algún problema envía el mensaje de error al celular del usuario que origino la transacción. ☐ R.6.3 El sistema consultara por los parámetros de configuración de la transacción solicitada. ☐ R.6.4 El sistema verificara, los permisos de acceso de la transacción solicitada de acuerdo al bin de la tarjeta al que tiene asignada la suscripción y si esta activa. ☐ R.6.5 El sistema realizara la verificación de la clave, en caso de que este parámetro este configurada en la transacción como obligatorio. ☐ R.6.6 En caso de que el cliente este invocando una transacción que requiera suscripción a un servicio adicional el sistema verificara si el cliente esta suscrito al servicio adicional respectivo. ☐ R.6.7 El sistema luego revisar todos los controles relacionados con la configuración de la transacción que el cliente esta solicitando debe invocar a la interfaz que proporcione la pasarela para el envío del mensaje. ☐ R.6.8 gestión de débito por comisión. ☐ R.6.9 Registro en bitácora transaccional de la transacción.

Caso de uso 7	**Gestión de generación de transacciones tipo PUSH ONLINE sin suscripción**
Actor:	Sistema Online
***Descripción*:**	Es un sistema que envía notificaciones de cualquier evento relacionado con el Banco y no necesariamente el cliente al que se envía el mensaje es un cliente de la institución financiera y tampoco se exige que este suscrito al servicio de Banca Celular.
Prioridad:	Necesario

REQUISITOS ASOCIADOS
R.7.1 Definición de la interfaz para la recepción de las operaciones PUSH-Online con la pasarela SMS. R.7.2 El sistema consultara por los parámetros de configuración de la transacción solicitada. R.7.3 El sistema verificara, los permisos de acceso de la transacción solicitada y si esta activa. R.7.4 El sistema luego revisar todos los controles relacionados con la configuración de la transacción que el cliente esta solicitando debe invocar a la interfaz que proporcione la pasarela para el envío del mensaje. R.7.5 Registro en bitácora transaccional de la transacción.

Caso de uso 8	**Gestión de generación de transacciones tipo PUSH BATCH con suscripción**
Actor:	Sistema Batch
***Descripción*:**	Es un sistema que envía requiere enviar notificación a clientes suscritos al servicio de Banca Celular.
Prioridad:	Necesario
REQUISITOS ASOCIADOS	

R.8.1 Definición de la interfaz para la recepción de las operaciones PUSH-Batch con la pasarela SMS. R.8.2 El sistema verificara el estatus de la suscripción del cliente al servicio banca celular, si tiene algún problema envía el mensaje de error al celular del usuario que origino la transacción. R.8.3 El sistema consultara por los parámetros de configuración de la transacción solicitada. R.8.4 El sistema verificara, los permisos de acceso de la transacción solicitada de acuerdo al bin de la tarjeta al que tiene asignada la suscripción y si esta activa. R.8.5 El sistema realizara la verificación de la clave, en caso de que este parámetro este configurada en la transacción como obligatorio. R.8.6 En caso de que el cliente este invocando una transacción que requiera suscripción a un servicio adicional el sistema verificara si el cliente esta suscrito al servicio adicional respectivo. R.8.7 El sistema luego revisar todos los controles relacionados con la configuración de la transacción que el cliente esta solicitando debe invocar a la interfaz que proporcione la pasarela para el envío del mensaje. R.8.8 gestión de débito por comisión. R.8.9 Registro en bitácora transaccional de la transacción.

Caso de uso 9	**Gestión de generación de transacciones tipo PUSH BATCH sin suscripción**
Actor:	Sistema Batch
***Descripción*:**	Es un sistema que envía notificaciones de cualquier evento relacionado con el Banco y no necesariamente el cliente al que se envía el mensaje es un cliente de la institución financiera y tampoco se exige que este suscrito al servicio de Banca Celular.
Prioridad:	Necesario

REQUISITOS ASOCIADOS
R.9.1 Los sistemas del banco que requieran enviar notificaciones a sus clientes al celular deben depositar sus mensajes en un repositorio central, especificando el numero celular, el mensaje a enviar y el tipo de transacción al que pertenece el mensaje. R.9.2 Un sistema automático cada n minutos 24/7 verifica si existen mensajes pendientes de envío en el repositorio central. R.9.3 El sistema automático debe verificar los horarios establecidos de envío para cada tipo de transacción que se encuentre en el repositorio central. R.9.4 Definición de la interfaz para la recepción de las operaciones PUSH-Batch con la pasarela SMS. R.9.5 El sistema consultara por los parámetros de configuración de la transacción solicitada. R.9.6 El sistema verificara, los permisos de acceso de la transacción solicitada y si esta activa. R.9.7 El sistema luego revisar todos los controles relacionados con la configuración de la transacción que el cliente esta solicitando debe invocar a la interfaz que proporcione la pasarela para el envío del mensaje. R.9.8 Registro en bitácora transaccional de la transacción.

2.3.4 *Requisitos de interfaces externas*

En esta sección se describe los requisitos que afecten a la interfase de usuario, interfases con otros sistemas de hardware y software e interfases de comunicaciones.

2.3.4.1 Interfaces de software

El sistema debe proporcionar interfaces de acceso a las operaciones con la pasarela SMS para las operaciones PULL.

La pasarela SMS debe proporcionar la interfase de envío de mensajes para las operaciones PUSH-Batch y Push-Online.

La pasarela SMS debe proporcionar una interfase de administrabilidad de las operaciones PULL, Push-Online y Push-Batch.

El sistema debe comunicarse con los componentes (Objetos COM, Servicios WEB) que proporcione la institución bancaria para el acceso a la información financiera y otra información que se requiera.

2.3.4.2 Interfaces de usuario

El sistema de información deberá ofrecer una interfaz de usuario intuitivo, fácil de aprender y sencillo de manejar. El sistema deberá presentar un alto grado de usabilidad. Lo deseable sería que el usuario se familiarice con el sistema en cuestión de pocas horas.

Las interfaces de usuario de la institución financiera deben ser orientadas al Web.

Estas interfaces deberán ser diseñadas para funcionar para Internet Explorer, por tratarse del navegador más utilizado para la institución financiera para la que desarrolla esta tesis.

2.3.4.3 Interfaces de comunicaciones

Las interfaces de comunicaciones serán implementadas por la institución bancaria en conjunto con las áreas de comunicaciones de los operadores celulares e irán de acuerdo a los requerimientos técnicos de las operadoras y las políticas de acceso y seguridad de la institución bancaria.

2.4 Conclusiones

La ingeniera de requisitos ha tomado gran importancia dentro de un proceso de implementación de un sistema de información y no es un tema a tratar dentro de este presente trabajo de graduación, sin embargo considero que el estándar IEEE 830 es muy amplio, y en este documento se ha tratado de aplicarlo de manera objetiva.

La obtención de casos de uso a pesar de ser una técnica simple requiere de mucha intuición del levantador de requisitos, asumo que la experiencia que vaya acumulando la persona a cargo de esta responsabilidad ayudara a obtener información realmente valedera, sin embargo en las recomendaciones se hace mención a una buena técnica que se aplico en el presente trabajo para la obtención de requisitos.

A continuación se resume los pasos que sugerimos se debe realizar dentro de la fase de planeación y elaboración con metodología RUP.

1. Listar las funciones del sistema, definir la frontera del sistema y luego identificar los actores y casos de uso
2. Es conveniente escribir todos los casos de uso en formato de ***alto nivel***. Clasifíquelos en primarios, secundarios u opcionales.
3. Dibuje un diagrama de casos de uso.
4. Relaciones los casos de uso y dé ejemplo de las relaciones en el diagrama correspondiente.
5. Escriba en el formato esencial expandido los casos de uso más importantes, influyentes y riesgosos, a fin de entender y estimar mejor la naturaleza y las dimensiones del problema. Para evitar análisis complejos posponga la escritura de la forma esencial expandida de los casos de uso menos importantes hasta los ciclos de desarrollo en que serán abordados.

6. En teoría los casos reales de uso se deben posponer hasta una fase de diseño en el ciclo de desarrollo, porque su creación conlleva a decisiones de diseño. Pese a ello a veces es necesario crear ***casos reales de uso*** durante la etapa inicial de requerimientos si:

    a. Las descripciones concretas facilitan notablemente la comprensión.
    b. Los clientes exigen especificar sus procesos en esta forma.

7. Clasifique los casos de uso.

2.5 Recomendaciones

- El éxito de los caso de uso se suele justificar con el hecho de que constituyen una técnica simple e intuitiva, sin embargo es difícil la obtención y especificación de casos de uso verdaderamente útiles. A continuación me permito recomendar una técnica de Cockburn [2] que utiliza el concepto de objetivo (goal) para organizar jerárquicamente los casos de uso. Distingue básicamente entre objetivos estratégicos (los procesos de negocio de la organización) y objetivos de usuario (las funciones del sistema). Los objetivos estratégicos se corresponden con un conjunto de objetivos de usuario y, de igual modo, un objetivo de usuario pude ser descompuesto, a su vez, en un conjunto de objetivos de usuario.

- Estudiar el estándar IEEE 830 de manera profunda antes de iniciar con el levantamiento de requisitos.

- Para verificar que las funciones X que debe tener el sistema sea verdad, la siguiente oración deberá tener sentido:

"El sistema deberá hacer <X>"

[2] Proyecto PROFIT "Definición y Aplicación de un Proceso de Software basado en UML Para el desarrollo de Aplicaciones web" FIT-070000-2000-411

Por ejemplo:

“El sistema deberá hacer la administración del sistema de banca celular”

- Un primer método con que se identifican los casos de uso se basa en los actores.
    1. Se identifican los actores relacionados con un sistema o empresa
    2. En cada actor, se identifican los procesos que inician o en que participan.

- Un segundo método de identificación de los casos de uso de basa en eventos.
    1. Se identifican los eventos externos a los que un sistema ha de responder.
    2. Se relacionan los eventos con los actores y casos de uso.

CAPITULO 3. ANÁLISIS ORIENTADO A OBJETOS

3.1 Introducción

La fase de planeación y elaboración ha concluido, los casos de uso han sido identificados, clasificados y programados. Sin embargo pueden existir ajustes en esta fase según como se avance en desarrollo de este trabajo de graduación. Estamos iniciando una transacción muy importante que es la fase de construcción en la cual se cumplen los ciclos de desarrollo iterativo. Las actividades iniciales de este ciclo están relacionadas con la administración del proyecto o puede ir en paralelo en donde se sincroniza la documentación generada en la fase de planeación y elaboración. Ahora si pasamos a la fase de análisis en el cual se investigan a fondo los problemas que involucra la construcción de un sistema de Banca Celular. Iniciamos por construir casos de uso esenciales expandidos para tener para alcanzar un conocimiento mas profundo de los procesos y de los requerimientos. Esto nos permite genera un modelo conceptual mucho mas preciso que nos facilita a conocer el dominio: sus conceptos, su terminología y sus relaciones, así mismo todos estos elementos que vamos generando están siendo registrados en paralelo en el documento glosario.

El comportamiento del sistema es una descripción de lo que hace, sin explicar la manera en que lo hace. Los diagramas de secuencia son notaciones de UML que ayudan mucho a analizar este comportamiento, en conjunto con los contratos de operaciones y los diagramas de estados

3.2 Caso expandido de usos: Esenciales

Un *caso de uso expandido* muestra mas detalles que uno de alto nivel: este tipo de casos son útiles para alcanzar un conocimiento mas profundo de los proceso y de los requerimientos.

Los *casos esenciales de uso* [Constantine97] son casos expandidos que se expresan en una forma teórica que contiene poca tecnología y pocos detalles de implementación; las decisiones de diseño se posponen y se abstraen de la realidad, especialmente las concernientes a la interfaz para el usuario. Un caso de este tipo describe el proceso a partir de sus actividades y motivos esenciales.

3.2.1 *Gestión de administración del sistema de Banca Celular*

Caso de uso:	**Gestión de administración del sistema de Banca Celular**
Actores:	Ejecutivo de atención al cliente
Propósito:	Realizar el soporte de administración del sistema Banca Celular
Resumen:	El administrador del sistema es la persona encargada de mantener los parámetros de configuración actualizados, definir las nuevas transacciones y grupos a los que pertenecen
Tipo:	Primario y esencial
Referencias cruzadas	*Todas Funciones*
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1.- Este caso de uso comienza cuando el administrador del sistema desea configurar los parámetros necesarios para definir cada una de las transacciones del sistema de Banca Celular.	
	2.- El sistema de Banca Celular debe realizar el mantenimiento de parámetros como servicio soportado, red adquiriente, Bines,
	Servicio, código de servicio, descripción
	Grupo de transacción, código, descripción

	Red adquiriente, código de red, descripción
	Bines, numero bin, descripción, servicio al que pertenece, tipo de producto, si el bin es local o es un sufijo que pertenece a otra institución, grupo asignado al bin
	Tipo de Operación, PULL, PUSH Online, PUSH Batch con suscripción y sin suscripción
	Operaciones soportadas, código de operación, descripción, tipo de operación al que pertenece
	Transacciones, código transacción, código de operación, código de servicio, red adquiriente, descripción. Tiene costo, tiene o requiere de autorización adicional, tiene clave, costos de red adquiriente y red autorizadora por tipo de cliente de suscripción, numero de tarjeta especial asociada a la transacción para transacciones sin suscripción, para transacciones de tipo push-batch sin suscripción días de semana de notificación, horario inicio y horario fin y fecha de inicio de notificaciones
	Asignación de transacciones a grupos
	Errores, código de error, mensaje a enviar al usuario, mensaje del sistema
	Estatus de transacción, código, descripción
15. El administrador del sistema recibe respuesta del sistema banca celular	
Cursos alternos	
Línea 2:	Si no puede actualizar registro. Indicar error

3.2.2 *Gestión BackOffice del producto Banca Celular*

Caso de uso:	**BackOffice**
Actores:	Ejecutivo de atención al cliente
Propósito:	Realizar el soporte al cliente del servicio Banca Celular
Resumen:	El ejecutivo de atención al cliente de la institución financiera, necesita una herramienta que le permita gestionar el soporte y atención de reclamos al cliente, así como operaciones básicas para poder mantener los datos actualizados de las suscripciones de los clientes, como por ejemplo sus números celulares
Tipo:	Primario y esencial
Referencias cruzadas	*Funciones* R1.1, R1.2, R1.3
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1.- Este caso de uso comienza cuando el ejecutivo de atención al cliente solicita una operación de servicio bancario a través de un sistema BackOffice.	
a. Si el ejecutivo solicita una operación de suscripción al servicio, véase el caso de uso *Suscripción del cliente*	
b. Si el ejecutivo solicita una operación de cambio del numero celular, véase la sección *Modifica teléfono de suscripción*	
c. Si el ejecutivo requiere una consulta de suscripciones, véase la sección *consulta de suscripciones*	

d. Si el ejecutivo requiere una consulta de servicios adicionales de una suscripción, véase la sección *consulta de servicios adicionales*	
e. Si el ejecutivo solicita una operación de inactivación del servicio banca celular, véase el caso de uso *Cancela suscripción principal y servicios adicionales*	
f. Si el ejecutivo solicita una operación de inactivación de un servicio adicional banca celular, véase el caso de uso *Cancela servicio adicional*	
g. Si el ejecutivo solicita un cambio de clave de servicio electrónico banca celular, véase el caso de uso *Transacción PULL* sección *Operación cambio de clave*	
h. Si el ejecutivo solicita una consulta de transacciones realizadas por un celular, véase la sección *Consulta de transacciones*	
i. Si el ejecutivo solicita la consulta de clave del servicio electrónico banca celular, véase sección *Consulta de clave*	
j. Si el ejecutivo realizo una suscripción errónea de un cliente puede solicitar la operación de reverso del servicio banca celular, siempre y cuando el reverso sea en el mismo día de suscripción. Si este es el caso véase la sección *Reverso de Suscripción*	

k. Si el ejecutivo realizo una suscripción a un servicio adicional de forma errónea, puede solicitar la operación de reverso de un servicio adicional banca celular, siempre y cuando el reverso sea en el mismo día de suscripción del servicio adicional. Si este es el caso véase la sección *Reverso de suscripción de un servicio adicional*	
	2.- El sistema de Banca Celular debe verificar que los campos genéricos para cada operación BackOffice lleguen correctamente desde la pasarela.
	3.- El sistema de Banca Celular debe verificar que la tarjeta de debito/crédito relacionada con la suscripción del cliente este registrada como un producto del banco
	4.- El sistema debe determinar el tipo de producto al que pertenece el BIN de la tarjeta ya sea Debito, Visa o MasterCard y adicionalmente se debe determinar si el es local o es un producto de otra institución asociada al banco como por ejemplo una cooperativa.
	5.- Debe verificar si la operación solicitada por el cliente esta asociada al BIN de la tarjeta, si la operación esta registrada para su utilización desde la red adquiriente y si esta activa. Si esta autentificación es correcta obtiene los parámetros de configuración de la transacción
	6.- Verifica si el cliente que solicita la transacción tiene o esta suscrito al servicio de banca celular y que la suscripción esta activa.

	7.- Si la transacción que solicita requiere de clave debe verificar que la clave del servicio haya sido ingresada por el cliente y este correcta
	8.- Si la transacción que solicita requiere de una suscripción adicional debe verificar que la operación que solicita el cliente este suscrita y que este activa la suscripción
	9.- Se consulta por el estado de la tarjeta de debito/crédito de suscripción al servicio. La tarjeta debe estar activa.
	10.- Graba una bitácora transaccional de la operación solicitada con estatus de operación bancaria en proceso
	11.- Ejecuta la operación BackOffice relacionada con el servicio solicitado
	a. Si el ejecutivo solicita una operación de suscripción al servicio, véase el caso de uso *Suscripción del cliente*
	b. Si el ejecutivo solicita una operación de cambio del número celular, véase la sección *Modifica teléfono de suscripción*
	c. Si el ejecutivo requiere una consulta de suscripciones, véase la sección *consulta de suscripciones*
	d. Si el ejecutivo requiere una consulta de servicios adicionales de una suscripción, véase la sección *consulta de servicios adicionales*
	e. Si el ejecutivo solicita una operación de suscripción de un servicio adicional, véase el caso de uso *Suscripción adicional*
	f. Si el ejecutivo solicita una operación de inactivación del servicio banca celular, véase el caso de uso *inactivación de Suscripción*

	g. Si el ejecutivo solicita una operación de inactivación de un servicio adicional banca celular, véase el caso de uso *inactivación de Suscripción de servicio adicional*
	h. Si el ejecutivo solicita un cambio de clave de servicio electrónico banca celular, véase el caso de uso *Transacción PULL* sección *Operación cambio de clave*
	i. Si el ejecutivo solicita una consulta de transacciones realizadas por un celular, véase la sección *Consulta de transacciones*
	j. Si el ejecutivo solicita la consulta de clave del servicio electrónico banca celular, véase sección *Consulta de clave*
	k. Si el ejecutivo realizo una suscripción errónea de un cliente puede solicitar la operación de reverso del servicio banca celular, siempre y cuando el reverso sea en el mismo día de suscripción. Si este es el caso véase la sección *Reverso de Suscripción*
	l. Si el ejecutivo realizo una suscripción a un servicio adicional de forma errónea, puede solicitar la operación de reverso de un servicio adicional banca celular, siempre y cuando el reverso sea en el mismo día de suscripción del servicio adicional. Si este es el caso véase la sección *Reverso de suscripción de un servicio adicional*
	12.- Verifica si la transacción tiene costo y realiza el cobro en línea
	a. Si el cliente esta suscrito con la tarjeta de débito, véase la sección *Operación pago con tarjeta de débito*
	b. Si el cliente esta suscrito con la tarjeta de crédito, véase la sección *Operación pago con tarjeta de crédito*

	13. Formatea el mensaje de salida a la pasarela dependiendo de la operación y al sistema BackOffice del ejecutivo de atención al cliente
	14. Actualiza la bitácora transaccional.
15. El usuario recibe en su celular un mensaje sms con la información solicitada o la confirmación de la operación bancaria solicitada.	
16. El ejecutivo de atención al cliente recibe el resultado de la operación de BackOffice solicitada	
Cursos alternos	
Línea 2:	Introducción de un mensaje incorrecto, Indicar error
Línea 3:	Bin de tarjeta no pertenece a ningún producto del Banco. Indicar error
Línea 5:	Si la operación solicitada no esta relacionada al bin de la tarjeta o no esta activa. Indicar error.
Línea 6:	Si la suscripción del cliente no existe o no esta activa. Indicar error.
Línea 7:	Si la clave es ingresada es incorrecta. Indicar error
Línea 8:	Si la operación solicitada por el cliente requiere de suscripción adicional y si el cliente no tiene esta suscripción. Indicar error
Línea 9:	Si el estado de la tarjeta de debito/crédito esta cancelada. Debe indicar error
Línea 11:	Si la operación bancaria solicitada no puede realizar el servicio. Indicar error
Línea 12:	Si no puede realizar el cobro por falta de fondos. Indicar error

Línea 14:	Si no puede actualizar. Indicar error

Sección: Modifica teléfono de suscripción	
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1.- El ejecutivo de atención al cliente especifica el número de tarjeta de suscripción, el número de celular actual, y el nuevo numero celular.	
	...
	11.- Graba la suscripción actual en una bitácora auditora y crea un nuevo registro de suscripción con el nuevo número de teléfono, cancela la suscripción anterior por cambio de número telefónico.
	…
15.- El usuario recibe un mensaje de notificación del cambio del número en su celular en caso de que la operación se haya realizado de manera exitosa.	
16. El sistema de BackOffice indica al ejecutivo de atención al cliente el resultado de la operación	
Cursos alternos	
Línea 11:	Si no registra correctamente el cambio del celular. Debe indicar el motivo

Caso de uso:	**BACK OFFICE**
Sección: Consulta de suscripciones	
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1.- El ejecutivo de atención al cliente especifica el bin o número de tarjeta de suscripción, el número de celular - opcional, la fecha inicial y final de suscripción.	
	…
	11.- El sistema consulta las suscripciones en el registro y devuelve las coincidencias.
	…
16. El sistema de BackOffice muestra al ejecutivo de atención al cliente el resultado de la operación	
Cursos alternos	
Línea 11:	Si no existen suscripciones registradas envía la respuesta respectiva

Caso de uso:	**BACK OFFICE**
Sección: Consulta de servicios adicionales de una suscripción	
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1.- El ejecutivo de atención al cliente especifica el número de tarjeta de suscripción y el número de celular.	
	…
	11.- El sistema consulta los servicios adicionales relacionados con la suscripción principal y devuelve las coincidencias.

	…
16. El sistema de BackOffice muestra al ejecutivo de atención al cliente el resultado de la operación	
Cursos alternos	
Línea 11:	Si no existen servicios adicionales relacionados con la suscripción envía la respuesta respectiva

Caso de uso:	**BACK OFFICE**
Sección: Cancela suscripción principal y servicios adicionales	
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1.- El ejecutivo de atención al cliente especifica el número de tarjeta de suscripción, numero de celular y clave de servicio banca celular.	
	…
	11.- El sistema verifica si el cliente tiene suscripción banca celular y servicios adicionales activos.
	12.- En caso de que el cliente tiene valores pendientes de cobro y procede al debito de la cuenta o el envío de la transacción de autorización de crédito
	13.- Cambia el estatus de cancelación de la suscripción.
	…
16. El sistema de BackOffice muestra al ejecutivo de atención al cliente el resultado de la operación	

Cursos alternos	
Línea 11:	Si no tiene suscripción activa o servicios adicionales activo. Indicar error
Línea 12:	Si no puede cobrar. Indicar al cliente que deposite dinero para proceder al debito
Línea 13:	Si no puede realizar el registro. Indicar error

Caso de uso:	**BACK OFFICE**
Sección: Cancela servicio adicional	
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1.- El ejecutivo de atención al cliente específica el número de tarjeta de suscripción, número de celular, clave de servicio banca celular y código de servicio adicional a cancelar.	
	...
	11.- El sistema verifica si el cliente tiene suscripción banca celular y servicios adicionales activos.
	12.- En caso de que el cliente tiene valores pendientes de cobro y procede al debito de la cuenta o el envío de la transacción de autorización de crédito
	13.- Cambia el estatus de cancelación del servicio adicional.
	…
16. El sistema de BackOffice muestra al ejecutivo de atención al cliente el resultado de la operación	

Cursos alternos	
Línea 11:	Si no tiene suscripción activa o servicios adicionales activo. Indicar error
Línea 12:	Si no puede cobrar. Indicar al cliente que deposite dinero para proceder al debito
Línea 13:	Si no puede realizar el registro. Indicar error

Caso de uso:	**BACK OFFICE**
Sección: Consulta de transacciones	
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1.- El ejecutivo de atención al cliente específica el número de tarjeta de suscripción, numero de celular, fecha inicial, fecha fina, numero de cuenta, tipo cuenta, estatus de transacción.	
	…
	11.- El sistema consulta en el registro transaccional del sistema banca celular.
	…
16. El sistema de BackOffice muestra al ejecutivo de atención al cliente el resultado de la operación	
Cursos alternos	
Línea 11:	Si no existen transacciones. Indicar error
Línea 11:	Si no puede consultar en el registro. Indicar motivo

Caso de uso:	**BACK OFFICE**

Sección: Consulta de clave de servicio banca celular	
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1.- El ejecutivo de atención al cliente especifica el número de tarjeta de suscripción, numero de celular, código del supervisor.	
	…
	11.- El sistema consulta en el registro la clave actual de suscripción del sistema banca celular.
	…
16. El sistema de BackOffice muestra al ejecutivo de atención al cliente el resultado de la operación	
Cursos alternos	
Línea 11:	Si no puede consultar la clave en el registro. Indicar motivo

Caso de uso:	**BACK OFFICE**
Sección: Reverso de suscripción principal y servicios adicionales	
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1.- El ejecutivo de atención al cliente especifica el número de tarjeta de suscripción, numero de celular y clave de servicio banca celular, usuario supervisor.	
	…

	11.- El sistema verifica si el cliente tiene suscripción banca celular y servicios adicionales activos y realizados en el mismo día del reverso.
	12.- Verifica si ha ingresado transacciones tipo PULL desde el celular suscrito. Si esto es así ya no procede el reverso, el ejecutivo de atención al cliente debe utilizar la opción de cancelación de suscripción
	13.- Se procede al reverso de los valores en caso de se haya cobrado al momento de la suscripción
	13.- Cambia al estatus de reversado la suscripción y servicios adicionales.
	…
16. El sistema de BackOffice muestra al ejecutivo de atención al cliente el resultado de la operación	
Cursos alternos	
Línea 11:	Los reversos solo pueden aceptarse en el mismo día de suscripción. Caso contrario el cliente debe solicitar la cancelación de la suscripción
Línea 13:	Si no puede realizar el reverso del cobro. Indicar el motivo
Línea 13:	Si no puede realizar el registro. Indicar error

Caso de uso:	**BACK OFFICE**
Sección: Reverso de un servicio adicional	
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**

1.- El ejecutivo de atención al cliente especifica el numero de tarjeta de suscripción, numero de celular, clave de servicio banca celular y usuario supervisor, código del servicio adicional a reversa.	
	...
	11.- El sistema verifica si el cliente tiene servicios adicionales activos y realizados en el mismo día del reverso.
	13.- Se procede al reverso de los valores en caso de se haya cobrado al momento de la suscripción del servicio adicional
	13.- Cambia al estatus de reversado la suscripción del servicio adicional.
	…
16. El sistema de BackOffice muestra al ejecutivo de atención al cliente el resultado de la operación	
Cursos alternos	
Línea 11:	Los reversos solo pueden aceptarse en el mismo día de suscripción del servicio adicional. Caso contrario el cliente debe solicitar la cancelación del servicio adicional
Línea 13:	Si no puede realizar el reverso del cobro. Indicar el motivo
Línea 13:	Si no puede realizar el registro. Indicar error

3.2.3 *Caso de uso gestión de suscripción de clientes.*

Caso de uso:	**Suscripción del cliente**
Actores:	Usuario

Propósito:	Registrar una suscripción al servicio Banca Celular de un cliente a desde su celular.
Resumen:	El cliente envía un mensaje al número asignado a la institución por ejemplo el 2626 solicitando una suscripción al servicio. Debe ingresar el número de su tarjeta de débito/crédito y su clave. El sistema autentifica al cliente, registra la suscripción y envía un mensaje de bienvenida al servicio
Tipo:	Primario y esencial
Referencias cruzadas	*Funciones* R1.1, R1.2, R1.3
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1.- Este caso de uso comienza cuando el cliente de la institución financiera desea suscribirse al servicio de banca celular por celular	
2.- El cliente de suscripción puede registrar al servicio enviando un mensaje sms desde el celular del cliente al número contratado por la institución para el servicio de banca celular por celular o bien el funcionario de atención al cliente a través de la intranet puede suscribir al cliente al servicio. Para cualquier mecanismo de suscripción el cliente de la institución bancaria debe indicar el numero de su tarjeta de debito/crédito y su clave	
	3.- El sistema de Banca Celular debe verificar que los parámetros ingresados por el cliente: operación para suscripción, tarjeta de débito/crédito y clave sean correctos.

	4.- El sistema de Banca Celular debe verificar que la tarjeta de debito/crédito sea una tarjeta propia de la institución financiera
	5.- Si el Bin de la tarjeta debito/crédito pertenece al banco debe determinar el tipo de producto al que pertenece Debito, Visa, MasterCard
	6.- Debe verificar si la operación esta asociada al BIN de la tarjeta, si esta registrada su utilización desde la red adquiriente y si esta activa. Si esta autentificación es correcta obtiene los parámetros de configuración de la transacción
	7.- Verifica si la tarjeta de débito/crédito existe y si esta con el estatus de activa
	8.- Debe verificar si la clave de la tarjeta ingresada es correcta
	9.- Verifica si el cliente ya tiene registrada una suscripción anterior con la misma tarjeta y celular, si no existe se vuelve a crear un nuevo registro de suscripción. Si ya existe y esta inactiva únicamente se la vuelve a activar. Si existe y esta reversada borra el registro de esta suscripción reversada y se vuelve a crear un nuevo registro.
	10.- Graba una bitácora transaccional
	11.- En caso de que la transacción esta configurada para el cobro por la suscripción en línea, el sistema procede a debitar el valor de la cuenta del cliente asociada a la tarjeta de debito. Si es una tarjeta de crédito envía o dispara una transacción de compra como si fuera de un pos virtual.
	12.- Actualiza la bitácora transaccional como OK o el error presentado

	13.- En caso de que el registro de la suscripción sea OK y la suscripción venga de otras redes de adquiriencia que no sea la red celular o vía SMS, se procede a notificar el registro de suscripción a la pasarela SMS, para que suscriba al cliente como parte de su control.
	14.- Se formatea un mensaje de respuesta OK de la suscripción. Si la suscripción es realizada desde la Intranet presenta un mensaje al funcionario de atención al cliente y se envía un mensaje SMS al cliente de bienvenida al servicio de Banca celular por celular. Si la suscripción fue realizada desde el celular, simplemente muestra el mensaje de bienvenida al servicio.
15. Si el cliente de suscripción origino la transacción desde su celular recibe un mensaje de bienvenida al servicio de banca celular. Esto indica que el proceso de suscripción es OK y puede empezar a utilizar ya el servicio. Si el cliente de suscripción es el ejecutivo de atención al cliente, recibe el mensaje de aceptación de suscripción del cliente y debe indicarle al cliente del servicio que le va a llegar un mensaje de bienvenida al servicio. Esto indicara que el servicio ya esta activo en su celular.	
Cursos alternos	

Línea 3:	Introducción de un neumónico de operación invalido. Indicar error
Línea 3:	Introducción de un número de tarjeta no numérico y tamaño incorrecto. Indicar error
Línea 3:	Introducción de una clave de tarjeta no numérica y tamaño incorrecto. Indicar error
Línea 4:	Introducción de una tarjeta debito/crédito que no pertenece la institución financiera. Indicar error
Línea 6:	Si la transacción no esta autorizada para la red adquiriente, no existe o no esta activa. Indicar error
Línea 7:	Si la tarjeta no existe o tiene estatus de no activa. Indicar error
Línea 8:	Si la clave es errónea. Indicar error
Línea 9:	En caso de no poder hacerlo. Indicar error
Línea 11:	Si no puede debitar, no realiza el registro de la suscripción del cliente. Indicar error por falta de fondos
Línea 13:	En caso de que se presente un error al reportar la suscripción a la pasarela. Se procede a reversar la suscripción y devolver los fondos al cliente en caso de haberse cobrado. Indicar error

3.2.4 *Caso de uso gestión de suscripción de un servicio adicional.*

Caso de uso:	**Suscripción de un servicio adicional**
Actores:	Usuario
Propósito:	Registrar una suscripción de un servicio adicional de Banca Celular de un cliente a desde su celular o por un funcionario de atención al cliente desde la intranet.

Resumen:	Existen servicios u operaciones financieras que no están disponibles únicamente con la suscripción del cliente a banca celular por considerarse de tratamiento delicado. Estas transacciones necesitan de una autorización adicional por parte del cliente, por lo tanto debe solicitar el acceso al servicio deseado vía celular o a través de un funcionario de atención al cliente desde la intranet de la institución financiera. Para poder acceder a este tipo de transacciones el cliente necesita realizar una suscripción adicional al servicio deseado. En este caso de uso esencial se presenta la interacción entre el actor que puede ser el cliente o un funcionario de atención al cliente y el sistema
Tipo:	Primario y esencial
Referencias cruzadas	*Funciones* R1.1, R1.2, R1.3
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1.- Este caso de uso comienza cuando el cliente de la institución financiera desea suscribirse a un servicio adicional de banca celular por celular	

2.- El cliente de suscripción puede registrarse al servicio adicional enviando un mensaje sms desde el celular del cliente al número contratado por la institución para el servicio de banca celular por celular o bien el funcionario de atención al cliente a través de la intranet puede suscribir el servicio adicional. Para cualquier mecanismo de suscripción el cliente de la institución bancaria debe indicar el número de su tarjeta de débito/crédito, clave de banca celular y el servicio adicional deseado.	
a. Si el cliente a solicitado el servicio de *Consulta de cheque cobrado* debe ingresar ADI*6 últimos números de tarjeta*clave actual*AVCHC	
b. Si el cliente a solicitado el servicio de *Notificación de consumos de tarjetas crédito* debe ingresar ADI*6 últimos números de tarjeta*clave actual*NOCTC	
c. Si el cliente a solicitado el servicio de *Notificación de retiros o avances en efectivo en cajeros automáticos de tarjeta débito/crédito* ADI*6 últimos números de tarjeta*clave actual*NOATM	
	3.- El sistema de Banca Celular debe verificar que los parámetros ingresados por el cliente: operación para suscripción de servicio adicional, tarjeta de debito/crédito, clave y servicio adicional deseado sean correctos.

	4.- El sistema de Banca Celular debe verificar que la tarjeta de debito/crédito sea una tarjeta propia de la institución financiera
	5.- Si el Bin de la tarjeta debito/crédito pertenece al banco debe determinar el tipo de producto al que pertenece Debito, Visa, MasterCard
	6.- Debe verificar si la operación esta asociada al BIN de la tarjeta, si esta registrada su utilización desde la red adquiriente y si esta activa. Si esta autentificación es correcta obtiene los parámetros de configuración de la transacción
	7.- Verifica si la tarjeta de debito/crédito existe y si esta con el estatus de activa
	8.- Debe verificar si la clave de la suscripción de banca celular es correcta
	9.- Verifica si el cliente este suscrito al servicio de banca por celular, si es el caso entonces procede a registrar la suscripción del servicio adicional. Si el cliente no tiene suscripción principal, se procede a registrar la suscripción al servicio banca por celular y luego se registra la suscripción adicional deseada.
	10.- Verifica si el cliente ya tiene registrada una suscripción adicional anterior con la misma tarjeta y celular, si no existe se vuelve a crear un nuevo registro de suscripción adicional. Si ya existe y esta inactiva únicamente se la vuelve a activar. Si existe y esta reversada borra el registro de esta suscripción adicional reversada y se vuelve a crear un nuevo registro de suscripción adicional.

	11.- Graba una bitácora transaccional
	12.- En caso de que la transacción esta configurada para el cobro por la suscripción en línea, el sistema procede a debitar el valor de la cuenta del cliente asociada a la tarjeta de debito. Si es una tarjeta de crédito envía o dispara una transacción de compra como si fuera de un pos virtual.
	13.- Actualiza la bitácora transaccional como OK o el error presentado
	14.- Se formatea un mensaje de respuesta OK de la suscripción adicional. Si la suscripción adicional es realizada desde la Intranet presenta un mensaje al funcionario de atención al cliente y se envía un mensaje SMS al cliente de bienvenida al servicio de Banca celular por celular. Si la suscripción fue realizada desde el celular, simplemente muestra el mensaje de aceptación del nuevo servicio.
15. Si el cliente origino la transacción de suscripción adicional desde su celular recibe un mensaje de aceptación de la suscripción del servicio adicional de banca celular. Esto indica que el proceso de suscripción adicional es OK y puede empezar a utilizar ya el servicio. Si el cliente de suscripción es el ejecutivo de atención al cliente, recibe el mensaje de aceptación de suscripción adicional del cliente y debe indicarle al cliente que puede empezar a utilizar el servicio.	

Cursos alternos	
Línea 3:	Introducción de un nemónico de operación invalido. Indicar error
Línea 3:	Introducción de un número de tarjeta no numérico y tamaño incorrecto. Indicar error
Línea 3:	Introducción de una clave de tarjeta no numérica y tamaño incorrecto. Indicar error
Línea 4:	Introducción de una tarjeta debito/crédito que no pertenece la institución financiera. Indicar error
Línea 6:	Si la transacción no esta autorizada para la red adquiriente, no existe o no esta activa. Indicar error
Línea 7:	Si la tarjeta no existe o tiene estatus de no activa. Indicar error
Línea 8:	Si la clave es errónea. Indicar error
Línea 9:	En caso de no poder registrar la suscripción principal o la suscripción adicional. Indicar error
Línea 10:	En caso de no poder hacerlo. Indicar error
Línea 12:	Si no puede debitar, no hace el registro de la suscripción adicional del cliente. Indicar error por falta de fondos

3.2.5 *Gestión de adquiriencia de transacciones tipo PULL*

Caso de uso:	Transacción PULL
Actores:	Cliente institución financiera
Propósito:	Enviar un mensaje SMS solicitando una operación bancaria

Resumen:	El cliente envía un mensaje al número asignado a la institución por ejemplo el 2626 solicitando se ejecute cierta operación bancaria. Debe ingresar la instrucción como un mensaje sms y enviar. El sistema autentifica la suscripción, registra la transacción, realiza la operación bancaria solicitada por el cliente y se envía la respuesta al celular del cliente
Tipo:	Primario y esencial
Referencias cruzadas	*Funciones* R1.1, R1.2, R1.3
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1.- Este caso de uso comienza cuando el cliente de la institución financiera solicita una operación o un servicio bancario a través de su celular.	
a. Si el cliente a solicitado el servicio de consulta de saldos, véase la sección *Operación consulta de saldos*	
b. Si el cliente a solicitado la operación consulta de cheque cobrado, véase la sección *Operación consulta de cheque cobrado*	
c. Si el cliente a solicitado cambio de clave de servicios banca por celular, véase la sección *Operación cambio de clave*	
d. Si el cliente a solicitado consulta de saldos a pagar de tarjeta de crédito, véase la sección *Operación consulta de saldos de tarjeta de*	

crédito	
e. Si el cliente a solicitado consulta de cupos de tarjeta de crédito, véase la sección *Operación consulta de cupos de tarjeta de crédito*	
f. Si el cliente a solicitado bloqueo de tarjetas, véase la sección *Operación bloqueo de tarjetas*	
	2.- El sistema de Banca Celular debe verificar que los campos genéricos para cada operación PULL lleguen correctamente desde la pasarela.
	3.- El sistema de Banca Celular debe verificar que la tarjeta de debito/crédito relacionada con la suscripción del cliente este registrada como un producto del banco
	4.- El sistema debe determinar el tipo de producto al que pertenece el BIN de la tarjeta ya sea Debito, Visa o MasterCard y adicionalmente se debe determinar si el es local o es un producto de otra institución asociada al banco como por ejemplo una cooperativa.
	5.- Debe verificar si la operación solicitada por el cliente esta asociada al BIN de la tarjeta, si la operación esta registrada para su utilización desde la red adquiriente y si esta activa. Si esta autentificación es correcta obtiene los parámetros de configuración de la transacción

	6.- Verifica si el cliente que solicita la transacción tiene o esta suscrito al servicio de banca celular y que la suscripción esta activa.
	7.- Si la transacción que solicita requiere de clave debe verificar que la clave del servicio haya sido ingresada por el cliente y este correcta
	8.- Si la transacción que solicita requiere de una suscripción adicional debe verificar que la operación que solicita el cliente este suscrita y que este activa la suscripción
	9.- Se consulta por el estado de la tarjeta de debito/crédito de suscripción al servicio. La tarjeta debe estar activa.
	10.- Graba una bitácora transaccional de la operación solicitada con estatus de operación bancaria en proceso
	11.- Ejecuta la operación bancaria relacionada con el servicio solicitado
	a. Si el cliente a solicitado la operación consulta de saldos, véase la sección *Operación consulta de saldos*
	b. Si el cliente a solicitado la operación consulta de cheque cobrado, véase la sección *Operación consulta de cheque cobrado*
	c. Si el cliente a solicitado cambio de clave de servicios banca por celular, véase la sección *Operación cambio de clave*
	d. Si el cliente a solicitado consulta de saldos a pagar de tarjeta de crédito, véase la sección *Operación consulta de saldos de tarjeta de crédito*
	e. Si el cliente a solicitado consulta de cupos de tarjeta de crédito, véase la sección *Operación consulta de cupos de tarjeta de*

	crédito
	f. Si el cliente a solicitado bloqueo de tarjetas, véase la sección *Operación bloqueo de tarjetas*
	12.- Verifica si la transacción tiene costo y realiza el cobro en línea
	a. Si el cliente esta suscrito con la tarjeta de débito, véase la sección *Operación pago con tarjeta de débito*
	b. Si el cliente esta suscrito con la tarjeta de crédito, véase la sección *Operación pago con tarjeta de crédito*
	13. Formatea el mensaje de salida a la pasarela dependiendo de la operación
	14. Actualiza la bitácora transaccional.
15. El usuario recibe en su celular un mensaje sms con la información solicitada o la confirmación de la operación bancaria solicitada.	
Cursos alternos	
Línea 2:	Introducción de un mensaje incorrecto, Indicar error
Línea 3:	Bin de tarjeta no pertenece a ningún producto del Banco. Indicar error
Línea 5:	Si la operación solicitada no esta relacionada al bin de la tarjeta o no esta activa. Indicar error.
Línea 6:	Si la suscripción del cliente no existe o no esta activa. Indicar error.
Línea 7:	Si la clave es ingresada es incorrecta. Indicar

	error
Línea 8:	Si la operación solicitada por el cliente requiere de suscripción adicional y si el cliente no tiene esta suscripción. Indicar error
Línea 9:	Si el estado de la tarjeta de debito/crédito esta cancelada. Debe indicar error
Línea 11:	Si la operación bancaria solicitada no puede realizar el servicio. Indicar error
Línea 12:	Si no puede realizar el cobro por falta de fondos. Indicar error
Línea 14:	Si no puede actualizar. Indicar error

Caso de uso:	**Transacción PULL**
Sección: pago con tarjeta de débito	
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1.- El sistema solicita el debito por cobro de comisión por la transacción solicitada, debe especificar el número de la tarjeta de crédito y la comisión a cobrar.	
	2.- Se genera una transacción de debito al autorizador de debito.
3.- El sistema debe obtener la cuenta y el tipo de cuenta de debito del cliente para posteriormente grabar en la bitácora transaccional	

Cursos alternos	
Línea 2:	Si no se debita el monto especificado. Debe indicar el motivo

Caso de uso:	**Transacción PULL**
Sección: pago con tarjeta de crédito	
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1.- El sistema solicita el consumo vía POS virtual por el valor de la comisión por la transacción solicitada, debe especificar el número de la tarjeta de crédito y la comisión a cobrar.	
	2.- Se genera una transacción de compra al autorizador de crédito VISA/MASTERCARD.
3.- El sistema debe obtener el número de autorización de la compra enviada para posteriormente grabar en la bitácora transaccional	

Cursos alternos	
Línea 2:	Si la compra no se puede realizar. Debe indicar el motivo

Caso de uso:	**Transacción PULL**
Sección: Operación consulta de saldos	
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1.- El cliente debe enviar un mensaje al 2525 y debe puede	

hacer las siguientes consultas de saldos.	
sca - consulta de saldos de cuenta de ahorros	
scc - consulta de saldos de cuenta de corriente	
s - consulta todas las cuentas asociadas a la suscripción	
	…
	11.- El sistema invoca al servicio de consulta de saldos de la institución financiera en función del numero de tarjeta de suscripción y el tipo de consulta que solicito el cliente
	…
	13.- El sistema formatea la respuesta que se va a proporcionar al cliente
	…
15.- El cliente recibe el mensaje de respuesta enviado por el banco en donde se presenta el resultado de su solicitud de consulta de saldos.	

Cursos alternos	
Línea 11:	Si el servicio de consulta de saldos no puede ejecutar la operación financiera. Indicar error

Caso de uso:	**Transacción PULL**
Sección: Operación consulta de cheque cobrado	
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1.- El cliente debe enviar un mensaje al 2525 y debe puede	

hacer la consulta del estatus de un cheque.	
chq 6dig*clave*N°chq	
en donde,	
chq - Operación de consulta de cheque	
6 dig - 6 último dígitos de la tarjeta de debito	
clave - 4 dígitos de la clave de banca celular	
Nº chq - Numero de cheque	…
	11.- El sistema debe obtener la cuenta corriente asociada a la tarjeta de debito de suscripción del cliente
	12.- El sistema invoca al servicio de consulta de estatus de cheques de la institución financiera en función del número de cuenta corriente y el numero del cheque e indica si el cheque esta cobrado o no
	…
	13.- El sistema formatea la respuesta que se va a proporcionar al cliente
	…
15.- El cliente recibe el mensaje de respuesta enviado por el banco en donde se presenta el resultado de su solicitud de consulta de cheque cobrado.	

Cursos alternos	
Línea 12:	Si el servicio de consulta de estatus de cheque no puede ejecutar la operación financiera. Indicar error

Caso de uso:	**Transacción PULL**
Sección: Operación cambio de clave	
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1.- El cliente debe enviar un mensaje al 2525 y puede realizar el cambio de clave del servicio banca celular El ejecutivo de atención al cliente desde el sistema de BackOffice también puede realizar la gestión de suscripción al servicio banca celular	
cla 6dig*ClaveAnterior*ClaveNueva	
en donde,	
cla - Operación de cambio de clave de banca celular	
6 dig - 6 último dígitos de la tarjeta de debito/crédito	
ClaveAnterior - 4 dígitos numéricos de la clave de banca celular	
ClaveAnterior - 4 dígitos numéricos de la nueva clave de banca celular	…
	11.- El sistema invoca al servicio de cambio de clave banca celular
	…
	13.- El sistema formatea la respuesta que se va a proporcionar al cliente
	…
15.- El cliente o el sistema BackOffice recibe el mensaje de enviado por el banco en donde se indica la aceptación o negación del cambio de clave de banca celular.	

Cursos alternos	
Línea 11:	Si el servicio de cambio de clave no puede ejecutar la operación financiera. Indicar error

Caso de uso:	**Transacción PULL**
Sección: Operación consulta de saldos tarjeta de crédito	
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1.- El cliente debe enviar un mensaje al 2525 y debe puede hacer la consulta del estatus de un cheque.	
stc 6dig*Clave	
en donde,	
stc - Operación de cambio de clave de banca celular	
6 dig - 6 último dígitos de la tarjeta de crédito de suscripción	
Clave - 4 dígitos numéricos de la clave de banca celular	
	11.- El sistema invoca al servicio de consulta de saldos de tarjeta de crédito en función del número de tarjeta. Debe devolver saldo disponible, saldo utilizado, valor mínimo a pagar, valor máximo a pagar, fecha máxima de pago
	…
	13.- El sistema formatea la respuesta que se va a proporcionar al cliente
	…

15.- El cliente recibe el mensaje de respuesta enviado por el banco en donde se indica los saldos de la tarjeta de crédito suscrita.	

Cursos alternos	
Línea 11:	Si el servicio de consulta de saldos de tarjeta de crédito no puede ejecutar la operación financiera. Indicar error

Caso de uso:	**Transacción PULL**
Sección: Operación consulta de cupos de tarjeta de crédito	
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1.- El cliente debe enviar un mensaje al 2525 y debe puede hacer la consulta del estatus de un cheque.	
ctc 6dig*Clave	
en donde,	
ctc - Operación de consulta de cupos de tarjeta de crédito	
6 dig - 6 último dígitos de la tarjeta de crédito de suscripción	
Clave - 4 dígitos numéricos de la clave de banca celular	
	11.- El sistema invoca al servicio de consulta de cupos de tarjeta de crédito en función del numero de tarjeta de suscripción y debe presentar el cupo total utilizado y disponible
	…

	13.- El sistema formatea la respuesta que se va a proporcionar al cliente
	…
15.- El cliente recibe el mensaje de respuesta enviado por el banco en donde se indica los cupos actuales de la tarjeta de crédito suscrita.	

Cursos alternos	
Línea 11:	Si el servicio de consulta de cupos de tarjeta de crédito no puede ejecutar la operación financiera. Indicar error

3.2.6 *Gestión de generación de transacciones tipo PUSH ONLINE con suscripción*

Caso de uso:	**Transacción PUSH Online - Con Suscripción**
Actores:	Generador de eventos
Propósito:	Enviar un mensaje SMS de aviso o notificación a un cliente que tiene suscripción al servicio de Banca Celular
Resumen:	Un sistema receptor de eventos de transacciones de cajero automático, pos, generan avisos o notificaciones a cliente que están suscritos al servicio Banca Celular a través un mensaje sms al celular suscrito.
Tipo:	Primario y esencial
Referencias cruzadas	*Funciones* R1.5
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**

1.- Este caso de uso comienza cuando el sistema receptor de eventos, recibe un evento de compra de un POS o un retiro o avance de efectivo desde un ATM, este sistema genera el evento al sistema de Banca celular para que este sea el que determine si envía o no el mensaje celular.	
2.- El sistema receptor de eventos dependiendo del origen de la transacción envía la notificación al sistema de Banca Celular para que este sea el que determine el envío del mensaje al cliente suscrito de banca celular.	
En las secciones que se especifican a continuación se presentan los casos de este tipo de operación PUSH Online con suscripción	
a. Ver sección "*Notificación de retiros/avances de efectivo desde cajeros automáticos propios*"	
b. Ver sección "Notificación de compras	

en POS propios"	
c. Ver sección "Notificación de autorizaciones de compra por teléfono"	
d. Ver sección "Notificación de Compras/Retiros/Avances en efectivo desde el exterior"	
e. Ver sección "Notificación de avisos en Batch "	
	3.- El sistema Banca Celular espera que el sistema receptor de eventos le envíe la siguiente información:
	OperaciónPushBatchCon Suscripcion*TipPas*NumTar*NumCel* Mensaje_Celular*CampoInformativo1*CampoInfromativo2
	en donde,
	OperaciónPushBatchConSuscripcion, indica que debe enviar un mensaje al cliente de la institución financiera que suscrito al servicio de Banca Celular.
	TipPas, este campo indica si el mensaje se lo va a enviar a través de correo electrónico o a través de la pasarela sms
	NumTar, numero de tarjeta es asignada por el administrador del sistema y es un numero de tarjeta que tiene un bin especial para este tipo de transacciones
	NumCel, es el número celular destino a donde se va a enviar el mensaje
	Mensaje_Celular, mensaje que desea enviar
	CampoInformativo1, opcional campo informativo que se va a graba en la bitácora transaccional
	CampoInformativo2, opcional campo informativo que se va a graba en la bitácora transaccional

	4.- El sistema de Banca Celular debe verificar que los campos genéricos para cada operación PUSH Online con suscripción llegue correctamente desde el sistema de generación de avisos online.
	5.- El sistema de Banca Celular debe verificar que el BIN de la tarjeta enviada por el sistema automático este registrada como producto del banco
	6.- El sistema debe determinar el tipo de producto al que pertenece el BIN de la tarjeta especial y adicionalmente se debe determinar si el es local o es un producto de otra institución asociada al banco como por ejemplo una cooperativa.
	7.- Debe verificar si la operación solicitada por el cliente esta asociada al BIN de la tarjeta especial, si la operación esta registrada para su utilización desde la red adquiriente y si esta activa. Si esta autentificación es correcta obtiene los parámetros de configuración de la transacción
	8.- Verifica si la tarjeta que envía el receptor de eventos esta suscrito al servicio de banca celular y que la suscripción esta activa.
	9.- Si la transacción que solicita requiere de clave debe verificar que la clave del servicio haya sido enviada por el sistema receptor de eventos y sea valida
	10.- Si la transacción que solicita requiere de una suscripción adicional debe verificar que la operación que solicita el sistema receptor de eventos este suscrita y que este activa
	11.- Se consulta por el estado de la tarjeta de debito/crédito de suscripción al servicio. La tarjeta debe estar activa.
	12.- Graba una bitácora transaccional de la operación solicitada con estatus de operación bancaria en proceso
	13.- Verifica si la transacción tiene costo y realiza el cobro en línea

	a. Si el cliente esta suscrito con la tarjeta de débito, véase la sección *Operación pago con tarjeta de débito*
	b. Si el cliente esta suscrito con la tarjeta de crédito, véase la sección *Operación pago con tarjeta de crédito*
	14.- El mensaje puede ser enviado por la pasarela sms o vía email. Para cualquiera de los casos se llama a la función específica
15. El usuario recibe en su celular un mensaje sms con la notificación enviada por la institución financiera.	
	16. Actualiza la bitácora transaccional.
	17. Se formatea el mensaje de respuesta al sistema automático receptor de eventos
18. El sistema automático receptor de eventos recibe la respuesta y actualiza información en sus bitácoras especificas (opcional)	
Cursos alternos	
Línea 4:	Introducción de un mensaje incorrecto, Indicar error
Línea 5:	Bin de tarjeta no pertenece a ningún producto del Banco. Indicar error
Línea 7:	Si la operación solicitada no esta relacionada al bin de la tarjeta o no esta activa. Indicar error.
Línea 8:	Si la suscripción de la tarjeta no existe o no esta activa. Indicar error.
Línea 9:	Si la clave ingresada es incorrecta. Indicar error
Línea 10:	Si el cliente no tiene suscripción al servicio adicional. Indicar error

Línea 11:	Si el estado de la tarjeta esta cancelada. Debe indicar error
Línea 12:	Si no puede registrar la transacción. Indicar error
Línea 13:	Si no puede realizar cobro. Indicar error
Línea 14:	Si la operación de envío de mensajes por correo o por la pasarela no puede enviar el mensaje. Indicar error
Línea 16:	Si no puede actualizar. Indicar error

Caso de uso:	**Transacción PUSH ONLINE Con suscripción**
Sección: Notificación de retiros/avances de efectivo desde cajeros automáticos propios	
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1.- El sistema automático recepta un evento de retiro de una tarjeta de debito o un avance de efectivo de una tarjeta de crédito en un cajero automático propio de la institución financiera.	
Genera la siguiente información	
Fecha	
Tarjeta	
Valor	
Tipo de transacción	
Cajero Automático	
2.- El sistema que realiza la captura del evento debe obtener desde el registro de suscripciones del servicio Banca Celular el numero celular asociado a la tarjeta de debito/crédito que realizo el retiro o	

avance en efectivo respectivamente.	
3.- Se formatea el mensaje o aviso de seguridad que se va a enviar al cliente y se procede a enviar la solicitud de aviso al sistema Banca Celular	
	...
	14.- El sistema envía la notificación del evento presentado en el cajero automático al cliente
15.- El cliente recibe en su celular el mensaje de seguridad	
	…
14.- El sistema que origino la transacción recibe la respuesta del sistema Banca Celular.	
Cursos alternos	
Línea 2:	Si la tarjeta no tiene registrado el servicio en Banca Celular simplemente ignora la transacción debido a que no encontró el número celular para enviar el aviso.

Caso de uso:	**Transacción PUSH ONLINE Con suscripción**
Sección: Notificación de compras en POS propios	
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**

1.- El sistema automático recepta un evento de compra de una tarjeta de debito/crédito en un POS propio de la institución financiera.	
Genera la siguiente información	
Fecha	
Tarjeta	
Valor	
Tipo de transacción	
Establecimiento	
2.- El sistema que realiza la captura del evento debe obtener desde el registro de suscripciones del servicio Banca Celular el numero celular asociado a la tarjeta de debito/crédito que realizo la compra.	
3.- Se formatea el mensaje o aviso de seguridad que se va a enviar al cliente y se procede a enviar la solicitud de aviso al sistema Banca Celular.	
	...
	14.- El sistema envía la notificación del evento presentado en el POS al cliente
15.- El cliente recibe en su celular el mensaje de seguridad	
	…
14.- El sistema que origino la transacción recibe la respuesta del sistema Banca Celular.	
Cursos alternos	

Línea 2:	Si la tarjeta no tiene registrado el servicio en Banca Celular simplemente ignora la transacción debido a que no encontró el número celular para enviar el aviso.

Caso de uso:	**Transacción PUSH ONLINE Con suscripción**
Sección: Notificación de autorizaciones de compra por teléfono	
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1.- El sistema automático recepta un evento de autorización telefónica de compra de una tarjeta de crédito propia en un establecimiento afiliado a la institución financiera.	
Genera la siguiente información	
Fecha	
Tarjeta	
Valor	
Tipo de transacción	
Establecimiento	
2.- El sistema que realiza la captura del evento debe obtener desde el registro de suscripciones del servicio Banca Celular el numero celular asociado a la tarjeta de crédito que realizo la compra.	
3.- Se formatea el mensaje o aviso de seguridad que se va a enviar al cliente y se procede a enviar la solicitud de aviso al sistema Banca Celular.	

	…
	14.- El sistema envía la notificación del evento presentado en el sistema de autorizaciones manuales por teléfono
15.- El cliente recibe en su celular el mensaje de seguridad	
	…
14.- El sistema que origino la transacción recibe la respuesta del sistema Banca Celular.	
Cursos alternos	
Línea 2:	Si la tarjeta no tiene registrado el servicio en Banca Celular simplemente ignora la transacción debido a que no encontró el número celular para enviar el aviso.

Caso de uso:	**Transacción PUSH ONLINE Con suscripción**
Sección: Notificación de Compras/Retiros/Avances en efectivo desde el exterior	
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1.- El sistema automático recepta un evento de autorización telefónica de compra, retiro o avance en efectivo de una tarjeta debito/crédito propia desde el exterior.	
Genera la siguiente información	
Fecha switch	
Hora switch	
Tipo de transacción	

Origen	
Destino	
ID	
2.- El sistema debe verificar si el origen y destino de la transacción esta autorizada por la institución financiera para la generación de avisos	
3.- El sistema que realiza la captura del evento debe obtener desde el registro de suscripciones del servicio Banca Celular el numero celular asociado a la tarjeta de crédito que realizo la compra.	
3.- Se formatea el mensaje o aviso de seguridad que se va a enviar al cliente y se procede a enviar la solicitud de aviso al sistema Banca Celular.	
	...
	14.- El sistema envía la notificación del evento presentado en el exterior
15.- El cliente recibe en su celular el mensaje de seguridad	
	…
14.- El sistema que origino la transacción recibe la respuesta del sistema Banca Celular.	
Cursos alternos	
Línea 2:	Si no esta autorizada el origen y destino de la transacción se descarta el evento
Línea 3:	Si la tarjeta no tiene registrado el servicio en Banca Celular simplemente ignora la transacción debido a que no encontró el número celular para enviar el aviso.

3.2.7 *Gestión de generación de transacciones tipo PUSH ONLINE sin suscripción*

Caso de uso:	**Transacción PUSH Online - Sin Suscripción**
Actores:	Sistema Online
Propósito:	Enviar un mensaje SMS de aviso o notificación que no esta suscrito al servicio de Banca Celular
Resumen:	Un sistema automático que genera un aviso o notificación y envía al cliente a través un mensaje sms al celular.
Tipo:	Primario y esencial
Referencias cruzadas	*Funciones* R1.6
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1.- Este caso de uso comienza cuando un sistema online requiere enviar un mensaje celular a un cliente que no esta suscrito al servicio de Banca Celular.	
2.- El sistema automático enviar la solicitud de notificación	
	3.- El sistema de Banca Celular espera la solicitud de notificación
	OperaciónPushBatchSin Suscripcion*TipPas*NumTar*NumCel* Mensaje_Celular*CampoInformativo1*CampoInfromativo2
	en donde,

	OperaciónPushOnlineSinSuscripcion, indica que debe enviar un mensaje a un celular destino y no debe verificar si esta suscrito al servicio de Banca Celular por celular
	TipPas, este campo indica si el mensaje se lo va a enviar a través de correo electrónico o a través de la pasarela sms
	NumTar, numero de tarjeta es asignada por el administrador del sistema y es un numero de tarjeta que tiene un bin especial para este tipo de transacciones
	NumCel, es el número celular destino a donde se va a enviar el mensaje
	Mensaje_Celular, mensaje que desea enviar
	CampoInformativo1, opcional campo informativo que se va a graba en la bitácora transaccional
	CampoInformativo2, opcional campo informativo que se va a graba en la bitácora transaccional
	a. Una aplicación practica de este tipo operación sms se encuentra en la sección la sección "*Notificación de remesas del exterior*"
	4.- El sistema de Banca Celular debe verificar que los campos genéricos para cada operación PUSH Online sin suscripción llegue correctamente desde el sistema de generación de avisos online.
	5.- El sistema de Banca Celular debe verificar que la tarjeta especial enviada por el sistema automático este registrada
	6.- El sistema debe determinar el tipo de producto al que pertenece el BIN de la tarjeta especial y adicionalmente se debe determinar si el es local o es un producto de otra institución asociada al banco como por ejemplo una cooperativa.
	7.- Debe verificar si la operación solicitada por el cliente esta asociada al BIN de la tarjeta especial, si la operación esta registrada para su utilización desde la red adquiriente y si esta activa. Si esta autentificación es correcta obtiene los parámetros de configuración de la transacción

	8.- Se consulta por el estado de la tarjeta de especial de suscripción al servicio push-online sin suscripción. La tarjeta debe estar activa.
	9.- Graba una bitácora transaccional de la operación solicitada con estatus de operación bancaria en proceso
	10.- El mensaje puede ser enviado por la pasarela sms o vía email. Para cualquiera de los casos se llama a la función específica
11.- El usuario recibe en su celular un mensaje sms con la notificación enviada por la institución financiera.	
	12. Actualiza la bitácora transaccional.
	13. Se formatea el mensaje de respuesta al sistema automático
14. El sistema automático recibe la respuesta y actualiza información en sus bitácoras especificas	
Cursos alternos	
Línea 4:	Introducción de un mensaje incorrecto, Indicar error
Línea 5:	Bin de tarjeta no pertenece a ningún producto del Banco. Indicar error
Línea 6:	Si la operación solicitada no esta relacionada al bin de la tarjeta o no esta activa. Indicar error.
Línea 7:	Si la suscripción del cliente especial no existe o no esta activa. Indicar error.
Línea 7:	Si el estado de la tarjeta especial esta cancelada. Debe indicar error

Línea 10:	Si la operación de envío de mensajes por correo o por la pasarela no puede enviar el mensaje. Indicar error
Línea 12:	Si no puede actualizar. Indicar error

Caso de uso:	**Transacción PUSH ONLINE Sin Suscripción**
Sección: Notificación de remesas del exterior	
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1.- El sistema automático cada n minutos accede al repositorio central de remesas para verificar si han llegado nuevas remesas desde el exterior, únicamente de las empresas remesadoras que tienen autorizado el canal de notificación sms	
2.- En caso que existan nuevas remesas verifica por cada remesa que en el campo numero de teléfono se tenga un numero celular valido y cuya serie corresponda a una de las tres operadoras celulares que dan servicio en el país. Por cada remesa formatea el mensaje de operación Push Online - sin suscripción y envía al sistema de Banca Celular	
	...
	10.- El sistema envía la notificación de la llegada de la remesa
11.- El beneficiario de la remesa recibe su notificación de llegada de	

la remesa	
	…
14.- El sistema automático de generación de aviso de remesas actualiza la base de remesas indicando que ha sido notificado vía sms.	

Cursos alternos	
Línea 2:	Si el número de celular enviado por la remesadora no es un número valido, se descarta el envío y pasa a la siguiente remesa.

3.2.8 *Gestión de generación de transacciones tipo PUSH BATCH con suscripción*

La gestión de transacción tipo push batch con suscripción es un caso de uso del sistema push online con suscripción, debido a que únicamente cambia el actor, en este caso el actor es un sistema batch que va a solicitar en lotes el envío de un mensaje sms al sistema de banca celular, previa verificación de la suscripción del cliente a este servicio.

Caso de uso:	**Transacción PUSH ONLINE Con suscripción**
Sección: Notificación de avisos en Batch	
Actor	**Sistema Batch**
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1.- Un sistema externo genera notificaciones en un proceso batch.	
Genera la siguiente información	

Id Cliente	
Mensaje a notificar	
2.- El sistema batch en función del ID del cliente debe obtener desde el registro de suscripciones del servicio Banca Celular el numero celular y la tarjeta asociado al id del cliente.	
3.- Se formatea el mensaje o aviso de seguridad que se va a enviar al cliente y se procede a enviar la solicitud de aviso al sistema Banca Celular.	
	...
	14.- El sistema envía la notificación generada por el sistema batch
15.- El cliente recibe en su celular el mensaje sms	
	…
14.- El sistema que origino la transacción recibe la respuesta del sistema Banca Celular.	
Cursos alternos	
Línea 2:	Si el id del cliente no tiene registrado el servicio en Banca Celular simplemente ignora la transacción debido a que no encontró el numero celular, ni tarjeta asociada de suscripción para enviar el aviso.

3.2.9 *Gestión de generación de transacciones tipo PUSH BATCH sin suscripción*

Caso de uso:	**Transacción PUSH BATCH - Sin Suscripción**
Actores:	Sistema BATCH
Propósito:	Enviar un mensaje SMS de aviso o notificación a un cliente no suscrito al servicio de Banca Celular
Resumen:	Un sistema Batch deposita los mensajes a enviar en un repositorio central. Un proceso automático cada 10 minutos recupera los mensajes a enviar del repositorio en función de un horario establecido.
Tipo:	Primario y esencial
Referencias cruzadas	*Funciones* R1.8
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1.- Este caso de uso comienza cuando un sistema batch de la institución financiera deposita en un repositorio central los mensajes o avisos a enviar a clientes no suscritos al servicio banca celular.	
2.- El sistema batch debe especificar el numero celular y el mensaje a enviar	

	3.- El sistema de banca celular cada 10 minutos consulta al repositorio central por nuevos mensajes a enviar en función de los horarios establecidos para cada tipo de mensaje a enviar. Ver la sección "*Configuración de transacciones*"
	4.- El sistema de Banca Celular espera la solicitud de operación push batch sin suscripción
	OperaciónPushBatchSin Suscripcion*TipPas*NumTar*NumCel* Mensaje_Celular*CampoInformativo1*CampoInfromativo2
	en donde,
	OperaciónPushBatchSinSuscripcion, indica que debe enviar un mensaje a un celular destino y no debe verificar si esta suscrito al servicio de Banca Celular por celular
	TipPas, este campo indica si el mensaje se lo va a enviar a través de correo electrónico o a través de la pasarela sms
	NumTar, numero de tarjeta es asignada por el administrador del sistema y es un numero de tarjeta que tiene un bin especial para este tipo de transacciones
	NumCel, es el número celular destino a donde se va a enviar el mensaje
	Mensaje_Celular, mensaje que desea enviar
	CampoInformativo1, opcional campo informativo que se va a graba en la bitácora transaccional
	CampoInformativo2, opcional campo informativo que se va a graba en la bitácora transaccional
	5.- El sistema de Banca Celular debe verificar que los campos genéricos para cada operación PUSH Batch sin suscripción llegue correctamente desde el proceso automático.
	6.- El sistema de Banca Celular debe verificar que el BIN de la tarjeta especial enviada por el sistema automático este registrado

	7.- El sistema debe determinar el tipo de producto al que pertenece el BIN de la tarjeta especial y adicionalmente se debe determinar si el es local o es un producto de otra institución asociada al banco como por ejemplo una cooperativa.
	8.- Debe verificar si la operación solicitada por el cliente esta asociada al BIN de la tarjeta especial, si la operación esta registrada para su utilización desde la red adquiriente y si esta activa. Si esta autentificación es correcta obtiene los parámetros de configuración de la transacción
	9.- Se consulta por el estado de la tarjeta de especial de suscripción al servicio push-online sin suscripción. La tarjeta debe estar activa.
	10.- Graba una bitácora transaccional de la operación solicitada con estatus de operación bancaria en proceso
	11.- El mensaje puede ser enviado por la pasarela sms o vía email. Para cualquiera de los casos se llama a la función específica
12.- El usuario recibe en su celular un mensaje sms con la notificación enviada por la institución financiera.	
	13. Actualiza la bitácora transaccional.
	14. Se formatea el mensaje de respuesta al sistema automático
15. El sistema automático recibe la respuesta y actualiza información en sus bitácoras especificas	

Línea 3:	En caso de existir ciertos tipos de mensajes pendientes de envío que no estén dentro del horario configurado para la transacción simplemente se descartan, hasta que el proceso automático vuelva a leerlos y los mensajes pendientes este dentro de sus horarios permitidos de envíos
Línea 5:	Introducción de un mensaje incorrecto, Indicar error
Línea 6:	Bin de tarjeta no pertenece a ningún producto del Banco. Indicar error
Línea 7:	Si la operación solicitada no esta relacionada al bin de la tarjeta o no esta activa. Indicar error.
Línea 8:	Si la suscripción del cliente especial no existe o no esta activa. Indicar error.
Línea 9:	Si el estado de la tarjeta especial esta cancelada. Debe indicar error
Línea 11:	Si la operación de envío de mensajes por correo o por la pasarela no puede enviar el mensaje. Indicar error
Línea 13:	Si no puede actualizar. Indicar error

Caso de uso:	**Transacción PUSH BATCH Sin suscripción**
Sección: Configuración de transacciones	
Actor	**Administrador del sistema**
Curso normal de los eventos	
Acción del actor	**Respuesta del sistema**
1.- Este caso de uso inicia cuando el administrador del sistema necesita configurar los días y horarios de envío de una transacción tipo PUSH - BATCH sin suscripción	

2.- Se debe configurar los días de la semana que una transacción puede enviar mensajes, y un rango los horarios en los que se puede enviar cada transacción, y desde que fecha se requiere habilitar la transacción	
	3.- El sistema de banca celular verifica que los parámetros ingresados estén correctos
	4.- Guarda la información de los horarios establecidos
5.- El Administrador recibe la confirmación de la configuración	
Cursos alternos	
Línea 3:	En caso de parámetros incorrectos indicar error
Línea 4:	Si no puede registrar la información. Indicar error

3.3 Modelo conceptual

Un modelo conceptual es una representación de conceptos en un dominio. Puede mostrarnos conceptos asociaciones entre conceptos y atributos de conceptos. Ofrece la ventaja de subrayar fuertemente una concentración en los conceptos del dominio, no en las entidades de software

Un modelo conceptual facilita además de conocer el dominio: sus conceptos, su terminología y sus relaciones.

Básicamente en el modelo conceptual cae la distinción fundamental entre el análisis orientado a objetos y el análisis estructurado: división por conceptos (objetos) y no por funciones.

Ver anexo 3.1 Modelo conceptual

3.4 Diagramas de secuencia

Los casos de uso indican como los actores interactúan con el sistema de software. Durante la interacción un actor genera eventos dirigidos a un sistema, solicitando alguna operación.

El diagrama de secuencia de un sistema es una representación que muestra en determinado escenario de un caso de uso[3], los eventos generados por actores externos, su orden y los eventos internos del sistema. A todos los sistemas se les trata como una caja negra; los diagramas se centran en los eventos que trascienden las fronteras del sistema y que fluyen de los actores a los sistemas.

Para poder construir los diagramas de secuencia se debe tener claro que es un evento y operación del sistema.

El *evento de un sistema*, es un hecho externo de entrada que un actor produce en un sistema. El evento da origen a una operación de respuesta.

La *operación de un sistema*, es una acción que se ejecuta en respuesta a un evento del sistema.

Ver anexo 3.2 Diagramas de secuencia

3.5 Contratos de operación

Un contrato[4] es un documento en donde se describe lo que una operación se propone lograr. Se redactan en un estilo declarativo y enfatizando *lo que* sucederá y no *como* se conseguirá. Los contratos suelen expresarse a partir de los cambios de estado de las precondiciones y poscondiciones. Puede elaborarse un contrato para un método de una clase de software o para una operación más global del sistema.

[3] El escenario de un caso de uso es una instancia o trayectoria realizada por medio del uso: un ejemplo real de su ejecución

[4] Bertrand Meyer fue el primero en difundir el término

Contrato	
Nombre:	Administrador sistema
Responsabilidades:	Crear una nueva transacción del servicio Banca Celular tipo PULL, PUSH, BackOffice
Tipo:	Sistema
Referencias cruzadas:	*Funciones* R1.1
Notas:	
Excepciones	
Si no puede registrar nueva transacción. Indicar error.	
Salida:	
Precondiciones	
El administrador debe definir a Banca celular SMS como un servicio	
El administrador debe definir el bin de la tarjeta	
El administrador debe crear un grupo de operaciones que pertenecen a un bin	
El administrador debe crear los tipos de operación PULL, PUSH, PUSH-Online con suscripción, PUSH-Batch con suscripción, PUSH-Online sin suscripción, PUSH-Batch sin suscripción	
El administrador debe definir las operaciones que brinda el servicio banca celular	
Poscondiciones	
Si se trata de una nueva definición de transacción iniciada por el administrador se instancia EspecificacionTransaccion, se modifica los atributos de número de transacción, descripción de transacción	
Se asocia a la instancia de EspecificacionTransaccion el código de red adquiriente, el código del servicio, el código de operación, si requiere clave, si requiere suscripción adicional, costos de adquiriente, costos de red autorizadora, por tipo de cliente, para transacciones de tipo push se define el numero de tarjeta especial, días y horarios de envío y fecha de inicio de notificaciones	

El administrador debe crear las instancias de asociación del grupo y las transacciones

Contrato	
Nombre:	TransaccionBACKOFFICE
Responsabilidades:	El funcionario de atención al cliente una operación bancaria de soporte solicitada por el cliente desde la Intranet
Tipo:	Sistema
Referencias cruzadas:	*Funciones* R1.2
Notas:	El funcionario de atención al cliente debe recibir la respuesta de la operación que solicitada en su pantalla
Excepciones	
El funcionario de atención al cliente debe especificar su usuario y clave para ingresar al sistema Intranet, si son incorrectos indicar error	
Si la operación de BackOffice no se puede completar, debe indicar error.	
Salida:	
Precondiciones	
BIN registrado por el administrador	
Transacción asociada a la operación solicitada por el ejecutivo de atención al cliente pertenezca al grupo de transacciones del BIN asignado por el administrador	
Poscondiciones	
Si se trata de una operación iniciada desde AtencionCliente se instancia al ReceptorTransaccion se asocia el código de operación solicitada y parámetros que requiere la operación de BackOffice	
PasarelaSMS crea una instancia al ReceptorTransaccion,	
ReceptorTransaccion solicita datos de la transacción asociada a la operación de suscripción enviada por el cliente celular, crea instancia	
Se ejecuta la operación BackOffice solicitada por el ejecutivo de atención al cliente, se crea instancia	
Se instancia el PagoOnline, cambia el atributo monto	

Se instancia BitacoraTransaccional y modifica los atributos descritos en el modelo conceptual
ReceptorTransaccion modifica atributo de respuesta de operación BackOffice y envía respuesta a la pantalla del ejecutivo de atención al cliente

Contrato	
Nombre:	RegistrarSuscripcion
Responsabilidades:	Registrar una suscripción al servicio Banca Celular de un cliente
Tipo:	Sistema
Referencias cruzadas:	*Funciones* R1.1, R1.2, R1.3
Notas:	El cliente debe recibir mensaje de bienvenida al servicio debe forma rápida.
Excepciones	
Si el BIN de tarjeta no pertenece a un producto de la institución financiera, indicar el error	
Si tarjeta no esta activa, indicar error	
Si la clave no es valida, indicar error	
Si no puede realizar el cobro, indicar error	
Salida:	
Precondiciones	
El cliente tiene tarjeta de débito/crédito de la institución financiera	
El cliente conoce el PIN de la tarjeta	
Autorización de operación para servicio, red adquiriente (SMS, Intranet).	
BIN registrado por el administrador	
Transacción asociada a la operación de suscripción pertenezca al grupo de transacciones del BIN asignado por el administrador	
Poscondiciones	
Si se trata de una nueva suscripción iniciada desde el cliente-celular se instancia a la PasarelaSMS	
Si se trata de una nueva suscripción iniciada por el ejecutivo de atención al cliente desde la intranet se instancia al ReceptorTransaccion	

PasarelaSMS o AtencionCliente crea una instancia al ReceptorTransaccion, se asocia el código de operación solicitada, el numero celular, el numero de tarjeta
ReceptorTransaccion solicita datos de la transacción asociada a la operación de suscripción enviada por el cliente celular
Si se trata de una nueva suscripción, se crea un registro de suscripción
Se asocia a la suscripción el tipo, numero de tarjeta y localidad y numero celular
Se instancia el PagoOnline, cambia el atributo monto
Se instancia BitacoraTransaccional y modifica los atributos descritos en el modelo conceptual

Contrato	
Nombre:	RegistrarSuscripcionAdicional
Responsabilidades:	Registrar una suscripción al servicio Banca celular de un cliente
Tipo:	Sistema
Referencias cruzadas:	*Funciones* R1.1, R1.2, R1.3
Notas:	El cliente debe recibir mensaje de bienvenida al servicio debe forma rápida.
Excepciones	
Si el BIN de tarjeta no pertenece a un producto de la institución financiera, indicar el error	
Si tarjeta no esta activa, indicar error	
Si la clave no es valida, indicar error	
Si no puede realizar el cobro, indicar error	
Salida:	
Precondiciones	
El cliente tiene tarjeta de débito/crédito de la institución financiera	
El cliente conoce el PIN de la tarjeta	
El cliente conoce el código del servicio adicional que desea suscribir	
Autorización de operación para servicio, red adquiriente (SMS, Intranet).	
BIN registrado por el administrador	

Transacción asociada a la operación de suscripción pertenezca al grupo de transacciones del BIN asignado por el administrador
Poscondiciones
Si se trata de una nueva suscripción a servicio adicional iniciada desde el cliente-celular se instancia a la PasarelaSMS
Si se trata de una nueva suscripción a servicio adicional iniciada por el ejecutivo de atención al cliente desde la intranet se instancia al ReceptorTransaccion
PasarelaSMS o AtencionCliente crea una instancia al ReceptorTransaccion, se asocia el código de operación solicitada, el numero celular, el numero de tarjeta
ReceptorTransaccion solicita datos de la transacción asociada a la operación de suscripción de servicio adicional enviada por el cliente celular
Si se trata de una nueva suscripción a servicio adicional, se crea un nuevo registro
Se asocia a la suscripción adicional el numero de tarjeta, numero celular y el código de operación adicional
Se instancia el PagoOnline, cambia el atributo monto
Se instancia BitacoraTransaccional y modifica los atributos descritos en el modelo conceptual

Contrato	
Nombre:	Transacción PULL
Responsabilidades:	Iniciar una operación bancaria solicitada por el cliente desde su celular
Tipo:	Sistema
Referencias cruzadas:	*Funciones* R1.4
Notas:	El cliente debe recibir en su celular un mensaje con el resultado de la operación bancaria solicitada
Excepciones	
Si el cliente no esta suscrito al servicio banca celular, indicar error	
Si tarjeta no esta activa, indicar error	
Si transacción requiere clave y si el cliente no ingresa o es incorrecta,	

indicar error	
Si el cliente requiere de suscripción adicional y no tiene o no esta activa, indicar error	
Si no puede realizar el cobro, indicar error	
Salida:	
Precondiciones	
El cliente esta suscrito al servicio banca celular	
Para ciertas transacciones debe conocer la clave del servicio banca celular	
Para ciertas transacciones debe conocer el numero de tarjeta de suscripción al servicio banca celular	
Para ciertas operaciones el cliente debe tener una suscripción adicional al servicio deseado	
BIN registrado por el administrador	
Transacción asociada a la operación solicitada por el cliente pertenezca al grupo de transacciones del BIN asignado por el administrador	
Poscondiciones	
Si se trata de una operación iniciada desde el cliente-celular se instancia a la PasarelaSMS	
PasarelaSMS crea una instancia al ReceptorTransaccion, se asocia el código de operación solicitada, el numero celular, el numero de tarjeta	
ReceptorTransaccion solicita datos de la transacción asociada a la operación de suscripción enviada por el cliente celular, crea instancia	
Se ejecuta la operación bancaria solicitada por el cliente, se crea instancia	
Se instancia el PagoOnline, cambia el atributo monto	
Se instancia BitacoraTransaccional y modifica los atributos descritos en el modelo conceptual	
ReceptorTransaccion modifica atributo de respuesta de operación bancaria y envía a la PasarelaSMS para formatear el mensaje de respuesta a cliente	

Contrato	
Nombre:	Transacción PUSH
Responsabilidades:	Enviar un mensaje o aviso de notificación de la institución bancaria al celular del cliente

Tipo:	Sistema
Referencias cruzadas:	*Funciones* R1.5, R1.6, R1.7, R1.8
Notas:	El cliente debe recibir en su celular un mensaje de notificación, aviso de advertencia, etc. que la institución bancaria desea comunicar
Excepciones	
Para notificación con suscripción debe verificar que la suscripción y tarjeta estén activas	
Para notificación sin suscripción el numero de teléfono celular que se especifica debe ser valido	
Si el cliente requiere de suscripción adicional y no tiene o no esta activa, indicar error	
Si no puede realizar el cobro, indicar error	
Salida:	
Precondiciones	
Para notificaciones a clientes suscritos al servicio Banca Celular el programa que solicita el envío del mensaje debe obtener el numero de celular asociado en el registro de suscripciones a partir del numero de identificación o numero de tarjeta del cliente	
Para notificación a cliente sin suscripción al servicio banca celular el sistema que solicita el envío del mensaje debe tener un numero celular destino valido, de tal manera que la pasarela no genere un error	
Para ciertas notificaciones a clientes suscritos al servicio banca celular, deben tener una suscripción al servicio adicional	
BIN registrado por el administrador	
Transacción asociada a la operación solicitada por sistema de notificación que pertenezca al grupo de transacciones del BIN asignado por el administrador	
Poscondiciones	
Un sistema-online o un sistema-batch que necesita enviar un aviso o notificación a un cliente suscrito o no suscrito al servicio banca celular crea una instancia al sistema ReceptorTransaccion	
ReceptorTransaccion solicita datos de la transacción asociada a la operación de notificación PUSH con suscripción o sin suscripción	
ReceptorTransaccion crea una nueva instancia a PasarelaSMS y envía el numero celular y mensaje a enviar	
Para clientes suscritos se instancia el PagoOnline, cambia el atributo monto	

Se instancia BitacoraTransaccional y modifica los atributos descritos en el modelo conceptual
ReceptorTransaccion modifica atributo de respuesta de operación push y envía resultado del envío al sistema online o batch que origino la transacción

3.6 Diagramas de estado

Muestran el conjunto de estados por los cuales pasa un objeto durante su vida en una aplicación, junto con los cambios que permiten pasar de un estado a otro.

Son útiles sólo para los objetos con un comportamiento significativo. Cada objeto está en un estado en cierto instante. El estado está caracterizado parcialmente por los valores algunos de los atributos del objeto. El estado en el que se encuentra un objeto determina su comportamiento. Cada objeto sigue el comportamiento descrito en el Diagrama de Estados asociado a su clase. Los Diagramas de Estados y escenarios son complementarios, los Diagramas de Estados son autómatas jerárquicos que permiten expresar concurrencia, sincronización y jerarquías de objetos, son grafos dirigidos y deterministas. La transición entre estados es instantánea y se debe a la ocurrencia de un evento.

Ver anexo 3.3 Diagramas de estados

3.7 Clasificación y programación de casos de uso. Versión 1.

A continuación se presenta la programación de los casos de uso que se van a desarrollar en la primera interación. A esta versión la hemos llamado 1.0. Se implementa la funcionalidad completa de todos los casos de uso esenciales. Únicamente para el caso de administración del sistema no se implementa en esta versión el sistema front end para la administración.

Primera Iteración	**Versión 1.0**	
Casos de uso	**Funcionalidad**	**Interfaz**
Gestión de administración del sistema de Banca Electrónica	Completa	No se implementa sistema front end

		para cliente
Gestión BackOffice del producto Banca Electrónica	Completa	OK
Gestión de suscripción de clientes.	Completa	OK
Gestión de suscripción de un servicio adicional.	Completa	OK
Gestión de adquiriencia de transacciones tipo PULL	Completa	OK
Gestión de generación de transacciones tipo PUSH ONLINE con suscripción	Completa	OK
Gestión de generación de transacciones tipo PUSH ONLINE sin suscripción	Completa	OK
Gestión de generación de transacciones tipo PUSH BATCH con suscripción	Completa	OK
Gestión de generación de transacciones tipo PUSH BATCH sin suscripción	Completa	OK

3.8 Conclusiones

- No existe un modelo conceptual apropiado para todos los casos o circunstancias.

- Todos los modelos conceptúales no son más que aproximaciones al dominio que intentamos entender. Podemos indicar que un buen modelo conceptual capta las abstracciones esenciales y la información indispensable para comprender el dominio dentro del contexto de los requerimientos.

- La identificación y ordenamiento de los eventos de un sistema para incluirlos en los diagramas de secuencia se logran de la revisión de los casos de uso.

- Un contrato puede emplearse con una operación de alto nivel que se aplica al sistema entero o con un método de bajo nivel de una clase particular.

- Los diagramas de estado resultan adecuados para describir el comportamiento de un objeto a través de diferentes casos de uso, sin embargo, no resultan del todo adecuados para describir el comportamiento que incluye a una serie de objetos colaborando entre sí. Por lo tanto, resulta útil combinar los diagramas de estado con otras técnicas. Por ejemplo, los diagramas de interacción son idóneos para la descripción del comportamiento de varios objetos en un único caso de uso, y los diagramas de actividades muestran de forma adecuada la secuencia general de acciones en diferentes objetos y casos de uso.

3.9 Recomendaciones

- Una técnica muy útil para identificar conceptos por su simplicidad propuesta en [Abbot83] consiste en identificar las frases nominales[5] en las descripciones textuales del dominio de un problema y considerarlas conceptos o atributos idóneos.

- Este método hay que utilizarlos con mucha prudencia; no es posible encontrar mecánicamente correspondencias entre sustantivo y concepto, y además las palabras del lenguaje natural son ambiguas. Los casos expandidos de uso son una excelente descripción en donde puede aplicarse esta técnica.

[5] Frase nominal [A. *Nominalsatz*]. Frase que no tiene verbo ni cópula: *omnia präclara, rara; perro ladrador, poco mordedor*. La naturaleza de la frase nominal está sometida a discusión. Hjelmslev (1948) sostiene que hay en ella tres elementos verbales implícitos: aspecto *infectum*, tiempo presente y modo indicativo. Benveniste (1950), por el contrario, afirma plausiblemente que la frase nominal es, por naturaleza, no temporal, no modal y no aspectual. Separa (como Hjelmslev) la frase nominal de la oración atributiva, y escribe: „Desde el punto de vista indoeuropeo, la segunda no es una variante más clara o más completa de la primera, ni la primera una forma deficiente de la segunda. Una y otra son posibles, pero no para la misma expresión. Una aserción nominal, completa en sí, coloca el enunciado fuera de toda localización temporal o modal y fuera de la subjetividad del hablante… Tiene valor de argumento, de prueba, de referencia. Se introduce en el discurso para mover y convencer, no para informar. Es, fuera del tiempo, de las personas y de la circunstancia, una verdad proferida como tal". Por ello se utiliza preferentemente en frases sentenciosas y proverbios, si bien en indoeuropeo, en griego y aun en latín, tuvo una vitalidad que excedía en mucho los estrechos marcos modernos de su empleo. "
[Lázaro Carreter, *Diccionario de términos*…, 294 s.]

- En todo análisis de un sistema, antes de iniciar el diseño lógico de cómo funcionara una aplicación de software, es necesario investigar y definir su comportamiento como una “caja negra”.

- Para preparar un contrato en los casos de uso:

  1. Identifique las operaciones del sistema a partir de los diagramas del sistema

  2. Elabora un contrato en cada operación del sistema

  3. Comience redactando la sección de *Responsabilidades*; después describa informalmente el propósito de la operación

  4. Complete luego la sección de Poscondiciones, describiendo en forma declarativa los cambios de estado de los objetos en el modelo conceptual

  5. Para describir las postcondiciones utilice las siguientes categorías:

     - Creación y eliminación de instancias

     - Modificación de los atributos

     - Asociaciones formadas y canceladas

- No nos debemos plantear el diseñar diagramas de estados para todas las clases en el sistema, sino sólo para aquellas que exhiban un comportamiento interesante de forma que la elaboración del diagrama de estados nos ayude a entender dicho comportamiento.

CAPITULO 4. DISEÑO ORIENTADO A OBJETOS

4.1 Introducción

El objetivo fundamental de la fase de diseño es obtener una solución lógica del sistema, que se funda en el paradigma de la programación orientada a objetos.

Los casos de uso realizados en la fase de análisis, describe el diseño concreto del caso de uso a partir de una tecnología de entrada y salida, así como de su implementación global, por lo que primero vamos a diseñar la interfaz de usuario en función de los casos de uso del BackOffice y de las operaciones PULL. En los contratos de colaboración desarrollados en la fase de análisis se incluye una primera conjetura óptima sobre las poscondiciones referentes al inicio de operaciones del sistema, sin embargo los contratos no muestran una solución de cómo los objetos de software van a cumplir con ellas. Se va a diseñar en UML los diagramas de interacción que explican gráficamente como los objetos interactúan a través de mensajes para realizar las tareas. Una vez terminados los diagramas de interacción vamos a identificar la especificación de las clases de software (y las interfaces) que participan en la solución del software y complementarlas con detalles de diseño de métodos y funciones tomando en consideración la definición de arquitectura multicapa utilizada en este sistema. Debo mencionar que gran parte del trabajo de este tema de graduación se plasmará en el diseño de mensajería realizada para las operaciones PULL y PUSH que se encuentra en el método Prctrn de objeto ServiciosElectronicos.Interfaz. Una vez diseñadas las clases iniciamos con el diseño de la base de datos y generaremos la documentación completa de la base de datos denominada dbServiciosSE que se encuentra plasmada en el diagrama Objeto-relación y en documentación del diccionario de base de datos. Por último definiremos la arquitectura del sistema a implementar desde diferentes vistas para que el desarrollador tenga una gran visión de lo que necesitamos construir. Al finalizar el capítulo me permitiré realizar unas conclusiones y recomendaciones rescatando las mejoras prácticas encontradas al escribir este capítulo.

4.2 Diseño de interfaz de usuario.

A continuación se presenta el diseño de las interfaces de usuario requeridas para el acceso a las operaciones SMS banca celular y las interfaces de usuario para el BackOffice.

4.2.1 *Diseño de mensajes SMS para las operaciones PULL.*

Iniciamos por presentar la sintaxis que hemos utilizado para diseñar las operaciones banca celular tipo PULL y que se ponen a disposición de los clientes para el acceso al servicio.

Sintaxis de un comando SMS

La sintaxis de un comando SMS esta dada por la siguiente estructura.

Nemónico par1 par1 ….par n

En donde,

Nemónico, es una ayuda para la memoria. Cualquier código que ayude a la persona de captura de datos recuerde la manera de teclear la operación, o que el usuario final recuerde como usar la información, puede ser considerado un nemónico, usando una combinación de letras y símbolos se logra una forma clara para codificar un producto de tal forma que el código sea visto y comprendido fácilmente. Para este producto en cada operación se utiliza un nemónico largo y nemónico corto. Por ejemplo para suscribirse al servicio banca celular se definen los nemónicos solicito o sol.

Par1, es un parámetro opcional o requerido que necesita la operación para poder ser ejecutada con éxito.

Si nos referimos a la tabla 4.1 en donde se encuentra la explicación de las operaciones PULL implementadas para este trabajo de graduación podemos observar que nos indica que para la operación de suscripción banca celular el usuario debe ingresar el siguiente mensaje.

Solicito + (N° Tarjeta Débito/Crédito) + Clave Tarjeta
O
Sol + (N° Tarjeta Débito/Crédito) + Clave Tarjeta

Entonces el mensaje que debe enviar el usuario es el siguiente:

“Solicito 4931760010080452 1425”

TARJETA DÉBITO			Ejemplo
Para consultar o solicitar	**Nemónico largo …escriba la palabra..**	**Nemónico corto …o escriba la abreviatura**	**Nem Corto**
Solicitud servicio BancaCelular	Solicito + (N° Tarjeta Débito/Crédito) + Clave Tarjeta	SOL + (N° Tarjeta Débito/Crédito) + Clave Tarjeta	SOL 4931760120080245 1111
Cambio de Clave de servicios electrónicos	Clave + (6 últimos números de tarjeta)* + (Clave anterior**) + (Nueva Clave)	CLA + (6 últimos números de tarjeta)* + (Clave anterior**) + (Nueva Clave)	CLA 080245 1111 1100
Saldos de todas cuentas	Saldos	S	SALDOS
Saldos de Cuenta Corriente	SaldosCC	SCC	SALDOSCC
Saldos de Cuenta Ahorros	SaldosCA	SCA	SALDOSCA
Consulta de Cheque cobrado	Cheque + (6 últimos números de tarjeta)* + (Clave actual) + Número de Cheque	CHQ + (6 últimos números de tarjeta)* + (Clave actual) + Número de cheque	CHEQUE 080245 1100 20002141
Aviso de cheque cobrado desde $	Adiciona + (6 últimos números de	ADI + (6 últimos números de tarjeta)*	ADICIONA 080245 1100 AVCHC

2000 en adelante	tarjeta)* + (Clave actual) + AVCHC	+ (Clave actual) + AVCHC	
TARJETA CREDITO			
Saldos a Pagar Tarjeta Crédito Visa/MasterCard	SaldoTC + (6 últimos números de tarjeta)* + (Clave actual)	STC + (6 últimos números de tarjeta)* + (Clave actual)	SALDOTC 080245 1100
Consulta de Cupo Tarjetas de Crédito Visa/MasterCard	CupoTC + (6 últimos números de tarjeta)* + (Clave actual)	CTC + (6 últimos números de tarjeta)* + (Clave actual)	CUPOTC 080245 1100
Notificación de consumos de tarjeta de crédito	Adiciona + (6 últimos números de tarjeta)* + (Clave actual) + NOCTC	ADI + (6 últimos números de tarjeta)* + (Clave actual) + NOCTC	ADICIONA 080245 1100 NOCTC
TARJETA DÉBITO/CREDITO			
Notificación de retiros o avances en efectivo en Cajeros Automáticos de tarjeta Débito/Crédito	Adiciona + (6 últimos números de tarjeta)* + (Clave actual) + NOATM	ADI + (6 últimos números de tarjeta)* + (Clave actual) + NOATM	ADICIONA 080245 1100 NOATM
Bloqueo temporal de la tarjeta de débito o crédito	BLOQUEO + (6 últimos números de tarjeta)* + (Pin Tarjeta) + (Clave actual)	BLQ + (6 últimos números de tarjeta)* + (Pin Tarjeta) + (Clave actual)	BLOQUEO 080245 1111 1100

Tabla 4.1 Diseño de comandos SMS para operaciones tipo PULL banca celular

4.2.2 *Diseño de interfaces de usuario para el BackOffice.*

En esta sección se presenta el diseño de interfaces de usuario para el BackOffice de acuerdo a los diferentes casos de uso.

4.2.2.1 Control de acceso usuario atención al cliente

Figura 4.1 Diseño de pantalla de seguridad

4.2.2.2 Control de acceso usuario supervisor

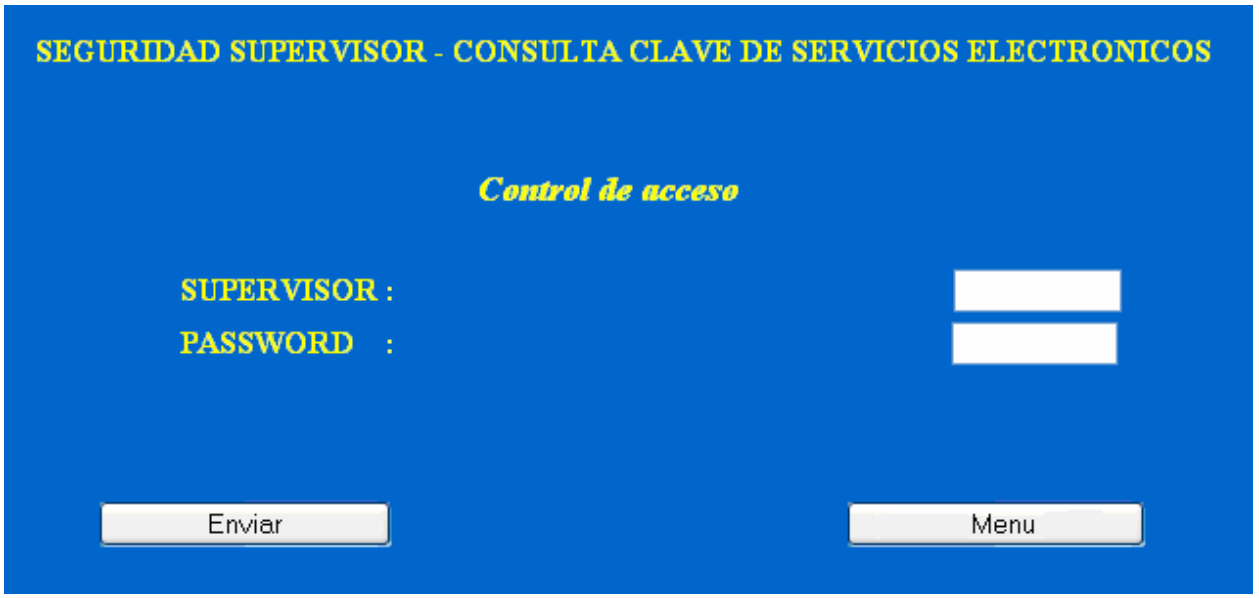


Figura 4.2 Diseño pantalla de seguridad para supervisor

4.2.2.3 Suscripción servicio banca celular

Figura 4.3 Diseño de pantalla de suscripción

4.2.2.4 Modificación de teléfono.

Figura 4.4 Diseño de pantalla de modificación de teléfono

4.2.2.5 Consulta suscripciones

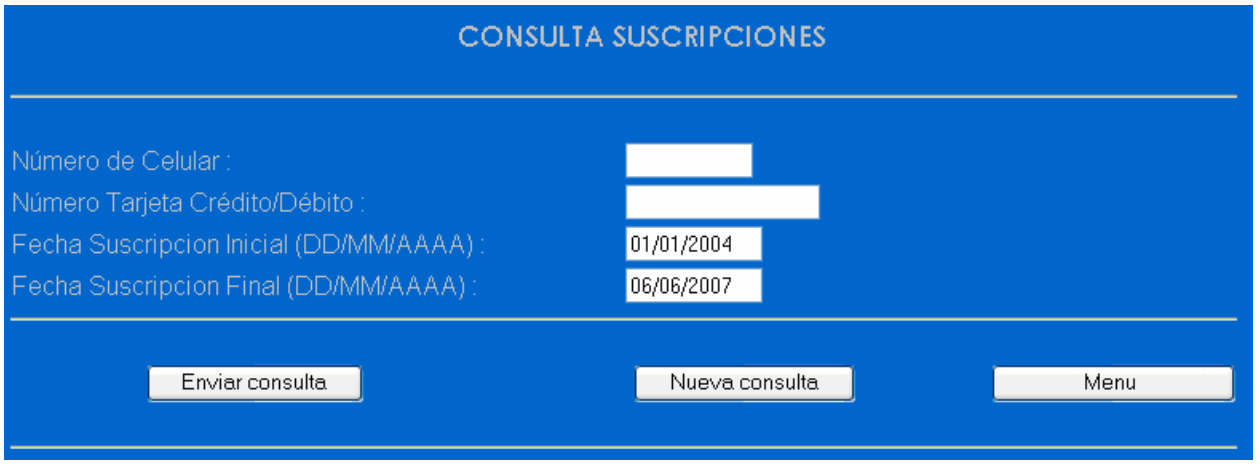


Figura 4.5 Diseño de pantalla para consulta de suscripciones

4.2.2.6 Consulta servicios adicionales de una suscripción

Figura 4.6 Diseño de pantalla para consulta de servicios adicionales de un cliente

4.2.2.7 Suscripción a un servicio adicional de un cliente

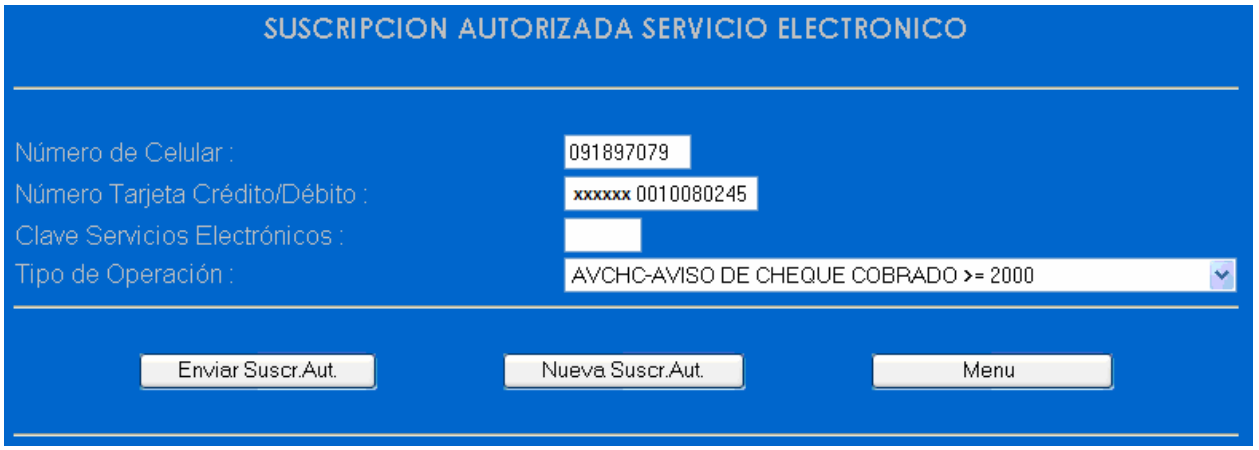


Figura 4.7 Diseño de pantalla para la suscripción de un servicio adicional

4.2.2.8 Inactiva suscripción banca celular

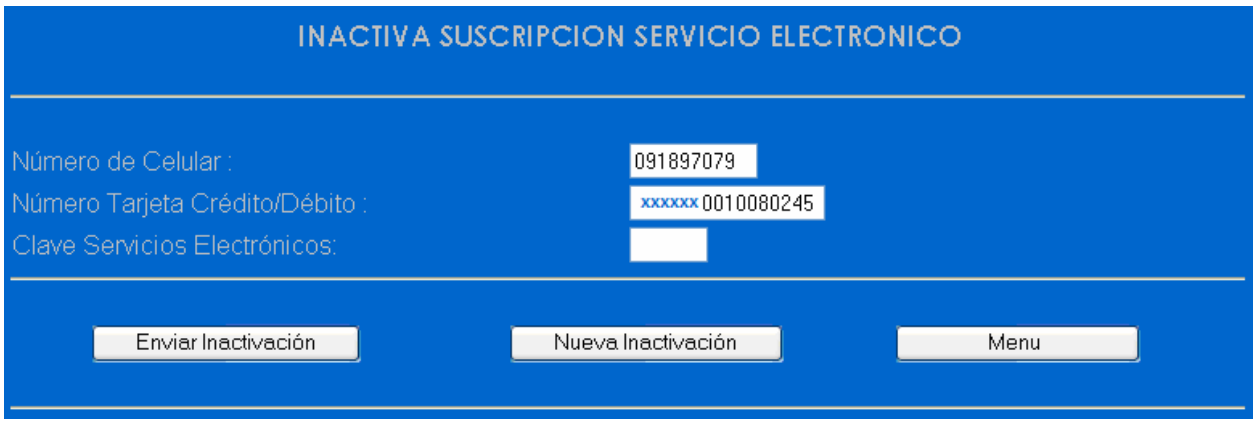


Figura 4.8 Diseño de pantalla para inactivación del servicio banca celular

4.2.2.9 Inactiva suscripción de un servicio adicional

Figura 4.9 Diseño de pantalla para inactivación de un servicio adicional

4.2.2.10 Cambio de clave de servicio banca celular

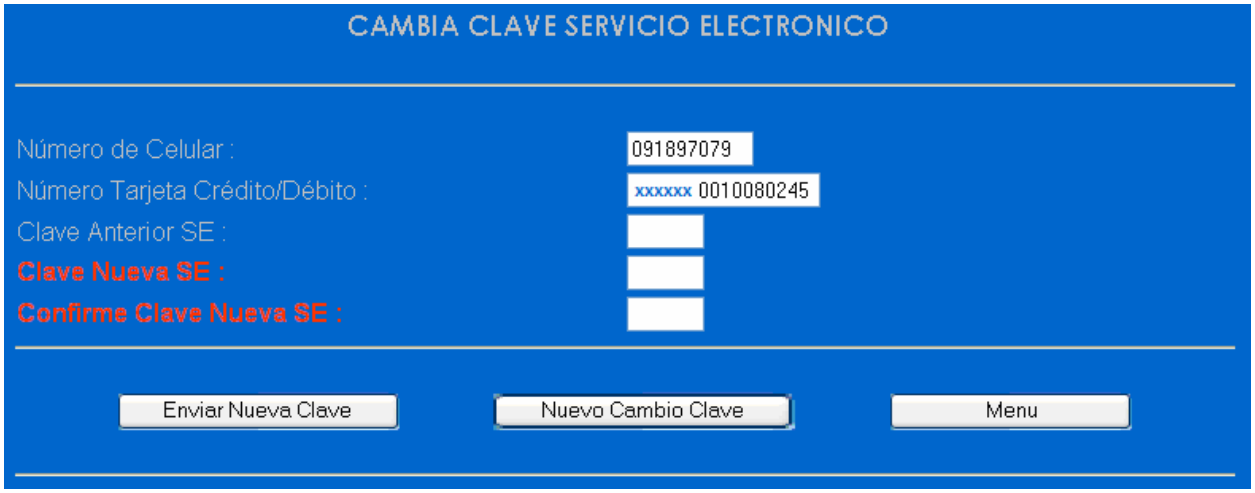


Figura 4.10 Diseño de pantalla para el cambio de clave del servicio banca celular

4.2.2.11 Consulta transacciones banca celular

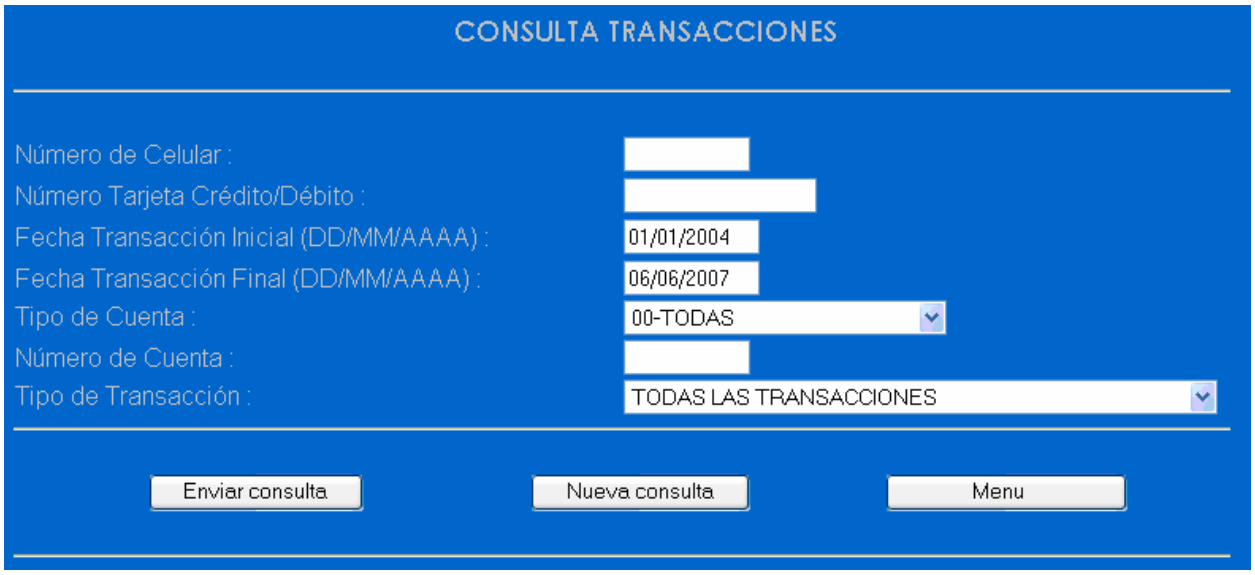


Figura 4.11 Diseño de pantalla para la consulta de transacciones banca celular

4.2.2.12 Consulta clave de servicio banca celular actual

Figura 4.12 Diseño de pantalla de consulta de clave de servicio banca celular

4.2.2.13 Reverso de suscripción banca celular

Figura 4.13 Diseño de pantalla para el reverso de una suscripción banca celular

4.2.2.14 Reverso suscripción de un servicio adicional

Figura 4.14 Diseño de pantalla para el reverso de una suscripción de un servicio adiciona

4.3 Diagramas de interacción

Los diagramas de interacción explican gráficamente como los objetos interactúan a través de mensajes para realizar las tareas. Explican gráficamente las interacciones existentes entre las instancias de los objetos en un formato de grafo o de red. La consideración cuidadosa de la asignación de responsabilidades y el diseño de la colaboración entre los objetos. Los diagramas de interacción no se pueden preparar si no se han generado los artefactos *Modelo conceptual* y *Contratos de operación del sistema*.

Modelo conceptual, a partir de este modelo el diseñador podrá definir las clases de software correspondientes a los conceptos. Los objetos de las clases participan en las interacciones que se describen gráficamente en los diagramas.
Contratos de operación del sistema, a partir de ellos el diseñador identifica las responsabilidades y las poscondiciones que han de llenar los diagramas de interacción.

En el anexo 4.1 se muestran los diagramas de interacción entre objetos para las transacciones en general de BACKOFFICE, PUSH, PULL.

4.4 Diagramas de clases

Este diagrama describe gráficamente las especificaciones de la clase de software y de las interfaces en una aplicación. Durante el diseño, el diagrama de clase se elabora para tener en cuenta los detalles concretos de la implementación del sistema.

Normalmente contiene la siguiente información:

- Clases, asociaciones, atributos.
- Interfaces con sus operaciones y constantes.
- Métodos.
- Información sobre los tipos de atributos
- Navegabilidad.
- Dependencias.

4.4.1 *Arquitecturas Multicapas*

Una vez concienciados del diseño, estableceremos la arquitectura del sistema. Esto incluye establecer si será un sistema simple diseñado para correr en una sola máquina, un sistema 'two-tiered' consistente en un cliente y un servidor, o un sistema 'multi-tiered' con objetos interfaz de usuario separados de los objetos 'business', separado de la base de datos, cada uno corriendo en plataformas distintas, para el caso de este trabajo de graduación escogemos un diseño ‘multi-tiered’ en donde vamos a separar la interfaz de usuario en el objeto ServiciosElectronicos y las reglas del negocio en el objeto ServiciosOperacion.

Una aproximación a dirigir el diagrama de clase para un sistema complejo es separar el diagrama en secciones que muestren la lógica de la aplicación, el diseño de la interfaz de usuario, y las clases implicadas con el almacenamiento de los datos. Esto se puede hacer físicamente segmentando el diagrama de clase, usando diagramas separados para cada sección, o simplemente añadiendo una propiedad a cada clase que 'tracks' cada 'tier' al cual pertenece.

A continuación se presenta el diagrama de paquetes.

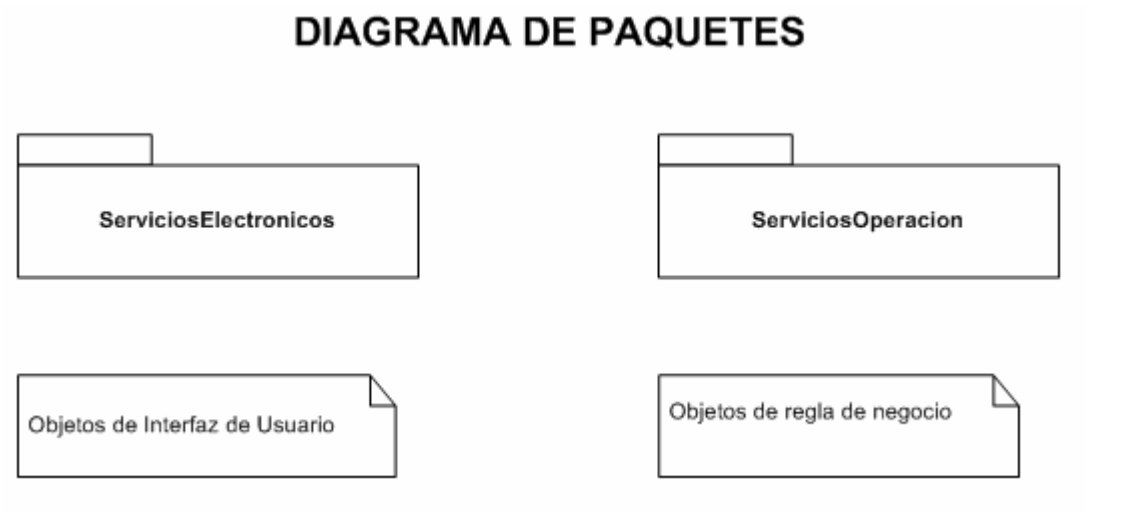


Figura 4.15 Diagrama de sistema 'multi-tiered' con objetos interfaz de usuario separados de los objetos 'business', que representa la arquitectura multicapas

4.4.2 *Diseño de Componentes*

Un componente es un grupo de objetos o componentes más pequeños que interaccionan entre ellos y se combinan para dar un servicio. Un componente es similar a una caja negra, en la cual los servicios del componente se especifican por su interfase o interfaces, sin ofrecer conocimiento del diseño e implementación internas del componente. El desarrollo basado en componentes es el proceso de ensamblar la combinación correcta de componentes en la configuración correcta para llevar acabo la funcionalidad deseada para un sistema. Los componentes se representan en el diagrama de clases de UML especificando la interfaz de una clase o paquete. Hay dos notaciones para mostrar una interfaz - una es mostrar la interfaz como una 'regular class symbol' con el estereotipo "interfaz", con una lista de operaciones soportadas por esta interfaz, detalladas en el 'operation department' (departamento de operación). La segunda alternativa es mostrar la interfaz como un círculo pequeño junto con la clase con una línea sólida, con el nombre de la interfaz en el círculo.

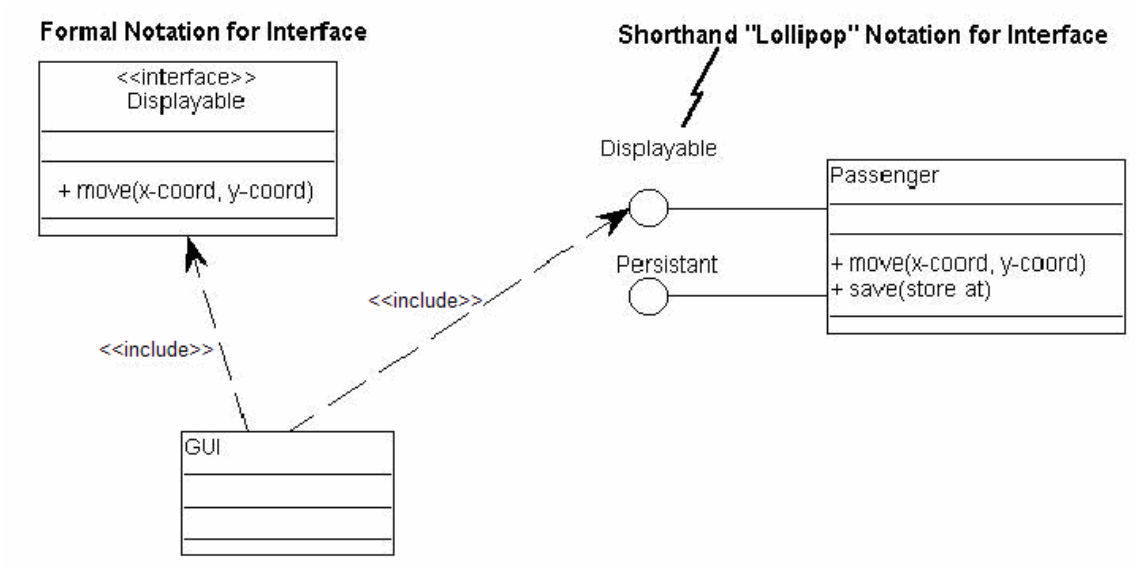


Figura 4.16 Notaciones para mostrar una interfaz de una clase de manera gráfica

Para el presente trabajo vamos a utilizar la notación formal. En la Figura 4.17 se muestra el diagrama de componentes diseñados para el sistema Banca Celular.

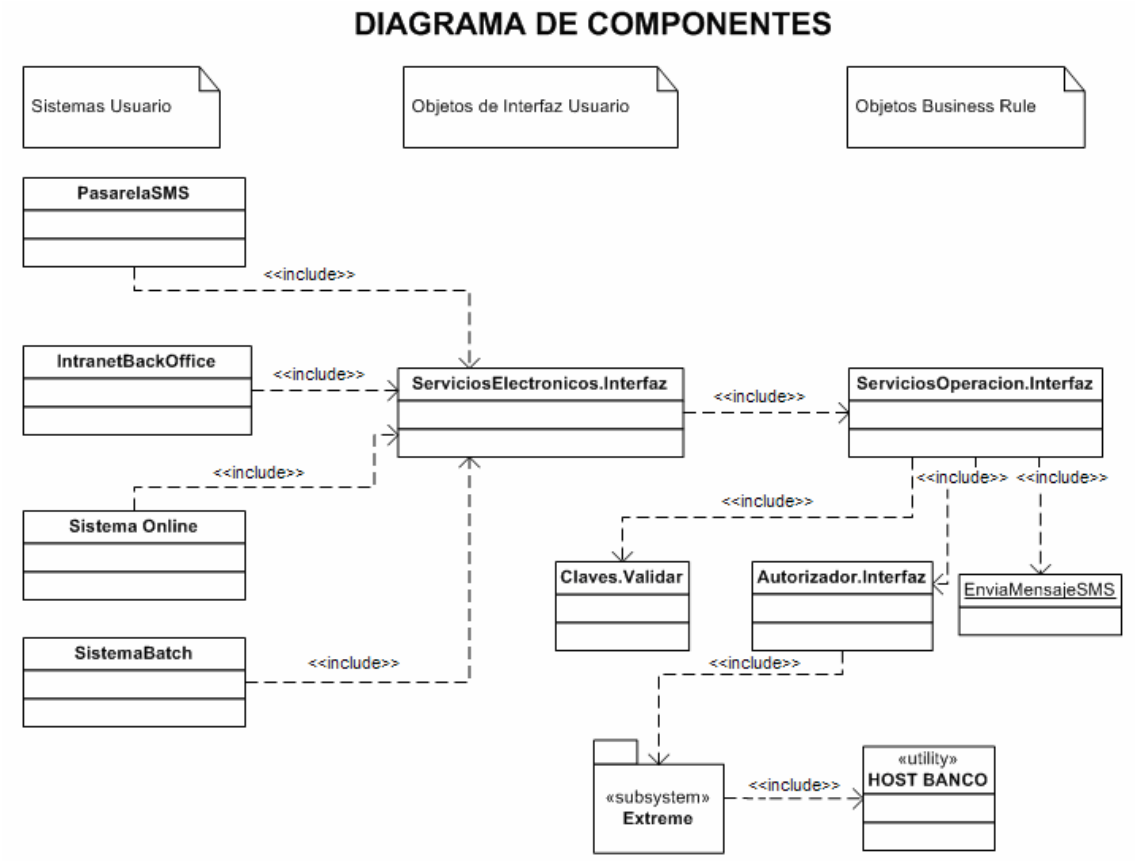


Figura 4.17 Diagrama de componentes del sistema Banca Celular

4.4.3 *Clase ServiciosElectronicos*

Esta clase permite dentro de una arquitectura de capas, provee de las interfaces necesarias para que los diferentes sistemas de usuario puedan acceder a las funciones requeridas para banca celular. Dentro de los sistemas usuarios que consumen esta interfaz esta la pasarela SMS, sistema de Intranet para el BackOffice, sistemas automáticos de notificación PUSH.

4.4.3.1 Diagrama y descripción de métodos de la clase ServiciosElectronicos

El diagrama de estructura estática se ha generado con Visio 2003 con lenguaje UML. Refiérase al anexo 4.2 en donde se muestra el diagrama de clase del objeto ServiciosElectronicos.Interfaz.

A continuación se presenta la documentación técnica de los diferentes métodos implementados para el acceso a las funciones de BackOffice y a las diferentes funciones de las operaciones PULL y PUSH.

4.4.3.2 Métodos de BackOffice

En esta sección se presenta la descripción de los métodos que la clase ServiciosElectronicos.Interfaz que se pone a disposición del front end para realizar operaciones de BackOffice del servicio banca celular.

4.4.3.2.1 Método SuscripcionSE().

Descripción: Función que permite suscribir a un cliente al servicio banca celular desde otros canales.

Visibilidad: Público.

Sintaxis:

Public Function SuscripcionSE(

ByVal TerNumCel, _

ByVal TerCodSer, _

ByVal TerCodRedAdq, _

ByVal TerIdeDisOri, _

ByVal TerUsuOri, _

ByVal TerNumTar, _

ByVal TerPin) As String

Parámetros de entrada:

TerNumCel(9) : Número de celular.

TerCodSer(3): Código de servicio = “SMS”.

TerCodRedAdq(3): Código de red adquiriente = “ITR” significa que es desde la intranet.

TerIdeDisOri(15): Identificación de dispositivo adquiriente donde se origina la transacción Ej.: PCDLandy (Puede ser el nombre o IP de la maquina origen)

UsuOri(10): Nombre del usuario que origina la transacción.

Tarjeta(16): Número de la tarjeta débito/crédito.

Pin(4) : Clave de la tarjeta débito/crédito.

Parámetro de salida: Función SuscripcionesSE() as string

Esta función devuelve la siguiente cadena de caracteres como resultado.

Retorno(3): Código de retorno

Si Retorno(3) <> “000”. A continuación viene descripción del error en los siguientes (145) caracteres.

Si Retorno(3) = “000” significa transacción OK

Viene a continuación los siguientes campos:

Identificación(13): Número de Ced/Ruc/Pas del Cliente

Producto(2): Tipo del producto (Tarjeta de Suscripción) del cliente

TD, Tarjeta de Débito

VI, Tarjeta VISA

MC, Tarjeta MasterCard

TC, Tarjeta Club

Nombre(30) : Nombre del tarjetahabiente

TipTarHab(1): “N”=Normal, “E”=Empleado, “V”=VIP, “P”=VIP Plus

TerEsLocal(1): “0” = Cooperativa, “1” = Institución local

Ejemplo de una transacción OK:
”0000102777091 TDJUAN DIEGO LANDY E1”
Ejemplo de una transacción con ERROR:
”001TARJETA CANCELADA”

4.4.3.2.2 Método: CambiaClaveSE().

Descripción: Función que permite cambiar la clave del servicio banca celular desde otros canales como la Intranet.

Visibilidad: Público.

Sintaxis:

```
Public Function CambiaClaveSE(
ByVal TerNumCel, _
ByVal TerCodSer, _
ByVal TerCodRedAdq, _
ByVal TerIdeDisOri, _
ByVal TerUsuOri, _
ByVal TerNumTar, _
ByVal TerClaveAntSE, _
ByVal TerClaveNueSE) As String
```

Parámetros de entrada:

TerNumCel(9) : Número de celular.
TerCodSer(3): Código de servicio = “SMS”.
TerCodRedAdq(3): Código de red adquiriente. Por ejemplo “ITR=INTRANET”
TerIdeDisOri(15): Identificación de dispositivo adquiriente donde se origina la transacción. Ejemplo: PCDLandy, que puede ser el nombre del PC o la dirección IP desde donde se origina el requerimiento.
UsuOri(10): Nombre del usuario que origina la transacción. Ejemplo el usuario de la intranet “dlandy”.
Tarjeta(16): Número de la tarjeta débito/crédito.
TerClaveAntSE(4): Clave de servicio banca celular actual.

TerClaveNueSE(4): Nueva clave de servicio banca celular con formato numérico de 4 dígitos.

Parámetro de salida: Función CambiaClaveSE() as string

Esta función devuelve la siguiente cadena de caracteres como resultado.

Retorno(3): Código de retorno

Si Retorno(3) <> “000” Transacción ERROR. A continuación viene descripción del error en los siguientes (145) caracteres.

Si Retorno(3) = “000” Transacción OK.

Viene a continuación los siguientes campos:

Identificación(13): Número de Ced/Ruc/Pas del Cliente

Producto(2): Tipo del producto (Tarjeta de Suscripción) del cliente

TD, Tarjeta de Débito

VI, Tarjeta VISA

MC, Tarjeta MasterCard

TC, Tarjeta Club

Nombre(30) : Nombre del tarjetahabiente

TipTarHab(1): “N”=Normal, “E”=Empleado, “V”=VIP, “P”=VIP Plus

TerEsLocal(1): “0” = Cooperativa, “1” = Institución local

Ejemplo de una transacción OK:

” 0000102777091 TDJUAN DIEGO LANDY E1”

Ejemplo de una transacción con ERROR:

”001TARJETA CANCELADA”

4.4.3.2.3 Método: ConsultaClaveSE().

Descripción: Método que realiza la consulta de la clave actual del servicio banca celular.

Visibilidad: Público.

Sintaxis:

Public Function ConsultaClaveSE(

ByVal TerNumCel, _

ByVal TerCodSer, _

ByVal TerCodRedAdq, _
ByVal TerIdeDisOri, _
ByVal TerUsuOri, _
ByVal TerNumTar, _
ByVal TerSupOri, _
ByVal TerSupNom) As String

Parámetros de entrada:

TerNumCel(9) : Número de celular.
TerCodSer(3): Código de servicio = “SMS”.
TerCodRedAdq(3): Código de red adquiriente. Si el front end es intranet colocar “ITR”.
TerIdeDisOri(15): Identificación de dispositivo adquiriente donde se origina la transacción Ejemplo: PCDLandy (Puede ser el nombre del PC o la dirección IP)
UsuOri(10): Nombre del usuario que origina la transacción.
Ejemplo el usuario de la intranet “dlandy”
Tarjeta(16): Número de la tarjeta débito/crédito
TerSupOri(10): Código del supervisor que autoriza la consulta de la clave del servicio banca celular. Ejemplo: Puede ser el código de un usuario supervisor de la intranet.
TerSupNom(40): Nombre del usuario supervisor.

Parámetro de salida: Función ConsultaClaveSE() as string

Esta función devuelve la siguiente cadena de caracteres como resultado.
Retorno(3): Código de retorno
Si Retorno(3) <> “000”. A continuación viene descripción de error en un los siguientes (145) caracteres.
Si Retorno(3) = “000” Transacción OK.
Viene a continuación los siguientes campos:
Identificación(13): Número de Ced/Ruc/Pas del Cliente
Producto(2): Tipo del producto (Tarjeta de Suscripción) del cliente

TD, Tarjeta de Débito
VI, Tarjeta VISA
MC, Tarjeta MasterCard
TC, Tarjeta Club

Nombre(30) : Nombre del tarjetahabiente
TipTarHab(1): “N”=Normal, “E”=Empleado, “V”=VIP, “P”=VIP Plus
TerEsLocal(1): “0” = Cooperativa, “1” = Institución local
TerClave(4): Número de la clave de los servicios electrónicos.

Ejemplo de una transacción OK:
:”0000102777091 TDJUAN DIEGO LANDY E12556”
Ejemplo de una transacción con ERROR:
”002SUSCRIPCION AL SERVICIO BANCA CELULAR INACTIVA”

4.4.3.2.4 Método: SuscripcionAutorizadaSE().

Descripción: Función que realiza la consulta de la clave de los servicios electrónicos. En este caso SMS.

Visibilidad: Público.

Sintaxis:

Public Function SuscripcionAutorizadaSE(
ByVal TerNumCel, _
ByVal TerCodSer, _
ByVal TerCodRedAdq, _
ByVal TerIdeDisOri, _
ByVal TerUsuOri, _
ByVal TerNumTar, _
ByVal TerClaveSE, _
ByVal TerCodOpe) As String

Parámetros de entrada:

TerNumCel(9) : Número de celular.
TerCodSer(3): Código de servicio = “SMS”.
TerCodRedAdq(3): Código de red adquiriente. Si el front end es intranet colocar “ITR”.
TerIdeDisOri(15): Identificación de dispositivo adquiriente donde se origina la

transacción. Ejemplo: PCDLandy (Puede ser el nombre del PC o dirección IP de la maquina origen)
UsuOri(10): Nombre del usuario que origina la transacción. Ejemplo el usuario de la intranet “dlandy”
Tarjeta(16): Número de la tarjeta débito/crédito
ClaveSE(4): Número de la clave de los servicios electrónicos.
TerCodOpe(5): Código de operación o servicio adicional, se explica a continuación como recuperar este código(s) con la función OperacionAutorizada.

Parámetro de salida: Función SuscripcionAutorizadaSE() as string
Esta función devuelve la siguiente cadena de caracteres como resultado.
Retorno(3): Código de retorno
Si Retorno(3) <> “000”. A continuación viene descripción de error en un los siguientes (145) caracteres.
Si Retorno(3) = “000”. Transacción OK.
Viene a continuación los siguientes campos:
Identificación(13): Número de Ced/Ruc/Pas del Cliente
Producto(2): Tipo del producto (Tarjeta de Suscripción) del cliente
TD, Tarjeta de Débito
VI, Tarjeta VISA
MC, Tarjeta MasterCard
TC, Tarjeta Club
Nombre(30) : Nombre del tarjetahabiente
TipTarHab(1): “N”=Normal, “E”=Empleado, “V”=VIP, “P”=VIP Plus
TerEsLocal(1): “0” = Cooperativa, “1” = Institución local

Ejemplo de una transacción OK:
”0000102777091 TDJUAN DIEGO LANDY E12556”
Ejemplo de una transacción con ERROR:
”001TARJETA CANCELADA”

4.4.3.2.5 Método: OperacionAutorizada().

Descripción: Función que realiza la consulta de los códigos de operaciones autorizadas o servicios adicionales que se tienen disponibles para un bin de tarjeta específica.

Visibilidad: Público.

Sintaxis:

Public Function OperacionAutorizada(
ByVal TerNumTar, _
ByRef RespIsTranOk, _
ByRef RespData) As ADODB.Recordset

Parámetros de entrada:
Tarjeta(16): Bin de la tarjeta de débito/crédito

Parámetro de salida:
RespIsTranOk: boolean, indica si la consulta fue o no exitosa.
RespData(145): Si la variable RespIsTranOk = False, este parámetro devuelve la descripción del mensaje de error:

Función OperaciónAutorizada() as ADODB.Recordset
Campos del recordset:
TO-CodOpe (5): Código de operación autorizada.
TT-Des (50): Descripción de la operación autorizada.
TT-TieCos (1): Si es = “1”, tiene comisión para la institución financiera adquiriente, “0”= no tiene costo.
TT-AdqCosNor (float): Comisión del adquiriente para clientes normales.
TT-AdqCosVip (float): Comisión del adquiriente para clientes vip.
TT-AdqCosVipPlus (float): Comisión del adquiriente para clientes vip plus.
TT-AdqCosEmp (float): Comisión del adquiriente para empleados.

4.4.3.2.6 Método: InactivaSuscripcionSE().

Descripción: Función que realiza la inactivación de la suscripción de los servicios banca celular de un tarjeta habiente.

Visibilidad: Público.

Sintaxis:

Public Function InactivaSuscripcionSE(
ByVal TerNumCel, _
ByVal TerCodSer, _
ByVal TerCodRedAdq, _
ByVal TerIdeDisOri, _
ByVal TerUsuOri, _
ByVal TerNumTar, _
ByVal TerClaveSE, _
ByVal TerCodMot) As String

Parámetros de entrada:

TerNumCel(9) : Número de celular.
TerCodSer(3): Código de servicio = “SMS”.
TerCodRedAdq(3): Código de red adquiriente. Si el front end es intranet colocar “ITR”.
TerIdeDisOri(15): Identificación de dispositivo adquiriente donde se origina la transacción. Ejemplo: PCDLandy (Puede ser el nombre del PC o dirección IP)
UsuOri(10): Nombre del usuario que origina la transacción.
Tarjeta(16): Número de la tarjeta de débito/crédito.
ClaveSE(4): Clave del servicio banca celular.
TerCodMot(2): Código de motivo por el que esta inactiva la suscripción.

Parámetro de salida: Función InactivaSuscripcionSE() as string
Esta función devuelve la siguiente cadena de caracteres como resultado.
Retorno(3): Código de retorno
Si Retorno(3) <> “000”. A continuación viene descripción de error en un los siguientes (145) caracteres.
Si Retorno(3) = “000” Transacción OK.
Viene a continuación los siguientes campos:
Identificación(13): Número de Ced/Ruc/Pas del Cliente
Producto(2): Tipo del producto (Tarjeta de Suscripción) del cliente
TD, Tarjeta de Débito

VI, Tarjeta VISA
MC, Tarjeta MasterCard
TC, Tarjeta Club
Nombre(30) : Nombre del tarjetahabiente
TipTarHab(1): “N”=Normal, “E”=Empleado, “V”=VIP, “P”=VIP Plus
TerEsLocal(1): “0” = Cooperativa, “1” = Institución local

Ejemplo:”0000102777091 TDJUAN DIEGO LANDY E1”
Ejemplo de una transacción con ERROR:
”003NO EXISTE SUSCRIPCION ACTIVA”

4.4.3.2.7 Método: InactivaSusAutSE().

Descripción: Función que realiza la inactivación de la suscripción autorizada o servicio adicional al servicio banca celular.

Visibilidad: Público.

Sintaxis:

```
Public Function InactivaSusAutSE(
ByVal TerNumCel, _
ByVal TerCodSer, _
ByVal TerCodRedAdq, _
ByVal TerIdeDisOri, _
ByVal TerUsuOri, _
ByVal TerNumTar, _
ByVal TerClaveSE, _
ByVal TerCodOpe,
ByVal TerCodMot) As String
```

Parámetros de entrada:

TerNumCel(9) : Número de celular.
TerCodSer(3): Código de servicio = “SMS”.

TerCodRedAdq(3): Código de red adquiriente. Si el front end es intranet colocar "ITR".

TerIdeDisOri(15): Identificación de dispositivo adquiriente donde se origina la transacción. Ejemplo: PCDLandy (Puede ser el nombre del PC o IP de la maquina origen)

UsuOri(10): Nombre del usuario que origina la transacción.

Tarjeta(16): Número de la tarjeta Débito/Crédito

ClaveSE(4) : Clave de los servicios electrónicos

TerCodOpe(5) : Código de operación autorizada.

TerCodMot(2) : Código del motivo por el que esta inactivo el servicio adicional.

Parámetro de salida: Función InactivaSusAutSE() as string

Esta función devuelve la siguiente cadena de caracteres como resultado.

Retorno(3): Código de retorno

Si Retorno(3): <> "000" A continuación viene descripción de error en un los siguientes (145) caracteres.

Si Retorno(3): "000" Transacción OK

Viene a continuación los siguientes campos:

Identificación(13): Número de Ced/Ruc/Pas del Cliente

Producto(2): Tipo del producto (Tarjeta de Suscripción) del cliente

TD, Tarjeta de Débito

VI, Tarjeta VISA

MC, Tarjeta MasterCard

TC, Tarjeta Club

Nombre(30) : Nombre del tarjetahabiente

TipTarHab(1): "N"=Normal, "E"=Empleado, "V"=VIP, "P"=VIP Plus

TerEsLocal(1): "0" = Cooperativa, "1" = Institución local

Ejemplo de una transacción OK:

"0000102777091 TDJUAN DIEGO LANDY E1"

Ejemplo de una transacción con ERROR:

"004SERVICIO ADICIONAL NO EXISTE"

4.4.3.2.8 Método: ReversaSuscripcionSE().

Descripción: Función que realiza el reverso de la suscripción y comisión banca celular. En caso de tener suscripciones adicionales también se produce el reverso de las mismas. La transacción de reverso se puede realizar únicamente en el mismo día de la suscripción, siempre y cuando el usuario no haya realizado transacciones tipo PULL desde el celular de suscripción.

Visibilidad: Público.

Sintaxis:

```
Public Function ReversaSuscripcionSE(
ByVal TerNumCel, _
ByVal TerCodSer, _
ByVal TerCodRedAdq, _
ByVal TerIdeDisOri, _
ByVal TerUsuOri, _
ByVal TerNumTar, _
ByVal TerClaveSE, _
ByVal TerSupOri, _
ByVal TerSupNom) As String
```

Parámetros de entrada:

TerNumCel(9) : Número de celular.

TerCodSer(3): Código de servicio = “SMS”.

TerCodRedAdq(3): Código de red adquiriente. Si el front end es intranet colocar “ITR”.

TerIdeDisOri(15): Identificación de dispositivo adquiriente donde se origina la transacción. Ejemplo: PCDLandy (Puede ser el nombre del PC o IP de la máquina origen)

UsuOri(10): Nombre del usuario que origina la transacción.

Tarjeta(16): Número de la tarjeta débito/crédito.

ClaveSE(4) : Clave del servicio banca celular.

TerSupOri(10): Código del supervisor origen. Ejemplo: Puede ser el código un usuario supervisor de la intranet.

TerSupNom(40): Nombre del usuario supervisor.

Parámetro de salida: Función ReversaSuscripcionSE() as string.

Esta función devuelve la siguiente cadena de caracteres como resultado.
Retorno(3): Código de retorno
Si Retorno(3) <> “000”. A continuación viene descripción de error en un los siguientes (145) caracteres.
Si Retorno(3) = “000” Transacción OK.
Viene a continuación los siguientes campos:
Identificación(13): Número de Ced/Ruc/Pas del Cliente
Producto(2): Tipo del producto (Tarjeta de Suscripción) del cliente
TD, Tarjeta de Débito
VI, Tarjeta VISA
MC, Tarjeta MasterCard
TC, Tarjeta Club
Nombre(30) : Nombre del tarjetahabiente
TipTarHab(1): “N”=Normal, “E”=Empleado, “V”=VIP, “P”=VIP Plus
TerEsLocal(1): “0” = Cooperativa, “1” = Institución local

Ejemplo de una transacción OK:
”0000102777091 TDJUAN DIEGO LANDY E1”
Ejemplo de una transacción con ERROR:
“005NO SE PUEDE REVERSAR LA SUSCRIPCION BANCA CELULAR. FUERA DE FECHA”

4.4.3.2.9 Método: ReversaSusAutSE().

Descripción: Función que realiza el reverso de la suscripción autorizada o servicio adicional banca celular y su comisión. Esta transacción debe ser utilizada el mismo día de suscripción.

Visibilidad: Público.

Sintaxis:

Public Function ReversaSusAutSE(
ByVal TerNumCel, _
ByVal TerCodSer, _
ByVal TerCodRedAdq, _
ByVal TerIdeDisOri, _
ByVal TerUsuOri, _
ByVal TerNumTar, _
ByVal TerClaveSE, _
ByVal TerCodOpe, _
ByVal TerSupOri, _
ByVal TerSupNom) As String

Parámetros de entrada:

TerNumCel(9) : Número de celular
TerCodSer(3): Código de servicio = “SMS”
TerCodRedAdq(3): Código de red adquiriente. Si el front end es intranet colocar “ITR”.
TerIdeDisOri(15): Identificación de dispositivo adquiriente donde se origina la transacción. Ejemplo: PCDLandy (Puede ser el nombre del PC o IP de la máquina origen)
UsuOri(10): Nombre del usuario que origina la transacción.
Tarjeta(16): Número de la tarjeta Débito/Crédito
ClaveSE(4) : Clave del servicio banca celular.
TerCodOpe(5) : Código de operación autorizada.

Parámetro de salida: Función ReversaSusAutSE() as string

Esta función devuelve la siguiente cadena de caracteres como resultado.
Retorno(3): Código de retorno.
Si Retorno(3) <> “000” A continuación viene descripción de error en un los siguientes (145) caracteres.
Si Retorno(3) = “000” Transacción OK.
Viene a continuación los siguientes campos:
Identificación(13): Número de Ced/Ruc/Pas del Cliente
Producto(2): Tipo del producto (Tarjeta de Suscripción) del cliente

TD, Tarjeta de Débito

VI, Tarjeta VISA

MC, Tarjeta MasterCard

TC, Tarjeta Club

Nombre(30) : Nombre del tarjetahabiente

TipTarHab(1): “N”=Normal, “E”=Empleado, “V”=VIP, “P”=VIP Plus

TerEsLocal(1): “0” = Cooperativa, “1” = Institución local

Ejemplo de una transacción OK:

”0000102777091 TDJUAN DIEGO LANDY E1”

Ejemplo de una transacción con ERROR:

“005NO SE PUEDE REVERSAR LA SUSCRIPCION ADICIONAL BANCA CELULAR. FUERA DE FECHA”

4.4.3.2.10 Método: ConsultaSuscripcionesSE().

Descripción: Función que realiza la consulta de las suscripciones banca celular.

Visibilidad: Público.

Sintaxis:

Public Function ConsultaSuscripcionesSE(

ByVal TerNumCel, _

ByVal TerCodSer, _

ByVal TerCodRedAdq, _

ByVal TerNumTar, _

ByVal TerFecSusIni, _

ByVal TerFecSusFin, _

ByRef RespIsTranOk, _

ByRef RespData) As ADODB.Recordset

Parámetros de entrada:

TerNumCel(9): Número de celular suscrito

TerCodSer(3): Código de servicio “SMS”

TerCodRedAdq(3): Código de red adquiriente. Si el front end es intranet colocar “ITR”.

TerNumTar(16): Número de la tarjeta de débito/crédito

TerFecSusIni(10): Fecha de suscripción inicio (AAAA/MM/DD)

TerFecSusFin(10): Fecha de suscripción final (AAAA/MM/DD)

RespIsTranOk: boolean, indica si la consulta fue o no exitosa.

RespData(145): Si la variable RespIsTranOk = False, este parámetro devuelve la descripción del mensaje de error.

Parámetros de salida:

Función ConsultaSuscripcionesSE() as ADODB.Recordset

Campos del recordset:

TS-TipCli, puede tomar los siguientes valores:

"N" TipCli="Normal"

"E" TipCli="Empleado"

"V" TipCli="Vip"

"P" TipCli="Vip Plus"

Caso contrario,

TipCli="Desconoc"

"TS-EsLocal", puede tomar los siguientes valores:

"1" Institucion="BA"

"0" Institucion="COOP"

Caso contrario,

Institucion="DESC"

"TS-Act", puede tomar los siguientes valores:

"0" Status="INA"

"1" Status="ACT"

"2" Status="REV"

Caso contrario

Status="DESC"

"TS-Pag", puede tomar los siguientes valores:

"0" Batch="NO"

"1" Batch="OK"

Caso contrario,

Batch="DESC"

"TS-PagOnLin", puede tomar los siguientes valores:

"0" OnLine="NO"

"1" OnLine="OK"

Caso contrario,

OnLine="DESC"
"TS-FecSus", fecha de suscripción.
"TS-HorSus", hora de suscripción.
"TRA-CodRedAdq", Código de red adquiriente.
"TS-IdeDisOri", dispositivo que origina la transacción.
"TS-UsuOri", usuario que origina la transacción.
"TS-NumCel", número del celular.
"TS-NumTar", número de tarjeta
"TS-NomCli", nombre del cliente.
"TS-NomOpe", nombre de la operadora.
TipCli, tipo de cliente.
Institución, tipo de institución.
Status, estado de suscripción.
Batch, pago en batch.
OnLine, paga en línea.

4.4.3.2.11 Método: ConsultaSusAutSE().

Descripción: Función que realiza la consulta de las suscripciones autorizadas que tiene una tarjeta.

Visibilidad: Público.

Sintaxis:

```
Public Function ConsultaSusAutSE(
ByVal TerCodSus,
ByVal TerNumCel, _
ByVal TerCodSer, _
ByVal TerCodRedAdq, _
ByVal TerNumTar, _
ByRef RespIsTranOk, _
ByRef RespData) As ADODB.Recordset
```

Parámetros de entrada:

TerCodSus: Código de suscripción. Este valor fue asignado por la base de datos al momento de suscribir al servicio banca celular.

TerNumCel(9): Número de celular suscrito.

TerCodSer(3): Código de servicio “SMS”.

TerCodRedAdq(3): Código de red adquiriente. Si el front end es intranet colocar “ITR”.

TerNumTar(16): Número de la tarjeta de débito/crédito.

RespIsTranOk: boolean, indica si la consulta fue o no exitosa.

RespData(145): Si la variable RespIsTranOk = False, este parámetro devuelve la descripción del mensaje de error:

Parámetros de salida:

Función ConsultaSusAutSE() as ADODB.Recordset

Campos del recordset:

"TSSE-Act" puede tomar los siguientes valores:

"0" significa que status="INA"

"1" significa que status="ACT"

"2" significa que status="REV"

Caso contrario

Status="DESC"

"TSSE-Pag", puede tomar los siguientes valores:

"0" Batch="NO"

"1" Batch="OK"

Caso contrario

Batch="DESC"

"TSSE-PagOnLin", puede tomar los siguientes valores:

"0", OnLine="NO"

"1", OnLine="OK"

Caso contrario

OnLine="DESC"

"TSSE-FecSus", fecha de suscripción.

"TSSE-HorSus", hora de suscripción.

"TRA-CodRedAdq", código de red adquiriente.

"TSSE-IdeDisOri", identificación del dispositivo origen.

"TSSE-UsuOri", nombre del usuario origen.
"TSSE-NumCel", número celular.
"TSSE-NumTar", número de tarjeta.
"TO-CodOpe", código de operación.
"DesOpe", nombre de operación.
Status, estatus de suscripción adicional.
"TMO-Motivo", código de motivo de inactivación de suscripción.
Batch, indica si el cobro es en proceso batch.
OnLine, indica si el cobro es en línea.

4.4.3.2.12 Método: ConsultaTransaccionesSE().

Descripción: Función que realiza la consulta de las suscripciones autorizadas que tiene una tarjeta.

Visibilidad: Público.

Sintaxis:

```
Public Function ConsultaTransaccionesSE(
ByVal TerNumCel, _
ByVal TerCodSer, _
ByVal TerCodRedAdq, _
ByVal TerNumTar, _
ByVal TerFecTraIni, _
ByVal TerFecTraFin, _
ByVal TerTipCue, _
ByVal TerNumCue, _
ByVal TerTipTra, _
ByRef RespIsTranOk, _
ByRef RespData) As ADODB.Recordset
```

Parámetros de entrada:
TerNumCel(9): Número de celular suscrito.
TerCodSer(3): Código de servicio “SMS”.

TerCodRedAdq(3): Código de red adquiriente. Si el front end es intranet colocar "ITR".

TerNumTar(16): Número de la tarjeta de débito/crédito.

TerFecTraIni(10):Fecha inicial (aaaa/mm/dd).

TerFecTraFin(10): Fecha final (aaaa/mm/dd).

TerTipCue(2): Tipo de cuenta del banco

"00"=TODAS

"10"=AH CUENTA AHORROS

"20"=CC CUENTA CORRIENTE

TerNumCue(10): Número de cuenta del banco

TerTipTra(2)

"10"=TODAS LAS TRANSACCIONES

"0"=OK TRANSACCIONES COMPLETAS

"1"=OK ESPERANDO RESPUESTA OPERADORA

"2"=TRANSACCIONES ERRONEAS

"3"=POR REVERSAR

"4"=REVERSADAS OK

"5"=SIN RESPUESTA DEL AUTORIZADOR

RespIsTranOk: boolean, indica si la consulta fue o no exitosa.

RespData(145): Si la variable RespIsTranOk = False, este parámetro devuelve la descripción del mensaje de error:

Parámetros de salida:

Function ConsultaTransaccionesSE As ADODB.Recordset

Campos del recordset

"TSET-TipTarHab", puede tomar los siguientes valores:

"N", entonces TipCli="Normal"

"E", entonces TipCli="Empleado"

"V", entonces TipCli="Vip"

"P", entonces TipCli="Vip Plus"

Caso contrario

TipCli="Desconoc"

"TSET-Status",estatus de la transacción.

"TSET-FecTra", fecha de transacción.

"TSET-HorTra", hora de transacción.

"TRA-CodRedAdq", código de red adquiriente.

"TO-CodOpe", código de operación.
"TraDes", descripción de transacción.
"TSET-IdeDisOri", identificación de dispositivo que origino la transacción.
"TSET-UsuOri", nombre de usuario que origino la transacción.
"TSET-SupOri", código del supervisor que autorizo la transacción.
"TSET-SupNom" nombre del supervisor que autorizo la transacción.
"TSET-NumTar", número de tarjeta.
"TSET-NumCel", número de celular.
"TO-CodOpeSA", código de operación del servicios adicional.
TipCli, tipo de cliente.
TipCue, tipo de cuenta.
"TSET-CueDes", número de cuenta destino.
"TSET-NumAut", número de autorizacion de VISA, MASTERCARD.
"TSET-ComAdq", valor de comisión cobrada para red adquiriente.
"TSET-ComAut", valor de comisión cobrada para red autorizadora.
"TSET-ComTot", valor de comisión total cobrada.
"TSET-CodRetBan", código de retorno de la operación bancaria.
"TSET-CodRetSMS", código de retorno entregado al SMS
"TSET-CodRetOpe", código del retorno de la operadora celular.
"TSET-MsgErr", mensaje de error.
"TST-StaDes", descripción del estado.

4.4.3.3 Método Proctrn de operación PULL – PUSH.

Para cualquier operación PULL o PUSH el método de acceso a la operación es el Proctrn de la clase ServiciosElectronicos.Interfaz.

Método: Proctrn()

Descripción: Método principal cuya función es el de actuar de interfaz entre los sistemas de la institución financiera y la pasarela SMS. Captura el mensaje de entrada enviado por la pasarela SMS en la variable DataInput y formatea el mensaje de salida en el parámetro DataOutput, que es entregado a la pasarela para que de formato al mensaje que se va a presentar, en el celular del cliente.

Visibilidad: Público.

Sintáxis:

Public Sub ProcTrn(
ByVal DataInput As String, _
ByRef DataOutput As String)

Parámetro de entrada:
BInput, variable en donde se espera el mensaje de entrada.

Parámetro de salida:
BOutput, variable en donde se formatea el mensaje de salida.

Operaciones PULL. Diseño de mensajería DataInput y DataOutput.

4.4.3.3.1 OPERACION "SUSCR".

Descripción: Operación de suscripción al servicio banca celular.
Parámetro de entrada: BInput
Nemónicos asociados a la pasarela:
Corto: sol
Largo: solicito
Formato:
Operación(5): "SUSCR"
Celular(9): Número del Celular de suscripción
CodSer(3):Código del Servicio (SMS=Celular)
RedOri(3): Código de Red de Adquiriencia (SMS= CELULAR, CMS=CARD MANAGEMENT SYSTEM, INT=PORTAL BANCO INTERNET, ITR= INTRANET DEL BANCO)
IdeDisOri(15) : Identificador del dispositivo desde donde se origina la transacción. En el caso del SMS el número de celular
UsuOri(10): Nombre del usuario que origina la transacción. En el caso de la adquiriencia de SMS el dueño del celular
Tarjeta(19): Número de tarjeta débito/crédito.

Pin (4): Clave de tarjeta débito/crédito.

Ejemplo del mensaje SMS ingresado por el usuario:

“sol 4931760012341324 2453”

O

“solicito 4931760012341324 2453”

Ejemplo trama de entrada:
Formato de trama de ejemplo desde la adquiriencia del SMS:
“SUSCR098171382SMSSMS098171382 098171382 4931760012341324 2453“

Parámetro de salida: BOutput
Formato:
Retorno(3):
Si es “000”, Transacción OK, a continuación vienen los siguientes campos.
Identificación(13): Número de ced/ruc/pas del cliente.
Producto(2): Tipo del producto (Tipo de tarjeta de suscripción) del cliente
TD, Tarjeta de Débito
VI, Tarjeta VISA
MC, Tarjeta MasterCard
TC, Tarjeta Club

Si es <> “000”, Transacción ERROR, a continuación vienen los siguientes campos.
MensajeError(145) en caso de que este campo se devuelva en blanco, entonces el mensaje de error se toma de plantilla de la pasarela.

Ejemplo de una trama de salida - transacción OK:
”0000102777091 TD”

Ejemplo de una trama de salida - transacción de ERROR:
“020TARJETA CANCELADA”

Plantilla ingresada en la pasarela:

“BIENVENIDO A BANCA CELULAR. SU IDENTIFICACION ES: <#CEDULA#> . RECUERDE QUE TIENE OPCION DE CAMBIAR SU CLAVE PERSONAL.”

Ejemplo del mensaje que se mostrara al usuario:

“BIENVENIDO A BANCA CELULAR. SU IDENTIFICACION ES: 0102777091. RECUERDE QUE TIENE OPCION DE CAMBIAR SU CLAVE PERSONAL.”

4.4.3.3.2 OPERACION “PASSW”.

Descripción: Operación de cambio de password al servicio SMS banca celular.
Parámetro de entrada: BInput
Nemónicos asociados a la pasarela:
Corto: cla
Largo: clave
Formato:
Operación(5): “PASSW”
Celular(9): Número del Celular de suscripción
CodSer(3):Código del Servicio (SMS=Celular)
RedOri(3): Código de Red de Adquiriencia (SMS= CELULAR, CMS=CARD MANAGEMENT SYSTEM, INT=PORTAL BANCO INTERNET, ITR= INTRANET DEL BANCO)
IdeDisOri(15) : Identificador del dispositivo desde donde se origina la transacción. En el caso del SMS el número de celular
UsuOri(10): Nombre del usuario que origina la transacción. En el caso de la adquiriencia de SMS el dueñ0 de celular
Tarjeta(19): Número de Tarjeta Débito/Crédito
ClaveAnt(4):Clave Anterior de suscripción del servicio SMS
ClaveNue(4):Clave Nueva de suscripción del servicio SMS

Ejemplo de mensaje SMS ingresado por el usuario:

“cla 341324 2453 1747”

O

“clave 341324 2453 1747”

Ejemplo de trama de entrada:

“PASSW098171382SMSSMS098171382 098171382 4931760012341324 24531747“

Parámetro de salida: BOutput
Formato:
Retorno(3):
Si es “000”, Transacción OK.
Si es <> “000”, Transacción ERROR, a continuación vienen los siguientes campos.
MensajeError(145) en caso de que este campo se devuelva en blanco, entonces el mensaje de error se toma de plantilla de la pasarela.

Ejemplo de una trama de salida - transacción OK:

”000”

Ejemplo de una trama de salida – transacción ERROR:

”030CLAVE ANTERIOR ES INCORRECTA”

Plantilla ingresada en la pasarela:
“TARJETA <#TARJETA#>: CAMBIO DE CLAVE SE SERVICIOS BANCA CELULAR OK”

Ejemplo de mensaje que se mostrara al usuario:

“TARJETA 4931760012341324: CAMBIO DE CLAVE SE SERVICIOS BANCA CELULAR OK”

4.4.3.3.3 OPERACION “SUSSA”.

Descripción: Operación de suscripción de servicio adicional.
Parámetro de entrada: BInput
Nemónicos asociados a la pasarela:
Corto: adi
Largo: adiciona
Formato:
Operación(5): “SUSSA”
Celular(9): Número del celular de suscripción.
CodSer(3): Código del Servicio (SMS=Celular).
RedOri(3): Código de Red de Adquiriencia (SMS= CELULAR, CMS=CARD MANAGEMENT SYSTEM, INT=PORTAL BANCO INTERNET, ITR= INTRANET DEL BANCO)
IdeDisOri(15): Identificador del dispositivo desde donde se origina la transacción. En el caso del SMS el número de celular
UsuOri(10): Nombre del usuario que origina la transacción. En el caso de la adquiriencia de SMS el dueño de celular
Tarjeta(19): Número de tarjeta débito/crédito.
Clave(4):Clave de suscripción del servicio banca celular.
CodOpe(5): Código de Operación del servicio adicional SMS.

AVCHC = Aviso de cheque cobrado >= 2000
NOATM = Notificación de retiros o avances en efectivo.
NOCTC = Notificación de compras vía POS.

Ejemplo de mensaje SMS ingresado por el usuario:
“ADI 341324 2453 NOATM”

Ejemplo de la trama de entrada:
“SUSSA098171382SMSSMS098171382 098171382 4931760012341324 2453NOATM “

Parámetro de salida: BOutput

Formato:

Retorno(3):

Si es “000”, Transacción OK, a continuación vienen los siguientes campos.

Identificación(13): Número de ced/ruc/pas del cliente.

Producto(2): Tipo del producto (Tipo de tarjeta de suscripción) del cliente

TD, Tarjeta de Débito

VI, Tarjeta VISA

MC, Tarjeta MasterCard

TC, Tarjeta Club

Si es <> “000”, Transacción ERROR, a continuación vienen los siguientes campos.

MensajeError(145) en caso de que este campo se devuelva en blanco, entonces el mensaje de error se toma de plantilla de la pasarela.

Ejemplo de una trama de salida - transacción OK:

”0000102777091 TD”

Ejemplo de una trama de salida - transacción de ERROR:

“020TARJETA CANCELADA”

Plantilla ingresada en la pasarela:

“SUSCRIPCION DEL SERVICIO ADICIONAL <#SERVICIO#> OK!”

Ejemplo del mensaje que se mostrara al usuario:

“SUSCRIPCION DEL SERVICIO ADICIONAL NOATM OK!”

4.4.3.3.4 OPERACION “BLTAR”.

Descripción: Operación de bloqueo de tarjeta de débito/crédito.

Parámetro de entrada: BInput

Nemónicos asociados a la pasarela:

Corto: BLQ

Largo: BLOQUEO

Formato:

Operación(5): “BLTAR”

Celular(9): Número del Celular de suscripción

CodSer(3):Código del Servicio (SMS=Celular)

RedOri(3): Código de Red de Adquiriencia (SMS= CELULAR, CMS=CARD MANAGEMENT SYSTEM, INT=PORTAL BANCO INTERNET, ITR= INTRANET DEL BANCO)

IdeDisOri(15) : Identificador del dispositivo desde donde se origina la transacción. En el caso del SMS el número de celular

UsuOri(10): Nombre del usuario que origina la transacción. En el caso de la adquiriencia de SMS el dueño del celular.

Tarjeta(19): Número de tarjeta débito/crédito.

Pin Tarjeta (4): Clave tarjeta débito/crédito.

Clave (4): Clave de banca celular.

Ejemplo del mensaje SMS ingresado por el usuario:

“BLQ 341324 1542 2453”

Ejemplo de trama de entrada:

“BLTAR098171382SMSSMS098171382 098171382 4931760012341324 15422453“

Parámetro de salida: BOutput

Formato:

Retorno(3):

Si es “000”, Transacción OK.

Si es <> “000”, Transacción ERROR, a continuación vienen los siguientes campos.

MensajeError(145) en caso de que este campo se devuelva en blanco, entonces el mensaje de error se toma de plantilla de la pasarela.

Ejemplo de una trama de salida - transacción OK:

”000”

Ejemplo de una trama de salida – transacción ERROR:

”030CLAVE ANTERIOR ES INCORRECTA”

Plantilla ingresada en la pasarela:

“BLOQUEO DE TARJETA <#TARJETA#> OK!”

Ejemplo de mensaje que se mostrara al usuario:

“BLOQUEO DE TARJETA 324 OK!”

4.4.3.3.5 OPERACION “BLTAR”.

Descripción: Operación que realiza la consulta de saldos de una tarjeta de débito. Dentro de los saldos que se presenta al cliente esta el saldo por confirmar, el saldo disponible y el saldo total.

Parámetro de entrada: BInput

Nemónicos asociados a la pasarela:

Corto: s, sca, scc

Largo: saldos saldoah, saldocc

Formato:

Operación(5): “SALDO”

Celular(9): Número del celular que origina la consulta de saldos.

CodSer(3):Código del Servicio (SMS=Celular)

RedOri(3): Código de Red de Adquiriencia (SMS= CELULAR, CMS=CARD MANAGEMENT SYSTEM, INT=PORTAL BANCO INTERNET, ITR= INTRANET DEL BANCO)

IdeDisOri(15) : Identificador del dispositivo desde donde se origina la transacción. En el caso del SMS el número de celular

UsuOri(10): Nombre del usuario que origina la transacción. En el caso de la adquiriencia de SMS el dueño del celular

Tarjeta(19): Número de Tarjeta Débito/Crédito

Clave(4): Clave de servicios electrónicos. Esta operación no tiene clave. Rellenamos con ceros.

Tipo(2): Tipo de Cuenta del Cliente a consultar CC, CA y si viene en blanco son todas las cuentas.

Cuenta (10): Cuenta del Cliente. Si tipo viene con CC o CA y si el campo cuenta viene en blanco el sistema asume por omisión la cuenta CC o CA. Si en este

campo viene el número de la cuenta entonces el sistema va a calcular el saldo respectivo, en caso de que tipo viene en blanco, este campo debe llegar en blanco.

Ejemplo a) del mensaje SMS ingresado por el usuario: El usuario desea consultar los saldos de la cuenta de ahorros asociado a la tarjeta de débito de la suscripción.

"sca"

Ejemplo a) del mensaje SMS ingresado por el usuario: El usuario desea consultar los saldos de la cuenta de ahorros asociado a la tarjeta de débito de la suscripción

"scc"

Ejemplo c) del mensaje SMS ingresado por el usuario: El usuario desea consultar los saldos de la cuenta de ahorros y cuenta corriente asociado a la tarjeta de débito de la suscripción

"s"

Ejemplo de trama de entrada a):

Solicita saldo de la cuenta de ahorros atachada a la tarjeta de débito.

"SALDO098171382SMSSMS098171382 098171382 4931760011233445 0000CA "

Ejemplo de trama de entrada b):

Solicita saldo de la cuenta corriente atachada a la tarjeta de débito.

"SALDO098171382SMSSMS098171382 098171382 4931760011233445 0000CC "

Ejemplo de trama de entrada c):

Solicita saldos de la cuenta de ahorros y cuenta corriente.

"SALDO098171382SMSSMS098171382 098171382 4931760011233445 0000 "

Parámetro de salida: BOutput

Formato:

Retorno(3):

Si es "000", Transacción OK. A continuación vienen los siguientes campos:

DescTipo1(32): Descripción de cuenta (Definir por Banco CUENTA CA)

Cuenta1 (10): Número de Cuenta 1
Disponible1 (13): Saldo disponible de la cuenta 1. Sin formato 12 Enteros 2 decimales con signo + o – carácter inicial
Confirmar1 (13): Saldo por confirmar de la cuenta 1. Sin formato 12 Enteros 2 decimales con signo + o – carácter inicial
Total1 (13): Saldo por confirmar de la cuenta 1. Sin formato 12 Enteros 2 decimales con signo + o – carácter inicial
DescTipo2(32): Descripción de cuenta (Definir por Banco CUENTA CC)
Cuenta2 (10): Número de Cuenta 2
Disponible2 (13): Saldo disponible de la cuenta 2. Sin formato 12 Enteros 2 decimales con signo + o – carácter inicial
Confirmar2 (13): Saldo por confirmar de la cuenta 2. Sin formato 12 Enteros 2 decimales con signo + o – carácter inicial
Total2 (13): Saldo por confirmar de la cuenta 2. Sin formato 12 Enteros 2 decimales con signo + o – carácter inicial

Si es <> “000”, Transacción ERROR, a continuación vienen los siguientes campos.
MensajeError(145) en caso de que este campo se devuelva en blanco, entonces el mensaje de error se toma de plantilla de la pasarela.

Ejemplo de una trama de salida - transacción OK:

Ejemplo:
”000”
“CUENTA AHORROS 702285 “
“+000000006156+000000000000+000000006156“
“CUENTACORRIENTE “
“+000000000000+000000000000+000000000000”

Ejemplo de una trama de salida – transacción ERROR:

”030TARJETA SUSPENDIDA TEMPORALMENTE”

Plantilla ingresada en la pasarela:

“<#DESCTIPO1#> <#NUMCTA1#>: DISP <#DISP1#>, POR CONF <#CONF1#>, TOTAL <#TOTAL1#>; <#DESCTIPO2#> <#NUMCTA2#>: DISP <#DISP2#>, POR CONF <#CONF2#>, TOTAL <#TOTAL2#>”

Ejemplo de mensaje que se mostrara al usuario:

“CUENTA AHORROS 285: DISP $61.56, POR CONF $0.00, TOTAL $61.56; CUENTA CORRIENTE: DISP $0.00, POR CONF $0.00, TOTAL $0.00”

4.4.3.3.6 OPERACION “SALTC”.

Descripción: Consulta de saldos de tarjeta de crédito.
Nemónicos asociados a la pasarela:
Corto: stc
Largo: saldotc
Parámetro de entrada: BInput
Formato:
Operación(5): “SALTC
Celular(9): Número del Celular que origina la consulta de saldos.
CodSer(3):Código del Servicio (SMS=Celular).
RedOri(3): Código de Red de Adquiriencia (SMS= CELULAR, CMS=CARD MANAGEMENT SYSTEM, INT=PORTAL BANCO INTERNET, ITR= INTRANET DEL BANCO).
IdeDisOri(15): Identificador del dispositivo desde donde se origina la transacción. En el caso del SMS el número de celular.
UsuOri(10): Nombre del usuario que origina la transacción. En el caso de la adquiriencia de SMS el dueño del celular.
Tarjeta(19): Número de tarjeta crédito.
Clave(4): Para esta operación el usuario no tiene que digitar la clave. Rellenamos el campo con ceros.

Ejemplo del mensaje SMS ingresado por el usuario:

“stc 890987 1524”

Ejemplo de trama de entrada:

"SALTC098171382SMSSMS098171382 098171382 4563580987890987 1524"

Parámetro de Salida: BOutput

Formato:

Retorno(3):

Si es "000", Transacción OK. A continuación vienen los siguientes campos:

DescTipo(32): Descripción de tarjeta de crédito.

Tarjeta (19): Número de Tarjeta de Crédito de la consulta

SalDisRot (13): Saldo disponible rotativo. Sin formato 10 Enteros 2 decimales con signo + o – carácter inicial

SalDisDif (13): Saldo disponible diferido. Sin formato 10 Enteros 2 decimales con signo + o – carácter inicial

SalDisTot (13): Saldo disponible total. Sin formato 10 Enteros 2 decimales con signo + o – carácter inicial

SalUtiRot (13): Saldo utilizado rotativo. Sin formato 10 Enteros 2 decimales con signo + o – carácter inicial

SalUtiDif (13): Saldo utilizado diferido. Sin formato 10 Enteros 2 decimales con signo + o – carácter inicial

SalUtiTot (13): Saldo utilizado total. Sin formato 10 Enteros 2 decimales con signo + o – carácter inicial

ValMinPag (13): Valor mínimo a pagar. Sin formato 10 Enteros 2 decimales con signo + o – carácter inicial

ValTotPag (13): Valor total a pagar. Sin formato 10 Enteros 2 decimales con signo + o – carácter inicial

FechaMaxPag (10): Fecha máxima de pago a para cancelar el valor de tarjeta (AAAA/MM/DD).

Si es <> "000", Transacción ERROR, a continuación vienen los siguientes campos.

MensajeError(145) en caso de que este campo se devuelva en blanco, entonces el mensaje de error se toma de plantilla de la pasarela.

Ejemplo de una trama de entrada – transacción OK:

”000
+000000015000+000000035000+000000050000+000000020000+000000011000+
000000031000+000000005000+0000000310002004/04/31”

Ejemplo de una trama de salida – transacción ERROR:

”030TARJETA DE CREDITO CANCELADA”

Plantilla ingresada en la pasarela:

<#DESCTIPO#> <#NUMTRJ#>: DISP.TOTAL <#DISPTOTAL#> UTIL.TOTAL <#UTILTOTAL#> MIN. PAGAR <#MINPAGAR#> TOT. PAGAR <#TOTPAGAR#> FECHA MAX. PAGO <#FECHAMAXPAGO#>

Ejemplo de mensaje que se mostrara al usuario:

“TARJETA VISA 890987: DISP TOTAL $50.00 UTIL. TOTAL $310.00, MIN. PAGAR $50.00 TOT. PAGAR $310.00 FECHA MAX. PAGO 2007/04/30”

4.4.3.3.7 OPERACION “CUPTC”.

Descripción: Consulta de cupos de tarjeta de crédito.
Nemónicos asociados a la pasarela:
Corto: ctc
Largo: cupotc
Parámetro de entrada: BInput
Formato:
Operación(5): “TARCU”
Celular(9): Número del Celular que origina la consulta de saldos
RedOri(3): Código de Red de Adquiriencia (SMS= CELULAR, CMS=CARD MANAGEMENT SYSTEM, INT=PORTAL BANCO INTERNET, ITR= INTRANET

DEL BANCO)

IdeDisOri(15) : Identificador del dispositivo desde donde se origina la transacción. En el caso del SMS el número de celular

UsuOri(10): Nombre del usuario que origina la transacción. En el caso de la adquiriencia de SMS el dueño del celular

Tarjeta(19): Número de Tarjeta Crédito

Clave(4): Clave de servicios electrónicos

Tipo(2): TO= Todos los cupos, RO=Rotativo, DI=Diferido.

Ejemplo del mensaje SMS ingresado por el usuario:

"stc 890987 1524"

Ejemplo de trama de entrada:

"TARCU098171382SMSSMS098171382 098171382 4563580987890987 1245TO"

Parámetro de Salida: BOutput

Formato:

Retorno(3): "000" Transacción OK

DescTipo(32): Descripción de Tarjeta de Crédito (Definir por Banco VISA)

Tarjeta (19): Número de Tarjeta de Crédito de la consulta

CupTotRot (13): Cupo total rotativo. Sin formato 12 Enteros 2 decimales con signo + o – carácter inicial

CupUtiRot (13): Cupo utilizado rotativo. Sin formato 12 Enteros 2 decimales con signo + o – carácter inicial

CupDisRot (13): Cupo disponible rotativo. Sin formato 12 Enteros 2 decimales con signo + o – carácter inicial

FechaMaxPagRot (10): Fecha máxima de pago rotativo para cancelar el valor de tarjeta (AAAA/MM/DD).

CupTotDif (13): Cupo total diferido. Sin formato 12 Enteros 2 decimales con signo + o – carácter inicial

CupUtiDif (13): Cupo utilizado diferido. Sin formato 12 Enteros 2 decimales con signo + o – carácter inicial

CupDisDif (13): Cupo disponible diferido. Sin formato 12 Enteros 2 decimales con signo + o – carácter inicial

FechaMaxPagDif (10): Fecha máxima de pago diferido para cancelar el valor de

tarjeta (AAAA/MM/DD).

Si es <> “000”, Transacción ERROR, a continuación vienen los siguientes campos. MensajeError(145) en caso de que este campo se devuelva en blanco, entonces el mensaje de error se toma de plantilla de la pasarela.

Ejemplo del mensaje SMS ingresado por el usuario:
“ctc 890987 1245”

Ejemplo de una trama de entrada – transacción OK:
”000”
“TARJETA VISA ”
+000000015000+000000085000+0000001000002007/04/30+000000000000+0000
00000000+0000000000000000/00/00”

Ejemplo de una trama de salida – transacción ERROR:
”030TARJETA DE CREDITO CANCELADA”

Plantilla ingresada en la pasarela:
“<#DESCTIPO#> <#NUMTRJ#> CUPO TOTAL. <#CUPTOTALROT#> CUPO DISP. <#CUPDISPROT#> FECHA MAX. PAGO <#FECHAMAXPAGOROT#>”

Ejemplo de mensaje que se mostrara al usuario:
“TARJETA VISA 890987: CUPO TOTAL $50.00 CUPO DISP. $310.00 FECHA MAX. PAGO 2007/04/30”

4.4.3.3.8 OPERACION “CONCH”.

Descripción: Consulta de cheque cobrado.
Nemónicos asociados a la pasarela:
Corto: chq
Largo: cheque
Parámetro de entrada: BInput
Formato:
Operación(5): “CONCH”

Celular(9): Número del Celular que origina la consulta de saldos

RedOri(3): Código de Red de Adquiriencia (SMS= CELULAR, CMS=CARD MANAGEMENT SYSTEM, INT=PORTAL BANCO INTERNET, ITR= INTRANET DEL BANCO)

IdeDisOri(15) : Identificador del dispositivo desde donde se origina la transacción. En el caso del SMS el número de celular

UsuOri(10): Nombre del usuario que origina la transacción. En el caso de la adquiriencia de SMS el dueño del celular

Tarjeta(19): Número de Tarjeta Débito

Clave(4): Clave de servicios electrónicos

NumChe(8):Número de cheque. Formato numérico. Justificado a la derecha y rellenado con espacios en blanco.

Ejemplo del mensaje SMS ingresado por el usuario:

"chq 80246 1252 100421"

Ejemplo de trama de entrada:

"CONCH098171382SMSSMS098171382 098171382 4931760010080246 1252100421 "

Parámetro de Salida: BOutput

Formato:

Retorno(3):

"000" Transacción OK

<> "000" Viene descripción de error en campo MensajeError(145) en caso de que este campo se devuelva en blanco, entonces el mensaje de error se toma de plantilla de la pasarela.

Ejemplo de una trama de salida – transacción OK:

"000"

Ejemplo de una trama de salida – transacción ERROR:

"040CHEQUE NO EXISTE"

Plantilla ingresada en la pasarela:

“EL CHEQUE NO. <#CHEQUE#> <#ESTADO#>”

Ejemplo de mensaje que se mostrara al usuario:
“EL CHEQUE NO. 100421 PAGADO”

4.4.3.4 Método Proctrn de operaciones PUSH.

4.4.3.5 OPERACION “NOTON”

Descripción: Notificación de avisos Push-Online sin suscripción al servicio banca celular.

PARAMETRO DE ENTRADA: BInput

Formato:

Operación(5): “NOTON”

Celular(9): Número del Celular a donde se desea enviar el mensaje sms.

CodSer(3):Código del Servicio (SMS=Celular)

RedOri(3): Código de Red de Adquiriencia (SMS= CELULAR, CMS=CARD MANAGEMENT SYSTEM, INT=PORTAL BANCO INTERNET, ITR= INTRANET DEL BANCO)

IdeDisOri(15) : Identificador del dispositivo desde donde se origina la transacción. En el caso del SMS el número de celular

UsuOri(10): Nombre del usuario que origina la transacción. En el caso de la adquiriencia de SMS el dueño del celular

Tarjeta(19): Número de tarjeta especial. Este número debe solicitar al administrador del sistema banca celular. Normalmente el BIN de la tarjeta especial es “911111”.

Clave(4): Clave de servicio banca celular. Rellenar con ceros.

CodOpePull(5): Código de operación Pull sin suscripción. Dependiendo del tipo de alerta que se desea enviar al cliente, este código es asignado por el administrador del sistema banca celular.

Mensaje(145): Mensaje que se mostrara al cliente en el celular.

TerUsuDes(150): Nombre del dueño del celular a donde se notifica.

TerTipPas(1): Tipo de pasarela utilizar. El dominio del campo es “L” – Pasarela SMS; “E”-Email; “S” – Sistema selecciona mejor alternativa.

Inf1(255): Campo informativo. Se puede enviar información que identifique el evento que origina la transacción. En el caso de notificación de remesas va el código de la empresa remesadora.

Inf2(255): Campo informativo. Se puede enviar información que identifique el evento que origina la transacción. En el caso de notificación de remesas va el código de la remesa del exterior.

Ejemplo del envío de un mensaje de notificación de la llegada de una remesas del exterior.

“NOTON098171382SMSREM098171382 098171382 91111100100802460000
NOTREESTIMADO CLIENTE DE AUSTROGIROS ACABA DE RECIBIR UNA
REMESA DEL EXTERIOR POR 150 USD
LANDY ANDRADE JUAN DIEGO

L001

1233

“

Parámetro de Salida: BOutput

Formato:

Retorno(3): “000” Transacción OK

<> “000” Viene descripción de error en campo MensajeError(145) en caso de que este campo se devuelva en blanco, entonces el mensaje de error se toma de plantilla de Pasarela.

4.4.1.4.2 OPERACION “NOTBA”

Descripción: Notificación de avisos Push-Batch sin suscripción al servicio banca celular.

PARAMETRO DE ENTRADA: BInput

Formato:

Operación(5): “NOTBA”

Celular(9): Número del Celular a donde se desea enviar el mensaje sms.

CodSer(3):Código del Servicio (SMS=Celular)

RedOri(3): Código de red de adquiriencia (SMS= CELULAR, CMS=CARD MANAGEMENT SYSTEM, INT=PORTAL BANCO INTERNET, ITR= INTRANET DEL BANCO)

IdeDisOri(15) : Identificador del dispositivo desde donde se origina la transacción. En el caso del SMS el número de celular

UsuOri(10): Nombre del usuario que origina la transacción. En el caso de la adquiriencia de SMS el dueño del celular

BinEspecial(19): Número de tarjeta especial. Este número debe solicitar al administrador del sistema banca celular. Se asigna el número de tarjeta especial a la operación push batch sin suscripción cuando se crea la transacción en el administrador del sistema. Normalmente el BIN de la tarjeta especial es “911111”.

Clave(4): Clave de servicio banca celular. Rellenar con ceros.

CodOpePull(5): Código de operación Pull sin suscripción. Dependiendo del tipo de alerta que se desea enviar al cliente, este código es asignado por el Administrador del sistema banca celular.

Mensaje(145): Mensaje que se mostrara al cliente en el celular.

TerUsuDes(150): Nombre del dueño del celular a donde se notifica.

TerTipPas(1): Tipo de pasarela utilizar. El dominio del campo es “L” – Pasarela SMS; “E”-Email; “S” – Sistema selecciona mejor alternativa.

Inf1(255): Campo informativo. Se puede enviar información que identifique el evento que origina la transacción. En el caso de notificación de remesas va el código de la empresa remesadora.

Inf2(255): Campo informativo. Se puede enviar información que identifique el evento que origina la transacción. En el caso de notificación de remesas va el código de la remesa del exterior.

Ejemplo del envío de un mensaje de promoción del servicio banca celular.

“NOTBA098171382SMSPPB098171382 098171382 911111
0000BANCE ESTIMADO CLIENTE. LA INSTITUCION FINANCIERA BRINDA EL NUEVO SERVICIO DE BANCA CELULAR. POR FAVOR SUSCRIBASE EN

CUALQUIER AGENCIA LANDY ANDRADE JUAN DIEGO

L

“

Parámetro de Salida: BOutput

Formato:

Retorno(3): “000” Transacción OK

<> “000” Viene descripción de error en campo MensajeError(145) en caso de que este campo se devuelva en blanco, entonces el mensaje de error se toma de plantilla de Pasarela.

4.4.3.5.1 OPERACION “NOLSU”

Descripción: Notificación de avisos push con suscripción al servicio banca celular.

PARAMETRO DE ENTRADA: BInput

Formato:

Operación(5): “NOLSU”

Celular(9): Número del Celular a donde se desea enviar el mensaje sms.

CodSer(3):Código del Servicio (SMS=Celular)

RedOri(3): Código de Red de Adquiriencia (SMS= CELULAR, CMS=CARD MANAGEMENT SYSTEM, INT=PORTAL BANCO INTERNET, ITR= INTRANET DEL BANCO)

IdeDisOri(15) : Identificador del dispositivo desde donde se origina la transacción. En el caso del SMS el número de celular

UsuOri(10): Nombre del usuario que origina la transacción.

En el caso de la adquiriencia de SMS el dueño del celular

BinEspecial(19): Número de tarjeta especial. Este número debe solicitar al administrador del sistema banca celular. Se asigna el número de tarjeta especial a la operación push batch sin suscripción cuando se crea la transacción en el administrador del sistema. Normalmente el BIN de la tarjeta especial es “911111”.

Clave(4): Clave de servicio banca celular. Rellenar con ceros.

CodOpePull(5): Código de operación Pull sin suscripción. Dependiendo del tipo de alerta que se desea enviar al cliente, este código es asignado por el Administrador del sistema banca celular.

Mensaje(145): Mensaje que se mostrara al cliente en el celular.

TerUsuDes(150): Nombre del dueño del celular a donde se notifica.

TerTipPas(1): Tipo de pasarela utilizar. El dominio del campo es “L” – Pasarela SMS; “E”-Email; “S” – Sistema selecciona mejor alternativa.

Inf1(255): Campo informativo. Se puede enviar información que identifique el evento que origina la transacción. En el caso de notificación de remesas va el código de la empresa remesadora.

Inf2(255): Campo informativo. Se puede enviar información que identifique el evento que origina la transacción. En el caso de notificación de remesas va el código de la remesa del exterior.

Ejemplo del envío de un mensaje de promoción del servicio banca celular.

“NOLSU098171382SMSCHA098171382 098171382 911111
0000NOATM ESTIMADO CLIENTE. POR SU SEGURIDAD LE INFORMAMOS QUE SE HA RETIRADO 50 USD DE SU CUENTA DEL CAJERO EL AEROPUERTO
LANDY ANDRADE JUAN DIEGO

L

“

Parámetro de Salida: BOutput

Formato:

Retorno(3): “000” Transacción OK

<> “000” Viene descripción de error en campo MensajeError(145) en caso de que este campo se devuelva en blanco, entonces el mensaje de error se toma de plantilla de Pasarela.

4.4.4 *Clase ServiciosOperacion.*

La función principal de esta clase es implementar la capa de regla del negocio. Dentro de las principales funcionalidades de esta capa esta el interactuar con diferentes componentes de la institución que realizan las operaciones bancarias solicitadas por el cliente. A continuación se describen los métodos y funciones diseñadas y la forma en la que tienen que ser invocados principalmente por la clase

ServiciosElectronicos que es la que interactúa con los diferentes sistemas usuarios que necesitan consumir las operaciones banca celular.

El diagrama de estructura estática se ha generado con Visio 2003 con lenguaje UML. Refiérase al anexo 4.3 en donde se presenta el diagrama de clase del objeto ServiciosOperacion.Interfaz.

A continuación se presenta la documentación técnica de los diferentes métodos implementados para el acceso a las Funciones de BackOffice y a las diferentes Funciones de las operaciones PULL y PUSH.

4.4.4.1 Método EnviaAPPC().

Descripción: Método que realice la consulta del estatus del cheque al Host de la institución bancaria.
Visibilidad: Privada.
Parámetros de entrada:
TerNumCue, variable de tipo string en donde se especifica el número de cuenta.
TerNumChe, variable de tipo string en donde se especifica el número de cheque.
Parámetros de salida:
RespIsTranOk, variable de tipo de boolean, que indica si el proceso de consulta se realizo con éxito.
DataOutput, variable de tipo string. Tiene la siguiente estructura.
CodRet: char(3), código de retorno.
MenEstChe: varchar(145), estado del cheque. Si CodRet = “000” los siguientes valores pueden tomar esta variable:
MenEstChe = “FUE PAGADO”
MenEstChe = “NO FUE PAGADO”
MenEstChe = ”CON ESTATUS DESCONOCIDO”
Si CodRet <> “000” entonces MenEstChe viene el mensaje del error presentado.

4.4.4.2 Método EnviaEMail().

Descripción: Método que envía correo electrónico.
Visibilidad: Privada.

Parámetros de entrada:

strDestino, variable de tipo string en donde se especifica la dirección de correo electrónico a donde se desea enviar el mail. Si desea enviar a más de uno debe enviar las direcciones separados por punto y coma.

strSubject, variable de tipo string en donde se especifica el titulo del correo electrónico.

strMensaje, variable de tipo string en donde se especifica el mensaje a enviar en el correo electrónico.

Parámetros de salida:

RespIsTranOk, variable de tipo bolean, TRUE indica que la ejecución de las instrucciones dentro del método fueron OK, false indica que se presento algún error dentro del método.

RespData, variable de tipo string, en caso de que RespIsTranOk=True, este parámetros viene en blanco, en caso de sea False en este parámetro viene el mensaje de error.

4.4.4.3 Método mensajeError().

Descripción: Método que recupera desde la base de datos la descripción de un error especifico. Adicionalmente graba una bitácora de errores. Clasifica los errores en operativos o de sistema.

Visibilidad: Publico.

Parámetros de entrada:

strTipErr, variable de tipo string en donde se especifica el tipo de error. “O”= Operativo; “S”=Sistema.

strMetFunSubGlo, variable de tipo string en donde se indica el nombre del método, función o subrutina en donde se produce el error.

intCodErr, variable de tipo entero en donde se especifica el error que se ha producido.

lngErrNumber, parámetro de tipo long. Para errores operativos normalmente este parámetro va en -1. Para errores de sistema o errores inesperados, se recupera el err.number que devuelve la excepción para grabar en bitácora.

strErrDescription, parámetro de tipo string. Para errores operativos normalmente este parámetro va en blanco. Para errores de sistema o errores inesperados, se recupera el err.description que devuelve la excepción para grabar en bitácora.

Parámetros de salida:

RespData, variable de tipo string. Tiene la siguiente estructura.

CodRet: char(3), código de retorno.

Mensaje: varchar(145), mensaje de error que se va a desplegar al usuario.

4.4.4.4 Función getError().

Descripción: Esta función recupera los datos asociados a un error específico.

Visibilidad: Privada.

Parámetros de entrada:

intCodError, variable de tipo entero, en done se especifica el código de error que se desea recuperar desde la base de datos.

Parámetros de salida:

getError: Variable de tipo recordset. A continuación se explican los atributos que devuelve.

TE-CodErr, variable de tipo entero, representa el código de error encontrado.

TE-DesUsu, variable de tipo string, representa el mensaje de error que se va a desplegar al usuario.

TE-DesSis, variable de tipo string, representa el mensaje que descrie del error del sistema presentado.

4.4.4.5 Método InsertaLogError().

Descripción: Este método graba una bitácora de errores en la base de datos.

Visibilidad: Privada.

Parámetros de entrada:

strTipErr, variable de tipo string, puede ser “O” para errores operativos, “S” para errores de sistema.

lngErrNumber, variable de tipo long, se especifica el número del error producido en la excepción.

strErrDescription, variable de tipo string, se especifica el número del error producido en la excepción.

intCodErr, variable de tipo entero, se especifica el número de error controlado.

strDesUsu, variable de tipo string, se especifica el mensaje que se va a desplegar la usuario.

strDesSis, variable de tipo string, se especifica el mensaje de sistema producido.

strPrograma, variable de tipo string en donde se especifica el nombre del programa en donde se produce el error.
srMetSubFun, variable de tipo string en donde se especifica el nombre del método, subrutina o función.
Parámetros de salida:
RespIsTranOk, variable de tipo bolean, trae indica que la ejecución de las instrucciones dentro del método fueron OK, false indica que se presento algún error dentro del método.
RespData, variable de tipo string, en caso de que RespIsTranOk=True, este parámetros viene en blanco, en caso de sea False en este parámetro viene el mensaje de error.

4.4.4.6 Método SetConeccionTarCre().

Descripción: Su función es la de establecer la comunicación con la base de datos de tarjetas de crédito.
Visibilidad: Privada.
Parámetros de entrada:
Parámetros de salida:
sMensaje, variable de tipo string, indica el mensaje de error presentado al intentar conectar a la base de datos.
lRetornoTarCre, variable de tipo long, si es diferente de cero significa que no se pudo establecer comunicación con la base de datos y viene el mensaje de error en sMensaje.

4.4.4.7 Método SetConeccionSmsAdmin().

Descripción: Su función es la de establecer la comunicación con la base de datos de la pasarela.
Visibilidad: Privada.
Parámetros de entrada:
Parámetros de salida:
sMensaje, variable de tipo string, indica el mensaje de error presentado al intentar conectar a la base de datos.
lRetornoSMSAdmin, variable de tipo long, si es diferente de cero significa que no se

pudo establecer comunicación con la base de datos y viene el mensaje de error en sMensaje.

4.4.4.8 Método SetConeccionTarDeb().

Descripción: Su función es la de establecer la comunicación con la base de datos de tarjeta de débito.
Visibilidad: Privada.
Parámetros de entrada:
Parámetros de salida:
sMensaje, variable de tipo string, indica el mensaje de error presentado al intentar conectar a la base de datos.
lRetornoTarDeb, variable de tipo long, si es diferente de cero significa que no se pudo establecer comunicación con la base de datos y viene el mensaje de error en sMensaje.

4.4.4.9 Método ConsultaTarCre().

Descripción: Su función es la de consultar por los datos de una determinada tarjeta de crédito. Devuelve los datos siempre y cuando la tarjeta este activa, caso contrario indica el error.
Visibilidad: Privada.
Parámetros de entrada:
TerNumTar, variable de tipo string, indica el número de la tarjeta de crédito que se desea consultar.
Parámetros de salida:
RespTarHab, variable de tipo string, indica el nombre del dueño de la tarjeta.
RespTipTarHab, variable de tipo string, indica el tipo de tarjetahabiente. Los valores pueden ser "V" – Vip, RespTipTarHab = "P" - Vip plus, "N" – Normal.
RespIdeCli, variable de tipo string, los valores pueden ser C=Cedula, R=RUC, P=Pasaporte.
RespIsTranOk, variable de tipo bolean, trae indica que la ejecución de las instrucciones dentro del método fueron OK, false indica que se presento algún error dentro del método.

DataOutput, variable de tipo string, en caso de que RespIsTranOk=True, este parámetros viene en blanco, en caso de sea False en este parámetro viene el mensaje de error.

4.4.4.10 **Método ConsultaTarDeb().**

Descripción: Su función es la de consultar por los datos de una determinada tarjeta de débito. Devuelve los datos siempre y cuando la tarjeta este activa, caso contrario indica el error.

Visibilidad: Privada.

Parámetros de entrada:

ByVal TerNumTar As String

Parámetros de salida:

RespTarHab, variable de tipo string, indica el nombre del dueño de la tarjeta.

RespTipTarHab, variable de tipo string, indica el tipo de tarjetahabiente. Los valores pueden ser "V" – Vip, RespTipTarHab = "P" - Vip plus, "N" – Normal, “E” - Empleado.

RespIdeCli, variable de tipo string, los valores pueden ser C=Cedula, R=RUC, P=Pasaporte.

RespIsTranOk, variable de tipo boolean, trae indica que la ejecución de las instrucciones dentro del método fueron OK, false indica que se presento algún error dentro del método.

DataOutput, variable de tipo string, en caso de que RespIsTranOk=True, este parámetros viene en blanco, en caso de sea False en este parámetro viene el mensaje de error.

4.4.4.11 **Función ConsultaOperadora().**

Descripción: función que devuelve el nombre de la operadora celular al que pertenece el número celular.

Parámetro de entrada:

Celular, variable de tipo string, se indica el número de teléfono del que se desea recupera la operadora.

Parámetro de salida:

ConsultaOperadora, variable de tipo string que indica la operadora. En caso de no encontrar una operadora celular asociada al número se devuelve en blanco. Los valores a devolver son “P”=Porta Celular; “A”=Alegro; “B”=MoviStar.

4.4.4.12 Función ObtieneOperadora().

Descripción: Nombre de la operadora celular asociada al número celular.

Parámetro de entrada:

TerNumCel, variable de tipo string, número de celular del que se desea obtener la operadora.

Parámetro de salida:

ObtieneOperadora, variable de tipo string, en caso de no encontrar una operadora celular asociada al número de devuelve “OPER. DESC”. Los valores a devolver son “PORTA”, “ALEGRO”, “MOVISTAR”.

4.4.4.13 Método VerificaSuscripcionClave().

Descripción: Método que verifica que la suscripción exista y valida que la clave sea correcta.

Visibilidad: Privada.

Parámetros de entrada:

TerNumTar, variable de tipo string, indica el número de tarjeta que se desea verificar la suscripción y la clave.

TerNumCel, variable de tipo string, número celular de suscripción.

TerCodSer, variable de tipo string, código de servicio, para caso de banca celular “SMS”.

TerTieCla, variable de tipo string, código que indica si se debe verificar o no la clave, “0” – No se debe verificar la clave, “1” – Se debe verificar la clave.

TerClave, variable de tipo string, clave banca celular asociada a la tarjeta que se va a verificar.

Parámetros de salida:

RespCodSus, variable de tipo long que devuelve el código de suscripción asociado al celular y a la tarjeta.

RespIsTranSusOk, variable de tipo Boolean, TRUE indica que existe la suscripción

y esta active, FALSE indica que la suscripción no existe o no esta active. Siempre la descripción del error viene en DataOutput.
RespIsTranClaOk, variable de tipo Boolean, TRUE indica que la clave banca celular es correcta, FALSE indica que la clave es incorrecta.
DataOutput, variable de tipo string. Tiene la siguiente estructura.
CodRet: char(3), código de retorno.
Mensaje: varchar(145), mensaje de error que se va a desplegar al usuario.

4.4.4.14 **Método VerificaSuscripcionE().**

Descripción: Método que verifica si una suscripción adicional esta active.
Visibilidad: Privada.
Parámetros de entrada:
TerNumTar, variable de tipo string, indica el número de tarjeta que se desea verificar la suscripción y la clave.
TerNumCel, variable de tipo string, número celular de suscripción.
TerCodSer, variable de tipo string, código de servicio, para caso de banca celular "SMS".
TerCodOpe, variable de tipo string, código de operación del servicio adicional a verificar.
Parámetros de salida:
RespIsTranOk, variable de tipo Boolean, TRUE indica
DataOutput variable de tipo string, que tiene la siguiente estructura.
CodRet: char(3), código de retorno.
Mensaje: varchar(145), mensaje de error que se va a desplegar al usuario.

4.4.4.15 **Método ConsultaSusNorOEspAct().**

Descripción: Método cuya función principal es el de consultar si la suscripción de un cliente banca celular, existe, si esta suscrito en e mismo día que llega este requerimiento y recupera en un recordset los datos de las suscripciones adicionales que tenga.
Visibilidad: Privada.

Parámetros de entrada:

TerNumTar, variable de tipo string, indica el número de tarjeta que se desea verificar la suscripción y la clave.

TerNumCel, variable de tipo string, número celular de suscripción.

TerCodSer, variable de tipo string, código de servicio, para caso de banca celular "SMS".

TerVerSusHoy, variable de tipo bolean, si es TRUE indica si la suscripción fue realizada en la fecha actual y FALSE no hace esta verificación solo si esta suscrito.

Parámetros de salida:

TerSusSE, variable de tipo recordset, devuelve las suscripciones adicionales relacionadas con la suscripción principal.

RespFecSus, variable de tipo string en donde se devuelve la fecha de suscripción principal.

RespHorSus, variable de tipo string en donde se devuelve la hora de suscripción principal.

RespIsTranOk, variable de tipo Boolean, TRUE indica que el método encontró la suscripción normal asociada. FALSE indica que se produjo un error.

DataOutput variable de tipo string, que tiene la siguiente estructura.

CodRet: char(3), código de retorno.

Mensaje: varchar(145), mensaje de error que se va a desplegar al usuario.

4.4.4.16 **Método ConsultaSusAutAct().**

Descripción: Método cuya función principal es el de consultar si la suscripción de un cliente banca celular, existe, si esta suscrito en e mismo día que llega este requerimiento y recupera en un recordset los datos de las suscripciones adicionales que tenga.

Visibilidad: Privada.

Parámetros de entrada:

TerNumTar, variable de tipo string, indica el número de tarjeta que se desea verificar la suscripción y la clave.

TerNumCel, variable de tipo string, número celular de suscripción.

TerCodSer, variable de tipo string, código de servicio, para caso de banca celular "SMS".

TerVerSusHoy, variable de tipo bolean, si es TRUE indica si la suscripción fue realizada en la fecha actual y FALSE no hace esta verificación solo si esta suscrito.

Parámetros de salida:

TerSusSE, variable de tipo recordset, devuelve las suscripciones adicionales relacionadas con la suscripción principal.

RespFecSus, variable de tipo string en donde se devuelve la fecha de suscripción principal.

RespHorSus, variable de tipo string en donde se devuelve la hora de suscripción principal.

RespIsTranOk, variable de tipo Boolean, TRUE indica que el método encontró la suscripción normal asociada. FALSE indica que se produjo un error.

DataOutput variable de tipo string, que tiene la siguiente estructura.

CodRet: char(3), código de retorno.

Mensaje: varchar(145), mensaje de error que se va a desplegar al usuario.

4.4.4.17 **Función OperacionAutorizada().**

Descripción: Función que devuelve los códigos de operación de las transacciones que necesitan autorización en función del BIN de la tarjeta.

Visibilidad: Publico.

Parámetros de entrada:

TerCodSer, variable de tipo string, código de servicio, para el caso de banca celular "SMS".

TerCodRedAdq, variable de tipo string en donde se especifica el código de la red adquiriente. "SMS-Short Message Service", "ITR" – Intranet, "CMS" – Card Management System.

TerBinTar, variable de tipo string en donde se especifica el BIN de la tarjeta.

TerEsLocal, variable de tipo string, que especifica si el bin es de la institución local "1". Si es institución con convenio es "0".

Parámetros de salida:

RespIsTranOk, variable de tipo boolean, trae indica que la ejecución de las instrucciones dentro del método fueron OK, false indica que se presento algún error dentro del método.

RespData, variable de tipo string, en caso de que RespIsTranOk=True, este parámetros viene en blanco, en caso de sea False en este parámetro viene el

mensaje de error.
OperacionAutorizada variable de tipo recordset, devuelve los datos de las transacciones que requieren autorización.

4.4.4.18 Función MotivoInactivacion().

Descripción: función que devuelve los códigos de motivos por las que se puede desactivar una suscripción banca celular.
Visibilidad: Público.
Parámetros de salida:
RespIsTranOk, variable de tipo boolean, trae indica que la ejecución de las instrucciones dentro del método fueron OK, false indica que se presento algún error dentro del método.
RespData, variable de tipo string, en caso de que RespIsTranOk=True, este parámetros viene en blanco, en caso de sea False en este parámetro viene el mensaje de error.
MotivoInactivacion, variable de tipo recordset, devuelve el código de motivo y descripción del motivo por la que se puede desactivar una suscripción principal y adicional.

4.4.4.19 Función ConsultaClave().

Descripción: función que consulta la clave de suscripción asociada a la suscripción banca celular y la convierte en decimal.
Visibilidad: Privada.
Parámetros de entrada:
TerCodSer, variable de tipo string, código de servicio, para caso de banca celular "SMS".
TerNumCel, variable de tipo string, número celular de suscripción.
TerNumTar, variable de tipo string, indica el número de tarjeta que se desea verificar la suscripción y la clave.
Parámetros de salida:
RespIsTranOk, variable de tipo boolean, "TRUE" indica que la ejecución de las instrucciones dentro del método fueron OK, "FALSE" indica que se presento algún

error dentro del método.

RespData, variable de tipo string, en caso de que RespIsTranOk=True, este parámetros viene en blanco, en caso de sea False en este parámetro viene el mensaje de error.

4.4.4.20 **Función TarjetaCuenta().**

Descripción: Función que devuelve el número de cuenta principal asociada a la tarjeta de débito.

Visibilidad: Privada.

Parámetros de entrada:

TerNumTar, variable de tipo string, indica el número de tarjeta que se desea verificar la suscripción y la clave.

Parámetros de salida:

RespIsTranOk, variable de tipo boolean, “TRUE” indica que la ejecución de las instrucciones dentro del método fueron OK, “FALSE” indica que se presento algún error dentro del método.

DataOutput variable de tipo string, que tiene la siguiente estructura.

CodRet: char(3), código de retorno.

Mensaje: varchar(145), mensaje de error que se va a desplegar al usuario.

4.4.4.21 **Método InsertaSuscripcion().**

Descripción: Método que realiza la inserción de una suscripción banca celular en la base de datos.

Visibilidad: Privada.

Parámetros de entrada:

TerCodSer, variable de tipo string, código de servicio, para el caso de banca celular “SMS”.

TerCodRedAdq, variable de tipo string en donde se especifica el código de la red adquiriente. “SMS-Short Message Service”, “ITR” – Intranet, “CMS” – Card Management System.

TerCodTra, variable de tipo string, se debe especificar el código de transacción relacionada con la operación de suscripción.

TerIdeDisOri, variable de tipo string, se debe especificar la identificación del dispositivo desde donde se origina la transacción.

TerUsuOri, variable de tipo string, se debe especificar la identificación del usuario que origina la transacción.

TerNomOpe, variable de tipo string, se especifica el nombre de la operadora al cual pertenece el teléfono.

TerNumCel, variable de tipo string, se especifica el número de celular a suscribir.

TerProBan, variable de tipo string, indica el producto al que pertenece la tarjeta de débito/crédito. Los producto pueden ser “TD” – Tarjeta de débito, “VI” – Tarjeta de crédito VISA, “MC” – Tarjeta de crédito MASTERCARD, “TC” – tarjeta de crédito tarjeta club.

TerEsLocal, variable de tipo string, indica si la tarjeta de débito/crédito pertenece a la institución propia o a una institución afiliada. “1” – Local, “0” – Afiliada.

TerTipTarHab, variable de tipo string, indica el tipo de cliente “N” – Normal, “V” – Vip, “P” – Vip Plus, “E” – Empleado.

TerNomCli, variable de tipo string, indica el nombre del cliente dueño de la tarjeta de débito/crédito.

TerNumTar, variable de tipo string, se indica el número de tarjeta de débito/crédito que esta utilizando para la suscripción.

TerIdeCli variable de tipo string, indica el tipo de identificación del cliente. “C” – Cédula, “R” – RUC, “P” = Pasaporte.

TerCla variable de tipo string, se debe especificar la clave asociada a la suscripción banca celular.

TerTransaccion, variable de tipo recordset, se especifican los datos de la transacción asociada a la operación de suscripción.

Parámetros de salida:

RespIsTranOk, variable de tipo boolean, “TRUE” indica que la ejecución de las instrucciones dentro del método fueron OK, “FALSE” indica que se presento algún error dentro del método.

DataOutput variable de tipo string, que tiene la siguiente estructura.

CodRet: char(3), código de retorno.

Mensaje: varchar(145), mensaje de error que se va a desplegar al usuario.

4.4.4.22 **Método InsertaTransaccionSerElec().**

Descripción: Método que inserta en el registro transaccional todas las operaciones solicitadas por los diferentes canales. Posteriormente se podrá consultar esta información para soporte al cliente, estadísticas, atención de reclamos, etc.

Visibilidad: Privada.

Parámetros de entrada:

TerFecTra, variable de tipo string, se especifica la fecha de la transacción en el formato “aaaa/MM/dd”.

TerHorTra, variable de tipo string, se especifica la hora de la transacción en el formato “hhmmss00”.

TerCodSer, variable de tipo string, código de servicio, para el caso de banca celular “SMS”.

TerCodRedAdq, variable de tipo string en donde se especifica el código de la red adquiriente. “SMS-Short Message Service”, “ITR” – Intranet, “CMS” – Card Management System.

TerCodTra, variable de tipo string, se debe especificar el código de transacción relacionada con la operación de suscripción.

TerCodOpe, variable de tipo string, se debe especificar el código de operación banca celular.

TerIdeDisOri, variable de tipo string, se debe especificar la identificación del dispositivo desde donde se origina la transacción.

TerUsuOri, variable de tipo string, se debe especificar la identificación del usuario que origina la transacción.

TerSupOri, variable de tipo string, para transacciones con cierto nivel de seguridad, se debe especificar el código del usuario supervisor utilizado.

TerSupNom variable de tipo string, para transacciones con cierto nivel de seguridad, se debe especificar el nombre del usuario supervisor utilizado.

TerNomOpe, variable de tipo string, se especifica el nombre de la operadora al cual pertenece el teléfono.

TerNumCel, variable de tipo string, se especifica el número de celular a suscribir.

TerProBan, variable de tipo string, indica el producto al que pertenece la tarjeta de débito/crédito. Los producto pueden ser “TD” – Tarjeta de débito, “VI” – Tarjeta de crédito VISA, “MC” – Tarjeta de crédito MASTERCARD, “TC” – tarjeta de crédito tarjeta club.

TerNumTar, variable de tipo string, se indica el número de tarjeta de débito/crédito que esta utilizando para la suscripción.

TerTipTarHab variable de tipo string, indica el tipo de identificación del cliente. “C” – Cédula, “R” – RUC, “P” = Pasaporte.

TerTipTarHab, variable de tipo string, indica el tipo de cliente “N” – Normal, “V” – Vip, “P” – Vip Plus, “E” – Empleado.

Parámetros de salida:

RespIsTranOk, variable de tipo boolean, “TRUE” indica que la ejecución de las instrucciones dentro del método fueron OK, “FALSE” indica que se presento algún error dentro del método.

DataOutput variable de tipo string, que tiene la siguiente estructura.

CodRet: char(3), código de retorno.

Mensaje: varchar(145), mensaje de error que se va a desplegar al usuario.

Parámetros de entrada opcionales:

Los siguientes campos son opcionales ya que pueden utilizarse como complemento informativo para identificar una transacción realizada a través de banca celular.

TerUsuDes, variable de tipo string, se especifica el usuario destino al que va destinada la operación.

TerMenDes, variable de tipo string, se especifica el mensaje destino que fue enviado al usuario.

TerInfo1, variable de tipo string, se puede utilizar para graba información importante por concepto de auditoria.

TerInfo2, variable de tipo string, se puede utilizar para graba información importante por concepto de auditoria.

4.4.4.23 **Método InsertaSuscripcionPasarela().**

Descripción: Método que reporta la suscripción en la pasalela SMS.

Visibilidad: Privada.

Parámetros de entrada:

i_tr, variable de tipo integer, número de transacción asignada 21.

i_login, variable de tipo string, usuario que accede al servicio.

i_operacion, tipos de operación que soporta esta transacción: “REV” – Reverso de suscripción en pasarela SMS, “INS” – Inserta suscripción en pasarela SMS.

i_cliente, variable de tipo integer, tipo de identificación del cliente. 0 = Cedula.

i_cedula, variable de tipo string, se especifica la cedula del cliente.

i_servicio, variable tipo string, envío en blanco.

i_producto, variable de tipo string, “TD” – Tarjeta de débito, “VI” – Visa, “MC” – MasterCard, “TC” – Tarjeta club.

i_cuenta, variable de tipo string, se especifica el número de tarjeta de suscripción.

i_telefono, variable de tipo string, se especifica el número celular.

i_tipo, variable de tipo string, se especifica en blanco.

i_nuevo_telf, variable de tipo string, se especifica en blanco.

i_mensaje, variable de tipo string, se especifica en blanco.

Parámetros de salida:

RespIsTranOk, variable de tipo boolean, “TRUE” indica que la ejecución de las instrucciones dentro del método fueron OK, “FALSE” indica que se presento algún error dentro del método.

DataOutput variable de tipo string, que tiene la siguiente estructura.

CodRet: char(3), código de retorno.

Mensaje: varchar(145), mensaje de error que se va a desplegar al usuario.

4.4.4.24 **Método EliminaSusPasarela().**

Descripción: Método que elimina la suscripción en la pasarela SMS, saca un respaldo de la suscripción y elimina el registro de suscripción.

Parámetros de entrada:

TerNumTar, variable de tipo string, se indica el número de tarjeta de débito/crédito que esta utilizando para la suscripción.

TerNumCel, variable de tipo string, se especifica el número de celular a suscribir.

Parámetros de salida:

RespIsTranOk, variable de tipo boolean, “TRUE” indica que la ejecución de las instrucciones dentro del método fueron OK, “FALSE” indica que se presento algún error dentro del método.

DataOutput variable de tipo string, que tiene la siguiente estructura.

CodRet: char(3), código de retorno.

Mensaje: varchar(145), mensaje de error que se va a desplegar al usuario.

4.4.4.25 **Método ActualizaTransaccionSerElec().**

Descripción: Después de ejecutar cualquier operación bancaria este método actualiza el registro transaccional con el resultado de la ejecución de la operación bancaria solicitada.

Visibilidad: Privada.

Parámetros de entrada:

TerFecTra, variable de tipo string, se especifica la fecha de la transacción en el formato “aaaa/MM/dd”.

TerHorTra, variable de tipo string, se especifica la hora de la transacción en el formato “hhmmss00”.

TerCodSer, variable de tipo string, código de servicio, para el caso de banca celular “SMS”.

TerCodRedAdq, variable de tipo string en donde se especifica el código de la red adquiriente. “SMS-Short Message Service”, “ITR” – Intranet, “CMS” – Card Management System.

TerCodTra, variable de tipo string, se debe especificar el código de transacción relacionada con la operación de suscripción.

TerCodOpeSE, variable de tipo string, se debe especificar el código de operación adicional banca celular.

TerIdeDisOri, variable de tipo string, se debe especificar la identificación del dispositivo desde donde se origina la transacción.

TerUsuOri, variable de tipo string, se debe especificar la identificación del usuario que origina la transacción.

TerNumCel, variable de tipo string, se especifica el número de celular a suscribir.

TerNumTar, variable de tipo string, se indica el número de tarjeta de débito/crédito que esta utilizando para la suscripción.

TerTipCueDes variable de tipo entero, se especifica el tipo de cuenta de donde se realizó el débito por comisión. 10 = Ahorros; 20=Corriente.

TerCueDes variable de tipo string, se especifica el número de la cuenta en donde se débito el valor de la comisión.

TerValComAdq variable de tipo double, valor de comisión que gana la red adquiriente.

TerValComAut variable de tipo double, valor de comisión que gana la red autorizador.

TerValComTot variable de tipo double, valor total de comisión que se cobra al cliente.

TerNumAut variable de tipo string, para los clientes suscritos con tarjeta de crédito los cobros se los hacen como si fueran transacciones de compra vía POS.

TerCodRetBan variable de tipo entero, código de respuesta devuelto por un componente previo la ejecución de una operación bancaria.
TerCodRetSMS variable de tipo string, devuelve el código de respuesta enviado a la pasarela.
TerCodRetOpe variable tipo entero, sirve para retroalimentar los estados que tiene un mensaje después de que la pasarela haya enviado el mensaje al operador celular.
TerMsgErr variable de tipo string, se guarda el mensaje de error presentado, luego de la ejecución de una operación bancaria.
TerMenDes variable de tipo string, actualiza el mensaje enviado al cliente.

Parámetros de salida:

RespIsTranOk, variable de tipo boolean, “TRUE” indica que la ejecución de las instrucciones dentro del método fueron OK, “FALSE” indica que se presento algún error dentro del método.
DataOutput variable de tipo string, que tiene la siguiente estructura.
CodRet: char(3), código de retorno.
Mensaje: varchar(145), mensaje de error que se va a desplegar al usuario.

Parámetros de salida opcionales:

TerTipPas variable de tipo string, tipo de medio que se utilizó para enviar el mensaje “E” – Email, “L=PasarelaSMS”, “S=Selecciona la mejor”

4.4.4.26 Método InactivaSuscripcionNorOAut().

Descripción: Método que desactiva el registro de una suscripción normal y suscripciones a servicios adicionales que estén registrados con el número de tarjeta y el número de celular.
Visibilidad: Privado.
Parámetros de entrada:

TerTipInaSus variable de tipo string, indica el tipo de inactivación que se va a realizar.
“N”=inactivación de suscripción normal y autorizadas;
“A”=Inactiva suscripción autorizada;
“S”=Reverso de suscripción normal;
”I”=Reverso de suscripción adicional.
TerCodSer, variable de tipo string, código de servicio, para el caso de banca celular

“SMS”.

TerCodOpeSE, variable de tipo string, se debe especificar el código de operación adicional banca celular.

TerNumCel, variable de tipo string, se especifica el número de celular a suscribir.

TerNumTar, variable de tipo string, se indica el número de tarjeta de débito/crédito que esta utilizando para la suscripción.

TerStatus variable de tipo string.

“0”=inactivación.

“1”=Reverso.

TerCodRedAdqUltMod, variable de tipo string en donde se especifica el código de la red adquiriente. “SMS-Short Message Service”, “ITR” – Intranet, “CMS” – Card Management System.

TerIdeDisUltMod, variable de tipo string, se debe especificar la identificación del dispositivo desde donde se origina la transacción.

TerUsuUltMod, variable de tipo string, se debe especificar la identificación del usuario que origina la transacción.

TerUsuDesUltMod, variable de tipo string, se debe especificar una breve descripción de la acción realizada sobre el registro de suscripción

TerIdeCli variable de tipo string, indica el tipo de identificación del cliente. “C” – Cédula, “R” – RUC, “P” = Pasaporte.

TerProBan, variable de tipo string, indica el producto al que pertenece la tarjeta de débito/crédito. Los producto pueden ser “TD” – Tarjeta de débito, “VI” – Tarjeta de crédito VISA, “MC” – Tarjeta de crédito MASTERCARD, “TC” – tarjeta de crédito tarjeta club.

TerCodMot variable de tipo string, indica el código del motivo de inactivación.

01 INACTIVA POR SOLICITUD DEL CLIENTE

02 INACTIVA POR CAMBIO DE CELULAR

03 INACTIVA POR FALTA DE PAGO

04 REVERSADA POR EL EJECUTIVO DE ATENCION AL CLIENTE

Parámetros de salida:

RespIsTranOk, variable de tipo boolean, “TRUE” indica que la ejecución de las instrucciones dentro del método fueron OK, “FALSE” indica que se presento algún error dentro del método.

DataOutput variable de tipo string, que tiene la siguiente estructura.

CodRet: char(3), código de retorno.

Mensaje: varchar(145), mensaje de error que se va a desplegar al usuario.

4.4.4.27 **Método InsertaSuscripcionSE().**

Descripción: Método que inserta la suscripción de un servicio adicional.

Parámetros de entrada:

TerCodSer, variable de tipo string, código de servicio, para el caso de banca celular "SMS".

TerCodRedAdq, variable de tipo string en donde se especifica el código de la red adquiriente. "SMS-Short Message Service", "ITR" – Intranet, "CMS" – Card Management System.

TerCodTra, variable de tipo string, se debe especificar el código de transacción relacionada con la operación de suscripción al servicio adicional.

TerIdeDisOri, variable de tipo string, se debe especificar la identificación del dispositivo desde donde se origina la transacción.

TerUsuOri, variable de tipo string, se debe especificar la identificación del usuario que origina la transacción.

TerNomOpe, variable de tipo string, se especifica el nombre de la operadora al cual pertenece el teléfono.

TerNumCel, variable de tipo string, se especifica el número de celular a suscribir.

TerProBan, variable de tipo string, indica el producto al que pertenece la tarjeta de débito/crédito. Los producto pueden ser "TD" – Tarjeta de débito, "VI" – Tarjeta de crédito VISA, "MC" – Tarjeta de crédito MASTERCARD, "TC" – tarjeta de crédito tarjeta club.

TerEsLocal, variable de tipo string, indica si la tarjeta de débito/crédito pertenece a la institución propia o a una institución afiliada. "1" – Local, "0" – Afiliada.

TerTipTarHab, variable de tipo string, indica el tipo de cliente "N" – Normal, "V" – Vip, "P" – Vip Plus, "E" – Empleado.

TerCodOpeSE variable de tipo string, indica el código de operación del servicio adicional a suscribir.

TerNumTar, variable de tipo string, se indica el número de tarjeta de débito/crédito que esta utilizando para la suscripción.

TerTransaccion, variable de tipo recordset, se especifican los datos de la transacción asociada a la operación adicional de suscripción.

TerCodSus variable de tipo long, código de suscripción asignado cuando realizo la suscripción principal.

Parámetros de salida:

RespIsTranOk, variable de tipo boolean, "TRUE" indica que la ejecución de las

instrucciones dentro del método fueron OK, “FALSE” indica que se presento algún error dentro del método.

DataOutput variable de tipo string, que tiene la siguiente estructura.

CodRet: char(3), código de retorno.

Mensaje: varchar(145), mensaje de error que se va a desplegar al usuario.

4.4.4.28 **Método CambiaPassword().**

Descripción: Método que registra una nueva clave en la suscripción banca celular del cliente.

Visibilidad: Público.

Parámetros de entrada:

TerCodSer, variable de tipo string, código de servicio, para el caso de banca celular “SMS”.

TerCodRedOri, variable de tipo string en donde se especifica el código de la red adquiriente. “SMS-Short Message Service”, “ITR” – Intranet, “CMS” – Card Management System.

TerIdeDisOri, variable de tipo string, se debe especificar la identificación del dispositivo desde donde se origina la transacción.

TerUsuOri, variable de tipo string, se debe especificar la identificación del usuario que origina la transacción.

TerNumCel, variable de tipo string, se especifica el número de celular a suscribir.

TerNumTar, variable de tipo string, se indica el número de tarjeta de débito/crédito que esta utilizando para la suscripción.

TerCla, variable de tipo string, contiene la nueva clave de suscripción banca celular.

TerPag As String, parámetro que indica si una suscripción esta pagada.

Parámetros de salida:

RespIsTranOk, variable de tipo boolean, “TRUE” indica que la ejecución de las instrucciones dentro del método fueron OK, “FALSE” indica que se presento algún error dentro del método.

DataOutput variable de tipo string, que tiene la siguiente estructura.

CodRet: char(3), código de retorno.

Mensaje: varchar(145), mensaje de error que se va a desplegar al usuario.

4.4.4.29 **Método ModificaSuscripcion().**

Descripción: Método cuya función principal es la de modificar el número celular de suscripción banca celular.

Visibilidad: Público.

Parámetros de entrada:

TerCodSer, variable de tipo string, código de servicio, para el caso de banca celular "SMS".

TerCodRedOri, variable de tipo string en donde se especifica el código de la red adquiriente. "SMS-Short Message Service", "ITR" – Intranet, "CMS" – Card Management System.

TerIdeDisOri, variable de tipo string, se debe especificar la identificación del dispositivo desde donde se origina la transacción.

TerUsuOri, variable de tipo string, se debe especificar la identificación del usuario que origina la transacción.

TerNumCel, variable de tipo string, se especifica el número celular de la suscripción actual.

TerNumTar, variable de tipo string, se indica el número de tarjeta de débito/crédito que esta utilizando para la suscripción.

TerNumCelNue, variable de tipo string, se especifica el número de celular nuevo a suscribir.

TerIdeCli, variable de tipo string, indica el tipo de identificación del cliente. "C" – Cédula, "R" – RUC, "P" = Pasaporte.

TerProBan, variable de tipo string, indica el producto al que pertenece la tarjeta de débito/crédito. Los producto pueden ser "TD" – Tarjeta de débito, "VI" – Tarjeta de crédito VISA, "MC" – Tarjeta de crédito MASTERCARD, "TC" – tarjeta de crédito tarjeta club.

Parámetros de salida:

RespIsTranOk, variable de tipo boolean, "TRUE" indica que la ejecución de las instrucciones dentro del método fueron OK, "FALSE" indica que se presento algún error dentro del método.

DataOutput variable de tipo string, que tiene la siguiente estructura.

CodRet: char(3), código de retorno.

Mensaje: varchar(145), mensaje de error que se va a desplegar al usuario.

4.4.4.30 Función sMsgClaHex().

Descripción: Transforma clave en binario.

Parámetro de entrada:

TerClave, variable de tipo string, se especifica la nueva de suscripción.

Parámetro de salida:

sMsgClaHex, variable de tipo string, devuelve la clave transformada a binario.

4.4.4.31 Método Suscribe().

Descripción: Método que realiza la suscripción de un cliente banca celular.

Visibilidad: Público.

Parámetros de entrada:

TerNumCel As String

TerNumTar As String

TerClave As String

TerTransaccion As Recordset

TerProBan As String

TerEsLocal As String

TerCodSer As String

TerCodRedAdq As String

TerIdeDisOri As String

TerUsuOri As String

Parámetros de salida:

RespIsTranOk, variable de tipo boolean, “TRUE” indica que la ejecución de las instrucciones dentro del método fueron OK, “FALSE” indica que se presento algún error dentro del método.

DataOutput variable de tipo string, que tiene la siguiente estructura.

CodRet: char(3), código de retorno.

Mensaje: varchar(145), mensaje de error que se va a desplegar al usuario.

4.4.4.32 Método Password().

Descripción: Método que cambia la clave de suscripción banca celular.

Visibilidad: Público.

Parámetros de entrada:

TerNumCel As String

TerNumTar As String

TerClaveAnt As String

TerClaveNue As String

TerTransaccion As Recordset

TerProBan As String

TerEsLocal As String

TerCodSer As String

TerCodRedAdq As String

TerIdeDisOri As String

TerUsuOri As String

Parámetro de salida:

RespIsTranOk, variable de tipo boolean, “TRUE” indica que la ejecución de las instrucciones dentro del método fueron OK, “FALSE” indica que se presento algún error dentro del método.

DataOutput variable de tipo string, que tiene la siguiente estructura.

CodRet: char(3), código de retorno.

Mensaje: varchar(145), mensaje de error que se va a desplegar al usuario.

4.4.4.33 **Método SuscribeAutorizada().**

Descripción: Método que realiza el registro de la suscripción de un servicio adicional banca celular.

Visibilidad: Público.

Parámetros de entrada:

TerNumCel As String

TerNumTar As String

TerCodOpeSE As String

TerClave As String

TerTransaccion As Recordset

TerProBan As String

TerEsLocal As String

TerCodSer As String

TerCodRedAdq As String

TerIdeDisOri As String

TerUsuOri As String

Parámetros de salida:

Parámetro de salida:

RespIsTranOk, variable de tipo boolean, “TRUE” indica que la ejecución de las instrucciones dentro del método fueron OK, “FALSE” indica que se presento algún error dentro del método.

DataOutput variable de tipo string, que tiene la siguiente estructura.

CodRet: char(3), código de retorno.

Mensaje: varchar(145), mensaje de error que se va a desplegar al usuario.

4.4.4.34 **Método VerificaTransaccion().**

Descripción: Cuando una transacción banca celular es solicitada este método verifica que la tarjeta y celular que realizan la solicitud tengan una suscripción al servicio y este activa, si la transacción requiere clave verifica su validez, y obtiene los datos de la tarjeta en caso de que este activa.

Visibilidad: Publico.

Parámetros de entrada:

TerNumCel As String

TerNumTar As String

TerCodSer As String

TerCodOpe As String

TerTieCla As String

TerCla As String

TerTieAut As String

TerProBan As String

Parámetro de salida:

RespTarHab, variable de tipo string, nombre del dueño de la tarjeta de débito/crédito.

RespTipTarHab, variable de tipo string, indica el tipo de cliente. El dominio es “V”-Vip; “P”-Vip Plus; “E”-Empleado; “N”-Normal.

RespIdeCli, variable de tipo string, se indica la identificación del cliente.

RespIsTranOk, variable de tipo boolean, “TRUE” indica que la ejecución de las instrucciones dentro del método fueron OK, “FALSE” indica que se presento algún

error dentro del método.

DataOutput variable de tipo string, que tiene la siguiente estructura.

CodRet: char(3), código de retorno.

Mensaje: varchar(145), mensaje de error que se va a desplegar al usuario.

4.4.4.35 **VerificaTransaccionCambioClave().**

Descripción: Cuando una transacción de cambio de clave es solicitada este método verifica que la tarjeta y celular que realizan la solicitud tengan una suscripción al servicio y este activa, si la transacción requiere clave verifica su validez, y obtiene los datos de la tarjeta en caso de que este activa.

Visibilidad: Privada.

Parámetros de entrada:

TerNumCel As String

TerNumTar As String

TerCodSer As String

TerCodOpe As String

TerTieCla As String

TerCla As String

TerTieAut As String

TerProBan As String

Parámetros de salida:

RespTarHab As String

RespTipTarHab As String

RespIdeCli As String

RespTerPag As String

RespIsTranOk, variable de tipo boolean, “TRUE” indica que la ejecución de las instrucciones dentro del método fueron OK, “FALSE” indica que se presento algún error dentro del método.

DataOutput variable de tipo string, que tiene la siguiente estructura.

CodRet: char(3), código de retorno.

Mensaje: varchar(145), mensaje de error que se va a desplegar al usuario.

4.4.4.36 **Método ObtieneComision().**

Descripción: Método cuya función principal es la de obtener los valores de comisión por transacción para la red adquiriente, red autorizadora.
Visibilidad: Privado.
Parámetros de entrada:
TerEsLocal As String
TerProBan As String
TerTipTarHab As String
TerTransaccion As Recordset
Parámetros de salida:
TerValComAdq variable de tipo double, que devuelve el valor de comisión de la red adquiriente.
TerValComAut, variable de tipo double, que devuelve el valor de comisión de la red autorizadora.

4.4.4.37 **Método SaldoTarCre().**

Descripción: Método que devuelve los saldos disponibles de una tarjeta de crédito.
Visiblidad: Público.
Parámetros de entrada:
TerNumCel As String
TerNumTar As String
TerCla As String
TerTransaccion As Recordset
TerProBan As String
TerEsLocal As String
TerCodSer As String
TerCodRedAdq As String
TerIdeDisOri As String
TerUsuOri As String
Parámetros de salida:
DataOutput variable de tipo string. Para mas detalles de su estructura refiérase a la documentación del parámetro de salida DataOutput de la operación PULL “SALTC”.

4.4.4.38 **Método InactivaSuscribeAutorizada().**

Descripción: Método cuya función principal es inactivar una suscripción al servicio adicional de una tarjeta y celular específicos.

Visibilidad: Público.

Parámetros de entrada:

TerNumCel As String

TerNumTar As String

TerCodOpeSE As String

TerCla As String, _

TerTransaccion As Recordset

TerProBan As String

TerEsLocal As String

TerCodSer As String

TerCodRedAdq As String

TerIdeDisOri As String

TerUsuOri As String

TerCodMot As String

Parámetro de salida:

DataOutput; Variable de tipo string. Los 3 primero caracteres el código de retorno e indica el resultado de la operación.

Retorno(3):

Si es “000”, Transacción OK. A continuación vienen los siguientes campos:

RespIdeCli(13): Identificación del cliente.

TerProBan(2): Tipo de producto de suscripción. Los valores de dominio son: “TD” – Tarjeta de débito; “VI” – Tarjeta de crédito; “MC”- Tarjeta de crédito; “TC” – Tarjeta club.

TerTarHabAux(30); Nombre del dueño de la tarjeta de débito/crédito.

RespTipTarHab(1): Tipo de tarjetahabiente. Los valores de dominio son: “N” – Norma; “V” - Vip; “P” – Vip Plus; “E” – Empleado.

TerEsLocal(1): Variable que indica si el producto pertenencia a institución local o pertenece a una institución afiliada. Los valores de dominio son: “1” – Local; “0” – Institución afiliada.

Si es <> “000”, Transacción ERROR, a continuación vienen los siguientes campos.

MensajeError(148) en caso de que este campo se devuelva en blanco, entonces el mensaje de error se toma de plantilla de la pasarela.

Ejemplo de una trama de salida – transacción OK:

”000

“0102777091 ”

“TD”

“LANDY ANDRADE JUAN DIEGO ”

“E”

“1”

Ejemplo de una trama de salida – transacción ERROR:

”067SERVICIO ADICIONAL NO ACTIVO”

4.4.4.39 **Método InactivaSuscribe().**

Descripción: Método cuya función principal es inactivar la suscripción principal y servicios adicionales que tiene un cliente asignado a una tarjeta de suscripción y número de teléfono celular.

Visibilidad; Público.

Parámetros de entrada:

ByVal TerNumCel As String

TerNumTar As String

TerCla As String

TerTransaccion As Recordset

TerProBan As String

TerEsLocal As String

TerCodSer As String

TerCodRedAdq As String

TerIdeDisOri As String

TerUsuOri As String

TerCodMot As String

Parámetro de salida:

DataOutput; Variable de tipo string. Los 3 primeros caracteres el código de retorno e indica el resultado de la operación.

Retorno(3):

Si es “000”, Transacción OK. A continuación vienen los siguientes campos:
RespIdeCli(13): Identificación del cliente.
TerProBan(2): Tipo de producto de suscripción. Los valores de dominio son: “TD” – Tarjeta de débito; “VI” – Tarjeta de crédito; “MC”- Tarjeta de crédito; “TC” – Tarjeta club.
TerTarHabAux(30); Nombre del dueño de la tarjeta de débito/crédito.
RespTipTarHab(1): Tipo de tarjetahabiente. Los valores de dominio son: “N” – Norma; “V” - Vip; “P” – Vip Plus; “E” – Empleado.
TerEsLocal(1): Variable que indica si el producto pertenencia a institución local o pertenece a una institución afiliada. Los valores de dominio son: “1” – Local; “0” – Institución afiliada.
Si es <> “000”, Transacción ERROR, a continuación vienen los siguientes campos.
MensajeError(148) en caso de que este campo se devuelva en blanco, entonces el mensaje de error se toma de plantilla de la pasarela.

4.4.4.40 **Método BinTar().**

Descripción: Método cuya función principal es el de obtener el bin de la tarjeta, verificar si el número de tarjeta pertenece a la institución local o a una institución afiliada y obtiene el tipo de producto al que pertenece el bin de la tarjeta.
Visibilidad: Privada.
Parámetro de entrada:
TerNumTar variable de tipo string, se indica el número o bin de la tarjeta de débito/crédito.
Parámetros de salida:
RespBinTar(), variable de tipo string, se coloca el bin de la tarjeta.
RespEsLocal, variable de tipo string, indica si el producto pertenece a la institución local o pertenece a una institución afiliada. Los valores de dominio son: “1” – Local; “0” – Institución afiliada.
RespProSMS, variable de tipo string, indica el tipo de producto al que pertenece la tarjeta de débito/crédito. Los valores de dominio son: “TD” – Tarjeta de débito; “VI” – Tarjeta de crédito; “MC”- Tarjeta de crédito; “TC” – Tarjeta club; “TE – Tarjeta Especial”.
RespIsTranOk, variable de tipo boolean, “TRUE” indica que la ejecución de las instrucciones dentro del método fueron OK, “FALSE” indica que se presento algún error dentro del método.

RespData, variable de tipo string, si el parámetro RespIsTranOk es “TRUE” este campo viene en blanco, si el parámetro RespIsTranOk= “FALSE”, en esta variable se indica el error presentado.

4.4.4.41 **Método RecuperaTranOriSerEle().**

Descripción: Recupera los datos en la bitácora transaccional de una suscripción adicional especifica. Esto se hacer para poder recuperar la comisión cobrada por la suscripción y la que deben acreditar después del reverso de la suscripción.
Visibilidad: Privado.
Parámetros de entrada:
TerNumTar, variable de tipo string, especifica el número de tarjeta de débito/crédito utilizada para la suscripción banca celular.
TerFecTra, variable de tipo string, especifica la fecha de transacción en formato yyyy/MM/dd.
TerHorTra, variable de tipo string, especifica la hora de transacción en formato hhmmss00.
TerCodOpe, variable de tipo string, se debe especificar el código “SUSSA”
Parámetros de salida:
rsTraSerEle, variable de tipo recordset, devuelve los atributos de la tabla TSET-SerEleTran.
RespIsTranOk, variable de tipo boolean, “TRUE” indica que la ejecución de las instrucciones dentro del método fueron OK, “FALSE” indica que se presento algún error dentro del método.
DataOutput variable de tipo string, que tiene la siguiente estructura.
CodRet: char(3), código de retorno.
Mensaje: varchar(145), mensaje de error que se va a desplegar al usuario.

4.4.4.42 **Método ReversaSuscribe().**

Descripción: Método cuya función principal es la de reversar una suscripción principal y servicios adicionales realizada en el mismo día de suscripción por un ejecutivo de atención al cliente y que haya generado cobro de comisiones. Esta opción únicamente es utilizada por el ejecutivo de atención al cliente en caso de

una suscripción errónea.

Visibilidad: Público.

Parámetros de entrada:

TerNumCel As String

TerNumTar As String

TerClave As String

TerTransaccion As Recordset

TerProBan As String

TerEsLocal As String

TerCodSer As String

TerCodRedAdq As String

TerIdeDisOri As String

TerUsuOri As String

TerSupOri As String

TerSupNom As String

Parámetro de salida:

DataOutput; Variable de tipo string. Los 3 primeros caracteres el código de retorno e indica el resultado de la operación.

Retorno(3):

Si es “000”, Transacción OK. A continuación vienen los siguientes campos:

RespIdeCli(13): Identificación del cliente.

TerProBan(2): Tipo de producto de suscripción. Los valores de dominio son: “TD” – Tarjeta de débito; “VI” – Tarjeta de crédito; “MC”- Tarjeta de crédito; “TC” – Tarjeta club.

TerTarHabAux(30); Nombre del dueño de la tarjeta de débito/crédito.

RespTipTarHab(1): Tipo de tarjetahabiente. Los valores de dominio son: “N” – Norma; “V” - Vip; “P” – Vip Plus; “E” – Empleado.

TerEsLocal(1): Variable que indica si el producto pertenencia a institución local o pertenece a una institución afiliada. Los valores de dominio son: “1” – Local; “0” – Institución afiliada.

DesRev(): Variable de tipo string en donde se indica los servicios adicionales que fueron reversados. No tiene un valor específico ya que por cada servicio adicional reversado en esta variable se especifica la siguiente información:

"SUSSA": código que indica que existía una suscripción adicional que fue reversada.

TO-CodOpeSA(5): En esta variable se indica el código de operación del servicio adicional revesado.

TSET-ComTot(12): Variable en donde se indica el valor de la comisión reversada por la suscripción al servicio por adicional. El formato es 10 enteros con 2 decimales sin separador decimal.
Si es <> “000”, Transacción ERROR, a continuación vienen los siguientes campos.
MensajeError(148) en caso de que este campo se devuelva en blanco, entonces el mensaje de error se toma de plantilla de la pasarela.

Ejemplo de una trama de salida – transacción OK:
”000
“0102777091 ”
“TD”
“LANDY ANDRADE JUAN DIEGO ”
“E”
“1”
“SUSSANOATM000000000100”
“SUSSANOCTC000000000030”

Ejemplo de una trama de salida – transacción ERROR:
”068SUSCRIPCION NO PUEDE SER REVERSADA. FUERA DE FECHA”

4.4.4.43 **Método ReversaSuscribeAutorizada().**

Descripción: Método cuya función principal es la de reversar una suscripción de un servicio adicional realizada en el mismo día por un ejecutivo de atención al cliente y que haya generado cobro de comisión. Esta opción únicamente es utilizada por el ejecutivo de atención al cliente en caso de una suscripción adicional errónea.
Visibilidad: Publico.
Parámetros de entrada:
TerNumCel As String
TerNumTar As String
TerCodOpeSE As String
TerClave As String
TerTransaccion As Recordset
TerProBan As String

TerEsLocal As String

TerCodSer As String

TerCodRedAdq As String

TerIdeDisOri As String

TerUsuOri As String

TerSupOri As String

TerSupNom As String

Parámetro de salida:

DataOutput; Variable de tipo string. Los 3 primeros caracteres el código de retorno e indica el resultado de la operación.

Retorno(3):

Si es “000”, Transacción OK. A continuación vienen los siguientes campos:

RespIdeCli(13): Identificación del cliente.

TerProBan(2): Tipo de producto de suscripción. Los valores de dominio son: “TD” – Tarjeta de débito; “VI” – Tarjeta de crédito; “MC”- Tarjeta de crédito; “TC” – Tarjeta club.

TerTarHabAux(30); Nombre del dueño de la tarjeta de débito/crédito.

RespTipTarHab(1): Tipo de tarjetahabiente. Los valores de dominio son: “N” – Norma; “V” - Vip; “P” – Vip Plus; “E” – Empleado.

TerEsLocal(1): Variable que indica si el producto pertenencia a institución local o pertenece a una institución afiliada. Los valores de dominio son: “1” – Local; “0” – Institución afiliada.

DesRev(): Variable de tipo string en donde se indica los servicios adicionales que fueron reversados. No tiene un valor específico ya que por cada servicio adicional reversado en esta variable se especifica la siguiente información:

"SUSSA": código que indica que existía una suscripción adicional que fue reversada.

TO-CodOpeSA(5): En esta variable se indica el código de operación del servicio adicional revesado.

TSET-ComTot(12): Variable en donde se indica el valor de la comisión reversada por la suscripción al servicio por adicional. El formato es 10 enteros con 2 decimales sin separador decimal.

Si es <> “000”, Transacción ERROR, a continuación vienen los siguientes campos.

MensajeError(148) en caso de que este campo se devuelva en blanco, entonces el mensaje de error se toma de plantilla de la pasarela.

Ejemplo de una trama de salida – Transacción OK:

”000”
“0102777091 ”
“TD”
“LANDY ANDRADE JUAN DIEGO ”
“E”
“1”
“SUSSANOATM000000000100”

Ejemplo de una trama de salida – Transacción ERROR:
”069SUSCRIPCION ADICIONAL NO PUEDE SER REVERSADA. FUERA DE FECHA”

4.4.4.44 Función DescripcionTarjeta().

Descripción: Función que obtiene la descripción del tipo producto ingresado.
Visibilidad: Privada.
Parámetro de entrada:
TerProBan, variable de tipo string, especifica el código del producto.
Parámetro de salida:
DescripcionTarjeta, variable de tipo string, devuelve la descripción del tipo de producto.

4.4.4.45 Método CupoTarCre().

Descripción: Consulta cupos de una tarjeta de crédito.
Visibilidad: Público.
Parámetros de entrada:
TerNumCel As String
TerNumTar As String
TerTipCup As String
TerCla As String
TerTransaccion As Recordset
TerProBan As String
TerEsLocal As String

TerCodSer As String
TerCodRedAdq As String
TerIdeDisOri As String
TerUsuOri As String

Parámetros de salida:

DataOutput variable de tipo string. Para mas detalles de su estructura refiérase a la documentación del parámetro de salida DataOutput de la operación PULL "CUPTC".

4.4.4.46 **Método ConsultaCheque().**

Descripción: Consulta el estado de un cheque girado por un cliente.

Visibilidad: Público.

Parámetros de entrada:

TerNumCel As String
TerNumTar As String
TerNumChe As String
TerCla As String
TerTransaccion As Recordset
TerProBan As String
TerEsLocal As String
TerCodSer As String
TerCodRedAdq As String
TerIdeDisOri As String
TerUsuOri As String

Parámetros de salida:

DataOutput variable de tipo string. Para mas detalles de su estructura refiérase a la documentación del parámetro de salida DataOutput de la operación PULL "CONCH".

4.4.4.47 **Método BloqueoTarjeta().**

Descripción: Método cuya función principal es la de bloquear temporalmente una tarjeta de débito/crédito.

Visibilidad: Público.

Parámetros de entrada:

ByVal TerNumCel As String

TerNumTar As String

TerClaveSE As String

TerCla As String

TerTransaccion As Recordset

TerProBan As String

TerEsLocal As String

TerCodSer As String

TerCodRedAdq As String

TerIdeDisOri As String

TerUsuOri As String

Parámetro de salida:

DataOutput variable de tipo string. Para mas detalles de su estructura refiérase a la documentación del parámetro de salida DataOutput de la operación PULL “BLTAR”.

4.4.4.48 Método SaldoTarDeb().

Descripción: Método que realiza la consulta de saldos de las cuentas asociadas a una tarjeta de débito.

Visibilidad: Público.

Parámetros de entrada:

TerNumCel As String

TerNumTar As String

TerTipCue As String

TerCla As String

TerTransaccion As Recordset

TerProBan As String

TerEsLocal As String

TerCodSer As String

TerCodRedAdq As String

TerIdeDisOri As String

TerUsuOri As String

Parámetro de salida:

DataOutput variable de tipo string. Para mas detalles de su estructura refiérase a la documentación del parámetro de salida DataOutput de la operación PULL "SALDO".

4.4.4.49 Método NotificacionOLBatchGenericaConSuscripcion().

Descripción: Método que realiza el envío de un mensaje SMS a un cliente suscrito al servicio banca celular y verifica que la operación utilizada sea de tipo push con suscripción.

Visibilidad: Público.

Parámetros de entrada:

TerTipPas, variable de tipo string, tipo de medio que se debe utilizar para enviar el mensaje. El dominio de la variable es el siguiente: "E" – Email, "L=PasarelaSMS", "S=Selecciona la mejor"

TerCodOpeOri: Código de operación autorizada desde donde se origina la transacción.

TerNumCel, variable de tipo string, se especifica el número de celular de suscripción y es el destino a donde se va a enviar el mensaje sms.

TerNumTar, variable de tipo string, se indica el número de tarjeta de débito/crédito de suscripción.

TerCla variable de tipo string, se debe especificar la clave asociada a la suscripción banca celular en caso de que la transacción de la operación push con suscripción este configurada con especificación de clave.

TerTransaccion, variable de tipo recordset, se especifican los datos de la transacción asociada a la operación push con suscripción.

TerProBan, variable de tipo string, indica el producto al que pertenece la tarjeta de débito/crédito. Los producto pueden ser "TD" – Tarjeta de débito, "VI" – Tarjeta de crédito VISA, "MC" – Tarjeta de crédito MASTERCARD, "TC" – tarjeta de crédito tarjeta club, "TE" – Tarjeta especial.

TerEsLocal, variable de tipo string, indica si la tarjeta de débito/crédito pertenece a la institución local o a una institución afiliada. El dominio de la variable es "1" – Local, "0" – Afiliada.

TerCodSer, variable de tipo string, código de servicio, para el caso de banca celular "SMS".

TerCodRedAdq, variable de tipo string en donde se especifica el código de la red adquiriente. “SMS-Short Message Service”, “ITR” – Intranet, “CMS” – Card Management System.

TerIdeDisOri, variable de tipo string, se debe especificar la identificación del dispositivo desde donde se origina la transacción.

TerUsuOri, variable de tipo string, se debe especificar la identificación del usuario que origina la transacción.

TerMensaje, variable de tipo string, se especifica el mensaje sms a enviar al usuario. No puede ser mayor a 148 caracteres.

TerInf1 variable de tipo string, campo informativo. Se puede utilizar para guardar datos que identifique el sistema que esta generando la notificación.

TerInf2 variable de tipo string, campo informativo. Se puede utilizar para guardar datos que identifiquen el evento que esta generando la notificación.

Parámetro de salida:

DataOutput; Variable de tipo string. Los 3 primeros caracteres el código de retorno e indica el resultado de la operación.

Retorno(3):

Si es “000”, Transacción OK. A continuación vienen los siguientes campos:

TerMensaje(148): Mensaje enviado al celular destino.

Si es <> “000”, Transacción ERROR, a continuación vienen los siguientes campos.

MensajeError(148) en caso de que este campo se devuelva en blanco, entonces el mensaje de error se toma de plantilla de la pasarela.

4.4.4.50 Método NotificacionOLBatchGenerica().

Descripción: Método que realiza el envío de un mensaje SMS a un celular valido y verifica que la operación utilizada sea de tipo push sin suscripción.

Visibilidad: Público.

Parámetros de entrada:

TerTipPas, variable de tipo string, tipo de medio que se debe utilizar para enviar el mensaje. El dominio de la variable es el siguiente: “E” – Email, “L=PasarelaSMS”, “S=Selecciona la mejor”

TerCodOpeOri: Código de operación autorizada desde donde se origina la transacción.

TerNumCel, variable de tipo string, se especifica el número de celular de

suscripción y es el destino a donde se va a enviar el mensaje sms.

TerNumTar, variable de tipo string, se indica el número de tarjeta de débito/crédito de suscripción.

TerCla, variable de tipo string, se debe especificar la clave asociada a la suscripción banca celular en caso de que la transacción de la operación push con suscripción este configurada con especificación de clave.

TerTransaccion, variable de tipo recordset, se especifican los datos de la transacción asociada a la operación push con suscripción.

TerProBan, variable de tipo string, indica el producto al que pertenece la tarjeta de débito/crédito. Los producto pueden ser "TD" – Tarjeta de débito, "VI" – Tarjeta de crédito VISA, "MC" – Tarjeta de crédito MASTERCARD, "TC" – tarjeta de crédito tarjeta club, "TE" – Tarjeta especial.

TerEsLocal, variable de tipo string, indica si la tarjeta de débito/crédito pertenece a la institución local o a una institución afiliada. El dominio de la variable es "1" – Local, "0" – Afiliada.

TerCodSer, variable de tipo string, código de servicio, para el caso de banca celular "SMS".

TerCodRedAdq, variable de tipo string en donde se especifica el código de la red adquiriente. "SMS-Short Message Service", "ITR" – Intranet, "CMS" – Card Management System.

TerIdeDisOri, variable de tipo string, se debe especificar la identificación del dispositivo desde donde se origina la transacción.

TerUsuOri, variable de tipo string, se debe especificar la identificación del usuario que origina la transacción.

TerMensaje, variable de tipo string, se especifica el mensaje sms a enviar al usuario. No puede ser mayor a 148 caracteres.

TerNomUsuCel, variable de tipo string, se debe especificar el nombre del dueño del celular en caso de tener esta información.

TerInf1 variable de tipo string, campo informativo. Se puede utilizar para guardar datos que identifique el sistema que esta generando la notificación.

TerInf2 variable de tipo string, campo informativo. Se puede utilizar para guardar datos que identifiquen el evento que esta generando la notificación.

Parámetro de salida:

DataOutput; Variable de tipo string. Los 3 primeros caracteres el código de retorno e indica el resultado de la operación.

Retorno(3):

Si es "000", Transacción OK. A continuación vienen los siguientes campos:

TerMensaje(148): Mensaje enviado al celular destino.
Si es <> “000”, Transacción ERROR, a continuación vienen los siguientes campos.
MensajeError(148) en caso de que este campo se devuelva en blanco, entonces el mensaje de error se toma de plantilla de la pasarela.

4.4.4.51 **Método Modifica().**

Descripción: Método que realiza el cambio del teléfono celular de suscripción.
Visibilidad: Público.
Parámetros de entrada:
TerNumCel As String
TerNumTar As String
TerNumCelNue As String
TerClave As String
TerTransaccion As Recordset
TerProBan As String
TerEsLocal As String
TerCodSer As String
TerCodRedAdq As String
TerIdeDisOri As String
TerUsuOri As String
Parámetro de salida:
DataOutput; Variable de tipo string. Los 3 primeros caracteres el código de retorno e indica el resultado de la operación.
Retorno(3):
Si es “000”, Transacción OK. A continuación vienen los siguientes campos:
RespIdeCli(13): Identificación del cliente.
TerProBan(2): Tipo de producto de suscripción. Los valores de dominio son: “TD” – Tarjeta de débito; “VI” – Tarjeta de crédito; “MC”- Tarjeta de crédito; “TC” – Tarjeta club.
TerTarHabAux(30); Nombre del dueño de la tarjeta de débito/crédito.
RespTipTarHab(1): Tipo de tarjetahabiente. Los valores de dominio son: “N” – Norma; “V” - Vip; “P” – Vip Plus; “E” – Empleado.
TerEsLocal(1): Variable que indica si el producto pertenencia a institución local o pertenece a una institución afiliada. Los valores de dominio son: “1” – Local; “0” –

Institución afiliada.
Si es <> “000”, Transacción ERROR, a continuación vienen los siguientes campos.
MensajeError(148) en caso de que este campo se devuelva en blanco, entonces el mensaje de error se toma de plantilla de la pasarela.

4.4.4.52 **Función ConsultaSuscripcionesSE().**

Descripción: Función que consulta las suscripciones banca celular.
Visibilidad: Público.
Parámetros de entrada:
TerNumCel, variable de tipo string, se especifica el número de celular de suscripción.
TerNumTar, variable de tipo string, se indica el número de tarjeta de débito/crédito de suscripción.
TerCodSer, variable de tipo string, código de servicio, para el caso de banca celular “SMS”.
TerFecSusIni, variable de tipo string en donde se especifica la fecha inicial de suscripción en el formato dd/MM/aaaa
TerFecSusFin, variable de tipo string en donde se especifica la fecha final de suscripción en el formato dd/MM/aaaa
Parámetros de salida:
RespIsTranOk, variable de tipo bolean, TRUE indica que la ejecución de las instrucciones dentro del método fueron OK, false indica que se presento algún error dentro del método.
DataOutput, variable de tipo string, en caso de que RespIsTranOk=True, este parámetro viene en blanco, en caso de sea False en este parámetro viene el código y mensaje de error. Tiene la siguiente estructura:
CodRet(3): código del error presentado.
MensajeError(148): Descripción del error presentado.
ConsultaSuscripcionesSE: Variable de tipo recordset en donde devuelve los campos de la base de datos dbServiciosE y la tabla TS-suscripción.

4.4.4.53 **Método ConsultaSusAutSE().**

Descripción: Método que realiza la consulta de suscripciones de los servicios adicionales suscritos por un cliente especificado por su número de tarjeta y número celular.

Visibilidad: Público.

Parámetros de entrada:

TerCodSus, variable de tipo long, se especifica el código de suscripción principal del servicio banca celular.

TerNumCel, variable de tipo string, se especifica el número de celular de suscripción.

TerNumTar, variable de tipo string, se indica el número de tarjeta de débito/crédito de suscripción.

TerCodSer, variable de tipo string, código de servicio, para el caso de banca celular "SMS".

TerCodRedAdq, variable de tipo string, en donde se especifica el código de red adquiriente.

Parámetros de salida:

RespIsTranOk, variable de tipo bolean, TRUE indica que la ejecución de las instrucciones dentro del método fueron OK, false indica que se presento algún error dentro del método.

DataOutput, variable de tipo string, en caso de que RespIsTranOk=True, este parámetro viene en blanco, en caso de sea False en este parámetro viene el código y mensaje de error. Tiene la siguiente estructura:

CodRet(3): código del error presentado.

MensajeError(148): Descripción del error presentado.

ConsultaSusAutSE: Variable de tipo recordset en donde devuelve los campos de la base de datos dbServiciosE y la tabla TSSE-SuscripcionSE

4.4.4.54 Función ConsultaTransaccionesSE().

Descripción: Consulta las transacciones banca celular de un celular.

Visibilidad: Público.

Parámetros de entrada:

TerNumCel, variable de tipo string, se especifica el número de celular de suscripción.

TerNumTar, variable de tipo string, se indica el número de tarjeta de débito/crédito de suscripción.

TerCodSer, variable de tipo string, código de servicio, para el caso de banca celular “SMS”.

TerCodRedAdq, variable de tipo string, en donde se especifica el código de red adquiriente.

TerFecTraIni, variable de tipo string en donde se especifica la fecha inicial de transacciones en el formato dd/MM/aaaa

TerFecTraFin, variable de tipo string en donde se especifica la fecha final de transacciones en el formato dd/MM/aaaa

TerTipCue, variable de tipo entero, se especifica el tipo de cuenta. Los valores del dominio del campo son “00” = Todos; “10” = Ahorros; “20 = Corriente.

TerNumCue, variable de tipo string, se especifica el número de cuenta de suscripción.

TerTipTra, variable de tipo string, se especifica el tipo de transacciones que se desea obtener. El dominio de la variable es el siguiente:

"0" - OK TRANSACCIONES COMPLETAS.

"1" - OK ESPERANDO RESPUESTA OPERADORA.

"2" - TRANSACCIONES ERRONEAS.

"3" - POR REVERSAR.

"4" - REVERSADAS OK.

"5" - SIN RESPUESTA DEL AUTORIZADOR.

"6" - COMISION REVERSADA - SUSCRIPCION SIGUE ACTIVA.

"10" - TODAS LAS TRANSACCIONES.

Parámetros de salida:

RespIsTranOk, variable de tipo bolean, TRUE indica que la ejecución de las instrucciones dentro del método fueron OK, false indica que se presento algún error dentro del método.

DataOutput, variable de tipo string, en caso de que RespIsTranOk=True, este parámetro viene en blanco, en caso de sea False en este parámetro viene el código y mensaje de error. Tiene la siguiente estructura:

CodRet(3): código del error presentado.

MensajeError(148): Descripción del error presentado.

ConsultaTransaccionesSE: Variable de tipo recordset en donde devuelve los campos de la base de datos dbServiciosE y la tabla TSET-SerEleTran

4.4.4.55 **Método ConsultaClaveSE().**

Descripción: Método que consulta la clave de una suscripción banca celular.
Visibilidad: Público.
Parámetros de entrada:
TerNumCel, variable de tipo string, se especifica el número de celular de suscripción y es el destino a donde se va a enviar el mensaje sms.
TerNumTar, variable de tipo string, se indica el número de tarjeta de débito/crédito de suscripción.
TerCla, variable de tipo string, se debe especificar la clave asociada a la suscripción banca celular en caso de que la transacción de la operación push con suscripción sea configurada con especificación de clave.
TerTransaccion, variable de tipo recordset, se especifican los datos de la transacción asociada a la operación push con suscripción.
TerProBan, variable de tipo string, indica el producto al que pertenece la tarjeta de débito/crédito. Los producto pueden ser "TD" – Tarjeta de débito, "VI" – Tarjeta de crédito VISA, "MC" – Tarjeta de crédito MASTERCARD, "TC" – tarjeta de crédito tarjeta club, "TE" – Tarjeta especial.
TerEsLocal, variable de tipo string, indica si la tarjeta de débito/crédito pertenece a la institución local o a una institución afiliada. El dominio de la variable es "1" – Local, "0" – Afiliada.
TerCodSer, variable de tipo string, código de servicio, para el caso de banca celular "SMS".
TerCodRedAdq, variable de tipo string en donde se especifica el código de la red adquiriente. "SMS-Short Message Service", "ITR" – Intranet, "CMS" – Card Management System.
TerIdeDisOri, variable de tipo string, se debe especificar la identificación del dispositivo desde donde se origina la transacción.
TerUsuOri, variable de tipo string, se debe especificar la identificación del usuario que origina la transacción.
TerSupOri, variable de tipo string, se especifica el código del supervisor que autoriza la consulta de la clave actual banca celular.
TerSupNom, variable de tipo string, se especifica el nombre del supervisor que autoriza la consulta de la clave actual banca celular.
Parámetro de salida:

DataOutput; Variable de tipo string. Los 3 primeros caracteres el código de retorno e indica el resultado de la operación.
Retorno(3):
Si es “000”, Transacción OK. A continuación vienen los siguientes campos:
RespIdeCli(13): Identificación del cliente.
TerProBan(2): Tipo de producto de suscripción. Los valores de dominio son: “TD” – Tarjeta de débito; “VI” – Tarjeta de crédito; “MC”- Tarjeta de crédito; “TC” – Tarjeta club.
TerTarHabAux(30); Nombre del dueño de la tarjeta de débito/crédito.
RespTipTarHab(1): Tipo de tarjetahabiente. Los valores de dominio son: “N” – Norma; “V” - Vip; “P” – Vip Plus; “E” – Empleado.
TerEsLocal(1): Variable que indica si el producto pertenencia a institución local o pertenece a una institución afiliada. Los valores de dominio son: “1” – Local; “0” – Institución afiliada.
TerClavel(4): Variable en donde se indica la clave actual del servicio banca celular.
Si es <> “000”, Transacción ERROR, a continuación vienen los siguientes campos.
MensajeError(148) en caso de que este campo se devuelva en blanco, entonces el mensaje de error se toma de plantilla de la pasarela.

4.4.4.56 **Método SetConeccionServiciosE().**

Descripción: Método que establece la comunicación con la base datos banca celular dbServiciosE en la variable oCnServiciosE.
Parámetros de salida:
sMensaje, variable de tipo string, mensaje que especifica el error en caso de no poder establecer una comunicación con la base de datos.
lRetornoServiciosE, variable de tipo long, Devuelve el estado de la comunicación. El dominio del campo es el siguiente. 0 - Comunicación establecida OK; <> 0 – Comunicación no establecida en el campo sMensaje se especifica el error.

4.4.4.57 **Método SetConeccionServiciosE2().**

Descripción: Método que establece la comunicación con la base datos banca celular dbServiciosE en la variable oCnServiciosE2.

Parámetros de salida:

sMensaje, variable de tipo string, mensaje que especifica el error en caso de no poder establecer una comunicación con la base de datos.

lRetornoServiciosE2, variable de tipo long, Devuelve el estado de la comunicación. El dominio del campo es el siguiente. 0 - Comunicación establecida OK; <> 0 – Comunicación no establecida en el campo sMensaje se especifica el error.

4.4.4.58 Método ObtieneTransaccion().

Descripción: Obtiene los datos de la transacción asociada a la operación banca celular solicitada en función del código del servicio, red que origina la solicitud, bin de la tarjeta, bin local, código de operación.

Visibilidad: Público.

Parámetros de entrada:

TerCodSer, variable de tipo string, código de servicio, para el caso de banca celular "SMS".

TerCodRedAdq, variable de tipo string en donde se especifica el código de la red adquiriente. "SMS-Short Message Service", "ITR" – Intranet, "CMS" – Card Management System.

TerBin, variable de tipo string, se indica el número de bin de la tarjeta de débito/crédito de suscripción.

TerEsLocal, variable de tipo string, indica si la tarjeta de débito/crédito pertenece a la institución local o a una institución afiliada. El dominio de la variable es "1" – Local, "0" – Afiliada.

TerCodOpe, variable de tipo string, se especifica el número de celular de suscripción y es el destino a donde se va a enviar el mensaje sms.

Parámetros de salida:

RespIsTranOk, variable de tipo bolean, TRUE indica que la ejecución de las instrucciones dentro del método fueron OK, false indica que se presento algún error dentro del método.

DataOutput, variable de tipo string, en caso de que RespIsTranOk=True, este parámetro viene en blanco, en caso de sea False en este parámetro viene el código y mensaje de error. Tiene la siguiente estructura:

CodRet(3): código del error presentado.

MensajeError(148): Descripción del error presentado.
ConsultaTransaccionesSE: Variable de tipo recordset en donde devuelve los campos de la base de datos dbServiciosE, las tablas TT-transacción] y [TO-Operacion].[TO-TipOpe].

4.4.4.59 Método Class_Initialize()

Descripción: Primer función que se ejecuta al momento de ser invocada por primera vez la clase ServiciosOperacion.Interfaz. Se utiliza para inicializaciones de variables.
Visibilidad: Privada.

4.4.4.60 Método SetConeccionServicios_R().

Descripción: Método que establece la comunicación con la base datos de remesas dbRemesagil en la variable oCnServiciosR.
Parámetros de salida:
sMensaje, variable de tipo string, mensaje que especifica el error en caso de no poder establecer una comunicación con la base de datos.
lRetornoServiciosR, variable de tipo long, Devuelve el estado de la comunicación. El dominio del campo es el siguiente. 0 - Comunicación establecida OK; <> 0 – Comunicación no establecida en el campo sMensaje se especifica el error.

4.4.4.61 Método SetConeccionServicios_SMS().

Descripción: Método que establece la comunicación con la base datos de la pasarela SMS en la variable oCnServiciosSMS.
Parámetros de salida:
sMensaje, variable de tipo string, mensaje que especifica el error en caso de no poder establecer una comunicación con la base de datos.
lRetornoServiciosSMS, variable de tipo long, Devuelve el estado de la comunicación. El dominio del campo es el siguiente. 0 - Comunicación establecida OK; <> 0 – Comunicación no establecida en el campo sMensaje se especifica el

error.

4.4.4.62 Función ConsultaRemesas_SE().

Descripción: Devuelve las remesas del exterior que todavía no se han notificado.
Visibilidad: Público.
Parámetros de salida:
RespIsTranOk, variable de tipo bolean, TRUE indica que la ejecución de las instrucciones dentro del método fueron OK, false indica que se presento algún error dentro del método.
DataOutput, variable de tipo string, en caso de que RespIsTranOk=True, este parámetro viene en blanco, en caso de sea False en este parámetro viene el código y mensaje de error. Tiene la siguiente estructura:
CodRet(3): código del error presentado.
MensajeError(148): Descripción del error presentado.
ConsultaRemesas_SE: Variable de tipo recordset en donde devuelve los campos de la base de datos de remesas, la tabla RemPago.

4.4.4.63 Método ActualizaRemesas_SE().

Descripción: Actualiza las remesas que hayan sido notificadas OK y las que no fueron notificadas para que en el siguiente proceso que se consulte por nuevas remesas solo se incluyan las nuevas remesas.
Visibilidad: Público.
Parámetros de entrada:
Band, variable de tipo boolean, indica si una remesa se notifico correctamente o no. Normalmente una remesa no se notifica cuando el número de teléfono especificado en la tabla de remesas, no corresponde a un número de celular correcto.
codRemesadora, variable de tipo string, indica el código de la empresa remesadora a la que pertenece la notificación.
codRemesa, variable de tipo string, indica el código de la remesa. En conjunto con el parámetro CodRemesadora es un valor que identifica de manera única una remesa.

Valor, variable de tipo string, cuando la notificación es OK, valor es el número de celular. Si la notificación no es OK, valor se graba en blanco para que el campo RePerConLla de la tabla de remesas no quede en NULL y no vuelva entrar este remesa en el siguiente proceso de notificación.

Parámetros de salida:

RespIsTranOk, variable de tipo bolean, TRUE indica que la ejecución de las instrucciones dentro del método fueron OK, false indica que se presento algún error dentro del método.

DataOutput, variable de tipo string, en caso de que RespIsTranOk=True, este parámetro viene “000”, en caso de sea False en este parámetro viene el código y mensaje de error. Tiene la siguiente estructura:

CodRet(3): código del error presentado.

MensajeError(148): Descripción del error presentado.

Parámetros de entrada opcionales:

Los siguientes campos son opciones y se deben especificar cuando la notificación haya sido exitosa.

Fecha, variable de tipo string, de indica fecha del envío del mensaje de notificación.

Hora, variable de tipo string, se indica la hora del envío del mensaje de notificación.

Observación, variable de tipo string, se indica una observación de que la notificación fue enviada exitosamente y el número celular destino.

Estado, variable de tipo string, cambia el estado de la remesa a notificada vía SMS.

Usuario, variable de tipo string, se indica el usuario de notificación.

Error, variable de tipo string, se indica el código de notificación al beneficiario. Este campo es variable y depende de la empresa remesadora. Este código se toma de la tabla CatalogoCompleintsLlamadas de la base de remesas.

4.4.4.64 Método RecuperaError().

Descripción: Recupera el código de error de notificación vía sms en función de la empresa remesadora.

Visibilidad: Público.

Parámetros de entrada:

TerRemCod variable de tipo string, se especifica el código de la empresa remesadora al que pertenece la remesa notificada.

TerRemCodErrBan, variable de tipo string, se especifica el código del banco

utilizado para una notificación.

Parámetros de salida:

RespRemGraStaTra variable de tipo string, devuelve el código de estado de notificación de la remesa,

RespRemCodComLla, variable de tipo string, devuelve el código de error de llamada.

RespRemDesComLla, variable de tipo string, devuelve la descripción de la notificación.

RespIsTranOk, variable de tipo bolean, TRUE indica que la ejecución de las instrucciones dentro del método fueron OK, false indica que se presento algún error dentro del método.

DataOutput, variable de tipo string, en caso de que RespIsTranOk=True, este parámetro viene “000”, en caso de sea False en este parámetro viene el código y mensaje de error. Tiene la siguiente estructura:

CodRet(3): código del error presentado.

MensajeError(148): Descripción del error presentado.

4.4.4.65 Método EnviaSMS().

Descripción: Método que envía un mensaje SMS a la operadora celular a través de la pasarela SMS.

Visibilidad: Privada.

Parámetros de entrada:

TerNumCel, variable de tipo string, se especifica el número celular destino.

TerMensaje, variable de tipo string, se especifica el mensaje a notificar. TerOrigen, variable de tipo string, campo determinado por la pasarela sms en “"PCRE”.

TerTipo, variable de tipo string, campo determinado por la pasarela sms en “R”.

Parámetros de salida:

RespIsTranOk, variable de tipo bolean, TRUE indica que la ejecución de las instrucciones dentro del método fueron OK, false indica que se presento algún error dentro del método.

DataOutput, variable de tipo string, en caso de que RespIsTranOk=True, este parámetro viene “000”, en caso de sea False en este parámetro viene el código y mensaje de error. Tiene la siguiente estructura:

CodRet(3): código del error presentado.

MensajeError(148): Descripción del error presentado.

4.4.4.66 **Método RecuperarCelular().**

Descripción: Recupera el celular de suscripción banca celular en función de la tarjeta de débito/crédito.

Visibilidad: Publico.

Parámetros de entrada:

Tarjeta, variable de tipo string, se especifica el número de tarjeta de débito y crédito de suscripción banca celular.

Parámetros de salida:

RespIsTranOk, variable de tipo bolean, TRUE indica que la ejecución de las instrucciones dentro del método fueron OK, false indica que se presento algún error dentro del método.

DataOutput, variable de tipo string, en caso de que RespIsTranOk=True, este parámetro viene “000”, en caso de sea False en este parámetro viene el código y mensaje de error. Tiene la siguiente estructura:

CodRet(3): código del error presentado.

MensajeError(148): Descripción del error presentado.

4.4.4.67 **Método ConsultarExcepcionOD().**

Descripción: Consulta si existe excepción de notificación entre una red adquiriente y una red autorizadora del evento origina en el switch de la institución bancaria.

Visibilidad: Público.

Parámetros de entrada:

OrigenLink, variable de tipo string, especifica la red adquiriente desde donde se origina la transacción.

DestLink variable de tipo string, especifica la red autorizadora desde donde se origina la transacción.

Parámetros de salida:

RespIsTranOk, variable de tipo bolean, TRUE indica que la ejecución de las instrucciones dentro del método fueron OK, false indica que se presento algún error dentro del método.

RespExcepOD, variable de tipo string, en caso de que RespIsTranOk=True, este

parámetro puede venir en “000” que significa que no se encontró excepción y se debe notificar el evento y si es “001” significa que si existe la excepción y no se debe notificar el evento. En caso de RespIsTranOk sea False hubo un problema de infraestructura al tratar de consultar la excepción y en este parámetro viene el código y mensaje de error. Tiene la siguiente estructura:
CodRet(3): código del error presentado.
MensajeError(148): Descripción del error presentado.

4.4.4.68 Función FechaSistema().

Descripción: Retorna la fecha del sistema en un solo formato.
Visibilidad: Privada.
Parámetros de Salida
FechaSistema, variable de tipo string que retorna la fecha del sistema en el formato “yyyy/MM/dd”

4.4.4.69 Función HoraSistema().

Descripción: Retorna la hora del sistema en un solo formato.
Visibilidad: Privada.
Parámetros de Salida
FechaSistema, variable de tipo string que retorna la hora del sistema en el formato “hhmmss00”

4.4.4.70 Método Class_Terminate().

Descripción: Este método se ejecuta cuando se destruye el objeto que referencia a la clase ServiciosOperacion.Interfaz. Se utiliza para destruir diferentes tipos de objetos como conexiones a base de datos.
Visibilidad: Privado.

4.4.4.71 Función ConsultaTpbPushBatch_SE().

Descripción: Consulta la tabla TPBPushBatch_SE. Devuelve solo las tuplas que tienen un estado no enviando. En la tabla se encuentra las tuplas con los mensajes a enviar a un cliente sobre un evento de sus cuentas o tarjetas en el banco.

Visibilidad: Público.

Parámetros de salida:

RespIsTranOk, variable de tipo bolean, TRUE indica que la ejecución de las instrucciones dentro del método fueron OK, false indica que se presento algún error dentro del método.

DataOutput, variable de tipo string, en caso de que RespIsTranOk=True, este parámetro viene "000", en caso de sea False en este parámetro viene el código y mensaje de error. Tiene la siguiente estructura:

CodRet(3): código del error presentado.

MensajeError(148): Descripción del error presentado.

ConsultaTpbPushBatch_SE(), variable de tipo recordset, los campos devueltos son los de la tabla TPB-PushBatch. Si RespIstranOK y el CodRet del DataOutput es "153", significa que no existen mensajes en la tabla TPB-PushBatch por notificar pero la ejecución del método fue correcta.

4.4.4.72 Método ActualizaTPBPushBatch_SE().

Descripción: Método que actualiza con los estados

Visibilidad: Público.

Parámetros de entrada:

band, variable de tipo string. El dominio del parámetro es el siguiente "E" - PUSH BATCH ENVIADO; "M" - PUSH BATCH ERROR; "P" - PUSH BATCH PENDIENTE DE ENVIO.

codTPB, variable de tipo long. Es el código de la tupla en la tabla TPB-PushBatch del mensaje a notificar en la base de datos dbServiciosE.

Parámetros de salida:

RespIsTranOk, variable de tipo bolean, TRUE indica que la ejecución de las instrucciones dentro del método fueron Ok, false indica que se presento algún error dentro del método.

DataOutput, variable de tipo string, en caso de que RespIsTranOk=True, este

parámetro viene “000”, en caso de sea False en este parámetro viene el código y mensaje de error. Tiene la siguiente estructura:

CodRet(3): código del error presentado.

MensajeError(148): Descripción del error presentado.

Parámetros de entrada opcionales:

Esto parámetros se deben especificar cuando el parámetro band tome los valores de “M” o “P”.

codRet, variable de tipo string, especifica el código del error producido.

Error, variable de tipo string, especifica la descripción del error producido.

4.4.4.73 **Método ConfiguraTransaccionSerElec_PushBatch().**

Descripción: Configura la funcionalidad de una transacción con operación push-batch sin suscripción.

Visibilidad: Público.

Parámetros de entrada:

TerCodTra, variable de tipo string, se especifica el código de la transacción a configurar.

TerDiasSem, variable de tipo string, especifica los días de la semana que se va a notificar los mensajes de esta transacción. “1-Domingo”;“2-Lunes”;”3-Martes”;”4-Miércoles”;”5-Jueves”;”6-Viernes”;”7-Sabado”; Ejemplo si se ubica la cadena “135” significa que van esta transacción tiene permisos de notificación los días Domingo, Martes y Jueves.

TerFecIni, variable de tipo string, indica desde cuando el sistema activara la transacción de mensajes. El formato es “aaaa/MM/dd”.

TerHorIni, variable de tipo string indica la hora de inicio en la que se activara esta transacción. El formato es “hhmmss00”.

TerHorFin, variable de tipo string indica la hora de finalización de activa de ka transacción. El formato es “hhmmss00”.

TerEstado, variable de tipo string, indica si la transacción esta active. “1” – Activa; “0” - Inactiva.

Parámetros de salida:

RespIsTranOk, variable de tipo bolean, TRUE indica que la ejecución de las instrucciones dentro del método fueron OK, false indica que se presento algún error dentro del método.

DataOutput, variable de tipo string, en caso de que RespIsTranOk=True, este

parámetro viene “000”, en caso de sea False en este parámetro viene el código y mensaje de error. Tiene la siguiente estructura:
CodRet(3): código del error presentado.
MensajeError(148): Descripción del error presentado.

4.4.4.74 Método ConsultaTTTransaccion().

Descripción: Consulta la tabla TT-Transacción. Devuelve una tupla de acuerdo al código de transacción consultado. Recupera solo los campos que permiten la configuración de los mensajes.
Visibilidad: Público.
Parámetros de entrada:
TerCodTra, variable de tipo string, se coloca el código de operación a consultar.
Parámetros de salida:
RespIsTranOk, variable de tipo bolean, TRUE indica que la ejecución de las instrucciones dentro del método fueron OK, false indica que se presento algún error dentro del método.
DataOutput, variable de tipo string, en caso de que RespIsTranOk=True, este parámetro viene “000”, en caso de sea False en este parámetro viene el código y mensaje de error. Tiene la siguiente estructura:
CodRet(3): código del error presentado.
MensajeError(148): Descripción del error presentado.
ConsultaTTTransaccion(), variable de tipo recordset devuelve los atributos de la tabla TT-Transaccion

4.5 Diagramas de arquitectura.

Al desarrollar una aplicación, considero de gran valor describir las siguientes vistas dentro de un documento de arquitectura. La notación que más ampliamente se está utilizando es el Lenguaje Unificado de Modelado o UML de sus siglas en inglés.

4.5.1 *Vista lógica.*

Muestra los componentes principales de diseño y sus relaciones de forma independiente de los detalles técnicos y de cómo la funcionalidad será

implementada en la plataforma de ejecución. A continuación se presenta el modelo de diseño de la aplicación, el cual es una vista lógica del modelo funcional y que describen la solución. Se describe la solución en términos de paquetes y clases de diseño. Como estamos utilizando como metodología el proceso unificado, dentro de esta vista se describe la realización de los casos de uso, subsistemas, paquetes y clases de los casos de uso más significativos arquitectónicamente. La Figura 4.3 muestra esta descripción.

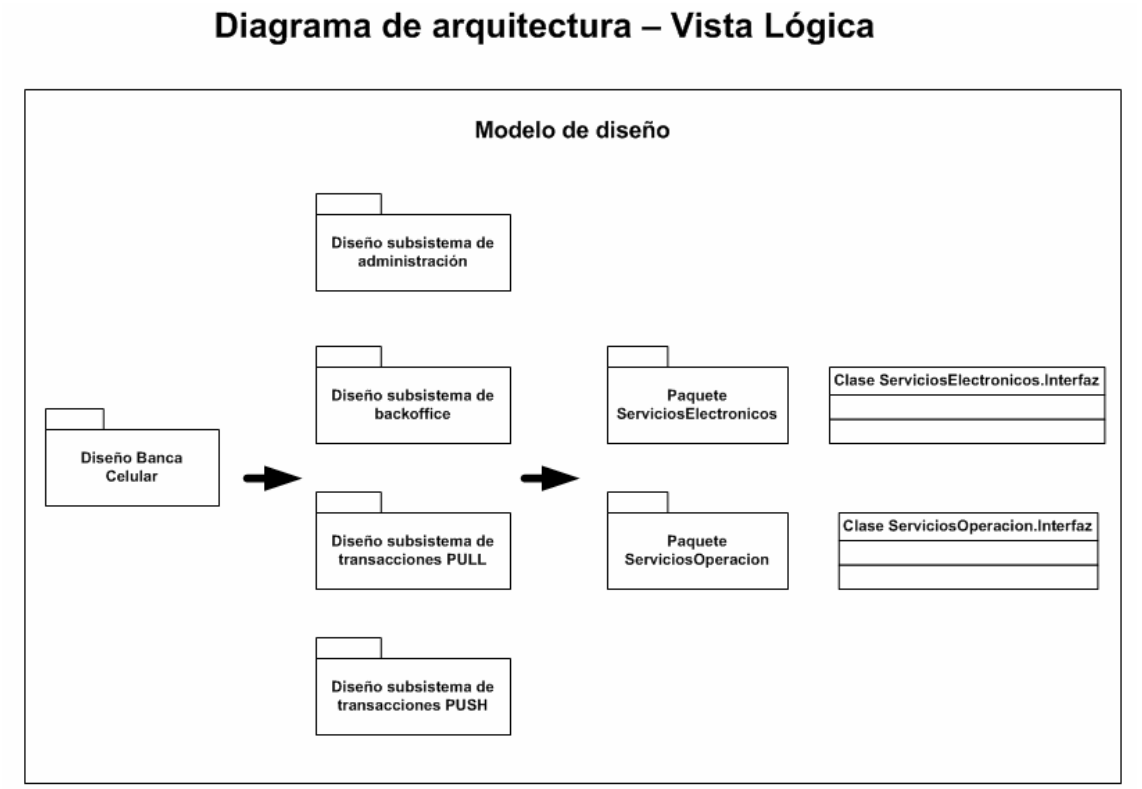


Figura 4.18 Diagrama de arquitectura. Vista de diseño conceptual.

En la actualidad, uno de los patrones de diseño más utilizado para cualquier tipo aplicaciones es el de Capas (Layers en inglés) donde, básicamente, se divide los elementos de diseño en paquetes de Interfaz de Usuario, Lógica de Negocio y Acceso a Datos y Servicios. La figura a continuación muestra una posible partición utilizando este patrón de diseño.

En el siguiente gráfico se pueden ver los paquetes de la capa de Interfaz del Usuario de color amarillo, los paquetes de la capa de lógica del negocio en color azul y finalmente, los paquetes de la capa de acceso a datos de color rojo.

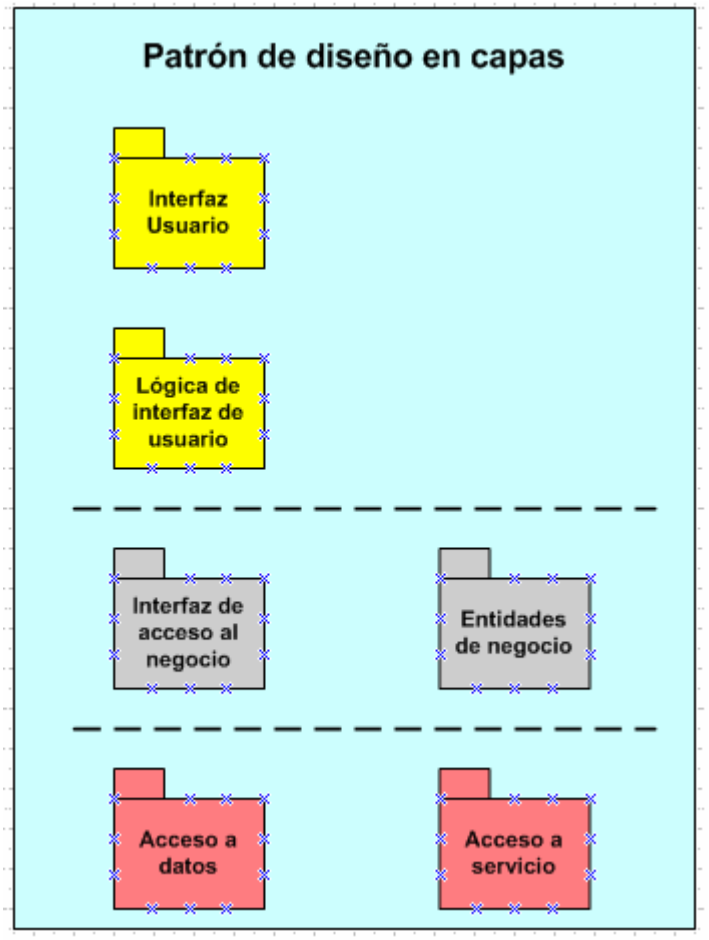


Figura 4.19 Patrón de diseño en capas.

Este es un modelo de diseño, cabe destacar que ésta es una división lógica, no se especifica en que procesador "vivirá" cada uno de los paquetes. Esto se muestra en la vista de implementación.

4.5.2 *Vista física.*

Esta vista ilustra la distribución del procesamiento entre los distintos equipos que conforman la solución, incluyendo los servicios y procesos de base. Los elementos definidos en la vista lógica se "mapean" a componentes de software (servicios, procesos, etc.) o de hardware que definen más precisamente como se ejecutará la solución.

En el gráfico se muestra la solución Web del BackOffice con tres nodos procesadores, Clientes, Servidor Web y Servidor de Base de Datos. Dentro de los nodos se ejecutan procesos, servicios y/o componentes y sus relaciones de dependencia, por ejemplo, el Internet Explorer "muestra" la página HTML que corresponde a la presentación o Interfaz del Usuario de la aplicación.

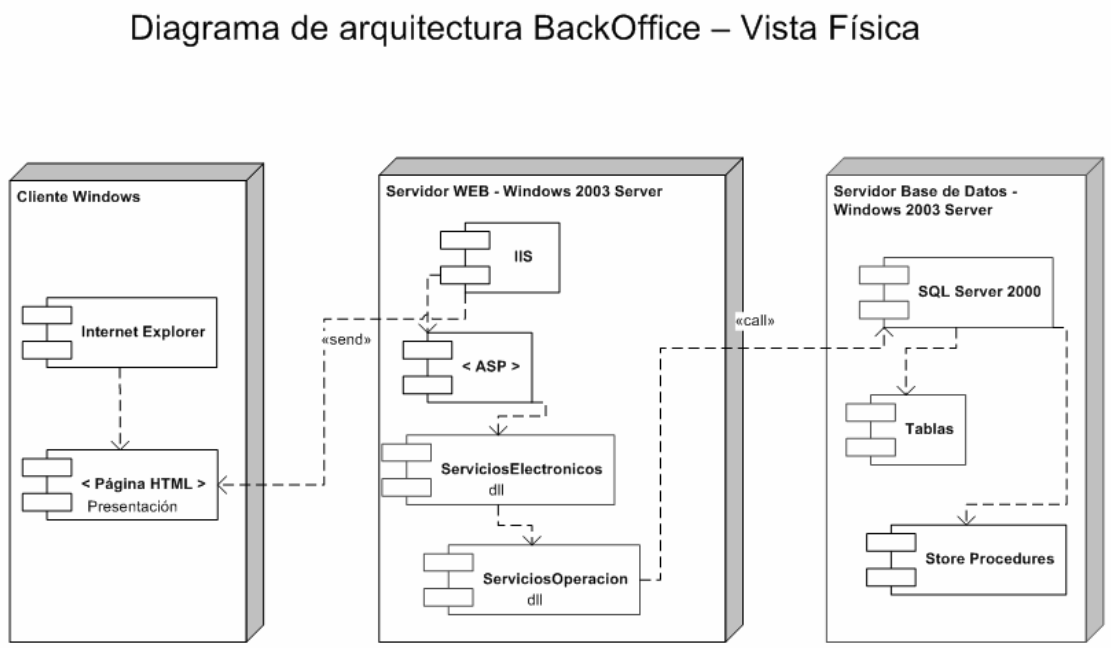


Figura 4.20 Diagrama de arquitectura. Vista de diseño físico del BackOffice.

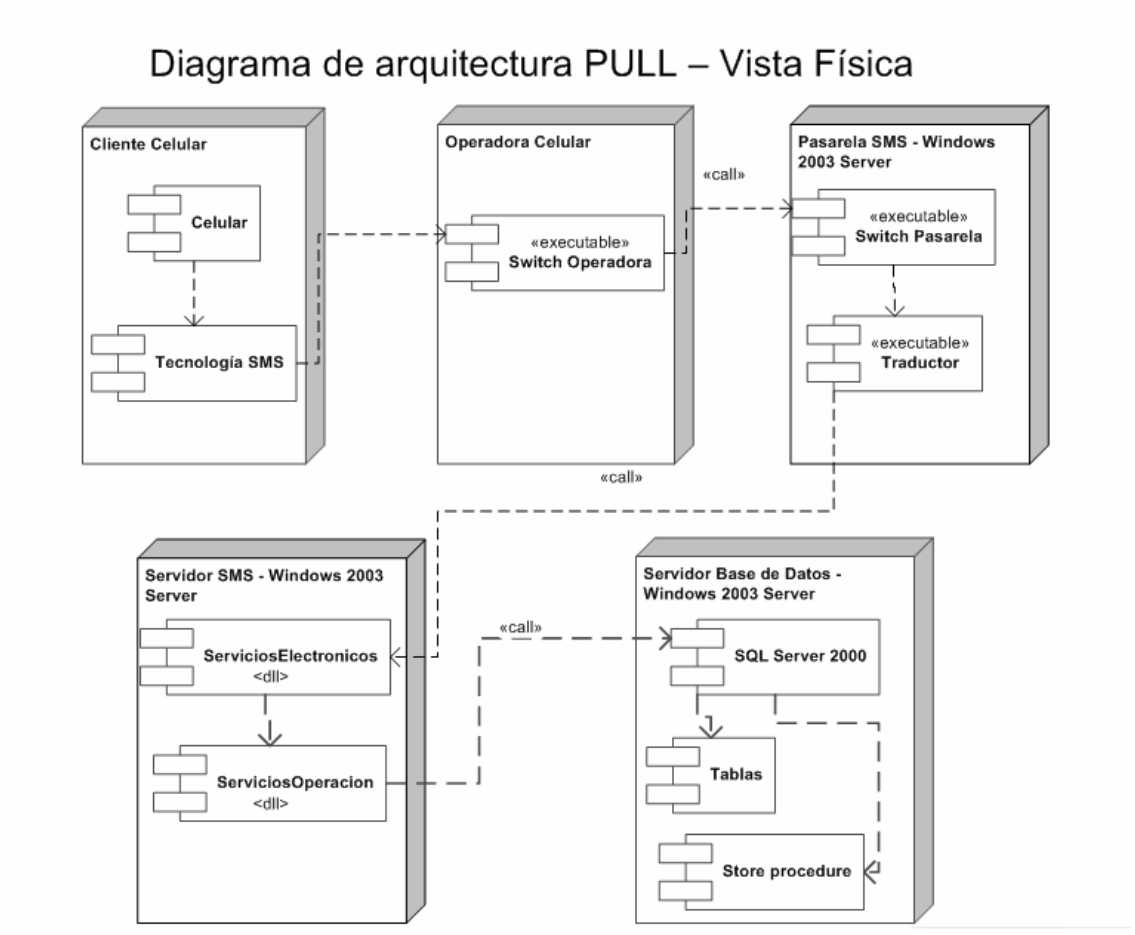


Figura 4.21 Diagrama de arquitectura. Vista de diseño físico de transacciones PULL.

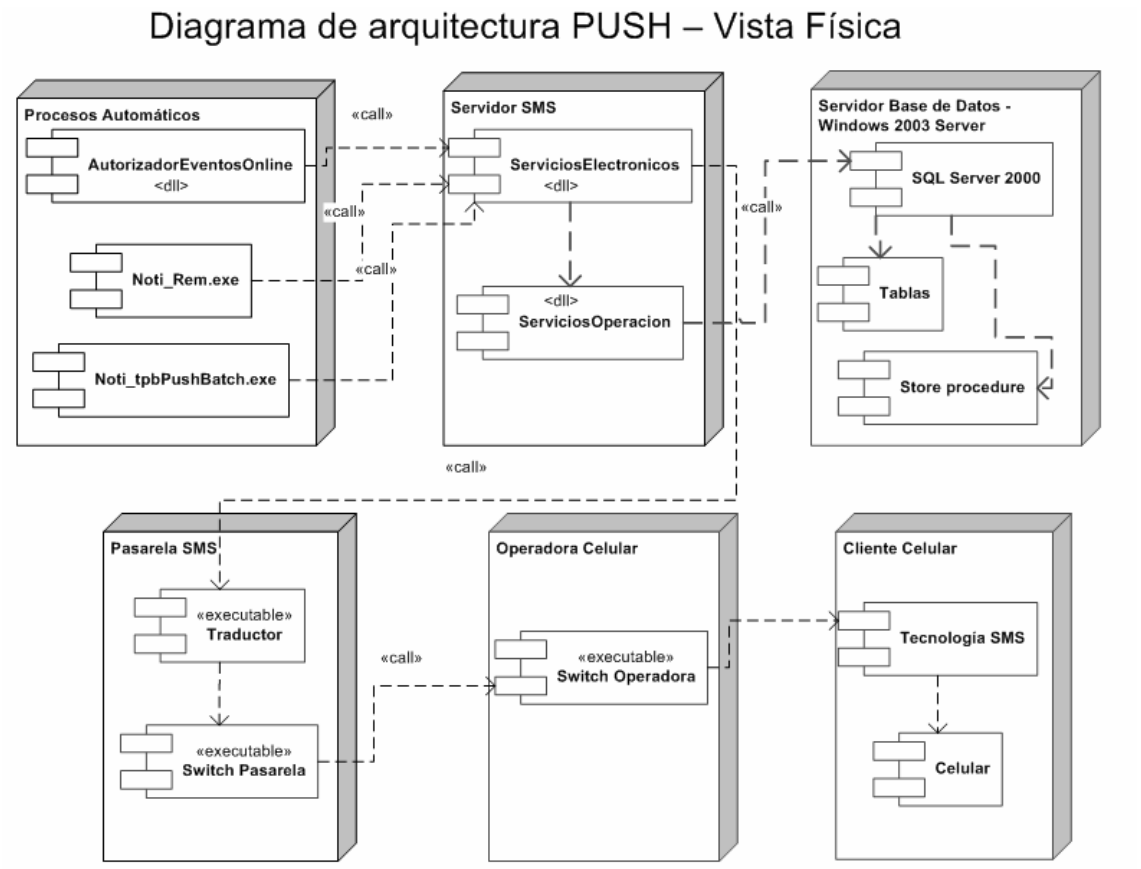


Figura 4.22 Diagrama de arquitectura. Vista de diseño físico de transacciones PUSH.

4.5.3 *Vista de implementación.*

La vista de implementación describe cómo se implementan los componentes físicos mostrados en vista de distribución agrupándolos en subsistemas organizados en capas y jerarquías, ilustra, además las dependencias entre éstos. Básicamente, se describe el mapeo desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos.

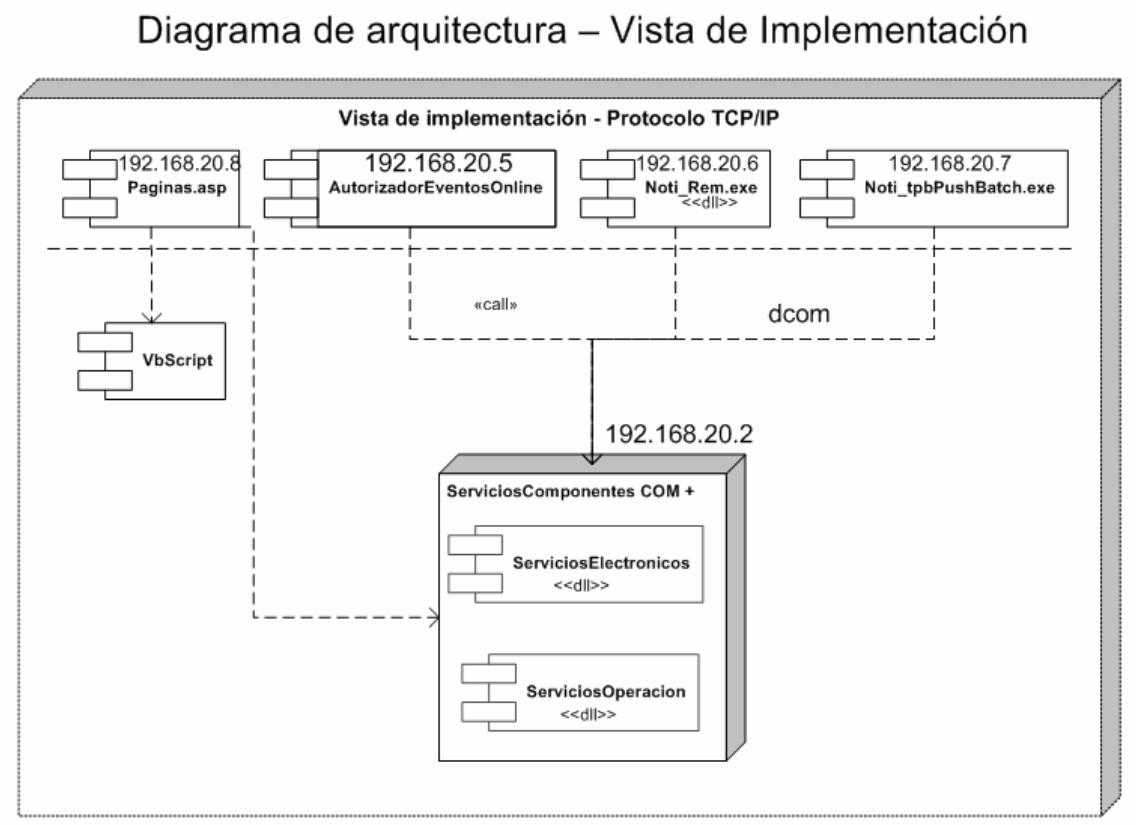


Figura 4.23 Diagrama de arquitectura. Vista de implementación.

4.6 Diagrama de base de datos dbServiciosE

El diseño de una base de datos es un proceso complejo que abarca decisiones a muy distintos niveles. La complejidad se controla mejor si se descompone el problema en subproblemas y se resuelve cada uno de estos subproblemas independientemente, utilizando técnicas específicas. Así, el diseño de una base de datos se descompone en diseño conceptual, diseño lógico y diseño físico.

4.6.1 *Diseño conceptual*

Parte de las especificaciones de requisitos de usuario y su resultado es el esquema conceptual de la base de datos. Un esquema conceptual es una descripción de alto nivel de la estructura de la base de datos, independientemente del SGBD (Sistema Gestor de Base de Datos) que se vaya a utilizar para manipularla. Un modelo conceptual es un lenguaje que se utiliza para describir esquemas conceptuales. El

objetivo del diseño conceptual es describir el contenido de información de la base de datos y no las estructuras de almacenamiento que se necesitarán para manejar esta información.

4.6.2 *Diseño lógico.*

Parte del esquema conceptual y da como resultado un esquema lógico. Un esquema lógico es una descripción de la estructura de la base de datos en términos de las estructuras de datos que puede procesar un tipo de SGBD. Un modelo lógico es un lenguaje usado para especificar esquemas lógicos (modelo relacional, modelo de red, etc.). El diseño lógico depende del tipo de SGBD que se vaya a utilizar, no depende del producto concreto.

Para ver detalles del diseño conceptual de la base de datos dbServiciosE refiérase al anexo 4.4 Diagrama Objeto–Relacion,

4.6.3 *Diseño físico*

Parte del esquema lógico y da como resultado un esquema físico. Un esquema físico es una descripción de la implementación de una base de datos en memoria secundaria: las estructuras de almacenamiento y los métodos utilizados para tener un acceso eficiente a los datos. Por ello, el diseño físico depende del SGBD concreto y el esquema físico se expresa mediante su lenguaje de definición de datos.

A continuación mostramos la implementación de una tabla en el Enterprise Manager del SQL Server 2000.

Design Table 'TS Suscripcion' in 'dbServiciosE' o...

Column Name	Data Type	Length	Allow Nulls
[TS-Cod]	bigint	8	
[TS-FecSus]	smalldatetime	4	
[TS-HorSus]	char	8	
[TS-CodSer]	char	3	
[TRA-CodRedAdq]	char	3	
[TS-IdeDisOri]	varchar	15	✓
[TS-UsuOri]	varchar	10	✓
[TS-NomOpe]	varchar	10	
[TS-NumCel]	char	9	
[TS-SmsPro]	char	2	
[TS-EsLocal]	char	1	
[TS-TipCli]	char	1	
[TS-NomCli]	varchar	30	
[TS-NumTar]	varchar	19	
[TS-IdeCli]	varchar	13	
[TS-Cla]	binary	20	✓
[TS-Act]	char	1	
[TS-Pag]	char	1	
[TS-PagOnLin]	char	1	
[TS-FecUltMod]	smalldatetime	4	✓
[TS-HorUltMod]	char	8	✓
[TS-RedUltMod]	varchar	3	✓
[TS-IdeDisUltMod]	varchar	15	✓
[TS-UsuUltMod]	varchar	10	✓

Columns

Description	
Default Value	
Precision	19
Scale	0
Identity	Yes
Identity Seed	1
Identity Increment	1
Is RowGuid	No
Formula	
Collation	

Figura 4.24 Diseño físico de la tabla TS-Suscripcion

4.6.4 *Diccionario de base de datos*

En esta sección se presenta un documento en donde se presenta la información completa de todas las tablas y relaciones creadas en la base de datos.

La estructura del documento es la siguiente:

TABLA	ATRIBUTO	TIPO	PK	FK	DESCRIPCION	NULL	DOMINI

Tabla 4.2 Estructura de diccionario de datos

En donde

Tabla, es el nombre de la tabla creada.

Atributo, es el nombre del campo que pertenece a la tabla.

Tipo, identifica el tipo de dato al que pertenece el atributo.

PK, indica si el atributo forma parte de la clave principal o no.

FK, indica si el atributo es una clave foránea.

Descripción, breve explicación de la funcionalidad del atributo.

Null, indica si el atributo acepta valores nulos.

Dominio, para ciertos atributos indica los valores que puede tener el atributo para no causar inconsistencias con la lógica del almacenamiento de datos.

Para verificar el diccionario de la base de datos dbServiciosE refiérase al Anexo 4.5.

4.7 Conclusiones

- Los diagramas de colaboración nos permiten decir mas en un espacio que los diagramas de secuencia y expresar además más información contextual; por ejemplo, el tipo de visibilidad entre objetos. También resulta más fácil expresar la lógica condicional y la concurrencia.

- El tema de la arquitectura tomó mucha importancia en los últimos años como algo fundamental al diseñar una solución informática. Lo importante es comprender que para la arquitectura existen varias perspectivas y que dentro de estás podemos mostrar diferentes vistas dependiendo de los puntos de vista de los interesados en la misma y qué, al diseñar una solución debemos pensar en ellos y dejar plasmado en algún documento las decisiones arquitectónicas que tomamos.

- Los diagramas de clase son los más importantes del diseño orientado a objetos, son la piedra angular de nuestro diseño. Contienen toda la información de todas las clases y sus relaciones con otras clases.

- Cuando se construyen los diagramas anteriores como casos de uso, secuencia, colaboración, las herramientas van obteniendo nombres de clases y generando los atributos y operaciones de cada clase siguiendo las indicaciones dadas por las especificaciones de requisitos en los casos de uso y escenarios.

- Al momento de hacer el primer diagrama de clases ya se tiene una lista de clases con algunos de sus atributos y operaciones. Sin embargo es necesario reflexionar y abstraer sobre la organización de esas clases estudiando las relaciones de herencia, agregación, etc.

- El diagrama de clases se ira refinando en las sucesivas iteraciones del modelo.

- Todos los gráficos y diagramas realizados en este capítulo fueron realizados con modelado UML de Microsoft Visio. Este lenguaje tiene la particularidad de crear modelos y no solo diagramas, genera código para C#, Visual Basic.NET y C++, y otras capacidades que ayudan mucho diseñador a la hora de analizar y diseñar una solución.

- Dentro de un arquitectura con división en capas la que se ha utilizado en esta tesis es una de las posibles divisiones en capas, podrían existir otras un poco más completas e incluso una que defina mayor cantidad de capas.

4.8 Recomendaciones

- Al momento de diseñar una solución con metodología RUP se debe generar los artefactos que se consideren los que más sirvan al programador como fuente de documentación para el desarrollo.

- Los diagramas de interacción constituyen uno de los artefactos más importantes que se generan en el diseño orientado a objetos. Para mejorar la calidad de su diseño se recomienda aplicar patrones, principios y expresiones codificadas. A continuación describimos unos breves tips para poder preparar un diagrama de colaboración:

  1. Elabora un diagrama por cada operación del sistema durante el ciclo actual de desarrollo.

  2. Si el diagrama se torna complejo, divídalo en diagramas más pequeños.
  3. Diseñe un sistema de objetos interactivos que realicen las tareas, usando como punto de partida las responsabilidades del contrato de operación, las post condiciones y las descripción de los casos de uso. Aplique el GRASP (Patrones para asignar responsabilidades) y otro patrones para desarrollar un buen diseño.

- Se sugiere que al diseñar una clase se debe pensar en cómo podemos identificar un objeto real, como una persona, un transporte, un documento o un paquete. Durante el proceso del diseño de las clases tomamos las propiedades que identifican como único al objeto y otras propiedades adicionales como datos que corresponden al objeto.

- Una estrategia que da buenos resultados a la hora de construir un diagrama de clases de diseño es la siguiente:

  1. Identifique todas las clases que participan en una solución de software. Para esto analice los diagramas de interación.
  2. Dibújelas en un diagrama de clases.

3. Duplique los atributos provenientes de los conceptos asociados del modelo conceptual.
4. Agregue los nombres de los métodos analizando los diagramas de interacción.
5. Incorpore la información sobre los tipos a los atributos y a los métodos.
6. Agregue las asociaciones necesarias para dar soporte a la visibilidad requerida de los atributos.
7. Agregue las flechas de navegabilidad a las asociaciones para indicar las dirección de visibilidad de los atributos.
8. Agregue las líneas de relaciones de dependencia para indicar la visibilidad no relacionada con los atributos.

CAPITULO 5. IMPLEMENTACION Y PRUEBAS DEL SISTEMA

5.1 Introducción

Una vez concluida la fase de diseño, disponemos de suficiente información para codificar la solución. Los artefactos del UML creados en la fase de diseño como los diagramas de colaboración, diagrama de clases y diagrama de base de datos sirven como información de entrada en el proceso de generación del código.

La creación de código en un lenguaje orientado a objetos no forma parte del análisis ni del diseño orientado a objetos. Sin embargo es recomendable utilizar un lenguaje que soporte el paradigma de la programación orientada a objetos ya que con los artefactos generados en la fase de diseño, suministran suficiente información necesaria para generar el código y el proceso de traducción desde los artefactos al código es un proceso relativamente fácil.

La creación de código lo vamos a realizar para la clase ServiciosElectronicos y ServiciosOperacion en un lenguaje orientado no precisamente a objetos pero si a eventos como lo es Visual Basic 6.0, básicamente por concepto de que este lenguaje ofrece un alto rendimiento y eficiencia en sistemas de alta transaccionalidad, como se espera en este tipo de soluciones. Las clases determinadas para este sistema lo soportan sin ningún problema. El front End para el BackOffice del sistema banca celular se lo desarrolla en ASP por la facilidad de interactuar con objetos COM. La herramienta que proporciona estos paquetes se lo puede encontrar en el paquete de Visual Studio 6.0.

El artefacto que genera la implementación va a ser el sistema banca celular, con esto no queremos decir que el camino sea fácil ni que podamos seguirlo mecánicamente, pues existen muchas variables. En general la fase de programación no es un paso fácil, todo lo contrario, en realidad los resultados obtenidos en la fase de diseño son un primer paso incompleto, en la programación y en las pruebas, se efectuará multitud de cambios y se descubrirán y resolverán problemas detallados. Una de las ventajas del proceso de desarrollo iterativo e incremental, es la posibilidad de introducir los resultados de un ciclo anterior al

iniciar el siguiente. Así los resultados del análisis y diseño subsecuentes se perfeccionan sin cesar y aprovechan el trabajo de la implementación precedente.

5.2 Código fuente.

En esta sección presentaremos el código fuente del método principal de la clase ServiciosElectronicos ProcTrn() y el código de las transacciones PULL, suscripción al servicio banca celular y consulta de saldos de la clase ServiciosOperacion.

Para una lectura completa del código refiérase al Capitulo 4 en donde se presenta una documentación completa de cada una de los métodos y funciones implementados y que se hace referencia en este código fuente.

5.2.1 *Código fuente ServiciosElectronicos.*

Clase: ServiciosElectronicos
Interfaz: Interfaz.

5.2.1.1 Método: Proctrn.

A continuación se presenta el código fuente.

```
'Sección de declaraciones de variables a nivel de clase.

Option Explicit
'Objeto en donde se instancia la clase ServiciosOperacion
Private objServiciosOperacion As Object
'Server SMS
Const Server = "10.1.1.12"
'dir para grabar logs
Private App_Path As String
'Mensaje de error
Private strMsgErr As String
'Nombre del archivo donde graba los logs de errores
Private strFileName As String
'Nombre del programa donde se producen los errores
Private strPrograma As String
'Variable en donde se especifica el mensaje entregado al cliente en caso de error
Private strMenCli As String
```

```
‘Codigo fuente Método Proctrn
Public Sub ProcTrn(ByVal DataInput As String, _
            ByRef DataOutput As String)
Dim TerOpe As String
Dim TerNumCel As String
Dim TerNumTar As String
Dim TerClave As String
Dim TerIdeDisOri As String
Dim TerUsuOri As String
Dim TerSupOri As String
Dim RespBinTar As String
Dim RespEsLocal As String
Dim RespTipTar As String
Dim RespProBan As String
Dim RespIsTranOk As Boolean
Dim RetCodMenError As String * 148
Dim SERTransaccion As Recordset
Dim TerCodOpe As String
Dim TerCodOpeSE As String
Dim TerCodSer As String
Dim TerClaveNue As String
Dim TerTieCla As String
Dim TerTieCos As String
Dim TerTipCue As String * 2
Dim TerTipCup As String * 2
Dim TerNumChe As String * 8
Dim TerClaveSE As String * 4
Dim TerCodRedAdq As String * 3
Dim TerNumCelNue  As String
Dim TerCodOpeOL As String * 5
Dim TerCodOpeBA As String * 5
Dim TerSupNom As String * 150
Dim TerMensaje As String * 145
Dim TerInf1 As String * 255
Dim TerInf2 As String * 255
Dim TerTipPas As String * 1
Dim strMetFunSub As String
Dim strMenErrSis As String
Dim lgnErrNumber As Long
Dim strErrDescription As String
Dim TerCodMot As String * 2
Dim TerValRec As String
Dim TerNumCelRec As String
Dim TerNumClaRec As String
On Error GoTo ERRORES

    TerValRec = ""
    TerNumCelRec = ""
```

```
TerCodMot = ""
lgnErrNumber = -1
strErrDescription = ""
strMenErrSis = ""
strMetFunSub = "PROCTRN"
RetCodMenError = ""
DataOutput = ""
strMsgErr = "DATAINPUT: "
'Graba log DataInput
GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, DataInput
'Obtiene numero celular
TerNumCel = Mid(DataInput, 6, 9)
'Obtiene número de tarjeta
TerNumTar = UCase(Trim(Mid(DataInput, 46, 19)))
'Obtiene la clave
TerClave = Trim(Mid(DataInput, 65, 4))
TerCodOpeOL = ""
TerCodOpeBA = ""

'Crea el objeto ServiciosOperacion
If objServiciosOperacion Is Nothing Then
  On Error Resume Next
  Set objServiciosOperacion = CreateObject("ServiciosOperacion.Interfaz", Server)
  If Err.Number <> 0 Then
    strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
    strMenErrSis = "ERROR: AL CREAR OBJETO (ServicioOperacion.Interfaz)"
    DataOutput = strMenCli
    GrabaErrorExcepc ion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
    Set objServiciosOperacion = Nothing
    RespIsTranOk = False
    Exit Sub
  End If
  On Error GoTo ERRORES
End If

'Valida que el numero de celular tenga únicamente dígitos numéricos
If Not IsNumeric(TerNumCel) Then
  On Error Resume Next
  objServiciosOperacion.MensajeError "O", strMetFunSub, 68, -1, "", DataOutput
  If Err.Number <> 0 Then
    strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description 'Recuperamos el error y la
descripcion de la excepcion
     strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.MensajeError 68" 'Mensaje en donde
se cayo el sistema
    DataOutput = strMenCli 'Mensaje al usuario del sistema
    GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
    Set objServiciosOperacion = Nothing
    RespIsTranOk = False
    Exit Sub
  End If
```

```
    On Error GoTo ERRORES
    Exit Sub
  End If

'Saca el bin y determina si es local o coop
TerCodOpe = UCase(Mid(DataInput, 1, 5)) 'Codigo de Operacion de Transaccion
TerCodSer = Trim(Mid(DataInput, 15, 3))
TerCodRedAdq = Trim(Mid(DataInput, 18, 3))
TerIdeDisOri = Trim(Mid(DataInput, 21, 15))
TerUsuOri = Trim(Mid(DataInput, 36, 10))



'Verifica que el número de tarjeta tenga únicamente dígitos numéricos
If Not IsNumeric(TerNumTar) Then
    On Error Resume Next
    objServiciosOperacion.MensajeError "O", strMetFunSub, 1, -1, "", DataOutput
    If Err.Number <> 0 Then
        strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
        strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.MensajeError 1"
        DataOutput = strMenCli 'Mensaje al usuario del sistema
        GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
        Set objServiciosOperacion = Nothing
        RespIsTranOk = False
        Exit Sub
    End If
    On Error GoTo ERRORES
    Exit Sub
  End If

'Verifica que el número de clave tenga únicamente dígitos numéricos
If Not IsNumeric(TerClave) Then
    On Error Resume Next
    objServiciosOperacion.MensajeError "O", strMetFunSub, 18, -1, "", DataOutput
    If Err.Number <> 0 Then
        strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
        strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.MensajeError 18"
        DataOutput = strMenCli
        GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
        Set objServiciosOperacion = Nothing
        RespIsTranOk = False
        Exit Sub
    End If
    On Error GoTo ERRORES
    Exit Sub
  End If

'Verifica si el bin esta autorizado para acceder al servicio banca celula y obtiene el producto
'al que pertenece y si es local o no
BinTar TerNumTar, RespBinTar, RespEsLocal, RespProBan, RespIsTranOk, DataOutput
```

```
    If Not RespIsTranOk Then
      Exit Sub
    End If

    'En caso de notificación OnLine sin suscripcion y con suscripcion
    If TerCodOpe = "NOTON" Or TerCodOpe = "NOLSU" Then
      TerCodOpeOL = Trim(Mid(DataInput, 69, 5))
      TerMensaje = Trim(Mid(DataInput, 74, 145))
      TerSupNom = Trim(Mid(DataInput, 219, 150))
      TerNumCelNue = Trim(Mid(DataInput, 6, 9))
      TerTipPas = Trim( UCase(Mid(DataInput, 369, 1)))
      Select Case TerTipPas
        Case "L", "E", "S"
        Case Else
            On Error Resume Next
            objServiciosOperacion.MensajeError "O", strMetFunSub, 102, -1, "", DataOutput
            If Err.Number <> 0 Then
              strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
              strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.MensajeError 102"
              DataOutput = strMenCli
              GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
              Set objServiciosOperacion = Nothing
              RespIsTranOk = False
              Exit Sub
            End If
            On Error GoTo ERRORES
            Exit Sub
      End Select

      'Obtiene los campos informativos
      TerInf1 = Trim(Mid(DataInput, 370, 255))
      TerInf2 = Trim(Mid(DataInput, 625, 255))

      'Recupera la transaccion asociada a la oepración solicitada por el cliente
      On Error Resume Next
       Set SERTransaccion = objServiciosOperacion.ObtieneTransaccion(TerCodSer, TerCodRedAdq, RespBinTar,
RespEsLocal, TerCodOpeOL, RespIsTranOk, DataOutput)
      If Err.Number <> 0 Then
        strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
        strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.ObtieneTransaccion"
        DataOutput = strMenCli
        GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
        Set objServiciosOperacion = Nothing
        RespIsTranOk = False
        Exit Sub
      End If
      On Error GoTo ERRORES

      If Not RespIsTranOk Then
        Exit Sub
```

```
End If

Select Case SERTransaccion.Fields("TO-TipOpe")
  Case "1" 'PUSH-ONLINE SIN SUSCRIPCION
     TerNumTar = Trim(SERTransaccion.Fields("TT-NumTar"))
  Case "3" 'PUSH-ONLINE CON SUSCRIPCION

  Case Else
     On Error Resume Next
     objServiciosOperacion.MensajeError "O", strMetFunSub, 99, -1, "", DataOutput
     If Err.Number <> 0 Then
        strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
        strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.MensajeError 99"
        DataOutput = strMenCli
        GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
        Set objServiciosOperacion = Nothing
        RespIsTranOk = False
        Exit Sub
     End If
     On Error GoTo ERRORES
     Exit Sub
End Select


ElseIf TerCodOpe = "NOTBA" Then 'Notificacion Batch
  'Obtiene el codigo de operacion batch
  TerCodOpeBA = Trim(Mid(DataInput, 69, 5))
  'Obtiene el mensaje a enviar
  TerMensaje = Trim(Mid(DataInput, 74, 145))
  'Obtiene el nombre del supervisor
  TerSupNom = Trim(Mid(DataInput, 219, 150))
  'Obtiene el numero celular a donde se desea notificar
  TerNumCelNue = Trim(Mid(DataInput, 6, 9))
  'Obtiene el tipo de pasarela
  TerTipPas = Trim(UCase(Mid(DataInput, 369, 1)))
  Select Case TerTipPas
   Case "L", "E", "S"
   Case Else
      On Error Resume Next
      objServiciosOperacion.MensajeError "O", strMetFunSub, 102, -1, "", DataOutput
      If Err.Number <> 0 Then
         strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
         strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.MensajeError 102"
         DataOutput = strMenCli
         GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
         Set objServiciosOperacion = Nothing
         RespIsTranOk = False
         Exit Sub
      End If
      On Error GoTo ERRORES
```

```
            Exit Sub
        End Select
        'Obtiene los campos informativos
        TerInf1 = Trim(Mid(DataInput, 370, 255))
        TerInf2 = Trim(Mid(DataInput, 625, 255))
        'Obtiene la transaccion asociada al codigo de operación solicitado por el cliente
        On Error Resume Next
         Set SERTransaccion = objServiciosOperacion.ObtieneTransaccion(TerCodSer, TerCodRedAdq, RespBinTar,
RespEsLocal, TerCodOpeBA, RespIsTranOk, DataOutput)
        If Err.Number <> 0 Then
           strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
           strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.ObtieneTransaccion"
           DataOutput = strMenCli
           GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
           Set objServiciosOperacion = Nothing
           RespIsTranOk = False
           Exit Sub
        End If
        On Error GoTo ERRORES
        If Not RespIsTranOk Then
           Exit Sub
        End If
        'PUSH-BATCH SIN SUSCRIPCION
        If SERTransaccion.Fields("TO-TipOpe") <> "2" Then
           On Error Resume Next
           objServiciosOperacion.MensajeError "O", strMetFunSub, 100, -1, "", DataOutput
           If Err.Number <> 0 Then
              strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
              strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.MensajeError 100"
              DataOutput = strMenCli
              GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
              Set objServiciosOperacion = Nothing
              RespIsTranOk = False
              Exit Sub
           End If
           On Error GoTo ERRORES
           Exit Sub
        End If
        'Obtiene el nuemro de tarjeta asociada a la transaccion
        TerNumTar = Trim(UCase(SERTransaccion.Fields("TT-NumTar")))
      Else
        On Error Resume Next
         Set SERTransaccion = objServiciosOperacion.ObtieneTransaccion(TerCodSer, TerCodRedAdq, RespBinTar,
RespEsLocal, TerCodOpe, RespIsTranOk, DataOutput)
        If Err.Number <> 0 Then
           strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
           strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.ObtieneTransaccion"
           DataOutput = strMenCli
           GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
           Set objServiciosOperacion = Nothing
```

```
        RespIsTranOk = False
        Exit Sub
      End If
      On Error GoTo ERRORES

    End If

    If Not RespIsTranOk Then
      Exit Sub
    End If
    'Obtiene el parámetro que indica si la transacciób requiere clave
    TerTieCla = SERTransaccion.Fields.Item(7).Value

    Select Case TerCodOpe
        Case "SUSCR" 'Operacion de Suscripcion
                On Error Resume Next
                 objServiciosOperacion.Suscribe TerNumCel, TerNumTar, TerClave, SERTransaccion, RespProBan,
RespEsLocal, TerCodSer, TerCodRedAdq, TerIdeDisOri, TerUsuOri, DataOutput
                If Err.Number <> 0 Then
                  strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
                  strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.Suscribe"
                  DataOutput = strMenCli
                  GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
                  Set objServiciosOperacion = Nothing
                  RespIsTranOk = False
                  Exit Sub
                End If
                On Error GoTo ERRORES


        Case "PASSW" 'Operacion de Cambio de Password
                TerClaveNue = Trim(Mid(DataInput, 69, 4))
                If Not IsNumeric(TerClaveNue) Then
                  On Error Resume Next
                  objServiciosOperacion.MensajeError "O", strMetFunSub, 31, -1, "", DataOutput
                  If Err.Number <> 0 Then
                    strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
                    strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.MensajeError 31"
                    DataOutput = strMenCli
                    GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
                    Set objServiciosOperacion = Nothing
                    RespIsTranOk = False
                    Exit Sub
                  End If
                  On Error GoTo ERRORES
                  Exit Sub
                End If
                On Error Resume Next
                objServiciosOperacion.Password TerNumCel, TerNumTar, TerClave, TerClaveNue,
SERTransaccion, RespProBan, RespEsLocal, TerCodSer, TerCodRedAdq, TerIdeDisOri, TerUsuOri, DataOutput
```

```
            If Err.Number <> 0 Then
              strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
              strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.Password"
              DataOutput = strMenCli
              GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
              Set objServiciosOperacion = Nothing
              RespIsTranOk = False
              Exit Sub
            End If
            On Error GoTo ERRORES

        Case "MOSUS" 'Operacion de modificacion de suscripcion
            TerNumCelNue = Trim(Mid(DataInput, 69, 9))
            If Not IsNumeric(TerNumCelNue) Then
              On Error Resume Next
              objServiciosOperacion.MensajeError "O", strMetFunSub, 68, -1, "", DataOutput
              If Err.Number <> 0 Then
                strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
                strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.MensajeError 68"
                DataOutput = strMenCli
                GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
                Set objServiciosOperacion = Nothing
                RespIsTranOk = False
                Exit Sub
              End If
              On Error GoTo ERRORES
              Exit Sub
            End If
            On Error Resume Next
            objServiciosOperacion.Modifica TerNumCel, TerNumTar, TerNumCelNue, TerClave,
SERTransaccion, RespProBan, RespEsLocal, TerCodSer, TerCodRedAdq, TerIdeDisOri, TerUsuOri, DataOutput
            If Err.Number <> 0 Then
              strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
              strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.Modifica"
              DataOutput = strMenCli
              GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
              Set objServiciosOperacion = Nothing
              RespIsTranOk = False
              Exit Sub
            End If
            On Error GoTo ERRORES

        Case "SUSSA" 'Suscripcion de Servicios Autorizados
            TerCodOpeSE = Trim(Mid(DataInput, 69, 5))
            On Error Resume Next
            objServiciosOperacion.SuscribeAutorizada TerNumCel, TerNumTar, TerCodOpeSE, TerClave,
SERTransaccion, RespProBan, RespEsLocal, TerCodSer, TerCodRedAdq, TerIdeDisOri, TerUsuOri, DataOutput
            If Err.Number <> 0 Then
              strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
              strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.SuscribeAutorizada"
```

```
            DataOutput = strMenCli
            GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
            Set objServiciosOperacion = Nothing
            RespIsTranOk = False
            Exit Sub
          End If
          On Error GoTo ERRORES
    Case "SALDO" 'Consulta de Saldos de CC y AH
          TerTipCue = UCase(Mid(DataInput, 69, 2))
          On Error Resume Next
          objServiciosOperacion.SaldoTarDeb TerNumCel, TerNumTar, TerTipCue, TerClave,
SERTransaccion, RespProBan, RespEsLocal, TerCodSer, TerCodRedAdq, TerIdeDisOri, TerUsuOri, DataOutput
          If Err.Number <> 0 Then
            strMsgErr = "ERROR N°: " & Er r.Number & " DESCRIPCION: " & Err.Description
            strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.SaldoTarDeb"
            DataOutput = strMenCli
            GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
            Set objServiciosOperacion = Nothing
            RespIsTranOk = False
            Exit Sub
          End If
          On Er ror GoTo ERRORES

    Case "SALTC"
           'Consulta de Saldos de TC
          On Error Resume Next
          objServiciosOperacion.SaldoTarCre TerNumCel, TerNumTar, TerClave, SERTransaccion,
RespProBan, RespEsLocal, TerCodSer, TerCodRedAdq, TerIdeDisOri, TerUsuOri, DataOutput
          If Err.Number <> 0 Then
            strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
            strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.SaldoTarCre"
            DataOutput = strMenCli
            GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
            Set objServiciosOperacion = Nothing
            RespIsTranOk = False
            Exit Sub
          End If
          On Error GoTo ERRORES


    Case "CUPTC" 'Consulta de Cupos de TC
          TerTipCup = Mid(DataInput, 69, 2)
          On Error Resume Next
          objServiciosOperacion.CupoTarCre TerNumCel, TerNumTar, TerTipCup, TerClave,
SERTransaccion, RespProBan, RespEsLocal, TerCodSer, TerCodRedAdq, TerIdeDisOri, TerUsuOri, DataOutput
          If Err.Number <> 0 Then
            strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
            strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.CupoTarCre"
            DataOutput = strMenCli
            GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
```

```
            Set objServiciosOperacion = Nothing
            RespIsTranOk = False
            Exit Sub
          End If
          On Error GoTo ERRORES


    Case "CONCH" 'Consulta de status de cheques
          TerNumChe = Mid(DataInput, 69, 8)
          If Not IsNumeric(TerNumChe) Then
            On Error Resume Next
            objServiciosOperacion.MensajeError "O", strMetFunSub, 36, -1, "", DataOutput
            If Err.Number <> 0 Then
              strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
              strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.MensajeError 36"
              DataOutput = strMenCli
              GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
              Set objServiciosOperacion = Nothing
              RespIsTranOk = False
              Exit Sub
            End If
            On Error GoTo ERRORES
            Exit Sub
          End If
          On Error Resume Next
          objServiciosOperacion.ConsultaCheque TerNumCel, TerNumTar, TerNumChe, TerClave,
SERTransaccion, RespProBan, RespEsLocal, TerCodSer, TerCodRedAdq, TerIdeDisOri, TerUsuOri, DataOutput
          If Err.Number <> 0 Then
            strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
            strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.ConsultaCheque"
            DataOutput = strMenCli
            GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
            Set objServiciosOperacion = Nothing
            RespIsTranOk = False
            Exit Sub
          End If
          On Error GoTo ERRORES


    Case "BLTAR" 'Bloqueo de Tarjeta
          TerClaveSE = Mid(DataInput, 69, 8)
          If Not IsNumeric(TerClaveSE) Then
            On Error Resume Next
            objServiciosOperacion.MensajeError "O", strMetFunSub, 31, -1, "", DataOutput
            If Err.Number <> 0 Then
              strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
              strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.MensajeError 31"
              DataOutput = strMenCli
              GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
              Set objServiciosOperacion = Nothing
```

```
            RespIsTranOk = False
            Exit Sub
          End If
          On Error GoTo ERRORES
          Exit Sub
        End If

        On Error Resume Next
        objServiciosOperacion.BloqueoTarjeta TerNumCel, TerNumTar, TerClaveSE, TerClave,
SERTransaccion, RespProBan, RespEsLocal, TerCodSer, TerCodRedAdq, TerIdeDisOri, TerUsuOri, DataOutput
        If Err.Number <> 0 Then
          strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
          strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.BloqueoTarjeta"
          DataOutput = strMenCli
          GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
          Set objServiciosOperacion = Nothing
          RespIsTranOk = False
          Exit Sub
        End If
        On Error GoTo ERRORES



    Case "INSUS" 'Operacion de in activación de Suscripcion
        'TerIdeDisOri = Trim(Mid(DataInput, 21, 15))
        'TerUsuOri = Trim(Mid(DataInput, 36, 10))
        TerCodMot = Trim(Mid(DataInput, 69, 2))
        On Error Resume Next
        objServiciosOperacion.InactivaSuscribe TerNumCel, TerNumTar, TerClave, SERTransaccion,
RespProBan, RespEsLocal, TerCodSer, TerCodRedAdq, TerIdeDisOri, TerUsuOri, TerCodMot, DataOutput
        If Err.Number <> 0 Then
          strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
          strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.InactivaSuscribe"
          DataOutput = strMenCli
          GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
          Set objServiciosOperacion = Nothing
          RespIsTranOk = False
          Exit Sub
        End If
        On Error GoTo ERRORES

    Case "INSUA" 'Operacion de in activación de Suscripcion de Servicios Autorizados
        TerCodOpeSE = Trim(Mid(DataInput, 69, 5))
        TerCodMot = Trim(Mid(DataInput, 74, 2))
        On Error Resume Next
        objServiciosOperacion.InactivaSuscribeAutorizada TerNumCel, TerNumTar, TerCodOpeSE,
TerClave, SERTransaccion, RespProBan, RespEsLocal, TerCodSer, TerCodRedAdq, TerIdeDisOri, TerUsuOri,
TerCodMot, DataOutput
        If Err.Number <> 0 Then
          strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
```

```
            strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.InactivaSuscribeAutorizada"
            DataOutput = strMenCli
            GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
            Set objServiciosOperacion = Nothing
            RespIsTranOk = False
            Exit Sub
          End If
          On Error GoTo ERRORES

    Case "COCLA"
          'Consulta de clave de servicios electrónicos
          TerSupOri = Trim(Mid(DataInput, 69, 10))
          TerSupNom = Trim(Mid(DataInput, 79, 40))
          On Error Resume Next
          objServiciosOperacion.ConsultaClaveSE TerNumCel, TerNumTar, TerClave, SERTransaccion,
RespProBan, RespEsLocal, TerCodSer, TerCodRedAdq, TerIdeDisOri, TerUsuOri, TerSupOri, TerSupNom,
DataOutput
          If Err.Number <> 0 Then
            strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
            strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.ConsultaClaveSE"
            DataOutput = strMenCli
            GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
            Set objServiciosOperacion = Nothing
            RespIsTranOk = False
            Exit Sub
          End If
          On Error GoTo ERRORES


    Case "CONSU"
          'Consulta de suscripcion
          On Error Resume Next
          objServiciosOperacion.ConsultaSuscripcionSE TerNumCel, TerNumTar, TerClave, SERTransaccion,
RespProBan, RespEsLocal, TerCodSer, TerCodRedAdq, TerIdeDisOri, TerUsuOri, DataOutput
          If Err.Number <> 0 Then
            strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
            strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.ConsultaSuscripcionSE"
            DataOutput = strMenCli
            GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
            Set objServiciosOperacion = Nothing
            RespIsTranOk = False
            Exit Sub
          End If
          On Error GoTo ERRORES



    Case "RVSUS"
          'Reversa la sucripcion y sucripcion autorizadas si las tiene
          TerSupOri = Trim(Mid(DataInput, 69, 10))
```

```
TerSupNom = Trim(Mid(DataInput, 79, 40))
On Error Resume Next
objServiciosOperacion.ReversaSuscribe TerNumCel, TerNumTar, TerClave, SERTransaccion, RespProBan, RespEsLocal, TerCodSer, TerCodRedAdq, TerIdeDisOri, TerUsuOri, TerSupOri, TerSupNom, DataOutput
If Err.Number <> 0 Then
  strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
  strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.ReversaSuscribe"
  DataOutput = strMenCli
  GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
  Set objServiciosOperacion = Nothing
  RespIsTranOk = False
  Exit Sub
End If
On Error GoTo ERRORES


Case "RVSUA"
'Reversa la sucripcion autorizada
TerCodOpeSE = Trim(Mid(DataInput, 69, 5))
TerSupOri = Trim(Mid(DataInput, 74, 10))
TerSupNom = Trim(Mid(DataInput, 84, 40))
On Error Resume Next
objServiciosOperacion.ReversaSuscribeAutorizada TerNumCel, TerNumTar, TerCodOpeSE, TerClave, SERTransaccion, RespProBan, RespEsLocal, TerCodSer, TerCodRedAdq, TerIdeDisOri, TerUsuOri, TerSupOri, TerSupNom, DataOutput
If Err.Number <> 0 Then
  strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
  strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.ReversaSuscribeAutorizada"
  DataOutput = strMenCli
  GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
  Set objServiciosOperacion = Nothing
  RespIsTranOk = False
  Exit Sub
End If
On Error GoTo ERRORES


Case "NOTON"
'notificación de avisos online sin suscripcion
On Error Resume Next
objServiciosOperacion.NotificacionOLBatchGenerica TerTipPas, TerCodOpe, TerNumCel, TerNumTar, TerClave, SERTransaccion, RespProBan, RespEsLocal, TerCodSer, TerCodRedAdq, TerIdeDisOri, TerUsuOri, TerMensaje, TerSupNom, TerInf1, TerInf2, DataOutput
If Err.Number <> 0 Then
  strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
  strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.NotificacionOLBatchGenerica"
  DataOutput = strMenCli
  GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
```

```
                Set objServiciosOperacion = Nothing
                RespIsTranOk = False
                Exit Sub
            End If
            On Error GoTo ERRORES



        Case "NOTBA"
            'notificación de avisos en batch sin suscripcion
            On Error Resume Next
            objServiciosOperacion.NotificacionOLBatchGenerica TerTipPas, TerCodOpe, TerNumCel,
TerNumTar, TerClave, SERTransaccion, RespProBan, RespEsLocal, TerCodSer, TerCodRedAdq, TerIdeDisOri,
TerUsuOri, TerMensaje, TerSupNom, TerInf1, TerInf2, DataOutput
            If Err.Number <> 0 Then
                strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
                strMenErrSis = "ERROR: En ProcTrn al llamar
objServiciosOperacion.NotificacionOLBatchGenerica"
                DataOutput = strMenCli
                GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
                Set objServiciosOperacion = Nothing
                RespIsTranOk = False
                Exit Sub
            End If
            On Error GoTo ERRORES


        Case "NOLSU"
            'notificación de remesas online con suscripcion
            On Error Resume Next
            objServiciosOperacion.NotificacionOLBatchGenericaConSuscripcion TerTipPas, TerCodOpe,
TerNumCel, TerNumTar, TerClave, SERTransaccion, RespProBan, RespEsLocal, TerCodSer, TerCodRedAdq,
TerIdeDisOri, TerUsuOri, TerMensaje, TerSupNom, TerInf1, TerInf2, DataOutput
            If Err.Number <> 0 Then
                strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
                strMenErrSis = "ERROR: En ProcTrn al llamar
objServiciosOperacion.NotificacionOLBatchGenericaConSuscripcion"
                DataOutput = strMenCli
                GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
                Set objServiciosOperacion = Nothing
                RespIsTranOk = False
                Exit Sub
            End If
            On Error GoTo ERRORES

        Case "AUTOR", "PAUTO", "REVAU", "REVPA", "DETCC", "DETLO", "TOTCC" 'Operacion de solicitud de
autorización para una compra comercial card
            On Error Resume Next
                objServiciosOperacion.PULLSinSuscripcion TerNumCel, TerNumTar, TerClave, SERTransaccion,
RespProBan, RespEsLocal, TerCodSer, TerCodRedAdq, TerIdeDisOri, TerUsuOri, DataInput, DataOutput
```

```
                If Err.Number <> 0 Then
                  strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
                  strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.PULLSinSuscripcion"
                  DataOutput = strMenCli
                  GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
                  Set objServiciosOperacion = Nothing
                  RespIsTranOk = False
                  Exit Sub
                End If
                On Error GoTo ERRORES

        Case "RECAR" 'Recarga automatica de tiempo aire
                'TerTipCue = UCase(Mid(DataInput, 69, 2))

                TerValRec = Mid(DataInput, 69, 12)
                TerNumCelRec = Mid(DataInput, 81, 9)
                TerNumClaRec = Mid(DataInput, 90, 4)
                On Error Resume Next
                objServiciosOperacion.RecargaCelular TerNumCel, TerNumTar, TerTipCue, TerNumClaRec,
SERTransaccion, RespProBan, RespEsLocal, TerCodSer, TerCodRedAdq, TerIdeDisOri, TerUsuOri, TerValRec,
TerNumCelRec, DataOutput
                If Err.Number <> 0 Then
                  strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
                  strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.RecargaCelular"
                  DataOutput = strMenCli
                  GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
                  Set objServiciosOperacion = Nothing
                  RespIsTranOk = False
                  Exit Sub
                End If
                On Error GoTo ERRORES


        Case Else
                DataOutput = "999OPERACION DESCONOCIDA"
    End Select
Exit Sub
ERRORES:

   'En caso de presetarse un error en la ejecucion de las instrucciones dentro de este método
   'Salta a esta sección la ejecución, en donde se graba en un Log la excepcion que genero el
   'error
   strMenErrSis = "ERROR: En ProcTrn N°: " & Err.Number & " DESCRIPCION: " & Err.Description
   lgnErrNumber = Err.Number
   strErrDescription = Err.Description
   On Error Resume Next
   objServiciosOperacion.MensajeError "S", strMetFunSub, 108, lgnErrNumber, strErrDescription, DataOutput
   If Err.Number <> 0 Then
     strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
     DataOutput = strMenCli
```

```
      GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
      Set objServiciosOperacion = Nothing
      RespIsTranOk = False
      Exit Sub
    End If
    On Error GoTo ERRORES
    RespIsTranOk = False
End Sub
```

5.2.1.2 Función SuscribeSE.

Esta función es invocada por la pagina SuscripcionSMS.asp para realizar la suscripción banca celular desde la intranet por un ejecutivo de atención al cliente.

```
Public Function SuscripcionSE(ByVal TerNumCel, _
                    ByVal TerCodSer, _
                    ByVal TerCodRedAdq, _
                    ByVal TerIdeDisOri, _
                    ByVal TerUsuOri, _
                    ByVal TerNumTar, _
                    ByVal TerPin) As String

On Error GoTo ERRORES

Dim DataInput As String
Dim TerIdeDisOriF  As String * 15
Dim TerUsuOriF As String * 10
Dim TerNumTarF As String * 19
Dim strMenErrSis As String
Dim lgnErrNumber As Long
Dim strErrDescription As String
Dim strMetFunSub As String


  'Verifica que el numero de celular sea numerico y tenga longitud de 9
  strMenErrSis = ""
  lgnErrNumber = -1
  strErrDescription = ""
  strMetFunSub = "SuscripcionSE"

  If objServiciosOperacion Is Nothing Then
    On Error Resume Next
    Set objServiciosOperacion = CreateObject("ServiciosOperacion.Interfaz", Server)
    If Err.Number <> 0 Then
      strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
      strMenErrSis = "ERROR: AL CREAR OBJETO (ServicioOperacion.Interfaz)"
      SuscripcionSE = strMenCli
      GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
```

```
        Set objServiciosOperacion = Nothing
        Exit Function
      End If
      On Error GoTo ERRORES
   End If


   TerNumCel = Trim(TerNumCel)
   If Not IsNumeric(TerNumCel) Or Len(TerNumCel) <> 9 Then
      On Error Resume Next
      objServiciosOperacion.MensajeError "O", strMetFunSub, 45, -1, "", SuscripcionSE
      If Err.Number <> 0 Then
         strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description 'Recuperamos el error y la
descripcion de la excepcion
        strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.MensajeError 45" 'Mensaje en donde
se cayo el sistema
        SuscripcionSE = strMenCli 'Mensaje al usuario del sistema
        GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
        Set objServiciosOperacion = Nothing
        Exit Function
      End If
      On Error GoTo ERRORES
      Exit Function
   End If

   TerCodSer = Trim(UCase(TerCodSer))
   If Len(TerCodSer) <> 3 Then
      On Error Resume Next
      objServiciosOperacion.MensajeError "O", strMetFunSub, 46, -1, "", SuscripcionSE
      If Err.Number <> 0 Then
         strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description 'Recuperamos el error y la
descripcion de la excepcion
         strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.MensajeError 46" 'Mensaje en donde
se cayo el sistema
        SuscripcionSE = strMenCli 'Mensaje al usuario del sistema
        GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
        Set objServiciosOperacion = Nothing
        Exit Function
      End If
      On Error GoTo ERRORES
      Exit Function
   End If

   TerCodRedAdq = Trim(UCase(TerCodRedAdq))
   If Len(TerCodRedAdq) <> 3 Then
      On Error Resume Next
      objServiciosOperacion.MensajeError "O", strMetFunSub, 47, -1, "", SuscripcionSE
      If Err.Number <> 0 Then
         strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description 'Recuperamos el error y la
descripcion de la excepcion
```

```
        strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.MensajeError 47" 'Mensaje en donde
se cayo el sistema
        SuscripcionSE = strMenCli 'Mensaje al usuario del sistema
        GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
        Set objServiciosOperacion = Nothing
        Exit Function
      End If
      On Error GoTo ERRORES
      Exit Function
    End If


    TerIdeDisOri = Trim(UCase(TerIdeDisOri))
    If TerIdeDisOri <> "" Then
      If Len(TerIdeDisOri) > 15 Then
        TerIdeDisOri = Mid(TerIdeDisOri, 1, 15)
      End If
    ElseIf TerIdeDisOri = "" Then
      On Error Resume Next
      objServiciosOperacion.MensajeError "O", strMetFunSub, 49, -1, "", SuscripcionSE
      If Err.Number <> 0 Then
        strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description 'Recuperamos el error y la
descripcion de la excepcion
        strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.MensajeError 49" 'Mensaje en donde
se cayo el sistema
        SuscripcionSE = strMenCli 'Mensaje al usuario del sistema
        GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
        Set objServiciosOperacion = Nothing
        Exit Function
      End If
      On Error GoTo ERRORES
      Exit Function
    End If



    TerUsuOri = Trim(UCase(TerUsuOri))
    If TerUsuOri <> "" Then
      If Len(TerUsuOri) > 10 Then
        TerUsuOri = Mid(TerUsuOri, 1, 10)
      End If
    ElseIf TerUsuOri = "" Then
      On Error Resume Next
      objServiciosOperacion.MensajeError "O", strMetFunSub, 49, -1, "", SuscripcionSE
      If Err.Number <> 0 Then
        strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description 'Recuperamos el error y la
descripcion de la excepcion
        strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.MensajeError 49" 'Mensaje en donde
se cayo el sistema
        SuscripcionSE = strMenCli 'Mensaje al usuario del sistema
```

```
        GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
        Set objServiciosOperacion = Nothing
        Exit Function
      End If
      On Error GoTo ERRORES
      Exit Function
    End If

    'Verifica que el numero de celular sea numerico y tenga longitud de 9
    TerNumTar = Trim(TerNumTar)
    If Not IsNumeric(TerNumTar) Or Len(TerNumTar) <> 16 Then
      On Error Resume Next
      objServiciosOperacion.MensajeError "O", strMetFunSub, 1, -1, "", SuscripcionSE
      If Err.Number <> 0 Then
         strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description 'Recuperamos el error y la
descripcion de la excepcion
        strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.MensajeError 1" 'Mensaje en donde se
cayo el sistema
        SuscripcionSE = strMenCli 'Mensaje al usuario del sistema
        GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
        Set objServiciosOperacion = Nothing
        Exit Function
      End If
      On Error GoTo ERRORES
      Exit Function
    End If

    'Verifica que el pin de la tarjeta sea numerico y tenga longitud de 4
    TerPin = Trim(TerPin)
    If Not IsNumeric(TerPin) Or Len(TerPin) <> 4 Then
      On Error Resume Next
      objServiciosOperacion.MensajeError "O", strMetFunSub, 51, -1, "", SuscripcionSE
      If Err.Number <> 0 Then
         strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description 'Recuperamos el error y la
descripcion de la excepcion
        strMenErrSis = "ERROR: En ProcTrn al llamar objServiciosOperacion.MensajeError 51" 'Mensaje en donde
se cayo el sistema
        SuscripcionSE = strMenCli 'Mensaje al usuario del sistema
        GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
        Set objServiciosOperacion = Nothing
        Exit Function
      End If
      On Error GoTo ERRORES
      Exit Function
    End If

    TerIdeDisOriF = TerIdeDisOri
    TerUsuOriF = TerUsuOri
    TerNumTarF = TerNumTar
    DataInput = "SUSCR" & TerNumCel & TerCodSer & TerCodRedAdq & TerIdeDisOriF & TerUsuOriF &
```

```
TerNumTarF & TerPin
  ProcTrn DataInput, SuscripcionSE
  Exit Function


ERRORES:

    strMenErrSis = "ERROR: En ProcTrn N°: " & Err.Number & " DESCRIPCION: " & Err.Description
    lgnErrNumber = Err.Number
    strErrDescription = Err.Description
    On Error Resume Next
    objServiciosOperacion.MensajeError "S", strMetFunSub, 50, lgnErrNumber, strErrDescription, SuscripcionSE
    If Err.Number <> 0 Then
      strMsgErr = "ERROR N°: " & Err.Number & " DESCRIPCION: " & Err.Description
      SuscripcionSE = strMenCli
      GrabaErrorExcepcion strFileName, strPrograma, strMetFunSub, strMsgErr, strMenErrSis
      Set objServiciosOperacion = Nothing
      Exit Function
    End If
    On Error GoTo 0
End Function
```

5.2.2 *Código fuente ServiciosOperacion.*

Clase: ServiciosOperacion

Interfaz: Interfaz.

5.2.2.1 **Método: Suscribe.**

Este método es el encargado de ejecutar toda la regla de negocio.

```
Public Sub Suscribe(ByVal TerNumCel As String, _
                ByVal TerNumTar As String, _
                ByVal TerClave As String, _
                ByVal TerTransaccion As Recordset, _
                ByVal TerProBan As String, _
                ByVal TerEsLocal As String, _
                ByVal TerCodSer As String, _
                ByVal TerCodRedAdq As String, _
                ByVal TerIdeDisOri As String, _
                ByVal TerUsuOri As String, _
                ByRef DataOutput As String)

Dim lRetorno As Long
```

```
Dim RetCodMenError As String * 148
Dim RespTarHab As String
Dim RespTipTarHab As String
Dim RespIdeCli As String * 13
Dim Mensaje As String
Dim RespIsTranOk As Boolean
Dim TerTieCla As String
Dim TerTarHabAux As String * 30
Dim TerCodTra As String
Dim strMetFunSub As String

 On Error GoTo ERRORES
 strMetFunSub = "Suscribe"
 DataOutput = ""
 'Si el producto de suscripcion es Visa, MasterCard o TarejtaClub se conecta a la
 'base de tarjetas de credito
 If TerProBan = "VI' Or TerProBan = "MC" Or TerProBan = "TC" Then
   Call SetConeccionTarCre(sMensaje, lRetorno)
 'Si el producto es tarjeta de debito se conecta a la base de tarjetas de débito
 ElseIf TerProBan = "TD" Then
   Call SetConeccionTarDeb(sMensaje, lRetorno)
 End If

 If lRetorno = 0 Then   ' Se conecto a Base de Tarjetas
   If TerProBan = "VI" Or TerProBan = "MC" Or TerProBan = "TC" Then
     'Consulta si la tarjeta crédito existe y si esta con estatus activa
     ConsultaTarCre TerNumTar, RespTarHab, RespTipTarHab, RespIdeCli, RespIsTranOk, DataOutput
   ElseIf TerProBan = "TD" Then
     'Consulta si la tarjeta de debito existe y si esta con estatus activa
     ConsultaTarDeb TerNumTar, RespTarHab, RespTipTarHab, RespIdeCli, RespIsTranOk, DataOutput
   End If

   If Not RespIsTranOk Then
     Exit Sub
   End If


   Select Case TerCodRedAdq
      Case "SMS" 'Adquiriencia del SMS
          TerIdeDisOri = TerNumCel
          TerUsuOri = TerNumCel


      Case "ITR" 'Adquiriencia de la Intranet
      Case "INT" 'Adquiriencia del Internet
      Case "CMS" 'Adquiriencia del CMS

   End Select

   'Verifica si la transaccion necesita clave
```

```
    TerTieCla = TerTransaccion.Fields.Item(7).Value
    'Obtiene el codigo de transaccion asociada a la operacion
    TerCodTra = TerTransaccion.Fields.Item(0).Value
    If TerTieCla = "1" Then
      'Valida Clave
      If ObjKey Is Nothing Then
        Set ObjKey = CreateObject("Claves.Validar")
      End If

      If Not ObjKey.ValidarClave(TerNumTar, TerClave, Mensaje) Then
        RetCodMenError = "014" & Mensaje
        DataOutput = RetCodMenError
        Exit Sub
      Else
        'Clave ok
        'Registra la suscripcion banca celular
        InsertaSuscripcion TerCodSer, TerCodRedAdq, TerCodTra, TerIdeDisOri, TerUsuOri,
ObtieneOperadora(TerNumCel), TerNumCel, TerProBan, _
                    TerEsLocal, RespTipTarHab, RespTarHab, TerNumTar, RespIdeCli, sMsgClaHex(TerClave),
TerTransaccion, RespIsTranOk, DataOutput
        If Not RespIsTranOk Then
          Exit Sub
        Else
          'Si proceso de suscripcion en la base de datos es correcto arma la trama de salida
          TerTarHabAux = ""
          Select Case TerCodRedAdq
            Case "SMS" 'Respuesta al SMS
              DataOutput = "000" & RespIdeCli & TerProBan
            Case "ITR" 'Adquiriencia de la Intranet
              TerTarHabAux = RespTarHab
              DataOutput = "000" & RespIdeCli & TerProBan & TerTarHabAux & RespTipTarHab & TerEsLocal
            Case "INT" 'Adquiriencia del Internet

            Case "CMS" 'Adquiriencia del CMS
              TerTarHabAux = RespTarHab
              DataOutput = "000" & RespIdeCli & TerProBan & TerTarHabAux & RespTipTarHab
          End Select



        End If
      End If

    Else
        'Sin Clave
        InsertaSuscripcion TerCodSer, TerCodRedAdq, TerCodTra, TerIdeDisOri, TerUsuOri,
ObtieneOperadora(TerNumCel), TerNumCel, TerProBan, _
                    TerEsLocal, RespTipTarHab, RespTarHab, TerNu mTar, RespIdeCli, sMsgClaHex(TerClave),
TerTransaccion, RespIsTranOk, DataOutput
        If Not RespIsTranOk Then
```

```
            Exit Sub
          Else
            TerTarHabAux = ""
            Select Case TerCodRedAdq
              Case "SMS" 'Respuesta al SMS
                DataOutput = "000" & RespIdeCli & TerProBan
              Case "ITR" 'Adquiriencia de la Intranet
                TerTarHabAux = RespTarHab
                DataOutput = "000" & RespIdeCli & TerProBan & TerTarHabAux & RespTipTarHab & TerEsLocal
              Case "INT" 'Adquiriencia del Internet

              Case "CMS" 'Adquiriencia del CMS
                TerTarHabAux = RespTarHab
                DataOutput = "000" & RespIdeCli & TerProBan & TerTarHabAux & RespTipTarHab & TerEsLocal
            End Select
          End If

      End If
    Else
      RetCodMenError = "003" & sMensaje
      DataOutput = RetCodMenError
      RespIsTranOk = False
      Exit Sub


    End If
    Exit Sub

ERRORES:


      mensajeError "S", strMetFunSub, 5, Err.Number, Err.Description, DataOutput
      RespIsTranOk = False


End Sub
```

5.2.2.2 Método SaldoTarDeb.

Método que consulta el saldo de las cuentas asociadas a una tarjeta de débito.

```
Public Sub SaldoTarDeb(ByVal TerNumCel As String, _
                ByVal TerNumTar As String, _
                ByVal TerTipCue As String, _
                ByVal TerCla As String, _
                ByVal TerTransaccion As Recordset, _
                ByVal TerProBan As String, _
                ByVal TerEsLocal As String, _
```

```
                ByVal TerCodSer As String, _
                ByVal TerCodRedAdq As String, _
                ByVal TerIdeDisOri As String, _
                ByVal TerUsuOri As String, _
                ByRef DataOutput As String)

Dim lRetorno As Long
Dim RetCodMenError As String * 148
Dim RespTarHab As String
Dim RespTipTarHab As String
Dim RespIdeCli As String * 13
Dim Mensaje As String
Dim TerTieCla As String * 1
Dim TerTieAut As String * 1
Dim TerCodOpe As String
Dim TerTieCos As String * 1
Dim RespIsTranOk As Boolean
Dim dFecMaxPago As Date
Dim lRetCode As Integer
Dim sMsgErr As String
Dim sDes1       As String * 32
Dim sCta1       As String * 10
Dim dSaldoDis1  As Double
Dim dSaldoCon1  As Double
Dim dSaldoTot1  As Double
Dim sDes2       As String * 32
Dim sCta2       As String * 10
Dim dSaldoDis2  As Double
Dim dSaldoCon2  As Double
Dim dSaldoTot2  As Double
Dim sTipCta     As String
Dim Signo As String
Dim TerValComAdq As Double
Dim TerValComAut As Double
Dim TerFecTra As String
Dim TerHorTra As String
Dim TerTipCueDes As Integer
Dim TerCueDes As String * 10
Dim TerNumAut As String
Dim TerCodRetSMS As String * 3
Dim TerCodRetOpe As Integer
Dim CobroComision  As Boolean
Dim TerCodTra As String
Dim TerMenDes As String
Dim strMetFunSub As String

On Error GoTo ERRORES
  'Recupera parámetros de la transaccion asociada a la operacion Saldo
  TerCodTra = TerTransaccion.Fields.Item(0).Value
  TerTieCla = TerTransaccion.Fields.Item(7).Value
```

```
TerTieAut = TerTransaccion.Fields.Item(6).Value
TerCodOpe = TerTransaccion.Fields.Item(1).Value
TerTieCos = TerTransaccion.Fields.Item(5).Value
TerCodRetSMS = ""
TerMenDes = ""
VerificaTransaccion TerNumCel, TerNumTar, TerCodSer, TerCodOpe, TerTieCla, TerCla, TerTieAut, _
            TerProBan, RespTarHab, RespTipTarHab, RespIdeCli, RespIsTranOk, DataOutput
If Not RespIsTranOk Then
  Exit Sub
End If

Select Case TerCodRedAdq
  Case "SMS" 'Adquiriencia del SMS
     TerIdeDisOri = TerNumCel
     TerUsuOri = TerNumCel
  Case "INT" 'Adquiriencia del Internet
  Case "ITR" 'Adquiriencia de la Intranet
  Case "CMS" 'Adquiriencia del CMS
End Select

'Inserta registro en tabla transacciones Banco
TerFecTra = Format(Date, "yyyy/mm/dd") 'Fecha de transaccion
TerHorTra = Format(Time, "hhmmss00") 'Hora de transaccion

InsertaTransaccionSerElec TerFecTra, TerHorTra, TerCodSer, TerCodRedAdq, _
                TerCodTra, "SALDO", "", TerIdeDisOri, _
                TerUsuOri, "", "", ObtieneOperadora(TerNumCel), _
                TerNumCel, TerProBan, TerNumTar, _
                RespTipTarHab, RespIsTranOk, DataOutput

If Not RespIsTranOk Then
  Exit Sub
End If
TerCodRetOpe = -1
'Instancia el objeto que consulta el saldo
If obj Is Nothing Then
  Set obj = CreateObject("AutorizadorSerElec.SMS")
End If

Select Case TerTipCue
  Case "CA"
     sTipCta = "10"
  Case "CC"
     sTipCta = "20"
  Case "TO"
     sTipCta = ""
  Case Else
     sTipCta = ""
End Select
```

```
  Call obj.ConsultaSaldoDebito(Trim(TerNumTar), sTipCta, _
  sDes1, sCta1, dSaldoDis1, dSaldoCon1, dSaldoTot1, _
  sDes2, sCta2, dSaldoDis2, dSaldoCon2, dSaldoTot2, lRetCode, sMsgErr)
  sDes1 = "CUENTA AHORROS"
  sDes2 = "CUENTA CORRIENTE"
  If lRetCode = 0 Then
    'Genera transaccion de cobro de comision
    CobroComision = False
    If TerTieCos = "1" Then
      ObtieneComision TerEsLocal, TerProBan, RespTipTarHab, TerTransaccion, TerValComAdq, TerValComAut
      If obj Is Nothing Then
        Set obj = CreateObject("AutorizadorSerElec.SMS")
      End If
      Dim TerTipCueDesAux As String
        Call obj.CobrarServicio(TerNumTar, CDbl(TerValComAdq + TerValComAut), "SMS01", TerNumEst, TerCat,
"0", _
                        TerNumAut, TerTipCueDesAux, TerCueDes, lRetCode, sMsgErr)
      'TerNumAut = "123456"
      'TerTipCueDes = 10
      'TerCueDes = "702285"
      'lRetCode = 0
      'lRetCode = 125
       'sMsgErr = "USUARIO NO TIENE FONDOS DE COMISION"
      TerTipCueDes = Val(TerTipCueDesAux)
      If lRetCode <> 0 Then
        If lRetCode >= 0 And lRetCode <= 999 Then
          sMsgErr = Trim(UCase(sMsgErr))
          RetCodMenError = sMsgErr
          DataOutput = Format(lRetCode, "000") & RetCodMenError
          If Len(sMsgErr) > 50 Then
            sMsgErr = Mid(sMsgErr, 1, 50)
          End If
        Else
          mensajeError "O", strMetFunSub, 30, -1, "", DataOutput
        End If
      Else
        CobroComision = True
      End If
    Else
      CobroComision = True
    End If


    If CobroComision Then
      DataOutput = "000"
      Signo = ""

      DataOutput = DataOutput & sDes1 & sCta1
      'Saldo disponible 1
      If dSaldoDis1 >= 0 Then
```

```
        Signo = "+"
      Else
        Signo = ""
      End If
      DataOutput = DataOutput & Signo & Format(dSaldoDis1 * 100, "000000000000")
      'Saldo por confirmar 1
      If dSaldoCon1 >= 0 Then
        Signo = "+"
      Else
        Signo = ""
      End If
      DataOutput = DataOutput & Signo & Format(dSaldoCon1 * 100, "000000000000")

      'Saldo Total 1
      If dSaldoTot1 >= 0 Then
        Signo = "+"
      Else
        Signo = ""
      End If
      DataOutput = DataOutput & Signo & Format(dSaldoTot1 * 100, "000000000000")

      DataOutput = DataOutput & sDes2 & sCta2
      'Saldo disponible 2
      If dSaldoDis2 >= 0 Then
        Signo = "+"
      Else
        Signo = ""
      End If
      DataOutput = DataOutput & Signo & Format(dSaldoDis2 * 100, "000000000000")

      'Saldo por confirmar 1
      If dSaldoCon2 >= 0 Then
        Signo = "+"
      Else
        Signo = ""
      End If
      DataOutput = DataOutput & Signo & Format(dSaldoCon2 * 100, "000000000000")

      'Saldo Total 2
      If dSaldoTot2 >= 0 Then
        Signo = "+"
      Else
        Signo = ""
      End If
      DataOutput = DataOutput & Signo & Format(dSaldoTot2 * 100, "000000000000")
    End If
  Else
    If lRetCode >= 0 And lRetCode <= 999 Then
      sMsgErr = Trim(UCase(sMsgErr))
      RetCodMenError = sMsgErr
```

```
        DataOutput = Format(lRetCode, "000") & RetCodMenError
        If Len(sMsgErr) > 50 Then
          sMsgErr = Mid(sMsgErr, 1, 50)
        End If
      Else
        mensajeError "O", strMetFunSub, 32, -1, "", DataOutput
      End If
    End If
    TerCodRetSMS = Mid(DataOutput, 1, 3)
    TerMenDes = Trim(UCase(DataOutput))
    'Graba codigo retorno Banco
    ActualizaTransaccionSerElec TerFecTra, TerHorTra, TerCodSer, TerCodRedAdq, TerCodTra, "", _
                        TerIdeDisOri, TerUsuOri, TerNumCel, TerNumTar, TerTipCueDes, _
                        TerCueDes, TerValComAdq, TerValComAut, TerValComAdq + TerValComAut, _
                        TerNumAut, lRetCode, TerCodRetSMS, TerCodRetOpe, sMsgErr, TerMenDes, RespIsTranOk,
DataOutput, "L"

    If Not RespIsTranOk Then
      Exit Sub
    End If
    Exit Sub

ERRORES:
    mensajeError "S", strMetFunSub, 33, Err.Number, Err.Description, DataOutput
    RespIsTranOk = False

End Sub
```

5.2.3 *Página SuscripcionSMS.asp*

```
<%@ LANGUAGE=VBScript %>
<%
response.buffer=true
userid=session("userid")
if len(userid)<>0 then
  password=session("password")
  oficina=session("oficina")
  fechajae=session("fechajae")
else
  response.clear
  response.redirect "http://server/sms/Seguridad.asp"
end if


horajae= time()
fechajae1 = date()
bandera=request("bandera")
txtNumCel=request("txtNumCel")
txtNumTar=request("txtNumTar")
```

```
txtClave=request("txtClave")
txtConClave=request("txtConClave")
txtClaveNueSE=request("txtClaveNueSE")
txtConClaveNueSE=request("txtConClaveNueSE")


bandera=Request("bandera")
if len(bandera)<> 0 then
        respuesta=""

        select case bandera
        case 1:


      TerCodSer="SMS"
      TerCodRedAdq="ITR"
      TerIdeDisOri=Request.ServerVariables("REMOTE_HOST")
      TerUsuOri=userid
      TerDataOutput=""

            Set Objeto = Server.CreateObject("ServiciosElectronicos.Interfaz")
            TerDataOutput = Objeto.SuscripcionSE (TxtNumCel,TerCodSer,TerCodRedAdq,TerIdeDisOri,
TerUsuOri,TxtNumTar,TxtClave)
      TerNumErr = mid(TerDataOutput,1, 3)

      if TerNumErr = "000" then
        TerDesErr = "TRANSACCION OK"
        TerIdeCli = mid(TerDataOutput,4, 13)
        TerProBan = mid(TerDataOutput,17, 2)
        Select Case TerProBan
          case "TD"
            TerProBan=TerProBan & " - TARJETA DEBITO"
          case "VI"
            TerProBan=TerProBan & " - VISA"
          case "MC"
            TerProBan=TerProBan & " - MASTERCARD"
          case "TC"
            TerProBan=TerProBan & " - TARJETA CLUB"
          case else
            TerProBan="PRODUCTO DESCONOCIDO"
        end Select


        TerTarHab = mid(TerDataOutput,19, 30)
        TerTipTarHab = mid(TerDataOutput,49, 1)
        Select Case TerTipTarHab
          case "N"
            TerTipTarHab=TerTipTarHab & " - NORMAL"
          case "E"
            TerTipTarHab=TerTipTarHab & " - EMPLEADO"
```

```
            case "V"
                TerTipTarHab=TerTipTarHab & " - VIP"
            case "P"
                TerTipTarHab=TerTipTarHab & " - VIP PLUS"
            case else
                TerTipTarHab="TIPO TARJETAHABIENTE DESCONOCIDO"
          end Select

          TerEsLocal = mid(TerDataOutput,50,1)
          Select Case TerEsLocal
            case "1"
                TerOrigen="INSTITUCION LOCAL"
            case "0"
                TerOrigen="COOPERATIVAS"
            case else
                TerOrigen="ORIGEN DESCONOCIDO"
          end Select
          bandera =5
        else
          TerDesErr =mid(TerDataOutput,4, len(terdataoutput)-3)
          bandera=3
        end if
        case 2:
        response.redirect "http://server/sms/SuscripcionSMS.asp"
        case 4:
        response.redirect "http://server/sms/MenuSMS.asp"
        case 6:
        response.redirect "http://server/sms/MenuSMS.asp"
    case 7: 'Cambio Clave
        TerCodSer="SMS"
        TerCodRedAdq="ITR"
        TerIdeDisOri=Request.ServerVariables("REMOTE_HOST")
        TerUsuOri=userid
        TerDataOutput=""
        Set Objeto = Server.CreateObject("ServiciosElectronicos.Interfaz")
      TerDataOutput = Objeto.CambiaClaveSE (TxtNumCel,TerCodSer,TerCodRedAdq,TerIdeDisOri,
TerUsuOri,TxtNumTar,TxtClave, txtClaveNueSE)
      TerNumErr = mid(TerDataOutput,1, 3)
        if TerNumErr = "000" then
          TerDesErr = "CAMBIO CLAVE SERVICIOS ELECTRONICOS OK"
          TerIdeCli = mid(TerDataOutput,4, 13)
          TerProBan = mid(TerDataOutput,17, 2)
          Select Case TerProBan
            case "TD"
                TerProBan=TerProBan & " - TARJETA DEBITO"
            case "VI"
                TerProBan=TerProBan & " - VISA"
            case "MC"
                TerProBan=TerProBan & " - MASTERCARD"
            case "TC"
```

```
            TerProBan=TerProBan & " - TARJETA CLUB"
          case else
            TerProBan="PRODUCTO DESCONOCIDO"
        end Select
        TerTarHab = mid(TerDataOutput,19, 30)
        TerTipTarHab = mid(TerDataOutput,49, 1)
        Select Case TerTipTarHab
          case "N"
            TerTipTarHab=TerTipTarHab & " - NORMAL"
          case "E"
            TerTipTarHab=TerTipTarHab & " - EMPLEADO"
          case "V"
            TerTipTarHab=TerTipTarHab & " - VIP"
          case "P"
            TerTipTarHab=TerTipTarHab & " - VIP PLUS"
          case else
            TerTipTarHab="TIPO TARJETAHABIENTE DESCONOCIDO"
        end Select
        TerEsLocal = mid(TerDataOutput,50,1)
        Select Case TerEsLocal
          case "1"
            TerOrigen="INSTITUCION LOCAL"
          case "0"
            TerOrigen="COOPERATIVAS"
          case else
            TerOrigen="ORIGEN DESCONOCIDO"
        end Select
        bandera = 8
      else
        TerDesErr =mid(TerDataOutput,4, len(terdataoutput)-3)
        bandera=3
      end if
          case 9: 'Llama a suscribir servicios adicionales
      response.redirect "http://server/sms/SuscripcionAutorizadaSMS.asp?n1=" & txtNumCel & "&n2=" &
txtNumTar
end select
end if
%>

<SCRIPT LANGUAGE="VBScript">

sub Suscripcion_OnClick
          dim registro
          Set registro = Document.SuscripcionSMS
          registro.bandera.value="1"
          if not IsNumeric(trim(registro.txtNumCel.value)) or len(trim(registro.txtNumCel.value))<> 9 then
            MSGBOX "El número de celular debe contener 9 caracteres numéricos",48,"ERROR INGRESO DE
DATOS"
          else
            if not IsNumeric(trim(registro.txtNumTar.value)) or len(trim(registro.txtNumTar.value))<> 16 then
```

```
                MSGBOX "El número de tarjeta de débito/crédito debe contener 16 caracteres
numéricos",48,"ERROR INGRESO DE DATOS"
      else
              if not IsNumeric(trim(registro.txtClave.value)) or len(trim(registro.txtClave.value))<> 4 then
                    MSGBOX "El pin de la tarjeta de débito/crédito debe contener 4 caracteres
numéricos",48,"ERROR INGRESO DE DATOS"
         else
            if trim(registro.txtClave.value) <> trim(registro.txtConClave.value) then
              MSGBOX "Clave y confirmacion de clave son diferentes. Por favor ingrese nuevamente clave y
confirmación",48,"ERROR INGRESO DE DATOS"
                    else
             registro.submit
            end if
         end if
              end if
            end if
end sub

sub NuevaSuscripcion_OnClick
            dim registro
            Set registro = Document.SuscripcionSMS
            registro.bandera.value="2"
   registro.submit
end sub

sub MenuBancaMobil_OnClick
            dim registro
            Set registro = Document.SuscripcionSMS
            registro.bandera.value="4"
   registro.submit
end sub

sub NuevaClave_OnClick

            dim registro
            Set registro = Document.SuscripcionSMS
            registro.bandera.value="7"
            if not IsNumeric(trim(registro.txtNumCel.value)) or len(trim(registro.txtNumCel.value))<> 9 then
              MSGBOX "El número de celular debe contener 9 caracteres numéricos",48,"ERROR INGRESO DE
DATOS"
            else
              if not IsNumeric(trim(registro.txtNumTar.value)) or len(trim(registro.tx tNumTar.value))<> 16 then
                MSGBOX "El número de tarjeta de débito/crédito debe contener 16 caracteres
numéricos",48,"ERROR INGRESO DE DATOS"
      else
              if not IsNumeric(trim(registro.txtClave.value)) or len(trim(registro.txtClave.value))<> 4 then
                    MSGBOX "El pin de la tarjeta débito/crédito debe contener 4 caracteres numéricos",48,"ERROR
INGRESO DE DATOS"
         else
              if not IsNumeric(trim(registro.txtClaveNueSE.value)) or len(trim(registro.txtClaveNueSE.value))<> 4
```

```
then
                        MSGBOX "La nueva clave de los servicios electrónicos debe contener 4 caracteres
numéricos",48,"ERROR INGRESO DE DATOS"
                      else
                        if trim(registro.txtClaveNueSE.value) <> trim(registro.txtConClaveNueSE.value) then
                  MSGBOX "Nueva clave y confirmacion son diferentes. Por favor ingrese nuevamente clave y
confirmación",48,"ERROR INGRESO DE DATOS"
                        else
                  registro.submit
                end if
              end if
            end if
                 end if
              end if

end sub

sub ServicioAdicional_OnClick
              dim registro
              Set registro = Document.SuscripcionSMS
              registro.bandera.value="9"
   registro.submit
end sub


</SCRIPT>



<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>
<head>
<title>Suscripcion Servicio BANCA CELULAR</title>
<meta name="GENERATOR" content="Microsoft FrontPage 5.0">
<meta name="Microsoft Theme" content="fondo 1111">
</head>
<body link="#FFFFFF" vlink="#00FFFF" alink="#00FFFF" text="#C0C0C0" bgcolor="#0066CC">
<STRONG><h1 ALIGN="CENTER"><font size="4" face="Century Gothic">SUSCRIPCION SERVICIO
ELECTRONICO BANCA CELULAR</font></h1>
<FONT SIZE=5 COLOR="#000080">
<hr>
<form name="SuscripcionSMS" method="POST" action="SuscripcionSMS.asp">
<center>
<%if bandera = 0 or bandera = 3 or bandera = 5 or bandera = 8 then %>
<TABLE CELLSPACING=0 BORDER=0 WIDTH=100%>
<TR>
<TD WIDTH="50%" VALIGN="MIDDLE"><P><font face="Arial">Número de Celular :</font></TD>
<TD WIDTH="50%" VALIGN="MIDDLE"><P><input type=text size=9 maxlength=9 name="txtNumCel" value =
<%=txtNumCel%>></TD>
</TR>
```

```
<TR>
<TD WIDTH="50%" VALIGN="MIDDLE"><P><font face="Arial">Número Tarjeta Crédito/Débito :</font></TD>

<TD WIDTH="50%" VALIGN="MIDDLE"><P><input type=text size=16 maxlength=16 name="txtNumTar" value =
<%=txtNumTar%>></TD>
</TR>
<TR>
<TD WIDTH="50%" VALIGN="MIDDLE"><P><font face="Arial" color=orangered><STRONG>Clave Tarjeta
Crédito/Débito :</STRONG></font></TD>
<TD WIDTH="50%" VALIGN="MIDDLE"><P><input type=passwcrd size=4 maxlength=4 name="txtClave" value =
<%=txtClave%>></TD>
</TR>
<TR>
<TD WIDTH="50%" VALIGN="MIDDLE"><P><font face="Arial" color=orangered><STRONG>Confirme Clave
Tarjeta Crédito/Débito :</STRONG></font></TD>
<TD WIDTH="50%" VALIGN="MIDDLE"><P><input type=password size=4 maxlength=4 name="txtConClave"
value = <%=txtConClave%>></TD>
</TR>
</TABLE>
<hr>
<p>
<%end if
if bandera=3 then%>
<TABLE CELLSPACING=5 BORDER=0 WIDTH=100%>
<tr>
<td WIDTH="30%" ><font face="Arial">N°Error:</a></font></td>
<td WIDTH="70%" ><FONT COLOR="#FFFFFF" face="Arial"><%=TerNumErr%></font></td>
</tr>
<tr>
<td WIDTH="30%" ><font face="Arial">Descripción:</a></font></td>
<td WIDTH="70%" ><FONT COLOR="#FFFFFF" face="Arial"><%=TerDesErr%></font></td>
</tr>
</TABLE>
<hr>
<p>
<%end if
if bandera=5 or bandera = 8 then%>
<TABLE CELLSPACING=5 BORDER=0 WIDTH=100%>
<tr>
<td WIDTH="30%" ><font face="Arial">Status:</a></font></td>
<td WIDTH="70%" ><FONT COLOR="#FFFFFF" face="Arial"><%=TerDesErr%></font></td>
</tr>
<tr>
<td WIDTH="30%" ><font face="Arial">Identificacion:</a></font></td>
<td WIDTH="70%" ><FONT COLOR="#FFFFFF" face="Arial"><%=TerIdeCli%></font></td>
</tr>
<tr>
<td WIDTH="30%" ><font face="Arial">Tipo Tarjeta:</a></font></td>
<td WIDTH="70%" ><FONT COLOR="#FFFFFF" face="Arial"><%=TerProBan%></font></td>
</tr>
```

```
<tr>
<td WIDTH="30%" ><font face="Arial">Cliente:</a></font></td>
<td WIDTH="70%" ><FONT COLOR="#FFFFFF" face="Arial"><%=TerTarHab%></font></td>
</tr>
<tr>
<td WIDTH="30%" ><font face="Arial">Tipo Cliente:</a></font></td>
<td WIDTH="70%" ><FONT COLOR="#FFFFFF" face="Arial"><%=TerTipTarHab%></font></td>
</tr>
<tr>
<td WIDTH="30%" ><font face="Arial">Lugar de Origen:</a></font></td>
<td WIDTH="70%" ><FONT COLOR="#FFFFFF" face="Arial"><%=TerOrigen%></font></td>
</tr>
<TR>
<TD WIDTH="50%" VALIGN="MIDDLE"><P><font face="Arial" color=orangered><STRONG>Nueva clave servicios
electrónicos:</STRONG></font></TD>
<TD WIDTH="50%" VALIGN="MIDDLE"><P><input type=password size=4 maxlength=4 name="txtClaveNueSE"
value = <%=txtClaveNueSE%>></TD>
</TR>
<TR>
<TD WIDTH="50%" VALIGN="MIDDLE"><P><font face="Arial" color=orangered><STRONG>Confirme nueva clave
servicios electrónicos:</STRONG></font></TD>
<TD WIDTH="50%" VALIGN="MIDDLE"><P><input type=password size=4 maxlength=4
name="txtConClaveNueSE" value = <%=txtConClaveNueSE%>></TD>
</TR>
</TABLE>
<hr>
<p>
<%end if %>
<TABLE CELLSPACING=0 BORDER=0 WIDTH=100%>
<TR>
<%If bandera = 0 or bandera = 3 then%>
<TD ALIGN="CENTER" WIDTH="50%" VALIGN="MIDDLE"><P><input type=button name="Suscripcion"
value="Enviar suscripción"></TD>
<%elseif BANDERA = 5 OR bandera = 3 then%>
<TD ALIGN="CENTER" WIDTH="50%" VALIGN="MIDDLE"><P><input type=button name="NuevaClave"
value="Enviar nueva clave"></TD>
<%elseif BANDERA = 8 then%>
<TD ALIGN="CENTER" WIDTH="50%" VALIGN="MIDDLE"><P><input type=button name="ServicioAdicional"
value="Servicios Adicionales"></TD>
<%end if%>
<TD ALIGN="CENTER" WIDTH="50%" VALIGN="MIDDLE"><P><input type=button name="NuevaSuscripcion"
value="Nueva Suscripción"></TD>
<TD ALIGN="CENTER" WIDTH="50%" VALIGN="MIDDLE"><P><input type=button name="MenuBancaMobil"
value="Menu Banca CELULAR"></TD>
</TR>
</TABLE>
</center>
<p>
<hr>
<center>
```

```
<table border=0>
<tr>
<td><font color="#FFFFFF" size=1><strong><CENTER>Fecha: </color></strong>
<%=fechajae1%></CENTER></td>
<td><font color="#FFFFFF" size=1><strong><CENTER>Usuario: </color></strong>
<%=userid%></CENTER></td>
<td><font color="#FFFFFF" size=1><strong><CENTER>Hora: </color></strong> <%=horajae%></CENTER></td>
</tr>
<tr><td></td><td><font color="#FFFFFF" size=1><strong><CENTER>Ultima Actualización: </color></strong>22-
Mayo-2007</CENTER></td><td></td></tr>
</table>
</center>
<input type=hidden name="bandera" value=<%=bandera%>>
</form>
</body>
</html>
```

5.3 Pruebas Orientadas a Objetos.

El objetivo fundamental es encontrar el mayor número de errores de la solución generada con una cantidad de esfuerzo racional en un tiempo realista.

5.3.1 *Modelos de Prueba AOO (Análisis Orientado a Objetos) y DOO (Diseño Orientado a Objetos).*

Los modelos de AOO y DOO no pueden probarse ejecutándoles, sino trata de verificar su exactitud y consistencia de AOO y DOO con relación al dominio del problema (exactitud = sintáctico y semántico, consistencia = clases y sus conexiones con otras clases)

Exactitud:

- Sintáctica, se revisa si tiene la apropiada simbología, convencionalidades
- Semántica, la conformidad del modelo con el dominio del problema (refleja el mundo real)

Consistencia:

- Relaciones entre objetos, se examina las clases y sus conexiones a otras clases con la ayuda de el modelo CRC y modelo objeto-relación

Verificación de exactitud y consistencia.

Exactitud:

- Sintáctica,

  Todos los artefactos desarrollados en las fases de planificación, análisis y diseño están apegados a los estándares de nomenclatura que propone RUP.

  Especificación de requisitos, desarrollados con IEEE 802.30.
  Diagramas de casos de uso, desarrollados con Visio de acuerdo al estándar UML.
  Casos de uso esenciales expandidos, de acuerdo a plantilla RUP.
  Diagramas de colaboración, desarrollados con Visio de acuerdo al estándar UML.
  Diagramas de Clase, estándar de las variables: TerXXXXXXXXXX en donde es el nombre descriptivo de la variable que se agrupan en tres caracteres. Por ejemplo TerCodRedAdq, esto significa variable en donde se guarda el código de la red adquiriente.
  Diagrama Objeto-Relación, el nombre de la tabla tiene la siguiente estructura TXY-XXX..YYY.. en donde, T=Tabla, X=Nombre descriptivo de la tabla, Y=Nombre descriptivo de la tabla por ejemplo TPB-PushBatch.

- Semántica,

  Todos los artefactos generados en el AOO utilizan de manera adecuada términos y simbología relacionada con la banca y la tecnología celular SMS. Los conceptos que se han utilizado son generales.

  Uno de los artefactos generados en el AOO es el modelo conceptual que trata abstraer conceptos del mundo real y apegarlos al modelo de la posible solución. Los términos utilizados en el modelo conceptual están de acuerdo al dominio del negocio.

  Términos técnicos han sido escritos en el documento glosario.

Consistencia:

Antes de la programación se probó la consistencia de los diferentes métodos diseñados para las diferentes operaciones PULL, PUSH, BACKOFFICE, con los diagramas de secuencia obtenidos en la fase de análisis.

Procedimiento de prueba de consistencia antes de la programación.

a) Selecciona el diagrama de secuencia a probar.
b) Por cada interacción entre el usuario del sistema y el sistema banca celular se verifica que el método(s) o función(s) realicen las tareas respectivas que en conjunto deben devolver un resultado al usuario del sistema.

Resultado:

"El resultado de estas pruebas son satisfactorias para cada diagrama de secuencia".

Procedimiento de prueba de consistencia después de la programación.

Estas pruebas tratan de verificar la consistencia de las clases implementadas, sus métodos y funciones y los diagramas de colaboración obtenidos en la fase de diseño.

c) Selecciona el diagrama de colaboración de la fase de diseño.
d) Verifica que la colaboración y responsabilidades entre objetos sean consistentes.

Resultado:

"El resultado de estas pruebas son satisfactorias para cada diagrama de colaboración desarrollado para las operaciones PULL, PUSH y BACKOFFICE".

Procedimiento de prueba de consistencia después de la integración entre objetos.

Se probó las relaciones entre objetos, este sistema no es rico en clases, debido a que cuando se pensó en el diseño del sistema, peso mucho el rendimiento y los tiempos de respuesta que debe tener frente a un requerimiento del cliente. Adicionalmente su modelado de negocio no exige crear clases en demasía sin justificarlas, sin embargo los métodos y funciones de la regla del negocio son generales para reutilizarlas en funciones como por ejemplo VerificarSuscripcion que se utiliza en todas las operaciones PULL.

Se revisa los diagramas de colaboración y modelo OR. Se verifica la colaboración entre los objetos ServiciosElectronicos y ServiciosOperacion de manera correcta y su interacción con los datos.

El diseño del sistema se ha revisado en función a los diagramas de secuencia desarrollados en AOO, donde se examinan el comportamiento del sistema y sus subsistemas, asignación de tareas y concurrencia. Podemos destacar que todos los escenarios planteados en los diagramas de secuencia fueron analizados y son soportados por las clases diseñadas. La concurrencia es un aspecto que en este tipo de diseños son muy soportados destacando que cada instancia de la clase se ejecuta en una área de memoria en un servidor de componentes.

5.3.2 *Definición de estrategias para pruebas Orientadas a objetos.*

Se inicia con la prueba de pequeña escala o la unidad, pasando por la validación o prueba del sistema, extendiéndose hasta las pruebas de desempeño (de forma temprana), Pruebas de aceptación y Pruebas de instalación

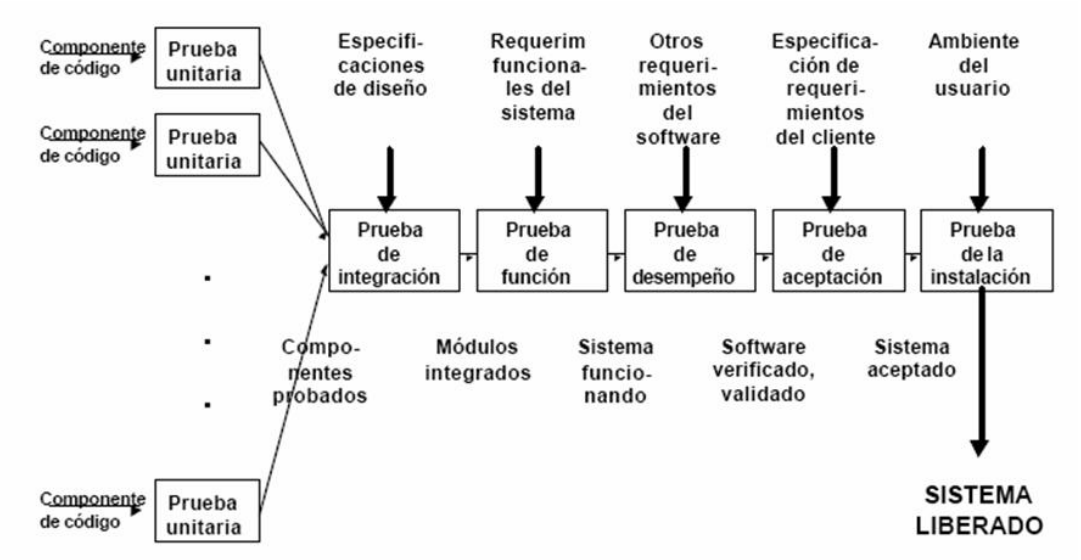


Figura 5.1 Estrategia de pruebas orientadas a objetos

Las pruebas de unidad, hemos realizado en el momento de desarrollo. No generamos ninguna documentación formal de este punto porque la gran cantidad de información que esto implica.

A continuación desarrollamos las pruebas de validación, pruebas de esfuerzo o de estrés, pruebas de rendimiento reales. Debemos recordar que el objetivo de en la fase de pruebas es el de encontrar el mayor número de errores, también es importante considerar que al momento de desarrollar cada unidad, método o función el programador realizan pruebas y se corrigen errores técnicos de tal manera que en las pruebas de validación se minimizan errores en programación más bien se trata de encontrar errores de regla del negocio.

Se presenta un pequeño marco teórico acerca de lo que trata cada una de las pruebas se que sugiere realizar en unas pruebas orientadas a objetos.

5.3.2.2 **Prueba de unidad**

Es la menor unidad a probar es la clase u objeto encapsulado. Pero una clase contiene Operaciones y una Operación puede existir como parte de un número de clases diferentes, por lo tanto no se puede tratar por aislado sino como parte de una clase.

5.3.2.3 Prueba de integración

Dispone de 2 estrategias:

Pruebas basadas en hilos, que es integrada por un conjunto de clases necesarias para responder a una entrada o evento del sistema, cada hilo se integra y se prueba por separado.

Pruebas basadas en uso, se comienza con las clases independientes que se usan muy poco, luego de concluir con las clases independientes nos vamos por la siguiente capa de clases dependientes que usan clases independientes, esta secuencia de probar por capas sigue hasta probar con todo el sistema

5.3.2.4 Prueba de validación (funcionalidad)

Se centra en las acciones visibles del usuario de I/O del sistema. Nos basamos de los casos de uso, donde se ven escenarios que tendrán la una alta probabilidad de errores al momento de interacción con el cliente.

5.3.2.5 Prueba de esfuerzo o desempeño

Sirven para determinar si el sistema está preparado para asegurar el procesamiento de determinadas cargas de trabajo. Probar el comportamiento erróneo del sistema. Mostrar defectos que normalmente no se manifestarían por sí solos. Las pruebas de esfuerzo son particularmente relevantes en los sistemas distribuidos

5.3.2.6 Prueba de aceptación (certificación)

Propias del usuario para confirmar que cumple con los requisitos. Normalmente cada institución tiene un departamento de auditoria que certifica el funcionamiento del producto antes del ingreso a producción.

5.3.2.7 Prueba de instalación

Una vez instalado se verifica que en el ambiente en producción su comportamiento es correcto, inclusive esta fase suele extenderse por largo tiempo en el sentido de probar el producto/servicio por tiempo continuo para monitorear la calidad del servicio.

5.3.3 *Pruebas de validación. Definición de casos de pruebas y resultados.*

5.3.3.1 *Operaciones PULL*

Para cada operación se ha planificado un set de pruebas. En el documento de planificación se indica el comando SMS que tiene que ingresar el usuario. En la primera fila se indican los procesos que ejecutan la operación y sobre cuales se evalúa la prueba, en la primera columna se indica la prueba realizada, por ultimo tenemos las columnas de resultado esperado, resultado obtenido y el resultado de la prueba. Gran parte de los errores encontrados son depurados y corregidos. Por cada prueba se presenta un Log de respaldo de las pruebas, en donde se puede verificar la trama de entrada al método Proctrn de la clase ServiciosElectronicos en el parámetro de entrada DataInput y la trama de salida devuelto por el método en el parámetro DataOutput. Los mensajes desplegados a los clientes en su celular no los podemos capturar para mostrar en este documento, sin embargo en el capítulo 4 se indicaba las plantillas utilizadas para formatear el mensaje que llegara al cliente por cada operación banca celular.

5.3.3.1.1 Pruebas operación "SUSCR"

OPERACIÓN SUSCR													
SOL + (N° Tarjeta Debito/Credito) + Clave Tarjeta	Sintaxis instrucción	Verifica autorización BIN	Verifica autorización de trans. para BIN	Verifica si suscr existe y si esta activa	Verifica estatus de tarjeta	InsertaRegistroTransaccionalAntesSus	InsertaSuscripcionBancaCelular	ActualizaRegistroTransaccionalDesSus	FormatoSalida	VerificaResultadoOperacion	ResultadoEsperado	ResultadoObtenido	ResultadoPrueba
Num tarjeta incorrecta	ok										err	err	ok
Celular incorrecto	ok										err	err	ok
Clave incorrecta	ok										err	err	ok
Bin no pertenece a institución		Ok									err	err	ok
Transaccion 00122 para red sms en estado inactiva			ok								err	err	ok
Envía suscripcion existente				ok							err	err	ok
Tarjeta cancelada					Ok						err	err	ok
Suscripcion ok						ok	ok	ok	ok	ok	ok	ok	ok

Tabla 5.1 Resultado de pruebas de operación PULL.

5.3.3.1.2 Log de soporte de pruebas "SUSCR"

FECHA: 2007/06/13 HORA: 13380400 FRMCLIENTESERVICIOS SUSCRIPCION**********INICIO**********

FECHA: 2007/06/13 HORA: 13380400 FRMCLIENTESERVICIOS SUSCRIPCION INPUT: SUSCR091897074SMSITRPCDLANDY JDLANDY dfdfdfdf xxxx

FECHA: 2007/06/13 HORA: 13380400 FRMCLIENTESERVICIOS SUSCRIPCION OUTPUT: 001ERROR: NUMERO DE TARJETA DEBE TENER 16 VALORES NUMERICOS

FECHA: 2007/06/13 HORA: 13380400 FRMCLIENTESERVICIOS SUSCRIPCION**********FIN**********

FECHA: 2007/06/13 HORA: 13411700 FRMCLIENTESERVICIOS SUSCRIPCION**********INICIO**********

FECHA: 2007/06/13 HORA: 13411700 FRMCLIENTESERVICIOS SUSCRIPCION INPUT: SUSCRdfdfdfSMSITRPCDLANDY JDLANDY NNNNNN0010080245 xxxx

FECHA: 2007/06/13 HORA: 13411700 FRMCLIENTESERVICIOS SUSCRIPCION OUTPUT: 068ERROR: NUMERO DE CELULAR CONTIENE VALORES NO NUMERICOS

FECHA: 2007/06/13 HORA: 13411700 FRMCLIENTESERVICIOS SUSCRIPCION**********FIN**********

FECHA: 2007/06/13 HORA: 13421000 FRMCLIENTESERVICIOS SUSCRIPCION**********INICIO**********

FECHA: 2007/06/13 HORA: 13421000 FRMCLIENTESERVICIOS SUSCRIPCION INPUT: SUSCR091897079SMSITRPCDLANDY JDLANDY NNNNNN0010080245 df

FECHA: 2007/06/13 HORA: 13421000 FRMCLIENTES ERVICIOS SUSCRIPCION OUTPUT: 018ERROR: NUMERO DE CLAVE CONTIENE VALORES NO NUMERICOS

FECHA: 2007/06/13 HORA: 13421000 FRMCLIENTESERVICIOS SUSCRIPCION**********FIN**********

FECHA: 2007/06/13 HORA: 13432000 FRMCLIENTESERVICIOS SUSCRIPCION**********INICIO**********

FECHA: 2007/06/13 HORA: 13432000 FRMCLIENTESERVICIOS SUSCRIPCION INPUT: SUSCR091897079SMSITRPCDLANDY JDLANDY 1245210010080245 xxxx

FECHA: 2007/06/13 HORA: 13432000 FRMCLIENTESERVICIOS SUSCRIPCION OUTPUT: 002ERROR: BIN NO REGISTRADO EN INSTITUCION XXX O RED APOYO

FECHA: 2007/06/13 HORA: 13432000 FRMCLIENTESERVICIOS SUSCRIPCION**********FIN**********

FECHA: 2007/06/13 HORA: 13465200 FRMCLIENTESERVICIOS SUSCRIPCION**********INICIO**********

FECHA: 2007/06/13 HORA: 13465200 FRMCLIENTESERVICIOS SUSCRIPCION INPUT: SUSCR096240926SMSSMSPCDLANDY JDLANDY NNNNNN0010080245 xxxx

FECHA: 2007/06/13 HORA: 13465200 FRMCLIENTESERVICIOS SUSCRIPCION OUTPUT: 098ERROR: NO EXISTE TRANSACCION SOLICITADA O NO ESTA ACTIVA

FECHA: 2007/06/13 HORA: 13465200 FRMCLIENTESERVICIOS SUSCRIPCION**********FIN**********

FECHA: 2007/06/13 HORA: 13495100 FRMCLIENTESERVICIOS SUSCRIPCION**********INICIO**********

FECHA: 2007/06/13 HORA: 13495100 FRMCLIENTESERVICIOS SUSCRIPCION INPUT: SUSCR091897079SMSSMSPCDLANDY JDLANDY NNNNNN0010080245 xxxx

FECHA: 2007/06/13 HORA: 13495200 FRMCLIENTESERVICIOS SUSCRIPCION OUTPUT: 013ERROR:

CLIENTE YA ESTA SUSCRITO AL SERVICIO BANCA CELULAR DE L A INSTITUCION XXX
FECHA: 2007/06/13 HORA: 13495200 FRMCLIENTESERVICIOS SUSCRIPCION**********FIN**********
FECHA: 2007/06/13 HORA: 13533400 FRMCLIENTESERVICIOS SUSCRIPCIO N**********INICIO**********
FECHA: 2007/06/13 HORA: 13533400 FRMCLIENTESERVICIOS SUSCRIPCION INPUT: SUSCR096240926SMSSMSPCDLANDY JDLANDY NNNNNN0010080245 xxxx
FECHA: 2007/06/13 HORA: 13533400 FRMCLIENTESERVICIOS SUSCRIPCION OUTPUT: 011ERROR: Tarjeta no activa. Motivo Cancelada. Por favor comunicarse con Servicios Bancarios...Gracias
FECHA: 2007/06/13 HORA: 13533400 FRMCLIENTESERVICIOS SUSCRIPCION**********FIN**********
FECHA: 2007/06/13 HORA: 13572700 FRMCLIENTESERVICIOS SUSCRIPCION**********INICIO**********
FECHA: 2007/06/13 HORA: 13572800 FRMCLIENTESERVICIOS SUSCRIPCION INPUT: SUSCR096240926SMSSMSPCDLANDY JDLANDY NNNNNN0010080245 xxxx
FECHA: 2007/06/13 HORA: 13572900 FRMCLIENTESERVICIOS SUSCRIPCION OUTPUT: 0000102777091 TD
FECHA: 2007/06/13 HORA: 13573000 FRMCLIENTESERVICIOS SUSCRIPCION**********FIN**********

5.3.3.1.3 Pruebas operación "PASSW"

PASSW													
CLA + (6 últimos números de tarjeta)* + (Clave anterior**) + (Nueva Clave)	Sintaxis instrucción	Verifica autorización BIN	Verifica autorización de trans. para BIN	Verifica si suscr existe y si esta activa	Verifica clave actual	InsertaRegistroTransaccionalAntesSus	InsertaSuscripcionBancaCelular	ActualizaRegistroTransaccionalDesSus	FormatoSalida	VerificaResultadoOperacion	ResultadoEsperado	ResultadoObtenido	ResultadoPrueba
6 últimos números incorrectos	ok										err	err	ok
Clave anterior incorrecta	ok										err	err	ok

Clave nueva incorrecta	ok										err	err	ok
Bin no pertenece a institución		ok									err	err	ok
Transacción para red sms en estado inactiva			ok								err	err	ok
Envía suscripcion inexistente				Ok							err	err	ok
Envía clave incorrecta					ok						err	err	ok
Cambio Clave ok						ok	ok	ok	ok	ok	ok	ok	ok

Tabla 5.2 Resultado de pruebas de operación PASSW.

5.3.3.1.4 Log de soporte de pruebas "PASSW"

FECHA: 2007/06/13 HORA: 17491500 FRMCLIENTESERVICIOS CMDSENPASS_CLICK**********INICIO**********

FECHA: 2007/06/13 HORA: 17491500 FRMCLIENTESERVICIOS CMDSENPASS_CLICK INPUT: PASSW098171480SMSSMSPCDLANDY JDLANDY NNNNNNNvvvvvvvvvvvv12341232

FECHA: 2007/06/13 HORA: 17491600 FRMCLIENTESERVICIOS CMDSENPASS_CLICK OUTPUT: 001ERROR: NUMERO DE TARJETA DEBE TENER 16 VALORES NUMERICOS

FECHA: 2007/06/13 HORA: 17491600 FRMCLIENTESERVICIOS CMDSENPASS_CLICK**********FIN**********

FECHA: 2007/06/13 HORA: 17511900 FRMCLIENTESERVICIOS CMDSENPASS_CLICK**********INICIO**********

FECHA: 2007/06/13 HORA: 17511900 FRMCLIENTESERVICIOS CMDSENPASS_CLICK INPUT: PASSW098171480SMSSMSPCDLANDY JDLANDY NNNNNN0010080245 12341232

FECHA: 2007/06/13 HORA: 17512000 FRMCLIENTESERVICIOS CMDSENPASS_CLICK OUTPUT: 015ERROR: CLIENTE NO SUSCRITO AL SERVICIO BANCA CELULAR DE LA INSTITUCION XXX

FECHA: 2007/06/13 HORA : 17512000 FRMCLIENTESERVICIOS CMDSENPASS_CLICK**********FIN**********

FECHA: 2007/06/13 HORA: 17513200 FRMCLIENTESERVICIOS CMDSENPASS_CLICK**********INICIO**********

FECHA: 2007/06/13 HORA: 17513200 FRMCLIENTESERVICIOS CMDSENPASS_CLICK INPUT: PASSW098171480SMSSMSPCDLANDY JDLANDY NNNNNN0010080245 wwwwwww

FECHA: 2007/06/13 HORA: 17513200 FRMCLIENTESERVICIOS CMDSENPASS_CLICK OUTPUT: 018ERROR: NUMERO DE CLAVE CONTIENE VALORES NO NUMERICOS

FECHA: 2007/06/13 HORA: 17513200 FRMCLIENTESERVICIOS CMDSENPASS_CLICK**********FIN**********

FECHA: 2007/06/13 HORA: 17515400 FRMCLIENTESERVICIOS CMDSENPASS_CLICK**********INICIO**********

FECHA: 2007/06/13 HORA: 17515400 FRMCLIENTESERVICIOS CMDSENPASS_CLICK INPUT: PASSW098171480SMSSMSPCDLANDY JDLANDY NNNNNN0010080245 1111qqqq

FECHA: 2007/06/13 HORA: 17515400 FRMCLIENTESERVICIOS CMDSENPASS_CLICK OUTPUT: 031ERROR: NUMERO DE CLAVE NUEVA CONTIENE VALORES NO NUMERICOS

FECHA: 2007/06/13 HORA: 17515400 FRMCLIENTESERVICIOS CMDSENPASS_CLICK**********FIN**********

FECHA: 2007/06/13 HORA: 17520300 FRMCLIENTESERVICIOS CMDSENPASS_CLICK**********INICIO**********

FECHA: 2007/06/13 HORA: 17520300 FRMCLIENTESERVICIOS CMDSENPASS_CLICK INPUT: PASSW098171480SMSSMSPCDLANDY JDLANDY NNNNNN0010080245 11111111

FECHA: 2007/06/13 HORA: 17520300 FRMCLIENTESERVICIOS CMDSENPASS_CLICK OUTPUT: 015ERROR: CLIENTE NO SUSCRITO AL SERVICIO BANCA CELULAR DE LA INSTITUCION XXX

FECHA: 2007/06/13 HORA: 17520300 FRMCL IENTESERVICIOS CMDSENPASS_CLICK**********FIN**********

FECHA: 2007/06/13 HORA: 17523300 FRMCLIENTESERVICIOS CMDSENPASS_CLICK**********INICIO**********

FECHA: 2007/06/13 HORA: 17523300 FRMCLIENTESERVICIOS CMDSENPASS_CLICK INPUT: PASSW098171480SMSSMSPCDLANDY JDLANDY 2222220010080245 11111111

FECHA: 2007/06/13 HORA: 17523400 FRMCLIENTESERVICIOS CMDSENPASS_CLICK OUTPUT: 002ERROR: BIN NO REGISTRADO EN INSTITUCION XXX O RED APOYO

FECHA: 2007/06/13 HORA: 17523400 FRMCLIENTESERVICIOS CMDSENPASS_CLICK**********FIN**********

FECHA: 2007/06/13 HORA: 17534100 FRMCLIENTESERVICIOS CMDSENPASS_CLICK**********INICIO**********

FECHA: 2007/06/13 HORA: 17534100 FRMCLIENTESERVICIOS CMDSENPASS_CLICK INPUT: PASSW091897079SMSSMSPCDLANDY JDLANDY NNNNNN0010080245 XXXXXXXX

FECHA: 2007/06/13 HORA: 17534100 FRMCLIENTESERVICIOS CMDSENPASS_CLICK OUTPUT: 000

FECHA: 2007/06/13 HORA: 17534100 FRMCLIENTESERVICIOS CMDSENPASS_CLICK**********FIN**********

FECHA: 2007/06/13 HORA: 17535200 FRMCLIENTESERVICIOS CMDSENPASS_CLICK**********INICIO**********

FECHA: 2007/06/13 HORA: 17535200 FRMCLIENTESERVICIOS CMDSENPASS_CLICK INPUT: PASSW091897079SMSSMSPCDLANDY JDLANDY NNNNNN0010080245 XXXXXXXX

FECHA: 2007/06/13 HORA: 17535300 FRMCLIENTESERVICIOS CMDSENPASS_CLICK OUTPUT: 000

FECHA: 2007/06/13 HORA: 17535300 FRMCLIENTESERVICIOS CMDSENPASS_CLICK**********FIN**********

FECHA: 2007/06/13 HORA: 17552400 FRMCLIENTESERVICIOS CMDSENPASS_CLICK**********INICIO**********

FECHA: 2007/06/13 HORA: 17552400 FRMCLIENTESERVICIOS CMDSENPASS_CLICK INPUT: PASSW091897079SMSSMSPCDLANDY JDLANDY NNNNNN0010080245 4323XXXX

FECHA: 2007/06/13 HORA: 17552500 FRMCLIENTESERVICIOS CMDSENPASS_CLICK OUTPUT: 016ERROR: CLAVE DE SUSCRIPCION BANCACEL INCORRECTA. VERIFIQUE SU CLAVE Y VUELVA A INTENTAR FECHA: 2007/06/13 HORA: 17552500 FRMCLIENTESERVICIOS CMDSENPASS_CLICK**********FIN**********

5.3.3.1.5 Pruebas operación "SALDO"

SALDO													
S,SCC,SCA	Sintaxis instrucción	Verifica autorización BIN	Verifica autorización de trans. para BIN	Verifica si suscr existe y si esta activa	Verifica estado de suscripcion	InsertaRegistroTransaccionalAntesSus	InsertaSuscripcionBancaCelular	ActualizaRegistroTransaccionalDesSus	FormatoSalida	VerificaResultadoOperacion	ResultadoEsperado	ResultadoObtenido	ResultadoPrueba
Bin no pertenece a institución		ok									err	err	ok
Transaccion cambio clave para red sms en estado inactiva			ok								err	err	ok
Envía suscripcion				ok							err	err	ok

inactiva													
Envía tipo de cuenta inexistente					ok					ok	err	err	ok
Consulta el saldo						ok	ok	ok	ok	ok	ok	ok	ok

Tabla 5.3 Resultado de pruebas de operación SALDO.

5.3.3.1.6 Log de soporte de pruebas "SALDO"

FECHA: 2007/06/13 HORA: 18074600 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********INICIO**********
FECHA: 2007/06/13 HORA: 18074600 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK INPUT: SALDO098171480SMSSMS098171480 098171480 NNNNNN0010080245 0000CA123223
FECHA: 2007/06/13 HORA: 18074600 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK OUTPUT: 015ERROR: CLIENTE NO SUSCRITO AL SERVICIO BANCA CELULAR
FECHA: 2007/06/13 HORA: 18074600 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********FIN**********
FECHA: 2007/06/13 HORA: 18082300 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********INICIO**********
FECHA: 2007/06/13 HORA: 18082300 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK INPUT: SALDO091897080SMSSMS091897080 091897080 NNNNNN001008024d 0000CA123223
FECHA: 2007/06/13 HORA: 18082400 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK OUTPUT: 001ERROR: NUMERO DE TARJETA DEBE TENER 16 VALORES NUMERICOS
FECHA: 2007/06/13 HORA: 18082400 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********FIN**********
FECHA: 2007/06/13 HORA: 18082900 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********INICIO**********
FECHA: 2007/06/13 HORA: 18082900 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK INPUT: SALDO09189707dSMSSMS09189707d 09189707d NNNNNN001008024d 0000CA123223
FECHA: 2007/06/13 HORA: 18082900 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK OUTPUT: 068ERROR: NUMERO DE CELULAR CONTIENE VALORES NO NUMERICOS
FECHA: 2007/06/13 HORA: 18082900 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********FIN**********
FECHA: 2007/06/13 HORA: 18084100 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********INICIO**********
FECHA: 2007/06/13 HORA: 18084100 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK INPUT: SALDO091897080SMSSMS091897080 091897080 NNNNNN0010080245 0000CC123223
FECHA: 2007/06/13 HORA: 18084200 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK OUTPUT: 009TARJETA NO TIENE CUENTA ASOCIADA
FECHA: 2007/06/13 HORA: 18084200 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********FIN**********
FECHA: 2007/06/13 HORA: 18085600 FRMCLIENTESERVICIOS

CMDSALDOSEND_CLICK**********INICIO**********
FECHA: 2007/06/13 HORA: 18085600 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK INPUT: SALDO091897080SMSSMS091897080 091897080 NNNNNN0010080245 0000CA712323
FECHA: 2007/06/13 HORA: 18085700 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK OUTPUT: 000CUENTA AHORROS 712323 +000000000278+000000000000+000000000278CUENTA CORRIENTE +000000000000+000000000000+000000000000
FECHA: 2007/06/13 HORA: 18085700 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********FIN**********

5.3.3.1.7 Pruebas operación "CONCH"

CONCH													
CHQ + (6 últimos números de tarjeta)* + (Clave actual) + Numero de cheque	Sintaxis instrucción	Verifica autorización BIN	Verifica autorización de trans. para BIN	Verifica si suscr existe y si esta activa	VerifSiSusTieCtaCor	InsertaRegistroTransaccionalAntesSus	InsertaSuscripcionBancaCelular	ActualizaRegistroTransaccionalDesSus	FormatoSalida	VerificaResultadoOperacion	ResultadoEsperado	ResultadoObtenido	ResultadoPrueba
Num tarjeta incorrecta	ok										err	err	ok
Clave incorrecta	ok										err	err	ok
Numero de cheque	ok										err	err	ok
Bin no pertenece a institución		ok									err	err	ok

Transaccion para red sms en estado inactiva			ok								err	err	ok
Suscripcion no tiene asociada una cuenta corriente					ok						err	err	ok
Cheque no existe						ok	ok	ok	ok	ok	err	err	ok
Consulta estado cheque ok						ok	ok	ok	ok	ok	ok	ok	ok

Tabla 5.4 Resultado de pruebas de operación CONCH.

5.3.3.1.8 Log de soporte de pruebas "CONCH"

FECHA: 2007/06/13 HORA: 18271000 FRMCLIENTESERVICIOS CMDCHQSEND_CLICK**********INICIO**********

FECHA: 2007/06/13 HORA: 18271000 FRMCLIENTESERVICIOS CMDCHQSEND_CLICK INPUT: CONCH09189707NSMSSMS098171381 098171381 NNNNNN0010080245 XXXX124521

FECHA: 2007/06/13 HORA: 18271000 FRMCLIENTESERVICIOS CMDCHQSEND_CLICK OUTPUT: 068ERROR: NUMERO DE CELULAR CONTIENE VALORES NO NUMERICOS

FECHA: 2007/06/13 HORA: 18271000 FRMCLIENTESERVICIOS CMDCHQSEND_CLICK**********FIN**********

FECHA: 2007/06/13 HORA: 18272300 FRMCLIENTESERVICIOS CMDCHQSEND_CLICK**********INICIO**********

FECHA: 2007/06/13 HORA: 18272300 FRMCLIENTESERVICIOS CMDCHQSEND_CLICK INPUT: CONCH091897079SMSSMS098171381 098171381 NNNNNN001008024P XXXX124521

FECHA: 2007/06/13 HORA: 18272300 FRMCLIENTESERVICIOS CMDCHQSEND_CLICK OUTPUT: 001ERROR: NUMERO DE TARJETA DEBE TENER 16 VALORES NUMERICOS

FECHA: 2007/06/13 HORA: 18272300 FRMCLIENTESERVICIOS CMDCHQSEND_CLICK**********FIN**********

FECHA: 2007/06/13 HORA: 18273500 FRMCLIENTESERVICIOS CMDCHQSEND_CLICK**********INICIO**********

FECHA: 2007/06/13 HORA: 18273500 FRMCLIENTESERVICIOS CMDCHQSEND_CLICK INPUT: CONCH091897079SMSSMS098171381 098171381 NNNNNN0010080245 QQQQ124521

FECHA: 2007/06/13 HORA: 18273500 FRMCLIENTESERVICIOS CMDCHQSEND_CLICK OUTPUT: 018ERROR:

NUMERO DE CLAVE CONTIENE VALORES NO NUMERICOS

FECHA: 2007/06/13 HORA: 18273500 FRMCLIENTESERVICIOS CMDCHQSEND_CLICK**********FIN**********

FECHA: 2007/06/13 HORA: 18274300 FRMCLIENTESERVICIOS CMDCHQSEND_CLICK**********INICIO**********

FECHA: 2007/06/13 HORA: 18274300 FRMCLIENTESERVICIOS CMDCHQSEND_CLICK INPUT: CONCH091897079SMSSMS098171381 098171381 NNNNNN0010080245 1234124521

FECHA: 2007/06/13 HORA: 18274300 FRMCLIENTESERVICIOS CMDCHQSEND_CLICK OUTPUT: 016ERROR: CLAVE DE SUSCRIPCION BANCACEL INCORRECTA. VERIFIQUE SU CLAVE Y VUELVA A INTENTAR

FECHA: 2007/06/13 HORA: 18274300 FRMCLIENTESERVICIOS CMDCHQSEND_CLICK**********FIN**********

FECHA: 2007/06/13 HORA: 18275300 FRMCLIENTESERVICIOS CMDCHQSEND_CLICK**********INICIO**********

FECHA: 2007/06/13 HORA: 18275300 FRMCLIENTESERVICIOS CMDCHQSEND_CLICK INPUT: CONCH091897079SMSSMS098171381 098171381 NNNNNN0010080245 XXXX124521

FECHA: 2007/06/13 HORA: 18275300 FRMCLIENTESERV ICIOS CMDCHQSEND_CLICK OUTPUT: 085ERROR: TARJETA NO TIENE ASOCIADA CUENTA CORRIENTE

FECHA: 2007/06/13 HORA: 18275300 FRMCLIENTESERVICIOS CMDCHQSEND_CLICK**********FIN**********

FECHA: 2007/06/13 HORA: 18275700 FRMCLIENTESERVICIOS CMDCHQSEND_CLICK**********INICIO**********

FECHA: 2007/06/13 HORA: 18275700 FRMCLIENTESERVICIOS CMDCHQSEND_CLICK INPUT: CONCH091897079SMSSMS098171381 098171381 NNNNNN0010080245 XXXX124521

FECHA: 2007/06/13 HORA: 18275800 FRMCLIENTESERVICIOS CMDCHQSEND_CLICK OUTPUT: 040ERROR: EL CHEQUE NO EXISTE

FECHA: 2007/06/13 HORA: 18275800 FRMCLIENTESERVICIOS CMDCHQSEND_CLICK**********FIN**********

FECHA: 2007/06/13 HORA: 18275900 FRMCLIENTESERVICIOS CMDCHQSEND_CLICK**********INICIO**********

FECHA: 2007/06/13 HORA: 18275900 FRMCLIENTESERVICIOS CMDCHQSEND_CLICK INPUT: CONCH091897079SMSSMS098171381 098171381 NNNNNN0010080245 XXXX124521

FECHA: 2007/06/13 HORA: 18280000 FRMCLIENTESERVICIOS CMDCHQSEND_CLICK OUTPUT: 000

FECHA: 2007/06/13 HORA: 18280000 FRMCLIENTESERVICIOS CMDCHQSEND_CLICK**********FIN**********

5.3.3.1.9 Pruebas operación "SUSSA"

SUSSA													
ADI + (6 últimos números de tarjeta)* + (Clave actual) + AVCHC	Sintaxis instrucción	Verifica autorización BIN	Verifica autorización de trans. para BIN	Verifica si suscr existe y si esta activa	Verifica estatus de tarjeta	InsertaRegistroTransaccionalAntesSus	InsertaSuscripcionBancaCelular	ActualizaRegistroTransaccionalDesSus	FormatoSalida	VerificaResultadoOperacion	ResultadoEsperado	ResultadoObtenido	ResultadoPrueba
Num tarjeta incorrecta	ok										err	err	ok
Num clave incorrecta	ok										err	err	Ok
Codigo ser adicional incorrecto	ok										err	err	ok
Suscripcion incorrecta				ok							err	err	Ok
Tarjeta estatus incorrecto					ok						err	err	Ok
Suscripcion ok						ok	ok	ok	ok	ok	ok	ok	Ok

Tabla 5.5 Resultado de pruebas de operación SUSSA.

5.3.3.1.10 Log de soporte de pruebas "SUSSA"

FECHA: 2007/05/04 HORA: 11252700 FRMCLIENTESERVICIOS CMDSASEND_CLICK**********INICIO**********

FECHA: 2007/05/04 HORA: 11252700 FRMCLIENTESERVICIOS CMDSASEND_CLICK INPUT:

SUSSA091897079SMSSMSPCDLANDY JDLANDY NNNNNN0010080245 XXXXAVCHC
FECHA: 2007/05/04 HORA: 11252700 FRMCLIENTESERVICIOS CMDSASEND_CLICK OUTPUT: 0000102777091 TD
FECHA: 2007/05/04 HORA: 11252700 FRMCLIENTESERVICIOS CMDSASEND_CLICK**********FIN**********
FECHA: 2007/05/04 HORA: 11252800 FRMCLIENTESERVICIOS CMDSASEND_CLICK**********INICIO**********
FECHA: 2007/05/04 HORA: 11252800 FRMCLIENTESERVICIOS CMDSASEND_CLICK INPUT: SUSSA091897079SMSSMSPCDLANDY JDLANDY NNNNNN0010080245 XXXXAVCHC
FECHA: 2007/05/04 HORA: 11252900 FRMCLIENTESERVICIOS CMDSASEND_CLICK OUTPUT: 019ERROR: SERVICIO ADICIONAL BANCA CELULAR YA ESTA ACTIVO
FECHA: 2007/05/04 HORA: 11252900 FRMCLIENTESERVICIOS CMDSASEND_CLICK**********FIN**********
FECHA: 2007/05/04 HORA: 11253200 FRMCLIENTESERVICIOS CMDSASEND_CLICK**********INICIO**********
FECHA: 2007/05/04 HORA: 11253200 FRMCLIENTESERVICIOS CMDSASEND_CLICK INPUT: SUSSA091897079SMSSMSPCDLANDY JDLANDY NNNNNN0010080245 XXXXCONCH
FECHA: 2007/05/04 HORA: 11253200 FRMCLIENTESERVICIOS CMDSASEND_CLICK OUTPUT: 0000102777091 TD
FECHA: 2007/05/04 HORA: 11253200 FRMCLIENTESERVICIOS CMDSASEND_CLICK**********FIN**********
FECHA: 2007/05/04 HORA: 11253800 FRMCLIENTESERVICIOS CMDSASEND_CLICK**********INICIO**********
FECHA: 2007/05/04 HORA: 11253800 FRMCLIENTESERVICIOS CMDSASEND_CLICK INPUT: SUSSA091897079SMSSMSPCDLANDY JDLANDY NNNNNN0010080245 XXXXNOCTC
FECHA: 2007/05/04 HORA: 11253800 FRMCLIENTESERVICIOS CMDSASEND_CLICK OUTPUT: 0000102777091 TD
FECHA: 2007/05/04 HORA: 11253800 FRMCLIENTESERVICIOS CMDSASEND_CLICK**********FIN**********
FECHA: 2007/05/04 HORA: 11254100 FRMCLIENTESERVICIOS CMDSASEND_CLICK**********INICIO**********
FECHA: 2007/05/04 HORA: 11254100 FRMCLIENTESERVICIOS CMDSASEND_CLICK INPUT: SUSSA091897079SMSSMSPCDLANDY JDLANDY NNNNNN0010080245 XXXXNOCTC
FECHA: 2007/05/04 HORA: 11254100 FRMCLIENTESERVICIOS CMDSASEND_CLICK OUTPUT: 019ERROR: SERVICIO ADICIONAL BANCA CELULAR YA ESTA ACTIVO
FECHA: 2007/05/04 HORA: 11254100 FRMCLIENTESERVICIOS CMDSASEND_CLICK**********FIN**********
FECHA: 2007/05/04 HORA: 11465000 FRMCLIENTESERVICIOS CMDSASEND_CLICK**********INICIO**********
FECHA: 2007/05/04 HORA: 11465000 FRMCLIENTESERVICIOS CMDSASEND_CLICK INPUT: SUSSA091897079SMSSMSPCDLANDY JDLANDY NNNNNN0010080245 XXXXAVCHC
FECHA: 2007/05/04 HORA: 11465100 FRMCLIENTESERVICIOS CMDSASEND_CLICK OUTPUT: 0000102777091 TD
FECHA: 2007/05/04 HORA: 11465100 FRMCLIENTESERVICIOS CMDSASEND_CLICK**********FIN**********
FECHA: 2007/05/04 HORA: 11465100 FRMCLIENTESERVICIOS CMDSASEND_CLICK**********INICIO**********
FECHA: 2007/05/04 HORA: 11465100 FRMCLIENTESERVICIOS CMDSASEND_CLICK INPUT: SUSSA091897079SMSSMSPCDLANDY JDLANDY NNNNNN0010080245 XXXXAVCHC
FECHA: 2007/05/04 HORA: 11465200 FRMCLIENTESERVICIOS CMDSASEND_CLICK OUTPUT: 019ERROR:

SERVICIO ADICIONAL BANCA CELULAR YA ESTA ACTIVO
FECHA: 2007/05/04 HORA: 11465200 FRMCLIENTESERVICIOS CMDSASEND_CLICK**********FIN**********
FECHA: 2007/05/04 HORA: 11465400 FRMCLIENTESERVICIOS CMDSASEND_CLICK**********INICIO**********
FECHA: 2007/05/04 HORA: 11465400 FRMCLIENTESERVICIOS CMDSASEND_CLICK INPUT: SUSSA091897079SMSSMSPCDLANDY JDLANDY NNNNNN0010080245 XXXXCONCH
FECHA: 2007/05/04 HORA: 11465500 FRMCLIENTESERVICIOS CMDSASEND_CLICK OUTPUT: 0000102777091 TD
FECHA: 2007/05/04 HORA: 11465500 FRMCLIENTESERVICIOS CMDSASEND_CLICK**********FIN**********
FECHA: 2007/05/04 HORA: 11465800 FRMCLIENTESERVICIOS CMDSASEND_CLICK**********INICIO**********
FECHA: 2007/05/04 HORA: 11465800 FRMCLIENTESERVICIOS CMDSASEND_CLICK INPUT: SUSSA091897079SMSSMSPCDLANDY JDLANDY NNNNNN0010080245 XXXXCONCH
FECHA: 2007/05/04 HORA: 11465900 FRMCLIENTESERVICIOS CMDSASEND_CLICK OUTPUT: 019ERROR: SERVICIO ADICIONAL BANCA CELULAR YA ESTA ACTIVO
FECHA: 2007/05/04 HORA: 11465900 FRMCLIENTESERVICIOS CMDSASEND_CLICK**********FIN**********
FECHA: 2007/05/04 HORA: 11470400 FRMCLIENTESERVICIOS CMDSASEND_CLICK**********INICIO**********
FECHA: 2007/05/04 HORA: 11470400 FRMCLIENTESERVICIOS CMDSASEND_CLICK INPUT: SUSSA091897079SMSSMSPCDLANDY JDLANDY NNNNNN0010080245 XXXXNOCTC
FECHA: 2007/05/04 HORA: 11470400 FRMCLIENTESERVICIOS CMDSASEND_CLICK OUTPUT: 0000102777091 TD
FECHA: 2007/05/04 HORA: 11470400 FRMCLIENTESERVICIOS CMDSASEND_CL ICK**********FIN**********
FECHA: 2007/05/04 HORA: 11471000 FRMCLIENTESERVICIOS CMDSASEND_CLICK**********INICIO**********
FECHA: 2007/05/04 HORA: 11471000 FRMCLIENTESERVICIOS CMDSASEND_CLICK INPUT: SUSSA091897079SMSSMSPCDLANDY JDLANDY NNNNNN0010080245 XXXXNOCTC
FECHA: 2007/05/04 HORA: 11471100 FRMCLIENTESERVICIOS CMDSASEND_CLICK OUTPUT: 019ERROR: SERVICIO ADICIONAL BANCA CELULAR YA ESTA ACTIVO
FECHA: 2007/05/04 HORA: 11471100 FRMCLIENTESERVICIOS CMDSASEND_CLICK**********FIN**********

5.3.3.1.11 Pruebas operación "SALTC"

SALTC													
STC + (6 últimos números de tarjeta)* + (Clave actual)	Sintaxis instrucción	Verifica autorización BIN	Verifica autorización de trans. para BIN	Verifica si suscr existe y si esta activa	Verifica estatus de tarjeta	InsertaRegistroTransaccionalAntesSus	InsertaSuscripcionBancaCelular	ActualizaRegistroTransaccionalDesSus	FormatoSalida	VerificaResultadoOperacion	ResultadoEsperado	ResultadoObtenido	ResultadoPrueba
Num tarjeta incorrecta	ok										err	err	ok
Num clave incorrecta	ok										err	err	ok
Clave incorrecta	ok										err	err	ok
Suscripcion incorrecta				ok							err	err	ok
Tarjeta estatus incorrecto					ok						err	err	ok
Saldo	ok					ok	ok	ok	ok	ok	ok	ok	ok

Tabla 5.6 Resultado de pruebas de operación SALTC.

5.3.3.1.12 Log de soporte de pruebas “SALTC”

DATAINPUT: SALTC099163814SMSSMS099163814 099163814 NNNNNN2600424001 XXXXTO
DATAOUTPUT: 000TARJETA VISA
+000000018106+000000000000+000000018106+000000031894+000000000000+000000031894+000000000000
+0000000000002007/06/07
DATAINPUT: SALTC099163814SMSSMS099163814 099163814 NNNNNN2600424001 XXXXTO
DATAOUTPUT: 000TARJETA VISA
+000000053333+000000000000+000000053333+000000056667+000000000000+000000056667+000000000000
+0000000000002007/06/07
DATAINPUT: SALTC099163814SMSSMS099163814 099163814 NNNNNN2600424001 XXXXTO
DATAOUTPUT: 000TARJETA VISA
+000000012023+000000000000+000000012023+000000377977+000000000000+000000377977+000000000000
+0000000000002007/06/01
DATAINPUT: SALTC099163814SMSSMS099163814 099163814 NNNNNN2600424001 XXXXTO
DATAOUTPUT: 000TARJETA MASTERCARD
+000000065379+000000000000+000000065379+000000034621+000000000000+000000034621+000000000000
+0000000000002007/06/07

5.3.3.1.13 Pruebas operación “CUPTC”

CUPTC													
CTC + (6 últimos números de tarjeta)* + (Clave actual)	Sintaxis instrucción	Verifica autorización BIN	Verifica autorización de trans. para BIN	Verifica si suscr existe y si esta activa	Verifica estatus de tarjeta	InsertaRegistroTransaccionalAntesSus	InsertaSuscripcionBancaCelular	ActualizaRegistroTransaccionalDesSus	FormatoSalida	VerificaResultadoOperacion	ResultadoEsperado	ResultadoObtenido	ResultadoPrueba
Num tarjeta incorrecta	ok										err	err	ok
Num clave	ok										err	err	ok

incorrecta													
Clave incorrecta	ok										err	err	ok
Suscripcion incorrecta				ok							err	err	ok
Tarjeta estatus incorrecto					ok						err	err	ok
Saldo	ok					ok	ok	ok	ok	ok	ok	ok	ok

Tabla 5.7 Resultado de pruebas de operación CUPTC.

5.3.3.1.14 Log de soporte de pruebas "CUPTC"

DATAINPUT: CUPTC099163814SMSSMS099163814 099163814 NNNNNN2600424001 XXXXTO
DATAOUTPUT: 000TARJETA MASTERCARD NNNNNNNNNNNNNNNN +000000580000+000000588032-0000000080322007/06/05+000000580000+000000442156+0000001378442007/06/05
DATAINPUT: CUPTC099163814SMSSMS099163814 099163814 NNNNNN2600424001 XXXXTO
DATAOUTPUT: 000TARJETA VISA NNNNNNNNNNNNNNNN +000000100000+000000106244-0000000062442007/06/03+000000100000+000000000000+0000001000002007/06/03
DATAINPUT: CUPTC099163814SMSSMS099163814 099163814 NNNNNN2600424001 XXXXTO
DATAOUTPUT: 000TARJETA VISA NNNNNNNNNNNNNNNN +000000100000+000000104395-0000000043952007/06/09+000000100000+000000000000+0000001000002007/06/09
DATAINPUT: CUPTC099163814SMSSMS099163814 099163814 NNNNNN2600424001 XXXXTO
DATAOUTPUT: 000TARJETA VISA NNNNNNNNNNNNNNNN +000000150000+000000115454+0000000345462007/06/10+000000150000+000000000000+0000001500002007/06/10

5.3.3.2 Operaciones PUSH

Para las operaciones PUSH en todas sus variaciones se prepara un documento de planificación de pruebas en donde se llevará el control del resultado de las pruebas.

5.3.3.2.1 Pruebas operación "NOTON".

NOTON – Push Online sin suscripción													
Sub Operación NOTRE	Sintaxis instrucción	Verifica autorización BIN	Verifica autorización de trans. para BIN	Verifica si transa esta activa	VerificaNumeroCelular	InsertaRegistroTransaccionalAntesSus	InsertaSuscripcionBancaCelular	ActualizaRegistroTransaccionalDesSus	FormatoSalida	VerificaResultadoOperacion	ResultadoEsperado	ResultadoObtenido	ResultadoPrueba
Trama	ok										err	err	ok
Suboperación no registrada		ok									err	err	ok
Red origen no registrada		ok									err	err	ok
Bin especial no registrado			ok								err	err	ok
Transaccion no activa				ok							err	err	ok
Numero celular incorrecto					ok						err	err	ok
Envía mensaje SMS						ok	ok	ok	ok	ok	err	err	ok

Tabla 5.8 Resultado de pruebas de operación NOTON.

5.3.3.2.2 Log de soporte de pruebas "NOTON/NOTRE"

FECHA: 2007/06/14 HORA: 12594600 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********INICIO**********
FECHA: 2007/06/14 HORA: 12594600 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK INPUT: NOTON091897079SMSREM091897079 091897079 NNNNNN0010080245 0000NOTREMENSAJE DE NOTIFICACION DE LLEGADA DE UNA REMESA
JUAN DIEGO LANDY ANDRADE
SINFORMACION1
INFORMACION2
FECHA: 2007/06/14 HORA: 13045900 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK OUTPUT: 098ERROR: NO EXISTE TRANSACCION SOLICITADA O NO ESTA ACTIVA
FECHA: 2007/06/14 HORA: 13045900 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********FIN**********
FECHA: 2007/06/14 HORA: 13051300 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********INICIO**********
FECHA: 2007/06/14 HORA: 13051300 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK INPUT: NOTON091897079SMSREM091897079 091897079 9111110010080245 0000NOTREMENSAJE DE NOTIFICACION DE LLEGADA DE UNA REMESA
JUAN DIEGO LANDY ANDRADE
SINFORMACION1
INFORMACION2
FECHA: 2007/06/14 HORA: 13052100 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK OUTPUT: 000MENSAJE DE NOTIFICACION DE LLEGADA DE UNA REMESA

FECHA: 2007/06/14 HORA: 13052100 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********FIN**********
FECHA: 2007/06/14 HORA: 13172900 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********INICIO**********
FECHA: 2007/06/14 HORA: 13172900 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK INPUT: NOTON091897079SMSSMS091897079 091897079 NNNNNN0010080245 0000NOTREESTIMADO CLIENTE, EL PAGO MINIMO ES $2,233.34 Y EL TOTAL ES $5,467.34 LA
FECHA TOPE ES 09/31 DE SU VISA 4563585895475862
JUAN DIEGO LANDY ANDRADE
SINFORMACION1
INFORMACION2
FECHA: 2007/06/14 HORA: 13173300 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK OUTPUT: 098ERROR: NO EXISTE TRANSACCION SOLICITADA O NO ESTA ACTIVA
FECHA: 2007/06/14 HORA: 13173300 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********FIN**********

5.3.3.2.3 Pruebas operación "NOLSU".

NOLSU - Push Online/Batch con suscripción													
Sub Operación NOATM/NOCTC	Sintaxis instrucción	Verifica autorización BIN	Verifica autorización de trans. para BIN	VerificaSuscripcionPrincipal	VerificaSuscripcionAdicional	InsertaRegistroTransaccionalAntesSus	InsertaSuscripcionBancaCelular	ActualizaRegistroTransaccionalDesSus	FormatoSalida	VerificaResultadoOperacion	ResultadoEsperado	ResultadoObtenido	ResultadoPrueba
Trama	ok										err	err	ok
Bin incorrecto		ok									err	err	ok
Transaccion autorizada			ok								err	err	ok
Transaccion activa			ok								err	err	ok
Red origen no registrada			ok								err	err	ok
Suscripcion banca celular				ok							err	err	ok
Suscripcion servicio adicional					ok						err	err	ok
Transaccion no activa				ok							err	err	ok
Envía mensaje SMS						ok	ok	ok	ok	ok	err	err	ok

Tabla 5.9 Resultado de pruebas de operación NOLSU.

5.3.3.2.4 Log de soporte de pruebas “NOLSU” con servicio adicional “NOATM”

“NOLSU/NOATM”

FECHA: 2007/06/14 HORA: 13282000 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********INICIO**********

FECHA: 2007/06/14 HORA: 13282000 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK INPUT: NOLSU091223234SMSSMS091223234 091223234 NNNNNN0010080245 0000NOATMMENSAJE DE NOTIFICACION DE ATM JUAN DIEGO LANDY ANDRADE

SINFORMACION1

INFORMACION2

FECHA: 2007/06/14 HORA: 13282000 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK OUTPUT: 103SERVICIO BANCACELL NO DISPONIBLE POR EL MOMENTO. INTENTE EN UNOS MINUTOS

FECHA: 2007/06/14 HORA: 13282000 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********FIN**********

FECHA: 2007/06/14 HORA: 13310600 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********INICIO**********

FECHA: 2007/06/14 HORA: 13310600 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK INPUT: NOLSU091223234SMSSMS091223234 091223234 NNNNNN0010080245 0000NOATMMENSAJE ATM JUAN DIEGO LANDY ANDRADE

SINFORMACION1

INFORMACION2

FECHA: 2007/06/14 HORA: 13320400 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********INICIO**********

FECHA: 2007/06/14 HORA: 13320400 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK INPUT: NOLSU091223234SMSSMS091223234 091223234 NNNNNN0010080245 0000NOATMMENSAJE ATM JUAN DIEGO LANDY ANDRADE SINFORMACION1

INFORMACION2

FECHA: 2007/06/14 HORA: 13393200 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********INICIO**********

FECHA: 2007/06/14 HORA: 13393200 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK INPUT: NOLSU091223234SMSSMS091223234 091223234 NNNNNN0010080245 0000NOATMMENSAJE NOTIFICACION ATM JUAN DIEGO LANDY ANDRADE

SINFORMA CION1

INFORMACION2

FECHA: 2007/06/14 HORA: 13394900 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK OUTPUT: 016ERROR: CLAVE DE SUSCRIPCION BANCACEL INCORRECTA. VERIFIQUE SU CLAVE Y VUELVA A INTENTAR

FECHA: 2007/06/14 HORA: 13394900 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********FIN**********

FECHA: 2007/06/14 HORA: 13411700 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********INICIO**********

FECHA: 2007/06/14 HORA: 13411700 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK INPUT: NOLSU091223234SMSSMS091223234 091223234 NNNNNN0010080245 0000NOATMMENSAJE NOTIFICACION ATM

JUAN DIEGO LANDY ANDRADE
SINFORMACION1
INFORMACION2
FECHA: 2007/06/14 HORA: 13411900 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK OUTPUT: 023ERROR: SUSCRIPCION ESPECIAL NO ACTIVA. POR FAVOR SUCRIBASE NUEVAMENTE AL SERVICIO ESPECIAL DESEADO
FECHA: 2007/06/14 HORA: 13411900 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********FIN**********
FECHA: 2007/06/14 HORA: 13453300 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********INICIO**********
FECHA: 2007/06/14 HORA: 13453300 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK INPUT: NOLSU091223234SMSSMS091223234 091223234 NNNNNN0010080245 0000NOATMMENSAJE NOTIFICACION ATM
JUAN DIEGO LANDY ANDRADE
SINFORMACION1
INFORMACION2
FECHA: 2007/06/14 HORA: 13453500 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK OUTPUT: 023ERROR: CLIENTE NO SUSCRITO AL SERVICIO ADICIONAL. POR FAVOR SUSCRIBASE AL SERVICIO
FECHA: 2007/06/14 HORA: 13453500 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********FIN**********
FECHA: 2007/06/14 HORA: 13455400 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********INICIO**********
FECHA: 2007/06/14 HORA: 13455400 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK INPUT: NOLSU091223234SMSSMS091223234 091223234 NNNNNN0010080245 0000NOATMMENSAJE NOTIFICACION ATM
JUAN DIEGO LANDY ANDRADE
SINFORMACION1
INFORMACION2
FECHA: 2007/06/14 HORA: 13455700 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK OUTPUT: 023ERROR: CLIENTE NO SUSCRITO AL SERVICIO ADICIONAL. POR FAVOR SUSCRIBASE AL SERVICIO
FECHA: 2007/06/14 HORA: 13455700 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********FIN**********
FECHA: 2007/06/14 HORA: 13463300 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********INICIO**********
FECHA: 2007/06/14 HORA: 13463300 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK INPUT: NOLSU091223234SMSSMS091223234 091223234 NNNNNN0010080245 0000NOATMMENSAJE NOTIFICACION ATM
JUAN DIEGO LANDY ANDRADE
SINFORMACION1
INFORMACION2
FECHA: 2007/06/14 HORA: 13463500 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK OUTPUT: 000MENSAJE NOTIFICACION ATM

FECHA: 2007/06/14 HORA: 13463500 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********FIN**********

5.3.3.2.5 Log de soporte de pruebas “NOLSU” con servicio adicional “NOCTC”

FECHA: 2007/06/14 HORA: 13474200 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********INICIO**********
FECHA: 2007/06/14 HORA: 13474200 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK INPUT: NOLSU091223233SMSSMS091223233 091223233 NNNNNN0010080245 0000NOCTCMENSAJE NOTIFICACION DE CONSUMO CON TARJETA DE CREDITO JUAN DIEGO LANDY ANDRADE
SINFORMACION1
INFORMACION2
FECHA: 2007/06/14 HORA: 13474400 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK OUTPUT: 000MENSAJE NOTIFICACION DE CONSUMO CON TARJETA DE CREDITO
FECHA: 2007/06/14 HORA: 13474400 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********FIN**********

5.3.3.2.6 Pruebas operación “NOTBA”.

NOTBA - Push Batch sin suscripción													
Sub Operación BANCE	Sintaxis instrucción	Verifica autorización BIN	Verifica autorización de trans. para BIN	Verifica si transa esta activa	VerificaNumeroCelular	InsertaRegistroTransaccionalAntesSus	InsertaSuscripcionBancaCelular	ActualizaRegistroTransaccionalDesSus	FormatoSalida	VerificaResultadoOperacion	ResultadoEsperado	ResultadoObtenido	ResultadoPrueba
Trama	ok										err	Err	ok
Suboperación no registrada		ok									err	Err	ok
Red origen no registrada		ok									err	Err	ok
Bin especial no registrado			ok								err	Err	ok

Transaccion no activa				ok							err	Err	ok
Numero celular incorrecto					ok						err	Err	ok
Envía mensaje SMS						ok	ok	ok	ok	Ok	err	Err	ok

Tabla 5.10 Resultado de pruebas de operación NOTBA.

5.3.3.2.7 Log de soporte de pruebas "NOTBA/BANCE"

*FECHA: 2007/06/14 HORA: 13484700 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********INICIO**********
FECHA: 2007/06/14 HORA: 13484700 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK INPUT: NOTBA091897079SMSSMS091897079 091897079 9111110010080245 0000BANCEMENSAJE DE PROMOCIONES BANCACELL JUAN DIEGO LANDY ANDRADE SINFORMACION1
INFORMACION2
FECHA: 2007/06/14 HORA: 13484700 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK OUTPUT: 098ERROR: NO EXISTE TRANSACCION SOLICITADA O NO ESTA ACTIVA
FECHA: 2007/06/14 HORA: 13484700 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********FIN**********
FECHA: 2007/06/14 HORA: 13494400 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********INICIO**********
FECHA: 2007/06/14 HORA: 13494400 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK INPUT: NOTBA091897079SMSPPB091897079 091897079 9111110010080245 0000BANCEMENSAJE DE PROMOCIONES BANCACELL JUAN DIEGO LANDY ANDRADE SINFORMACION1
INFORMACION2
FECHA: 2007/06/14 HORA: 13494400 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK OUTPUT: 000MENSAJE DE PROMOCIONES BANCACELL
FECHA: 2007/06/14 HORA: 13494400 FRMCLIENTESERVICIOS CMDSALDOSEND_CLICK**********FIN**********</br>*

5.3.3.2.8 Pruebas configuración de operaciones PUSH Batch.

Para los mensajes PUSH Batch, cada operación debe ser previamente configurada diferentes parámetros. Un proceso automático que se ejecuta en el Server, se activa cada 10 minutos y lee la configuración de cada transaccion que tenga operación tipo PUSH BATCH sin suscripción. Si la operación esta autorizada para el envío de notificaciones SMS, se consulta la tabla TPB-PushBatch de las base de datos dbServiciosE que es el lugar en donde los sistemas interesados deben grabar los mensajes a enviar para que el proceso envíen los mensajes. A continuación se muestra los parámetros que deben configurarse por cada operación.

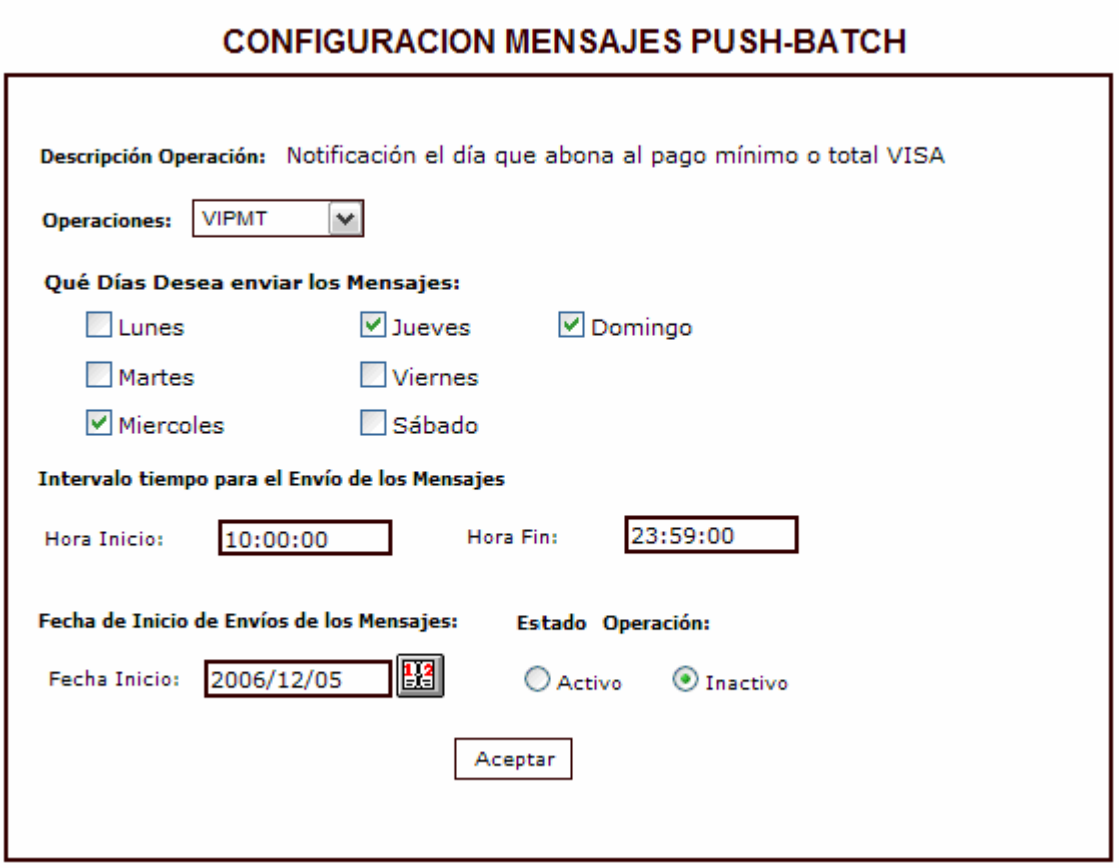


Figura 5.2 Configuración de mensajes Push batch sin suscripción

Programa de configuración de operaciones Push - Batch								
Pruebas	CargaOperacionesPushBatch	Días de envío	Intervalo de tiempo de envío de mensajes	Fecha de Inicio de notificación	Estado de la operación	ResultadoEsperado	ResultadoObtenido	ResultadoPrueba
Carga Operaciones	err					ok	err	err
Cambiar parámetros de configuración	ok	ok	ok	ok	ok	ok	ok	ok
Envío de mensajes por proceso automático						ok	ok	ok

Tabla 5.11 Resultado de pruebas de procesos automáticos - PUSH Batch sin suscripción.

5.3.3.3 Operaciones BACKOFFICE

Para las pruebas del BackOffice se ha preparado un set de pruebas. La tabla 5.12 muestra las pruebas realizadas y los resultados obtenidos. Se presentó un error al tratar de realizar el reverso de una suscripción. Este error fue depurado y corregido sin embargo en el documento se encuentra el resultado de la prueba obtenida.

Planificación y pruebas BackOffice																	
Prueba BackOffice	Módulo Seguridad	Módulo menú principal	Módulo Suscripción	Módulo Consulta Suscripciones	Módulo de cambio de teléfono	Módulo de consulta de servicios adicionales	Módulo de suscripción de servicio adicional	Módulo in activación de servicio adicional	Módulo in activación de servicio	Módulo Cambio Clave	Módulo ConsultaClave	Módulo Consulta Transacciones	Modulo Reversa Serv. Adicional	Modulo Rever susc. Principal	ResultadoEsperado	ResultadoObtenido	
Usuario Incorrecto	ok														ok	ok	ok
Password Incorrecto	ok														ok	ok	ok
Despliegue de menú		ok													ok	ok	ok
Suscripción tarjeta incorrecta			ok												err	err	ok
Suscripción ok			ok												ok	ok	ok
Consulta de suscripciones				ok											ok	ok	er
Verificar suscripcion anterior ok y estado actual					ok										ok	ok	ok
Ingresa clave de suscripción y nuevo teléfono celular					ok										ok	ok	ok
Consulta de servicios adicionales suscritos de la suscripcion ok anterior						ok									err	err	ok
Suscripción de un servicio adicional							ok								ok	ok	ok
Suscribe otro servicio adicional							ok								ok	ok	ok
Consulta servicios adicionales de una suscripcion						ok									ok	ok	ok
Inactiva servicio adicional								ok							ok	ok	ok
Consulta servicios adicionales de una suscripcion, verifica estatus						ok									ok	ok	ok
in activación de una suscripción principal y servicios adicionales															err	err	ok

Suscripción de un servicio adicional							ok								ok	ok	ok
Suscribe otro servicio adicional							ok								ok	ok	ok
Consulta servicios adicionales de una suscripcion						ok									ok	ok	ok
Inactiva servicio adicional								ok							ok	ok	ok
Inactiva suscripcion banca celular y servicios adicionales en caso de tener asociados									ok						ok	ok	ok
CambioClaveBancaCelular										ok					ok	ok	ok
Consulta clave											ok				ok	ok	ok
Consulta transacciones												ok			ok	ok	ok
Reversa servicio adicional													ok		ok	ok	ok
Reversa suscripcion principal														ok	ok	ok	er

Tabla 5.12 Resultado de pruebas de las transacciones del BACKOFFICE banca celular.

5.3.3.3.2 Prueba usuario incorrecto.

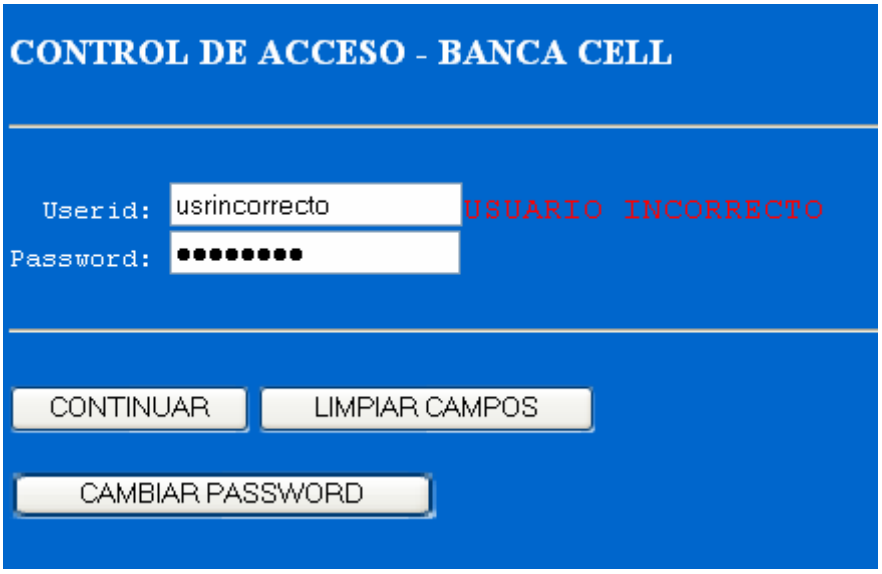


Figura 5.3 Módulo seguridad. Ingreso de usuario incorrecto.

5.3.3.3.3 Prueba password incorrecto.

Figura 5.4 Módulo seguridad. Ingreso de password incorrecto.

5.3.3.3.4 Prueba menú BackOffice de banca celular.

En caso de que el usuario y password sean validados correctamente por el módulo de seguridad se despliega este menú en donde se presentan todas las opciones que tiene el funcionario de atención al cliente como soporte al cliente.

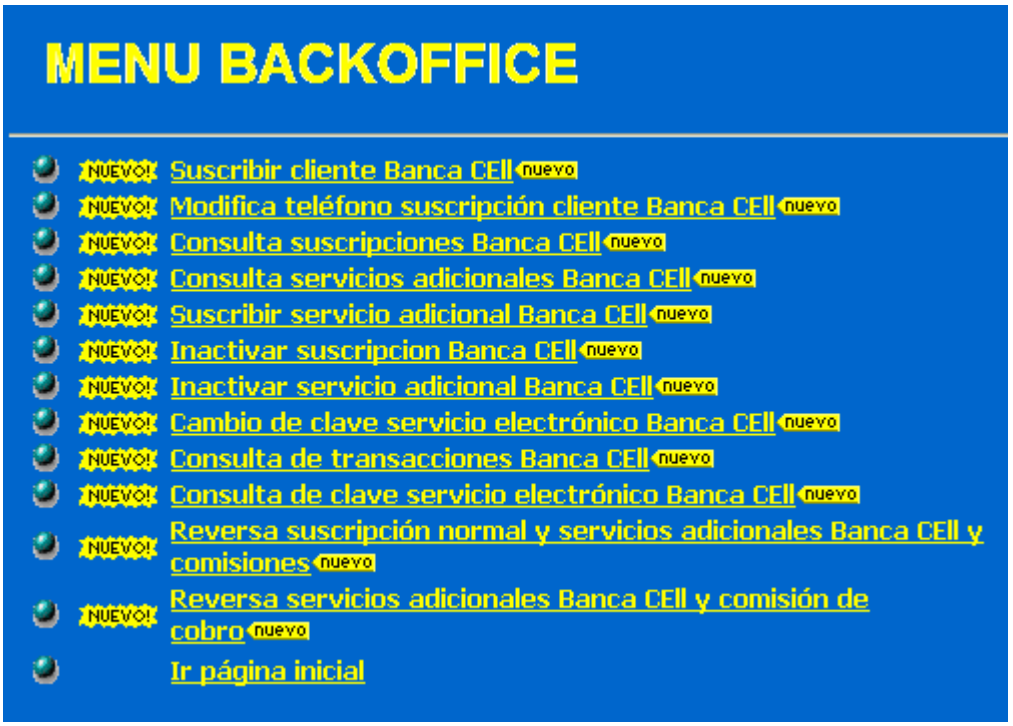


Figura 5.5 Módulo menú BackOffice. Opciones al usuario previa autentificación.

5.3.3.3.5 Prueba suscripción con tarjeta incorrecta.

En esta prueba el usuario trata de suscribirse al sistema con un número de tarjeta incorrecta.

Figura 5.6 Módulo suscripción. Tarjeta incorrecta.

5.3.3.3.6 Prueba suscripción banca celular exitosa.

En esta prueba el usuario ingresa un número de celular correcto, un número de tarjeta correcta y estatus activa y el PIN de la tarjeta correcta. El resultado es la suscripción correcta al servicio banca celular.

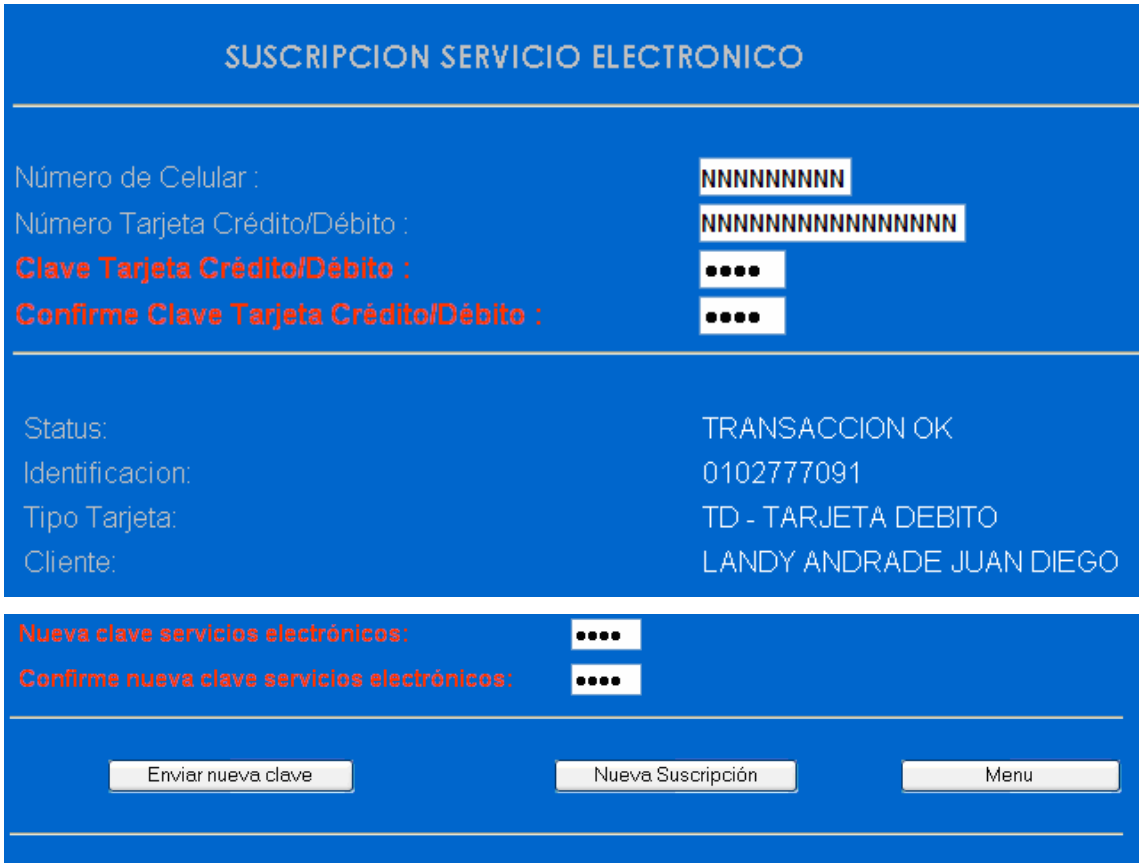


Figura 5.7 Módulo suscripción. Suscripción correcta.

5.3.3.3.7 Prueba consulta de suscripciones.

En esta prueba el usuario ingresa el número de tarjeta de crédito/débito y el rango de fechas cuando se realizo la suscripción. El resultado se presenta en la figura 5.8. Se encuentra la suscripción anterior.

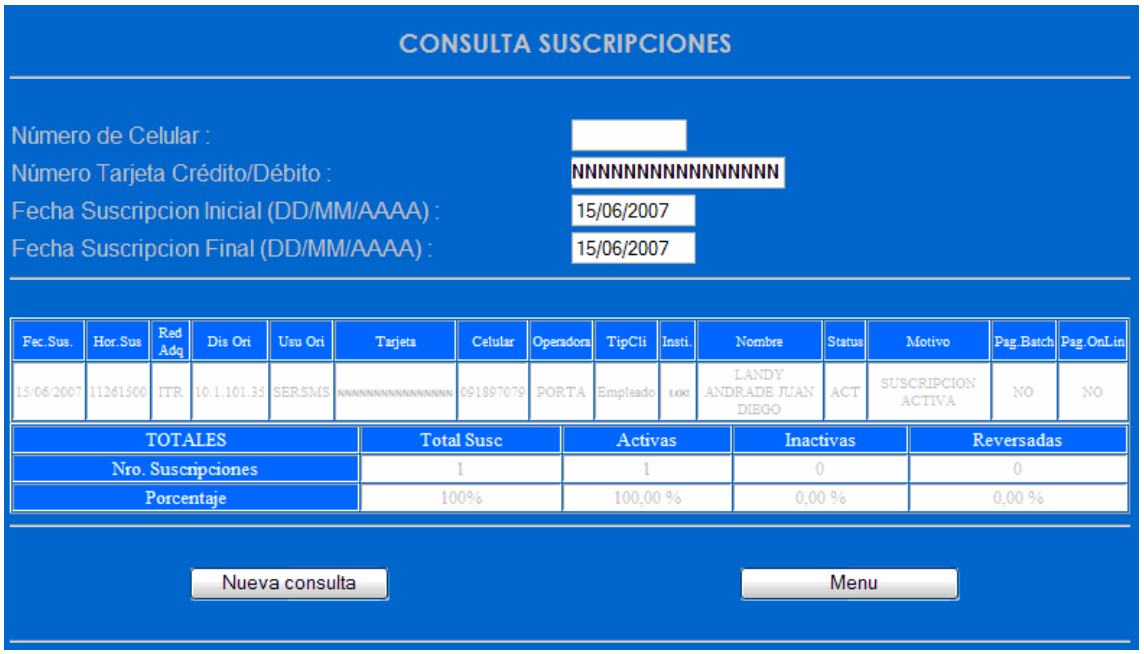


Figura 5.8 Módulo consulta de suscripciones.

5.3.3.3.8 Prueba modifica teléfono de suscripción.

En esta prueba el usuario hace un cambio de teléfono a la suscripción. Se solicita se ingrese la clave del servicio banca celular.

Figura 5.9 Módulo modificación de teléfono.

5.3.3.3.9 Prueba suscripción de servicio adicional.

El usuario asocia un servicio adicional a la suscripción principal, debe ingresar la clave del servicio banca celular. La prueba es correcta.

Figura 5.10 Módulo suscripción a un servicio adicional.

5.3.3.3.10 Prueba suscripción de otro servicio adicional.

Se agrega a la suscripción principal otro servicio adicional. En este caso la notificación de retiros de cajeros automáticos. La prueba es satisfactoria.

Figura 5.11 Módulo suscripción a un servicio adicional.

5.3.3.3.11 Prueba de suscripción de un servicio adicional ya suscrito anteriormente.

En esta prueba se trata de verificar que pasa si se suscribe un servicio adicional que ya tenia previamente el cliente suscrito. La prueba es satisfactoria.

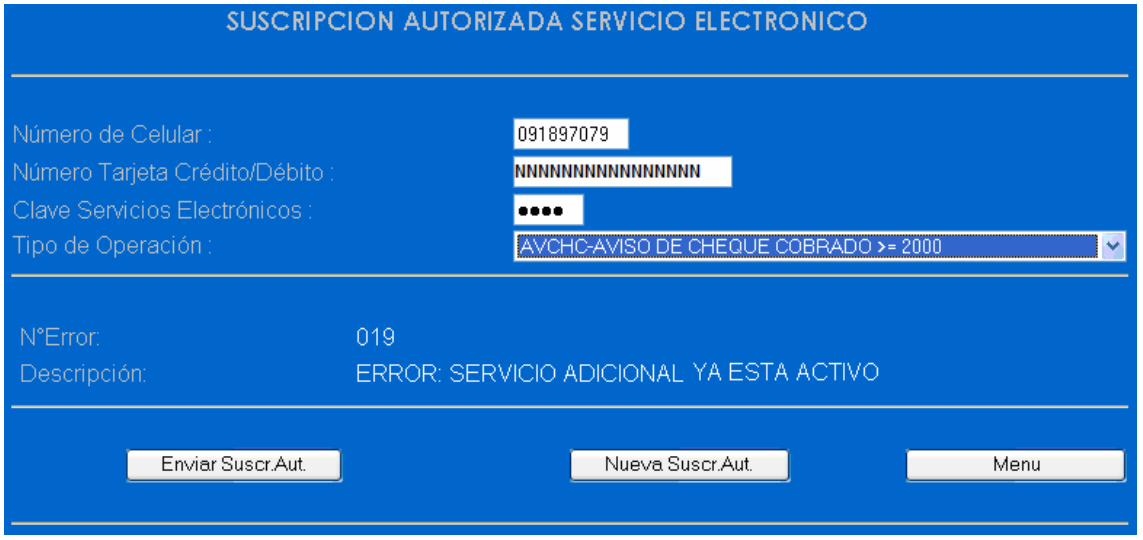


Figura 5.12 Módulo suscripción de un servicio adicional ya asociado previamente.

5.3.3.3.12 Prueba consulta de servicios adicionales asociados a una suscripción principal.

En esta prueba el usuario consulta los servicios adicionales suscritos a un cliente. La prueba es satisfactoria.

Figura 5.13 Módulo consulta de servicios adicionales.

5.3.3.3.13 Prueba inactiva servicio adicional.

Esta prueba inactiva un servicio adicional asociado a la suscripción principal del cliente. La prueba es satisfactoria.

Figura 5.14 Módulo consulta de servicios adicionales.

5.3.3.3.14 Prueba inactivación de suscripción del servicio banca celular.

En esta prueba se realiza la inactivación de una suscripción banca celular. En caso de que el cliente tenga suscripciones a servicios adicionales también los inactiva. La prueba se realiza con éxito.

Figura 5.15 Módulo in activación de suscripción principal banca celular.

5.3.3.3.15 Prueba cambio clave servicio banca celular.

En esta prueba se realiza el cambio de clave del servicio banca celular en caso de que el cliente así lo desee.

Figura 5.16 Módulo cambio de clave banca celular.

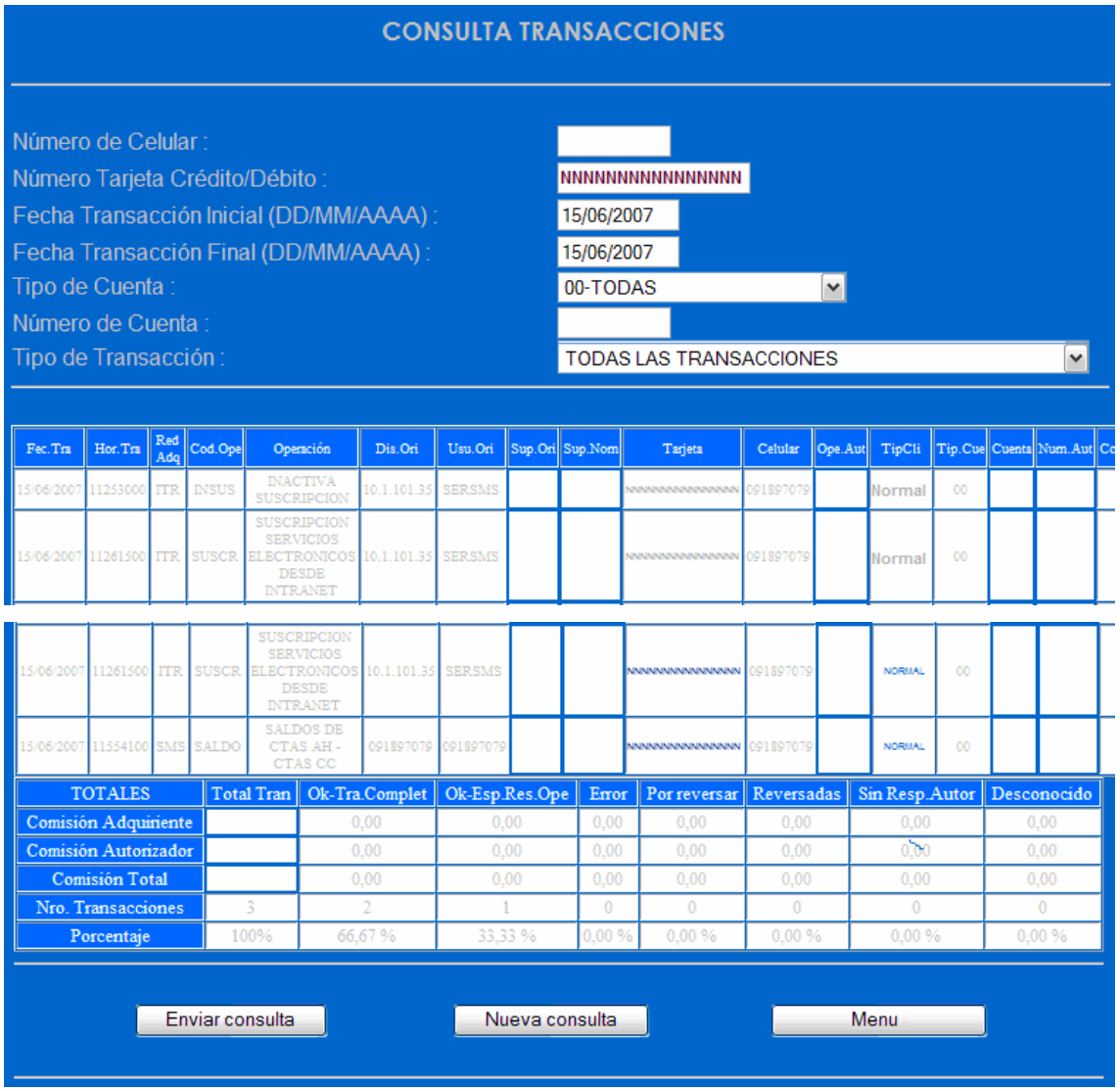
5.3.3.3.16 Prueba consulta clave servicio banca celular.

En esta prueba el usuario consulta la clave actual del servicio banca celular. A este modulo se ingresa con autentificación de un usuario supervisor.

Figura 5.17 Módulo consulta de clave banca celular.

5.3.3.3.17 ***Prueba consulta de transacciones banca celular.***

En esta transaccion se muestran la consulta de transacciones realizadas por un cliente. La información mostrada esta conforme con las transacciones realizadas en las pruebas y la base de datos. El resultado de la prueba es satisfactorio.



CONSULTA TRANSACCIONES

Número de Celular :
Número Tarjeta Crédito/Débito : NNNNNNNNNNNNNNNN
Fecha Transacción Inicial (DD/MM/AAAA) : 15/06/2007
Fecha Transacción Final (DD/MM/AAAA) : 15/06/2007
Tipo de Cuenta : 00-TODAS
Número de Cuenta :
Tipo de Transacción : TODAS LAS TRANSACCIONES

Fec.Tra	Hor.Tra	Red Adq	Cod.Ope	Operación	Dis.Ori	Usu.Ori	Sup.Ori	Sup.Nom	Tarjeta	Celular	Ope.Aut	TipCli	Tip.Cue	Cuenta	Num.Aut	Co
15/06/2007	11253000	ITR	INSUS	INACTIVA SUSCRIPCION	10.1.101.35	SERSMS			NNNNNNNNNNNNNNNN	091897079		Normal	00			
15/06/2007	11261500	ITR	SUSCR	SUSCRIPCION SERVICIOS ELECTRONICOS DESDE INTRANET	10.1.101.35	SERSMS			NNNNNNNNNNNNNNNN	091897079		Normal	00			
15/06/2007	11261500	ITR	SUSCR	SUSCRIPCION SERVICIOS ELECTRONICOS DESDE INTRANET	10.1.101.35	SERSMS			NNNNNNNNNNNNNNNN	091897079		NORMAL	00			
15/06/2007	11554100	SMS	SALDO	SALDOS DE CTAS AH - CTAS CC	091897079	091897079			NNNNNNNNNNNNNNNN	091897079		NORMAL	00			

TOTALES	Total Tran	Ok-Tra.Complet	Ok-Esp.Res.Ope	Error	Por reversar	Reversadas	Sin Resp.Autor	Desconocido
Comisión Adquiriente		0,00	0,00	0,00	0,00	0,00	0,00	0,00
Comisión Autorizador		0,00	0,00	0,00	0,00	0,00	0,00	0,00
Comisión Total		0,00	0,00	0,00	0,00	0,00	0,00	0,00
Nro. Transacciones	3	2	1	0	0	0	0	0
Porcentaje	100%	66,67 %	33,33 %	0,00 %	0,00 %	0,00 %	0,00 %	0,00 %

Enviar consulta | Nueva consulta | Menu

Figura 5.18 Módulo consulta de clave banca celular.

5.3.3.3.18 Prueba reverso de suscripcion adicional.

Esta opción el usuario debe utilizar en caso de que cometer un error al suscribir de manera errónea un servicio adicional, por ejemplo código de servicio adicional no deseado pero ya se debitó una comisión por la suscripción al servicio adicional. El usuario debe reversar el registro de la suscripción adicional incorrecta y acreditar la comisión cobrada, siempre y cuando se encuentre dentro de la ventana de tiempo de devolución de la comisión. Si el reverso ya esta fuera de fecha el usuario debe utilizar la opción de in activación del servicio adicional y se debe establecer otro

mecanismo de devolución del dinero.

Figura 5.19 Módulo reverso de un servicio adicional.

5.3.3.3.19 Prueba verificación de estatus de reverso luego de la in activación del servicio adicional.

En la figura 5.20 se muestra los estados de reversados de los servicios adicionales. La prueba es satisfactoria.

Figura 5.20 Módulo reverso de un servicio adicional.

5.3.3.3.20 Prueba reverso de suscripción banca celular.

Esta opción el usuario debe utilizar en caso de que cometer un error al suscribir de manera errónea a un cliente al servicio banca celular, por ejemplo el número celular ingresado incorrecto pero ya se debitó una comisión por la suscripción. El usuario debe reversar el registro de la suscripción principal y el sistema adicionalmente verifica si existieron suscripciones adicionales para reversarlos y acreditar la comisión cobrada por la suscripcion principal y adicionales realizadas, siempre y cuando se encuentre dentro de la ventana de tiempo de devolución de la comisión. Si el reverso ya esta fuera de fecha el usuario debe utilizar la opción de in activación del servicio banca celular y se debe establecer otro mecanismo de devolución del dinero.

5.3.4 *Pruebas de esfuerzo o estrés.*

Estas pruebas tienen como objetivo verificar como se comportan los objetos

cuando deben atender una carga importante de transaccionalidad.

Datos de prueba.

a) Objeto ServiciosElectronicos
b) 5000 transacciones,
c) 5 procesos automáticos disparando las transacciones desde 2 servidores.
d) Transacciones de entrada PULL, PUSH.

Resultado de prueba.

a) Los objetos ServiciosElectronicos y ServiciosOperacion presentan un comportamiento normal de procesamiento.

b) Todas las transacciones son contestadas, y sus respuestas son direccionadas de manera correcta al programa de test adquiriente.

c) Los tiempos de respuestas están en el orden de 50 ms (milisegundos).

d) Comportamiento del objeto de regla de negocio conforme con los datos de entrada.

e) Registro en base de datos OK. Los parámetros de entrada y salida son grabados correctamente.

f) Cambios de configuraciones de las transacciones Online, los resultados obtenidos están conformes con los resultados presentados.

5.3.5 *Pruebas de rendimiento reales.*

Las pruebas de rendimiento reales son conformes con los datos obtenidos en las pruebas antes del ingreso a producción y se adjunta en el anexo 5.1 parte del Log de las pruebas realizadas.

5.4 Implementación – Esquema tecnológico

En esta sección presentamos el esquema tecnológico a implementar en la institución financiera. La Fig. 5.23 se muestra todos los componentes que forman parte de la solución y que se debe implementar.

Figura 5.21 Esquema tecnológico a implementar.

A continuación describimos los diferentes elementos que necesitamos implementar para tener toda la infraestructura tecnológica a punto para instalar Lan solución banca celular.

5.4.2 *Infraestructura física.*

a) Comunicaciones con las operadoras celulares. Estos pueden ser enlaces VPN (Virtual Protocol Network) vía Internet o enlaces dedicados.
b) Firewall de comunicaciones para establecer los VPN de forma segura.
c) La institución debe solicitar al proveedor de Internet dirección IP pública para los enlaces VPN.
d) En la granja de servidores de la institución financiera deben correr el protocolo de comunicaciones TCP/IP. (TCP=Transfer Control Protocol; IP=Internet Protocol)
e) Servidor, en donde se ejecutaran las aplicaciones banca celular, pasarela y base de datos.

5.4.3 *Infraestructura lógica.*

a) Instalación en el servidor del sistema operativo Windows 2003 Server.
b) Instalación del IIS (Internet Information Server).
c) Instalación del servidor de componentes COM+. Normalmente este se instala con el IIS.
d) Instalación en el servidor del sistema SQL Server 2000 Enterprise Edition.
e) Instalación de pasarela. Es importante escoger correctamente la pasarela, sin embargo es importante mencionar que la pasarela debe soportar los protocolos smpp (short message Peer to Peer), en con soporte de la versión 3.3 o versión 3.4. La pasarela debe soportar la ejecución de smpp sobre TCP/IP ya que la información va a viajar a través de VPN en Internet. Normalmente las pasarelas son provistas por compañías autorizadas por las operadoras de un país y la selección de cual escoger esta en función del costo y servicio. Sin embargo para cualquier pasarela se debe configurar los parámetros del servidor de mensajería.

   Normalmente los siguientes son los parámetros que entrega la operadora celular y que deben ser ingresados en la pasarela.

   IP Server, dirección IP del servidor de mensajería de la operadora.
   Login, nombre del usuario entregado por la operadora celular.

Pass, clave de acceso al servidor de mensajería.

System Type, alias asignado por la operadora.

Source Address, es el número de shortcode asignado a su empresa por la operadora celular.

Puerto TX, puerto reservado para el establecimiento del socket para los mensajes transmitidos desde la operadora.

Puerto RX, puerto reservado para el establecimiento del socket para los mensajes recibidos por la operadora.

Source TON, Type of number, valor es 0x03 (hex), especifica la red.

Source NPI, (Number plan indicator), número indicador del plan, cuando no es conocido se debe setear 0x01, se recomienda 0x00.

Destination TON, Type of number, es configurado para internacional en 0x00.

Destination NPI, Number plan indicator, es configurado para "ISDN/E.164" con 0x00, se recomienda 0x01.

f) Instalación de base de datos dbServiciosE.

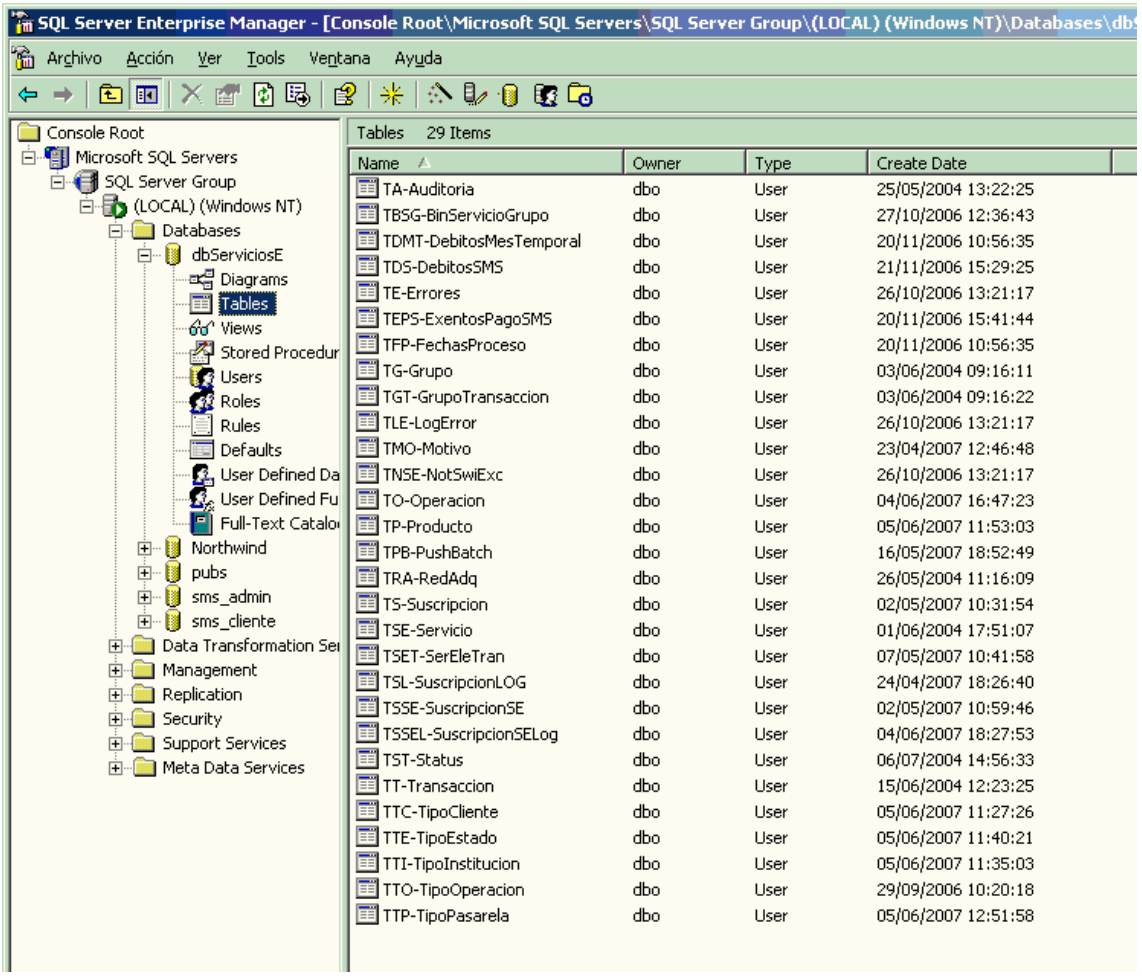


Figura 5.22 Base de datos dbServiciosE en SQLServer 2000 Enterprise Edition.

g) Registro e instalación de los objetos COM+ ServiciosElectrónicos.dll y ServiciosOperacion.dll

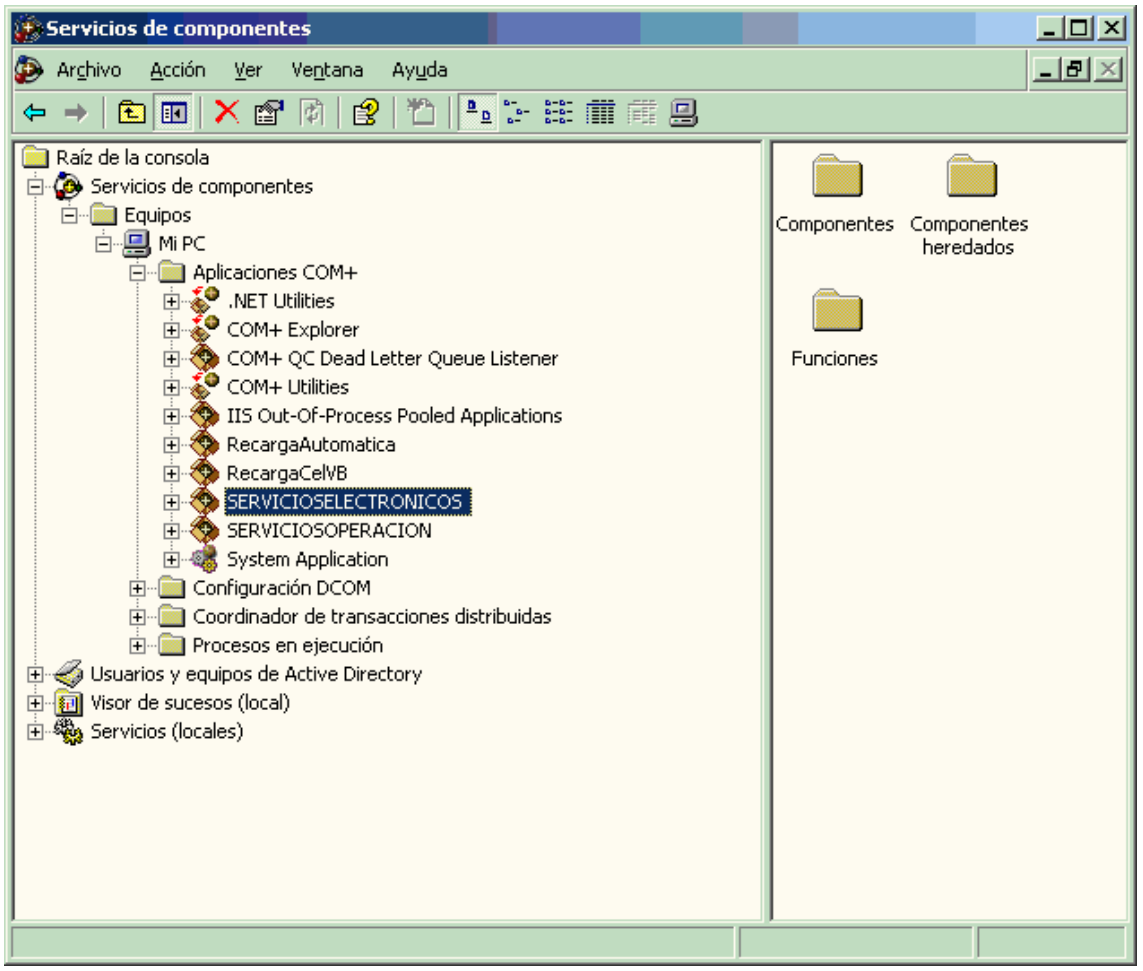


Figura 5.23 Servidor de componentes, el gráfico muestra los paquetes ServiciosElectronicos y ServiciosOperacion.

h) Instalación de la Intranet para el BackOffice.

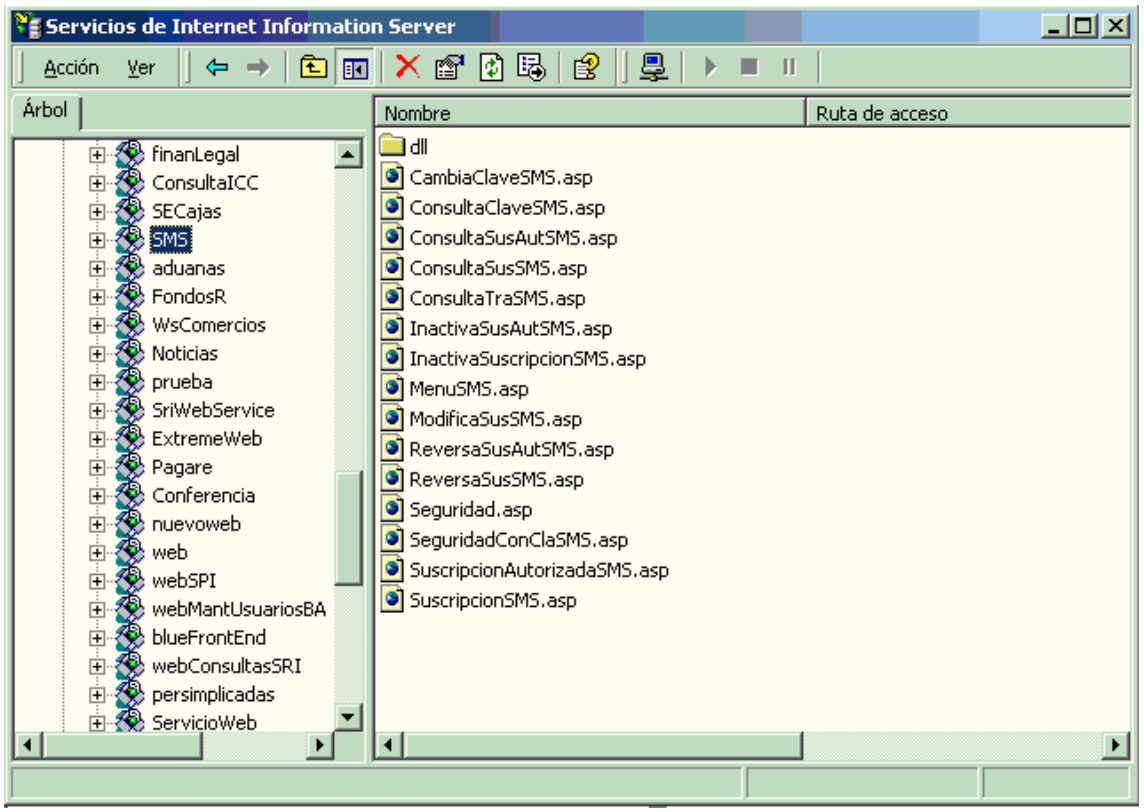


Figura 5.24 Directorio virtual SMS en donde apunta a las páginas ASP en el servidor IIS.

5.5 Conclusiones.

- Para conseguir una solución más adecuada, el desarrollo debe basarse en un modelado del análisis y diseño concienzudo antes de iniciar la codificación. La parte esencial de la aplicación, el modelo conceptual, las capas arquitectónicas, las principales asignaciones de responsabilidades, y las interacciones más importantes entre objetos se determinan de manera más satisfactoria en una investigación formal y en el proceso de diseño, antes que “apresurándose a codificar”.

- Una ventaja del análisis, diseño y programación orientada a objetos cuando se emplean en conjunto con una metodología como la que hemos propuesta en esta solución RUP, ofrece una guía completa de principio a fin, parea codificar la solución a partir de los requerimientos.

- Las tecnologías COM proporcionan todavía una gran ventaja para la reutilización de componentes y son muy eficientes cuando soportan alta carga transaccional.

- Las fases de pruebas, con lleva un sin numero de tareas tediosas y que generan gran cantidad de información. Sin embargo es muy importante que se realice una adecuada planificación de pruebas antes del ingreso a producción de cualquier solución informática. Una planificación adecuada consume tiempo y recursos, pero vale la pena minimizar los errores que se puedan presentar a los usuarios y a la empresa, sobre todo en aquellos en donde pueden presentarse perjuicios económicos a los actores involucrados.

5.6 Recomendaciones.

- Durante la programación podemos construir prototipos que ofrecen un excelente ambiente para examinar rápidamente métodos alternos y normalmente vale la pena dedicar poco o mucho tiempo al diseño por la codificación.

- Es conveniente que los diagramas generados en la fase de diseño sean actualizados de forma semiautomática, para que incluyan los cambios en las siguientes fases de codificación. Las herramientas CASE (Computer-aided software enginering, ingeniería de software asistida por computadora), realizan este trabajo, estas herramientas pueden leer el código fuente y producir automáticamente los diagramas de clases y de colaboración. Este es un aspecto de la ingeniera inversa, que consiste en generar modelos lógicos partiendo de un código fuente ejecutable. Pera nuestro caso particular no hemos trabajado con una herramienta CASE pero existe la herramienta Microsoft Visio 2003 for Enterprise Arquitects que permite hacer una reingeniería inversa de diagrama de clases y de base de datos.

- En cada iteración es recomendable pruebas para evitar la propagación de

errores en las próximas iteraciones como por ejemplo atributos mal definidos crean operaciones innecesarias.

- Es preferible que para la revisión de la consistencia del AOO los siguientes pasos deban ser realizados a lo largo de la evolución del AOO:

  a) Determinar si se necesitarían mas clases o si las responsabilidades están apropiadamente agrupadas entre clases

  b) Determinar si las responsabilidades muy invocadas pueden unirse y provocar una sola responsabilidad

- En las llamadas entre objetos COM, se debe tener un adecuado control de excepciones a nivel de creación del objeto y de llamada del método o función con esto se puede tomar medidas vía programación cuando existe una desincronización entre los objetos, especialmente cuando se encuentran en diferentes servidores, en donde uno de ellos puede ser reiniciado, causando la inestabilidad del servicio y si no están bien controlados este tipo de errores, los objetos no son automáticamente recuperables mientras no se descarguen de memoria desde la consola de servicios de componentes COM+.

CONCLUSIONES

- Al fabricar una solución informática de alta calidad, que satisfaga las demandas de los usuarios finales con una planificación y un presupuesto predecibles, es necesario establecer un modelo de proceso que contemple la asignación de tareas y responsabilidades de forma disciplinada dentro de la organización, garantizando con ello que la calidad esté presente en cada etapa del ciclo de vida del proyecto. Para ello, se ha apostado por implantar el Proceso Unificado de Rational o RUP (Rational Unified Process), adaptándolo a las necesidades de los clientes en conjunción con una serie de normas y estándares internacionales para el diseño, codificación y pruebas del software. El resultado es un proceso de fabricación con un desarrollo incremental por iteraciones dirigidas por el riesgo, y con un control continuo de la calidad. El cambio se acepta de manera natural y es gestionado adecuadamente beneficiando con ello el resultado del proyecto.

- La ingeniera de requisitos ha tomado gran importancia dentro de un proceso de implementación de un sistema de información sin embargo considero que el estándar IEEE 830 es muy amplio, y en este documento se ha tratado de aplicarlo de manera objetiva.

- La construcción de casos de uso a pesar de ser una técnica simple requiere de mucha intuición del levantador de requisitos, asumo que la experiencia que vaya acumulando la persona a cargo de esta responsabilidad ayudará a obtener información realmente valedera.

- Un modelo conceptual es una aproximación al dominio que intentamos entender. Podemos indicar que un buen modelo conceptual capta las abstracciones esenciales y la información indispensable para comprender el dominio dentro del contexto de los requerimientos.

- La identificación y ordenamiento de los eventos de un sistema para incluirlos en los diagramas de secuencia se logran de la revisión de los casos de uso.

- Un contrato puede emplearse con una operación de alto nivel que se aplica al sistema entero o con un método de bajo nivel de una clase particular.

- Los diagramas de estado describir el comportamiento de un objeto a través de diferentes casos de uso, sin embargo, no resultan del todo adecuados para describir el comportamiento que incluye a una serie de objetos colaborando entre sí. Por lo tanto, resulta útil combinar los diagramas de estado con otras técnicas. Por ejemplo, los diagramas de interacción son idóneos para la descripción del comportamiento de varios objetos en un único caso de uso, y los diagramas de actividades muestran de forma adecuada la secuencia general de acciones en diferentes objetos y casos de uso.

- Los diagramas de colaboración nos permiten decir más en un espacio, que los diagramas de secuencia y expresar además más información contextual; por ejemplo, el tipo de visibilidad entre objetos. También resulta más fácil expresar la lógica condicional y la concurrencia.

- La arquitectura de un sistema tomó mucha importancia en los últimos años como algo fundamental al diseñar una solución informática. Lo importante es comprender que para la arquitectura existen varias perspectivas y que dentro de estás podemos mostrar diferentes vistas dependiendo de los puntos de vista de los interesados en la misma y qué, al diseñar una solución debemos pensar en ellos y dejar plasmado en algún documento las decisiones arquitectónicas que tomamos.

- Se sugiere que al diseñar una clase se debe pensar en cómo podemos identificar un objeto real, como una persona, un transporte, un documento o un paquete. Durante el proceso del diseño de las clases tomamos las propiedades que identifican como único al objeto y otras propiedades adicionales como datos que corresponden al objeto.

- Los diagramas de clase son los más importantes del diseño orientado a objetos, son la piedra angular de nuestro diseño. Contienen toda la información de todas las clases y sus relaciones con otras clases.

- Al momento de hacer el primer diagrama de clases ya se tiene una lista de clases con algunos de sus atributos y operaciones. Sin embargo es necesario reflexionar y abstraer sobre la organización de esas clases estudiando las relaciones de herencia, agregación, etc.

- El diagrama de clases se ira refinando en las sucesivas iteraciones del modelo.

- Dentro de una arquitectura con división en capas, la que se ha utilizado en esta tesis es una de las posibles divisiones en capas, por lo que pueden existir otras un poco más completas e incluso una que defina mayor cantidad de capas. Esto depenedera del tipo de solución que se esté desarrollando.

- Para conseguir una solución más adecuada, el desarrollo debe basarse en un modelado del análisis y diseño concienzudo antes de iniciar la codificación. La parte esencial de la aplicación, el modelo conceptual, las capas arquitectónicas, las principales asignaciones de responsabilidades, y las interacciones más importantes entre objetos se determinan de manera más satisfactoria en una investigación formal y en el proceso de diseño, antes que “apresurándose a codificar”.

- La mayor ventaja del análisis, diseño y programación orientada a objetos cuando se emplean en conjunto con una metodología como la que hemos propuesta en esta solución RUP, ofrece una guía completa de principio a fin, parea codificar la solución a partir de los requerimientos.

- Los objetos COM proporcionan todavía una gran ventaja para la reutilización de componentes y son muy eficientes cuando soportan alta carga transaccional.

- Las fases de pruebas, con lleva un sin numero de tareas tediosas y que generan gran cantidad de información. Sin embargo es muy importante que se realice una adecuada planificación de pruebas antes del ingreso a producción de cualquier solución informática. Una planificación adecuada consume tiempo y recursos, pero vale la pena minimizar los errores que se puedan presentar a los usuarios y a la empresa, sobre todo en aquellos en donde pueden presentarse perjuicios económicos a los actores involucrados.

- Finalmente considero importante indicar que el objetivo general y los específicos planteados en este trabajo de graduación han sido cubiertos en su totalidad. Esta solución informática proporciona a las empresas del sector financiero del pais una plataforma robusta, flexible, etc. en donde se pueden generar un sin numero de transacciones para brindar diferentes tipos de servicios a los clientes, su expansión estará sujeta a las especificaciones funcionales de cada área gerencial, que a su vez puede contribuir con la generación de nuevos ingresos para la institución financiera.

RECOMENDACIONES

- El artefacto plan de desarrollo de software ayuda a aterrizar las ideas de los ejecutivos de la empresa en una solución informática real. Por este motivo es necesario destinar todo el tiempo y recursos necesarios para que el alcance del proyecto este apegado al presupuesto y eviten desfases en tiempo y recursos.

- El éxito de los casos de uso se suele justificar con el hecho de que constituyen una técnica simple e intuitiva, sin embargo es difícil la obtención y especificación de casos de uso verdaderamente útiles. A continuación me permito recomendar una técnica de Cockburn que utiliza el concepto de objetivo para organizar jerárquicamente los casos de uso. Distingue básicamente entre objetivos estratégicos (los procesos de negocio de la organización) y objetivos de usuario (las funciones del sistema). Los objetivos estratégicos se corresponden con un conjunto de objetivos de usuario y, de igual modo, un objetivo de usuario pude ser descompuesto, a su vez, en un conjunto de objetivos de usuario.

- Una técnica muy útil para identificar conceptos por su simplicidad propuesta en [Abbot83] consiste en identificar las frases nominales en las descripciones textuales del dominio de un problema y considerarlas conceptos o atributos idóneos. Este método hay que utilizarlos con mucha prudencia; no es posible encontrar mecánicamente correspondencias entre sustantivo y concepto, y además las palabras del lenguaje natural son ambiguas. Los casos expandidos de uso son una excelente descripción en donde puede aplicarse esta técnica.

- En todo análisis de un sistema, antes de iniciar el diseño lógico de cómo funcionará una aplicación de software, es necesario investigar y definir su comportamiento como una “caja negra”.

- No se debe diseñar diagramas de estados para todas las clases en el sistema, sino sólo para aquellas que exhiban un comportamiento interesante de forma que la elaboración del diagrama de estados nos ayude a entender dicho comportamiento.

- Al momento de diseñar una solución con metodología RUP se debe generar los artefactos que se consideren los que más sirvan al programador como fuente de documentación para el desarrollo, caso contrario se puede generar documentación ambigua que consuman tiempo y recursos que no se utiliza.

- Los diagramas de interacción constituyen uno de los artefactos más importantes que se generan en el diseño orientado a objetos. Para mejorar la calidad de su diseño se recomienda aplicar patrones, principios y expresiones codificadas.

- Se sugiere que al diseñar una clase se debe pensar en cómo podemos identificar un objeto real, como una persona, un transporte, un documento o un paquete. Durante el proceso del diseño de las clases tomamos las propiedades que identifican como único al objeto y otras propiedades adicionales como datos que corresponden al objeto.

- Durante la programación podemos construir prototipos que ofrecen un excelente ambiente para examinar rápidamente métodos alternos y normalmente vale la pena dedicar poco o mucho tiempo al diseño por la codificación.

- Es conveniente que los diagramas generados en la fase de diseño sean actualizados de forma semiautomática, para que incluyan los cambios en las siguientes fases de codificación. Las herramientas CASE (Computer-aided software enginering, ingeniería de software asistida por computadora),

realizan este trabajo, estas herramientas pueden leer el código fuente y producir automáticamente los diagramas de clases y de colaboración. Este es un aspecto de la ingeniera inversa, que consiste en generar modelos lógicos partiendo de un código fuente ejecutable. Pera nuestro caso particular no hemos trabajado con una herramienta CASE pero existe la herramienta Microsoft Visio 2003 for Enterprise Arquitects que permite hacer una reingeniería inversa de diagrama de clases y de base de datos.

- En cada iteración es recomendable pruebas para evitar la propagación de errores en las próximas iteraciones como por ejemplo atributos mal definidos crean operaciones innecesarias.

- En las llamadas entre objetos COM, se debe tener un adecuado control de excepciones a nivel de creación del objeto y de llamada del método o función con esto se puede tomar medidas vía programación cuando existe una desincronización entre los objetos, especialmente cuando se encuentran en diferentes servidores, en donde uno de ellos puede ser reiniciado, causando la inestabilidad del servicio y si no están bien controlados este tipo de errores, los objetos no son automáticamente recuperables mientras no se descarguen de memoria desde la consola de servicios de componentes COM+.

- Se recomienda el uso de este trabajo de investigación a todas las personas involucradas en el área de las TIC para que impulsemos juntos diferentes tipos de soluciones para el área de las telecomunicaciones celulares en nuestro país, si bien es cierto este trabajo enfoca un tipo de soluciones para el sector bancario, pero con creatividad se pueden enmarcar este tipo de solución a los diferentes sectores productivos del pais. Recordemos que el nivel de penetración de la telefonía celular en el país alcanza el indices altísimos, siendo en la actualidad el telefóno celular el dispositivo principal con el que un usuario puede interactuar con la empresa y viceversa.

- Además, recomiendo esta solución informática y específicamente el presente documento como material de referencia para implementar

soluciones empresariales con el uso de este tipo de tecnologías en todas las disciplinas relacionadas con el área de las TIC, de manera que se motive al estudiante el emprendimiento de nuevos proyectos relacionados con el tema.

- Finalmente, se sugiere este tipo de solución como modelo a seguir, para implementaciones futuras en cualquier empresa del sector financiero por su adaptabilidad con el mínimo esfuerzo a los diferentes escenarios tecnológicos utilizados en este tipo de empresas.

GLOSARIO

TÉRMINO	CATEGORÍA	COMENTARIOS
BIN	definición	Bank Identification Number, son los primeros seis dígitos de una tarjeta financiera. Participan en este esquema las tarjetas de debito, crédito, prepago, recarga. Los BIN se utilizan para identificar a los banco emisores y son el mecanismo de encaminamiento principal para las demandas por autorizaciones en tiempo real. Para mayor información vea el ISO 7812.
BIN	tipo	Seis primeros dígitos de una tarjeta de debito/crédito
BitacoraTransaccional	tipo	Registra las transacciones realizadas en el sistema BancaCelular
BitacoraTransaccional: CodOpe:string	atributo	Código de operación que requirió el cliente
BitacoraTransaccional: CodSer:string	atributo	Código del servicio Banca Celular
BitacoraTransaccional:CodError:Integer	atributo	Código de error en caso de presentarse
BitacoraTransaccional:CodRedAdq:string	atributo	Código de la red adquiriente desde donde se disparo la transacción
BitacoraTransaccional:FechaTran:Date	atributo	Fecha de la transacción
BitacoraTransaccional:HoraTran:Date	atributo	Hora de la transacción
BitacoraTransaccional:Mensaje:string	atributo	Mensaje enviado al cliente
BitacoraTransaccional:NumCel:string	atributo	Numero de celular a donde se envía el mensaje
BitacoraTransaccional:NumTar:Integer	atributo	Numero de tarjeta de 16 dígitos numéricos, suscrita
BitacoraTransaccional:NumTra:string	atributo	Código de la transacción relacionada con la

		operación solicitada por el cliente
CASE	Definición	CASE - Computer-aided software enginering, ingeniería de software asistida por computadora
Caso de uso	definición	Es un documento narrativo que describe la secuencia de eventos de un actor (agente externo) que utiliza un sistema para completar un proceso.[Jacobson92]
Catalogo de transacciones	tipo	Grupo de transacciones que forman un catalogo y que están asignados aun bin respectivo
Celular	Tipo	Dispositivo celular del cliente
Celular: NumCelular:string	atributo	Numero celular entregado por la operadora
Cliente	Tipo	Cliente de la institución financiera que realiza transacciones al sistema banca celular
Ejecutivo atención al cliente	tipo	Funcionario de la institución financiera que tiene contacto directo con el cliente
ERS	abreviatura	Para esta etapa se utilizan las recomendaciones de la IEEE según su estándar 830-1993. Dichas recomendaciones describen el proceso de creación de la especificación de requisitos software.
Especificación transacción	tipo	definición de una transacción banca celular
Especificación transacción: Descripcion:String	atributo	Nombre asignada a la transacción
Especificación transacción: NumTra:Integer	atributo	Numero de transacción asignada
Gestión BackOffice del producto Banca Electrónica	caso de uso	Descripción del proceso del soporte de un funcionario de atención al cliente del servicio Banca Celular

Gestión de administración del sistema de Banca Electrónica	caso de uso	Descripción de un proceso del administrador del sistema Banca Celular
Gestión de adquiriencia de transacciones tipo PULL	caso de uso	Descripción del proceso de envío de un mensaje SMS solicitando una operación bancaria
Gestión de generación de transacciones tipo PUSH BATCH con suscripción	caso de uso	Descripción del proceso de envío de un mensaje SMS de aviso o notificaciones desde un proceso batch a un cliente que tiene suscripción al servicio de Banca Celular
Gestión de generación de transacciones tipo PUSH BATCH sin suscripción	caso de uso	Descripción del proceso de envío de un mensaje SMS de aviso o notificaciones desde un proceso batch a un cliente que no tiene suscripción al servicio de Banca Celular
Gestión de generación de transacciones tipo PUSH ONLINE con suscripción	caso de uso	Descripción del proceso de envío de un mensaje SMS de aviso o notificación a un cliente que tiene suscripción al servicio de Banca Celular
Gestión de generación de transacciones tipo PUSH ONLINE sin suscripción	caso de uso	Descripción del proceso de envío de un mensaje SMS de aviso o notificación a un cliente que no tiene suscripción al servicio de Banca Celular
Gestión de suscripción de clientes.	caso de uso	Descripción del proceso de suscripción al servicio Banca Celular desde el teléfono del cliente o desde el funcionario de atención al cliente.
Gestión de suscripción de un servicio adicional.	caso de uso	Descripción del proceso de suscripción a un servicio adicional del Banca Celular desde el teléfono del cliente o desde el funcionario de atención al cliente.
IEEE	abreviatura	Instituto de ingenieros eléctricos y electrónicos
IP	abreviatura	Protocolo Internet

MensajesSMS	tipo	envía las transacciones a las operadoras celulares
MensajesSMS: NumCelular:string	atributo	Numero celular destino
MensajesSMS:mensaje	atributo	Mensaje a enviar
Operación PULL	caso de uso	Cuando un usuario inicia una transacción desde su celular y espera como resultado un mensaje de texto de la institución financiera.
Operación PUSH-Batch	caso de uso	Cuando un sistema automático procesa y envía notificaciones o avisos a los celulares pero no son generados por eventos en línea y son alimentados normalmente por archivos.
Operación PUSH-Online	caso de uso	Cuando un sistema automático genera un evento en línea y envía un mensaje SMS al celular de un usuario.
OperacionBancaCelular	tipo	definición de las operaciones que soporta el servicio banca celular
OperacionBancaCelular:CodOpe:string	atributo	Código de operación
OperacionBancaCelular:Descripcion:string	atributo	Nombre de la operación
OperacionBancaCelular:TipOpe:string	atributo	Tipo de operación, pull, push.
PagoOnline	tipo	Realiza el cobro de las comisiones online
PagoOnline:Monto:decimal	atributo	Valor a cobrar al cliente por transacción
PagoOnline:TipPago:string	atributo	Tipo de pago con tarjeta de debito/crédito
Pasarela SMS	Definición	La pasarela de mensajeria SMS/IP trata de ser como un servidor WEB, realiza sus tareas de cambio de formato y ajusted del mensaje,pero precisa de contenidos que realmente le den utilidad.
POS	definición	POS) point-of-sale terminal (POS) Punto de venta, dispositivo electrónico que permite

		leer tarjetas de debito/crédito y conecta con el banco adquiriente para obtener una autorización de compra
Receptor transacción	Tipo	Recibe las transacciones originadas por los diferentes usuarios del sistema, autoriza y autentifica la transacción en función de la operación solicitada por el cliente
Receptor Transaccion:CodOpe:string	atributo	Código de operación solicitada por los diferentes usuarios del sistema
Receptor Transaccion:Date	atributo	Hora de recepción de la transacción
Receptor Transaccion:Fecha:Date	atributo	Fecha de recepción de la transacción
RedAdquiriente	tipo	Redes autorizadas que pueden enviar transacciones al sistema banca celular
RedAdquiriente:CodRedAdq:string	tipo	Código de red adquiriente: SMS - Red celular, ITR – Intranet
RedAdquiriente:Descripcion:string	tipo	Nombre de la red adquiriente
RUP	abreviatura	RUP - Rational Unified Process. Se trata de una metodología para describir el proceso de desarrollo de software.
Servicio	tipo	Servicio autorizado - Banca Celular
Servicio:CodSer:string	atributo	Código de servicio
Servicio:Descripcion:string	atributo	Nombre del servicio
Sistema Batch	tipo	sistema que envía notificaciones en proceso o lotes
Sistema Online	tipo	Sistema receptor de eventos generados por el cliente sobre sus cuentas
SMPP	abreviatura	Short Messaage peer-to-peer protocol

SMS	abreviatura	SMS - Short Message Service o Servicio de Mensajes, es un servicio disponible en los teléfonos móviles que permite el envío de mensajes cortos de hasta 160 caracteres (también conocidos como mensajes de texto, o más coloquialmente, textos o incluso textos o mensajes) entre teléfonos móviles, teléfonos fijos y otros dispositivos de mano.
Suscripción	Tipo	Registro de las suscripciones al servicio Banca celular
Suscripcion:IdeCliente:string	atributo	Numero de identificación del cliente asociado a la tarjeta
Suscripcion:TipIdeCliente:string	atributo	Tipo de identificación del cliente asociado a la tarjeta
Tarjeta	Tipo	Tarjeta de debito/crédito asociada al cliente de la institución financiera que pueden disparar transacciones al sistema
Tarjeta:Numtarjeta:string	atributo	Numero de tarjeta de 16 dígitos numéricos
Tarjeta:TipTarjeta	atributo	Tipo de tarjeta debito/crédito
TCP/IP	Definición	Conjunto básico de protocolos de comunicación de redes, popularizado por Internet, que permiten la transmisión de información en redes de computadoras. El nombre TCP/IP proviene de dos protocolos importantes de la familia, el Transmission Control Protocol (TCP) y el Internet Protocol (IP)
TIC	Definición	Tecnologías de la información y comunicación, son las tecnologías que se necesitan para la gestión de transformación de la

		información.
VPN	Definición	Virtual Private Network, en castellano significa red privada virtual, es una tecnología de red que permite una extensión de la red local sobre una red pública o no controlada, como por ejemplo el Internet.

BIBLIOGRAFIA

1. Booch Grady, Rumbaugh James, Jacobson, “El proceso unificado de desarrollo de software”, Addison Wesley, 2000.

2. Booch Grady, Rumbaugh James, Jacobson, “El lenguaje unificado de modelado”, Addison Wesley, 2000.

3. Larman, Craig, "UML Y PATRONES”, Introducción al análisis y diseño orientado a objetos, Prentice Hall, 2003.

4. Arlow Jim, Neustadt Ila, “UML AND THE UNIFIED PROCESS”, Practical Object-OPriented Analisi & Design, Addison-Wesley, 2002.

5. Rumbaugh James, Blaha Michael, Premerlani William, Eddy Frederick, Lorensen William, “OBJECT-ORIENTED Modeling and Design”, Prenticel Hall, 1991.

6. Proceso de Desarrollo - Tutorial de Desarrollo Orientado a Objetos http://www.clikear.com/manuales/uml/procesodesarrollo.asp
7. Pasarelas de mensajería SMS. http://es.tldp.org/Presentaciones/200103hispalinux/seco/pdf/pasarelas-sms.pdf

8. Pasarelas de mensajería SMS. http://andressh.net/download/PasarelasSMS/pasarelas-sms.html

9. Modelado de Sistemas com UML http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/index.html

10. UML – Diagramas de colaboración. http://www.creangel.com/uml/home.php

11. Especificaciones de los requisitos de software http://www.cali.gov.co/publico2/documentos/informatica/IEEE830_ext.pdf

12. Arquitectura de software
http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art110.asp

13. Banca electrónica. El futuro
http://ciberhabitat.gob.mx/banco/elfuturo.htm

14. Especificaciones técnicas SMPP 1.0
http://www.quios.com/docs/SMPP_interface_1.0_tech_spec.pdf

15. Curso Manual Tutorial – Telefonía celular digital
http://www.conocimientosweb.net/dcmt/ficha15159.html

16. Recursos de ASPFácil – Objetos COM
http://www.aspfacil.com/recursos/default.asp?cat=9

ANEXOS

ANEXO 1.1 ANALISIS FINANCIERO

ANEXO 1.1 (a) ANALISIS FINANCIERO

PARAMETROS DE ENTRADA		
Comisión por 20 mensajes PULL	1.50	usd
Comisión por cada mensaje adicional	0.08	usd
Mensaje adicional	0	
Costo mensaje por pasarela SMS	0.02	usd
Mensajes enviados por suscripcion	5	
Mensajes PUSH	0.00%	
Costo mensaje por sistema banca celular	0.01	usd
Costo mensual de internet	200.00	usd
Pago mensual recurso humano	100.00	usd
Gastos mantenimiento operativo	30.00	usd

					MENSAJES ENVIADOS EFECTIVOS						
MES	Numero de suscripciones activas	Ingreso por suscriptor	Ingreso por mensaje adicional	Total ingresos mes	PULL	PULL ADI	TOTAL PULL	PUSH	Total mensajes enviados pasarela	Costo Mensual por pasarela	Costo mensual sistema banca celular
Jul-07	0	0.00	0	0.00	0	0	0	0	0	0	0
Ago-07	0	0.00	0	0.00	0	0	0	0	0	0	0
Sep-07	0	0.00	0	0.00	0	0	0	0	0	0	0
Oct-07	1000	1500.00	0	1500.00	5000	0	5000	0	5000	100	50
Nov-07	2000	3000.00	0	3000.00	10000	0	10000	0	10000	200	100
Dic-07	3000	4500.00	0	4500.00	15000	0	15000	0	15000	300	150
Ene-08	4000	6000.00	0	6000.00	20000	0	20000	0	20000	400	200
Feb-08	5000	7500.00	0	7500.00	25000	0	25000	0	25000	500	250
Mar-08	6000	9000.00	0	9000.00	30000	0	30000	0	30000	600	300
Abr-08	7000	10500.00	0	10500.00	35000	0	35000	0	35000	700	350
May-08	8000	12000.00	0	12000.00	40000	0	40000	0	40000	800	400
Jun-08	9000	13500.00	0	13500.00	45000	0	45000	0	45000	900	450

SISTEMA BANCA CELULAR												
FLUJO DE FONDOS Julio 2007 - Junio 2008												
	Jul-07	Ago-07	Sep-07	Oct-07	Nov-07	Dic-07	Ene-08	Feb-08	Mar-08	Abr-08	May-08	Jun-08
Ingreso por comisión de servicios	0.00	0.00	0.00	1500.00	3000.00	4500.00	6000.00	7500.00	9000.00	10500.00	12000.00	13500.00
Gastos x internet 256 kbytes sin comprension 1:1	200.00	200.00	200.00	200.00	200.00	200.00	200.00	200.00	200.00	200.00	200.00	200.00
Gastos x mantenimiento	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00
Costo de arrendamiento pasarela	0.00	0.00	0.00	100.00	200.00	300.00	400.00	500.00	600.00	700.00	800.00	900.00
Costo de arrendamiento sistema banca celular	0.00	0.00	0.00	50.00	100.00	150.00	200.00	250.00	300.00	350.00	400.00	450.00
Pago de Sueldos Recurso Humano Técnico	500.00	500.00	0.00									
Pago de Sueldos Recurso Humano	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Compra de Server	2032.00											
Compra de Equipo Auxiliar - Linux	500.00											
Compra de licencia Windows 2003 Server	750.00											
Compra de licencia de SQL Server 2000 Enterprise Edition	800.00											
Inversión	18000.00											
Flujo de fondos neto	-13088.00	-830.00	-330.00	1020.00	2370.00	3720.00	5070.00	6420.00	7770.00	9120.00	10470.00	11820.00

Los primeros dos meses no existen ingresos porque se encuentra el sistema en implementación en la institución
El tercer mes gratis por lanzamiento del producto

TASA DE INTERES ANUAL =	20%
TASA DE INTERES MENSUAL =	1.67%
VAN =	S/. 35,505.08
TIR=	19%

CONCLUSION

Proyecto es viable debido a que la TIR alcanza un porcentaje 19%

Actualmente el sistema bancario del pais paga tasas en depositos o inversiones que fluctúan entre el 4 y el 6 %

ANEXO 1.1 (b) PUNTO DE EQUILIBRIO

SISTEMA BANCA CELULAR												
ANALISIS DE COSTOS DEL PROYECTO Julio 2007 - Junio 2008												
	Jul-07	**Ago-07**	**Sep-07**	**Oct-07**	**Nov-07**	**Dic-07**	**Ene-08**	**Feb-08**	**Mar-08**	**Abr-08**	**May-08**	**Jun-08**
Gastos x internet 256 kbytes sin comprension 1:1	200.00	200.00	200.00	200.00	200.00	200.00	200.00	200.00	200.00	200.00	200.00	200.00
Gastos x mantenimiento	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00
Costo de arrendamiento pasarela	0.00	0.00	0.00	100.00	200.00	300.00	400.00	500.00	600.00	700.00	800.00	900.00
Costo de arrendamiento sistema banca celular	0.00	0.00	0.00	50.00	100.00	150.00	200.00	250.00	300.00	350.00	400.00	450.00
Pago de Sueldos Recurso Humano Técnico	500.00	500.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Pago de Sueldos Recurso Humano	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Compra de Server	2032.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Compra de Equipo Auxiliar - Linux	500.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Compra de licencia Windows 2003 Server	750.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Compra de licencia de SQL Server 2000 Enterprise Ed	800.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Total Gastos x Mes	4912.00	830.00	330.00	480.00	630.00	780.00	930.00	1080.00	1230.00	1380.00	1530.00	1680.00

Inversión de accionistas	18000.00	U.S.D.		
Costo Total Proyecto :	15792.00	U.S.D.		
Bancos	2208.00	U.S.D.	Activo de la empresa	
Costo de suscripción activa	1.50	U.S.D.		
Cobros por comision de servicio	10528		Utilidad:	0 U.S.D.

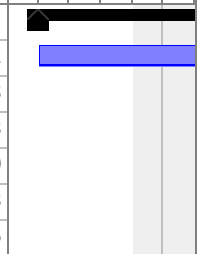
Punto de Equilibrio

CONCLUSION

Indica que en el año se deben realizar al menos 10528
cobros por comisión de servicio banca celular, debemos recordar que basta que un cliente realice una transaccion banca celular para cobrar por el servicio

ANEXO 1.2 CRONOGRAMA DE ACTIVIDADES

Id		Nombre de tarea	Duración	Calendario de tareas	% completado	Area que ejecuta la Tarea	Inversión Prevista	Inversion Real	Comienzo
1		**FASE DE PLANEACION Y DE ELABORACION**	**26 días**	**Ninguno**	**69%**		**0.00**	**0.00**	**mié 06/03/01**
2		Definir el plan preliminar	5 días	Ninguno	0%		0.00	0.00	mié 06/03/01
3		Elaborar el informe preliminar de investigación	3 días	Ninguno	0%		0.00	0.00	mié 06/03/08
4	✓	Definir los requerimientos	5 días	Ninguno	100%		0.00	0.00	lun 06/03/13
5	✓	Definir los casos de uso	10 días	Ninguno	100%		0.00	0.00	lun 06/03/20
6	✓	Registrar los términos en el glosario	2 días	Ninguno	100%		0.00	0.00	lun 06/04/03
7	✓	Perfeccionar el plan	1 día	Ninguno	100%		0.00	0.00	mié 06/04/05
8	✓	**FASE DE CONSTRUCCION - Iteración 1**	**312 días**	**Ninguno**	**100%**		**0.00**	**0.00**	**jue 06/04/06**
9	✓	**Fase de Análisis**	**73 días**	**Ninguno**	**100%**		**0.00**	**0.00**	**jue 06/04/06**
10	✓	Definir los casos de uso esenciales	30 días	Ninguno	100%		0.00	0.00	jue 06/04/06
11	✓	Perfeccionar los diagramas de casos de uso	4 días	Ninguno	100%		0.00	0.00	vie 06/05/19
12	✓	Perfeccionar el modelo conceptual	5 días	Ninguno	100%		0.00	0.00	vie 06/05/26
13	✓	Perfeccionar el glosario	10 días	Ninguno	100%		0.00	0.00	vie 06/06/02
14	✓	Definir los diagramas de secuencia de los sistemas	10 días	Ninguno	100%		0.00	0.00	vie 06/06/16
15	✓	Definir los contratos de operaciones	10 días	Ninguno	100%		0.00	0.00	vie 06/06/30
16	✓	Definir los diagramas de estados	4 días	Ninguno	100%		0.00	0.00	vie 06/07/14
17	✓	**Fase de Diseño**	**79 días**	**Ninguno**	**100%**		**0.00**	**0.00**	**jue 06/07/20**
18	✓	Definir los casos reales de uso	30 días	Ninguno	100%		0.00	0.00	jue 06/07/20
19	✓	Definir los reportes, la interfaz de usuario del usuar	5 días	Ninguno	100%		0.00	0.00	vie 06/09/01
20	✓	Perfeccionar la arquitectura del sistema	6 días	Ninguno	100%		0.00	0.00	vie 06/09/08
21	✓	Definir los diagramas de interacción	8 días	Ninguno	100%		0.00	0.00	lun 06/09/18
22	✓	Definir los diagramas de clases de diseño	15 días	Ninguno	100%		0.00	0.00	jue 06/09/28
23	✓	Definir el esquema de la base de datos	15 días	Ninguno	100%		0.00	0.00	jue 06/10/19
24	✓	**Fase de implementación y pruebas**	**160 días**	**Ninguno**	**100%**		**0.00**	**0.00**	**lun 06/11/13**
25	✓	Programación de sistemas	85 días	Ninguno	100%		0.00	0.00	lun 06/11/13



Proyecto: TesisCronogramaActivi
Fecha: mié 07/08/01

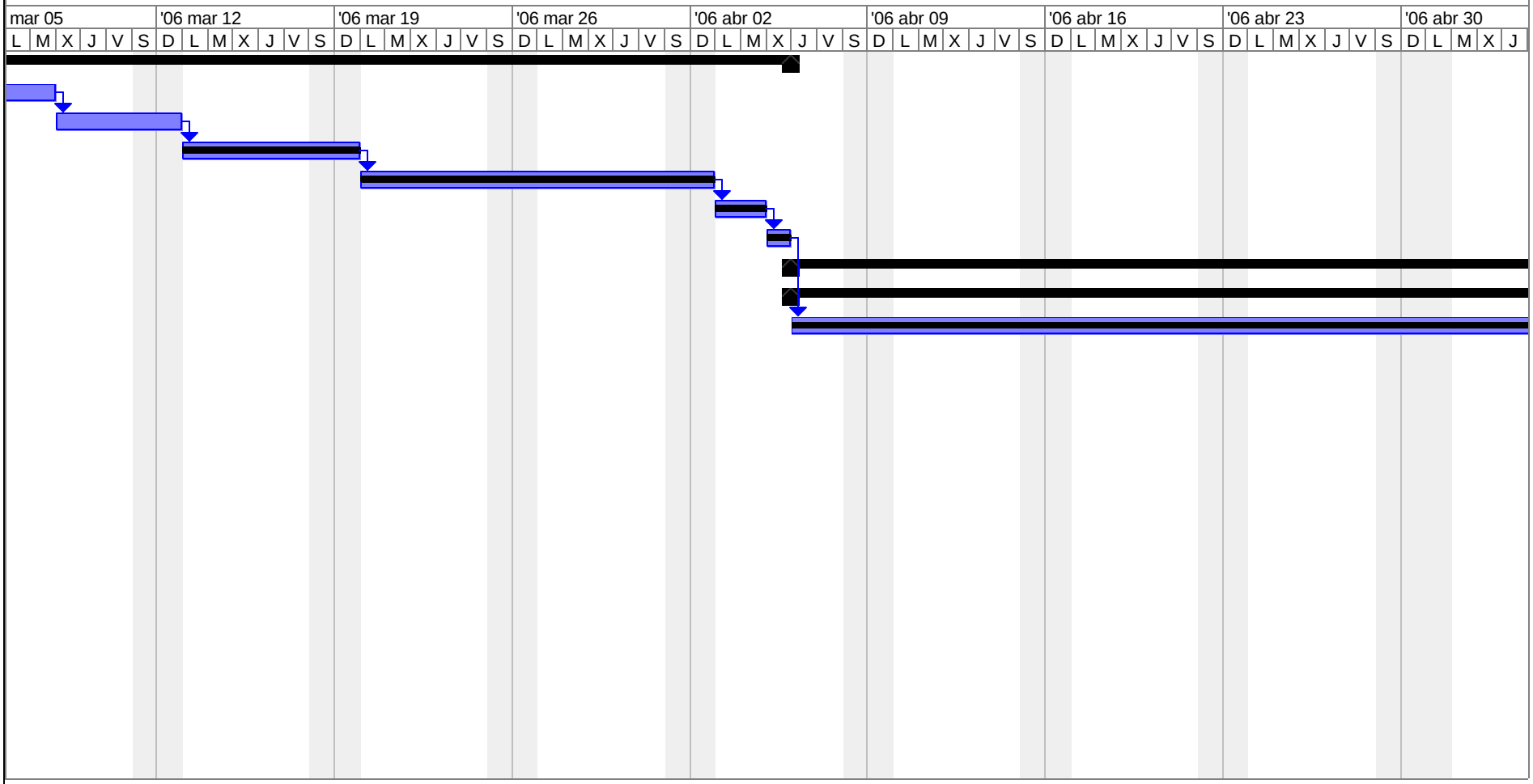
Tarea		Hito		Tareas externas	
División		Resumen		Hito externo	
Progreso		Resumen del proyecto		Fecha límite	

Id		Nombre de tarea	Duración	Calendario de tareas	% completado	Area que ejecuta la Tarea	Inversión Prevista	Inversion Real	Comienzo
26		Pruebas de unidades	15 días	Ninguno	100%		0.00	0.00	mié 07/03/14
27		Planificacion de pruebas	10 días	Ninguno	100%		0.00	0.00	mié 07/04/04
28		Pruebas de AOO	10 días	Ninguno	100%		0.00	0.00	mié 07/04/18
29		Pruebas de DOO	10 días	Ninguno	100%		0.00	0.00	jue 07/05/03
30		Pruebas de validación	20 días	Ninguno	100%		0.00	0.00	jue 07/05/17
31		Pruebas de estrés	5 días	Ninguno	100%		0.00	0.00	jue 07/06/14
32		Pruebas de rendimiento en producción	5 días	Ninguno	100%		0.00	0.00	jue 07/06/21

26 '06
M X J V S D

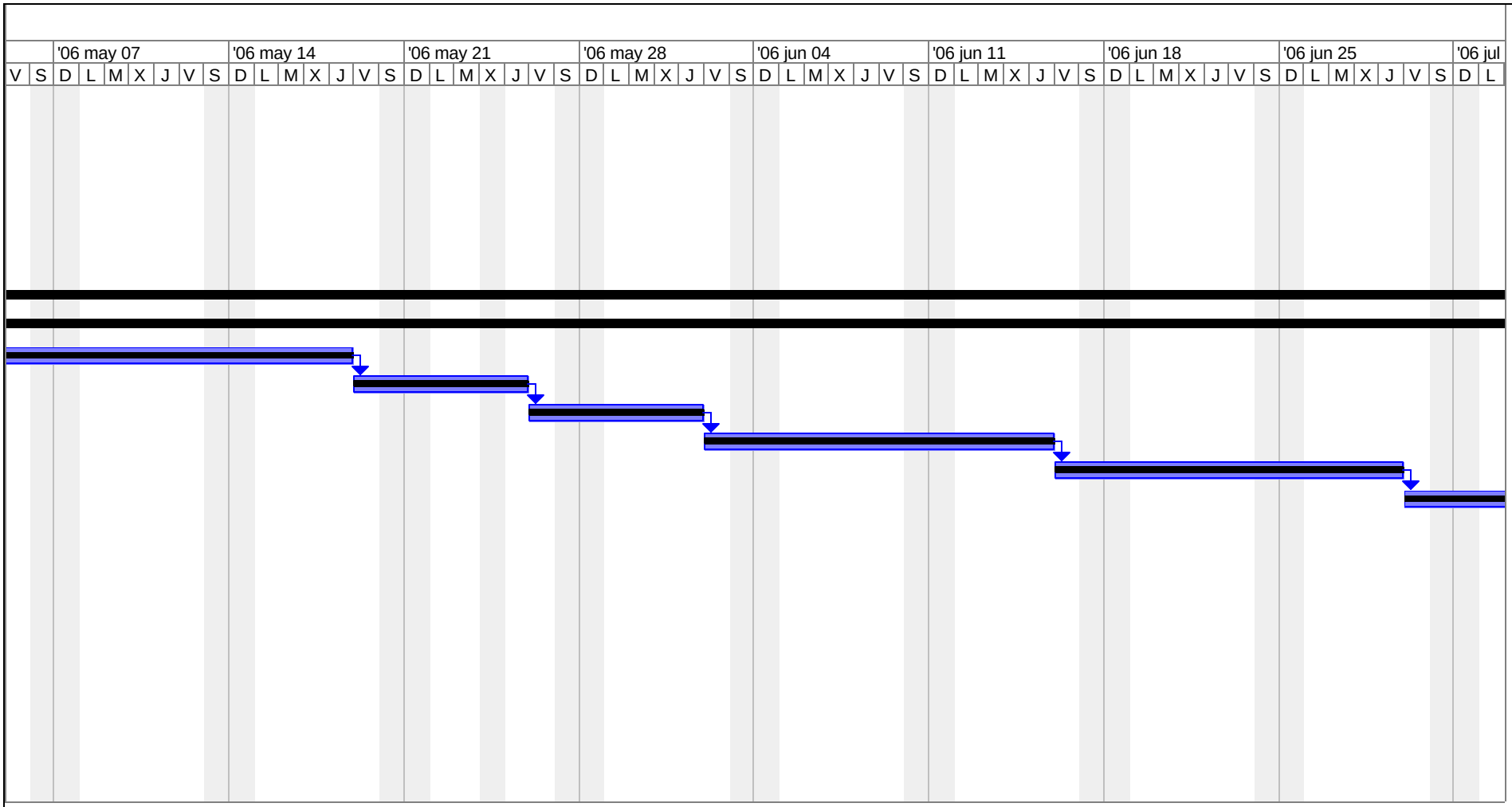
Proyecto: TesisCronogramaActivi
Fecha: mié 07/08/01

Tarea
División
Progreso
Hito
Resumen
Resumen del proyecto
Tareas externas
Hito externo
Fecha límite



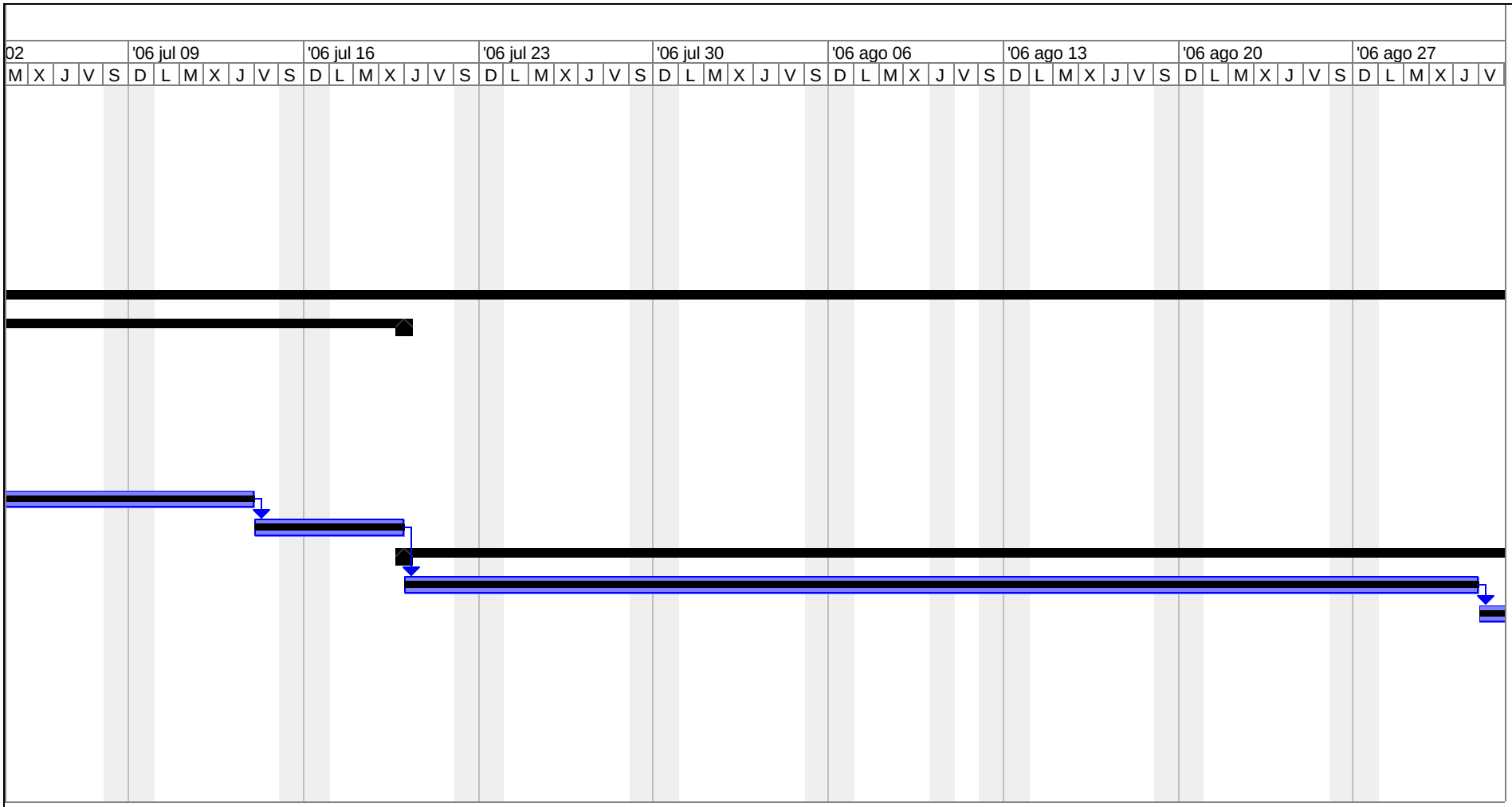
Proyecto: TesisCronogramaActivi
Fecha: mié 07/08/01

Tarea		Hito		Tareas externas	
División		Resumen		Hito externo	
Progreso		Resumen del proyecto		Fecha límite	



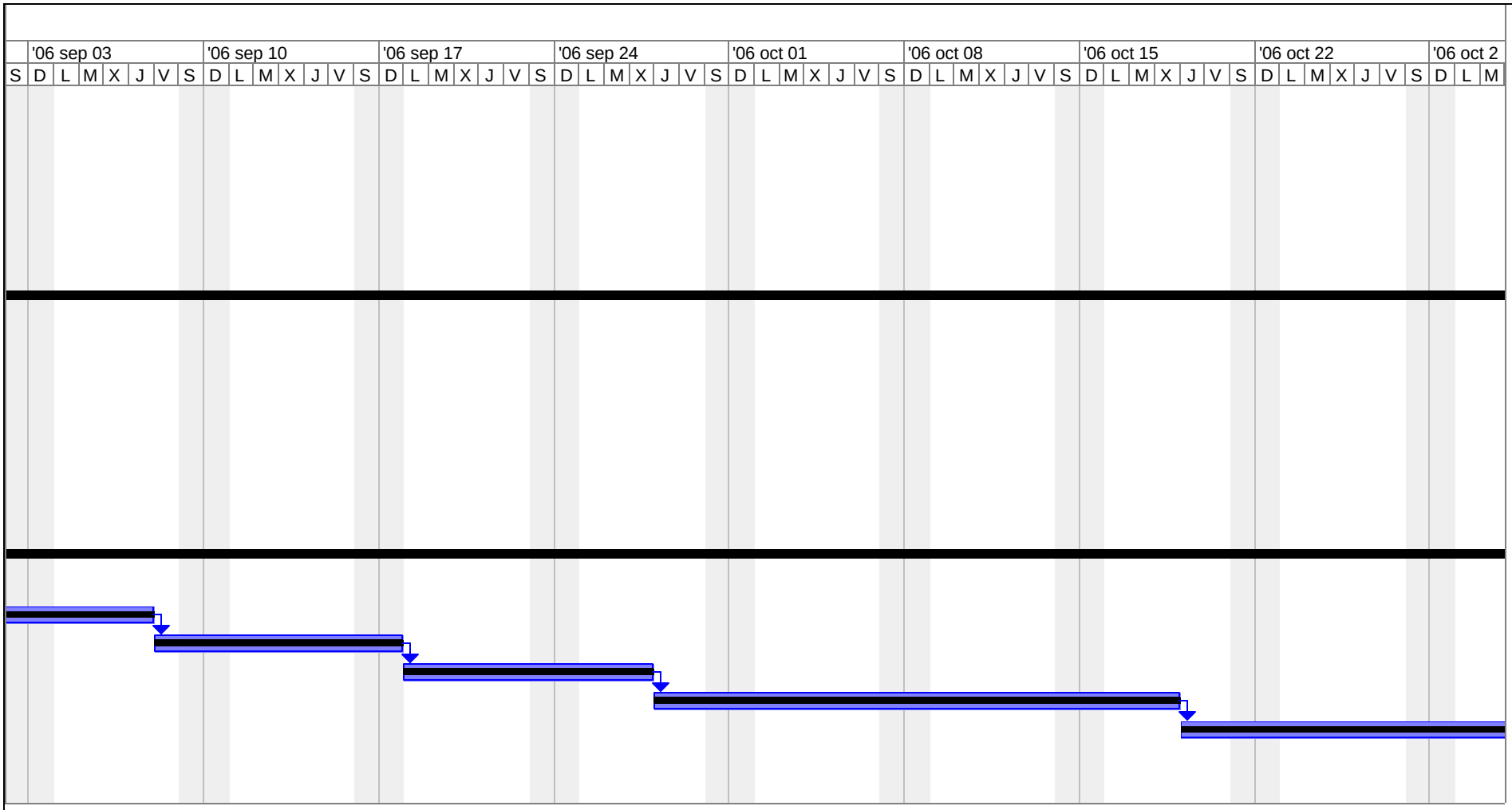
Proyecto: TesisCronogramaActivi
Fecha: mié 07/08/01

Tarea		Hito		Tareas externas	
División		Resumen		Hito externo	
Progreso		Resumen del proyecto		Fecha límite	



Proyecto: TesisCronogramaActivi
Fecha: mié 07/08/01

Tarea		Hito		Tareas externas	
División		Resumen		Hito externo	
Progreso		Resumen del proyecto		Fecha límite	



Proyecto: TesisCronogramaActivi
Fecha: mié 07/08/01

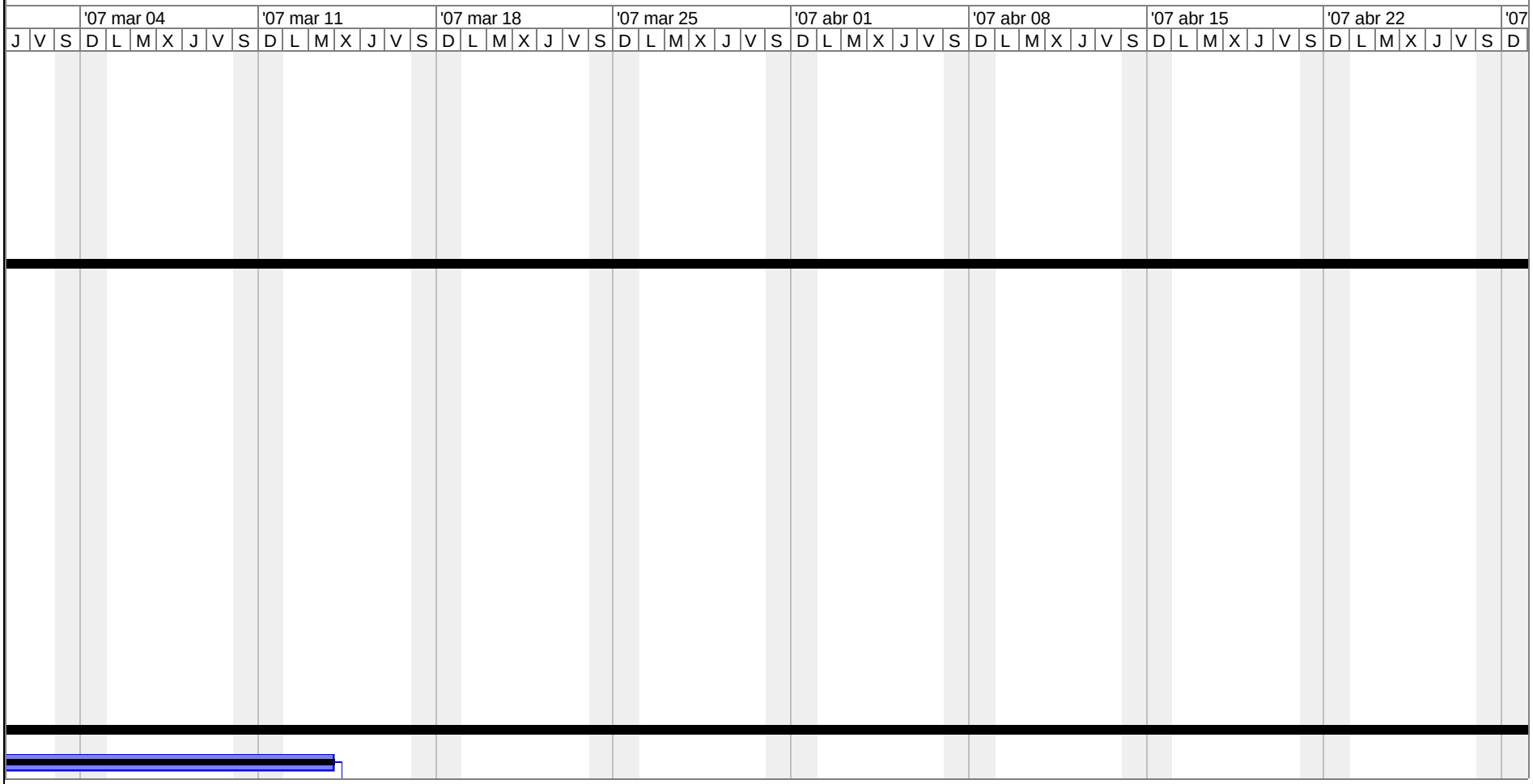
Tarea		Hito		Tareas externas	
División		Resumen		Hito externo	
Progreso		Resumen del proyecto		Fecha límite	

Proyecto: TesisCronogramaActivi
Fecha: mié 07/08/01

Tarea		Hito		Tareas externas	
División		Resumen		Hito externo	
Progreso		Resumen del proyecto		Fecha límite	

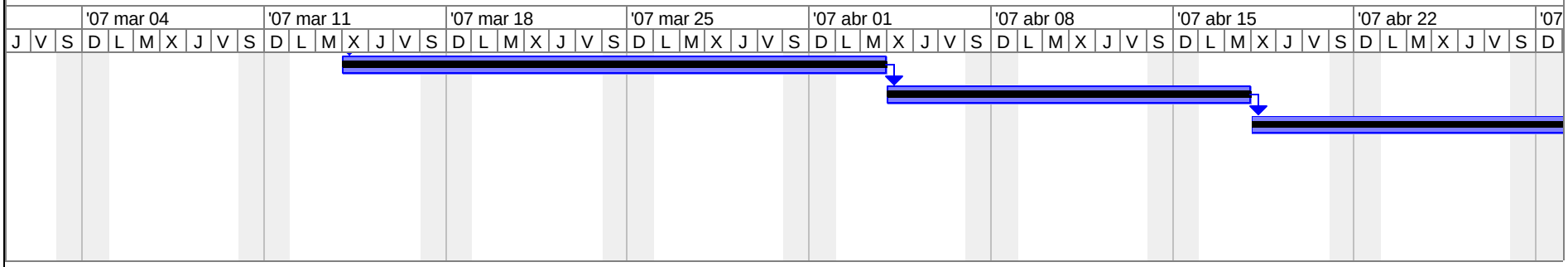
Proyecto: TesisCronogramaActivi
Fecha: mié 07/08/01

Tarea		Hito		Tareas externas	
División		Resumen		Hito externo	
Progreso		Resumen del proyecto		Fecha límite	



Proyecto: TesisCronogramaActivi
Fecha: mié 07/08/01

Tarea		Hito		Tareas externas	
División		Resumen		Hito externo	
Progreso		Resumen del proyecto		Fecha límite	

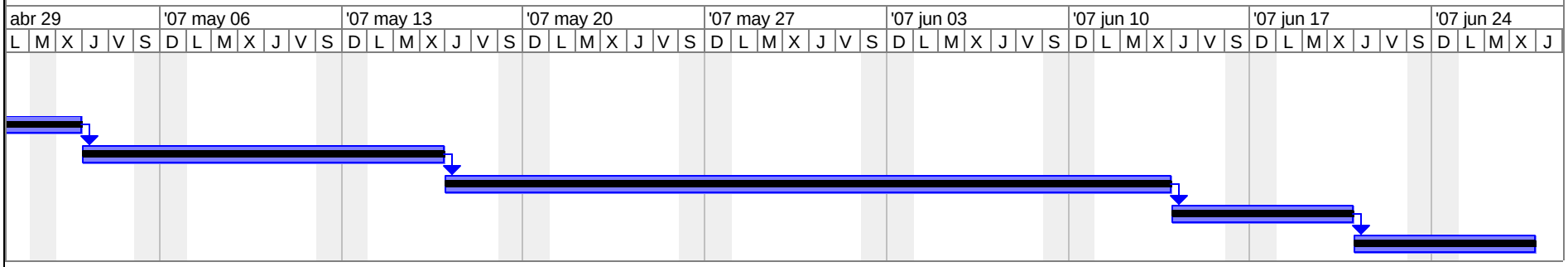


Proyecto: TesisCronogramaActivi
Fecha: mié 07/08/01

Tarea		Hito		Tareas externas	
División		Resumen		Hito externo	
Progreso		Resumen del proyecto		Fecha límite	

Proyecto: TesisCronogramaActivi
Fecha: mié 07/08/01

Tarea		Hito		Tareas externas	
División		Resumen		Hito externo	
Progreso		Resumen del proyecto		Fecha límite	



Proyecto: TesisCronogramaActivi
Fecha: mié 07/08/01

Tarea		Hito		Tareas externas	
División		Resumen		Hito externo	
Progreso		Resumen del proyecto		Fecha límite	

ANEXO 1.3 MATRIZ DE RIESGOS

ANEXO 1.3 MATRIZ DE RIESGOS - PROYECTO BANCA CELULAR CON TECNOLOGIA SMS

#	Enunciado del Riesgo		(1-5)	(1-3)					
	Descripción/Condición	Consecuencia	Probabilidad	Impacto	Exposición	Estrategia Mitigación	Contingencia	Disparador Contingencia	Asignación
1	Desconocimiento de la tecnología celular	Retrazo en el proyecto	3	3	9	Capacitación		Gerencia/Sistemas	Sistemas
2	Falta de compromiso de la institución para definiciones relacionadas con el proyecto por parte del usuario	Retrazo en el Proyecto	1	3	3	Informes gerencia general			Gerencia
3	Falla en contratación pasarela	Fracaso del proyecto	1	3	3		Contratación de otra pasarela		Gerencia
4	Establecimiento de VPN con operadoras	Retrazo en el proyecto	2	3	6	Informes a directivos de operadoras	Conexiones dedicadas con operadoras	Gerencia/Comunicaciones	Comunicaciones
5	Falla en instalación de servidor	Retrazo en el Proyecto	1	3	3	Soporte proveedor hardware	Servidor de respaldo	Sistemas	Sistemas

6	Fallas en la pasarela	Dismunción de calidad del servicio	2	3	6	Adecuada planificación de pruebas y certificación antes del ingreso a producción	Adecuada planificación de pruebas y certificación despúes del ingreso a producción	Auditoría/Sistemas	Sistemas
7	Fallas en los componentes de servicio de operación bancaria	Fallas en el servicio	3	3	9	Adecuada planificación de pruebas y certificación antes del ingreso a producción	Adecuada planificación de pruebas y certificación antes del ingreso a producción	Auditoría/Sistemas	Auditoría/Sistemas

Revisado

Director de Tesis

Nombre: Ing. Pablo Pintado

Fecha 2006/Jul/20

Institución financiera

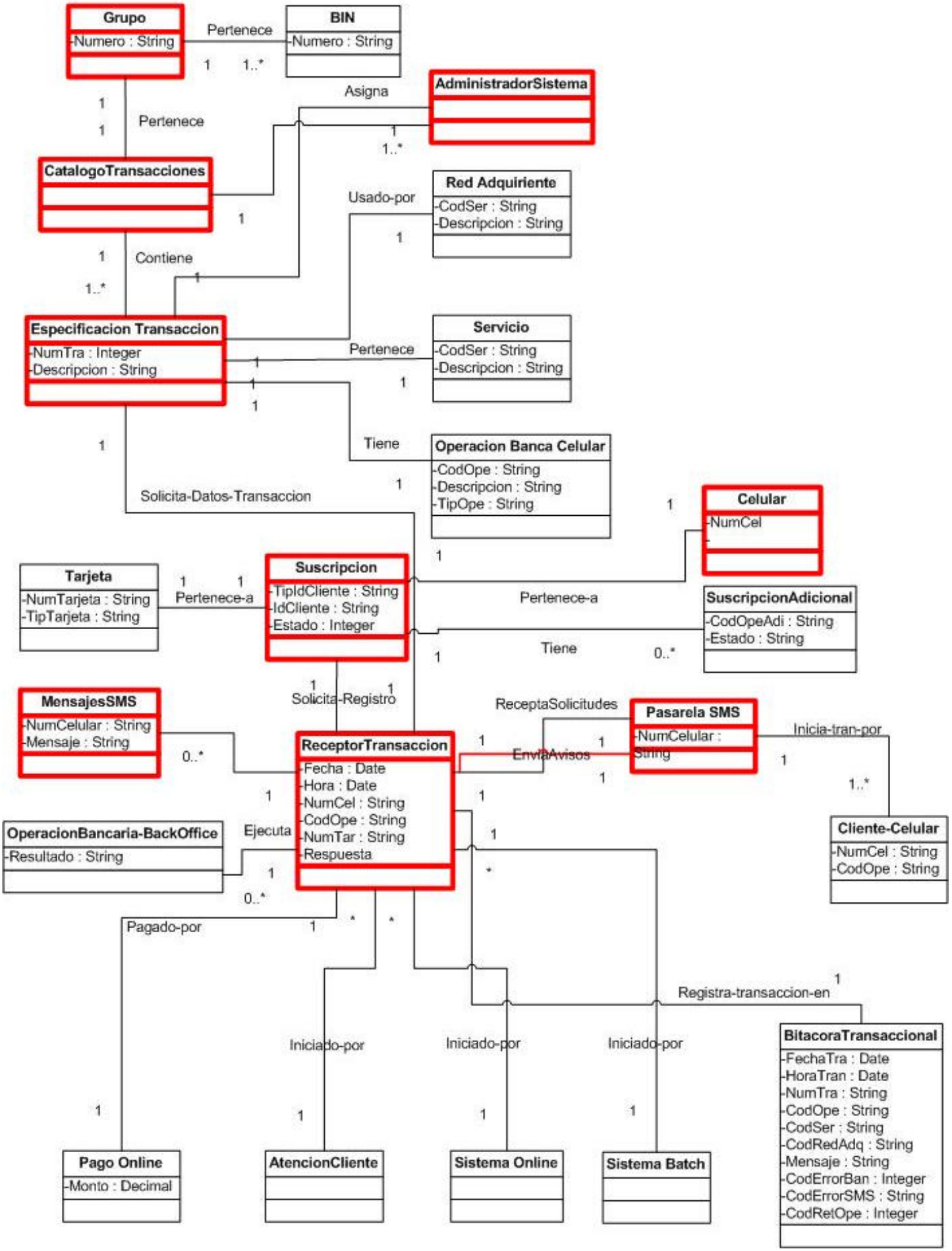
Desarrollador Tesis

Nombre: Ing. Juan Diego Landy

Fecha 2006/Jul/20

ANEXO 3.1 MODELO CONCEPTUAL

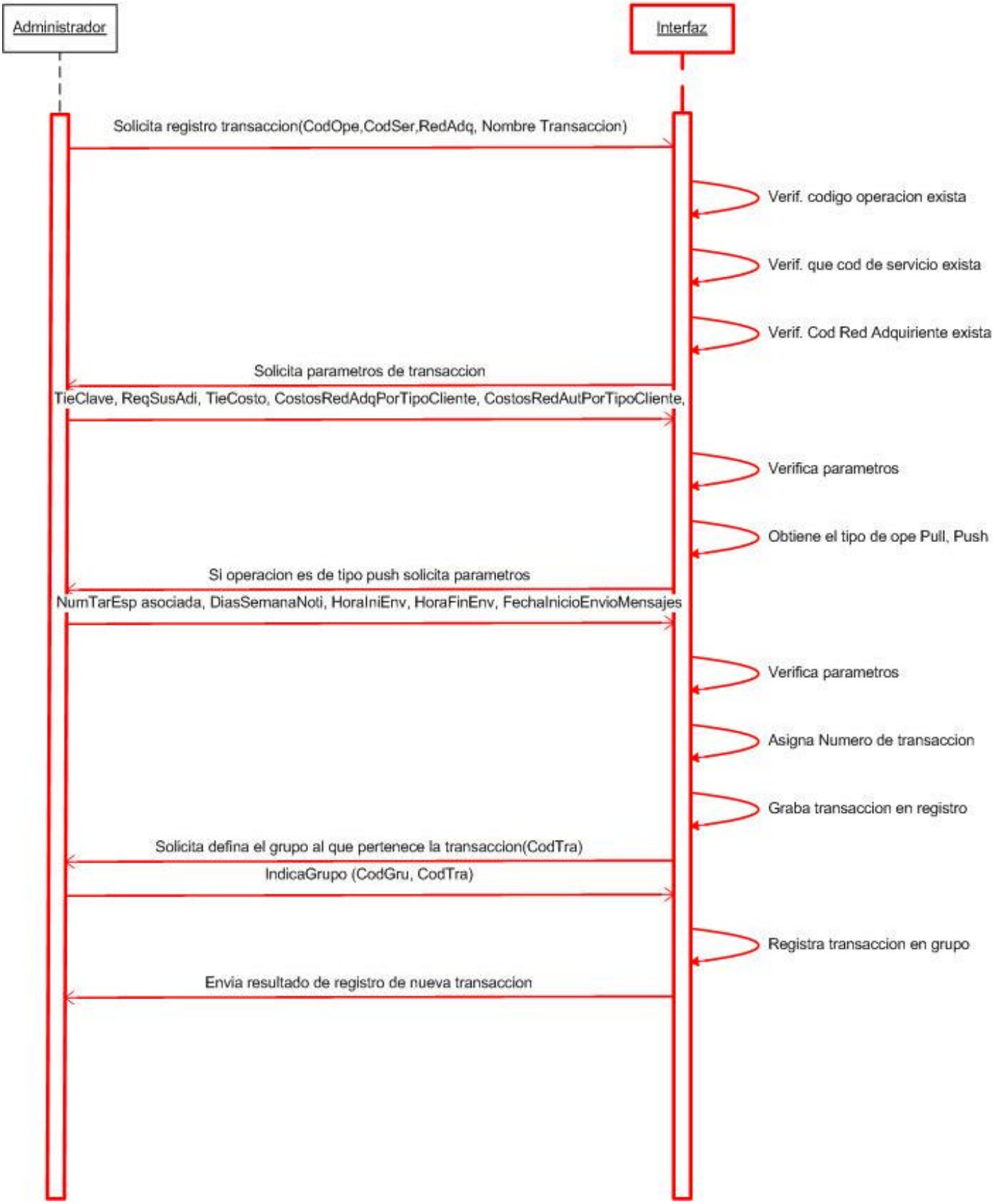
ANEXO 3.1 MODELO CONCEPTUAL – BANCA CELULAR



ANEXO 3.2 DIAGRAMAS DE SECUENCIA

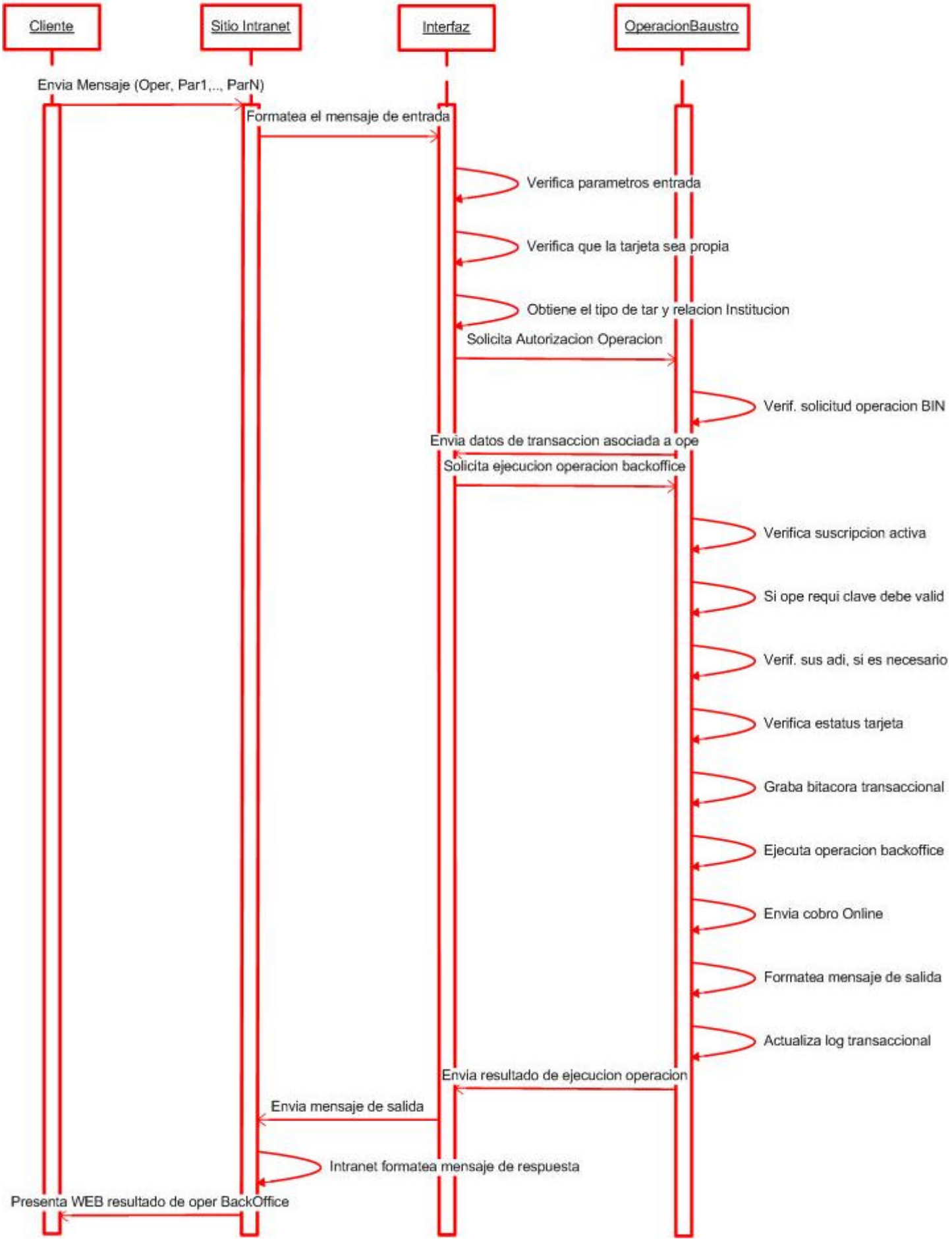
ANEXO 3.2 (a) DIAGRAMA DE SECUENCIA: ADMINISTRACIÓN DEL SISTEMA

Escenario: Creación de transacción Banca Celular



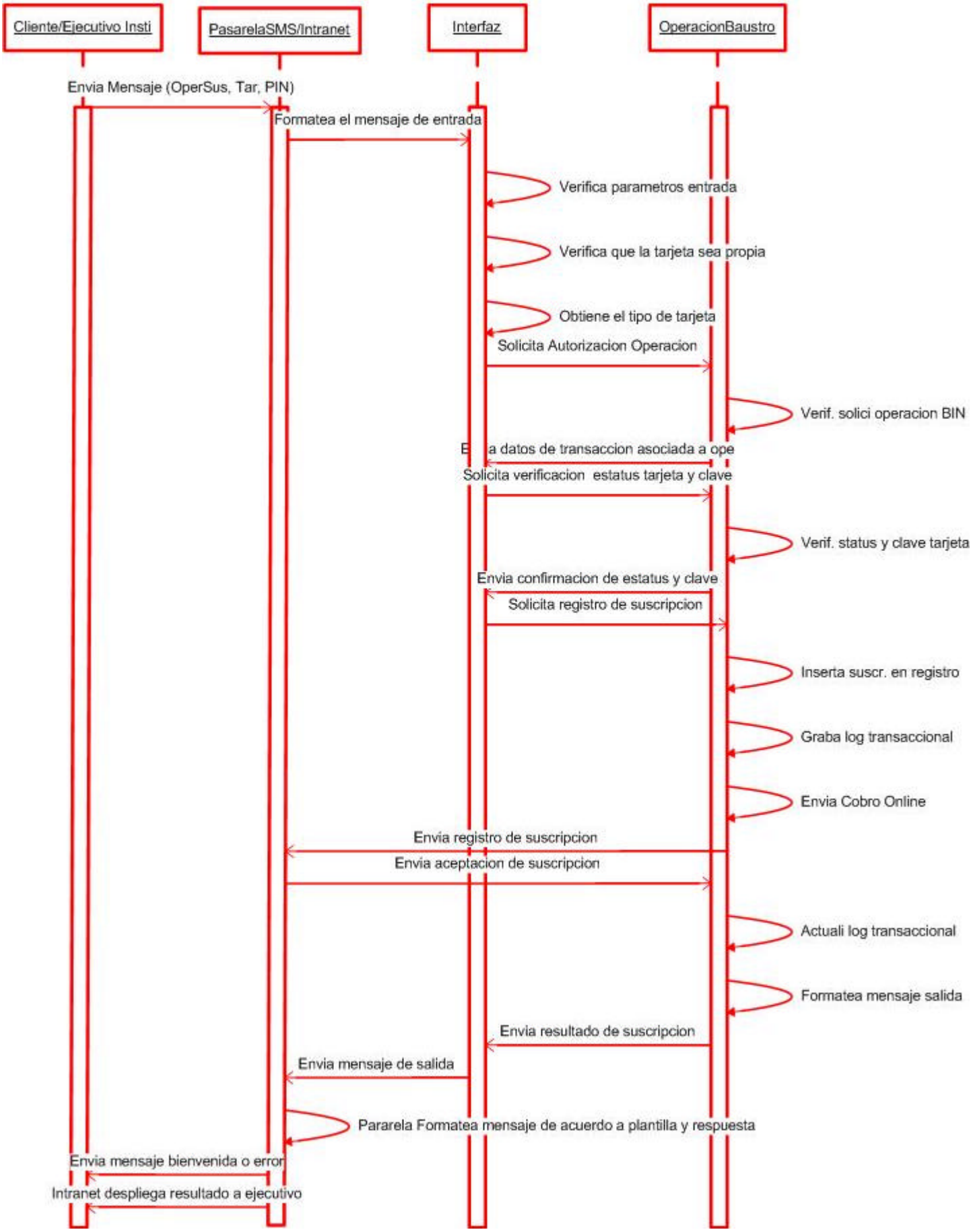
ANEXO 3.2 (b) DIAGRAMA DE SECUENCIA: TRANSACCIÓN BACKOFFICE

Escenario: Intranet



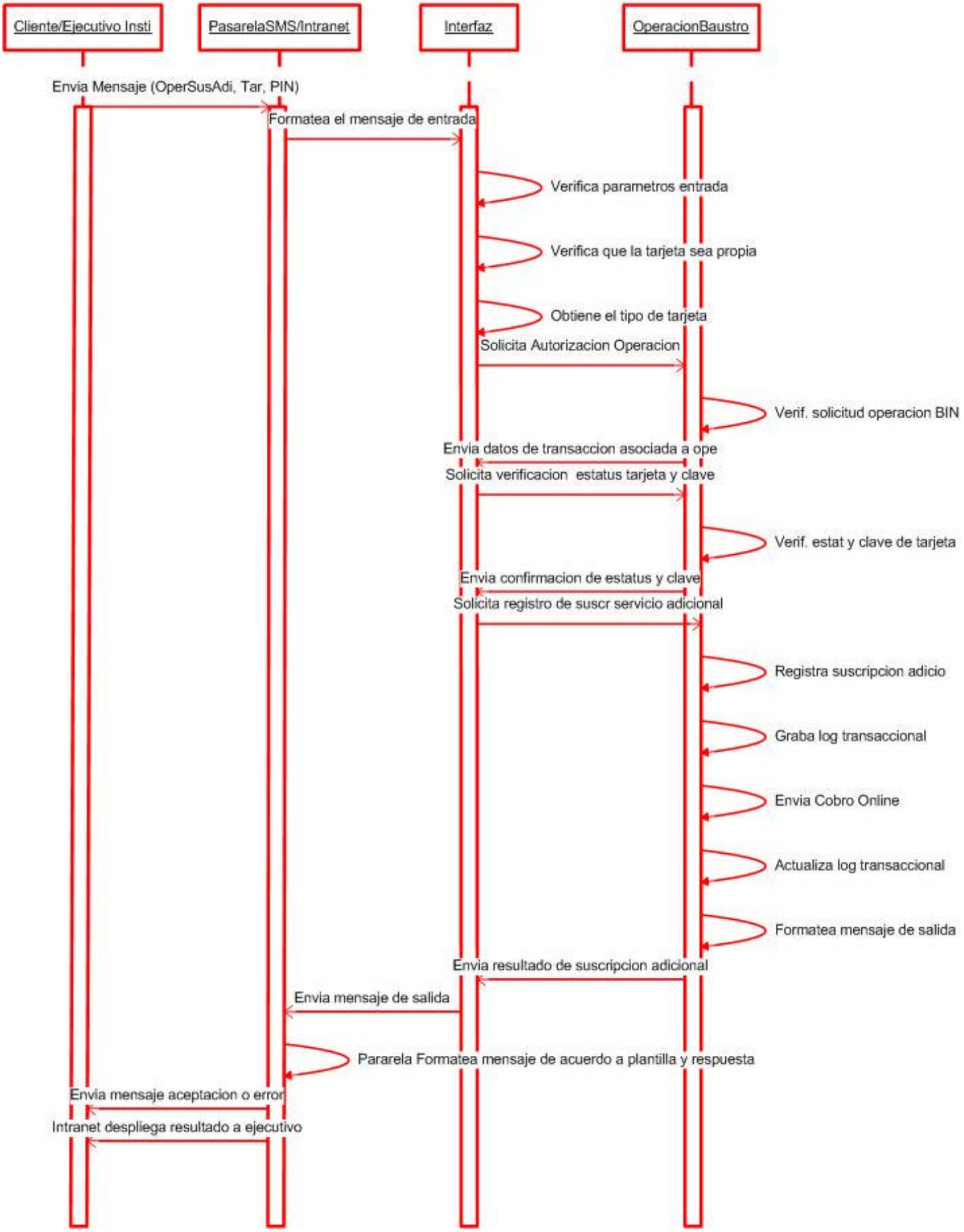
ANEXO 3.2 (c) DIAGRAMA DE SECUENCIA: SUSCRIPCION DE USUARIO

Escenario: SMS/Intranet



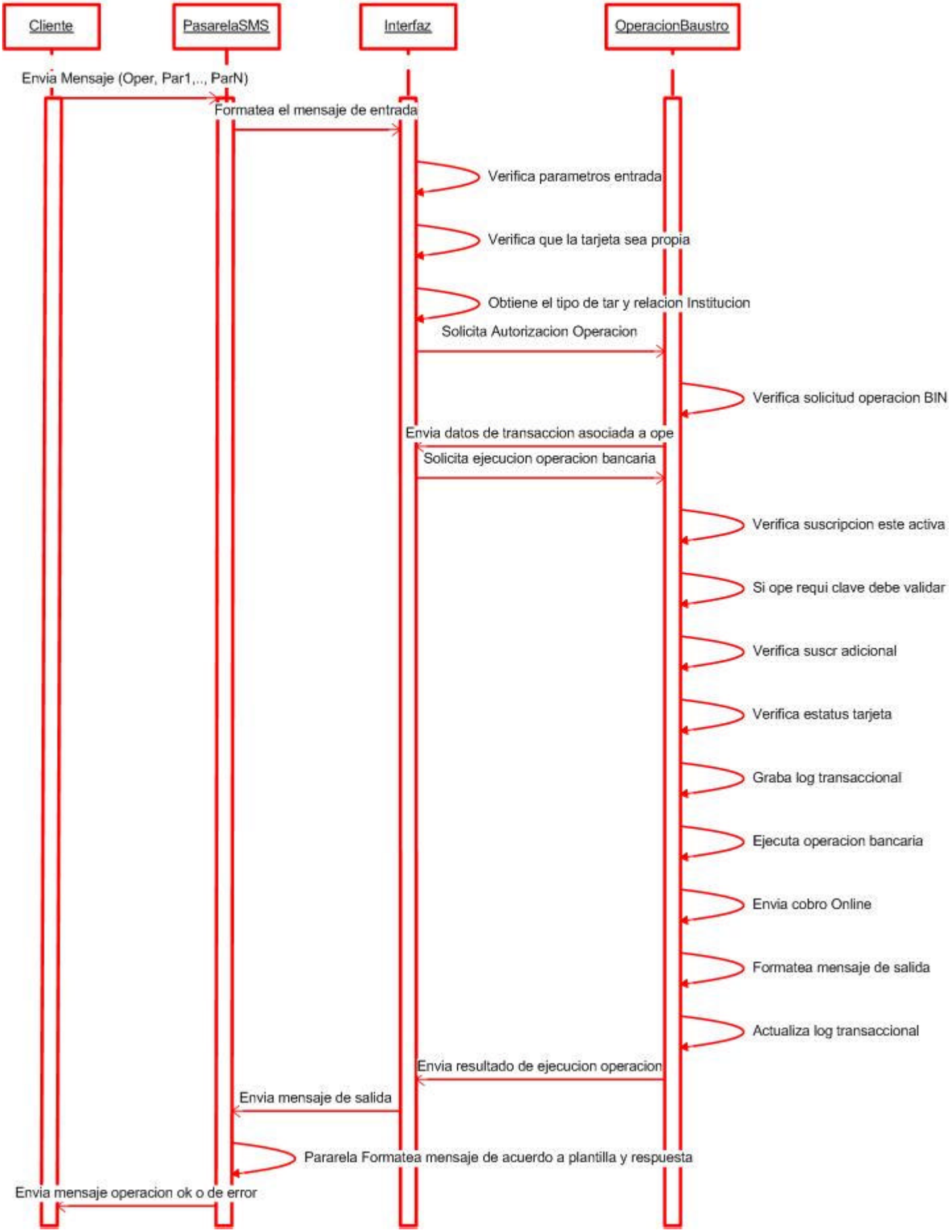
ANEXO 3.2 (d) DIAGRAMA DE SECUENCIA: SUSCRIPCION SERVICIO ADICIONAL

Escenario: SMS/Intranet



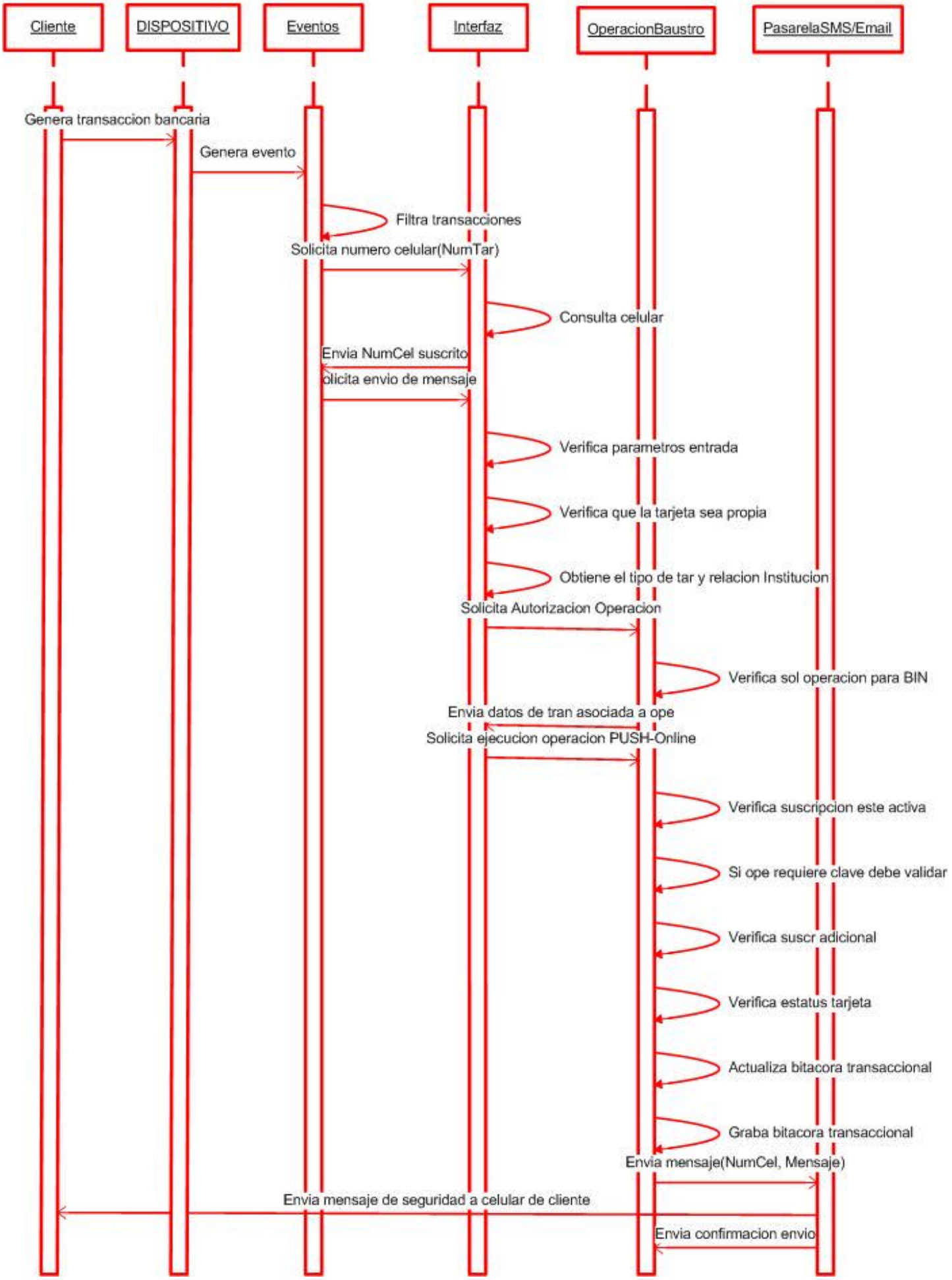
ANEXO 3.2 (e) DIAGRAMA DE SECUENCIA: TRANSACCIONES PULL

Escenario: SMS



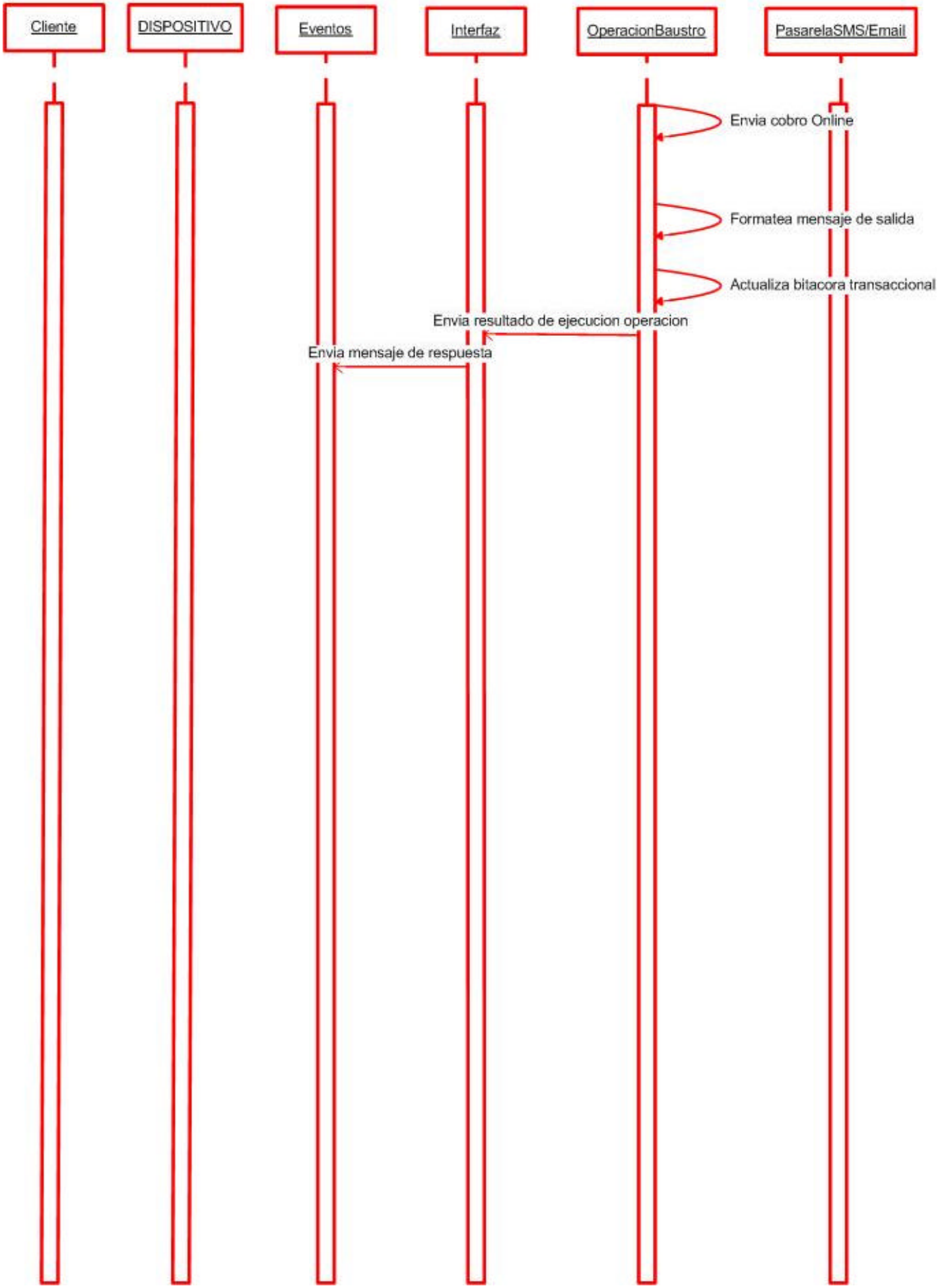
ANEXO 3.2 (f) DIAGRAMA DE SECUENCIA: TRANSACCIONES PUSHONLINE CON SUSCRIPCIÓN

Escenario: SMS



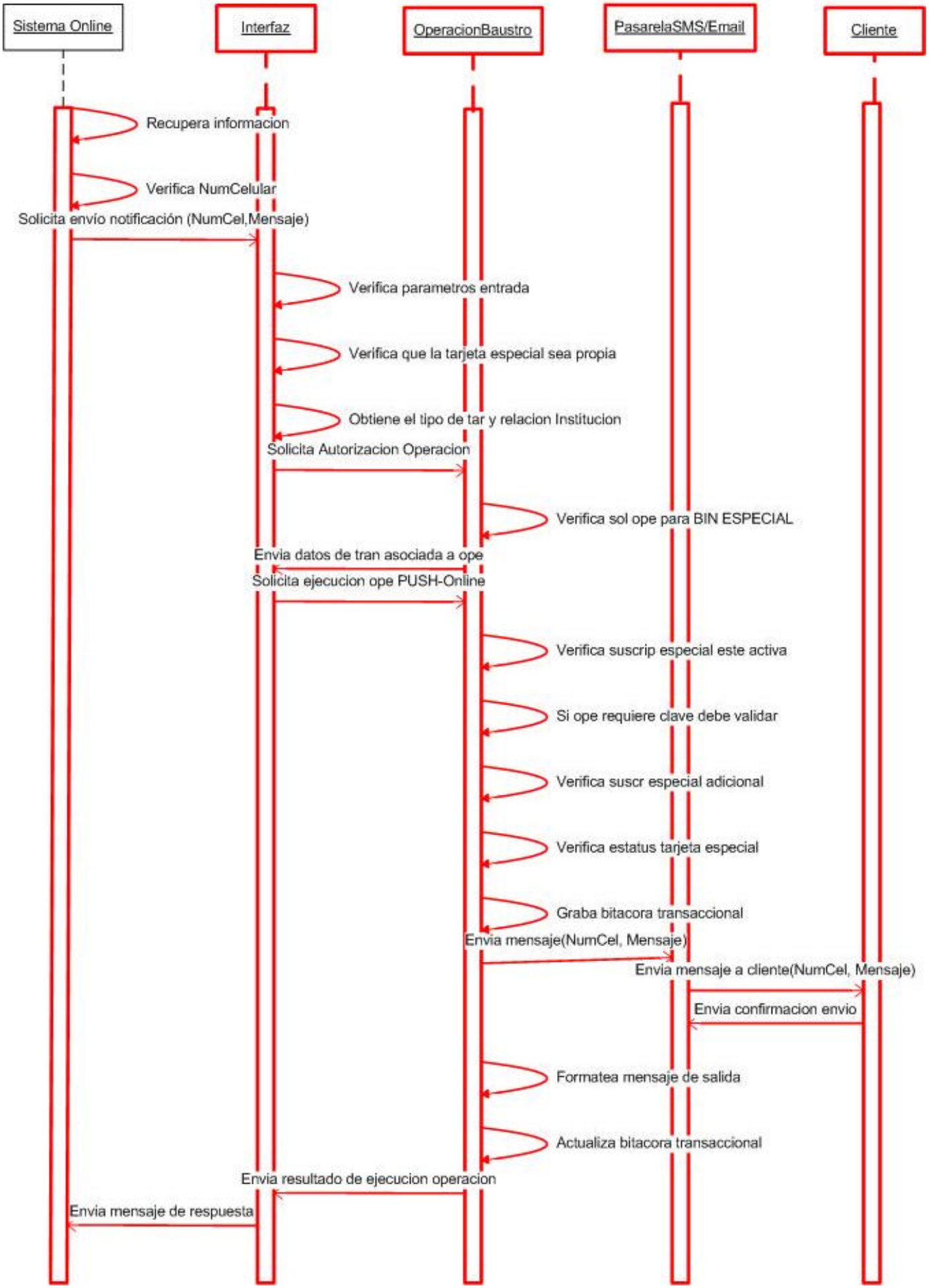
ANEXO 3.2 (f) DIAGRAMA DE SECUENCIA: TRANSACCIONES PUSHONLINE CON SUSCRIPCIÓN - Continuación

Escenario: SMS



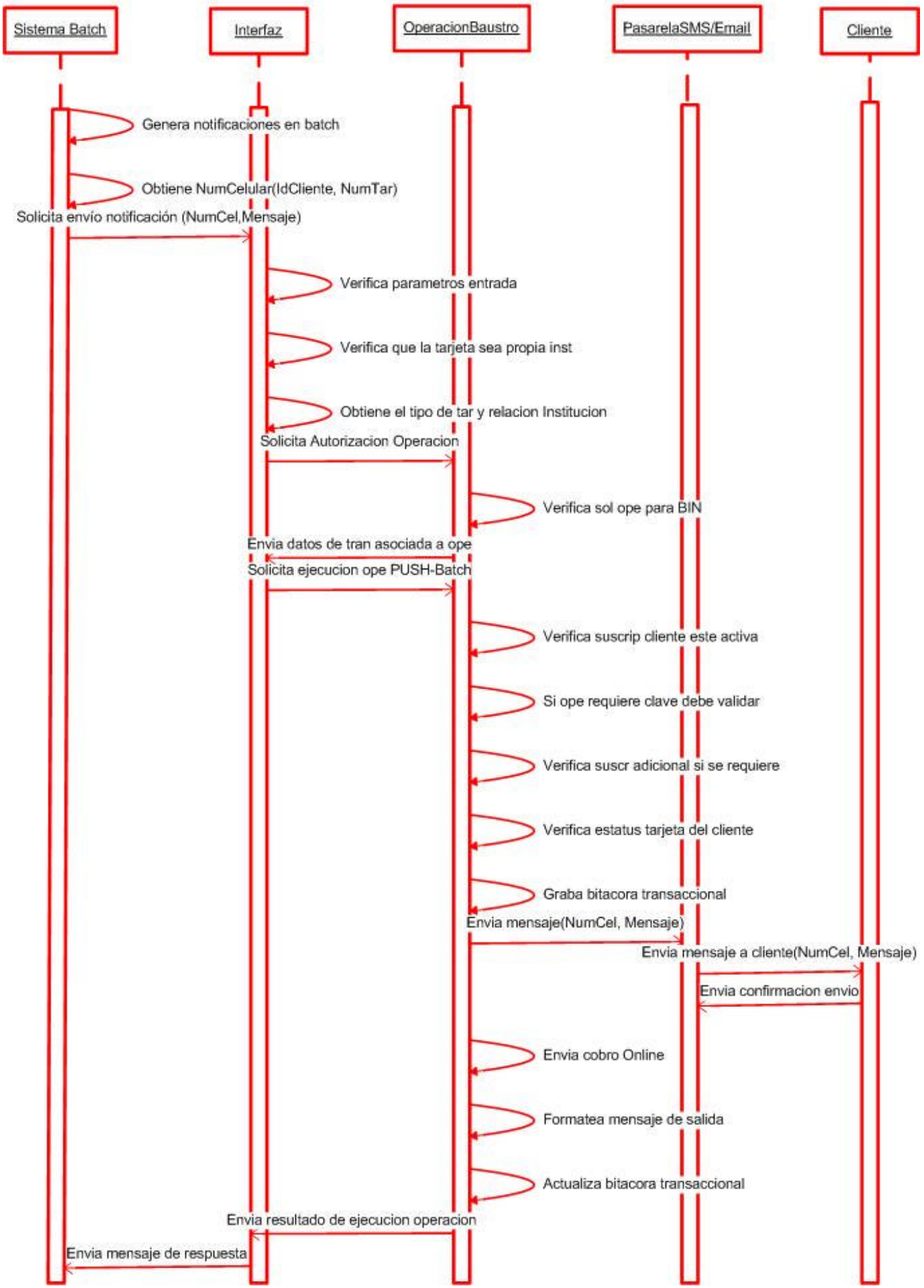
ANEXO 3.2 (g) DIAGRAMA DE SECUENCIA: TRANSACCIONES PUSHONLINE SIN SUSCRIPCIÓN

Escenario: SMS



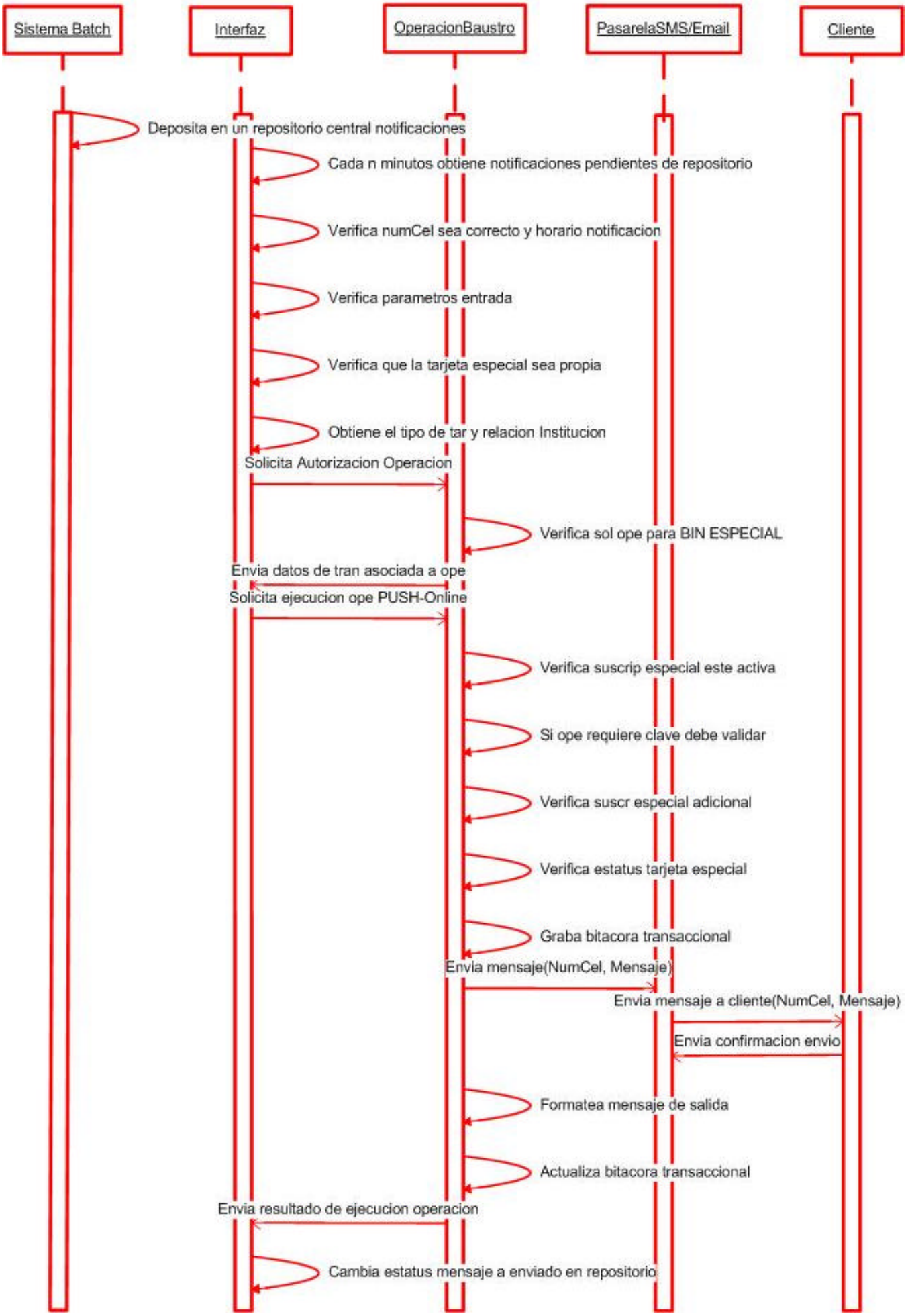
ANEXO 3.2 (h) DIAGRAMA DE SECUENCIA: TRANSACCIONES PUSHBATCH CON SUSCRIPCIÓN

Escenario: SMS



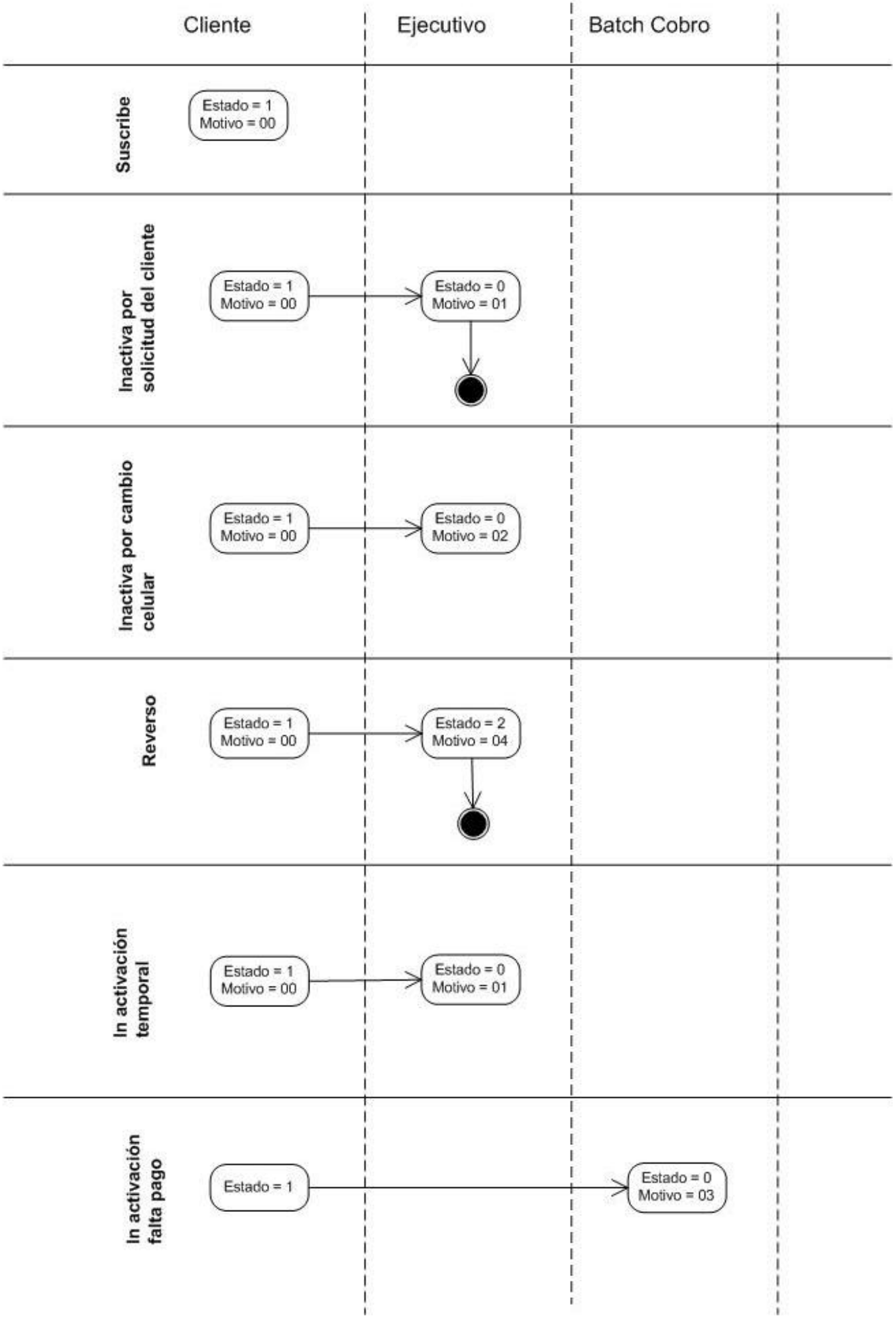
ANEXO 3.2 (i) DIAGRAMA DE SECUENCIA: TRANSACCIONES PUSHBATCH SIN SUSCRIPCIÓN

Escenario: SMS



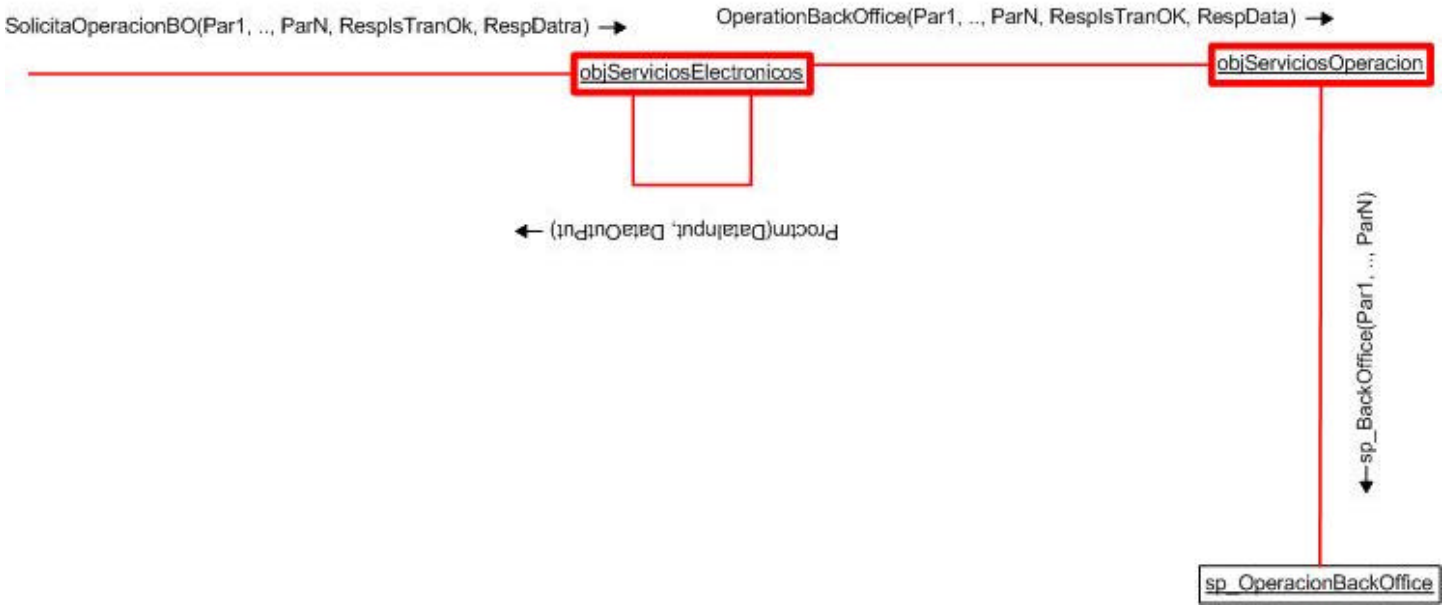
ANEXO 3.3 DIAGRAMA DE ESTADOS

ANEXO 3.3 DIAGRAMA DE ESTADOS DE UNA SUSCRIPCIÓN PRINCIPAL Y SUSCRIPCIÓN DE UN SERVICIO ADICIONAL



ANEXO 4.1 DIAGRAMAS DE COLABORACION

ANEXO 4.1 (a) DiAGRAMA DE COLABORACION – TRANSACCIONESBACKOFFICE



ANEXO 4.1 (b) DIAGRAMA DE COLABORACIÓN – TRANSACCIONES PULL

ANEXO 4.1 (c) DIAGRAMA DE COLABORACIÓN – TRANSACCIONES PUSH

ANEXO 4.2 DIAGRAMA DE CLASE SERVICIOSELECTRONICOS

ANEXO 4.2 DIAGRAMA DE CLASE SERVICIOSELECTRONICOS.INTERFAZ

ServiciosElectronicos.Interfaz

-objServiciosOperacion : Object
-Server : <unspecified> = "10.1.1.12"
-App_Path : String
-strMsgErr : String
-strFileName : String
-strPrograma : String
-strMenCli : String

-BinTar(in TerNumTar : String, inout RespBinTar : String, inout RespEsLocal : String, inout RespProSMS : String, inout RespIsTranOk : Boolean, inout RespError : String)
-GrabaErrorExcepcion(in strFileName : String, in strPrograma : String, in strMetFunSub : String, in strMsgErr : String, in strMenErrSis : String)
+ProcTrn(in DataInput : String, inout DataOutput : String)
+OperacionAutorizada(in TerNumTar, inout RespIsTranOk, inout RespData) : Recordset
+MotivoInactivacion(inout RespIsTranOk, inout RespData) : Recordset
+SuscripcionSE(in TerNumCel, in TerCodSer, in TerCodRedAdq, in TerIdeDisOri, in TerUsuOri, in TerNumTar, in TerPin) : String
+CambiaClaveSE(in TerNumCel, in TerCodSer, in TerCodRedAdq, in TerIdeDisOri, in TerUsuOri, in TerNumTar, in TerClaveAntSE, in TerClaveNueSE) : String
+ModificaSuscripcionSE(in TerNumCel, in TerCodSer, in TerCodRedAdq, in TerIdeDisOri, in TerUsuOri, in TerNumTar, in TerClave, in TerNumCelNue) : String
+SuscripcionAutorizadaSE(in TerNumCel, in TerCodSer, in TerCodRedAdq, in TerIdeDisOri, in TerUsuOri, in TerNumTar, in TerClaveSE, in TerCodOpe) : String
+InactivaSusAutSE(in TerNumCel, in TerCodSer, in TerCodRedAdq, in TerIdeDisOri, in TerUsuOri, in TerNumTar, in TerClaveSE, in TerCodOpe, in TerCodMot) : String
+InactivaSuscripcionSE(in TerNumCel, in TerCodSer, in TerCodRedAdq, in TerIdeDisOri, in TerUsuOri, in TerNumTar, in TerClaveSE, in TerCodMot) : String
+ReversaSuscripcionSE(in TerNumCel, in TerCodSer, in TerCodRedAdq, in TerIdeDisOri, in TerUsuOri, in TerNumTar, in TerClaveSE, in TerSupOri, in TerSupNom) : String
+ReversaSusAutSE(in TerNumCel, in TerCodSer, in TerCodRedAdq, in TerIdeDisOri, in TerUsuOri, in TerNumTar, in TerClaveSE, in TerCodOpe, in TerSupOri, in TerSupNom) : String
+ConsultaClaveSE(in TerNumCel, in TerCodSer, in TerCodRedAdq, in TerIdeDisOri, in TerUsuOri, in TerNumTar, in TerSupOri, in TerSupNom) : String
+ConsultaSuscripcionesSE(in TerNumCel, in TerCodSer, in TerCodRedAdq, in TerNumTar, in TerFecSusIni, in TerFecSusFin, inout RespIsTranOk, inout RespData) : Recordset
+ConsultaSusAutSE(in TerCodSus, in TerNumCel, in TerCodSer, in TerCodRedAdq, in TerNumTar, inout RespIsTranOk, inout RespData) : Recordset
+ConsultaTransaccionesSE(in TerNumCel, in TerCodSer, in TerCodRedAdq, in TerNumTar, in TerFecTraIni, in TerFecTraFin, in TerTipCue, in TerNumCue, in TerTipTra, inout RespIsTranOk, inout RespData) : Recordset
+ModificaTelSusSE(in TerNumCel, in TerCodSer, in TerCodRedAdq, in TerIdeDisOri, in TerUsuOri, in TerNumTar, in TerClaveSE) : String
-Class_Initialize()
-ValidaCel(in TerNumCel : String, inout FinCel : String, inout Peso : Integer) : Boolean
+ConsultaRemesas(inout RespIsTranOk : Boolean, inout RespData : String)
+NotificaRemesas(in TerRemCod : String, in TerNumGir : String, in TerUsuDes : String, in TerTelefono : String, in RemMensaje : String, inout RespIsTranOk : Boolean, inout RespData : String)
+NotificaConsumosTC(in Fecha : String, in Dispositivo : String, in Secuencial : String, in TipoDisp : String, in Tarjeta : String, in TipoTran : String, in Valor : Double, in Reverso : String, in CodEstab : String, in TipoTarjeta : String)
+NotificaConsumosATM(in Fecha : String, in Dispositivo : String, in Secuencial : String, in TipoDisp : String, in Tarjeta : String, in TipoTran : String, in Valor : Double, in Reverso : String, in CodEstab : String, in TipoTarjeta : String)
+DaDirectorio(in strDirectorio : String)
-Maneja_Archivos_Log(in Mensaje : String)
+ConsultaTpbPushBatch(inout RespIsTranOk : Boolean, inout RespData : String)
+NotificaPushBatch(in TerCodTPB : Long, in TerCodOpe : String, in TerMensaje : String, in PushBatchMensaje : String, in TerRedOri : String, in TerUsuDes : String, in TerTelefono : String, in TerTipPas : String, in inf1 : String, in inf2 : String, inout RespIsTranOk : Boolean, inout RespData : String)

ANEXO 4.3 DIAGRAMA DE CLASE SERVICIOSOPERACION

ANEXO 4.3 DIAGRAMA DE CLASE SER

ServiciosOpera

-oCnServiciosE : Connection
-BoCnServiciosE : String
-lRetornoServiciosE : Long
-oCnServiciosE2 : Connection
-BoCnServiciosE2 : String
-lRetornoServiciosE2 : Long
-sMensaje : String
-oCnTarCre : Connection
-BoCnTarCre : String
-lRetornoTarCre : Long
-oCnTarDeb : Connection
-BoCnTarDeb : String
-lRetornoTarDeb : Long
-oCnSmsAdmin : Connection
-BoCnSmsAdmin : String
-lRetornoSmsAdmin : Long
-TerNumEst : Double
-TerCat : Long
-oCnServiciosR : Connection
-BoCnServiciosR : String
-lRetornoServiciosR : Long
-oCnServiciosSMS : Connection
-BoCnServiciosSMS : String
-lRetornoServiciosSMS : Long
-m_OutLook : Application
-objMail : MailItem
-ObjKey : Object
-strFileName : String
-strPrograma : String
-strMenCli : String
-obj : Object
-strSubject : String
-strDomPorta : String
-strDomAlegro : String
-strDomMovi : String

-EnviaAPPC(in TerNumCue : String, in TerNumChe : String, inout RespIsTranOk : Boolean, inout DataOutput : String)

.VICIOSOPERACION.INTERFAZ

acion.Interfaz

-EnviaAPC(in TerNumCel : String, in TerNumCel : String, inout RespIsTranOk : Boolean, inout DataOutput : String)
-EnviaEMail(in strDestino : String, in strSubject : String, in strMensaje : String, inout RespIsTranOk : Boolean, inout RespData : String)
+mensajeError(in strTipErr : String, in strMetFunSubGlo : String, in intCodErr : Integer, in lngErrNumber : Long, in strErrDescription : String, inout RespData : String)
-getError(in intCodError : Integer, inout RespIsTranOk : Boolean, inout RespData : String) : Recordset
-InsertaLogError(in strTipErr : String, in lngErrNumber : Long, in strErrDescription : String, in intCodErr : Integer, in strDesUsu : String, in strDesSis : String, in strPrograma : String, in strMetSubFun : String, inout RespIsTranOk : Boolean, inout RespData : String)
-SetConeccionTarCre(inout sMensaje : String, inout IRetornoTarCre : Long)
-SetConeccionSmsAdmin(inout sMensaje : String, inout IRetornoSmsAdmin : Long)
-SetConeccionTarDeb(inout sMensaje : String, inout IRetornoTarDeb : Long)
-ConsultaTarCre(in TerNumTar : String, inout RespTarHab : String, inout RespTipTarHab : String, inout RespIdeCli : String, inout RespIsTranOk, inout DataOutput : String)
-ConsultaTarDeb(in TerNumTar : String, inout RespTarHab : String, inout RespTipTarHab : String, inout RespIdeCli : String, inout RespIsTranOk, inout DataOutput : String)
-ConsultaOperadora(in Celular : String) : String
-ConsultaOperadoraCelular(in Celular : String, inout RespOperadora : String, inout RespIsTranOk : Boolean, inout DataOutput : String)
-ObtieneOperadora(in TerNumCel : String) : String
-VerificaSuscripcionClave(in TerNumTar : String, in TerNumCel : String, in TerCodSer : String, in TerTieCla : String, in TerClave : String, inout RespPago : String, inout RespCodSus : Long, inout RespIsTranSusOk : Boolean, inout RespIsTranClaOk : Boolean, inout DataOutput : String)
-VerificaSuscripcionE(in TerNumTar : String, in TerNumCel : String, in TerCodSer : String, in TerCodOpe : String, inout RespIsTranOk : Boolean, inout DataOutput : String)
-ConsultaSusNorOEspAct(in TerNumTar : String, in TerNumCel : String, in TerCodSer : String, inout TerSusSE : Recordset, in TerVerSusHoy : Boolean, inout RespFecSus : String, inout RespHorSus : String, inout RespIsTranOk : Boolean, inout DataOutput : String)
-ConsultaSusAutAct(in TerNumTar : String, in TerNumCel : String, in TerCodSer : String, in TerCodOpeSE : String, inout TerSusSE : Recordset, in TerVerSusHoy : Boolean, inout RespIsTranOk : Boolean, inout DataOutput : String)
+OperacionAutorizada(in TerCodSer : String, in TerCodRedAdq : String, in TerBinTar : String, in TerEsLocal : String, inout RespIsTranOk : Boolean, inout RespData : String) : Recordset
+MotivoInactivacion(inout RespIsTranOk : Boolean, inout RespData : String) : Recordset
-ConsultaClave(in TerCodSer : String, in TerNumCel : String, in TerNumTar : String, inout RespIsTranOk : Boolean, inout RespData : String) : String
-TarjetaCuenta(in TerNumTar : String, inout RespIsTranOk : Boolean, inout DataOutput : String) : String
-InsertaSuscripcion(in TerCodSer : String, in TerCodRedAdq : String, in TerCodTra : String, in TerIdeDisOri : String, in TerUsuOri : String, in TerNomOpe : String, in TerNumCel : String, in TerProBan : String, in TerEsLocal : String, in TerTipTarHab : String, in TerNomCli : String, in TerNumTar : String, in TerIdeCli : Strir
-InsertaTransaccionSerElec(in TerFecTra : String, in TerHorTra : String, in TerCodSer : String, in TerCodRedAdq : String, in TerCodTra : String, in TerCodOpe : String, in TerCodOpeSA : String, in TerIdeDisOri : String, in TerUsuOri : String, in TerSupOri : String, in TerSupNom : String, in TerNomOpe : String, in TerNu
-InsertaSuscripcionEclipSoft(in i_tr : Integer, in i_login : String, in i_operacion : String, in i_cliente : Integer, in i_cedula : String, in i_servicio : String, in i_producto : String, in i_cuenta : String, in i_telefono : String, in i_tipo : String, in i_nuevo_telf : String, in i_mensaje : String, inout RespIsTranOk : Boolean, inout DataOut
-EliminaSusEclipSoft(in TerNumTar : String, in TerNumCel : String, inout RespIsTranOk : Boolean, inout DataOutput : String)
-ActualizaTransaccionSerElec(in TerFecTra : String, in TerHorTra : String, in TerCodSer : String, in TerCodRedAdq : String, in TerCodTra : String, in TerCodOpeSE : String, in TerIdeDisOri : String, in TerUsuOri : String, in TerNumCel : String, in TerNumTar : String, in TerTipCueDes : Integer, in TerCueDes : String, in T
-InactivaSuscripcionNorOAut(in TerTipInaSus : String, in TerCodSer : String, in TerCodOpeSE : String, in TerNumCel : String, in TerNumTar : String, in TerStatus : String, in TerCodRedAdqUltMod : String, in TerIdeDisUltMod : String, in TerUsuUltMod : String, in TerDesUltMod : String, in TerIdeCli : String, in TerProBa
-InsertaSuscripcionSE(in TerCodSer : String, in TerCodRedAdq : String, in TerCodTra : String, in TerIdeDisOri : String, in TerUsuOri : String, in TerNomOpe : String, in TerNumCel : String, in TerProBan : String, in TerEsLocal : String, in TerTipTarHab : String, in TerCodOpeSE : String, in TerNumTar : String, in TerTrar
+CambiaPassword(in TerCodSer : String, in TerRedOri : String, in TerIdeDisOri : String, in TerUsuOri : String, in TerNumCel : String, in TerNumTar : String, in TerCla : String, inout RespIsTranOk : Boolean, inout DataOutput : String)
+ModificaSuscripcion(in TerCodSer : String, in TerRedOri : String, in TerIdeDisOri : String, in TerUsuOri : String, in TerNumCel : String, in TerNumTar : String, in TerNumCelNue : String, in TerIdeCli : String, in TerProBan : String, inout RespIsTranOk : Boolean, inout DataOutput : String)
-sMsgClaHex(in TerClave : String) : String
+Suscribe(in TerNumCel : String, in TerNumTar : String, in TerClave : String, in TerTransaccion : Recordset, in TerProBan : String, in TerEsLocal : String, in TerCodSer : String, in TerCodRedAdq : String, in TerIdeDisOri : String, in TerUsuOri : String, inout DataOutput : String)
+Password(in TerNumCel : String, in TerNumTar : String, in TerClaveAnt : String, in TerClaveNue : String, in TerTransaccion : Recordset, in TerProBan : String, in TerEsLocal : String, in TerCodSer : String, in TerCodRedAdq : String, in TerIdeDisOri : String, in TerUsuOri : String, inout DataOutput : String)
+SuscribeAutorizada(in TerNumCel : String, in TerNumTar : String, in TerCodOpeSE : String, in TerClave : String, in TerTransaccion : Recordset, in TerProBan : String, in TerEsLocal : String, in TerCodSer : String, in TerCodRedAdq : String, in TerIdeDisOri : String, in TerUsuOri : String, inout DataOutput : String)
-VerificaTransaccion(in TerNumCel : String, in TerNumTar : String, in TerCodSer : String, in TerCodOpe : String, in TerTieCla : String, in TerCla : String, in TerTieAut : String, in TerProBan : String, inout RespTarHab : String, inout RespTipTarHab : String, inout RespIdeCli : String, inout ResplsTranOk : Boolean, inout D
-VerificaTransaccionCambioClave(in TerNumCel : String, in TerNumTar : String, in TerCodSer : String, in TerCodOpe : String, in TerTieCla : String, in TerCla : String, in TerTieAut : String, in TerProBan : String, inout RespTarHab : String, inout RespTipTarHab : String, inout RespIdeCli : String, inout RespTerPag : Strin
-ObtieneComision(in TerEsLocal : String, in TerProBan : String, in TerTipTarHab : String, in TerTransaccion : Recordset, inout TerValComAdq : Double, inout TerValComAut : Double)
+SaldoTarCre(in TerNumCel : String, in TerNumTar : String, in TerCla : String, in TerTransaccion : Recordset, in TerProBan : String, in TerEsLocal : String, in TerCodSer : String, in TerCodRedAdq : String, in TerIdeDisOri : String, in TerUsuOri : String, inout DataOutput : String)
+InactivaSuscribeAutorizada(in TerNumCel : String, in TerNumTar : String, in TerCodOpeSE : String, in TerCla : String, in TerTransaccion : Recordset, in TerProBan : String, in TerEsLocal : String, in TerCodSer : String, in TerCodRedAdq : String, in TerIdeDisOri : String, in TerUsuOri : String, in TerCodMot : String, ino
+InactivaSuscribe(in TerNumCel : String, in TerNumTar : String, in TerCla : String, in TerTransaccion : Recordset, in TerProBan : String, in TerEsLocal : String, in TerCodSer : String, in TerCodRedAdq : String, in TerIdeDisOri : String, in TerUsuOri : String, in TerCodMot : String, inout DataOutput : String)
-BinTar(in TerNumTar : String, inout RespBinTar : String, inout RespEsLocal : String, inout RespProSMS : String, inout ResplsTranOk : Boolean, inout RespData : String)
-RecuperaTranOriSerEle(in TerNumTar : String, in TerFecTra : String, in TerHorTra : String, in TerCodOpe : String, inout rsTraSerEle : Recordset, inout RespIsTranOk : Boolean, inout DataOutput : String)
+ReversaSuscribe(in TerNumCel : String, in TerNumTar : String, in TerClave : String, in TerTransaccion : Recordset, in TerProBan : String, in TerEsLocal : String, in TerCodSer : String, in TerCodRedAdq : String, in TerIdeDisOri : String, in TerUsuOri : String, in TerSupOri : String, in TerSupNom : String, inout DataOut
+ReversaSuscribeAutorizada(in TerNumCel : String, in TerNumTar : String, in TerCodOpeSE : String, in TerClave : String, in TerTransaccion : Recordset, in TerProBan : String, in TerEsLocal : String, in TerCodSer : String, in TerCodRedAdq : String, in TerIdeDisOri : String, in TerUsuOri : String, in TerSupOri : String, ir
-DescripcionTarjeta(in TerProBan : String) : String
+CupoTarCre(in TerNumCel : String, in TerNumTar : String, in TerTipCup : String, in TerCla : String, in TerTransaccion : Recordset, in TerProBan : String, in TerEsLocal : String, in TerCodSer : String, in TerCodRedAdq : String, in TerIdeDisOri : String, in TerUsuOri : String, inout DataOutput : String)
+ConsultaCheque(in TerNumCel : String, in TerNumTar : String, in TerNumChe : String, in TerCla : String, in TerTransaccion : Recordset, in TerProBan : String, in TerEsLocal : String, in TerCodSer : String, in TerCodRedAdq : String, in TerIdeDisOri : String, in TerUsuOri : String, inout DataOutput : String)
+BloqueoTarjeta(in TerNumCel : String, in TerNumTar : String, in TerClaveSE : String, in TerCla : String, in TerTransaccion : Recordset, in TerProBan : String, in TerEsLocal : String, in TerCodSer : String, in TerCodRedAdq : String, in TerIdeDisOri : String, in TerUsuOri : String, inout DataOutput : String)
+SaldoTarDeb(in TerNumCel : String, in TerNumTar : String, in TerTipCue : String, in TerCla : String, in TerTransaccion : Recordset, in TerProBan : String, in TerEsLocal : String, in TerCodSer : String, in TerCodRedAdq : String, in TerIdeDisOri : String, in TerUsuOri : String, inout DataOutput : String)
+RecargaCelular(in TerNumCel : String, in TerNumTar : String, in TerTipCue : String, in TerCla : String, in TerTransaccion : Recordset, in TerProBan : String, in TerEsLocal : String, in TerCodSer : String, in TerCodRedAdq : String, in TerIdeDisOri : String, in TerUsuOri : String, in TerValRec : String, in TerCelDes : Strin
+PULLSinSuscripcion(in TerNumCel : String, in TerNumTar : String, in TerCla : String, in TerTransaccion : Recordset, in TerProBan : String, in TerEsLocal : String, in TerCodSer : String, in TerCodRedAdq : String, in TerIdeDisOri : String, in TerUsuOri : String, in DataInput : String, inout DataOutput : String)
+NotificacionOLBatchGenericaConSuscripcion(in TerTipPas : String, in TerCodOpeOri : String, in TerNumCel : String, in TerNumTar : String, in TerCla : String, in TerTransaccion : Recordset, in TerProBan : String, in TerEsLocal : String, in TerCodSer : String, in TerCodRedAdq : String, in TerIdeDisOri : String, in TerU:
+NotificacionOLBatchGenerica(in TerTipPas : String, in TerCodOpeOri : String, in TerNumCel : String, in TerNumTar : String, in TerCla : String, in TerTransaccion : Recordset, in TerProBan : String, in TerEsLocal : String, in TerCodSer : String, in TerCodRedAdq : String, in TerIdeDisOri : String, in TerUsuOri : String, in
+Modifica(in TerNumCel : String, in TerNumTar : String, in TerNumCelNue : String, in TerClave : String, in TerTransaccion : Recordset, in TerProBan : String, in TerEsLocal : String, in TerCodSer : String, in TerCodRedAdq : String, in TerIdeDisOri : String, in TerUsuOri : String, inout DataOutput : String)
+ConsultaSuscripcionesSE(in TerNumCel : String, in TerNumTar : String, in TerCodSer : String, in TerFecSusIni : String, in TerFecSusFin : String, inout RespIsTranOk : Boolean, inout DataOutput : String) : Recordset
+ConsultaSusAutSE(in TerCodSus : Long, in TerNumCel : String, in TerNumTar : String, in TerCodSer : String, in TerCodRedAdq : String, inout RespIsTranOk : Boolean, inout DataOutput : String) : Recordset
+ConsultaTransaccionesSE(in TerNumCel : String, in TerNumTar : String, in TerCodSer : String, in TerCodRedAdq : String, in TerFecTraIni : String, in TerFecTraFin : String, in TerTipCue : Integer, in TerNumCue : String, in TerTipTra : String, inout RespIsTranOk : Boolean, inout DataOutput : String) : Recordset
+ConsultaClaveSE(in TerNumCel : String, in TerNumTar : String, in TerCla : String, in TerTransaccion : Recordset, in TerProBan : String, in TerEsLocal : String, in TerCodSer : String, in TerCodRedAdq : String, in TerIdeDisOri : String, in TerUsuOri : String, in TerSupOri : String, in TerSupNom : String, inout DataOutpu
-SetConeccionServiciosE(inout sMensaje : String, inout IRetornoServiciosE : Long)
-SetConeccionServiciosE2(inout sMensaje : String, inout IRetornoServiciosE2 : Long)
+ObtieneTransaccion(in TerCodSer : String, in TerCodRedAdq : String, in TerBinTar : String, in TerEsLocal : String, in TerCodOpe : String, inout RespIsTranOk : Boolean, inout RespData : String) : Recordset
-Class_Initialize()
-GrabaErrorExcepcion(in strFileName : String, in strPrograma : String, in strMetFunSub : String, in strMsgErr : String, in strMenErrSis : String)
-SetConeccionServicios_R(inout sMensaje : String, inout IRetornoServiciosR : Long)
-SetConeccionServicios_SMS(inout sMensaje : String, inout IRetornoServiciosSMS : Long)
+ConsultaRemesas_SE(inout RespIsTranOk : Boolean, inout DataOutput : String) : Recordset
+ActualizaRemesas_SE(in band : Boolean, in codRemesadora : String, in codRemesa : String, in valor : String, inout RespIsTranOk : Boolean, inout DataOutput : String, in Fecha : String, in hora : String, in Observacion : String, in Estado : String, in Usuario : String, in Error : String)
+RecuperaError(in TerRemCod : String, in TerRemCodErrBan : String, inout RespRemGraStaTra : String, inout RespRemCodComLla : String, inout RespRemDesComLla : String, inout RespIsTranOk : Boolean, inout RespData : String)
-EnviaSMS(in TerNumCel : String, in TerMensaje : String, in TerOrigen : String, in TerTipo : String, inout DataOutput : String, inout RespIsTranOk : Boolean)
+RecuperarCelular(in Tarjeta : String, inout RespIsTranOk : Boolean, inout DataOutput : String, inout Celular : String)
+ConsultarExcepcionOD(in OrigenLink : String, in DestLink : String, inout RespIsTranOk : Boolean, inout DataOutput : String)
+VerificarExcepcion(in Tarjeta : String, in Operacion : String, inout RespIsTranOk : Boolean, inout DataOutput : String, inout Verificar : Boolean)
-FechaSistema() : String
-HoraSistema() : String
-Class_Terminate()
+ConsultaTpbPushBatch_SE(inout RespIsTranOk : Boolean, inout DataOutput : String) : Recordset
+ActualizaTPBPushBatch_SE(in band : String, in codTPB : Long, inout RespIsTranOk : Boolean, inout DataOutput : String, in codRet : String, in Error : String)
+ConfiguraTransaccionSerElec_PushBatch(in TerCodTra : String, in TerDiasSem : String, in TerFecIni : String, in TerHorIni : String, in TerHorFin : String, in TerEstado : String, inout RespIsTranOk : Boolean, inout DataOutput : String)
+ConsultaTTTransaccion(in TerCodTra : String, inout RespIsTranOk : Boolean, inout DataOutput : String) : Recordset

ng, in TerCla : String, in TerTransaccion : Recordset, inout RespIsTranOk : Boolean, inout DataOutput : String)
ımCel : String, in TerProBan : String, in TerNumTar : String, in TerTipTarHab : String, inout RespIsTranOk : Boolean, inout DataOutput : String, in TerUsuDes : String = "", in TerMenDes : String = "", in TerInfo1 : String = "", in TerInfo2 : String = "")
tput : String)

TerValComAdq : Double, in TerValComAut : Double, in TerValComTot : Double, in TerNumAut : String, in TerCodRetBan : Integer, in TerCodRetSMS : String, in TerCodRetOpe : Integer, in TerMsgErr : String, in TerMenDes : String, inout RespIsTranOk : Boolean, inout DataOutput : String, in TerTipPas : String = "")
ın : String, in TerCodMot : String, inout RespIsTranOk : Boolean, inout DataOutput : String)
ısaccion : Recordset, in TerCodSus : Long, inout RespIsTranOk : Boolean, inout DataOutput : String)

ɔataOutput : String)
ıg, inout RespIsTranOk : Boolean, inout DataOutput : String)

ıut DataOutput : String)

put : String)
n TerSupNom : String, inout DataOutput : String)

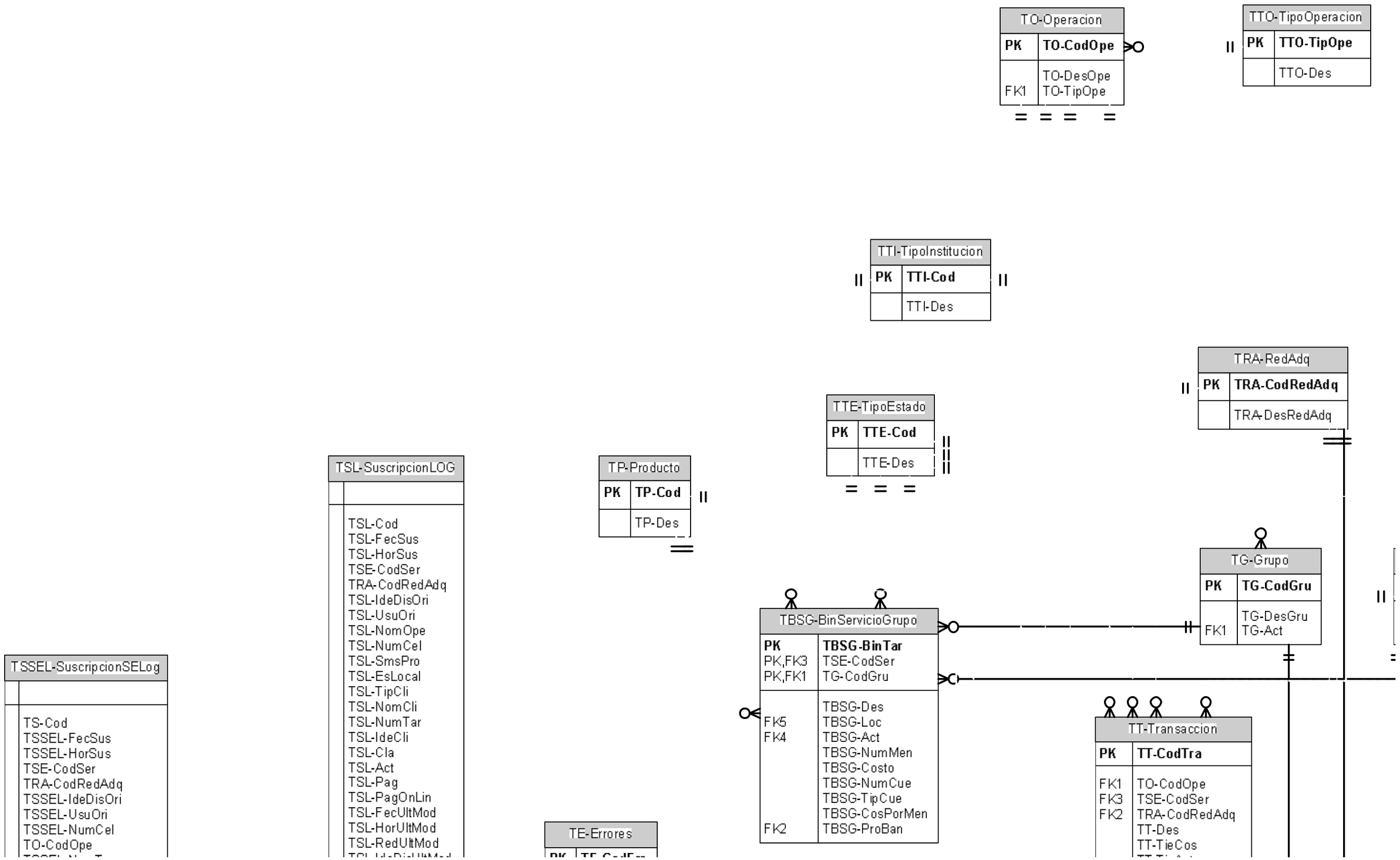
ıg, inout DataOutput : String)

suOri : String, in TerMensaje : String, in TerInf1 : String, in TerInf2 : String, inout DataOutput : String)
ı TerMensaje : String, in TerNomUsuCel : String, in TerInf1 : String, in TerInf2 : String, inout DataOutput : String)

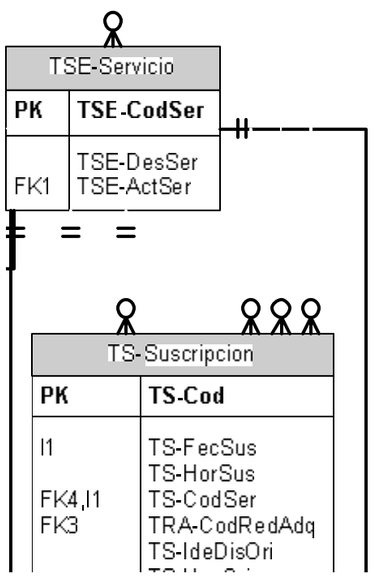
ıt : String)

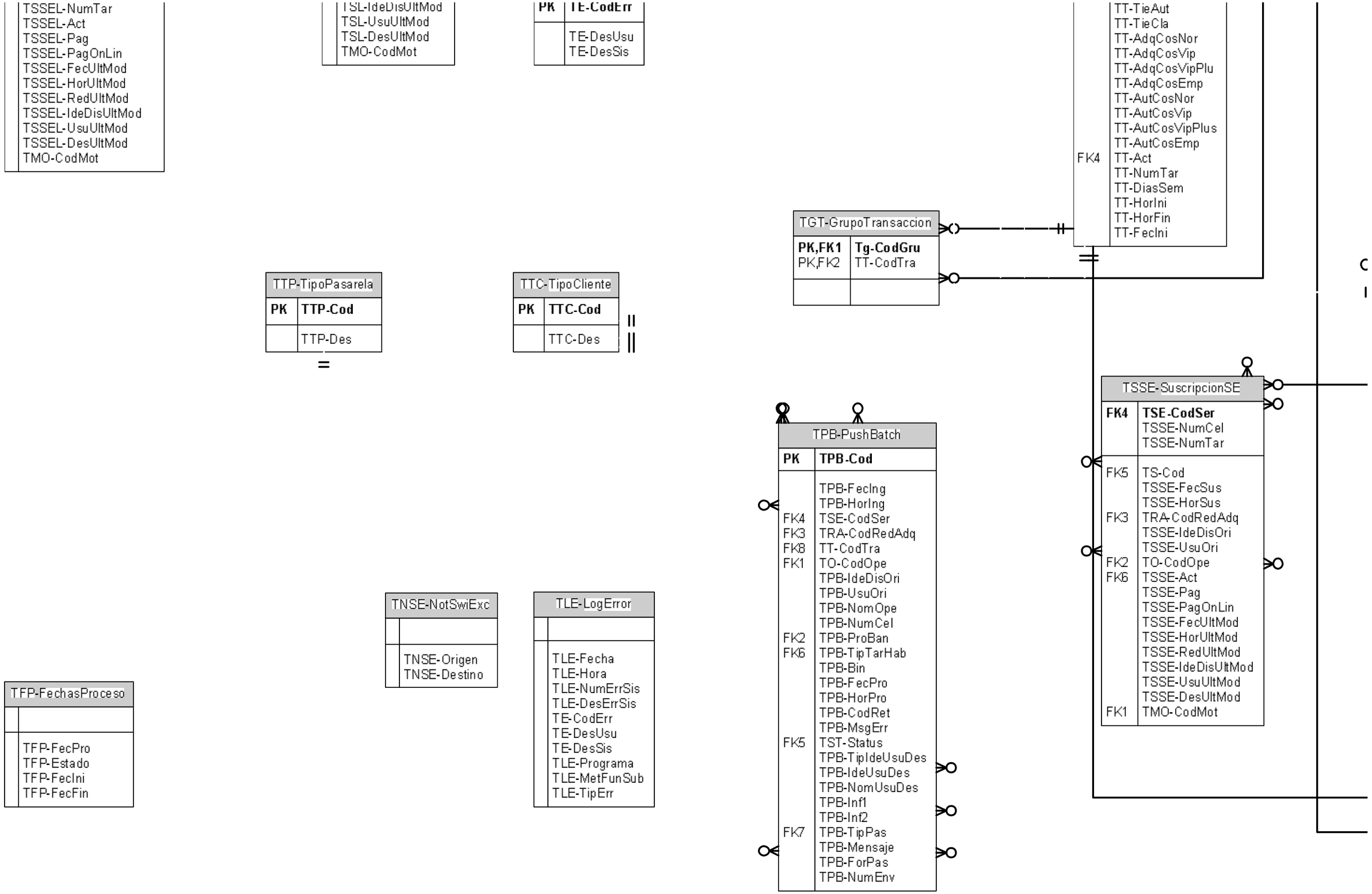
ANEXO 4.4 DIAGRAMA OBJETO-RELACION

ANEXO 4.4 DIAGRAMA OBJETO – RELACIÓN DE LA BASE DE DATC



)S DBSERVICIOSE




TSSEL-NumTar
TSSEL-Act
TSSEL-Pag
TSSEL-PagOnLin
TSSEL-FecUltMod
TSSEL-HorUltMod
TSSEL-RedUltMod
TSSEL-IdeDisUltMod
TSSEL-UsuUltMod
TSSEL-DesUltMod
TMO-CodMot
TSL-IdeDisUltMod
TSL-UsuUltMod
TSL-DesUltMod
TMO-CodMot
PK TE-CodErr
TE-DesUsu
TE-DesSis
FK4
TT-TieAut
TT-TieCla
TT-AdqCosNor
TT-AdqCosVip
TT-AdqCosVipPlu
TT-AdqCosEmp
TT-AutCosNor
TT-AutCosVip
TT-AutCosVipPlus
TT-AutCosEmp
TT-Act
TT-NumTar
TT-DiasSem
TT-HorIni
TT-HorFin
TT-FecIni
TGT-GrupoTransaccion
PK,FK1 Tg-CodGru
PK,FK2 TT-CodTra
TTP-TipoPasarela
PK TTP-Cod
TTP-Des
TTC-TipoCliente
PK TTC-Cod
TTC-Des
TSSE-SuscripcionSE
FK4 TSE-CodSer
TSSE-NumCel
TSSE-NumTar
FK5 TS-Cod
TSSE-FecSus
TSSE-HorSus
FK3 TRA-CodRedAdq
TSSE-IdeDisOri
TSSE-UsuOri
FK2 TO-CodOpe
FK6 TSSE-Act
TSSE-Pag
TSSE-PagOnLin
TSSE-FecUltMod
TSSE-HorUltMod
TSSE-RedUltMod
TSSE-IdeDisUltMod
TSSE-UsuUltMod
TSSE-DesUltMod
FK1 TMO-CodMot
TPB-PushBatch
PK TPB-Cod
TPB-FecIng
TPB-HorIng
FK4 TSE-CodSer
FK3 TRA-CodRedAdq
FK8 TT-CodTra
FK1 TO-CodOpe
TPB-IdeDisOri
TPB-UsuOri
TPB-NomOpe
TPB-NumCel
FK2 TPB-ProBan
FK6 TPB-TipTarHab
TPB-Bin
TPB-FecPro
TPB-HorPro
TPB-CodRet
TPB-MsgErr
FK5 TST-Status
TPB-TipIdeUsuDes
TPB-IdeUsuDes
TPB-NomUsuDes
TPB-Inf1
TPB-Inf2
FK7 TPB-TipPas
TPB-Mensaje
TPB-ForPas
TPB-NumEnv
TNSE-NotSwiExc
TNSE-Origen
TNSE-Destino
TLE-LogError
TLE-Fecha
TLE-Hora
TLE-NumErrSis
TLE-DesErrSis
TE-CodErr
TE-DesUsu
TE-DesSis
TLE-Programa
TLE-MetFunSub
TLE-TipErr
TFP-FechasProceso
TFP-FecPro
TFP-Estado
TFP-FecIni
TFP-FecFin

	TS-UsuOri
	TS-NomOpe
I1	TS-NumCel
FK2,I2	TS-SmsPro
FK7	TS-EsLocal
FK5	TS-TipCli
	TS-NomCli
I1	TS-NumTar
I3,I2	TS-IdeCli
	TS-Cla
FK6,I3,I2	TS-Act
	TS-Pag
	TS-PagOnLin
	TS-FecUltMod
	TS-HorUltMod
	TS-RedUltMod
	TS-IdeDisUltMod
	TS-UsuUltMod
	TS-DesUltMod
FK1	TMO-CodMot

TST-Status	
PK	**TST-Status**
	TST-StaDes

TMO-Motivo	
PK	**TMO-CodMot**
	TMO-DesMot TMO-TipMot

TSET-SerEleTran	
PK,I1	**TSET-FecTra**
PK	TSET-HorTra
PK,FK4,I1	TSE-CodSer
PK,FK3	TRA-CodRedAdq
PK,FK7	TT-CodTra
PK	TSET-IdeDisOri
PK	TSET-UsuOri
PK,I1	TSET-NumCel
PK,I1	TSET-NumTar
PK	TO-CodOpeSA
FK1	TO-CodOpe
	TSET-SupOri
	TSET-SupNom
	TSET-NomOpe
FK2	TSET-ProBan
	TSET-TipCueDes
I1	TSET-CueDes
FK6	TSET-TipTarHab
	TSET-ComAdq
	TSET-ComAut
	TSET-ComTot
	TSET-NumAut
	TSET-CodRetBan
	TSET-CodRetSMS
	TSET-CodRetOpe
	TSET-MsgErr
FK5,I1	TSET-Status
	TSET-UsuDes
	TSET-MenDes
	TSET-Inf1
	TSET-Inf2
	TSET-TipPas

ANEXO 4.5 DICCIONARIO DE BASE DE DATOS

ANEXO 4.5 DICCIONARIO DE BASE DE DATOS DBSERVICIOSSE

TABLA	ATRIBUTO	TIPO DATO	PK	FK	DESCRIPCION	NULL	DOMINIO
TBSG-BinServicioGrupo	**Registra los bines de las tarjetas de débito/crédito autorizados para el servicio banca celular**						
	[TBSG-BinTar]	[varchar] (19)	PK		Bin de tarjeta		
	[TSE-CodSer]	[char] (3)	PK	FK	Codigo de servicio	No	
	[TG-CodGru]	[char] (9)	PK	FK	Codigo de grupo	No	
	[TBSG-Des]	[varchar] (50)			Descripcion del BIN	No	
	[TBSG-Loc]	[char] (1),		FK	Indica si el bin pertenece a la institucion local o a una institucion afiliada a la institucion local.	No	1=Es local; 0 = Intitucion afiliada
	[TBSG-Act]	[char] (1)		FK	Indica si el bin esta activo.	No	0=Inactiva; 1=Activa
	[TBSG-NumMen]	[int]			Numero de mensajes del paquete para el cobro	Si	
	[TBSG-Costo]	[float]			Costo mensual del servicio	Si	
	[TBSG-NumCue]	[varchar] (10)			Número de cuenta desde donde debe ser debitado los valores por comisión. Aplica para bines de instituciones afiliadas.	Si	
	[TBSG-TipCue]	[char] (1)			Tipo de cuenta desde donde debe ser debitado los valores por comisión. Aplica para bines de instituciones afiliadas.	Si	A=Ahorros;C=Cuenta corriente
	[TBSG-CosPorMen]	[float] NULL			Valor por mensaje adicional	Si	
	[TBSG-ProBan]	[char] (2)		FK	Tipo de producto al que pertnece el BIN	No	N=Normal;E=Empleado;

[TE-Errores]	**Registra los errores definidos para banca celular**						
	[TE-CodErr]	[int]	PK		Código de error	Si	
	[TE-DesUsu]	[varchar] (138)			Mensaje del error que se presentará al sistema	No	
	[TE-DesSis]	[varchar] (255)			Mensaje del error de sistema	No	
[TG-Grupo]	**Define los grupos de transacciones**						
	[TG-CodGru]	[char] (9)	PK		Código de grupo	No	
	[TG-DesGru]	[char] (50)			Nombre del grupo	No	
	[TG-Act]	[char] (1)	FK		Estado que indica si el grupo esta activo o no	No	1=Activo;0=Inactivo
[TGT-GrupoTransaccion]	**Determina las transacciones autorizadas para un determinado grupo**						
	[Tg-CodGru]	[char] (9)	PK	FK	Código del grupo	No	
	[TT-CodTra]	[char] (5)	PK	FK	Código de transacción	No	
[TLE-LogError]	**Bitácora de errores presentados en la operación del sistema**						
	[TLE-Fecha]	[smalldatetime]			Fecha de error	No	
	[TLE-Hora]	[char] (8)			Hora de error	No	
	[TLE-NumErrSis]	[bigint] NULL ,			Error de excepcion, en caso de ser error de sistema	Si	
	[TLE-DesErrSis]	[varchar] (255)			Descripcion de excepcion, en caso de ser error de sistema	Si	
	[TE-CodErr]	[int] NULL ,			Código de error presentado	Si	
	[TE-DesUsu]	[varchar] (138)			Mensaje de error presentado al usuario	Si	
	[TE-DesSis]	[varchar] (255)			Mensaje de error del sistema	Si	

	[TLE-Programa]	[varchar] (100)			Programa en donde se produce el error	Si	
	[TLE-MetFunSub]	[varchar] (100)			Método, función o subrutina	Si	
	[TLE-TipErr]	[char] (1)			Tipo de error presentado	No	S=Sistema;O=Operativo
[TMO-Motivo]	**Registra los motivos para la inactivación y reverso de una suscripción**						
	[TMO-CodMot]	[char] (2)	PK		Código de motivo	No	
	[TMO-DesMot]	[varchar] (50)			Descripción de motivo	No	
	[TMO-TipMot]	[char] (1)			Tipo de motivo	No	S=Sistema;U=Usuario
[TNSE-NotSwiExc]	**Registra las redes adquiriente y autorizadora que no deben generar notificacion PUSH**						
	[TNSE-Origen]	[varchar] (20)	PK		Nombre del link origen	No	
	[TNSE-Destino]	[varchar] (20)	PK		Nombrel del link destino	No	
[TO-Operacion]	**Registra las operaciónes PULL, PUSH del servicio banca celular**						
	[TO-CodOpe]	[char] (5)	PK		Código de operación	No	
	[TO-DesOpe]	[varchar] (50)			Nombre de operación	No	
	[TO-TipOpe]	[char] (1)		FK	Tipo de operación	Si	
[TPB-PushBatch]	**Registra las notificacion que se desean enviar a un celular**						
	[TPB-Cod]	[bigint] IDENTITY	PK		Código de notificación. Es autonumérico.	No	
	[TPB-FecIng]	[smalldatetime]			Fecha de ingreso de la notificación	No	
	[TPB-HorIng]	[char] (8)			Hora de ingreso de la notificación	No	

	[TSE-CodSer]	[char] (3)		FK	Codigo de servicio	No	
	[TRA-CodRedAdq]	[char] (3)		FK	Codigo de red adquiriente	No	
	[TT-CodTra]	[char] (5)		FK	Código de transacción	No	
	[TO-CodOpe]	[char] (5)		FK	Código de operación	No	
	[TPB-IdeDisOri]	[varchar] (15)			Identificación de dispositivo que carga la notificación	No	
	[TPB-UsuOri]	[varchar] (10)			Nombre del usuario que carga la transacción	No	
	[TPB-NomOpe]	[varchar] (10)			Nombre de la operadora	No	
	[TPB-NumCel]	[char] (9)			Número celular	No	
	[TPB-ProBan]	[char] (2)		FK	Código de producto bancario	No	
	[TPB-TipTarHab]	[char] (1)		FK	Tipo de tarjeta habiente	No	
	[TPB-Bin]	[varchar] (19)			Bin de la tarjeta	No	
	[TPB-FecPro]	[smalldatetime]			Fecha de envío		
	[TPB-HorPro]	[char] (8)			Hora de envío	Si	
	[TPB-CodRet]	[varchar] (5)			Código de retorno	Si	
	[TPB-MsgErr]	[varchar] (200)			Mensaje de error presentado al enviar notificación	Si	
	[TST-Status]	[char] (1)		FK	Estado de la notificación	No	E=Enviado;M=Error;P=Pendiente de envío
	[TPB-TipIdeUsuDes]	[varchar] (150)			Tipo de identificación del usuario destino	Si	C=Cédula;R=Ruc;P=Pasaporte
	[TPB-IdeUsuDes]	[varchar] (150)			Identificación del usuario destino	Si	
	[TPB-NomUsuDes]	[varchar] (150)			Nombre del usuario destino	Si	
	[TPB-Inf1]	[varchar] (20)			Campo informativo	Si	

	[TPB-Inf2]	[varchar] (20)			Campo informativo	Si	
	[TPB-TipPas]	[char] (1)		FK	Tipo de pasarela	No	L=Pasarela SMS;E=Correo electrónico;S=El sistema selecciona por donde enviar
	[TPB-Mensaje]	[varchar] (150)			Mensaje SMS	No	
	[TPB-ForPas]	[char] (1)			Forza la pasarela	No	0=No sugiere la pasarela de envío;1=Sugiere la pasarela de envío
	[TPB-NumEnv]	[varchar] (50)			Número de mensajes enviados	Si	
[TRA-RedAdq]	**Registra las redes adquirientes**						
	[TRA-CodRedAdq]	[char] (3)	PK		Código de red adquiriente	No	
	[TRA-DesRedAdq]	[varchar] (50)			Nombre de red adquiriente	No	
[TS-Suscripcion]	**Registra las suscripciones de los clientes al servicio banca celular**						
	[TS-Cod]	[bigint] IDENTITY	PK		Código autonumérico	No	
	[TS-FecSus]	[smalldatetime]			Fecha de suscripción	No	
	[TS-HorSus]	[char] (8)			Hora de suscripción	No	
	[TS-CodSer]	[char] (3)		FK	Código de servicio	No	
	[TRA-CodRedAdq]	[char] (3)		FK	Código de red adquiriente	No	
	[TS-IdeDisOri]	[varchar] (15)			Identificación de dispositivo origen	Si	

	[TS-UsuOri]	[varchar] (10)			Nombre del usuario que realiza la suscripción	Si	
	[TS-NomOpe]	[varchar] (10)			Nombre de operadora del celular	No	
	[TS-NumCel]	[char] (9)			Número del celular	No	
	[TS-SmsPro]	[char] (2)		FK	Producto de suscripción	No	TD=Tarjeta dédito;VI=Tarjeta de crédito Visa;MC=MasterCard;TC=Tarjeta club
	[TS-EsLocal]	[char] (1)		FK	Indica si la suscripcion es de un cliente local o de una institución afiliada	No	0=Local;1=Institución afiliada
	[TS-TipCli]	[char] (1)		FK	Tipo de cliente	No	C=Cédula;R=Ruc;P=Pasaporte
	[TS-NomCli]	[varchar] (30)			Nombre del dueño de la tarjeta de suscripción	No	
	[TS-NumTar]	[varchar] (19)			Número de tarjeta	No	
	[TS-IdeCli]	[varchar] (13)			Identificacion del cliente	No	
	[TS-Cla]	[binary] (20)			Clave de servicio banca celular	Si	
	[TS-Act]	[char] (1)		FK	Indica si la suscripción esta activa o no	No	1=Activa;0=Inactiva;2=Reverso
	[TS-Pag]	[char] (1)			Indica si el cliente paga en forma batch	No	
	[TS-PagOnLin]	[char] (1)			Indica si el cliente paga comisión en forma online	No	
	[TS-FecUltMod]	[smalldatetime]			Fecha de modificación	Si	
	[TS-HorUltMod]	[char] (8)			Hora de modificación	Si	
	[TS-RedUltMod]	[varchar] (3)			Red desde donde se realiza la modificación de la suscripción	Si	

	[TS-IdeDisUltMod]	[varchar] (15)			Identificación del dispositivo origen	Si	
	[TS-UsuUltMod]	[varchar] (10)			Usuario que realiza la modificación de la suscripción	Si	
	[TS-DesUltMod]	[varchar] (50)			Descripción de la modificación	Si	
	[TMO-CodMot]	[char] (2)		FK	Código del motivo de inactivación	No	
[TSE-Servicio]	**Registra el servicio soportado**						
	[TSE-CodSer]	[char] (3)			Código del servicio	No	
	[TSE-DesSer]	[char] (50)			Nombre del servicio	No	
	[TSE-ActSer]	[char] (1)			Estado del servicio	No	0=Inactivo;1=Activo
[TSET-SerEleTran]	**Bitácora transaccional de mensajes recibidos/enviados a los clientes, asi como registro de transacciones del backoffice**						
	[TSET-FecTra]	[smalldatetime]	PK		Fecha de transaccion	No	
	[TSET-HorTra]	[char] (8)	PK		Hora de transaccion	No	
	[TSE-CodSer]	[char] (3)	PK	FK	Código del servicio	No	
	[TRA-CodRedAdq]	[char] (3)	PK	FK	Código de red adquiriente	No	
	[TT-CodTra]	[char] (5)	PK	FK	Código de transacción	No	
	[TO-CodOpe]	[char] (5)		FK	Código de operación	No	
	[TSET-IdeDisOri]	[varchar] (15)	PK		Identificación del dispositivo origen	No	
	[TSET-UsuOri]	[varchar] (10)	PK		Usuario origen	No	
	[TSET-SupOri]	[varchar] (10)			Código del supervisor	Si	
	[TSET-SupNom]	[varchar] (40)			Nombre del suspervisor	Si	
	[TSET-NomOpe]	[varchar] (10)			Nombre de operadora	No	
	[TSET-NumCel]	[char] (9)	PK		Número de celular	No	

	[TSET-ProBan]	[char] (2)		FK	Codigo del producto asociado a la tarjeta de suscripción	No	
	[TSET-NumTar]	[varchar] (19)	PK		Número de tarjeta	No	
	[TO-CodOpeSA]	[char] (5)	PK		Código de operación de servicio adicional	No	
	[TSET-TipCueDes]	[int]			Tip de cuenta asociada a la suscripción	Si	A=Ahorros;C=Corriente
	[TSET-CueDes]	[char] (10)			Número de cuenta asociada a la suscripción	Si	
	[TSET-TipTarHab]	[char] (1)		FK	Tipo de tarjeta habiente	No	N=Normal:E=Empleado;V=Vip;P=Vip plus
	[TSET-ComAdq]	[float]			Comisión cobrada para la red adquiriente	No	
	[TSET-ComAut]	[float]			Comisión cobrada para la red autorizadora	No	
	[TSET-ComTot]	[float]			Comisión total	No	
	[TSET-NumAut]	[varchar] (10)			Número de auorización	Si	
	[TSET-CodRetBan]	[int]			Código de retorno de la operación bancaria	Si	
	[TSET-CodRetSMS]	[char] (3)			Códido de retorno entregado a la pasarela	Si	
	[TSET-CodRetOpe]	[int]			Código de retorno de la operadora celular	Si	
	[TSET-MsgErr]	[varchar] (255)			Mensaje de error	Si	
	[TSET-Status]	[char] (1)		FK	Estatus	No	
	[TSET-UsuDes]	[varchar] (150)			Usuario destino	Si	
	[TSET-MenDes]	[text]			Mensaje enviado al destino	Si	
	[TSET-Inf1]	[varchar] (255)			Campo informativo	Si	

	[TSET-Inf2]	[varchar] (255)			Campo informativo	Si	
	[TSET-TipPas]	[char] (1)			Tipo de pasarela utilizada	Si	
[TSL-SuscripcionLOG]	**Log de todos los eventos realizados en las suscripciones de los clientes**						
	[TSL-Cod]	[bigint]			Código autonumérico	Si	
	[TSL-FecSus]	[smalldatetime]			Fecha de suscripción	No	
	[TSL-HorSus]	[char] (8)			Hora de suscripción	No	
	[TSE-CodSer]	[char] (3)			Código de servicio	No	
	[TRA-CodRedAdq]	[char] (3)			Código de red adquiriente	No	
	[TSL-IdeDisOri]	[varchar] (15)			Identificación de dispositivo origen	Si	
	[TSL-UsuOri]	[varchar] (10)			Nombre del usuario que realiza la suscripción	Si	
	[TSL-NomOpe]	[varchar] (10)			Nombre de operadora del celular	No	
	[TSL-NumCel]	[char] (9)			Número del celular	No	
	[TSL-SmsPro]	[char] (2)			Producto de suscripción	No	TD=Tarjeta dédito;VI=Tarjeta de crédito Visa;MC=MasterCard;TC=Tarjeta club
	[TSL-EsLocal]	[char] (1)			Indica si la suscripcion es de un cliente local o de una institución afiliada	No	0=Local;1=Institución afiliada
	[TSL-TipCli]	[char] (1)			Tipo de cliente	No	C=Cédula;R=Ruc;P=Pasaporte
	[TSL-NomCli]	[varchar] (30)			Nombre del dueño de la tarjeta de suscripción	No	

	[TSL-NumTar]	[varchar] (19)			Número de tarjeta	No	
	[TSL-IdeCli]	[varchar] (13)			Identificacion del cliente	No	
	[TSL-Cla]	[binary] (50)			Clave de servicio banca celular	No	
	[TSL-Act]	[char] (1)			Indica si la suscripción esta activa o no	No	1=Activa;0=Inactiva;2=Reverso
	[TSL-Pag]	[char] (1)			Indica si el cliente paga en forma batch	No	
	[TSL-PagOnLin]	[char] (1)			Indica si el cliente paga comisión en forma online	No	
	[TSL-FecUltMod]	[smalldatetime]			Fecha de modificación	Si	
	[TSL-HorUltMod]	[char] (8)			Hora de modificación	Si	
	[TSL-RedUltMod]	[varchar] (3)			Red desde donde se realiza la modificación de la suscripción	Si	
	[TSL-IdeDisUltMod]	[varchar] (15)			Identificación del dispositivo origen	Si	
	[TSL-UsuUltMod]	[varchar] (10)			Usuario que realiza la modificación de la suscripción	Si	
	[TSL-DesUltMod]	[varchar] (50)			Descripción de la modificación	Si	
	[TMO-CodMot]	[char] (2)			Código del motivo de inactivación	Si	
[TSSE-SuscripcionSE]	**Registra las suscripciones de los clientes a servicios adicionales banca celular**						
	[TS-Cod]	[bigint]		FK	Código de suscripción banca celular	No	
	[TSSE-FecSus]	[smalldatetime]			Fecha de suscripción	No	
	[TSSE-HorSus]	[char] (8)			Hora de suscripción	No	
	[TSE-CodSer]	[char] (3)		FK	Código de servicio	No	
	[TRA-CodRedAdq]	[char] (3)		FK	Código de red adquiriente	No	

	[TSSE-IdeDisOri]	[varchar] (15)			Identificación de dispositivo origen	Si	
	[TSSE-UsuOri]	[varchar] (10)			Usuario que origina la suscripción	Si	
	[TSSE-NumCel]	[char] (9)			Número de celular	No	
	[TO-CodOpe]	[char] (5)		FK	Código de operación de suscripción adicional	No	
	[TSSE-NumTar]	[varchar] (19)			Número de tarjeta	No	
	[TSSE-Act]	[char] (1)		FK	Indica si la suscripción al servicio adicional esta activa o no	No	0=Inactiva;1=Activa
	[TSSE-Pag]	[char] (1)			Indica si el pago de comisiones es en batch	No	
	[TSSE-PagOnLin]	[char] (1)			Indica si el pago de comisiones es en línea	No	
	[TSSE-FecUltMod]	[smalldatetime]			Fecha de última modificación de la suscripción	Si	
	[TSSE-HorUltMod]	[char] (8)			Hora de última modificación de la suscripción	Si	
	[TSSE-RedUltMod]	[char] (3)			Red adquiriente desde donde se origino la modificación	Si	
	[TSSE-IdeDisUltMod]	[varchar] (15)			Dispositivo desde donde se originó la modificación	Si	
	[TSSE-UsuUltMod]	[varchar] (10)			Usuario que modificó la transacción	Si	
	[TSSE-DesUltMod]	[varchar] (50)			Descripción de la modificación realizada sobre el registro	Si	
	[TMO-CodMot]	[char] (2)		FK	Código del motivo de inactivacion o reverso	No	
[TSSEL-SuscripcionSELog]	**Bitácora de cambios realizados sobre los registros de suscripción a servicios adicionales banca celular**						

	[TS-Cod]	[bigint]			Código de suscripción banca celular	Si	
	[TSSEL-FecSus]	[smalldatetime]			Fecha de suscripción	No	
	[TSSEL-HorSus]	[char] (8)			Hora de suscripción	No	
	[TSE-CodSer]	[char] (3)			Código de servicio	No	
	[TRA-CodRedAdq]	[char] (3)			Código de red adquiriente	No	
	[TSSEL-IdeDisOri]	[varchar] (15)			Identificación de dispositivo origen	No	
	[TSSEL-UsuOri]	[varchar] (10)			Usuario que origina la suscripción	No	
	[TSSEL-NumCel]	[char] (9)			Número de celular	No	
	[TO-CodOpe]	[char] (5)			Código de operación de suscripción adicional	No	
	[TSSEL-NumTar]	[varchar] (19)			Número de tarjeta	No	
	[TSSEL-Act]	[char] (1)			Indica si la suscripción al servicio adicional esta activa o no	No	0=Inactiva;1=Activa
	[TSSEL-Pag]	[char] (1)			Indica si el pago de comisiones es en batch	No	
	[TSSEL-PagOnLin]	[char] (1)			Indica si el pago de comisiones es en línea	No	
	[TSSEL-FecUltMod]	[smalldatetime]			Fecha de última modificación de la suscripción	Si	
	[TSSEL-HorUltMod]	[char] (8)			Hora de última modificación de la suscripción	Si	
	[TSSEL-RedUltMod]	[char] (3)			Red adquiriente desde donde se origino la modificación	Si	
	[TSSEL-IdeDisUltMod]	[varchar] (15)			Dispositivo desde donde se originó la modificación	Si	
	[TSSEL-UsuUltMod]	[varchar] (10)			Usuario que modificó la transacción	Si	

	[TSSEL-DesUltMod]	[varchar] (50)			Descripción de la modificación realizada sobre el registro	Si	
	[TMO-CodMot]	[char] (2)			Código del motivo de inactivacion o reverso	Si	
[TST-Status]	**Registra los estados de los registros banca celular**						
	[TST-Status]	[char] (1)	PK		Código de estado	No	
	[TST-StaDes]	[varchar] (50)			Descripción de estado	No	
[TT-Transaccion]	**Registra las transacciones banca celular**						
	[TT-CodTra]	[char] (5)	PK		Código de transaccion	No	
	[TO-CodOpe]	[char] (5)		FK	Código de operación	No	
	[TSE-CodSer]	[char] (3)		FK	Código de servicio	No	
	[TRA-CodRedAdq]	[char] (3)		FK	Código de red adquiriente	No	
	[TT-Des]	[char] (50)			Nombre de transacción	No	
	[TT-TieCos]	[char] (1)			Especifica si tiene costo la transacción	No	0=No;1=Si
	[TT-TieAut]	[char] (1)			Especifica si esta transaccion requiere suscripcion al servicio adicional	No	0=No;1=Si
	[TT-TieCla]	[char] (1)			Especifica si esta transaccion requiere especificacion de clave banca celular	No	0=No;1=Si
	[TT-AdqCosNor]	[float]			Comision de transaccion de red adquiriente de un cliente normal	Si	
	[TT-AdqCosVip]	[float]			Comision de transaccion de red adquiriente de un cliente vip	Si	
	[TT-AdqCosVipPlu]	[float]			Comision de transaccion de red adquiriente de un cliente vip plus	No	
	[TT-AdqCosEmp]	[float]			Comision de transaccion de red adquiriente de un cliente empleado	No	

	[TT-AutCosNor]	[float]			Comision de transaccion de red autorizadora de un cliente normal	Si	
	[TT-AutCosVip]	[float]			Comision de transaccion de red autorizadora de un cliente vip	Si	
	[TT-AutCosVipPlus]	[float]			Comision de transaccion de red autorizadora de un cliente vip plus	Si	
	[TT-AutCosEmp]	[float]			Comision de transaccion de red autorizadora de un cliente empleado	Si	
	[TT-Act]	[char] (1)		FK	Indica si la transacción esta activa o no	No	0=Inactiva;1=Activa
	[TT-NumTar]	[varchar] (19)			Número de tarjeta especial asociada a la transacción	Si	
	[TT-DiasSem]	[varchar] (7)			Dias de la semana autorizadas para notificacion tipo PUSH	Si	1234567
	[TT-HorIni]	[char] (8)			Hora de inicio de de notificacion tipo PUSH	Si	
	[TT-HorFin]	[char] (8)			Hora fin de notificacion tipo PUSH	Si	
	[TT-FecIni]	[smalldatetime]			Fecha de inicio de notificacion tipo PUSH	Si	
[TTO-TipoOperacion]	**Registra los tipos de operación PULL/PUSH**						
	[TTO-TipOpe]	[char] (1)	PK		Tipo de operación	No	
	[TTO-Des]	[varchar] (30)			Nombre del tipo de operación	No	
[TTC-TipoCliente]	**Registra los tipos de clientes**						
	[TTC-Cod	[char] (1)	PK		Tipo de cliente	No	
	[TTC-Des]	[varchar] (50)			Nombre del tipo de cliente	No	
[TP-Producto]	**Registra los tipos de productos de la institucion financiera**						
	[TP-Cod	[char] (2)	PK		Tipo de producto	No	

	[TP-Des]	[varchar] (50)			Nombre del producto	No	
[TTE-TipoEstado]	**Registra los estados de control**						
	[TTE-Cod	[char] (1)	PK		Código del estado	No	
	[TTE-Des]	[varchar] (50)			Descripción del estado	No	
[TTI-TipoInstitucion]	**Registra los tipos de institución**						
	[TTI-Cod	[char] (1)	PK		Código del tipo de institución	No	
	[TTI-Des]	[varchar] (50)			Descripción del tipo de institución	No	
[TTP-TipoPasarela]	**Registra los tipos de pasarela**						
	[TTP-Cod	[char] (1)	PK		Código de pasarela	No	
	[TTP-Des]	[varchar] (50)			Descripción del tipo de pasarela	No	

ANEXO 5.1 PRUEBAS DE RENDIMIENTO REALES

ANEXO 5.1 LOG DE PRUEBAS DE RENDIMIENTO REALES

Por tratarse de una bitácora de pruebas reales las respuestas de cada transacción han sido eliminadas por concepto de confidencialidad de la información.

FECHA: 2007/06/18 HORA: 06063300 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: SALDO099340381SMSSMS099340381
099340381 NNNNNN0010468002 0000
FECHA: 2007/06/18 HORA: 06124800 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: SALDO093793094SMSSMS093793094
093793094 NNNNNN0170142736 0000CA
FECHA: 2007/06/18 HORA: 06264300 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: SALDO095024651SMSSMS095024651
095024651 NNNNNN0170126424 0000CC
FECHA: 2007/06/18 HORA: 06264300 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: SALDO095024651SMSSMS095024651
095024651 NNNNNN0170096106 0000CC
FECHA: 2007/06/18 HORA: 06501100 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: SALDO085341279SMSSMS085341279
085341279 NNNNNN0070051243 0000
FECHA: 2007/06/18 HORA: 06574700 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: SALDO094999540SMSSMS094999540
094999540 NNNNNN0090120093 0000CC
FECHA: 2007/06/18 HORA: 07271800 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: S ALDO086265314SMSSMS086265314
086265314 NNNNNN0090077913 0000CA
FECHA: 2007/06/18 HORA: 07375300 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: SALDO093181623SMSSMS093181623
093181623 NNNNNN0090111241 0000CC
FECHA: 2007/06/18 HORA: 07375300 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: SALDO093181623SMSSMS093181623
093181623 NNNNNN0090129474 0000CC
FECHA: 2007/06/18 HORA: 07464400 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: SALDO093229632SMSSMS0 93229632
093229632 NNNNNN0090111159 0000CA
FECHA: 2007/06/18 HORA: 07490200 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: SALDO093962460SMSSMS093962460
093962460 NNNNNN0050037444 0000CC
FECHA: 2007/06/18 HORA: 07581900 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: SALDO097175074SMSSMS097175074
097175074 NNNNNN0470015137 0000
FECHA: 2007/06/18 HORA: 08000000 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: NOTON095138643SMSR EM095138643
095138643 9111110010080246 0000NOTREAVISO DE REMESAS: SR(A) ETELVINA CLAVIJO PAUCAR. FAVOR ACERCARSE A COBRAR SU GIRO EN EL INSTITUCION FINANCIERA ETELVINA CLAVIJO PAUCAR
L008
24202
FECHA: 2007/06/18 HORA: 08000000 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: NOTON092879585SMSR EM092879585
092879585 9111110010080246 0000NOTREAVISO DE REMESAS: SR(A) DOLORES SANISACA CHALCO. FAVOR ACERCARSE A COBRAR SU GIRO EN EL INSTITUCION FINANCIERA DOLORES SANISACA CHALCO
L008
24204
FECHA: 2007/06/18 HORA: 08000000 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: NOTON095017762SMSR EM095017762
095017762 9111110010080246 0000NOTREAVISO DE REMESAS: SR(A) JOSE VICENTE LATA. FAVOR ACERCARSE A COBRAR SU GIRO EN EL INSTITUCION FINANCIERA JOSE VICENTE LATA
L008
24198
FECHA: 2007/06/18 HORA: 08000000 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: NOTON092203702SMSR EM092203702
092203702 9111110010080246 0000NOTREAVISO DE REMESAS: SR(A) \ARIADEL PILAR ORTIZ PONCE. FAVOR ACERCARSE A COBRAR SU GIRO EN EL INSTITUCION FINANCIERA \ARIADEL PILAR ORTIZ PONCE
L008
24205
FECHA: 2007/06/18 HORA: 08000100 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: NOTON092246345SMSR EM0 92246345
092246345 9111110010080246 0000NOTREAVISO DE REMESAS: SR(A) MARGARITA APUANGO CAGUANA. FAVOR ACERCARSE A COBRAR SU GIRO EN EL INSTITUCION FINANCIERA MARGARITA APUANGO CAGUANA
L008

24208
FECHA: 2007/06/18 HORA: 08000100 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: NOTON092896792SMSR EM092896792 092896792 9111110010080246 0000NOTREAVISO DE REMESAS: SR(A) LUIS LEON CHALCO. FAVOR ACERCARSE A COBRAR SU GIRO EN EL INSTITUCION FINANCIERA LUIS LEON CHALCO
L008
24206
FECHA: 2007/06/18 HORA: 08000100 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: NOTON092914235SMSR EM092914235 092914235 9111110010080246 0000NOTREAVISO DE REMESAS: SR(A) LIDIA VEGA GUTIERREZ. FAVOR ACERCARSE A COBRAR SU GIRO EN EL INSTITUCION FINANCIERA LIDIA VEGA GUTIERREZ
L008
24209
FECHA: 2007/06/18 HORA: 08000100 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: NOTON092879315SMSR EM092879315 092879315 9111110010080246 0000NOTREAVISO DE REMESAS: SR(A) GLORIA ZUMBA ESPADERO. FAVOR ACERCARSE A COBRAR SU GIRO EN EL INSTITUCION FINANCIERA GLORIA ZUMBA ESPADERO
L008
24207
FECHA: 2007/06/18 HORA: 08000100 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: N OTON095121953SMSREM095121953 095121953 9111110010080246 0000NOTREAVISO DE REMESAS: SR(A) ROSA RAMON ULLOA. FAVOR ACERCARSE A COBRAR SU GIRO EN EL INSTITUCION FINANCIERA ROSA RAMON ULLOA
L008
24199
FECHA: 2007/06/18 HORA: 08000100 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: NOTON093117182SMSR EM093117182 093117182 9111110010080246 0000NOTREAVISO DE REMESAS: SR(A) JENNY MARITZA ZHANAY BARBECHO. FAVOR ACERCARSE A COBRAR SU GIRO EN EL INSTITUCION FINANCIERA JENNY MARITZA ZHANAY BARBECHO
L010
217627
FECHA: 2007/06/18 HORA: 08000200 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: NOTON084199236SMSR EM084199236 084199236 9111110010080246 0000NOTREAVISO DE REMESAS: SR(A) ALEJANDRA CAMPOVERDE NIAMA. FAVOR ACERCARSE A COBRAR SU GIRO EN EL INSTITUCION FINANCIERA ALEJANDRA CAMPOVERDE NIAMA
L010
217640
FECHA: 2007/06/18 HORA: 08000200 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: NOTON098137803SMSR EM098137803 098137803 9111110010080246 0000NOTREAVISO DE REMESAS: SR(A) BLANCA ROSARIO SANANGO SANANGO. FAVOR ACERCARSE A COBRAR SU GIRO EN EL INSTITUCION FINANCIERA BLANCA ROSARIO SANANGO SANANGO
L010
217585
FECHA: 2007/06/18 HORA: 08000200 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: NOTON099976701SMSR EM099976701 099976701 9111110010080246 0000NOTREAVISO DE REMESAS: SR(A) MARIA ADELINA MURUDUMBAY M.. FAVOR ACERCARSE A COBRAR SU GIRO EN EL INSTITUCION FINANCIERA MARIA ADELINA MURUDUMBAY M.
L010
217649
FECHA: 2007/06/18 HORA: 08000200 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: NOTON088767867SMSR EM088767867 088767867 9111110010080246 0000NOTREAVISO DE REMESAS: SR(A) PABLO JESUS QUIZHPI TENEZACA. FAVOR ACERCARSE A COBRAR SU GIRO EN EL INSTITUCION FINANCIERA PABLO JESUS QUIZHPI TENEZACA
L010
217696
FECHA: 2007/06/18 HORA: 08000300 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: NOTON092900923SMSR EM092900923 092900923 9111110010080246 0000NOTREAVISO DE REMESAS: SR(A) AMANDINA FENICIA CABRERA RODRIGUEZ. FAVOR ACERCARSE A COBRAR SU GIRO EN EL INSTITUCION FINANCIERA AMANDINA FENICIA CABRERA RODRIGUEZ
L010
217590
FECHA: 2007/06/18 HORA: 08000300 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: NOTON084224262SMSR EM084224262 084224262 9111110010080246 0000NOTREAVISO DE REMESAS: SR(A) MARIA ALCIRA HUERTA PERALTA. FAVOR ACERCARSE A COBRAR SU GIRO EN EL INSTITUCION FINANCIERA MARIA ALCIRA HUERTA PERALTA
L010
217591
FECHA: 2007/06/18 HORA: 08000300 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: NOTON092515408SMSR EM092515408 092515408 9111110010080246 0000NOTREAVISO DE REMESAS: SR(A) MAGDALENA BUNAY PIZHA. FAVOR ACERCARSE A COBRAR SU GIRO

EN EL INSTITUCION FINANCIERA MAGDALENA BUNAY PIZHA
L010
217592
FECHA: 2007/06/18 HORA: 08000300 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: NOTON092255728SMSR EM 092255728 092255728 9111110010080246 0000NOTREAVISO DE REMESAS: SR(A) LUCIA VERA ULLOA. FAVOR ACERCARSE A COBRAR SU GIRO EN EL INSTITUCION FINANCIERA LUCIA VERA ULLOA
L010
217599
FECHA: 2007/06/18 HORA: 08000300 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: NOTON092256905SMSR EM092256905 092256905 9111110010080246 0000NOTREAVISO DE REMESAS: SR(A) ANA NARCISA VERA. FAVOR ACERCARSE A COBRAR SU GIRO EN EL INSTITUCION FINANCIERA ANA NARCISA VERA
L010
217600
FECHA: 2007/06/18 HORA: 08000400 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: NOTON092957487SMSR EM092957487 092957487 9111110010080246 0000NOTREAVISO DE REMESAS: SR(A) DOLORES MERCEDES LEON. FAVOR ACERCARSE A COBRAR SU GIRO EN EL INSTITUCION FINANCIERA DOLORES MERCEDES LEON
L010
217536
FECHA: 2007/06/18 HORA: 08000400 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: NOTON095309011SMSR EM095309011 095309011 9111110010080246 0000NOTREAVISO DE REMESAS: SR(A) DOLORES GUALLPA. FAVOR ACERCARSE A COBRAR SU GIRO EN EL INSTITUCION FINANCIERA DOLORES GUALLPA
L010
217632
FECHA: 2007/06/18 HORA: 08000400 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: NOTON098264583SMSREM098264583 098264583 9111110010080246 0000NOTREAVISO DE REMESAS: SR(A) FRANCISCO LLIGUICOTA GUALLPA. FAVOR ACERCARSE A COBRAR SU GIRO EN EL INSTITUCION FINANCIERA FRANCISCO LLIGUICOTA GUALLPA
L010
217634
FECHA: 2007/06/18 HORA: 08000400 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: NOTON095385495SMSR EM095385495 095385495 9111110010080246 0000NOTREAVISO DE REMESAS: SR(A) MARIA TERESA DUTAN PALAGUACHI. FAVOR ACERCARSE A COBRAR SU GIRO EN EL INSTITUCION FINANCIERA MARIA TERESA DUTAN PALAGUACHI
L010
217635
FECHA: 2007/06/18 HORA: 08000400 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: NOTON093423866SMSR EM093423866 093423866 9111110010080246 0000NOTREAVISO DE REMESAS: SR(A) MARIO LEONCIO BERMEO CABRERA. FAVOR ACERCARSE A COBRAR SU GIRO EN EL INSTITUCION FINANCIERA MARIO LEONCIO BERMEO CABRERA
L010
217512
FECHA: 2007/06/18 HORA: 08000400 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: NOTON094758465SMSR EM094758465 094758465 9111110010080246 0000NOTREAVISO DE REMESAS: SR(A) SHIRLEY MARIUXI GARAY MONTALVO. FAVOR ACERCARSE A COBRAR SU GIRO EN EL INSTITUCION FINANCIERA SHIRLEY MARIUXI GARAY MONTALVO
L010
217522
FECHA: 2007/06/18 HORA: 08000500 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: NOTON099851324SMSR EM099851324 099851324 9111110010080246 0000NOTREAVISO DE REMESAS: SR(A) JOSE MARIA FAJARDO. FAVOR ACERCARSE A COBRAR SU GIRO EN EL INSTITUCION FINANCIERA JOSE MARIA FAJARDO
L010
217510
FECHA: 2007/06/18 HORA: 08000500 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: NOTON091639163SMSR EM091639163 091639163 9111110010080246 0000NOTREAVISO DE REMESAS: SR(A) ANGEL STALYN LLANOS REINOSO. FAVOR ACERCARSE A COBRAR SU GIRO EN EL INSTITUCION FINANCIERA ANGEL STALYN LLANOS REINOSO
L014
101510
FECHA: 2007/06/18 HORA: 08000500 SERVICIOSELECTRONICOS.INTERFAZ PROCTRN DATAINPUT: NOTON088694754SMSR EM088694754 088694754 9111110010080246 0000NOTREAVISO DE REMESAS: SR(A) MARIA MANUELA AUCANSHALA ZUNA. FAVOR ACERCARSE A COBRAR SU GIRO EN EL INSTITUCION FINANCIERA MARIA MANUELA AUCANSHALA ZUNA
L014
101521