

INDICE DE CONTENIDOS

AGRADECIMIENTO	I
DEDICATORIAS	II
RESUMEN	V
ABSTRACT	VI
CAPITULO I	
LA INFORMACION A REPLICAR	
Introducción	1
1.1 Definición de la información que se requiere consolidar	2
1.2 Verificación e igualdad de estructuras de las bases de datos	6
1.3 Explicación del esquema de replicación	12
1.3.1 El uso de las tablas externas y la generación de código Sql dinámico	12
1.3.2 El esquema de replicación en Marcimex S.A.	13
CAPITULO II	
IMPLEMENTACION DE REPLICACION EN AGENCIAS	
Introducción	17
2.1 Creación del ambiente de replicación a nivel de agencia	17
2.2 Proceso de envío de replicación	21
CAPITULO III	
IMPLEMENTACION DE REPLICACION EN LA MATRIZ	
Introducción	24
3.1 Creación y definición de parámetros	24
3.2 Creación del módulo de replicación en la matriz	28
3.3 Validación de información y control de errores	30
CAPITULO IV	
DESARROLLO DE MANUALES DE CONSULTA	
Introducción	32
4.1 Desarrollo de manual de consulta de usuario en la agencia	32
4.2 Desarrollo de manual de consulta de usuario de matriz	37
CAPITULO V	
DOCUMENTACION DE LA APLICACION	
Introducción	42
5.1 Tablas	42

5.1.1 SIS_TABLAS_REPLICADAS	42
5.1.1.1 Script de creación	42
5.1.2 SIS_ABREV_OBJETOS	43
5.1.2.1 Script de creación	43
5.1.3 SIS_REPLICACION	44
5.1.3.1 Script de creación	45
5.1.4 SIS_ERRORES_CARGA	46
5.1.4.1 Script de creación	46
5.1.5 SIS_PLANOS_SUBIDOS	47
5.1.5.1 Script de creación	47
5.2 Tablas Externas	48
5.2.1 DATOS_MATRIZ	48
5.2.2 DINAMICAS	49
5.3 Procedimientos	50
5.3.1 SIS_CREA_TRIGGER	50
5.3.2 SIS_BORRA_TRIGGER	51
5.3.3 SIS_CREA_TABLA_EXT	52
5.3.4 SIS_BORRA_TABLA_EXT	53
5.3.5 SIS_CREA_PROCS_CARGA	54
5.3.6 SIS_BORRA_PROCS_CARGA	55
5.3.7 SIS_GENERA_PLANO_TABLAS	56
5.4 Procedimientos creados dinámicamente	57
5.4.1 DINAMICOS	57
5.5 Funciones	58
5.5.1 SIS_EXTRAE_ABREV	58
5.6 Triggers	59
5.6.1 DINAMICOS	59
5.7 Formas	60
5.7.1 SIS001	60
5.7.2 SIS002	62
5.7.3 SIS003	64
5.8 Scripts	66
5.8.1 GENERA_ESQUEMA.SQL	66
5.8.2 COMPARAR_ESTRUCTURAS.SQL	67
CAPITULO VI	
CONCLUSIONES Y RECOMENDACIONES	
6.1 Conclusiones	68
6.2 Recomendaciones	69

BIBLIOGRFIA	71
ANEXOS	72

AGRADECIMIENTO

Queremos agradecer el presente trabajo monográfico a nuestro director de tesis el Ing. Fernando Balarezo, por dirigirnos correctamente y darnos las guías y apoyo constante para la culminación de esta monografía. Además al personal de la empresa Marcimex S.A. que nos dio la colaboración necesaria.

DEDICATORIA

“Dedico este trabajo a toda mi familia, que con su apoyo incondicional me dieron ánimos en todo momento para continuar y culminar esta tesis y carrera.”

Julio Tamayo S.

DEDICATORIA

“Dedico este trabajo a mi familia y muy especialmente a mi hija María Caridad que ha sido la fuente de inspiración que me da ánimos de seguir adelante en cada actividad que realizo diariamente”

Juan Carlos Castro P.

LAS IDEAS EXPUESTAS EN ESTE TRABAJO Y QUE APARECEN COMO PROPIAS DE LOS
AUTORES, SON DE NUESTRAS RESPONSABILIDAD.

Juan Carlos Castro P.
0103200093

Julio Tamayo S.
0102148822

RESUMEN

La Empresa MARCIMEX S.A. requiere el desarrollo de una aplicación capaz de centralizar en su oficina principal residente en la ciudad de Cuenca cierta información clave para procesos de dirección y control que se encuentra dispersa en cada una de las sucursales de la empresa. Esta aplicación consiste en un sistema de replicación asíncrona basado en el uso de SQL DINAMICO y TABLAS EXTERNAS. Para el desarrollo de la aplicación, se tomaron las necesidades de información de los departamentos de la empresa. Luego la aplicación inicia con una etapa de homologación de estructuras de datos en las agencias con respecto a las existentes en la matriz, posteriormente se desarrolla el sistema remoto que llamaremos a aquel que se instala en cada agencia que consta de cuatro etapas: escoger que tablas se replicarían en la matriz, proceso de captura de cambios durante el día, generación de archivos planos o cápsulas de réplica y envío de dicha información a la matriz. La etapa final del sistema es para la oficina principal o sitio master de réplica que es en donde se consolidará la información proveniente de las agencias, en esta etapa tenemos los siguiente procesos: Escoger las tablas a consolidar, generación de procesos dinámicos de carga de información y tablas externas, proceso de consolidación de información replicada y por último el control y corrección de errores durante el proceso. De esta manera queda implementado el sistema de replicación de la empresa MARCIMEX S.A.

ABSTRACT

The company MARCIMEX S.A. requires the development of an application able to centralize in its resident main office in Cuenca city the key information for direction and control processes, which is dispersed in each one of the branches of the company. This application consists of a system of asynchronous replication based on the use of DYNAMIC SQL and EXTERNAL TABLES. For the development of the application, in first place the necessities of information in each departments of the company were taken. Soon the application initiates with a stage of comparison of structures of data in agencies, with respect of existing ones in the main office, soon is developed the remote system that we will call to that it would settle in each agency that consists of four stages: choose that tables would be talked back in the main office, process of capture changes during the day, generation of flat files or capsules of replication and shipment of this information to the main office, the final stage of the system is for the main office or master site of replication that it is in where the original information from the agencies will be consolidated, in this stage we have following the process: choose the tables to consolidate, generation of dynamic processes of information load and external tables, process of consolidation of replicated information and finally the control and correction of errors during the process of consolidation. This way it would be designed and constructed the system of replication of the company MARCIMEX S.A.

CAPITULO 1

INTRODUCCION

Marcimex S.A., es una empresa de comercialización de electrodomésticos al por mayor y al por menor con cobertura a nivel nacional, formada por una oficina matriz en la ciudad de Cuenca con varias agencias o puntos de venta y bodegas para distribución de productos, localizadas en todo el país como son Quito, Guayaquil, Ambato, Riobamba, Santo Domingo de los Colorados, Quinde, Quevedo, La Maná, Portoviejo, Manta, Machala, Pasaje y Loja.

El soporte de esta infraestructura se lo realiza con un sistema de información cuya base de datos en la oficina matriz, consolida la información a nivel nacional. Esta base de datos sirve para soportar tanto las transacciones de las aplicaciones en la matriz como son Contabilidad, Proveedores, Compras, Nómina, Negociaciones de Cartera, Importaciones y además soporta la información generada por los sistemas de Cartera, Inventarios y Ventas que son alimentados por la información ingresada a través de aplicaciones distribuidas en las agencias y bodegas.

Las agencias de Marcimex S.A. se dividen en agencias de usuarios que son los puntos de facturación directa al público o usuario final, y en agencias de mayoreo que son puntos de facturación para clientes mayoristas.

Las aplicaciones manejan las transacciones completas de una agencia como punto de venta, es decir, manejo de clientes, crédito, facturación, Inventarios, Cartera y contabilización de agencia; por lo que cada agencia maneja su base de datos con sus programas y sus transacciones, las agencias de mayoreo tienen además características propias de manejo como por ejemplo la consignación de productos, razón por la cuál tienen programas y bases de datos diferentes entre los aplicaciones de usuarios y comerciantes. En el caso de las bodegas, las aplicaciones distribuidas manejan todo el movimiento del inventario, es decir sus ingresos y salidas de mercadería a través de sus programas y en su base de datos local.

Al no tener un sistema de información con conexiones en línea, donde la información generada en una agencia se refleje inmediatamente en la matriz, actualmente Marcimex S.A. a través de su departamento de Sistemas, maneja el envío de información de Cartera, Inventarios, Contabilidad, Ventas, a través de ciertos programas que fijan cierta cantidad de información mínima necesaria para su control y consolidación en la matriz. Con este esquema cada vez que generamos una nueva fuente de información o cuando necesitamos más información de la que actualmente se recibe en la matriz, es necesario generar más código de programas para traer dicha información. Además es necesario mantener y actualizar los programas que cargan dicha información de las agencias en la matriz, y hacer los controles operativos necesarios para asegurar que toda la información fue consolidada completamente.

La presente tesis, pretende simplificar el manejo de este problema a través de la replicación de información de las agencias generando archivos planos mediante un módulo que guarda automáticamente todas las acciones de las tablas de información para luego cargarlas en el servidor matriz mediante el uso de tablas externas de la base de datos principal. La generación de los procedimientos almacenados y triggers es automática, a través de código dinámico que crea, modifica y elimina estos objetos de la base de datos según las necesidades de replicar información.

Finalmente se pretende tener un sistema que automáticamente replica la información de sus agencias sin la modificación de código y sin el uso de código adicional cuando varíen las condiciones de tablas de información que se necesitan replicar. Por su lado en la recepción de la información, se maneja una carga de datos replicados completa, de igual manera sin necesidad de código adicional que desarrollar, podemos ampliar o limitar las tablas de información que necesitamos cargar. En este punto, "la carga de información", realizará los controles que garanticen la integridad de los datos, de manera de obtener información centralizada confiable.

LA INFORMACION A REPLICAR

1.1 Definición de la información que se requiere consolidar.

Marcimex S.A., maneja un sistema de información cuya consolidación se realiza en una base de datos única y centralizada. Las características y estructura de esta base de datos es implementada en cada una de sus agencias, para soportar sus transacciones, según los procesos de información que se manejan en cada una de ellas.

Al no tener un sistema de información con conexiones en línea, donde la información generada en una agencia se refleje inmediatamente en la matriz, si no es a través de ciertos programas que fijan cierta cantidad de información mínima necesaria para el control, es totalmente indispensable y necesario fijar nuevas necesidades de información de cada uno de los departamentos de la empresa.

Dichas necesidades deben estar apartadas de las limitaciones actuales del sistema, por lo que procederemos a realizar una encuesta a los departamentos de la empresa sobre sus requerimientos de información desde las agencias. Para aquello hemos preparado la siguiente encuesta:

La encuesta fue realizada entre los diferentes departamentos de la empresa como Ventas, Cartera, Contabilidad, Contraloría, Inventarios, Mercadeo y Tesorería; éstas encuestas son presentadas como anexos.

Como resultado de la necesidad de información de cada departamento tenemos el siguiente cuadro de tablas de información que debemos traer desde las agencias y actualizar todos los días.

Cuadro 2

Departamento	Necesidad	Tabla
VENTAS	Obligatorios	IN_FACTURAS
		IN_DETALLE_FACTURAS
		IN_MOVIMIENTOS_INVENTARIO
		CA_VENTAS_TARJETA_CREDITO
		CA_CLIENTES
		IN_CONSIGNA_X_CLIENTE
		IN_CONSIGNA_CLIENTE_DETALLE
		IN_ARTICULOS_X_ESTADO
	Deseables	CA_CREDITOS
		CA_VENCIMIENTOS
		CA_SOLICITUDES
		CA_DETALLE_SOLICITUDES
CARTERA	Obligatorios	CA_CLIENTES
		CA_CREDITOS
		CA_VENCIMIENTOS
		CA_MOVIMIENTOS
		CA_MOVIMIENOS_X_VENCIM
		CA_SOLICITUDES
		CA_DETALLE_SOLICITUDES
		CJ_NOTAS_DEBITO_UB
		CA_REFERENCIAS_PERSONALES
		CA_REFERENCIAS_COMERCIALES
		CA_REFERENCIAS_BANCARIAS
		CA_REFERENCIAS_PATRIMONIOS
		CA_REFERENCIAS_TARJ_CREDITO
MERCADEO	Obligatorios	IN_FACTURAS
		IN_DETALLE_FACTURAS

		CA_VENTAS_TARJETA_CREDITO
		IN_MOVIMIENTOS_INVENTARIO
TESORERIA	Obligatorios	CJ_MOVIMIENTOS
		CJ_DOCUMENTOS
		CJ_DOCUMENTOS_DETALLE
		CJ_NOTAS_DEBITO_UB
CONTRALORIA	Obligatorios	IN_FACTURAS
		IN_DETALLE_FACTURAS
		CA_CLIENTES
		CA_CREDITOS
		CA_VENCIMIENTOS
		CA_MOVIMIENTOS
		CA_MOVIMIENTOS_X_VENCIM
		CJ_MOVIMIENTOS
		CJ_DOCUMENTOS
		CJ_DOCUMENTOS_DETALLE
		CJ_ANTICIPOS_CLIENTE
		CJ_NOTAS_DEBITO_UB
CONTABILIDAD	Obligatorios	CJ_DOCUMENTOS
		CJ_DOCUMENTOS_DETALLE
		CJ_MOVIMIENTOS
		CJ_DATOS_CABECERA_PLANTILLA
		CJ_DATOS_DETALLE_PLANTILLA
		CJ_PLANTILLA_CONTABLE
		CJ_PLANTILLA_CUENTAS
		CJ_TRX_SIN_PLANTILLA
		CJ_TRX_SIN_PLANTILLA_DETALLE
LOGISTICA / INVENTARIOS	Obligatorios	IN_ARTICULOS_X_ESTADO
		IN_MOVIMIENTOS_INVENTARIO
		IN_CONSIGNA_X_CLIENTE
		IN_CONSIGNA_CLIENTE_DETALLE
		IN_ENTRADAS_SALIDAS
		IN_ENT_SAL_DETALLE
	Deseables	IN_FACTURAS
		IN_DETALLE_FACTURAS

De esta información procedemos a unificar las tablas necesarias, eliminando las que se repiten entre los diferentes departamentos. A continuación tenemos el resumen de las mismas.

Resumen de Tablas a Replicar a Partir de la Encuesta al Personal

NOMBRES DE TABLAS	NOMBRES DE TABLAS
IN_DETALLE_FACTURAS	IN_ARTICULOS_X_ESTADO
IN_ENTRADAS_SALIDAS	IN_FACTURAS
IN_CONSIGNA_CLIENTE_DETALLE	IN_CONSIGNA_X_CLIENTE
IN_MOVIMIENTOS_INVENTARIO	IN_ENT_SAL_DETALLE
CA_CLIENTES	CA_VENTAS_TARJETA_CREDITO
CA_SOLICITUDES	CA_VENCIMIENTOS
CA_MOVIMIENTOS_X_VENCIM	CA_MOVIMIENTOS
CA_REFERENCIAS_COMERCIALES	CA_REFERENCIAS_PERSONALES
CA_REFERENCIAS_TARJ_CREDITO	CA_REFERENCIAS_PATRIMONIOS
CA_CREDITOS	CA_DETALLE_SOLICITUDES
CA_REFERENCIAS_BANCARIAS	CA_VENTAS_TARJETA_CREDITO
CJ_ANTICIPOS_CLIENTE	CJ_MOVIMIENTOS
CJ_DOCUMENTOS	CJ_DOCUMENTOS_DETALLE
CJ_DATOS_CABECERA_PLANTILLA	CJ_PLANTILLA_CONTABLE
CJ_TRX_SIN_PLANTILLA_DETALLE	CJ_PLANTILLA_CUENTAS
CJ_TRX_SIN_PLANTILLA	CJ_DATOS_DETALLE_PLANTILLA
CJ_NOTAS_DEBITO_UB	

1.2 Verificación e igualdad de estructuras de las bases de datos.

Una vez definidas las tablas que son requeridas replicar, es necesario para el presente trabajo, realizar una verificación que garantice la igualdad de estructura entre las bases de datos remotas y la base de la agencia matriz, tanto en orden de campos como en tamaños de los mismos, ya que nuestra tesis se basa en la igualdad de estructuras de las tablas dentro de la base de datos tanto en nombre de campo, orden, tipo de dato y tamaño.

Para realizar la verificación de estructuras, vamos a determinar como estructura base la que tenemos en el servidor principal de la empresa y luego la comparamos con las estructuras de las agencias, para lo cual se dieron los siguientes pasos:

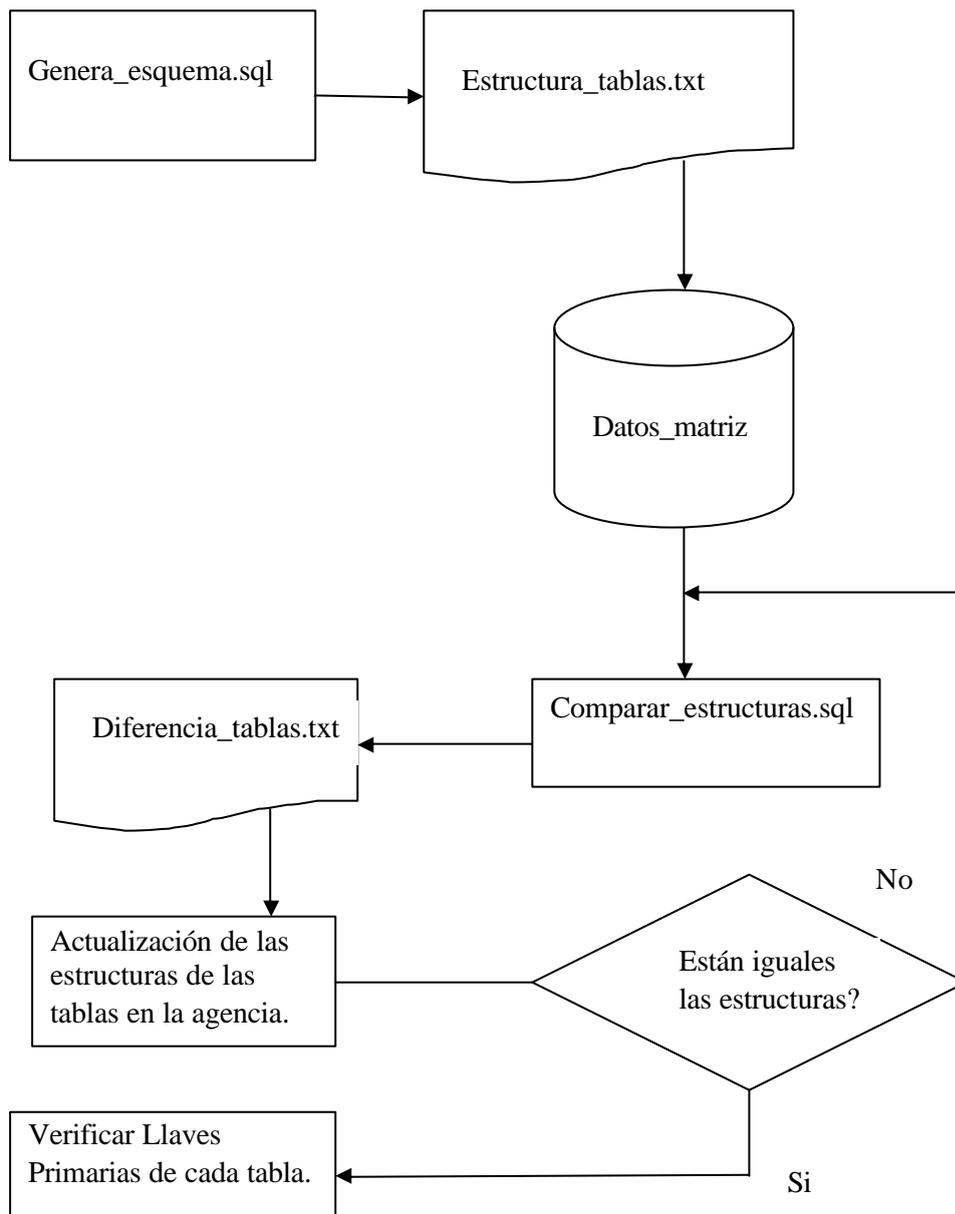
- a) A través del programa **genera_esquema.sql** vamos a determinar un archivo de salida (c:\estructura_tablas.txt) con la estructura de las tablas a replicar de la matriz.
- b) Una vez en la agencia creamos el directorio '**c:\comparadb**' en el sistema operativo donde guardamos el archivo **estructura_tablas.txt**, y a su vez creamos el directorio

lógico en la base de datos direccionado al directorio físico. Se crea el directorio en la base de datos con la siguiente sentencia:

```
create directory comparadb as 'c:\comparadb';
```

- c) Posteriormente creamos la tabla externa **datos_matriz** en la base de datos para que tome al archivo estructura_tablas.txt
- d) Luego en cada agencia procedemos a correr un programa **comparar_estructuras.sql** que nos ayudará a determinar la diferencia entre las estructuras de las tablas de la agencia versus la estructura de las tablas de la matriz mediante la comparación entre la tabla externa que contiene las estructuras de las tablas de la matriz contra el diccionario de datos de la agencia. El resultado se guarda en un archivo de salida llamado diferencia_tablas.txt
- e) Determinadas las diferencias procedemos a realizar la actualización de las estructuras de las tablas; en este punto es necesario poner toda la atención a fin de precautelar la información y evitar pérdidas. Procedemos a repetir este proceso hasta no tener diferencias de estructuras entre la agencia y la matriz.
- f) Por último procedemos a verificar la igualdad de las llaves primarias de cada tabla, tanto en orden, tipo de dato y tamaño de campo.

Diagrama del Proceso de Verificación e Igualdad de Estructuras en las Agencias.



Como ejemplo para esta tesis vamos a proceder a realizar esta verificación con la agencia Mariano Cueva.

- a) Generamos a través del programa **genera_esquema.sql** de la matriz el archivo `estructura_tablas.txt`
- b) Luego de crear el entorno en la agencia (directorios y tabla externa) se procede a correr el script `comparar_estructuras.sql` obteniendo la siguiente información de diferencias:

Resultados de la Verificación: **Tablas de Matriz con diferencias en Agencia**

#	NOMBRE_TABLA	NOMBRE_COLUMNA	TIPO_DATO	LONG.	PRECISION	ESCALA	ID_COL.
1	CA_CREDITOS	SECUENCIA_NEGOCIACION	NUMBER	22	4	0	38
2	CA_CREDITOS	TASA_NEGOCIACION	NUMBER	22	7	4	39
3	CA_DETALLE_SOLICITUDES	CANTIDAD_APROBADA	NUMBER	22	7	2	28
4	CA_DETALLE_SOLICITUDES	CANTIDAD_REAL	NUMBER	22	7	2	27
5	CA_DETALLE_SOLICITUDES	CODIGO_NO_APROBACION	NUMBER	22	3	0	29
6	CA_DETALLE_SOLICITUDES	CODIGO_NO_DESPACHO	NUMBER	22	3	0	30
7	CA_REFERENCIAS_PERSONALES	DIRECCION_REFERENCIA_PERSONAL	VARCHAR2	80			7
8	CA_SOLICITUDES	BODEGA_ORIGEN	NUMBER	22	4	0	37
9	CA_SOLICITUDES	COD_SUCURSAL_VENDED	NUMBER	22	4	0	39
10	CA_SOLICITUDES	CONSIGNA_FACTURA	VARCHAR2	1			38
11	CA_SOLICITUDES	DSCTO_EN_PUNTOS	NUMBER	22	7	0	32
12	CA_SOLICITUDES	DSCTO_PROMOCION	NUMBER	22	13	2	31
13	CA_SOLICITUDES	TOTAL_DESCUENTO	NUMBER	22	13	2	30
14	CA_SOLICITUDES	TOTAL_FLETES	NUMBER	22	13	2	36
15	CJ_TRX_SIN_PLANTILLA_DETALLE	ESTADO	VARCHAR2	2			11
16	IN_CONSIGNA_CLIENTE_DETALLE	ADICIONADO_POR	VARCHAR2	10			14
17	IN_CONSIGNA_CLIENTE_DETALLE	CANTIDAD	NUMBER	22	7	2	12
18	IN_CONSIGNA_CLIENTE_DETALLE	CODIGO_ARTICULO	NUMBER	22	3	0	7
19	IN_CONSIGNA_CLIENTE_DETALLE	CODIGO_CLIENTE	NUMBER	22	8	0	1
20	IN_CONSIGNA_CLIENTE_DETALLE	CODIGO_EMPRESA	NUMBER	22	4	0	2
21	IN_CONSIGNA_CLIENTE_DETALLE	CODIGO_ESTADO	NUMBER	22	2	0	8
22	IN_CONSIGNA_CLIENTE_DETALLE	CODIGO_LINEA	NUMBER	22	5	0	6
23	IN_CONSIGNA_CLIENTE_DETALLE	CODIGO_MARCA	NUMBER	22	3	0	5
24	IN_CONSIGNA_CLIENTE_DETALLE	CODIGO_SERVIDOR	NUMBER	22	4	0	3
25	IN_CONSIGNA_CLIENTE_DETALLE	CODIGO_SUCURSAL	NUMBER	22	3	0	4
26	IN_CONSIGNA_CLIENTE_DETALLE	CODIGO_TRANSACCION	NUMBER	22	5	0	11
27	IN_CONSIGNA_CLIENTE_DETALLE	ESTADO	VARCHAR2	1			18
28	IN_CONSIGNA_CLIENTE_DETALLE	FECHA	DATE	7			10
29	IN_CONSIGNA_CLIENTE_DETALLE	FECHA_ADICION	DATE	7			15
30	IN_CONSIGNA_CLIENTE_DETALLE	FECHA_MODIFICACION	DATE	7			17
31	IN_CONSIGNA_CLIENTE_DETALLE	MODIFICADO_POR	VARCHAR2	10			16
32	IN_CONSIGNA_CLIENTE_DETALLE	NUM_COMPROBANTE	NUMBER	22	8	0	9
33	IN_CONSIGNA_CLIENTE_DETALLE	SECUENCIA_FACTURA	NUMBER	22	10	0	13
34	IN_CONSIGNA_X_CLIENTE	ADICIONADO_POR	VARCHAR2	10			12
35	IN_CONSIGNA_X_CLIENTE	BUFFER	NUMBER	22	5	0	9
36	IN_CONSIGNA_X_CLIENTE	CODIGO_ARTICULO	NUMBER	22	3	0	7
37	IN_CONSIGNA_X_CLIENTE	CODIGO_CLIENTE	NUMBER	22	8	0	1
38	IN_CONSIGNA_X_CLIENTE	CODIGO_EMPRESA	NUMBER	22	4	0	2
39	IN_CONSIGNA_X_CLIENTE	CODIGO_ESTADO	NUMBER	22	2	0	8
40	IN_CONSIGNA_X_CLIENTE	CODIGO_LINEA	NUMBER	22	5	0	6
41	IN_CONSIGNA_X_CLIENTE	CODIGO_MARCA	NUMBER	22	3	0	5
42	IN_CONSIGNA_X_CLIENTE	CODIGO_SERVIDOR	NUMBER	22	4	0	3
43	IN_CONSIGNA_X_CLIENTE	CODIGO_SUCURSAL	NUMBER	22	3	0	4
44	IN_CONSIGNA_X_CLIENTE	COSTO_IDD	NUMBER	22	11	2	16
45	IN_CONSIGNA_X_CLIENTE	COSTO_TDD	NUMBER	22	11	2	17
46	IN_CONSIGNA_X_CLIENTE	DIAS_COSTO	NUMBER	22	7	2	18
47	IN_CONSIGNA_X_CLIENTE	FECHA_ADICION	DATE	7			13
48	IN_CONSIGNA_X_CLIENTE	FECHA_MODIFICACION	DATE	7			15
49	IN_CONSIGNA_X_CLIENTE	FECHA_MODIFICA_BUFFER	DATE	7			19
50	IN_CONSIGNA_X_CLIENTE	FECHA_ULT_CONSIGNA	DATE	7			11
51	IN_CONSIGNA_X_CLIENTE	MODIFICADO_POR	VARCHAR2	10			14
52	IN_CONSIGNA_X_CLIENTE	STOCK	NUMBER	22	7	2	10

Resultados de la Verificación: **Tablas de Mariano Cueva con diferencia.**

#	NOMBRE_TABLA	NOMBRE_COLUMNA	TIPO_DATO	LONG.	PRECISION	ESCALA	ID_COL.
1	CA_CREDITOS	NO EXISTE					
2	CA_CREDITOS	TASA_NEGOCIACION	NUMBER	22	7	4	38
3	CA_DETALLE_SOLICITUDES	NO EXISTE					
4	CA_DETALLE_SOLICITUDES	NO EXISTE					
5	CA_DETALLE_SOLICITUDES	NO EXISTE					
6	CA_DETALLE_SOLICITUDES	NO EXISTE					
7	CA_REFERENCIAS_PERSONALES	DIRECCION_REFERENCIA_PERSONAL	VARCHAR2	20			7
8	CA_SOLICITUDES	NO EXISTE					
9	CA_SOLICITUDES	NO EXISTE					
10	CA_SOLICITUDES	NO EXISTE					
11	CA_SOLICITUDES	DSCTO_EN_PUNTOS	NUMBER	22	7	0	30
12	CA_SOLICITUDES	DSCTO_PROMOCION	NUMBER	22	13	2	32
13	CA_SOLICITUDES	TOTAL_DESCUENTO	NUMBER	22	13	2	31
14	CA_SOLICITUDES	NO EXISTE					
15	CJ_TRX_SIN_PLANTILLA_DETALLE	ESTADO	VARCHAR2	1			11
16	NO EXISTE	NO EXISTE					
17	NO EXISTE	NO EXISTE					
18	NO EXISTE	NO EXISTE					
19	NO EXISTE	NO EXISTE					
20	NO EXISTE	NO EXISTE					
21	NO EXISTE	NO EXISTE					
22	NO EXISTE	NO EXISTE					
23	NO EXISTE	NO EXISTE					
24	NO EXISTE	NO EXISTE					
25	NO EXISTE	NO EXISTE					
26	NO EXISTE	NO EXISTE					
27	NO EXISTE	NO EXISTE					
28	NO EXISTE	NO EXISTE					
29	NO EXISTE	NO EXISTE					
30	NO EXISTE	NO EXISTE					
31	NO EXISTE	NO EXISTE					
32	NO EXISTE	NO EXISTE					
33	NO EXISTE	NO EXISTE					
34	NO EXISTE	NO EXISTE					
35	NO EXISTE	NO EXISTE					
36	NO EXISTE	NO EXISTE					
37	NO EXISTE	NO EXISTE					
38	NO EXISTE	NO EXISTE					
39	NO EXISTE	NO EXISTE					
40	NO EXISTE	NO EXISTE					
41	NO EXISTE	NO EXISTE					
42	NO EXISTE	NO EXISTE					
43	NO EXISTE	NO EXISTE					
44	NO EXISTE	NO EXISTE					
45	NO EXISTE	NO EXISTE					
46	NO EXISTE	NO EXISTE					
47	NO EXISTE	NO EXISTE					
48	NO EXISTE	NO EXISTE					
49	NO EXISTE	NO EXISTE					
50	NO EXISTE	NO EXISTE					
51	NO EXISTE	NO EXISTE					
52	NO EXISTE	NO EXISTE					

Los resultados de la verificación nos muestran las tablas de matriz que tienen diferencia con las tablas de la agencia, y también nos presenta los resultados de verificación con las tablas de la agencia que tienen diferencia con la estructura de la matriz.

En el ejemplo de los resultados entre matriz y la agencia Mariano Cueva, se advierte que tenemos diferencias en las tablas ca_creditos, ca_detelle_solicitudes, ca_referencias_personales, ca_solicitudes, cj_trx_sin_plantilla_contable, donde nos faltan algunos campos y en otros casos tenemos diferentes tamaños de campos.

También tenemos la lista de tablas que no existen en la base de datos de la agencia como son: in_consigna_cliente_detalle e in_consigna_x_cliente; en este ultimo caso dichas tablas no existen en la agencia Mariano Cueva, debido a que solo se utilizan para agencias de mayoreo.

Ejemplo de comparación de diferencias: Tablas de matriz

#	NOMBRE_TABLA	NOMBRE_COLUMNA	TIPO_DATO	LONG.	PRECISION	ESCALA	ID_COL.
1	CA_CREDITOS	SECUENCIA_NEGOCIACION	NUMBER	22	4	0	38
2	CA_CREDITOS	TASA_NEGOCIACION	NUMBER	22	7	4	39

Comparación de diferencias: Tablas de agencia Mariano Cueva

#	NOMBRE_TABLA	NOMBRE_COLUMNA	TIPO_DATO	LONG.	PRECISION	ESCALA	ID_COL.
1	CA_CREDITOS	NO EXISTE					
2	CA_CREDITOS	TASA_NEGOCIACION	NUMBER	22	7	4	38

Vemos claramente que la tabla CA_CREDITOS de la matriz tiene la columna SECUENCIA_NEGOCIACION; mientras que la misma tabla en la agencia Mariano Cueva no tiene dicha columna.

Definidas las diferencias se procede a igualar las estructuras de las tablas manualmente hasta no tenerlas.

Hay que tener cuidado para evitar pérdidas de información, por lo que se recomienda realizar:

- Detener la transaccionalidad de la agencia.
- Respaldar las tablas a modificar
- Igualar las estructuras respecto a las tablas de matriz
- Copiar la información respaldada hacia las tablas con su nueva estructura.
- Verificar los resultados
- Correr nuevamente el programa comparar_estructuras.sql y verificar la igualdad.
- Eliminar las tablas de respaldo que se crearon.

1.3 Explicación del esquema de replicación.

1.3.1 El uso de las tablas externas y la generación de código Sql dinámico.

El manejo de replicación de información que se presenta en esta tesis, es un método sencillo y alternativo para generar replicación que, como, se muestra en la empresa a implementarse, facilitará la consolidación de información. Es un software que se basará principalmente en dos conceptos tecnológicos *Tablas Externas* y *SQL Dinámico*.

Tablas Externas: Una tabla externa no es nada más que una tabla estándar de solo lectura que maneja la base de datos Oracle, pero cuya estructura interna (campos) y externa (tipo de datos) depende de un archivo plano que debe estar estructurado en su totalidad bajo un mismo patrón de campos, es decir, un campo separado de otro ya sea mediante un carácter o un número igual de campos en toda su estructura. Sobre dicha tabla podemos realizar consultas de tipo SELECT, entre las muchas ventajas que nos presenta esta modalidad de recuperación de información sin duda alguna la más importante es la facilidad y versatilidad de su uso, en vista de que se pueden crear utilizando SQL DINAMICO y filtrar la información antes de pasarla, por ejemplo, a una tabla interna de la base de datos de Oracle, siendo una herramienta sumamente poderosa para operaciones de migración de información. La tabla externa apunta a su archivo de texto mediante el uso de un objeto de la base de datos el *directorio*.

Directorio: Es un objeto de la base de datos, que facilita el uso de las tablas externas. Procedemos a crear primero en el sistema operativo el directorio, y luego usamos éste directorio para crear el objeto *directorio* de la base de datos.

A continuación un ejemplo de creación de un directorio:

Primero se crea el directorio en el sistema operativo **c:\comparadb**.

Luego creamos el directorio de la base de datos con la siguiente sentencia

```
create directory comparadb as 'c:\comparadb';
```

En la base de datos se crea el objeto DIRECTORIO **comparadb**. No es necesario que tengan el mismo nombre el directorio de la base como el del sistema operativo.

El manejo de tablas externas viene como una nueva ventaja a partir de la versión de Oracle 9i .

Principales ventajas:

- El uso de archivos planos como tablas de datos.
- Acceso a los archivos vía comandos PL/Sql
- Elimina el uso la herramienta SqlLoader. Herramienta que sirve para cargar datos desde archivos planos, pero que requiere instalación de la misma, formatos de la información, y manejos de control de carga de datos.

Sql Dinámico: El diseño original del lenguaje PL/SQL implicaba el uso del mecanismo de acoplamiento temprano debido a que en un principio dicho lenguaje solo podía contener instrucciones de tipo DML (lenguaje de manipulación de datos) y no DDL (lenguaje de definición de datos) a partir de la versión 2.1 de PL/SQL que se estreno conjuntamente con ORACLE 7 versión 7.1 aparece lo que se conoce con el nombre de SQL DINAMICO, que no es mas que código SQL que se genera y ejecuta en tiempo de ejecución, a diferencia del código estático que es acoplado y generado en tiempo de compilación, el SQL DINAMICO utiliza el método de acoplamiento tardío, acoplamiento no es mas que identificar el área de almacenamiento asociada a un identificador dentro del código de un programa. El primer paquete que proporciona ORACLE para operar con SQL DINAMICO es el paquete DBMS_SQL, dicho paquete fue perfeccionado para ORACLE 8, Oracle 8i mejora aun mas la situación, proporcionando SQL dinámico nativo integrado en el lenguaje. En la presente tesis hacemos uso de los dos métodos de manipulación, DBMS_SQL y SQL DINAMICO NATIVO, sin duda alguna el SQL DINAMICO NATIVO ofrece mas simplicidad al momento de armar el código pero su restricción mas importante es el tamaño muy reducido de código que puede contener, es decir, de lo desarrollado en esta tesis al aplicar SQL DINAMICO NATIVO nos limitaba en el tamaño de la generación del código; mientras que el paquete DBMS_SQL no tiene restricción en cuanto al tamaño de código dinámico que empleemos.

Ventajas:

- Puede variar el código dependiendo de las necesidades en tiempo de ejecución.
- El acceso a la base de datos es más rápido.

1.3.2 El esquema de replicación en Marcimex S.A..

Es importante señalar que como la replicación de la información comienza a partir de su implementación en cada agencia, es necesario tener un respaldo de toda la información antes de la implementación, este respaldo se debe subir al servidor central de la matriz como el primer paso para igualar la información.

Por otro lado, si bien es cierto que este esquema no depende de las versiones de las bases de datos, hemos visto que en ciertas versiones algunas tablas donde se guarda el diccionario de datos de la base de datos, difieren en nombres de campos, por lo que hay que tener cuidado de esto en la etapa de generación de procedimientos y triggers almacenados de la base de datos que sirven para las novedades o información a replicar desde los servidores de las agencias.

El esquema de replicación se divide en dos etapas:

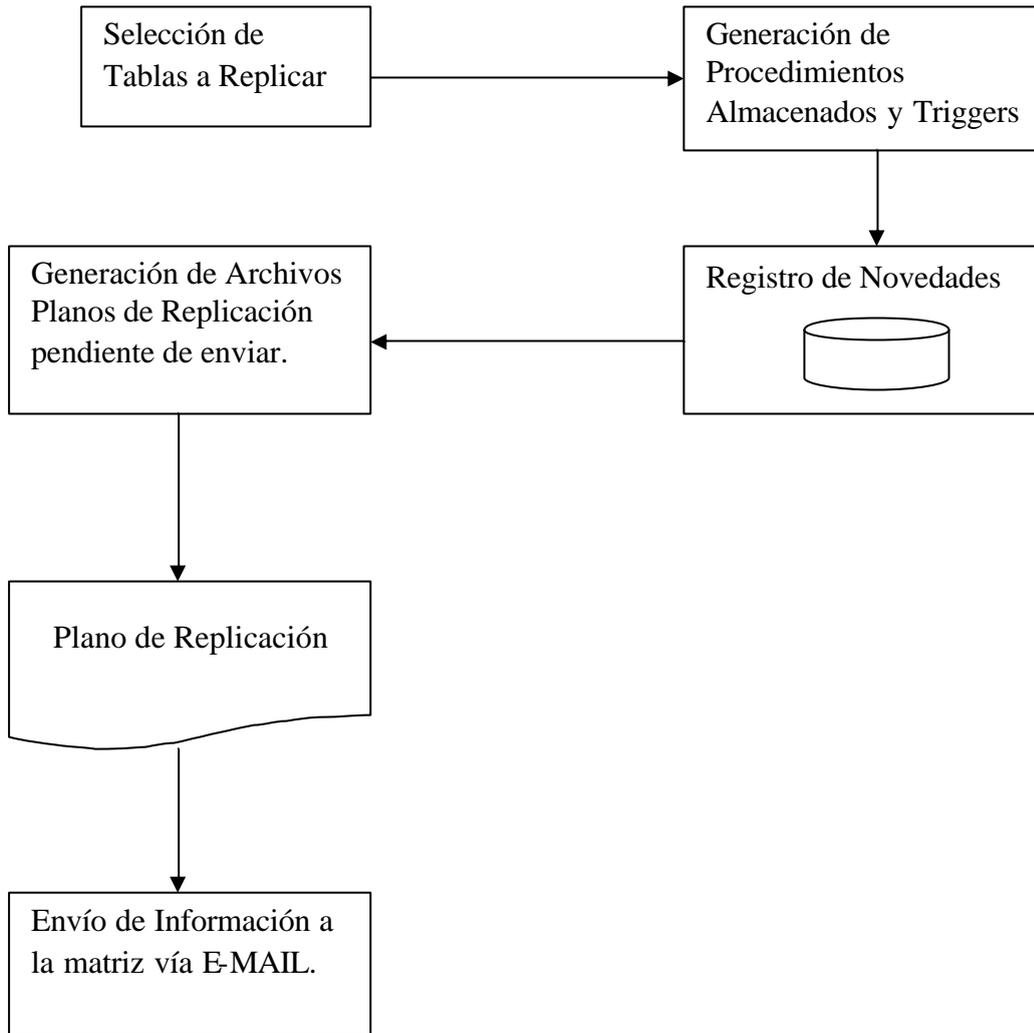
- **Primera etapa:** La activación de las novedades o datos que queremos replicar y que se han modificado en un período de tiempo en el servidor de una agencia.
- **Segunda etapa:** La consolidación y control de la información generada en las agencias dentro del servidor central.

1) Primera etapa:

Asumiendo que se efectuó el proceso de validación de estructuras de información de las tablas a ser replicadas en cada agencia, según los resultados de las necesidades de información a ser replicada desde las agencias hacia la matriz, para la activación de las novedades o datos a ser replicados es necesario considerar el siguiente desarrollo:

- a) Mediante una programa, mostramos todas las tablas que existen en la base de datos de cada agencia, en ésta pantalla identificamos las tablas que requieren replicación. Luego de esto la aplicación procede a generar dinámicamente los procedimientos y triggers almacenados en la base de datos para cumplir el objetivo de comenzar a generar las novedades (inserciones, actualizaciones y eliminaciones de registros) de dichas tablas de información.
- b) Una vez efectuado el paso anterior, se guardan todas las novedades en tablas de la aplicación de replicación que conservarán dicha información con un estado de pendiente de enviar en cada uno de los registros.
- c) En la noche, durante el cierre diario de las agencias de Marcimex, se procede a efectuar una tarea mediante el uso de otro programa, el cuál genera los archivos planos desde las tablas de replicación de datos y las guarda en un directorio del servidor para posteriormente enviarlas vía email hacia la matriz.
- d) Dichos archivos planos tienen una estructura según las tablas que se replican, y sus nombres van acompañados de un número secuencial según los envíos realizados y el código de la agencia donde se generan. Esto nos sirve para que en la consolidación de información no exista confusión con el manejo de estos archivos.
- e) Una vez completado el ciclo de generación de novedades y envío de las mismas, se procede a actualizar los registros como enviados para no tomarlos en cuenta en el próximo envío.

Diagrama del Proceso del Esquema de Replicación en la Agencia.

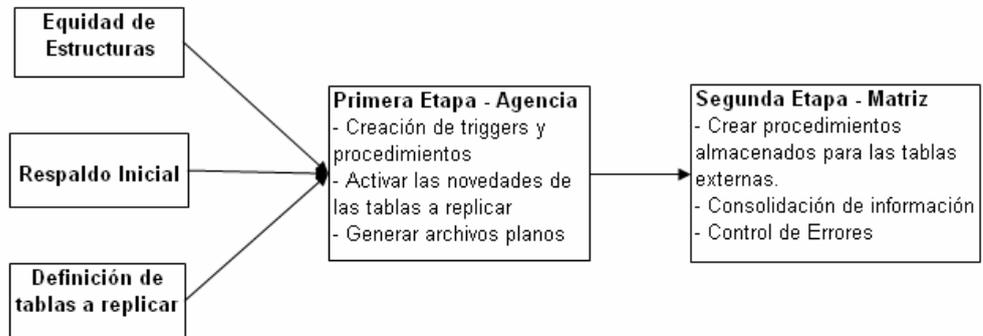


2) Segunda etapa:

- Previa la recepción de los archivos planos enviados por las agencias, mediante la ejecución de la aplicación de la replicación para la matriz, se generan los procedimientos y triggers almacenados para la base de datos, pero esta vez los hace para generar los procedimientos dinámicos que crean las tablas externas y procedimientos de lectura de los archivos planos en el servidor de matriz.
- Una vez generado el ambiente de procedimientos y tablas externas para la recepción de datos, se procede con la consolidación de la información, esto se lo hace mediante una pantalla que pide la agencia y la secuencia a subir. Esta secuencia es el número consecutivo de los envíos de información de cada agencia, por lo que en la mayoría de los casos será la secuencia de cada día de envío de información.

- c) Por último es necesario manejar y controlar posibles errores al momento de consolidar la información, dichos errores son guardadas en tablas para consultarlos y tomar acciones correctivas según sea el caso.

DIAGRAMA GENERAL DE REPLICACION



CAPITULO 2

IMPLEMENTACION DE REPLICACION EN AGENCIAS

Introducción.

En este capítulo se explicará como funciona el esquema de replicación presentado, respecto a su primera etapa que es la generación de las novedades de los registros de cada tabla a replicar. Además se explicará como manejar el envío de información desde las agencias y bodegas hacia la matriz para proceder a la segunda etapa de la replicación.

2.1 Creación del ambiente de replicación a nivel de agencia

La replicación de información en la agencia requiere de la implementación de una pantalla para indicar que tablas de información son replicadas y generar todo el código de procedimientos y triggers almacenados dentro de la base de datos para proceder con la implementación de las novedades de la tabla, dichas novedades son los eventos que le puede suceder a un registro, es decir, una inserción de nuevo registro, una actualización de registros o eliminación de registros.

Pasos a seguir:

1) Definir las tablas que vamos a replicar

Mediante el formulario SIS001 que veremos en la siguiente figura:

Selección de Tablas para Réplica

Usuario: SISTEMAS

Host
Cliente

Tablas del Sistema

- AP_APLICACIONES
- AP_BARRIOS
- AP_CANTONES
- AP_CARDINAL
- AP_CIUDES
- AP_EMITOR
- AP_EMPLEADOS
- AP_EMPRESAS
- AP_FECHAS_FESTIVAS
- AP_FORMAS_DE_PAGO
- AP_INSTITUCIONES
- AP_MONEDAS
- AP_NACIONES
- AP OPCIONES_DEL SISTEMA
- AP_PAISES

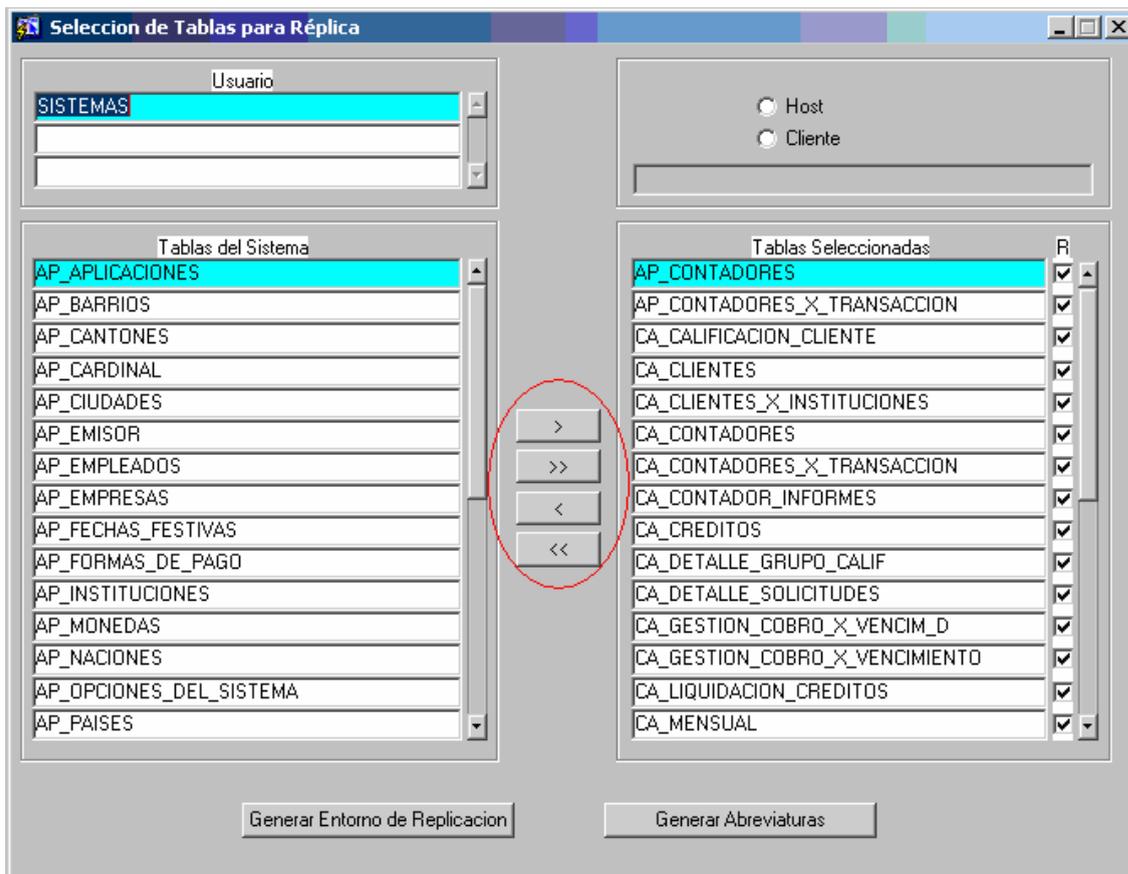
Tablas Seleccionadas

- AP_CONTADORES
- AP_CONTADORES_X_TRANSACCION
- CA_CALIFICACION_CLIENTE
- CA_CLIENTES
- CA_CLIENTES_X_INSTITUCIONES
- CA_CONTADORES
- CA_CONTADORES_X_TRANSACCION
- CA_CONTADOR_INFORMES
- CA_CREDITOS
- CA_DETALLE_GRUPO_CALIF
- CA_DETALLE_SOLICITUDES
- CA_GESTION_COBRO_X_VENCIM_D
- CA_GESTION_COBRO_X_VENCIMIENTO
- CA LIQUIDACION_CREDITOS
- CA_MENSUAL

> >> < <<

Generar Entorno de Replicacion Generar Abreviaturas

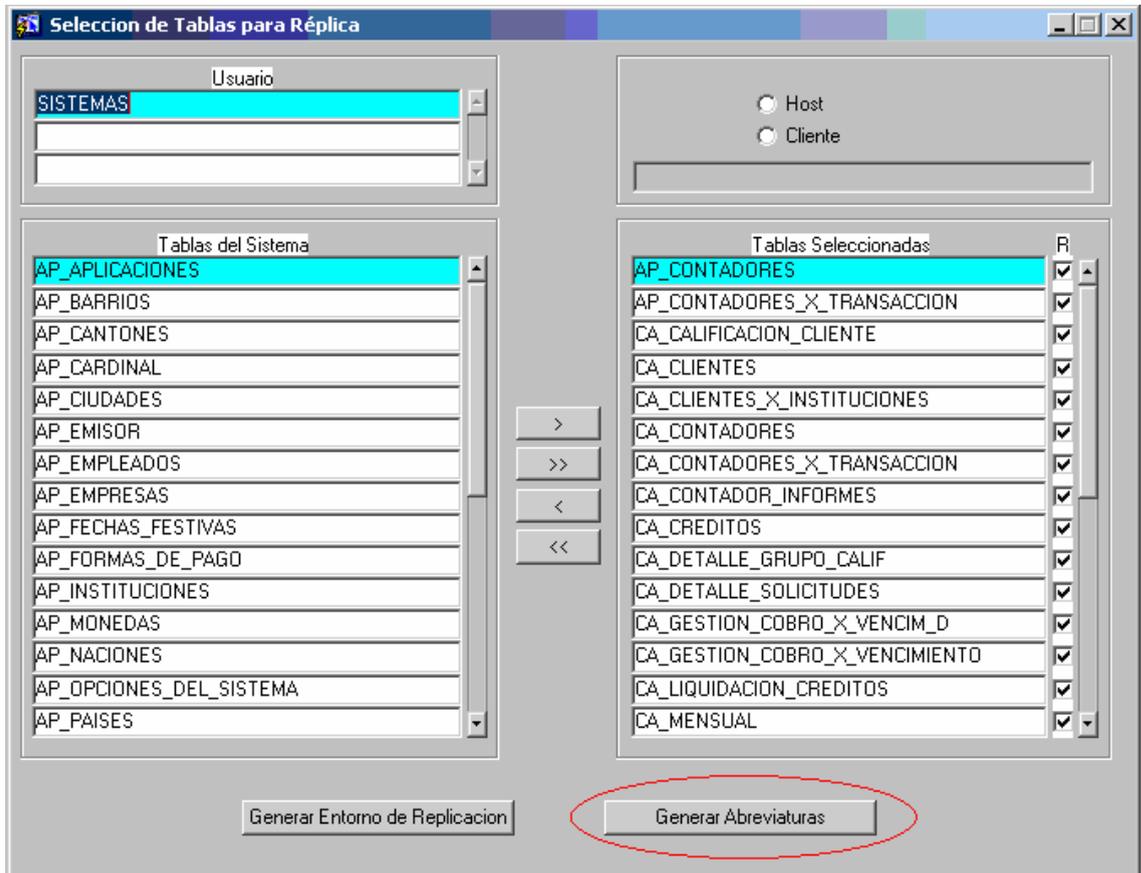
Seleccionaremos primeramente el usuario propietario de las tablas que vamos a replicar, al seleccionar el usuario en la parte inferior del mismo aparecerán todas las tablas de su propiedad y mediante los botones PASAR UNO o PASAR TODAS las seleccionaremos,



una vez seleccionadas se pasaran al panel derecho en donde estarán ubicadas valga la redundancia todas las tablas de dicho usuario que han sido seleccionadas para el proceso de replicación .

2) Creación de Abreviaturas

A veces las tablas de los diversos esquemas de base de datos en las que trabajamos, poseen nombres sumamente largos y debido a que en el proceso de replica por lo general los generadores de "código dinámico" utilizan dichos nombres concatenándolos con ciertos prefijos o sufijos según sea el caso para crear nuevos objetos (triggers, tablas externas y procedimientos de carga) dichos objetos poseerán nombres aun más largos que a veces sobrepasaran el largo máximo permitido por el motor de la base de datos. Debido a ello, vamos a crear abreviaturas para cada tabla replicada es decir vamos a convertir internamente el nombre de la tabla real en un nombre virtual que lo utilizaremos internamente en el proceso de replicación para referirnos a dicha tabla. Esto lo realizaremos haciendo click en el botón "Crear Abreviaturas" ubicado en la parte inferior izquierda del formulario SIS001



3) Definir el esquema de replicación

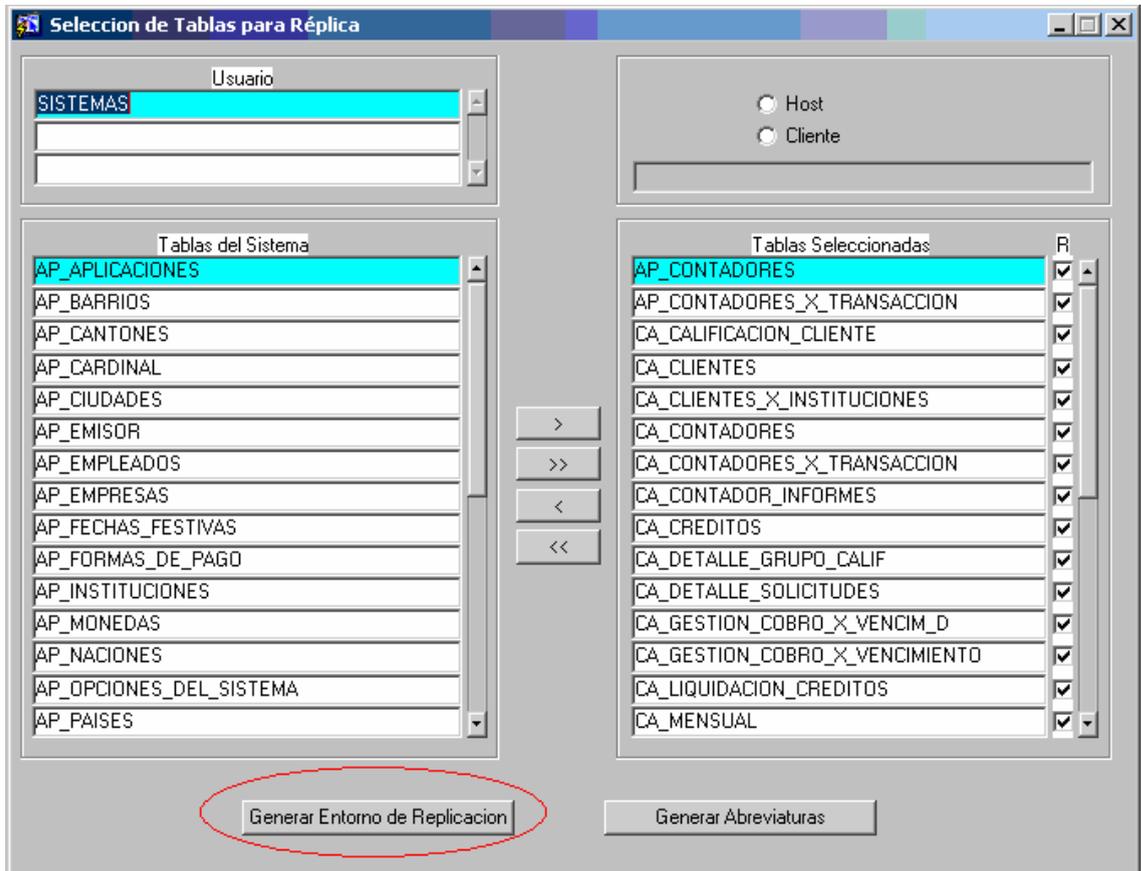
Dentro del proceso de replica de información tenemos dos esquemas claramente establecidos:

- Host
- Cliente

Para el ambiente de replicación de las agencias siempre escogeremos el modo Cliente



Posteriormente seleccionaremos el botón “Generar entorno”



Generar Entorno en modo Cliente realizará lo siguiente:

- Graba los cambios realizados en la pantalla, es decir guarda las tablas seleccionadas dentro de las tablas de control del modulo de replicación.
- Creará los triggers de replicación uno por cada tabla que se replique, dichos triggers son el centro mismo del proceso de replica en el lado del Cliente, son los encargados de monitorear los cambios que se realicen sobre las tablas seleccionadas para la replica y guardar la información necesaria sobre dichos cambios ("rowid "de la tabla, tipo de acción que se ejecuto, datos de control, etc.) en la(s) tablas de control cuya información posteriormente será ocupada para la creación de los archivos planos resultantes que se enviaran al HOST para su respectiva carga en la tabla destino final de la replica.

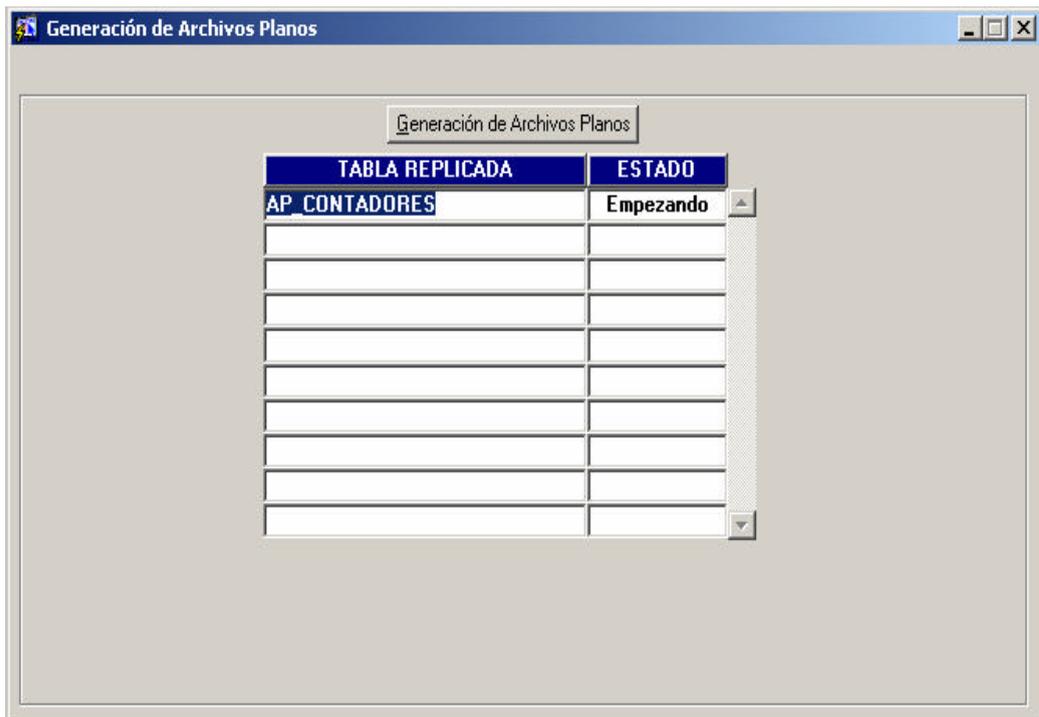
2.2 Proceso de envío de replicación.

Luego de que los triggers de replicación sean creados, estos continuamente monitorearán los cambios que se realicen sobre la información de las tablas seleccionadas para la replica y guarda dicha información como base para posteriormente generar los archivos planos con la información a replicar.

1) Ingreso a la pantalla de generación de archivos planos (SIS002)

Esta pantalla iniciará el proceso de generación de archivos planos al pulsar el botón

y nos da la siguiente información:



En ese momento comienza a tomar toda la información de las tablas a replicar y que no ha sido replicada aún, esto se controla de acuerdo a un estado ("P") de los registros a replicar.

Luego cuando termina presenta la siguiente pantalla:



TABLA REPLICADA	ESTADO
CJ_PLANTILLA_CONTABLE	Generado
CJ_PLANTILLA_CUENTAS	Generado
CJ_SECUENCIA_DOCUMENTO_SU	Generado
IN_ARTICULOS_X_ESTADO	Generado
IN_DETALLE_FACTURAS	Generado
IN_ENTRADAS_SALIDAS	Generado
IN_ENT_SAL_DETALLE	Generado
IN_FACTURAS	Generado
IN_MOVIMIENTOS_INVENTARIO	Generado
PROCESO TERMINADO	

En ese momento quedan generados los archivos planos y marcados los registros como ya enviados (estado="E").

2) Envío de Archivos planos de Replica

A diferencia de un sistema de replicación síncrona en donde el canal de comunicaciones entre los sitios remotos y el sitio receptor de replica es por lo general una línea de telecomunicaciones directa, ya sea esta alámbrica o inalámbrica que nos permite asegurarnos la confiabilidad de los datos que por ella transitan; en la replica asíncrona que es el centro de nuestro estudio el canal de comunicaciones y mas aun el proceso de envío y recepción de información dependerán en gran parte de la intervención del usuario, en tareas como:

- El proceso de empaquetamiento de archivos planos generados en el punto de replicación remoto.
- El proceso de copia en medio magnético de dichos archivos para su posterior envío, o bien el envío por correo electrónico de los mismos al sitio master.
- El proceso de recepción del medio (correo, disco, etc.).
- El desempaquetamiento de la información comprimida desde el medio.
- El correcto posicionamiento de los archivos de replicación en sus correctas ubicaciones para la posterior carga de la información en el master, etc.

Por esta razón el usuario debe tener en mente que de el orden y correcto posicionamiento de la información en los directorios precisos dependerá también la

correcta carga de información replicada para lo cual hemos fijado dos directorios fundamentales que servirán como depósitos de información para la generación de archivos planos en el sitio remoto y la carga de dicha información una vez que estos se encuentren en el servidor de matriz. En la empresa MARCIMEX hemos fijado el directorio en el cual se generaran los archivos planos desde el modulo SIS002.FMX en: Z:\APPL\REPLICA\PLANOS\ y los nombres de archivos de replica que se generaran en dicho directorio tendrán la siguiente estructura: NOMBREDETABLA_SECUENCIAENVIO.CODIGOSUCURSAL.

Es decir para la tabla CA_CREDITOS con secuencia de envío 150 de la sucursal 41, el nombre del archivo sería CA_CREDITOS_150.41.

Luego de generar los archivos de replicación del día en el directorio antes mencionado, dicha información deberá ser empaquetada y posteriormente enviada por correo electrónico a la matriz (Cuenca) para su posterior carga.

Bien podríamos haber escogido otro medio de envío de información a la matriz como discos o medios ópticos de almacenamiento pero en las actuales circunstancias escogimos el E-MAIL por tres razones fundamentales:

- Todas las agencias de la empresa MARCIMEX S.A. poseen cuenta de correo electrónico.
- Los archivos de cierres diarios (Cartera, Contabilidad e Inventarios) todos los días son enviados por este medio a la agencia, ahorraríamos tiempo enviando también la replica.
- El usuario conoce bien el manejo de correo electrónico y no necesitará capacitación adicional evitando posteriores errores hasta la adquisición de experiencia por parte del usuario.

CAPITULO 3

IMPLEMENTACION DE REPLICACION EN LA MATRIZ.

Introducción.

En este capítulo explicaremos el proceso de consolidación de la información en la matriz cumpliendo así la siguiente etapa de la replicación. Trataremos aspectos como: La recepción de los archivos planos que vienen de las agencias o bodegas, definición y creación del entorno de la replicación en la matriz para consolidar la información, funcionamiento del modulo de carga de datos y por último el manejo y control de errores.

3.1 Creación y definición de parámetros

Los archivos de mail de las agencias con información de su replicación van a llegar todos los días y de todas las agencias y bodegas, por lo que es necesario organizar este trabajo; como definición creamos un solo directorio llamado `y:\mails_replica` donde guardaremos todos los mails que llegan. En este directorio reposarán estos archivos mientras no se procedan con su consolidación, posteriormente podríamos guardarlos en un directorio de respaldo de manera que quede organizado el manejo de estos archivos.

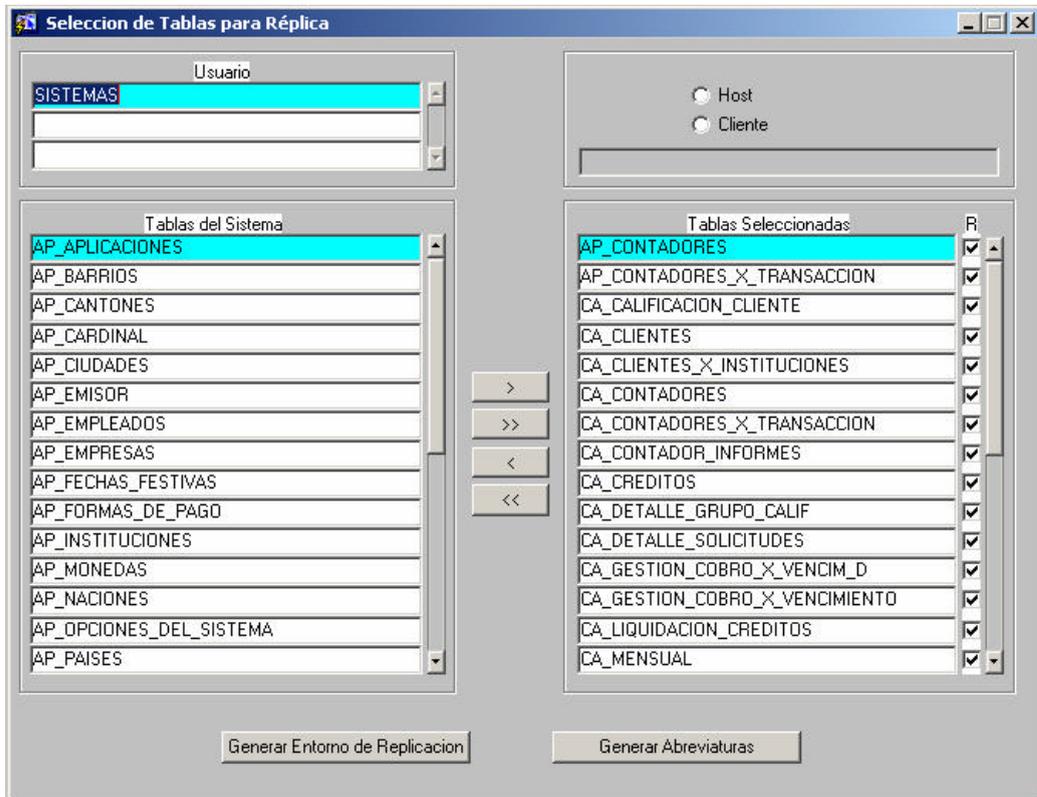
Luego procedemos a crear el directorio **dir_datos_carga** en la base de datos que apunta al directorio físico del sistema operativo `y:\planos_replica`, este directorio nos sirve para saber de donde toma las tablas externas su archivo de referencia; por lo que en el proceso de consolidación, el modulo SIS003 tomará los archivos planos del directorio `y:\mails_replica` y los pasará en el formato deseado al directorio `y:\planos_replica` para ser tomados como tablas externas.

Posteriormente se crean en la base de datos las tablas externas y procedimientos de carga de cada una de las tablas a replicar mediante el programa SIS001 operado con la opción de HOST. Cabe mencionar que las tablas que serán cargadas deben ser las mismas que fueron definidas para la replicación en las agencias.

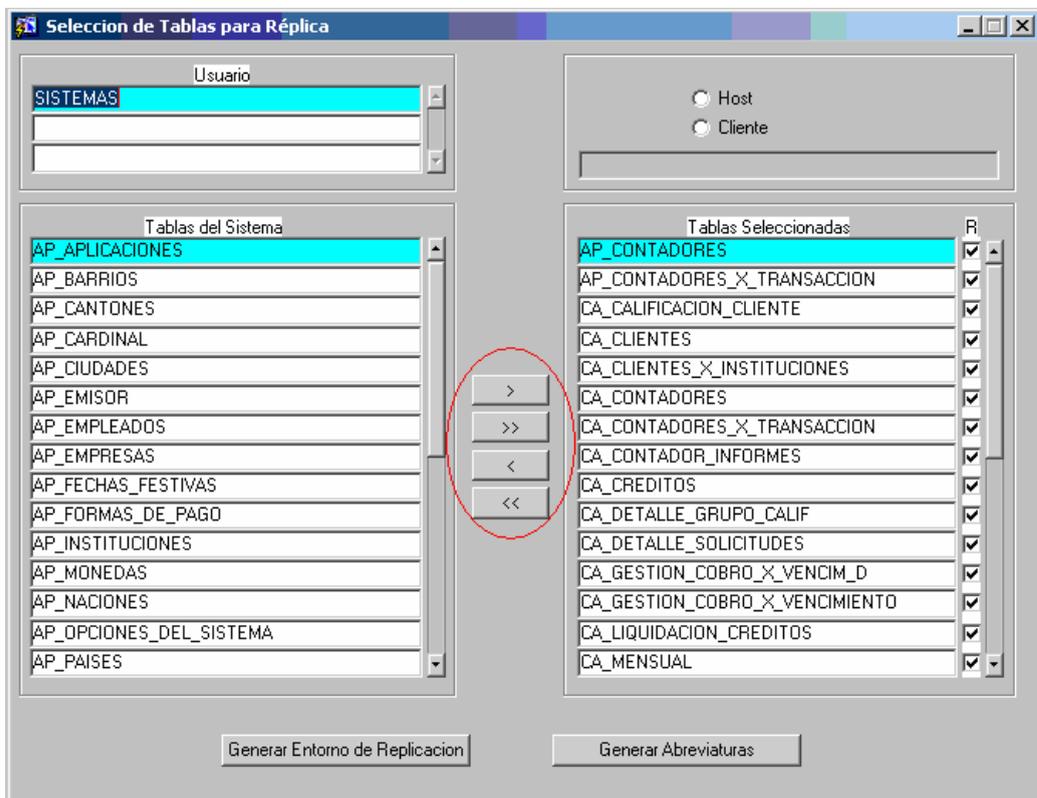
Pasos a seguir:

2) Definir las tablas que vamos a replicar

Mediante el formulario SIS001 que veremos en la siguiente figura:



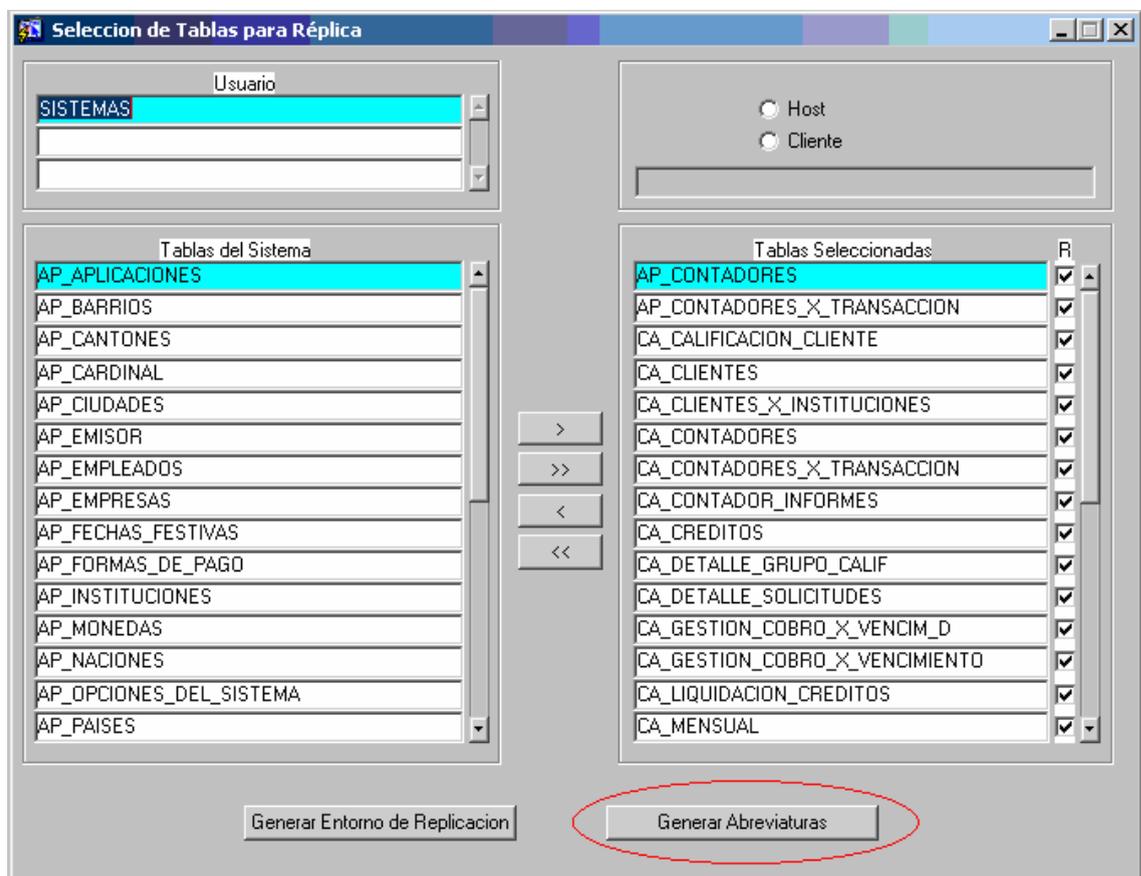
Seleccionaremos primeramente el usuario propietario de las tablas que vamos a replicar, al seleccionar el usuario en la parte inferior del mismo aparecerán todas las tablas de su propiedad y mediante los botones PASAR UNO o PASAR TODAS las seleccionaremos.



Una vez seleccionadas se pasaran al panel derecho en donde estarán ubicadas, valga la redundancia, todas las tablas de dicho usuario que han sido seleccionadas para el proceso de replicación .

2) Creación de Abreviaturas

Al igual que en el proceso de generación de replica de las agencias, existen algunas tablas de los diversos esquemas de base de datos en las que trabajamos, que poseen nombres sumamente largos por lo que es necesario crear abreviaturas para cada tabla replicada es decir vamos a convertir internamente el nombre de la tabla real en un nombre virtual que lo utilizaremos internamente en el proceso de replicación para referirnos a dicha tabla. Esto lo realizaremos haciendo click en el botón “Generar Abreviaturas” ubicado en la parte inferior derecha del formulario SIS001.



3) Definir el esquema de replicación

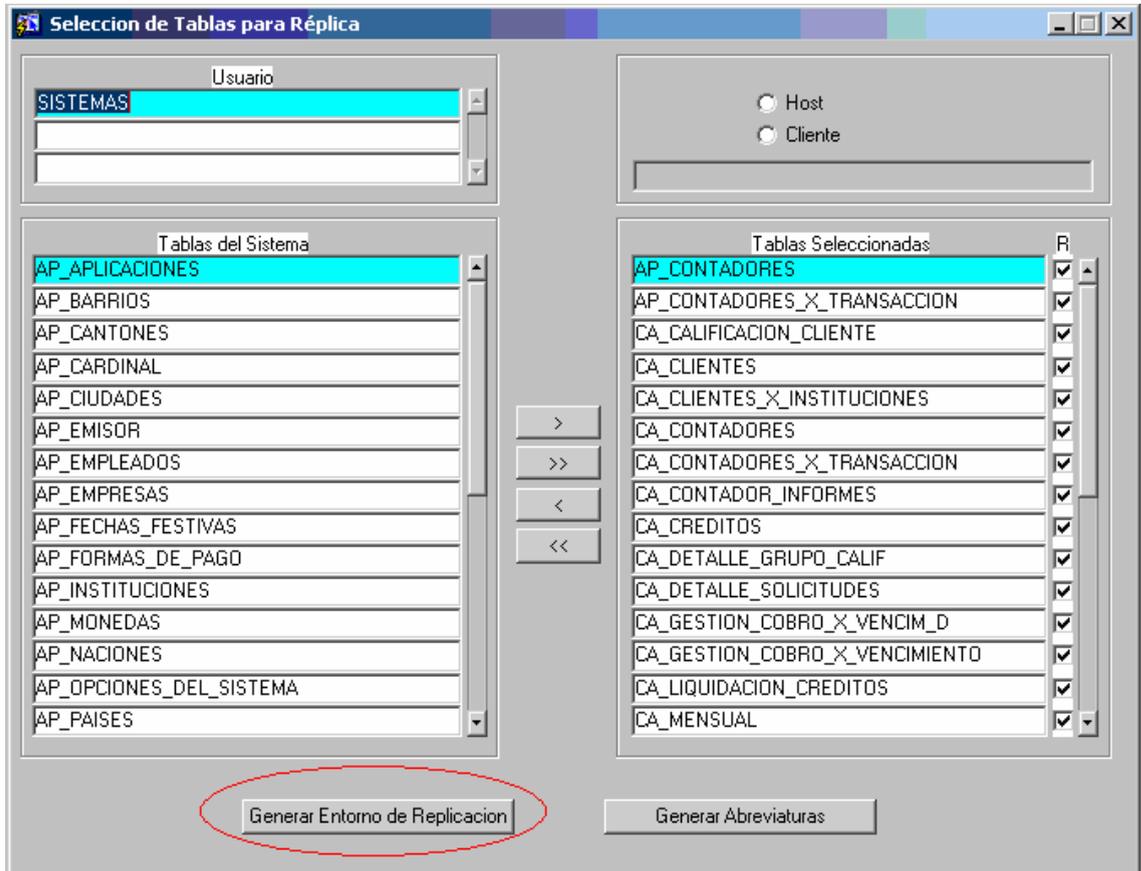
Dentro del proceso de replica de información tenemos dos esquemas claramente establecidos:

- Host
- Cliente

Para el ambiente de replicación de la matriz siempre escogeremos el modo Host

Host
 Cliente

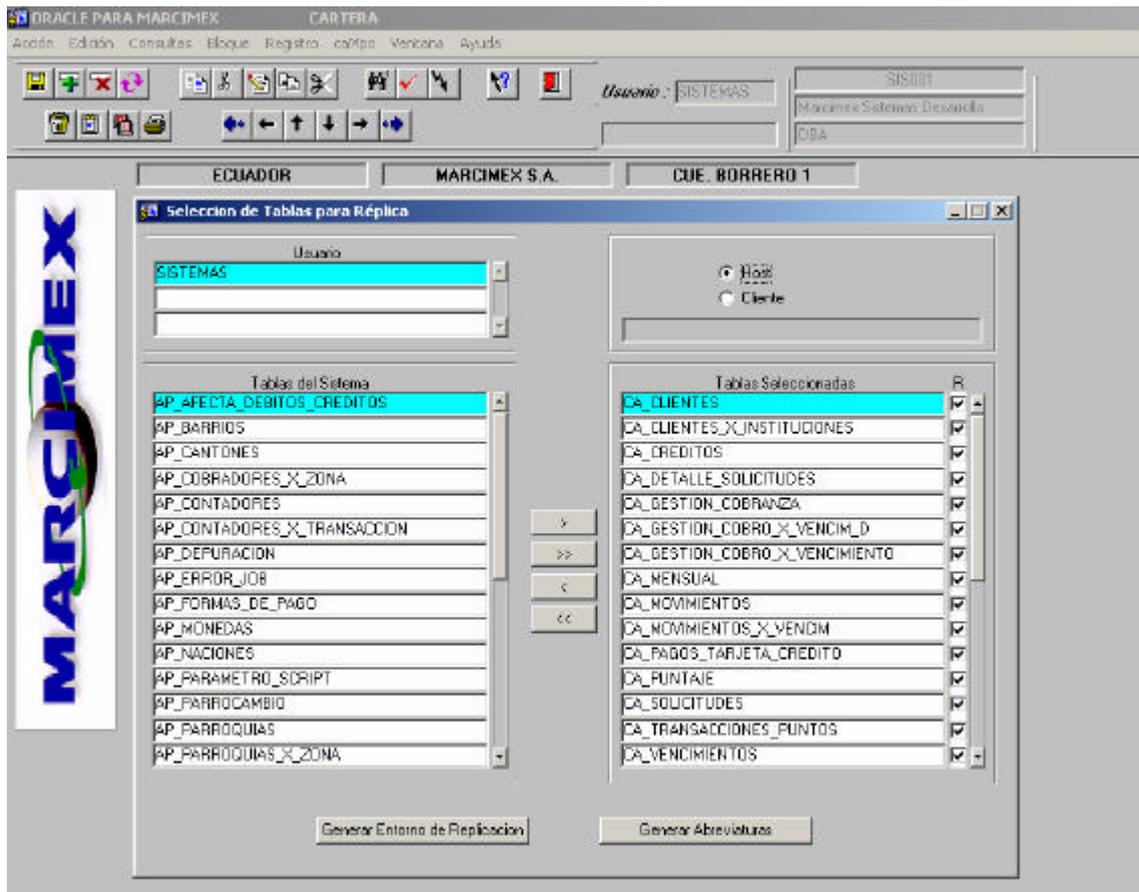
Posteriormente seleccionaremos el botón “Generar Entorno de Replicación”



Generar Entorno de Replicación en modo Host realizará lo siguiente:

- Graba los cambios realizados en la pantalla, es decir, las tablas que fueron marcadas quedan guardadas como las tablas a ser consolidadas por la replicación.
- Creará las tablas externas y procedimientos almacenados para la carga de datos de cada tabla que se consolide, esta creación de las tablas externas y procedimientos se lo hace con código dinámico, es decir, se crean automáticamente la codificación de creación de un objeto de la base de datos y dentro de la mismo proceso se ejecutan esas sentencias y se crean los objetos inmediatamente, todo esto se lo hace en un cursor (llamados sucesivos) hasta completar las condiciones del mismo dando un gran poder de manejo en la

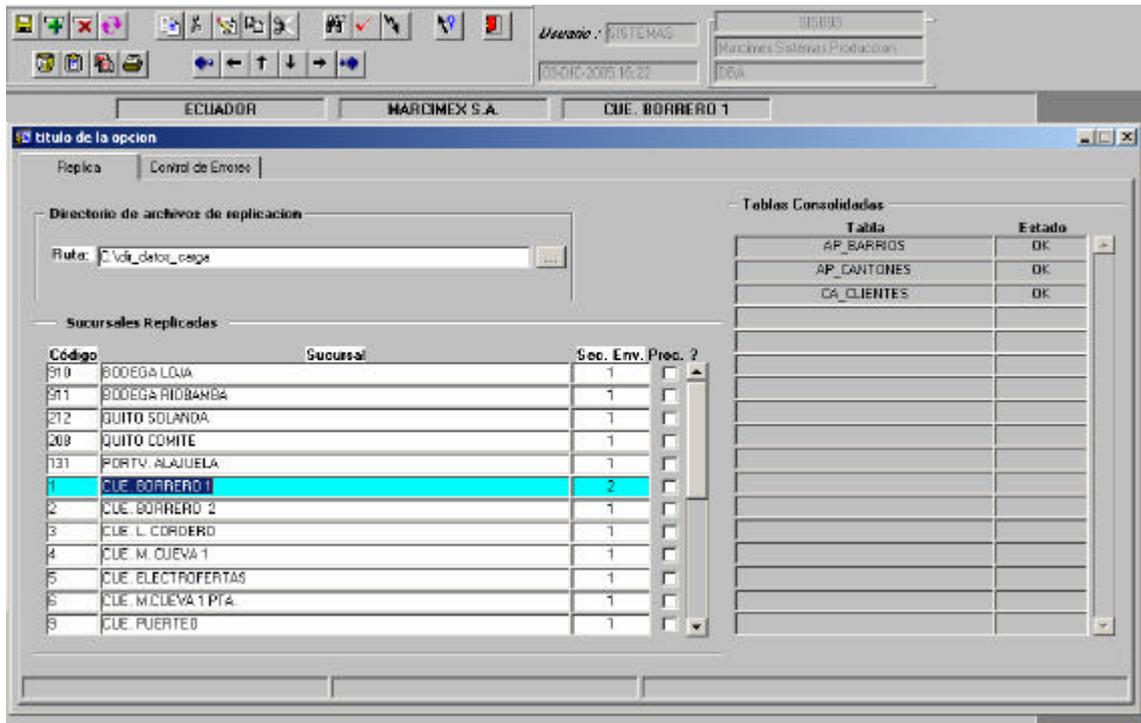
creación de objetos, evitando un proceso manual por parte de un programador de sql.



Una vez generado el ambiente para las tablas replicadas en la matriz, el sistema de replicación queda listo para proceder a la consolidación de la información a través del modulo SIS003.

3.2 Creación del modulo de replicación en la matriz.

La pantalla SIS003 es la encargada de subir la información desde los archivos planos enviados de las agencias hacia las tablas del servidor matriz.



En esta pantalla, tenemos un bloque con todas las sucursales que tienen un servidor de agencia, de las cuáles vamos a consolidar su información en el servidor matriz. En el caso de la pantalla tenemos a la agencia 1 CUE. BORRERO 1, de la cuál vamos a subir la secuencia de envío 1. Para ello debemos hacer los siguientes pasos:

- 1) Posesionarnos en el registro de la agencia que queremos subir la información.
- 2) Elegir la ruta de donde tomaremos los archivos planos, definidos en el directorio de la base de datos, para nuestro caso es el y:\planos_replica.
- 3) Finalmente damos un click sobre campo Proc.? , y automáticamente comenzará el proceso de consolidación de información para esa agencia y para esa secuencia.
- 4) En caso de existir errores durante el proceso de carga, podremos ver dichos errores pulsando el tab Control de Errores. 

Al momento de realizar el proceso de consolidación, vamos revisando las tablas que se proceden a consolidar, mediante los resultados ubicados en la parte derecha de la pantalla principal. Si alguna tabla no completara su consolidación, entonces quedará marcada con el error que no permitió dicha consolidación.

Tablas Consolidadas	
Tabla	Estado
AP_BARRIOS	OK
AP_CANTONES	OK
CA_CLIENTES	OK

3.3 Validación de información y control de errores.

Para garantizar que la consolidación de la información se llevó a cabo satisfactoriamente, es necesario fijar ciertas validaciones y controles que nos ayudarán de manera global a determinar que el proceso de carga de la información fue completa.

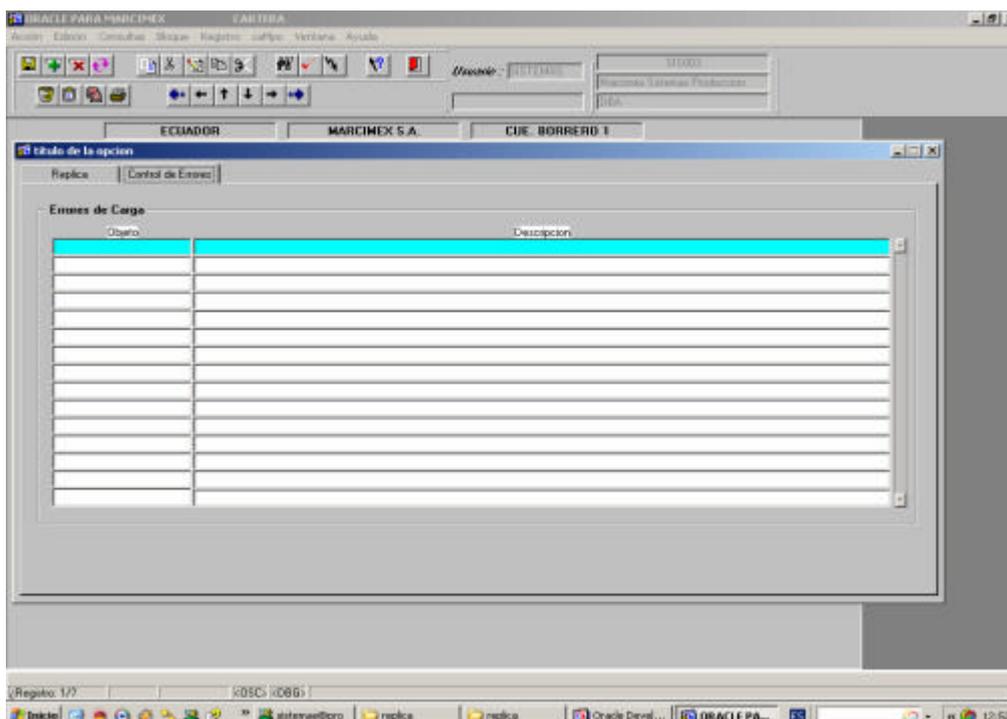
Validación de información.

En primer lugar determinemos que información vamos a verificar por cada carga de datos. Para ello hemos considerado llevar una validación respecto a:

- El número de tablas que deben consolidarse por sucursal, es decir, la cantidad de tablas a subir debe corresponder con el número de archivos físicos llegados por sucursal y por secuencia.
- La cantidad de registros a cargar, si por algún motivo el momento de cargar una tabla, ésta no se completara, nos presentará en la pantalla un mensaje de error en el campo Estado, en lugar del OK.
- Una validación adicional, consiste en verificar que todas las sucursales hayan subido la secuencia del día al que corresponde, para ello la persona encargada de la consolidación de la información, llevará un cuadro con las agencias y sus respectivas secuencias cargadas ya.

Control de Errores.

El momento que se presenta un desfase de información, el sistema controla el error a través de la pantalla *Control de Errores*. Dicha pantalla presentará la tabla que produjo el descuadre, junto con la descripción del error encontrado. Esta información la tenemos por cada sucursal y secuencia de envío que se esté consolidando.



Categorías de Error.- Para una mejor determinación de los posibles errores que se presenten en la fase de consolidación de la información dentro del modo de replicación desarrollado, hemos considerado unas categorías de dichos errores. Estas categorías son detalladas a continuación:

- a) Por archivos: Podemos tener errores al abrir los archivos de texto de los cuáles se basan las tablas externas y sus causas pueden ser:
 - Archivo dañado
 - No existe el archivo.
 - Directorio equivocado.
- b) Por cambio de estructura: De presentarse un cambio de estructura entre los datos de las tablas de agencia y la estructura de las tablas de matriz, se presentarán errores al cargar como:
 - Error en valor o número.
 - Longitud de dato erróneo.
 - Actualizar campos con datos equivocados.
- c) Por problemas con la Base de Datos: Este problema puede ser de distinta índole como los siguientes casos:
 - Error de espacio en el Tablespace de una tabla.
 - Base de datos caída o no activa.
 - Llaves foráneas activas
 - Violación de llaves primarias e índices únicos.
- d) Permisos de ejecución: A pesar que este tipo de error debe estar ya superado el momento de la implementación del módulo de replicación, es necesario decirlo, de manera que el usuario que ejecuta la consolidación de la información tenga todos los permisos "grant" hacia los objetos dentro de la base de datos.

Los siguientes puntos son necesarios para garantizar la carga completa de la información:

- Una vez determinado el error, debemos corregirlo según su detalle, esto se lo hace operativamente dependiendo del error.
- Corregido el error, el sistema nos permite cargar nuevamente la secuencia de envío de la agencia que no pudo ser completada, por lo que comenzará un nuevo intento sólo con las tablas que presentaron error. Este sistema permite tener identificados a las tablas que causaron error de carga.
- Verificar que no exista tabla alguna por subir para la secuencia de envío y sucursal dada.

CAPITULO 4

DESARROLLO DE MANUALES DE CONSULTA

Introducción.

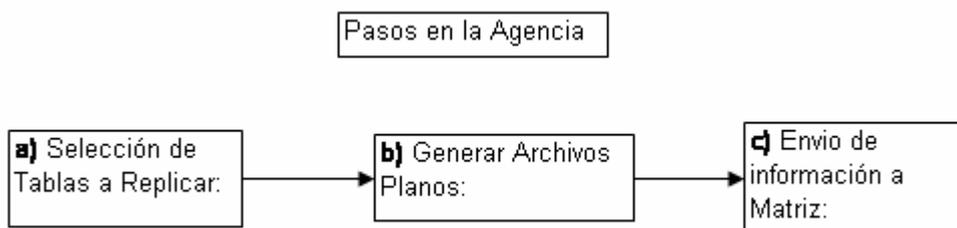
En este capítulo se explicará el funcionamiento de los procesos necesarios realizar operativamente para que la replicación sea llevada adecuadamente. Primero trataremos sobre una manual de consulta para un usuario de este sistema en una agencia (cliente), explicándole todos los pasos a seguir hasta cumplir con el primer objetivo de generar la replicación diaria de información de su agencia.

Segundo explicaremos como procederá un usuario de la aplicación en la Matriz (host), donde éste procede con la consolidación de la información a través de la parte del sistema de replicación que trata la carga y consolidación de la información de agencias.

4.1 Desarrollo de Manual de consulta de usuario de agencia.

El manual de consulta para un usuario de agencia se divide en dos: Una primera pantalla que se utiliza fundamentalmente al iniciar la replicación en esa agencia y que luego es necesaria para dar mantenimiento a dicha replicación; y, una segunda pantalla que es la que se ejecutará diariamente para generar y enviar la información replicada.

Diagrama de Pasos en Agencia:

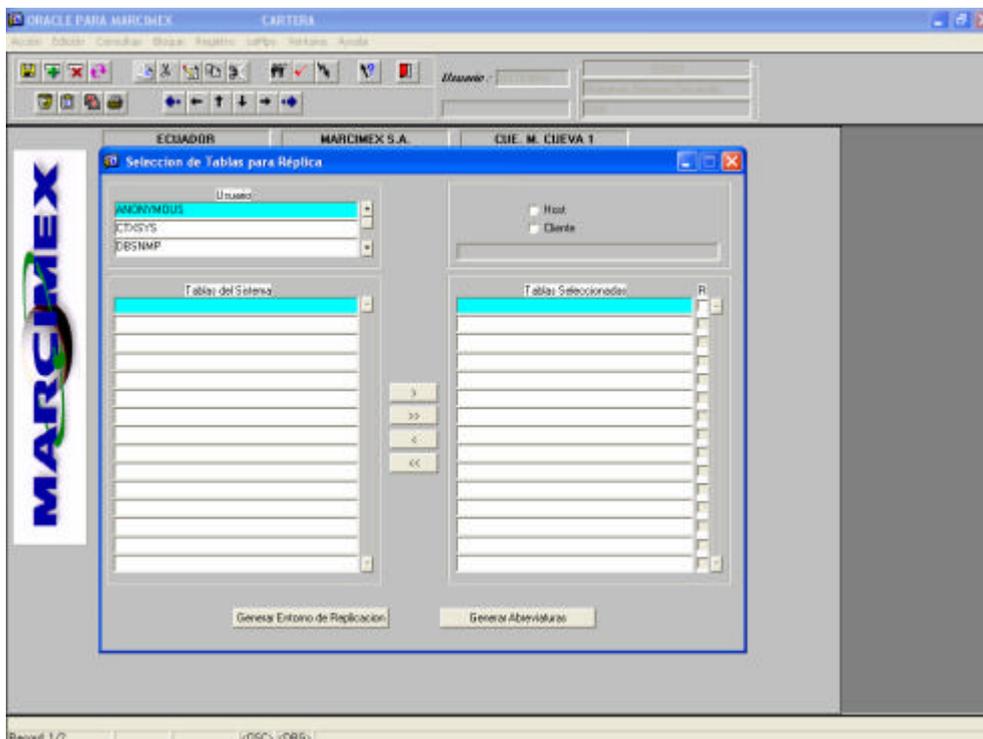


Selección de Tablas para Replica - Pantalla SIS001

Para una mejor disposición de la ubicación a las pantallas de la replicación, se implementarán los accesos directos de cada pantalla sobre el escritorio del servidor de la agencia.



Acceso: Escritorio del Servidor de Agencia → SIS001 usamos el ícono:



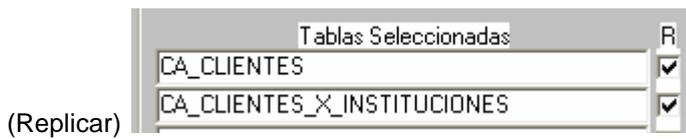
Al ingresar en esta pantalla el cursor se detiene en el bloque <<Usuario>>, inicialmente están listados todos los usuarios de la base de datos, de aquí señalaremos algunos pasos necesarios para poner a punto el esquema de replicación en la agencia.

Procedemos primero a buscar al usuario "SISTEMAS" que es el dueño de los objetos (tablas) que vamos a replicar. Esto lo hacemos pulsando <<F7>> ingresar criterio de búsqueda y luego <<F8>> ejecutar la consulta en el bloque <<Usuario>>. En ese momento el bloque <<Tablas del Sistema>> se llenará con todas las tablas que tenga la base de datos para ser replicadas.

2) Luego procedemos entonces a escoger las tablas que según las definiciones de los usuarios de matriz, son necesarias ser replicadas, damos un clic sobre el registro que contiene a la tabla a replicar y procedemos a pasarla al siguiente bloque <<Tablas Seleccionadas>> pulsando el botón ; y así sucesivamente hasta determinar todas las tablas que son necesarias replicar. Cabe indicar que podemos usar el botón si deseamos pasar todas las tablas de ese usuario al bloque << Tablas Seleccionadas>>; mientras las botones y cumplen respectivamente la función de quitar el registro señalado o todos los registros desde el bloque <<Tablas Seleccionadas>> hacia el bloque inicial <<Tablas del Sistema>>.

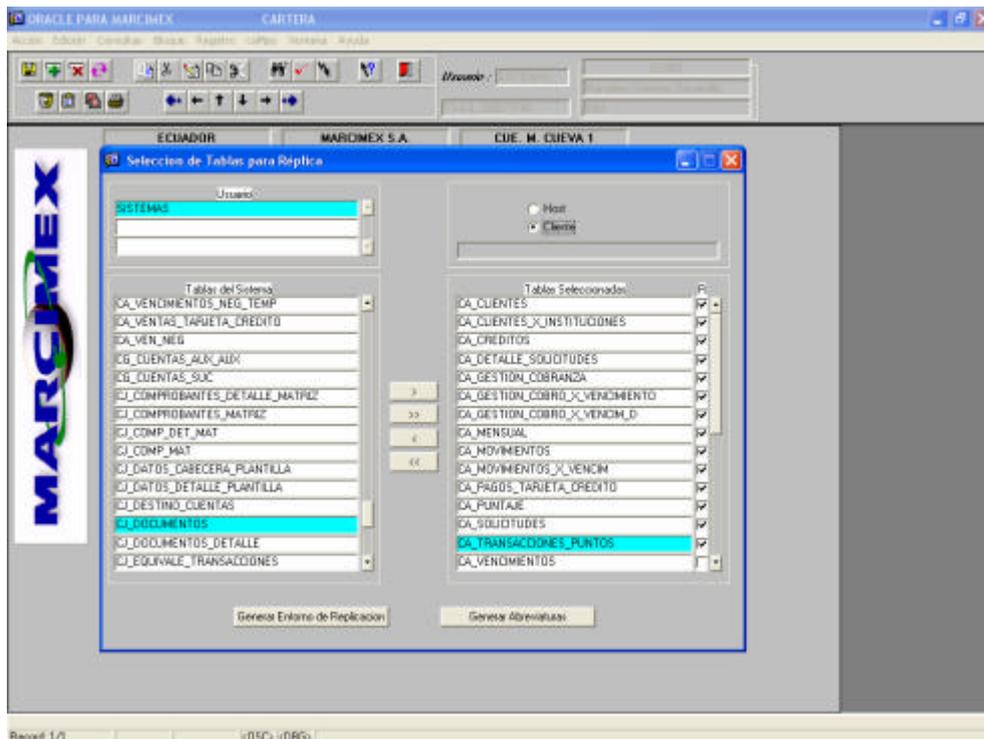
3) Una vez definidas las tablas a replicar escogemos el esquema de replicación, en el caso de

las agencias es “Cliente” Host Cliente y procedemos a tajar las tablas en el campo R



(Replicar)

Nos queda la información como la siguiente pantalla:



4) Luego de procedemos a crear las abreviaturas de las tablas seleccionadas, esto es necesario porque al momento de generar los procedimientos almacenados para la generación de los archivos planos así como los triggers almacenados para la replicación de cada tabla, los nombres de los objetos generados son muy largos y tenemos un límite de 30 caracteres en la base de datos por nombre de objeto.

Presionamos entonces el botón y se generan las abreviaturas.

5) Finalmente procedemos a presionar el botón con el que se procede a grabar las tablas seleccionadas y a crear desde ese momento el ambiente de replicación de información de la agencia.

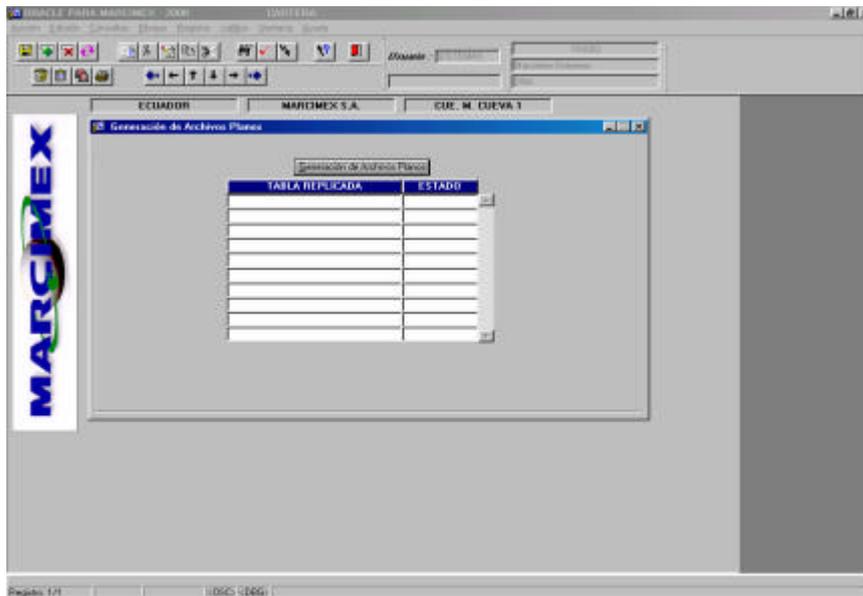
b) Generación de Archivos Planos - Pantalla SIS002

Al final de la jornada de trabajo, cuando ya se han realizado todas las acciones del cierre de día de una agencia, debemos proceder con la generación de archivos planos de la replicación del día; es la última acción del usuario en el sistema.

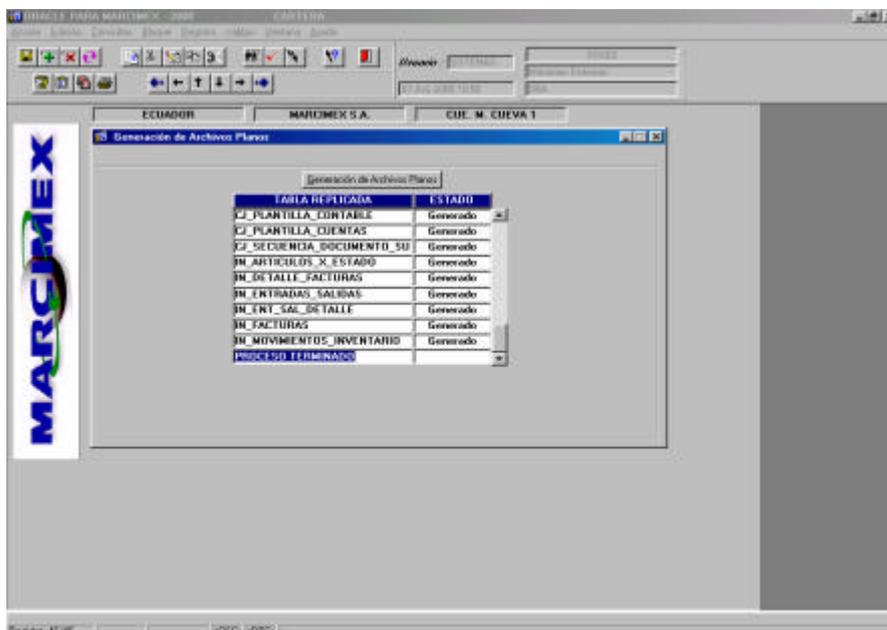


Acceso: Escritorio del Servidor de Agencia → SIS002 usamos el ícono:

Nos presenta la siguiente pantalla donde pulsamos el botón **Generación de Archivos Planos** y comienza a generarse los archivos planos de las tablas a replicar.



Al final del proceso deben completarse la generación de todas las tablas y cuando tengamos la siguiente pantalla entonces el proceso habrá terminado.



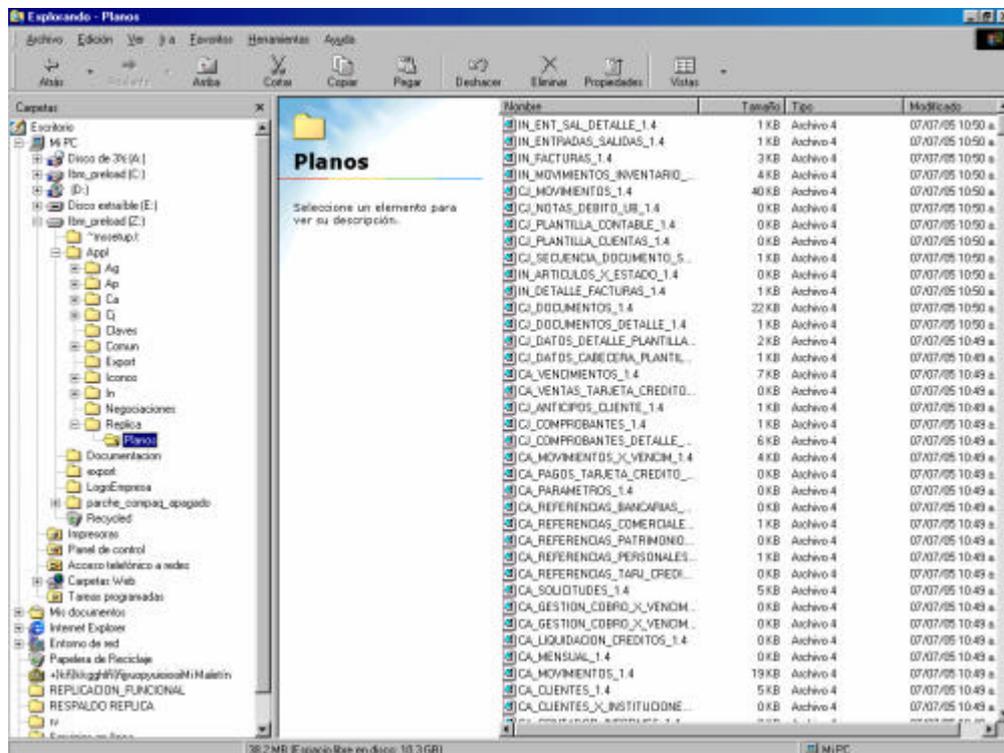
Entonces se habrán generado los archivos planos de la siguiente manera:

- Se guardan en el directorio z:\appl\replica\planos.
- Se crea un archivo plano por cada tabla replicada.
- El nombre de cada archivo se conforma de:

Nombre de la tabla (ca_credits), seguido de un guión bajo (_), luego una secuencia sucesiva de envío(1). Y tendrá el código de la agencia donde se realiza la replica como extensión. Ejm: ca_credits_1.4

c) Ubicación de archivos y envío a matriz

La siguiente pantalla nos muestra que archivos se generaron y la carpeta que los contiene de la generación de archivos planos de replicación de la agencia Mariano Cueva (# 4).



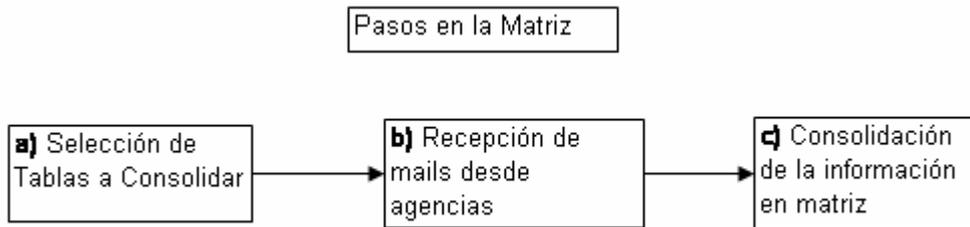
Una vez ubicados los archivos los seleccionamos los correspondientes a la generación de esta secuencia (ejm secuencia = 1), y los enviamos vía e-mail, para esta tesis probaremos el envío de correo a la dirección de correo electrónico replica@marcimex.com.ec.

Para un mejor uso de los archivos y no tener problemas de olvido de selección de alguno de ellos, es recomendable pasar los archivos generados diariamente en el directorio z:\appl\replica\planos a otro archivo de respaldo, de manera que cuando se proceda a la búsqueda y selección de los archivos generados en el día, tengamos una sola secuencia en el directorio especificado.

4.2 Desarrollo de Manual de Consulta de usuario de matriz.

El manual de consulta para un usuario de Matriz se divide en dos: Una primera pantalla que se utiliza fundamentalmente para la configuración inicial, y luego el posterior mantenimiento de la consolidación en la matriz; y, una segunda pantalla que es la que se ejecutará diariamente para consolidar la información replicada.

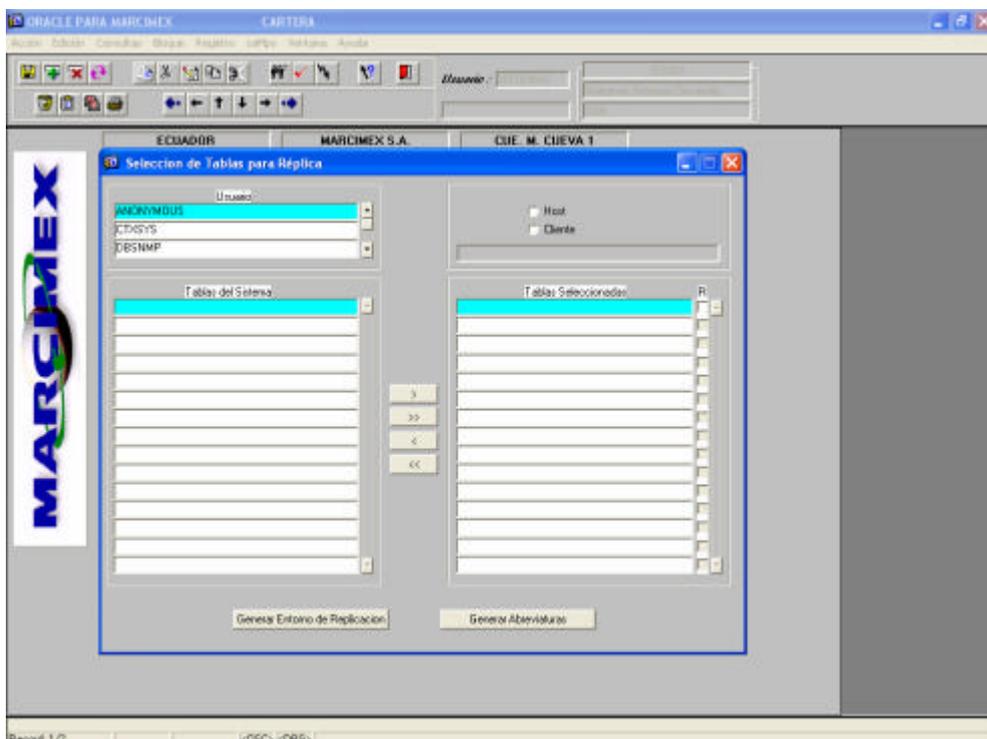
Diagrama de Pasos en Matriz:



Selección de Tablas para Consolidación - Pantalla SIS001

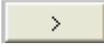


Acceso: Escritorio del Equipo de Consolidación → SIS001 usamos el ícono:

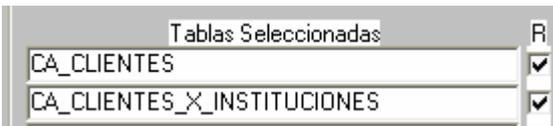


Al ingresar en esta pantalla el cursor se detiene en el bloque <<Usuario>>, inicialmente están listados todos los usuarios de la base de datos. Señalaremos algunos pasos necesarios para poner a punto el esquema de consolidación de información en la matriz.

Procedemos primero a buscar al usuario "SISTEMAS" que es el dueño de los objetos (tablas) que vamos a consolidar. Esto lo hacemos pulsando <<F7>> ingresar criterio de búsqueda y luego <<F8>> ejecutar la consulta en el bloque <<Usuario>>. En ese momento el bloque <<Tablas del Sistema>> se llenará con todas las tablas de dicho esquema que tenga la base de datos.

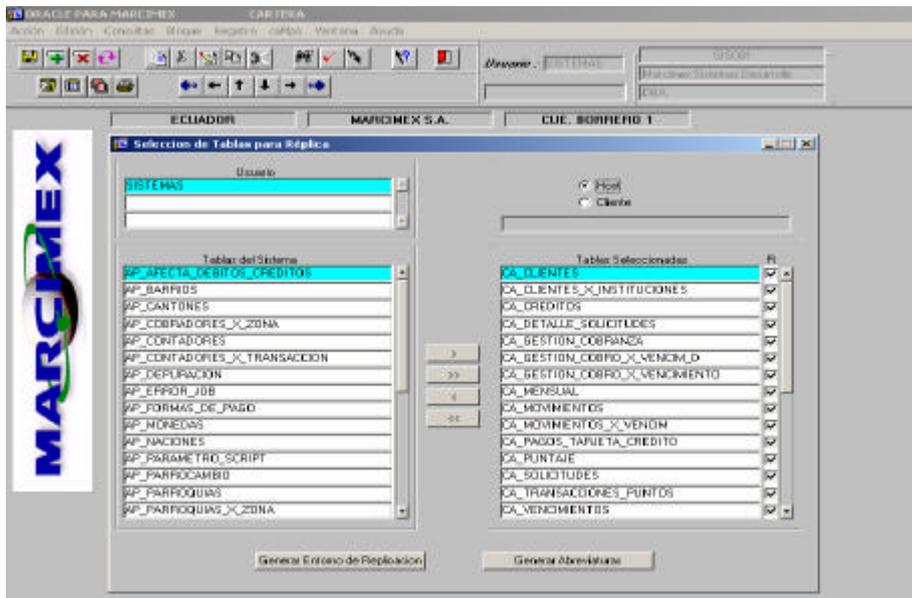
Luego procedemos entonces a escoger las tablas que fueron definidas para ser replicadas en las agencias, damos un clic sobre el registro que contiene a la tabla a consolidar y procedemos a pasarla al siguiente bloque <<Tablas Seleccionadas>> pulsando el botón ; y así sucesivamente hasta determinar todas las tablas que son necesarias para el proceso de consolidación. Cabe indicar que podemos usar el botón  si deseamos pasar todas las tablas de ese usuario al bloque << Tablas Seleccionadas>>; mientras los botones  y  cumplen respectivamente la función de quitar el registro señalado o todos los registros desde el bloque <<Tablas Seleccionadas>> hacia el bloque inicial <<Tablas del Sistema>>.

Una vez seleccionadas las tablas que van a intervenir en el proceso de consolidación, seleccionamos el entorno <<Host>> mediante el radio button correspondiente  y procedemos a tajar las tablas en el campo R (Replicar)



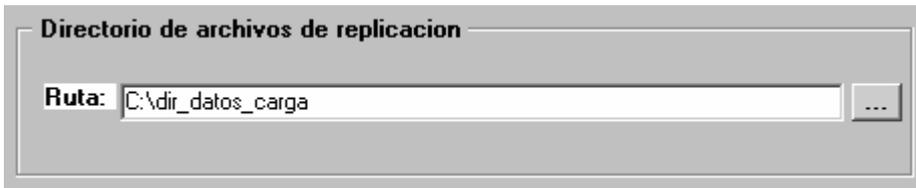
nos queda la información como la siguiente

pantalla:

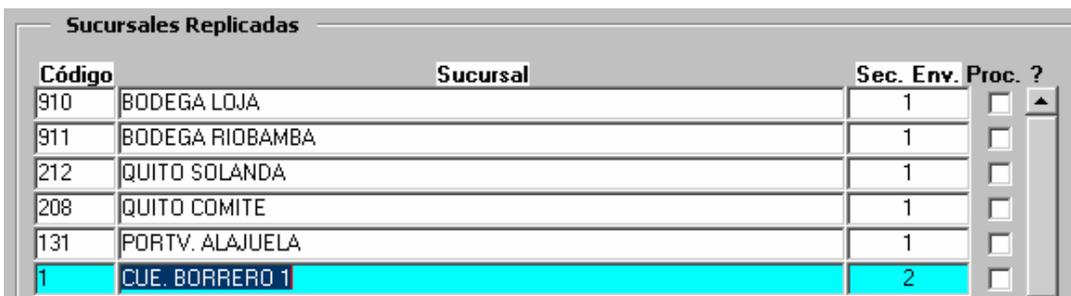


Para proceder con la consolidación realizamos lo siguiente:

Elegimos el directorio donde se encuentran ubicados los archivos con la información de la replicación de las agencias. Esto lo hacemos a través del campo <<Ruta>>



Nos posicionaremos sobre la agencia que vamos a consolidar.



Código	Sucursal	Sec. Env.	Proc. ?
910	BODEGA LOJA	1	<input type="checkbox"/>
911	BODEGA RIOBAMBA	1	<input type="checkbox"/>
212	QUITO SOLANDA	1	<input type="checkbox"/>
208	QUITO COMITE	1	<input type="checkbox"/>
131	PORTV. ALAJUELA	1	<input type="checkbox"/>
1	CUE. BORRERO 1	2	<input type="checkbox"/>

En este punto cabe señalar que la secuencia a subir dada por el campo <<Sec. Env.>> del bloque “Sucursales Replicadas” debe ser la secuencia correspondiente al lote de archivos planos de replicación enviados ese día por la agencia.

Proc.

Procedemos a tajar en el campo <<Proc. ?>> y automáticamente comienza el proceso de consolidación. Durante este proceso en el bloque <<Tablas Consolidadas>> se mostraran progresivamente el avance de la consolidación de cada una de las tablas. Al final del proceso de carga de cada tabla, en el campo <<Estado>> del mismo bloque, se colocará el estado en el que finalizo la carga de dicha tabla. Ejemplo:



Tabla	Estado
AP_BARRIOS	OK
AP_CANTONES	OK
CA_CLIENTES	OK

Si todas las tablas tienen el estado OK, decimos que el proceso de consolidación de dicha agencia ha concluido exitosamente, caso contrario podríamos revisar los errores generados durante el proceso mediante el Tab <<Control de Errores>>



CAPITULO 5

DOCUMENTACION DE LA APLICACION

Introducción.

En este capitulo describiremos los detalles técnicos de los objetos y formas que componen el sistema de replica objeto de nuestro estudio.

5.1 Tablas

5.1.1 SIS_TABLAS_REPLICADAS

Descripción: En esta tabla se guardan los nombres de todas las tablas que queremos que se repliquen en el servidor de matriz, así como un estado que nos indica si el objeto esta activo (A) o inactivo (I) para la replicación (únicamente se replicaran las tablas con estado A). El contenido de esta tabla a nivel de nombres de tablas y estados tiene que ser idéntico tanto en la agencia como en la matriz, en caso que exista una tabla que este definida en la agencia y no en la matriz, el sistema de replica asumiría que dicha tabla no debe ser cargada y estaríamos replicando y enviando dicha información a la matriz sin que en esta llegue a consolidarse. De modo contrario si tendríamos una tabla definida en la matriz y no en la agencia, al momento de cargar los datos no existiría la tabla a la cual se hace referencia y ocurriría un error de “no data found” en el servidor de matriz.

5.1.1.1 Script de Creación:

```
create table SIS_TABLAS_REPLICADAS
(
  CODIGO_SERVIDOR    NUMBER(4) not null,
  DUENO              VARCHAR2(10) not null,
  NOMBRE_TABLA       VARCHAR2(30) not null,
  ESTADO             VARCHAR2(1) not null,
  ADICIONADO_POR     VARCHAR2(10) not null,
  FECHA_ADICION      DATE not null,
  MODIFICADO_POR     VARCHAR2(10),
  FECHA_MODIFICACION DATE
)
tablespace TB_APPL_RP
pctfree 10
pctused 40
intrans 1
maxtrans 255
storage
```

```
(
  initial 304K
  minextents 1
  maxextents unlimited
);
```

```
alter table SIS_TABLAS_REPLICADAS
  add constraint SIS_PK_TABLAS_REPLICADAS primary key
(CODIGO_SERVIDOR, DUENO, NOMBRE_TABLA)
  using index
  tablespace TB_APPL_RP_IDX
  pctfree 10
  initrans 2
  maxtrans 255
  storage
  (
    initial 200K
    minextents 1
    maxextents unlimited
  );
```

5.1.2 SIS_ABREV_OBJETOS

Descripción: En vista de que la mayor parte de objetos dinámicos (creados en tiempo de ejecución) como triggers, tablas externas y procedimientos de carga utilizados en el esquema de replicación se generan antecediendo un prefijo al nombre de la tabla que afectan, por ejemplo el trigger que guarda los cambios realizados sobre la tabla AP_SUCURSALES recibe el nombre de T_RPL_AP_SUCURSALES, existen algunas tablas que poseen un nombre cuya longitud sumado el prefijo exceden el límite permitido por Oracle para un nombre de objeto (30 caracteres) tuvimos que crear una tabla de correspondencia entre el nombre real de la tabla (AP_SUCURSALES) y un nombre generado por el sistema mediante un número secuencial por ejemplo para la tabla AP_SUCURSALES su nombre interno sería T000001. Con esto obtendremos nombres internos de tablas compuestos de siete caracteres, terminando de esta manera el problema con los nombres largos. Dicha correspondencia entre los nombres reales y los nombres internos de tablas las guardaremos en esta tabla.

5.1.2.1 Script de Creación:

```
create table SIS_ABREV_OBJETOS
  (
    NOMBRE_OBJETO_REAL VARCHAR2(255) not null,
```

```

    APLICATIVO          VARCHAR2(5) not null,
    ABREVIATURA_INTERNA VARCHAR2(50),
    DUENO               VARCHAR2(50) not null,
    TIPO_OBJETO        VARCHAR2(20)
)
tablespace TB_APPL_RP
pctfree 10
pctused 40
initrans 1
maxtrans 255
storage
(
    initial 64K
    minextents 1
    maxextents unlimited
);
alter table SIS_ABREV_OBJETOS
add constraint PK_SIS_ABREV_OBJ primary key (DUENO, APLICATIVO,
NOMBRE_OBJETO_REAL)
using index
tablespace TB_APPL_RP_IDX
pctfree 10
initrans 2
maxtrans 255
storage
(
    initial 64K
    minextents 1
    maxextents unlimited
);

```

5.1.3 SIS_REPLICACION

Descripción: Esta tabla es el centro mismo de la replicación, cuando un usuario realiza una operación DML ya sea por inserción, actualización o eliminación sobre alguna tabla cuyo nombre conste en la tabla SIS_TABLAS_REPLICADAS como activo, dicho cambio será registrado en esta tabla mediante los siguientes campos:

- ACCION: Operación DML realizada I = inserción, M = modificación, D = eliminación.
- NOMBRE_TABLA: Nombre de la tabla sobre la cual se realizó el cambio.
- ROW_ID_TABLA: La identificación única del registro que se alteró.
- PK1, PK2 ... PK10: Los valores de la llave primaria del registro alterado.
- ESTADO: Nos indica si el registro esta pendiente de enviar a la matriz para su consolidación.
- SECUENCIA_ENVIO: Indica en que secuencia se envió a la matriz dicho cambio.
- FECHA_ENVÍO: Fecha en la cual se envió el cambio.

- Finalmente tenemos campos de auditoría como son ADICIONADO_POR, FECHA_ADICIÓN, MODIFICADO_POR, FECHA_MODIFICACION.

5.1.3.1 Script de Creación:

```

create table SIS_REPLICACION
(
  ACCION          VARCHAR2(1) not null,
  NOMBRE_TABLA   VARCHAR2(30) not null,
  ROWID_TABLA    VARCHAR2(30) not null,
  PK1             VARCHAR2(30),
  PK2             VARCHAR2(30),
  PK3             VARCHAR2(30),
  PK4             VARCHAR2(30),
  PK5             VARCHAR2(30),
  PK6             VARCHAR2(30),
  PK7             VARCHAR2(30),
  PK8             VARCHAR2(30),
  PK9             VARCHAR2(30),
  PK10           VARCHAR2(30),
  ESTADO         VARCHAR2(1),
  SECUENCIA_ENVIO NUMBER(4),
  FECHA_ENVIO    DATE,
  ADICIONADO_POR VARCHAR2(10) not null,
  FECHA_ADICION  DATE not null,
  MODIFICADO_POR VARCHAR2(10),
  FECHA_MODIFICACION DATE
)
tablespace TB_APPL_RP
pctfree 10
pctused 40
initrans 1
maxtrans 255
storage
(
  initial 304K
  minextents 1
  maxextents unlimited
);
alter table SIS_REPLICACION
add constraint SIS_PK_REPLICACION primary key (ACCION, NOMBRE_TABLA)
using index
tablespace TB_APPL_RP_IDX
pctfree 10
initrans 2
maxtrans 255
storage

```

```
(
  initial 104K
  minextents 1
  maxextents unlimited
);
```

5.1.4 SIS_ERRORES_CARGA

Descripción: Durante el proceso de carga de información podrían presentarse varios tipos de errores, por citar un ejemplo, podría ser que por algún motivo cambie la estructura de datos de la tabla de la agencia contra la estructura de la matriz, entonces en el proceso de carga, no correspondería la información que viene con la estructura en la que sería consolidada localmente, dicho error se debe registrar para al fin del proceso de carga poder decidir que hacer con el, si se lo podría pasar por alto para traerlo posteriormente o si se tratase de información estratégica que tendríamos que solucionarlo ese instante. Esto por citar un ejemplo de los muchos casos que podrían ocurrir durante el proceso de carga. En definitiva en esta tabla se guardara el error vinculado con el proceso de carga del cual proviene y de la tabla a la cual hace referencia, junto con la descripción técnica del mismo.

5.1.4.1 Script de Creación:

```
create table SIS_ERRORES_CARGA
(
  NOMBRE_DEL_OBJETO VARCHAR2(30) not null,
  DESCRIPCION       VARCHAR2(255) not null,
  ADICIONADO_POR   VARCHAR2(30),
  FECHA_ADICION    DATE,
  CODIGO_SUCURSAL  NUMBER(4) not null,
  SECUENCIA_ENVIO  NUMBER(10) not null
)
tablespace TB_APPL_RP
pctfree 10
pctused 40
intrans 1
maxtrans 255
storage
(
  initial 2M
  minextents 1
  maxextents unlimited
);
alter table SIS_ERRORES_CARGA
  add constraint PK_ERR_CARGA primary key (NOMBRE_DEL_OBJETO,
  SECUENCIA_ENVIO, CODIGO_SUCURSAL, DESCRIPCION)
  using index
```

```

tablespace SYSTEM
pctfree 10
initrans 2
maxtrans 255
storage
(
  initial 64K
  minextents 1
  maxextents unlimited
);

```

5.1.5 SIS_PLANOS_SUBIDOS

Descripción: Luego del proceso de consolidación de información en la Matriz realizada por el modulo SIS003, automáticamente dicho modulo actualiza el estado de el paquete de datos consolidado y le asigna una fecha que corresponde al momento en el cual terminó de cargarse la última fila del conjunto de datos de la replicación, la ausencia de esta fecha denota que dicha secuencia aún no ha sido cargada, cabe señalar que no es posible el salto de secuencias para la consolidación es decir si la agencia número 4, envía hoy la secuencia 10, el día de mañana el sistema esperara automáticamente la carga de la secuencia 11 de dicha agencia y no podríamos cargar las secuencias subsiguientes si las hubiere.

5.1.5.1 Script de Creación:

```

create table SIS_PLANOS_SUBIDOS
(
  CODIGO_SERVIDOR    NUMBER(4) not null,
  CODIGO_SUCURSAL    NUMBER(4) not null,
  NOMBRE_TABLA       VARCHAR2(30) not null,
  SECUENCIA_ENVIO    NUMBER(4) not null,
  FECHA_SUBIDA       DATE not null,
  ADICIONADO_POR     VARCHAR2(10) not null,
  FECHA_ADICION      DATE not null,
  MODIFICADO_POR     VARCHAR2(10),
  FECHA_MODIFICACION DATE
)
tablespace TB_APPL_RP
pctfree 10
pctused 40
initrans 1
maxtrans 255
storage
(
  initial 2M
  minextents 1

```

```

        maxextents unlimited
    );
alter table SIS_PLANOS_SUBIDOS
    add constraint SIS_PK_PLANOS_SUBIDOS primary key (CODIGO_SUCURSAL,
CODIGO_SERVIDOR, SECUENCIA_ENVIO, NOMBRE_TABLA)
    using index
    tablespace TB_APPL_RP_IDX
    pctfree 10
    initrans 2
    maxtrans 255
    storage
    (
        initial 800K
        minextents 1
        maxextents unlimited
    );

```

5.2 Tablas externas.

5.2.1 Nombre: datos_matriz

Descripción: Es una tabla externa cuya estructura de campos es idéntica a la estructura del archivo estructura_tablas.txt; nos sirve para tomar la información de este archivo y comparar con las estructuras de las tablas de cada agencia.

Datos_matriz

```

create table datos_matriz (
nombre_tabla varchar2(30),
nombre_columna varchar2(30),
tipo_de_datos varchar2(106),
longitud number,
precision_dat number,
escala number,
id_columna number
)
organization external
(
type oracle_loader
default directory comparadb
access parameters (
        records delimited by newline
        fields terminated by ';'
    )
)

```

```
location ('datos_matriz.txt')
)
reject limit 1000;
```

5.2.2 Nombre: EXT_nombre_tabla (Dinámica)

Descripción: Es una tabla externa cuya estructura de campos es idéntica a la estructura de campos del archivo plano generado desde el sitio de replicación, y que se monta sobre dicho archivo de tal manera que cualquier operación de selección que realicemos sobre la tabla, la estaremos haciendo sobre el propio archivo de texto. Estas tablas son creadas desde el procedimiento SIS_CREA_TABLA_EXT. En realidad las tablas externas fueron derivadas del programa externo de carga masiva de información LOADER.EXE que viene junto con el paquete de la base datos 8i y 9i o en sus respectivos clientes. Las tablas externas poseen ventajas y desventajas frente a su similar LOADER, entre las ventajas tenemos las siguientes:

- **Orden:** Al tener predefinidos los formatos de nuestros archivos externos de texto y su estructura (delimitación y tipo de datos) almacenados en nuestra base de datos, de una u otra forma veríamos a dichos archivos mucho mas cercanos al contexto de nuestra base de datos y por que no, los veríamos formando parte integral de la misma como cualquier otra tabla interna del sistema.
- **Manipulación:** Al convertir nuestros archivos de texto en "Tablas", podríamos realizar sobre éstas cierto tipo de sentencias SQL y PL/SQL como Selects y Cursores que permitirían manejar su contenido con mucha más facilidad que cargándolo en bloque con LOADER.
- **Optimización de Recursos:** El comando LOADER al formar parte integral de un paquete de programas ORACLE (Base de Datos 8i, 9i o Cliente) nos sugiere que este paquete debe estar instalado en la máquina en donde reposan los archivos planos, mientras que la tabla externa hace referencia a un directorio específico dentro del servidor principal, mismo que podríamos referenciar desde un cliente mediante una unidad de red virtual.
- **Dinámica:** Al ser un objeto de base de datos dicha entidad podría ser generada dinámicamente mediante sentencias DDL ejecutadas desde un paquete Sql Dinámico como DBMS_SQL o SQL DINAMICO NATIVO.

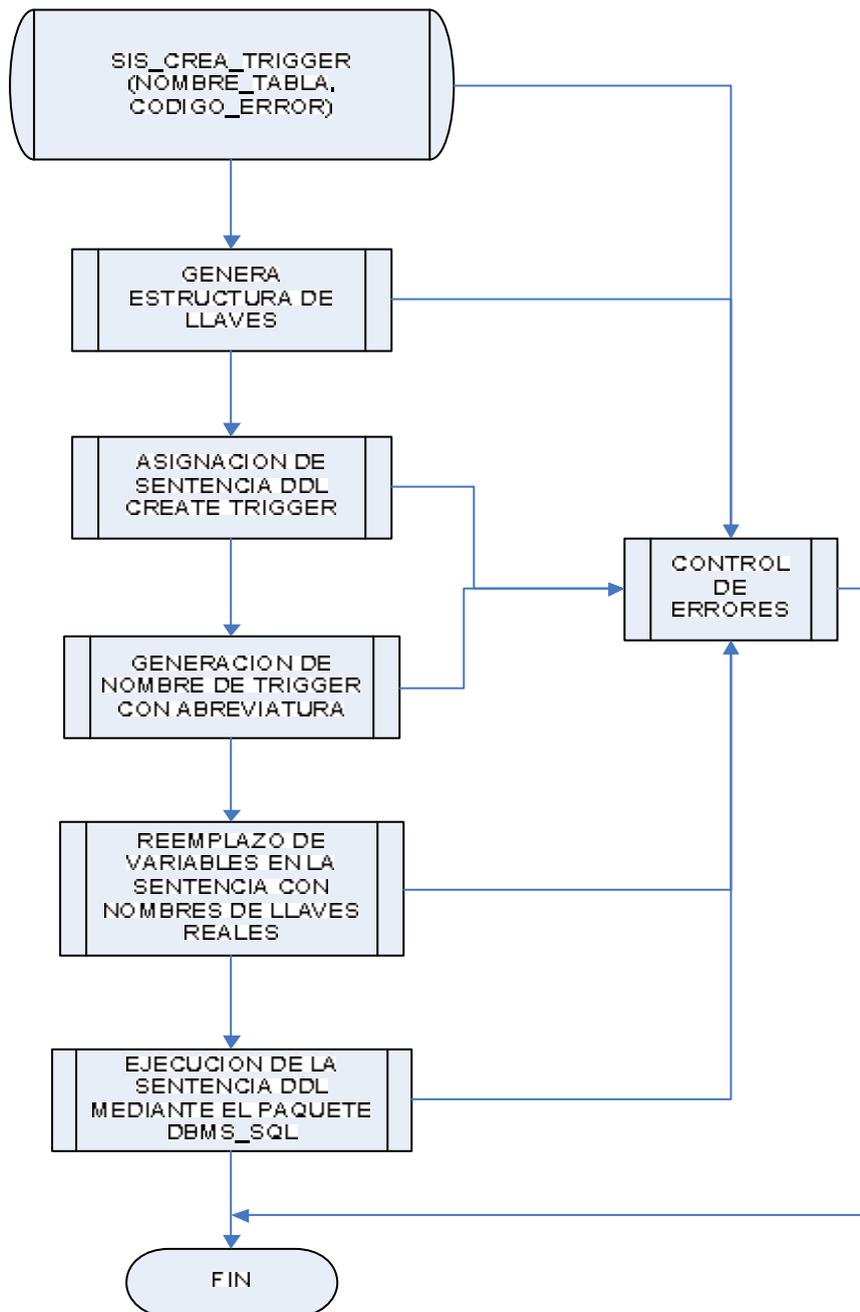
La principal desventaja frente a LOADER radica en la velocidad de acceso a los registros virtuales del archivo plano.

5.3.- Procedimientos

5.3.1 SIS_CREA_TRIGGER

Descripción: Es el procedimiento que generará dinámicamente los triggers que en el futuro monitorearán los cambios realizados sobre el grupo de tablas a replicarse. Debido al tamaño del script de creación del trigger final que pasaremos al motor de PL/SQL de la base de datos, hemos decidido utilizar el paquete DBMS_SQL para su generación y no SQL NATIVO.

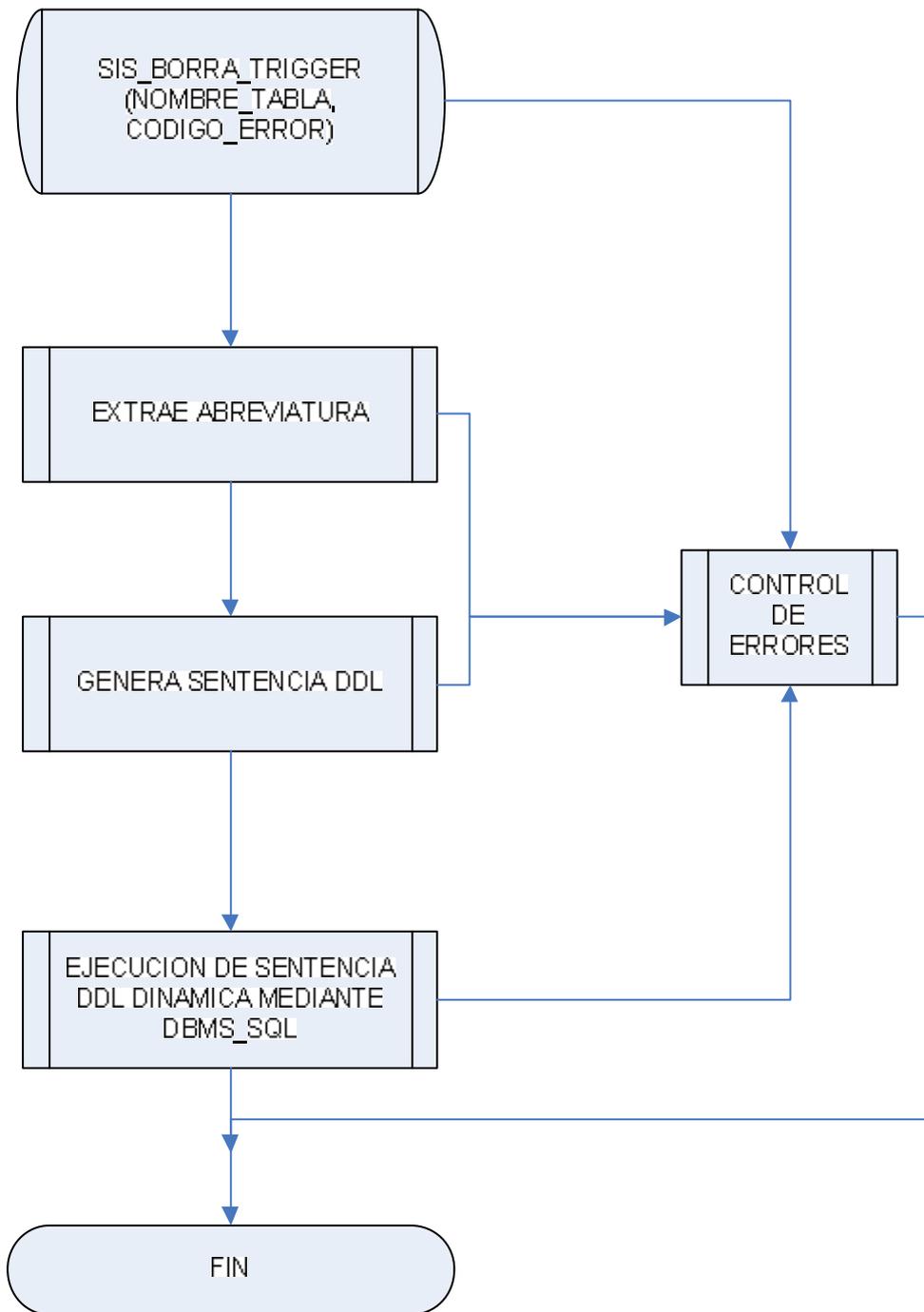
Diagrama:



5.3.2 SIS_BORRA_TRIGGER

Descripción: A la inversa del procedimiento anterior, este proceso eliminará de la(s) tabla(s) el trigger que monitorea los cambios realizados sobre dicha tabla en caso de que se decida que esa tabla ya no formará parte del grupo de replicación en un momento dado.

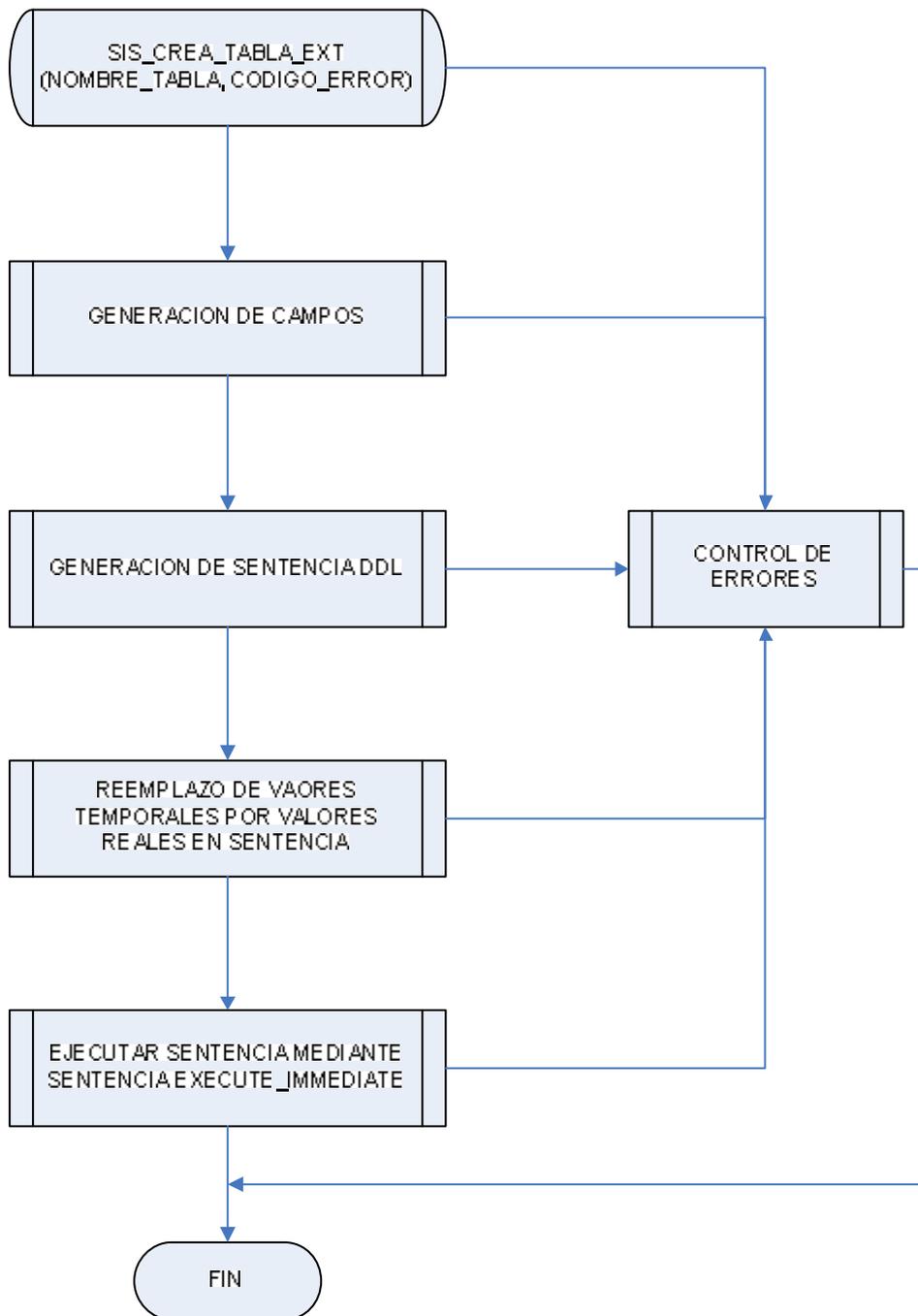
Diagrama:



5.3.3 SIS_CREA_TABLA_EXT

Descripción: Este proceso generará dinámicamente la tabla externa que será la encargada de recibir en el servidor de matriz la información proveniente de los archivos planos desde el sitio de replicación. Existirá una tabla externa por cada tabla que forme parte del grupo de replica, su nombre tendrá la siguiente estructura: *EXT_abrvnombretabla*.

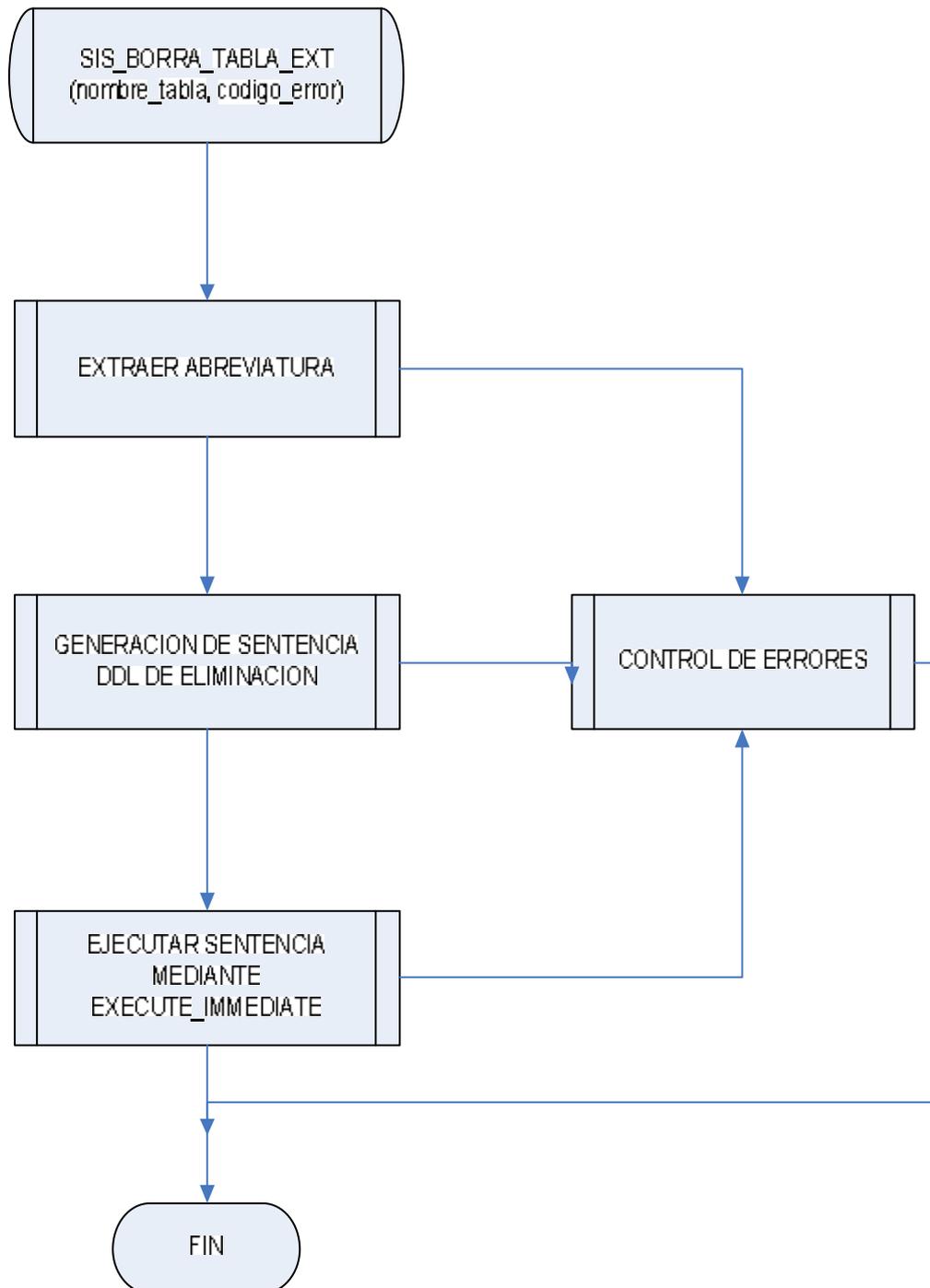
Diagrama:



5.3.4 SIS_BORRA_TABLA_EXT

Descripción: Eliminará de la base de datos una tabla externa específica en caso de que el objeto sobre el cual actúe deje de formar parte del grupo de replicación.

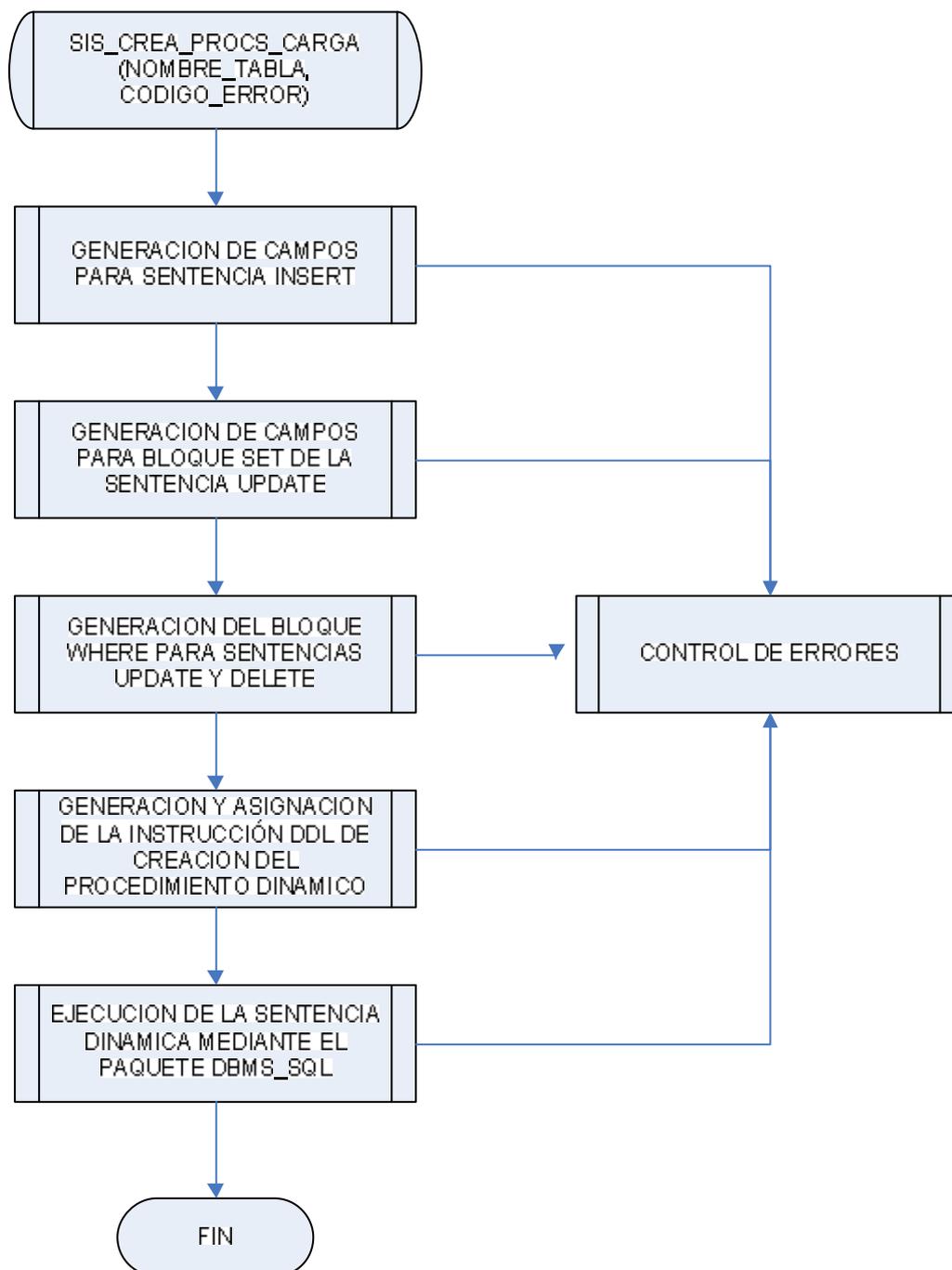
Diagrama:



5.3.5 SIS_CREA_PROCS_CARGA

Descripción: Crea dinámicamente el procedimiento que lee la información de la tabla externa correspondiente a una tabla a replicarse, y en ésta, realiza las operaciones DML de inserción, modificación y/o eliminación provenientes del sitio de replicación, existirá un procedimiento de carga por cada tabla que forme parte del grupo de replicación debido a que mantiene una correspondencia con los campos de la tabla externa sobre la cual actúe.

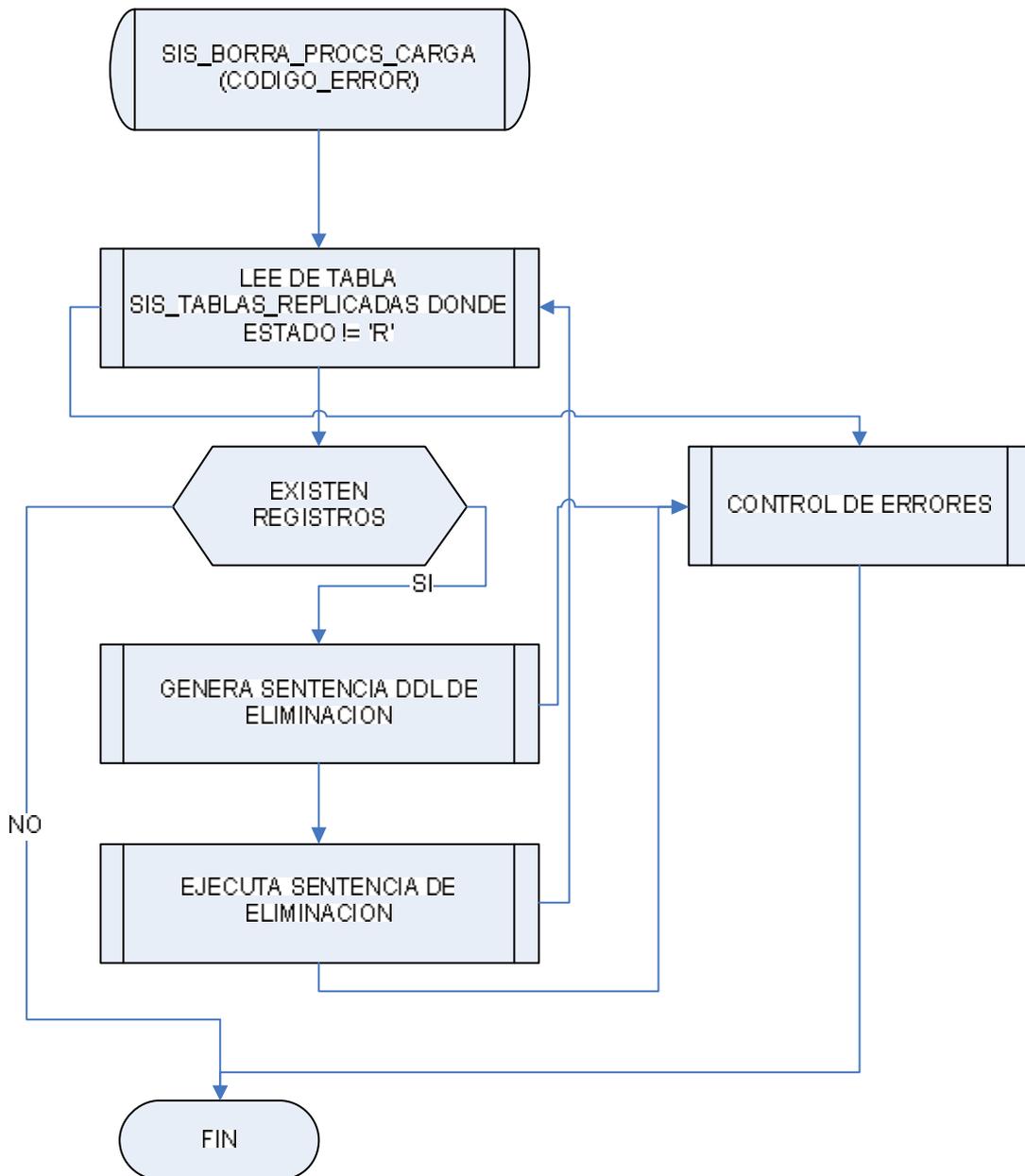
Diagrama:



5.3.6 SIS_BORRA_PROCS_CARGA

Descripción: Elimina un procedimiento de carga del servidor de matriz, en caso de que la tabla a la cuál esta relacionado deje de formar parte del grupo de replica en un momento dado.

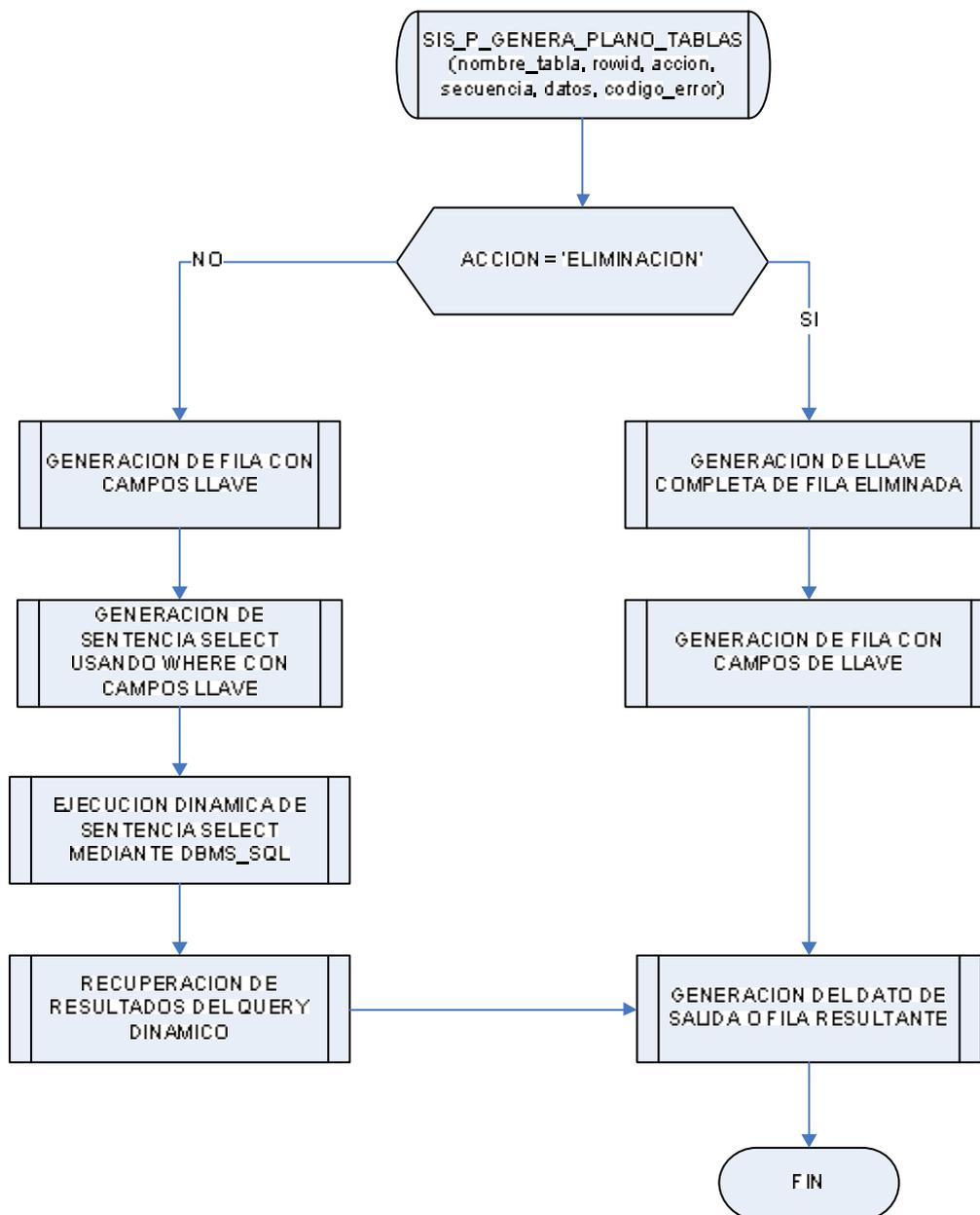
Diagrama:



5.3.7 SIS_GENERA_PLANO_TABLAS

Descripción: En el sitio de replicación, este procedimiento recoge todos los cambios realizados en un tiempo determinado desde la tabla SIS_REPLICACION que no hayan sido aún enviados al servidor de matriz y los convierte en un archivo plano separado por el signo “\$” en el cuál consta información sobre el cambio realizado, secuencia de envío y el registro completo de la tabla replicada.

Diagrama:

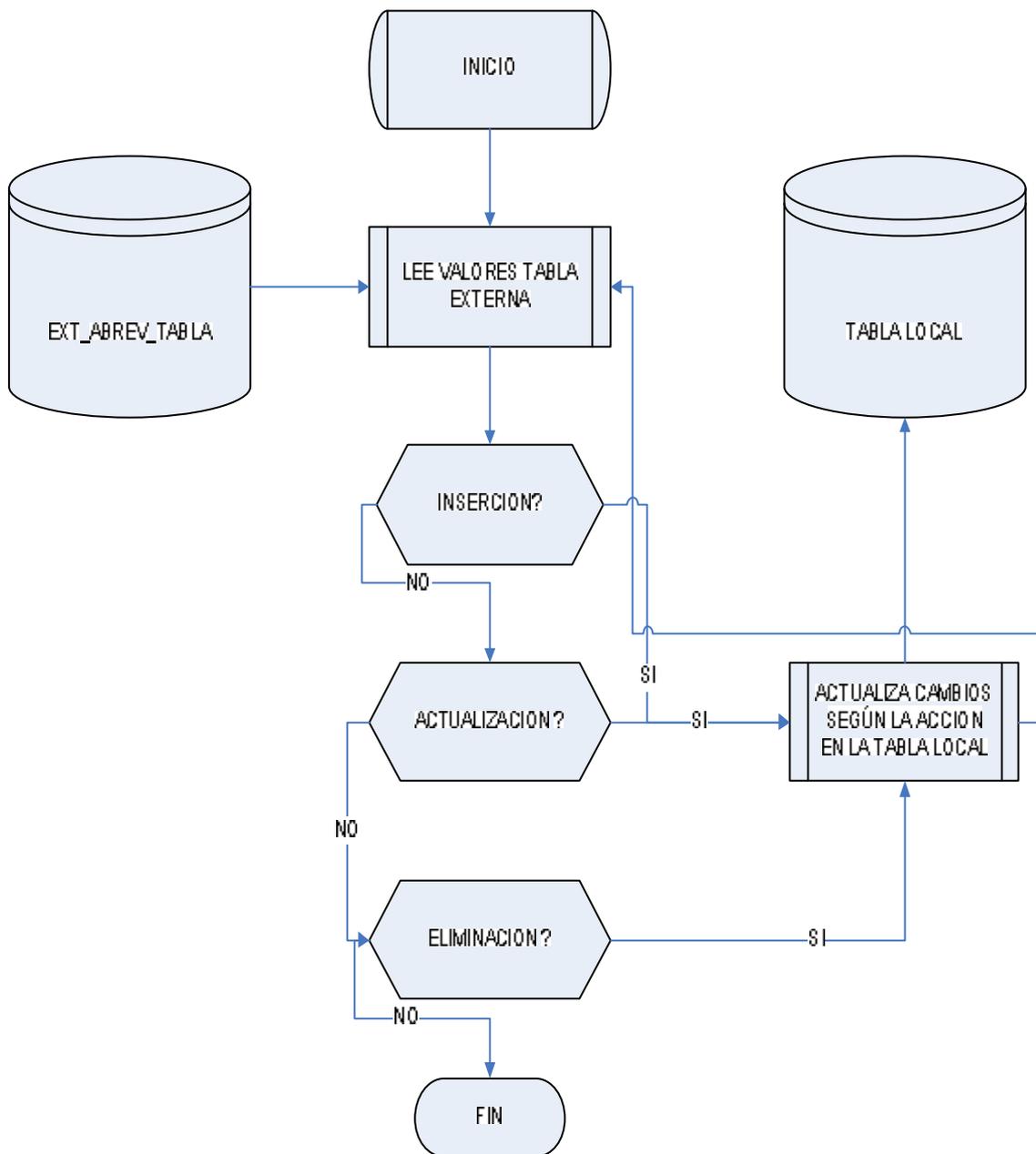


5.4 Procedimientos creados Dinámicamente

5.4.1 SIS_C_nombre_tabla

Descripción: Lee los datos desde la tabla externa y realiza las operaciones DML de inserción, modificación y eliminación en la tabla interna correspondiente. Son creados desde el procedimiento SIS_CREA_PROCS_CARGA.

Diagrama:

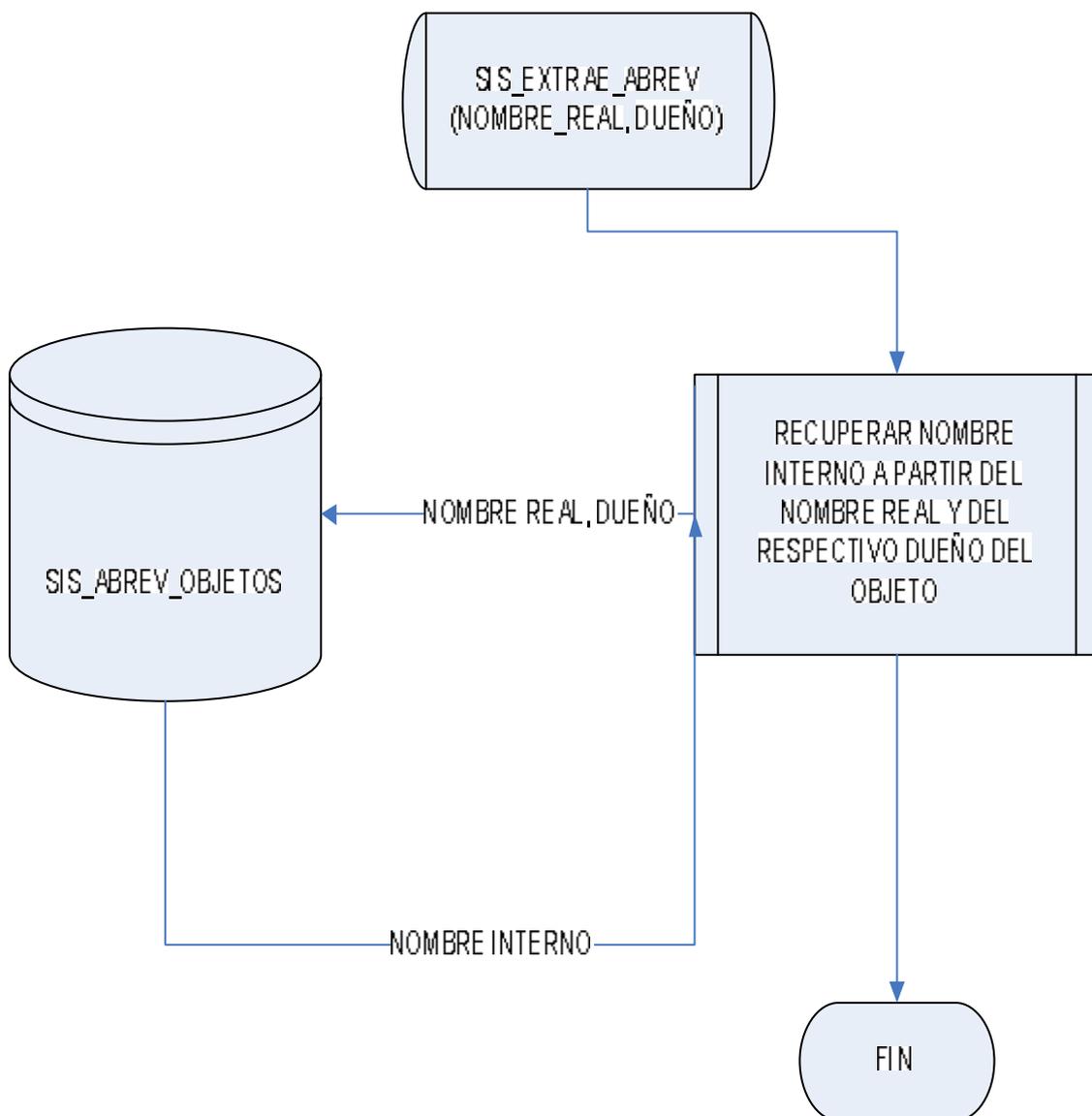


5.5 .- Funciones

5.5.1 SIS_EXTRAE_ABREV

Descripción: Es la encargada de recibir como parámetro el nombre real de la tabla y nos retornará el correspondiente nombre interno de la misma mediante la información guardada en la tabla SIS_ABREV_OBJETOS.

Diagrama:

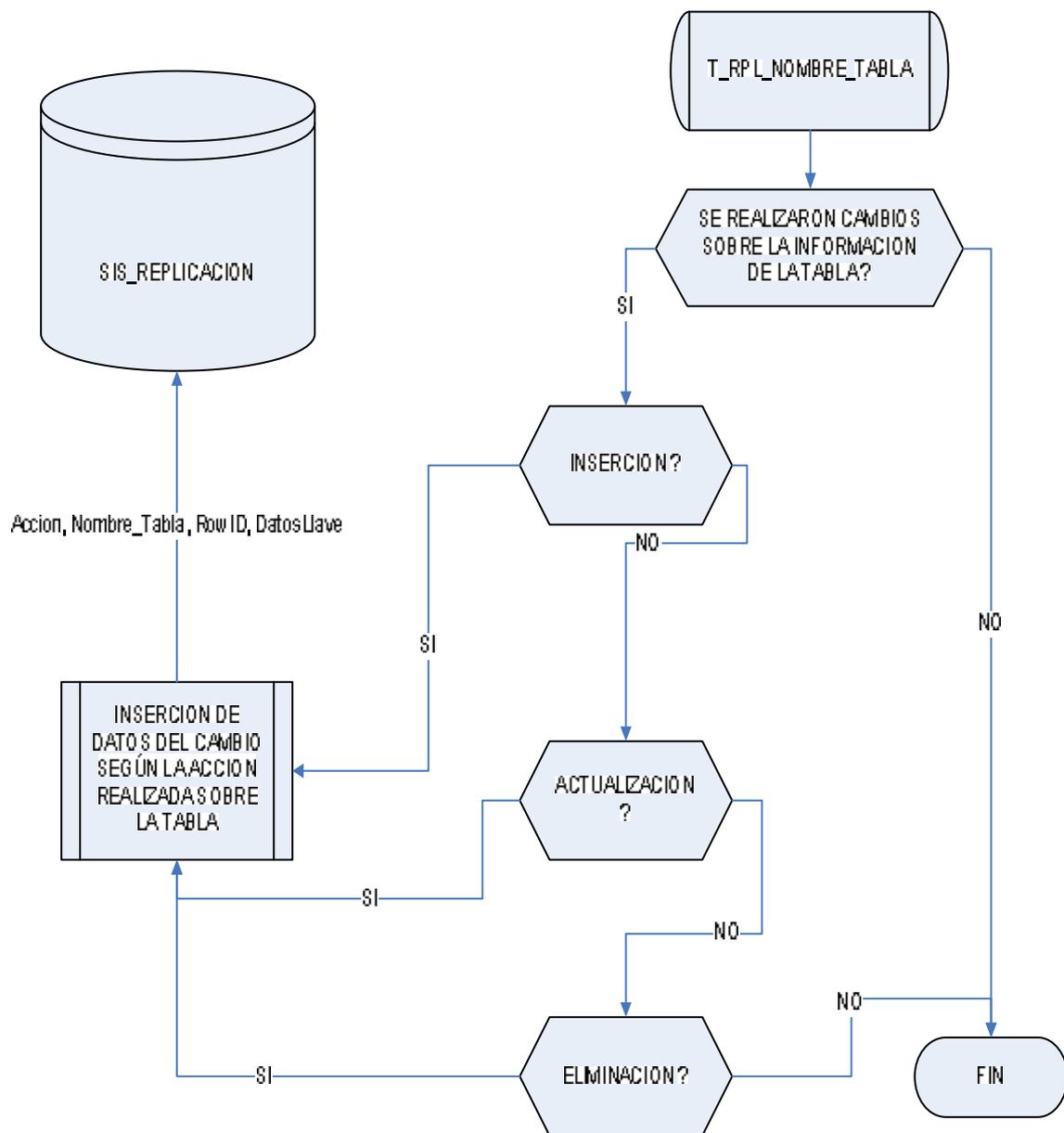


5.6 .- Triggers

5.6.1 T_RPL_nombre_tabla

Descripción: Es un trigger generado desde el procedimiento SIS_CREA_TRIGGER y su trabajo es monitorear cualquier cambio realizado sobre la tabla en la cual esta creado y guardar la información de dicho cambio en la tabla SIS_REPLICACION.

Diagrama:

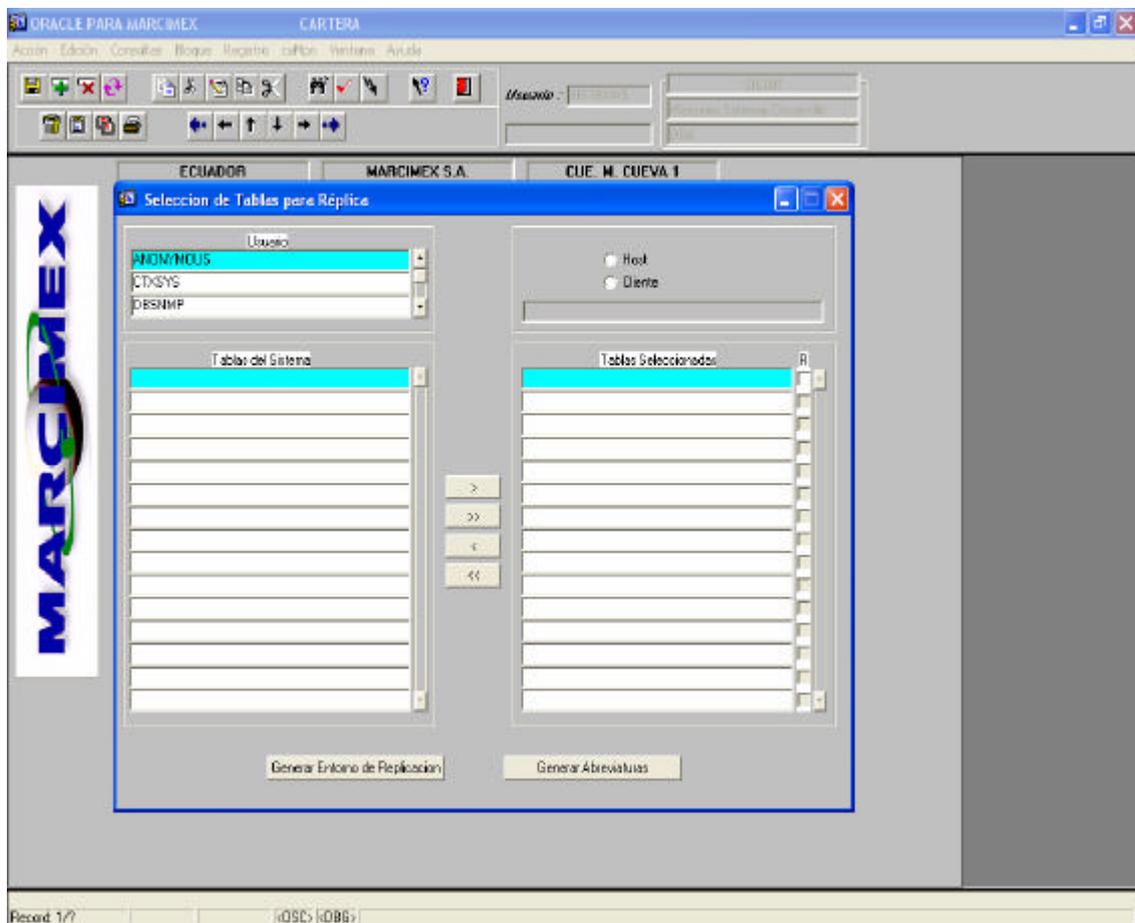


5.7.- Formas

5.7.1 SIS001

Descripción: Es la forma encargada de crear el ambiente de replicación tanto en el sitio de replicación como en el servidor de matriz, en esta forma crearemos el grupo de replicación que esta conformado por todas aquellas tablas que queremos que se repliquen, también es la forma encargada de generar los nombres internos (abreviaturas) de las tablas que conforman el grupo de replicación. Desde esta forma podremos también administrar la eliminación lógica de los objetos dinámicos en el momento en que se decida que cierta tabla ya no forme parte del grupo de replicación.

Pantalla :



Tablas:

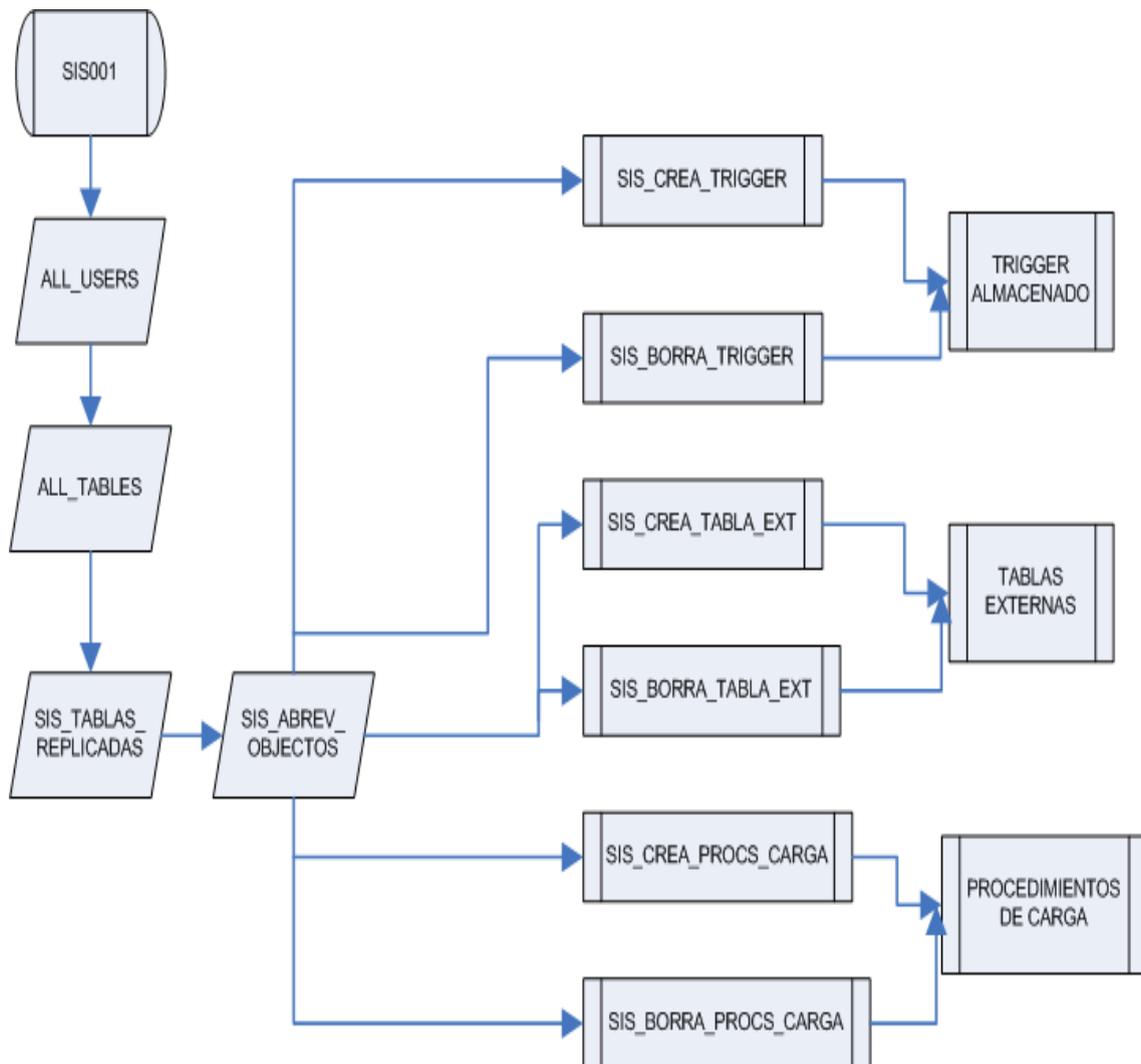
Nombre	Descripción	Acción
SYS.ALL_USERS	Tabla del diccionario de datos que contiene todos los usuarios de la	SELECT

	base de datos.	
SYS.ALL_TABLES	Tabla que contiene todos las tablas de la base de datos por usuario.	SELECT
SIS_TABLAS_REPLICADAS	Tabla del sistema de replicación que guarda el nombre de las tablas que queremos replicar	SELECT INSERT DELETE
SIS_ABREV_OBJETOS	Tabla que guarda la relación del nombre de tabla original con una nueva en abreviatura.	SELECT INSERT

Procedimientos Almacenados:

Nombre	Descripción	Parámetros
SIS_CREA_TRIGGER	Procedimiento que crea los trigger almacenados para cada tabla seleccionada a replicarse.	(nombre_tabla, error)
SIS_BORRA_TRIGGER	Procedimiento que borra los trigger almacenados par cada tabla que es retirada del sistema de replicación.	(nombre_tabla, error)
SIS_CREA_TABLA_EXT	Procedimiento que crea las tablas externas para cada tabla seleccionada a ser consolidada.	(nombre_tabla, error)
SIS_CREA_PROCS_CARGA	Crea los procedimientos almacenados para la lectura y carga de datos de cada tabla seleccionada para la consolidación de información.	(nombre_tabla, error)
SIS_BORRA_TABLA_EXT	Elimina de la base de datos las tablas externas de cada tabla quitada del sistema de consolidación de información.	(nombre_tabla, error)
SIS_BORRA_PROCS_CARGA	Elimina los procedimientos lectura y carga de datos de la base de datos, correspondiente a cada tabla que es eliminada del sistema de consolidación.	(nombre_tabla, error)

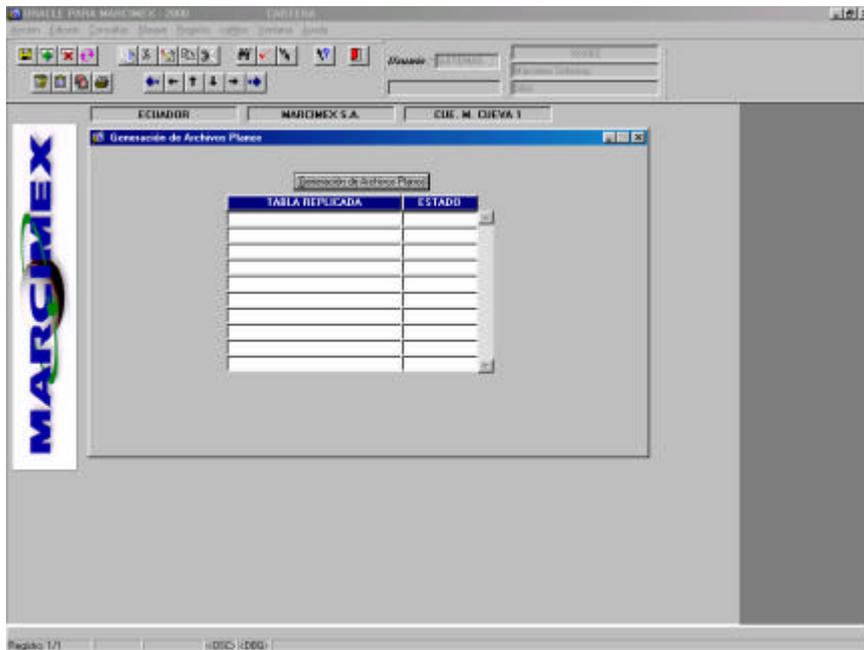
Diagrama:



5.7.2 SIS002

Descripción: Es la forma que genera los archivos planos de replicación que viajarán desde el sitio de replicación a el maestro para su respectiva carga, esta forma toma información de la tabla SIS_REPLICACION y genera un archivo plano con todos los registros modificados durante un espacio de tiempo y que no han sido enviadas aun al servidor de matriz, mas alguna información de seguridad como es la secuencia de envío.

Pantalla :



Tablas:

Nombre	Descripción	Acción
SIS_TABLAS_REPLICADAS	Tabla del sistema de replicación que guarda el nombre de las tablas que queremos replicar	SELECT
SIS_REPLICACION	Tabla del sistema de replicación que guarda todas las novedades de las tablas seleccionadas para la replicación desde las agencias.	SELECT UPDATE

Procedimientos Almacenados:

Nombre	Descripción	Parámetros
SIS_GENERA_PLANO_TABLAS	Procedimiento que crea los archivos planos con la información a replicar de cada tabla.	(nombre_tabla, rowid_tabla,Lv_accion, Ln_secuencia,Lv_datos, Lv_Cod_Error)

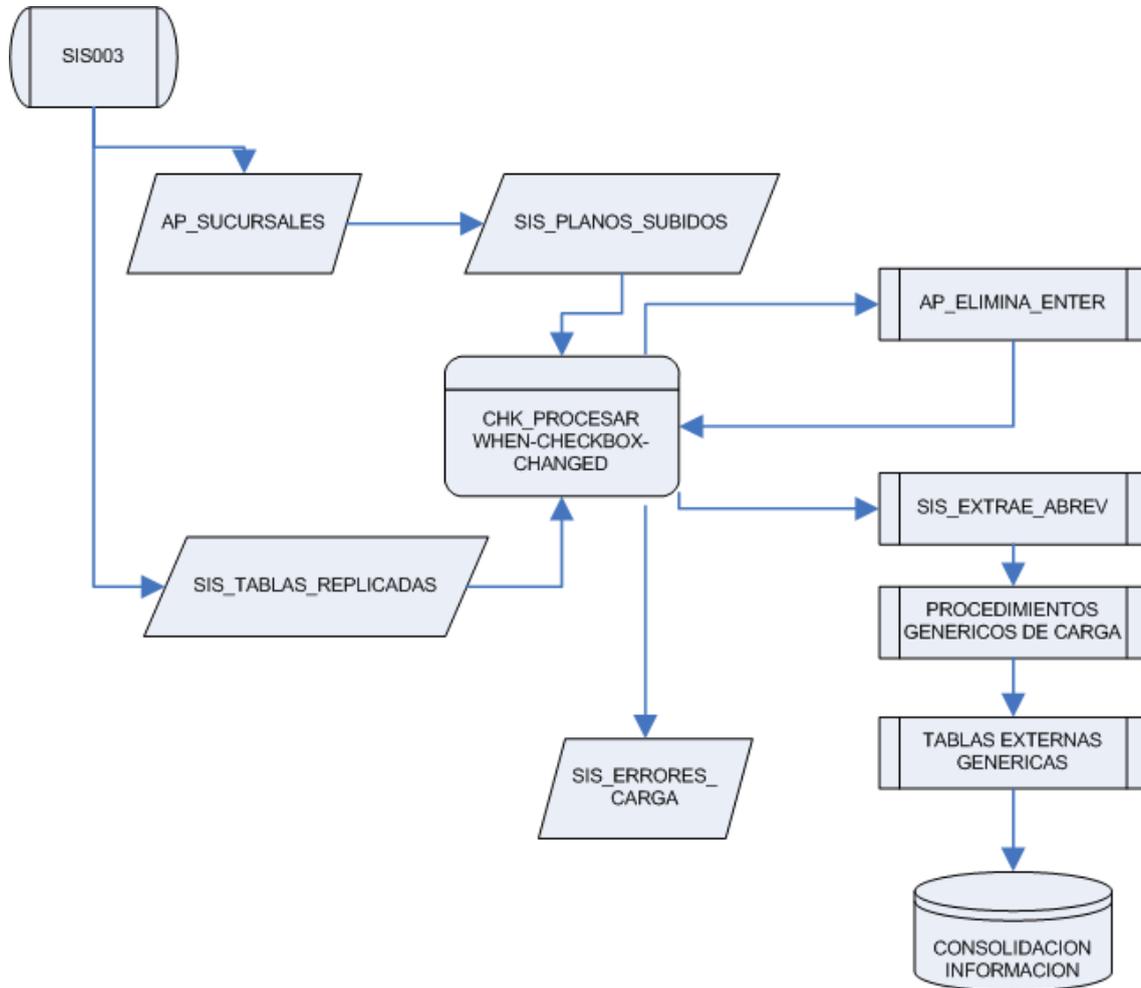
Tablas:

Nombre	Descripción	Acción
AP_SUCURSALES	Tabla del sistema de Marcimex, que guarda los nombres de todas las sucursales de la empresa	SELECT
SIS_PLANOS_SUBIDOS	Tabla del sistema de replicación que guarda control de las cargas de información para cada sucursal, secuencia y tabla consolidada.	SELECT UPDATE INSERT
SIS_TABLAS_REPLICADAS	Tabla del sistema de replicación que guarda el nombre de las tablas que queremos replicar o consolidar.	SELECT
SIS_ERRORES_CARGA	Tabla que guarda los posibles errores que se presenten durante la carga de información.	DELETE INSERT

Procedimientos Almacenados:

Nombre	Descripción	Parámetros
AP_ELIMINA_ENTER	Procedimiento que ayuda a descartar los caracteres <<enter>> de las tramas de datos.	(v_linea_archivo_lectura)
SIS_EXTRAE_ABREV	Procedimiento que sirve para tomar el nombre de tabla abreviada de cada una de las tablas replicadas.	(nombre_tabla,user)

Diagrama:



5.8.- Scripts

5.8.1 Genera_esquema.sql

Descripción: Este script genera una información con la estructura de las tablas de la base de datos de matriz. Luego guardamos el resultado en un archivo llamado estructura_tablas.txt.

Genera_esquema.sql

```
declare
  cursor c_tablas
is
  select a.TABLE_NAME ||';'|| a.COLUMN_NAME ||';'|| a.DATA_TYPE ||';'|| a.DATA_LENGTH
||';'|| a.DATA_PRECISION ||';'|| a.DATA_SCALE ||';'|| a.COLUMN_ID ||';' linea
```

```

from sys.all_tab_columns a
where a.OWNER in ('SISTEMAS','FINANCIAL')
and substr(a.TABLE_NAME,1,1) between 'A' and 'Z'
order by a.TABLE_NAME;
begin
for r1 in c_tablas loop
dbms_output.put_line(r1.linea);
end loop;
end;

```

5.8.2 Comparar_estructuras.sql

Descripción: Este script nos ayudará a determinar la diferencia entre las estructuras de las tablas de la agencia versus la estructura de las tablas de la matriz mediante la comparación entre la tabla externa que contiene las estructuras de las tablas de la matriz contra el diccionario de datos de la agencia

Comparar_estructuras

```

select a.nombre_tabla from datos_matriz a
where not exists (select 'X'
from sys.all_tab_columns b
where b.TABLE_NAME=ltrim(rtrim(a.nombre_tabla))
and b.COLUMN_NAME=ltrim(rtrim(a.nombre_columna))
and b.DATA_TYPE=ltrim(rtrim(a.tipo_de_datos))
and b.DATA_LENGTH=a.longitud
and nvl(b.DATA_PRECISION,0)=nvl(a.precision_dat,0)
and nvl(b.DATA_SCALE,0)=nvl(a.escala,0)
and nvl(b.COLUMN_ID,0)=nvl(a.id_columna,0)
and b.OWNER=user )
/

```

CAPITULO 6

CONCLUSIONES Y RECOMENDACIONES

En este capítulo presentaremos las debidas conclusiones y recomendaciones determinadas de acuerdo a una serie de experiencias presentadas en el desarrollo de este trabajo.

6.1 Conclusiones.

Luego de haber concluida la aplicación de Replicación Asíncrona basada en el uso de Tablas Externas y SQL Dinámico hemos sacado las siguientes conclusiones del mismo:

- El uso de Tablas Externas para realizar el intercambio de datos entre diferentes bases de datos geográficamente dispersas de pequeña y mediana carga transaccional, constituye una poderosa y económica opción debido a que convierte al archivo plano externo en una tabla virtual de Oracle en la que se puede realizar todo tipo de consultas utilizando código SQL común.
- Si bien las Tablas Externas son una solución sencilla para el manejo de información contenida en archivos planos, el precio de tiempo que debemos pagar por esa sencillez es sumamente costoso si lo comparamos con la carga de datos en masa de un utilitario como SQL Loader.
- Las Tablas Externas incorporan de una u otra manera a la estructura física de la base de datos central de la empresa a los archivos planos que la alimentan; es decir los archivos planos que alimentan a la base de datos desde entidades externas ya dejan de ser entes ajenos dispersos a veces sin ninguna estructura lógica que los ate al sistema al que pertenecen.
- El uso de cursores sobre las tablas externas facilita el manejo de la información contenida en los mismos y nos ayuda a filtrar sus datos como si se tratase de una verdadera tabla común de Oracle.
- El manejo de SQL Dinámico nos permite moldear en tiempo de ejecución nuestra aplicación, haciendo que este se acople a la estructura de datos que en ese preciso instante de tiempo este presente en la base de datos.
- Al estar este sistema de replicación sujeto a la estructura de las tablas que componen el diccionario de datos de la base, se concluye que siempre se deben tener las estructuras de las agencias y del servidor principal correctamente uniformes tanto en tamaño de campos, longitud y tipo de dato de cada campo y característica de llave primaria.

- Las etapas para realizar una replica asíncrona se pueden resumir en:
 - Moldeado del entorno de replicación en la Sucursal.
 - Selección de Tablas que enviarán sus cambios a la Matriz.
 - Generación de Mecanismos de Captura y almacenamiento de Cambios.
 - Generación de archivos planos que encapsulan la información que ha cambiado entre el instante de tiempo de la última replicación efectiva y el estado actual.
 - Envío de dichos archivos a la Matriz.
 - Moldeado del entorno de replicación en la Matriz.
 - Generar las estructuras de tablas externas que empataran con la estructura de los planos de replicación que lleguen de las Sucursales.
 - Generar los procedimientos de carga que dinámicamente generan las sentencias y cursores que suben la información contenida en las tablas externas hacia las tabas maestras de la Matriz.
 - Almacenar los posibles errores que pueda generar los procedimientos de carga.
 - Corregir dichos errores antes de que carguemos el siguiente paquete de datos.

6.2 Recomendaciones.

Como recomendaciones finales ya sean tanto para garantizar el correcto funcionamiento de la replicación como para mejorar el funcionamiento de la misma en un futuro, presentamos las siguientes:

- Verificar los permisos tanto de sistema operativo como de objetos de la base de datos en ambos fases de la replicación antes de comenzar con el primer envío de datos. Debido a que el SQL Dinámico no nos avisa del error al momento de la compilación sino cuando este realmente se dé en tiempo real.

- Verificar la igualdad de estructuras entre las distintas bases de datos remotas antes de cargar la información en la matriz, esto es fundamental para evitar errores de incompatibilidades de datos.
- Debido a la cantidad de transacciones que una empresa pudiera tener en su base de datos, se recomienda realizar el proceso de envío de información en periodos cortos de tiempo, como base en la empresa Marcimex S.A. podríamos decir que el envío diario es mas que suficiente para que los archivos enviados no sean demasiadamente grandes ya que así como se exportan los datos, de igual manera se exportan tiempos que están íntimamente relacionados con los tamaños de las cápsulas de replicación.
- No se recomienda utilizar el método de carga de datos con tablas externas con archivos planos que sobrepasen las diez mil líneas de código debido al tiempo de subida de los mismos, si este es el caso se podría diseñar un sistema de replicación basado en SQL Loader para que cargue la información en tablas temporales antes de subirlas definitivamente al sistema maestro de la Matriz.
- Una mejora fundamental que se debería realizar sobre el sistema sería colocar un procedimiento que respete la integridad referencial de las tablas durante el proceso de carga. Debido a que la estructura de tablas del sistema de gestión de la empresa Marcimex S.A. no lleva la integridad referencial en la base de datos sino en la aplicación, no fue necesario colocar dicha funcionalidad en el sistema de replicación base del presente trabajo.
- Tal vez si los errores tuvieran una pequeña documentación incluida en la aplicación para ayudar al administrador del sistema a solucionarlo sería una mejora muy buena para el desempeño de la carga de datos.
- Colocar un tipo de verificación como es el CRC o la PARIDAD en la copia de información en el sistema de replica nos ayudaría a estar seguros de que la totalidad de información que viene es la que se carga mas aún cuando el medio en el cual viaja la información es susceptible de daños en su estructura física que podrían comprometer a la información contenida en el. En Marcimex S.A. existirá una persona que se encargue específicamente del tema replicación por lo que dicho proceso se lo hará manualmente.
- Para el correcto funcionamiento de este tipo de replicación, es recomendable arrancar con una importación y exportación inicial de cada una de las bases de datos de las agencias de Marcimex S.A hacia su servidor principal para garantizar que arrancamos con una igualdad de la información. También es recomendable que cada cierto tiempo (sugerencia cada año) se proceda con esta operación.

BIBLIOGRAFIA

- URMAN Scout, Programación Avanzada con PL/SQL, McGraw Hill, Madrid, 2001.
- Oracle 9i, Manual de Documentación, Manual Oracle On-Line cod: B1043.

Internet:

- <http://www.dbasupport.com/oracle/ora9i/ors.shtml> : Página de Fundamentos de Replicación Avanzada de Oracle.
- www.oracle.com : Pagina principal de Oracle.
- <http://www.experts-exchange.com/Databases/Oracle/> : Pagina de consultas a expertos referentes a temas especificos de oracle.
- <http://www.orape.net/article31.html> : Fundamentos y ejemplos del uso de tablas externas sobre Oracle 9i.

ANEXOS