



UNIVERSIDAD DEL AZUAY

FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN

ESCUELA DE INGENIERÍA DE SISTEMAS

***“SISTEMA DE PEDIDOS EN AMBIENTE WEB
PARA LA GESTIÓN DE LA FUERZA DE VENTAS”***

Monografía previa a la obtención del título de

Analista en Informática

Autor:

VERONICA DEL ROSARIO AÑAZCO CENTENO

Director de Monografía:

ING. LENIN ERAZO GARZÓN

CUENCA, ECUADOR - AÑO 2011

Dedicatoria

*A mi hijos, fuente de mi inspiración, progreso y
obtención de cada uno de los objetivos trazados.*

Agradecimientos

A Dios por que ha estado conmigo a cada paso que doy cuidándome y dándome fortaleza para continuar, a mis padres, que fueron las personas que me apoyaron a lo largo de mi carrera estudiantil, profesional y personal, a mi esposo, quien siempre estuvo fomentando el desarrollo de esta monografía y, en especial, al Ing. Lenin Erazo por la dirección de éste trabajo.

Índice de Contenidos

Capítulo I Introducción.....	1
1.1 Definiciones y Conceptos.....	2
1.2 Tareas realizadas por un sistema de Pedidos en Ambiente..	
Web.	4
1.3 Seguridades.....	5
1.4 Arquitectura	5
Capítulo II Análisis y Diseño del Software.....	7
Introducción.....	7
2.1 Identificación y Definición del Problema.....	7
2.2 Diagnóstico de la Situación Actual.....	7
2.3 Optimización de la situación actual.....	9
2.4 Diseño de procesos	10
2.5 Diseño de la Base de Datos.....	25
Capítulo III Codificación del Software Desarrollo	34
Introducción.....	34
3.1 Antecedentes	34
3.2 Aplicativo Web	34
3.3 Integrador entre el Aplicativo Web y el ERP.....	35
3.4 Desarrollo de Vistas para extracción de información.....	37
3.5 Facturación desde Pedidos y Notas de Entrega	37
Resumen.....	37
Capítulo IV Instalación y Pruebas.....	38
Introducción.....	38
4.1 Características del equipo donde se aloja la página.....	38
4.2 Instalación del Software en que se realizará el módulo	38
4.3 Construcción de la cadena de Conexión.....	39

4.4 Pruebas.....	39
Resumen.....	41
Capítulo V Conclusiones y Recomendaciones.....	42
Glosario.....	44
Bibliografía.....	45
Anexos.....	46
Anexo A.....	47

Gráficos:

Gráfico 1.4. Arquitectura 3 Capas.....	6
Gráfico 2.4.1.1 Autenticación de Usuario.....	11
Gráfico 2.4.1.2.1 Creación Clientes.....	12
Gráfico 2.4.1.2.2 Modificación Clientes.....	13
Gráfico 2.4.1.3 Confirmación Datos Clientes.....	15
Gráfico 2.4.1.4.1 Creación Pedidos.....	17
Gráfico 2.4.1.4.2 Modificación Pedidos.....	18
Gráfico 2.4.1.4.3 Consulta Pedidos.....	19
Gráfico 2.4.1.4.4 Cerrar Pedidos.....	20
Gráfico 2.4.1.5.1 Facturación desde Pedidos.....	22
Gráfico 2.4.1.5.2 Facturación desde Notas de Entrega.....	24
Gráfico 2.5. Entidad / Relación.....	25
Gráfico 4.4. Pruebas.....	40

Resumen

El presente trabajo tiene como objetivo la creación de un sistema informático de Pedidos en Ambiente Web, aplicado a la necesidad de optimizar el manejo de información en línea de una comercializadora de consumo masivo. Inicia con una introducción, proporcionando conceptos, metodologías y teorías de sistemas informáticos. A continuación, se definen las necesidades y objetivos de la comercializadora y se elabora un diagnóstico de la situación actual de la empresa, optimizando la misma y rediseñando los procesos a desarrollarse.

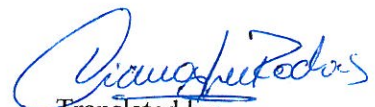
En el desarrollo del aplicativo web se ha utilizado Visual Basic .Net utilizando como base de datos Oracle. Se ha implementado un interfaz integrador entre el aplicativo Web y el ERP de la empresa a fin de automatizar el traslado de la información y mejorar la productividad de la misma.

ABSTRACT

The goal of the present work is to create a Web Environment Filter data processing system in order to optimize the management of the on-line information for a Massive Internet Enterprise. We begin with an introduction providing concepts, methodologies and Systems theories; modules, programs, procedures, etc; information about the enterprise, its goals and needs. An assessment of the company's current situation, its improvement, a design of the developing processes and a data dictionary containing the tables and the line of work are also presented.

The Web application, an integrated circuit of Web application and ERP (Enterprise resource planning), and receipt forms were developed. The operative systems, programs, data base, and other utility software employed during the configuration of the system are detailed in the installation process.




Translated by,

Diana Lee Rodas

CAPITULO I

Introducción

Los sistemas de información apuntan a lograr un aprovechamiento óptimo de los activos intangibles de la organización (por ejemplo, la información de las transacciones internas). Normalmente se implementan sistemas informáticos que le permiten asistir al personal en sus tareas operativas a partir de la automatización de procesos rutinarios ligados a controles internos. Por ejemplo: a partir de un proceso de facturación, se puede saber si dispone del stock necesario y actualiza el mismo y a la vez, genera los registros necesarios en el plan de cuentas para el área contable como de la cuenta corriente del cliente, pero, por que no utilizar esa información para poder clasificar la rentabilidad de sus productos, conocer la conducta de consumo de sus clientes y anticiparse a sus necesidades, generar tableros de control sobre las variables críticas y muchas otras alternativas de mejora en la gestión de la información? Precisamente esta es la más rica de las aplicaciones de la tecnología: no solo utilizarla como herramienta de gestión y control transaccional, sino ponerla al servicio de la dirección y mandos medios, ofreciéndoles información útil para la toma de decisiones.

Este proyecto busca aprovechar al máximo las posibilidades tecnológicas poniéndolas al servicio de la organización.

Actualmente las necesidades de disponibilidad de información se hace cada vez mas distribuida y los empleados requieren de esta en diferentes lugares geográficamente distribuidos o de amplio desplazamiento.

Una alternativa para solucionar esta necesidad es la incorporación de un sistema que permita la toma de pedidos a través de la web internet, y así integrarse al ERP de la empresa, para lo cual se diseñara y elaborara un sistema informático que analicé y diseñe la mejor alternativa.

Esto permitirá la reducción significativa de los tiempos de procesamiento de la información, mantener información en línea y no depender del re proceso de la información lo que genera un soporte adecuado para la toma de decisiones por parte de la Gerencia dentro de la empresa.

1.1 Definiciones y conceptos

Sistema:

Un sistema es un conjunto de partes o elementos organizadas y relacionadas que interactúan entre sí para lograr un objetivo. Los sistemas reciben (entrada) datos, y proveen (salida) información.

Un sistema puede ser abstracto o conceptual (un software).

Cada sistema existe dentro de otro más grande, por lo tanto un sistema puede estar formado por subsistemas y partes, y a la vez puede ser parte de un supersistema.

Un grupo de elementos no constituye un sistema si no hay una relación e interacción, que de la idea de un "todo".

Cualquier conjunto de dispositivos que colaboran en la realización de una tarea; se refiere también a cualquier colección o combinación de programas, procedimientos, datos y equipamiento utilizado en el procesamiento de información.

Grupo de elementos o conjunto de programas interdependientes que interactúan regularmente formando un todo.

Módulo:

Un módulo es un software que agrupa un conjunto de subprogramas y estructuras de datos. Los módulos son unidades que pueden ser compiladas por separado y los hace reusables y permite que múltiples programadores trabajen en diferentes módulos en forma simultánea, produciendo ahorro en los tiempos de desarrollo.

Los módulos promueven la modularidad y el encapsulamiento, pudiendo generar programas complejos de fácil comprensión.

Puede tomarse como sinónimo de subrutina o de unidad de software, aunque este último es más abarcativo.

Programa:

Es una secuencia de instrucciones que indican al hardware de un ordenador qué operaciones debe realizar con los datos, pueden estar incorporados al propio hardware, o bien pueden existir de manera independiente en forma de software.

Sinónimo de software, el conjunto de instrucciones que ejecuta un ordenador o computadora. El término puede referirse al código fuente original o a la versión ejecutable (en lenguaje máquina) de un componente de software. "

Conceptualizo un programa, como un conjunto de instrucciones secuenciales, lógicas y ejecutables, producto de un análisis efectuado para el logro de un fin común.

Procedimiento:

Un tipo de subrutina. Porción de código dentro de un programa más grande, que realiza una tarea específica y es relativamente independiente del resto del código. La mayoría de los lenguajes de programación incluyen soporte para la creación de procedimientos (u otros tipos de subrutinas, como funciones o módulos).

Los procedimientos suelen utilizarse para reducir la duplicación de códigos en un programa, permitir reusar los códigos, descomponer problemas complejos en piezas simples (mejorando la mantenibilidad y facilidad de extensión del código), mejora la lectura del código de un programa, oculta o regula parte de un programa, etc.

Los procedimientos son ejecutados cuando son llamados desde otros procedimientos, funciones o módulos. Los procedimientos pueden recibir parámetros, pero no necesitan devolver un valor como las funciones. Cada lenguaje de programación tiene su propia implementación de procedimientos (si es que posee procedimientos).

Dato:

Es un conjunto discreto, de factores objetivos sobre un hecho real o bien un registro de transacciones que por sí mismo tiene poca o ninguna relevancia o propósito.

Antecedente o fundamento que permite llegar más fácilmente al conocimiento de una cosa, el cual por sí mismo no tiene relevancia alguna.

Partiendo de lo anterior, defino a dato como un factor de un elemento el cual por sí mismo no tiene propósito o significado alguno.

A continuación se menciona la definición de información, puesto que al conformarse los datos como un conjunto integrado con características comunes dan origen a ella.

Información:

A diferencia de los datos, la información tiene significado (relevancia y propósito). No sólo puede formar potencialmente al que la recibe, sino que está organizada para algún propósito.

Los datos se convierten en información cuando su creador les añade significado; es decir; al conformarse como un conjunto integrado con características comunes. Una vez que se ha logrado dicha integración, se obtiene una secuencialidad en el proceso del tratamiento de la información dando origen a un programa, el cual es definido en seguida.

1.2 Tareas realizadas por un sistema que genere Pedidos a través de ambiente Web.

El objetivo de un sistema que genere pedidos a través de un ambiente Web es el de realizar el pedido, generar un comprobante de inventario de reserva del producto o sea una nota de entrega, facturar desde los pedidos y notas de entrega, controlar el estado de los pedidos, evaluar y operar con éxito el área comercial de la organización; a través de diferentes procesos y consultas que son generadas en línea por esta herramienta rompiendo los esquemas tradicionales de toma de información en papel por parte de la fuerza de ventas para luego ser digitalizada y reprocesadas en la organización.

Al tomar la información de los repositorios de datos de la empresa garantiza integridad y confiabilidad de la información al momento de realizar el proceso de ventas ante el cliente, del mismo modo nos entrega información en el instante en el que suceden los hechos al actualizar los mismos repositorios de manera inmediata.

El aplicativo de generación de pedidos debe ser lo suficientemente flexible para adaptarse a las reglas del negocio mediante políticas de ventas, políticas de crédito y políticas institucionales, de igual manera, debe ser dinámico ante los cambios permanentes que tienen las organizaciones en sus políticas y funcionamiento, tratando de que el aplicativo se adapte a los cambios de las políticas con facilidad y no las políticas cambien en función del aplicativo.

Las funciones de estos aplicativos son variadas y depende de la empresa, su nivel de personalización, pero básicamente, deben tener un modulo de seguridad y accesos básico en todo sistema informático, un modulo de preventa que engloba la creación de clientes, la toma de pedidos, y un modulo de informes en donde se deben plasmar la información suficiente y necesaria para cumplir con el objetivo de un sistema de Pedidos.

1.3 Seguridades

El requisito principal es la seguridad por esto un sistema de estas características debe exigir que todos los usuarios deben autenticarse como usuario verificado en los sistemas corporativos. Esto garantiza que el personal de ventas en el acceso a la información crítica sea un usuario autorizado y su nivel de autorización en la misma.

También una conexión no segura puede causar una pérdida a las empresas de datos propietarias, a su vez reduce la ventaja competitiva, de aquí que el sistema debe tener fuertes capacidades de encriptación para los datos de aplicación, ya que se transmite a través de una red usando un estándar de comunicación. Un sistema de cifrado de alta seguridad como Triple AES (Advanced Encryption Standard) o DES (Data Encryption Standard) es necesario para que los datos de la aplicación estén protegidos.

1.4 Arquitectura

La arquitectura de esta aplicación va en tres capas. Es un estilo de programación, su objetivo primordial es la separación de la capa de presentación, capa de negocio y la capa de datos.

Capa de Presentación.- Esta capa es la que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso; ésta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser “amigable”.

Capa de negocio.- Aquí es donde, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él.

Capa de Datos.- Es donde residen los datos y es la encargada de acceder a los mismos, está formada por uno o más gestores de base de datos que realizan todo el almacenamiento de los datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

En ésta aplicación, la capa de presentación se denomina Pedidos; contiene la definición de las características de la interface gráfica que utilizará el cliente a través del navegador. Esta capa a su vez se comunica con la capa de negocios, denominada PedidosCN, la cual contiene las validaciones de las reglas del negocio necesarias sobre los datos introducidos por el usuario. Finalmente está la capa de acceso a datos, denominada PedidosCD que se encarga de comunicarse con el repositorio de datos Oracle, y de enviar la información necesaria para las funciones y procedimientos residentes en la misma, y de recibir los datos procesados para devolverlos hacia la capa de negocios, la que a su vez se encargara de transmitirlos hacia la capa de presentación.

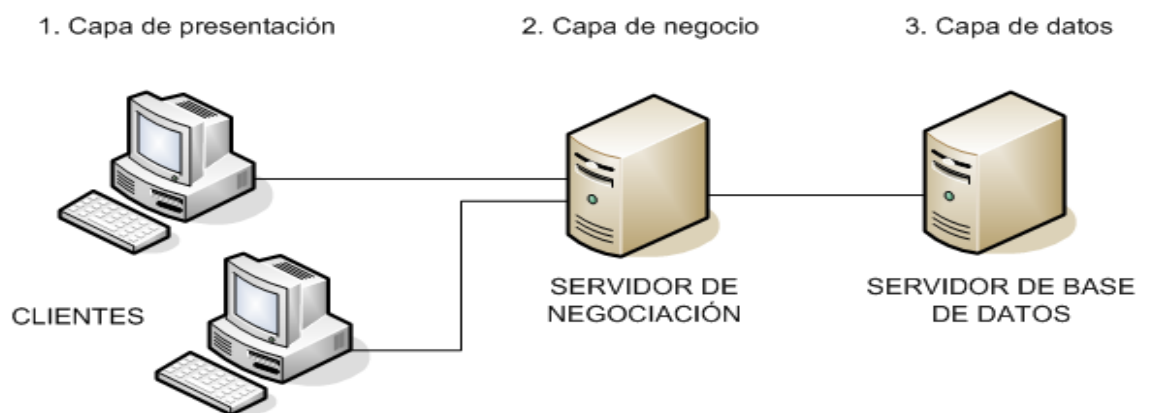


Gráfico 1.4 ARQUITECTURA EN 3 CAPAS

CAPITULO II

ANALISIS Y DISEÑO DEL SOFTWARE

Introducción

Este capítulo toma la base conceptual estudiada en el capítulo anterior y lo plasma a la práctica, desarrollando cada uno de los pasos para la elaboración del sistema que genera pedidos en ambiente web y aplicándolos a la realidad de una empresa.

2.1 Identificación y Definición del problema

La fuerza de ventas realiza la toma de pedidos y gestión de clientes de forma manual, luego al finalizar su jornada de trabajo se dirigen a las oficinas, liquidan su día de trabajo y realizan reportes manuales, entregan sus reportes y pedidos para su respectiva digitalización y cuadre lo que genera redundancia de trabajo, exceso de personal, pérdida de tiempo, falta de control, falta de información en línea para la toma de decisiones y la posibilidad de fraude por parte de la fuerza de ventas ya sea en el campo o dejando de hacer su trabajo al ser difícil su supervisión y control en el instante.

2.2 Diagnóstico de la Situación Actual

a) Descripción de la Organización y/o entorno Afectado por el Sistema

La empresa es una comercializadora de productos de consumo masivo, cuya principal gestión es la venta directa de sus productos visitando tienda por tienda a nivel nacional, para lograr este fin cuenta con 7 agencias a nivel nacional y una fuerza de ventas de más de 80 personas entre vendedores, entregadores, supervisores y jefes de ventas en su sistema de comercialización de pre-venta.

Adicional a la fuerza de ventas cuenta con una estructura administrativa y de bodegas para atender los requerimientos del personal de ventas y realizar la digitalización y control de lo realizado por la fuerza de ventas, cada agencia cuenta con personal de administración y bodegas.

La comercializadora cuenta con un ERP desarrollado en Oracle que genera información sobre sus operaciones y maneja toda la transaccionalidad de la empresa y sus distintos módulos (Contabilidad, Bancos, Cuentas x Cobrar, Cuentas x Pagar, Comercialización, Nominas, Inventarios), este ERP maneja la operatividad de la empresa una vez que los vendedores regresan de sus rutas diarias, momento en el cual existe una gran aglomeración de personal tratando de liquidar sus ventas del día y haciendo que el proceso sea lento y con mucha probabilidad de tener errores, de igual manera al no tener la información al día ya que el proceso de re-digitación toma un día no se puede tener una logística adecuada ni tomar decisiones de manera oportuna.

Objetivos actuales:

- Procesar la información más oportunamente: Tener al día la información y no llevar retraso en la facturación o entrega de producto más allá de dos días.
- Tener mejor control sobre la información: Existe mucha información errónea o al momento de re digitar no se entiende la letra del vendedor o están mal los códigos lo que genera pérdida de tiempo y muchas veces sobre las ventas.
- Optimizar recursos humanos: Existe mucho personal en la parte administrativa para la re digitación de la información y esta puede ser redistribuida en actividades de control y de supervisión para mejorar la productividad de la empresa.
- Tomar decisiones oportunamente: Una gran deficiencia para la compañía es la toma de decisiones oportuna por la falta de información, un objetivo claro a más del aumento de la productividad es la posibilidad de tomar decisiones en el instante en que se requiera y esto se logra únicamente con información oportuna, correcta y ágil y un adecuado control sobre el recurso humano.

b) Descripción del proceso de Preventa

Preventa: El proceso de pre-venta es el proceso en donde se registran los pedidos de los clientes. El proceso es manual, cada vendedor o pre-vendedor tiene una ruta asignada la cual se aplica para ese día con entrega en 48 horas, se le entrega el rutero impreso a cada vendedor con los clientes a visitar y la cartera de cada uno de éstos, los vendedores salen a realizar su ruta visitando cliente por cliente y tomando los pedidos a mano al igual que las novedades, al finalizar el día los vendedores regresan a la oficina para revisar los pedidos realizados, realizar informes de lo realizado en el día, notificar las novedades, hacer aprobar los pedidos a crédito y recibir alguna retroalimentación del trabajo realizado, la documentación es entregada a la liquidadora la cual procede a ingresar en el sistema, verificar códigos y precios y proceder a la facturación al día siguiente si no ha existido algún inconveniente y el siguiente paso es la entrega del producto con su respectiva factura.

2.3 Optimización de la situación actual

Antes de iniciar este proceso de mejora a través de una búsqueda de una solución en ambiente web para tener mejor control e información más oportuna la empresa realizó una serie de optimizaciones, tanto en la parte administrativa, organizacional así como tecnológica, las cuales se describen a continuación:

Se realizó un rediseño del proceso optimizando tiempos, se eliminaron tareas que generaban duplicidad del trabajo y se realizó una capacitación a todo el personal tanto de ventas como administrativo para mejorar el flujo de información.

Dentro de este rediseño del proceso también se diseñó el software de ingreso de información en ambiente web (ingreso de pedidos) con lo cual se optimizó el tiempo de ingreso y se mejoró la operatividad del mismo.

2.4 Diseño de procesos

El requerimiento general que busca la implementación de esta solución es básicamente un aplicativo que me permita tener mi información en línea y no esperar el reproceso de la misma para poder tomar decisiones o esperar hasta el final del día para actualizar las bases de datos.

Dotar a la fuerza de ventas de una herramienta que le permita tomar decisiones en el punto de venta, le permita transmitir información certera al cliente de una manera oportuna y clara y que le facilite su labor diaria optimizando tiempos tanto en el transcurso del día así como al finalizar el mismo.

Consideramos que una solución así es la que va a ayudar a la optimización tanto del área comercial como administrativa de la organización, reduciendo tiempos, teniendo el control sobre las acciones y tareas, obteniendo información en línea y una mejor forma de trabajo lo que puede generar en una oportuna toma de decisiones en cada área para ser más productivos.

El aplicativo va a tener un módulo de control de pedidos el cual debe ceñirse al siguiente proceso:

2.4.1 Módulo Realiza Pedidos a través de la Pagina Web

2.4.1.1 Inicio y Autenticación de Usuario

Si el usuario (cliente), se encuentra registrado en la Base de Datos Clientes y se encuentra autorizado para acceder al sistema, el aplicativo mostrará el nombre del usuario (cliente), caso contrario mostrará un mensaje de “Usuario No Autorizado”, el mismo que tendrá que registrarse.

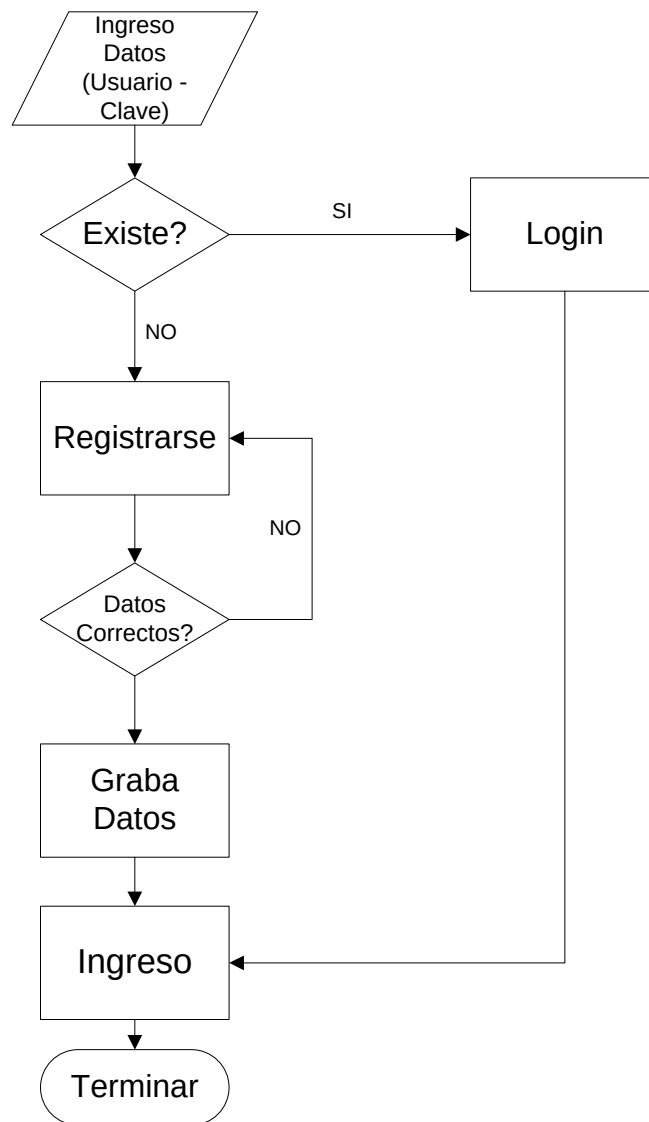


Gráfico 2.4.1.1 AUTENTIFICACION DE USUARIO

2.4.1.2 Opción Clientes

Nuevos Clientes: Permitirá la creación de nuevos clientes vía web, los datos a almacenar son: código cliente, cédula/ruc, nombre, dirección, teléfono, ciudad, categoría (por default se crea como consumidor final). El código de cliente el sistema otorga automáticamente. Los clientes también pueden ser creados desde el ERP y allí se les asigna mail y clave para entrar al aplicativo web.

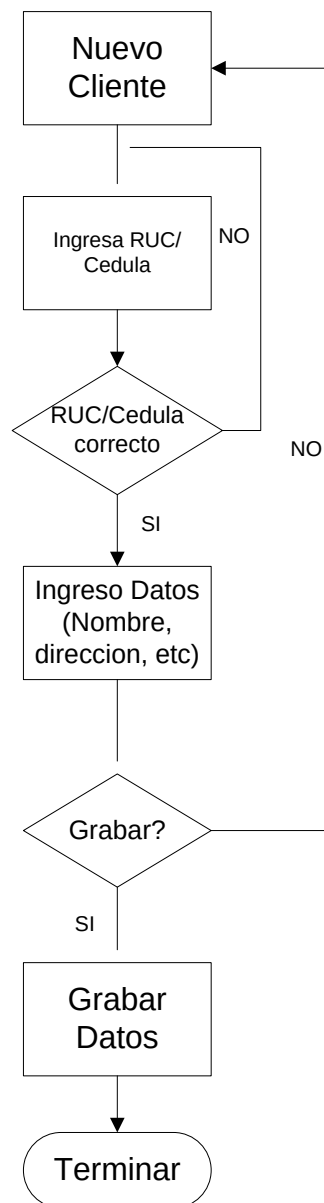


Gráfico 2.4.1.2.1 CREACION CLIENTES

Modificación Clientes: Permitirá la actualización de los datos del cliente, descritos anteriormente. Excepto Categoría, Lista de Precios, Política, Agente vendedor.

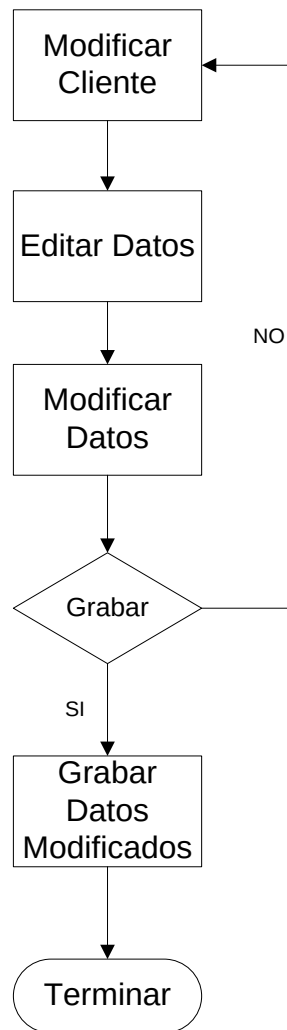


Gráfico 2.4.1.2.2 MODIFICACION CLIENTES

2.4.1.3 Confirmación Datos Clientes

Creado el cliente en el ERP a través de web y no tener los datos completos éstos se listarán en una de las opciones que tiene el supervisor para completar los datos necesarios para realizar un pedido, éste supervisor es el encargado de analizar al cliente (llamarlo por teléfono, visitarlo y llegar a identificarlo en una de nuestras categorías según las políticas de la empresa), los datos a completar son, que vendedor lo atiende, categoría, lista de precios, política, ruta, una vez con los datos completos envía una notificación al cliente para que puede realizar pedidos a través de la web. Completados los datos del cliente, éste deja de enlistarse en la opción que tiene el supervisor.

Los campos a completar:

Categoría.- Es una característica de los clientes, un grupo al que pertenecen por ejemplo: minorista, mayorista, distribuidores etc.

Lista de Precios.- Es la lista de precios con la que se le venderá al cliente, dependiendo de la categoría del cliente hay precios a minoristas, mayoristas, consumidor final, cada uno de éstos tienen precios diferentes.

Política.- Son parámetros de descuento, número de pagos, plazos, establecidos por la empresa para la venta y cobro de facturas.

Agente Vendedor.- Es un empleado o agente vendedor que va a estar pendiente del cliente, de sus pedidos, despachos, pagos, etc.

Ruta.- Es el sector en donde se encuentra la ubicación del cliente.

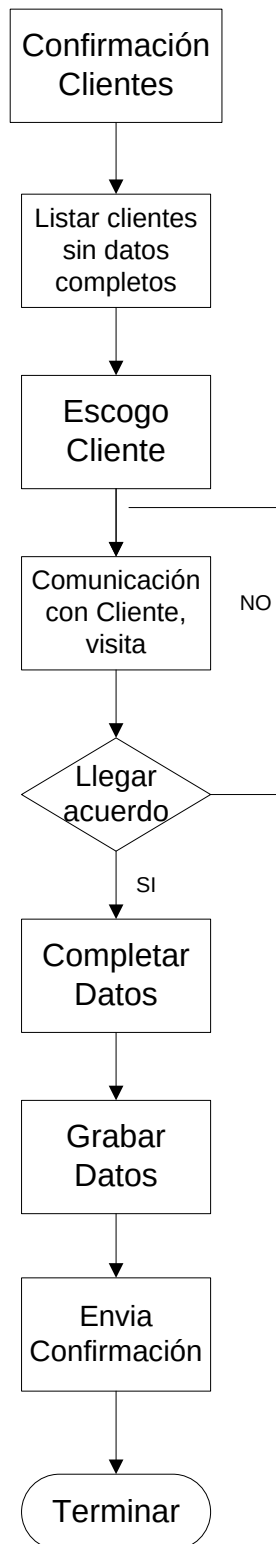


Gráfico 2.4.1.3 CONFIRMACION DATOS CLIENTE

2.4.1.4 Opción Pedidos

Ingresar Pedido: Permitirá la creación de un pedido y dependiendo del stock genera un comprobante de inventarios donde apartará o reservará el producto, cabe recalcar que después de la generación del comprobante de inventario se cambia el estado del pedido; a pedido con despacho parcial si es que solo se despacho una parte y si fue en su totalidad se actualiza el estado a finalizado, si no se realizó el comprobante de inventario o sea no se reservó el producto entonces el estado del pedido es de pendiente.

Las sub-opciones de Crear Pedido, son las siguientes:

Agregar Productos

Ingreso de los productos: código, cantidad la misma que está definida en unidades,

Buscar Producto-Stock: Búsqueda del producto a través del código y/o nombre del producto.

- Visualización en pantalla del resultado de la búsqueda: código, nombre, unidad de medida, precio.
- Visualización del stock del producto.
- Visualización del precio del producto.

Una vez ingresados los productos se realizarán los cálculos internos obteniendo el subtotal, descuentos si los hubiere, impuesto al valor agregado (IVA) y la totalidad del pedido.

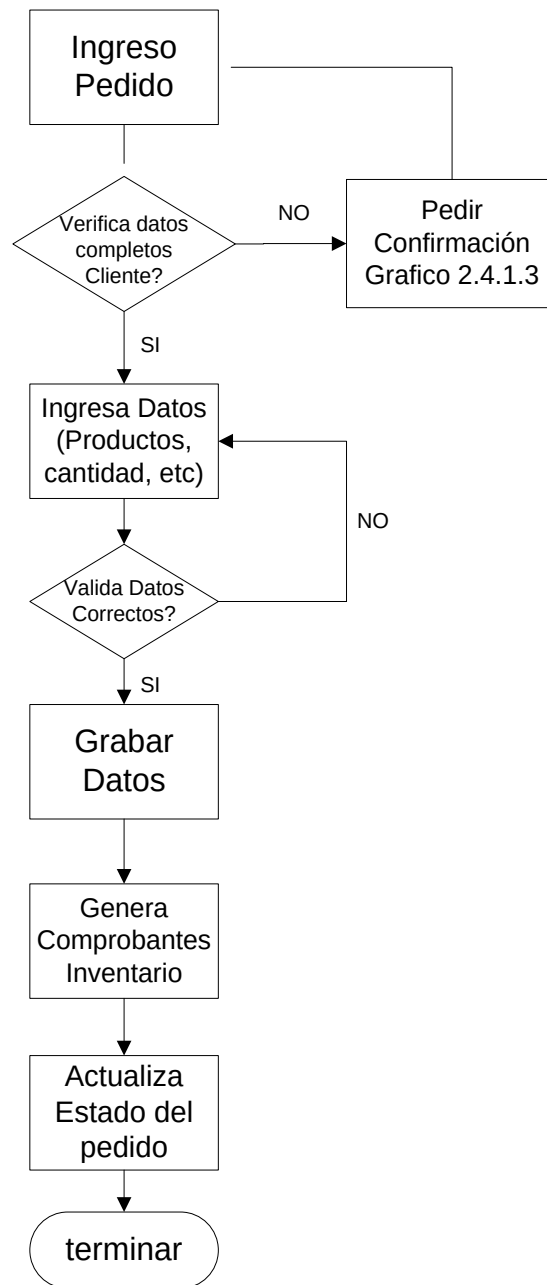


Gráfico 2.4.1.4.1 INGRESO PEDIDOS

Modificar Pedido: Permitirá la modificación de las cantidades, siempre y cuando el pedido aún no se lo haya despachado, cerrado o finalizado.

Eliminar Ítems: Permitirá la eliminación de un ítem que pertenece al pedido, siempre y cuando el pedido aún no se lo haya despachado, cerrado o finalizado.

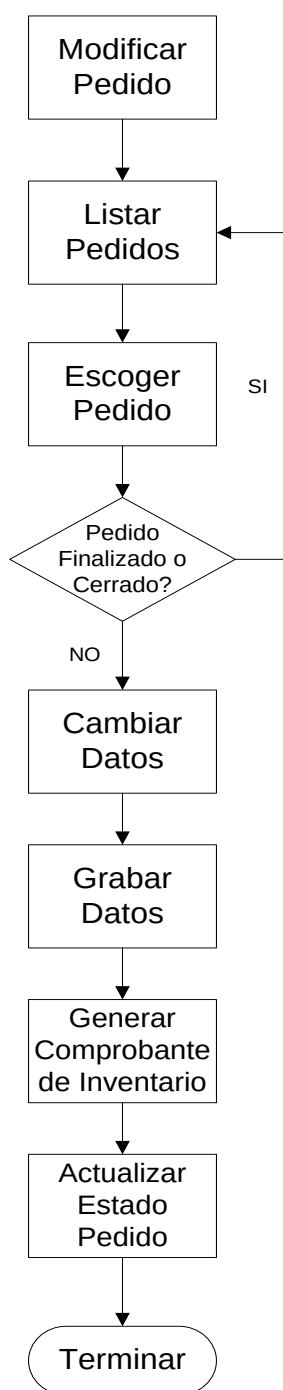


Gráfico 2.4.1.4.2 MODIFICAR PEDIDOS

Consultar Pedido: Este proceso permite consultar el pedido, podrá estar éste en estado pendiente, despacho parcial, cerrado o finalizado, se lo visualizará en pantalla, con todos sus datos, política, ítems, precios, etc.

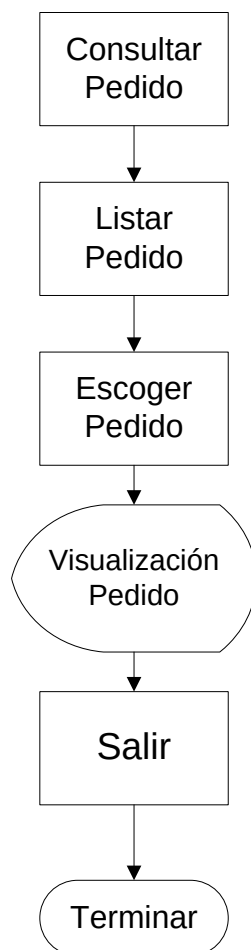


Gráfico 2.4.1.4.3 CONSULTAR PEDIDOS

Cerrar Pedido: Este proceso se efectuará únicamente si no se ha procedido con la facturación del pedido. El cliente puede cerrarlo en cualquier momento.

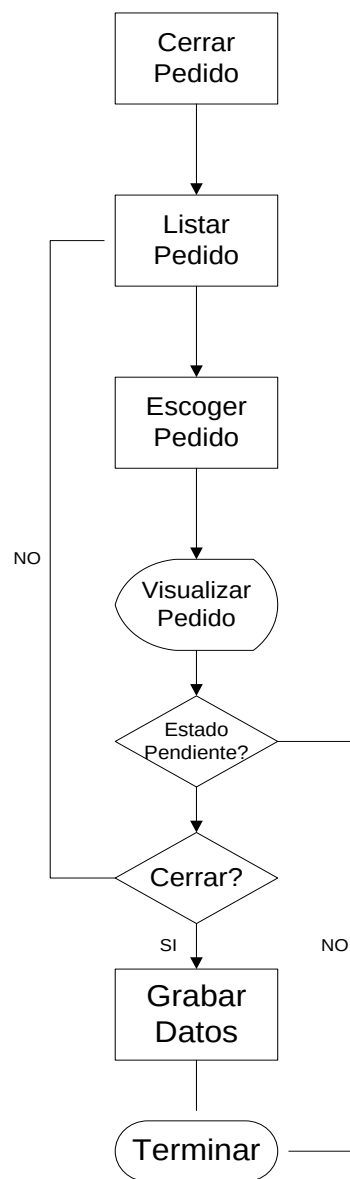


Gráfico 2.4.1.4.4 CERRAR PEDIDOS

Cuando el cliente haya ingresado todos los productos, cantidades que integran un pedido de un cliente y ya no existan modificaciones ni eliminaciones de ítems, el aplicativo grabará la información referente al pedido en la Base de Datos transaccional de la comercializadora.

2.4.1.5 Opción Facturación

La facturación se la realiza desde el ERP; tenemos 2 tipos:

Facturación desde Pedidos:

La primera es facturación desde pedidos pendientes o con despacho parcial, el pedido se crea desde la aplicación web y a medida que se vaya facturando el pedido quedará en diferentes estados, si al momento de facturar no tengo todos los ítems disponibles se facturará solo lo que tengo en stock y el resto quedará pendiente por lo tanto el pedido tomará un estado de despacho parcial, en cambio si se factura todo lo que está en el pedido, éste toma un estado de finalizado.

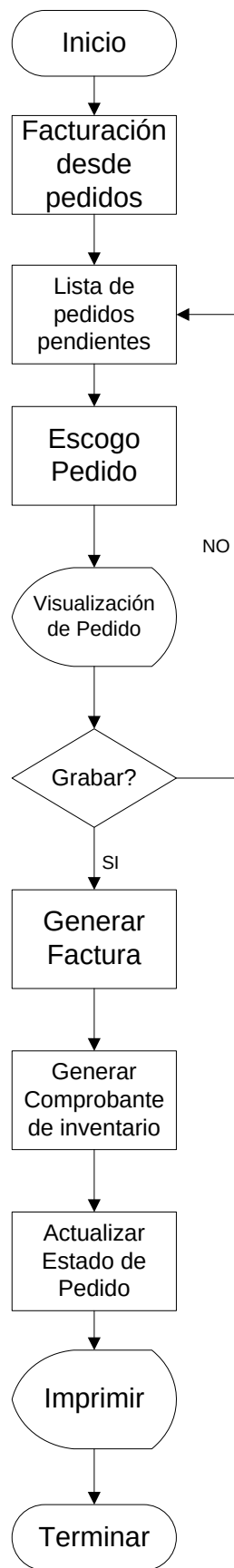


Gráfico 2.5.1.5.1 FACTURACION DESDE PEDIDOS

Facturación desde Notas de Entrega:

La segunda, facturación desde notas de entrega (comprobante de inventario), cuando realizo un pedido y si tengo stock en mi bodega automáticamente me reservará el producto generando un comprobante de inventario, es por esta razón que tenemos este tipo de facturación; ya que cuando se generó este comprobante se actualiza el estado del pedido ya sea a despacho parcial o finalizado dependiendo del stock disponible que haya tenido en ese momento.

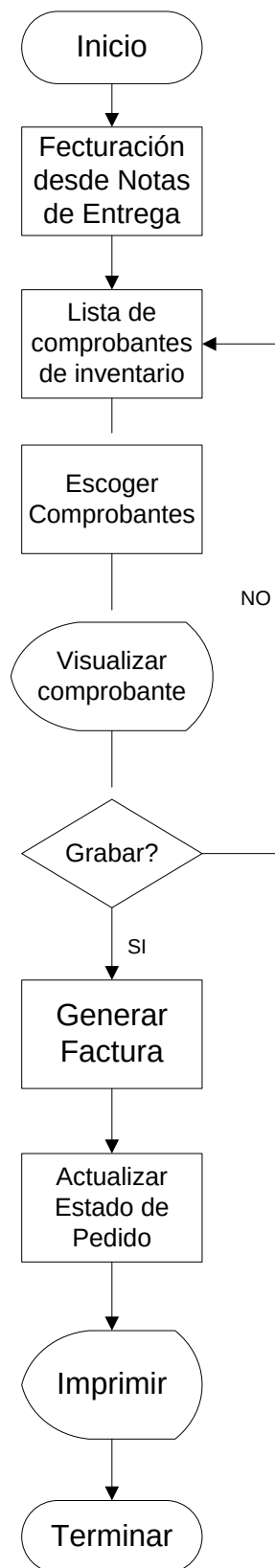


Gráfico 2.4.1.5.2 FACTURACION DESDE NOTAS DE ENTREGA

2.5 Diseño de base de Datos

- Diagrama Entidad-Relación.

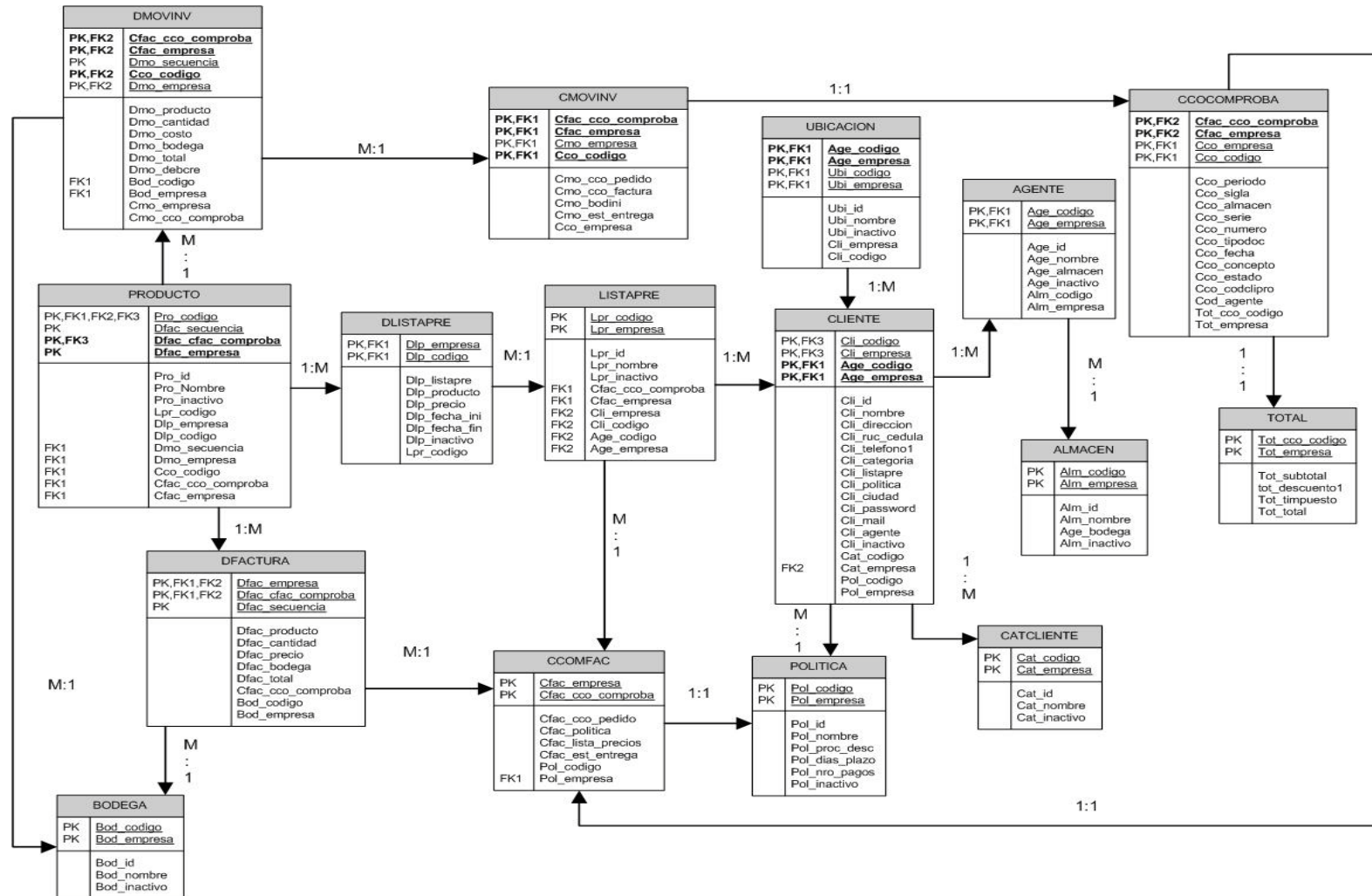


Gráfico 2.5. ENTIDAD / RELACION

- **Diccionario de Datos.**

Tabla Catcliente.- En esta tabla se guardan datos de las categorías de los clientes como son subdistribuidores, distribuidores, mayoristas, minoristas, consumidor final, etc; sus campos:

Cat_codigo.- Es un número secuencial para identificar al registro, código interno único.

Cat_empresa.- Este campo identifica a la empresa en la que estamos trabajando.

Cat_id.- Es un código para identificar al registro.

Cat_nombre.- Es el nombre que damos a la categoría.

Cat_inactivo.- En este campo identificamos si la categoría está activa o inactiva.

Tabla Ubicacion.- Esta tabla contiene datos de las provincias, ciudades que existen a nivel nacional.

Ubi_codigo.- Es un número secuencial para identificar al registro, código interno único de la tabla.

Ubi_empresa.- Este campo identifica a la empresa en la que estamos trabajando.

Ubi_id.- Es un código para identificar al registro.

Ubi_nombre.- Es el nombre de las provincias, ciudades.

Ubi_inactivo.- En este campo identificamos si la ubicación está activa o inactiva.

Tabla Bodega.- Esta tabla contiene datos de las bodegas que existen a nivel nacional.

Bod_codigo.- Es un número secuencial para identificar al registro, código interno único de la tabla.

Bod_empresa.- Este campo identifica a la empresa en la que estamos trabajando.

Bod_id.- Es un código para identificar al registro.

Bod_nombre.- Es el nombre de la bodega.

Bod_inactivo.- En este campo identificamos si la bodega está activa o inactiva.

Tabla Almacen.- Esta tabla contiene datos de los almacenes o agencias que existen a nivel nacional.

Alm_codigo.- Es un número secuencial para identificar al registro, código interno único de la tabla.

Alm_empresa.- Este campo identifica a la empresa en la que estamos trabajando.

Alm_id.- Es un código para identificar al registro.

Alm_nombre.- Es el nombre del almacén o agencia.

Age_bodega.- Este campo se escoge entre las bodegas creados en la tabla bodega.

Alm_inactivo.- En este campo identificamos si el almacén está activo o inactivo.

Tabla Agente.- Esta tabla contiene datos de los vendedores que atienden a los clientes.

Age_codigo.- Es un número secuencial para identificar al registro, código interno único.

Age_empresa.- Este campo identifica a la empresa en la que estamos trabajando.

Age_id.- Es un código para identificar al registro.

Age_nombre.- Es el nombre del vendedor.

Age_almacen.- Este campo se escoge entre los almacenes creados en la tabla almacen.

Age_inactivo.- En este campo identificamos si vendedor está activo o inactivo.

Tabla Politica.- En ésta tabla se guardan datos de las políticas de cobro de la empresa.

Pol_codigo.- Es un número secuencial para identificar al registro, código interno único.

Pol_empresa.- Este campo identifica a la empresa en la que estamos trabajando.

Pol_id.- Es un código para identificar al registro.

Pol_nombre.- Es el nombre que damos a la política.

Pol_porc_desc.- Porcentaje de descuento al cliente.

Pol_dias_plazo.- Días de plazo para cancelar la deuda adquirida.

Pol_nro_pagos.- El número de pagos hasta cancelar la factura.

Pol_inactivo.- En este campo identificamos si la política está activa o inactiva.

Tabla Listapre.- Esta tabla contiene datos de las listas de precios que existen en la empresa.

Lpr_codigo.- Es un número secuencial para identificar al registro, código interno único.

Lpr_empresa.- Este campo identifica a la empresa en la que estamos trabajando.

Lpr_id.- Es un código para identificar al registro.

Lpr_nombre.- Es el nombre que damos a la lista de precios.

Lpr_inactivo.- En este campo identificamos si la lista de precios está activa o inactiva.

Tabla Cliente.- En esta tabla se guardan los datos del cliente.

Cli_codigo.- Es un número secuencial para identificar al registro, código interno único.

Cli_empresa.- Este campo identifica a la empresa en la que estamos trabajando.

Cli_id.- Es un código para identificar al registro.

Cli_nombre.- Es el nombre del Cliente.

Cli_Dirección.- Se ingresa la dirección del cliente.

Cli_ruc_cedula.- Se ingresa el ruc o la cédula del cliente.

Cli_telefono1.- Se ingresa el número de teléfono.

Cli_categoria.- Este campo se escoge entre las categorías creadas en la tabla catcliente.

Cli_listapre.- Este campo se escoge entre las listas de precios creadas en la tabla listapre.

Cli_politica.- Este campo se escoge entre las políticas creadas en la tabla politica.

Cli_ciudad.- Este campo se escoge entre las ciudades creadas en la tabla ubicacion.

Cli_agente.- Este campo se escoge entre los vendedores creados en la tabla agente.

Cli_mail.- Es este campo se guarda el mail del cliente.

Cli_password.- En este campo se guarda la clave con la que se registra.

Cli_inactivo.- En este campo identificamos si el cliente está activo o inactivo.

Tabla Producto.- Esta tabla contiene información del ítem que se comercializa.

Pro_codigo.- Es un número secuencial para identificar al registro, código interno único.

Pro_empresa.- Este campo identifica a la empresa en la que estamos trabajando.

Pro_id.- Es un código para identificar al registro.

Pro_nombre.- Es el nombre que damos a la lista de precios.

Pro_inactivo.- En este campo identificamos si la lista de precios está activa o inactiva.

Tabla Dlistapre.- En ésta tabla se guardan datos de precios de cada uno de los ítems comercializados con su respectiva lista de precios.

Dlp_empresa.- Este campo identifica a la empresa en la que estamos trabajando.

Dlp_listapre.- Este campo se escoge entre las listas de precios creadas en la tabla listapre.

Dlp_codigo.- Es un número secuencial para identificar al registro, código interno único.

Dlp_producto.- Este campo se escoge entre las listas de precios creadas en la tabla listapre.

Dlp_precio.- Es el precio sin iva del producto.

Dlp_fecha_ini.- La fecha desde que inicia el producto con ese precio

Dlp_fecha_fin.- La fecha hasta la que es válido ese precio para el producto.

Dlp_inactivo.- En este campo identificamos si el precio está activo o inactivo.

Tabla Ccomproba.- En ésta tabla se guardan los datos de los comprobantes creados, como son pedidos, notas de entrega, facturas, esta es la cabecera de la parte contable de la empresa, donde distinguimos al comprobante por tipo de documento o sea si es pedido, nota de entrega, factura etc.

Cco_empresa.- Este campo identifica a la empresa en la que estamos trabajando.

Cco_codigo.- Es un número secuencial para identificar al registro, código interno único.

Cco_periodo.- El año en el que se realiza la transacción.

Cco_sigla.- La sigla del comprobante por ejemplo para pedidos PED, para facturas FA, para notas de entrega NET.

Cco_almacen.- La agencia a la que pertenece la transacción, es el código del almacén que fue extraído de la tabla de almacén.

Cco_serie.- La serie o sea el punto de emisión del pedido, factura.

Cco_numero.- El número secuencial del pedido dependiendo del almacén y la serie.

Cco_tipodoc.- El tipo de documento que es; por ejemplo: Pedido, Factura.

Cco_fecha.- Fecha del comprobante.

Cco_concepto.- Un concepto, observación que se haga del documento.

Cco_estado.- El estado es un valor número en el que se identifica si el comprobante está grabado en su totalidad, eliminado o en proceso.

Cco_codclipro.- Es el código del cliente que fue extraído de la tabla de cliente.

Cco_agente.- Es el código del agente vendedor que fue extraído de la tabla de agente.

Tabla Ccomfac.- Esta tabla contiene información de la cabecera de los pedidos y facturas realizadas, forma parte del anexo del modulo de ventas.

Cfac_empresa.- Este campo identifica a la empresa en la que estamos trabajando.

Cfac_cco_comproba.- Es el código del comprobante de la tabla ccomproba, un número secuencia único.

Cfac_cco_pedido.- En el caso de que se estuviera almacenando una factura en este campo se guarda el pedido del cual se originó la factura, en cambio si es un pedido, este campo va como null.

Cfac_politica.- Es el código de la política que fue extraído de la tabla de política.

Cfac_lista_precios.- Es el código de la lista de precios que fue extraído de la tabla de lista de precios.

Cfac_est_entrega.- Es un campo numérico en el que se guardan el estado del pedido para diferenciar si está pendiente, si está como despacho parcial o si está finalizado.

Tabla Dfactura.- Esta tabla contiene información del detalle de los pedidos y facturas, forma parte del anexo del módulo de ventas.

Dfac_empresa.- Este campo identifica a la empresa en la que estamos trabajando.

Dfac_cfac_comproba.- Es el campo cfac_cco_comproba de la tabla ccomfac, un número secuencia único, que a su vez fue extraído de la tabla ccomproba (cco_codigo).

Dfac_secuencia.- Un número secuencial para identificar cuántas líneas tiene el detalle del comprobante.

Dfac_producto.- Es el código del producto que fue extraído de la tabla de productos.

Dfac_cantidad.- La cantidad pedida o facturada.

Dfac_precio.- El precio del producto dependiendo de la lista de precios.

Dfac_bodega.- Es el código de la bodega que fue extraída de la tabla de bodega.

Dfac_total.- El total de lo pedido o facturado la cantidad por el precio.

Tabla Cmovinv.- Esta tabla contiene información de la cabecera de la nota de entrega o sea el comprobante de inventario de la factura, forma parte del anexo del módulo de inventario.

Cmo_empresa.- Este campo identifica a la empresa en la que estamos trabajando.

Cmo_cco_comproba.- Es el código del comprobante de la tabla ccomproba, un número secuencia único.

Cmo_cco_pedido.- En este campo va el código del pedido (cco-codigo) de la tabla ccomproba.

Cmo_cco_factura.- En este campo va el código de la factura (cco-codigo) de la tabla ccomproba.

Cmo_bodini.- Es el código de la bodega que fue extraída de la tabla de bodega.

Cmo_est_entrega.- Este es un campo numérico en el cual identifico si la nota de entrega ya fue facturada para que no se enliste en la opción facturación desde notas de entrega.

Tabla Dmovinv.- Esta tabla contiene información del detalle de la nota de entrega o sea el comprobante de inventario de la factura, forma parte del anexo del módulo de inventario.

Dmo_empresa.- Este campo identifica a la empresa en la que estamos trabajando.

Dmo_cmo_comproba.- Es el campo cmo_cco_comproba de la tabla cmovinv, un número secuencia único, que a su vez fue extraído de la tabla ccomproba (cco_codigo).

Dmo_secuencia.- Un número secuencial para identificar cuántas líneas tiene el detalle del comprobante.

Dmo_producto.- Es el código del producto que fue extraído de la tabla de productos.

Dmo_cantidad.- La cantidad egresada del stock del producto.

Dmo_costo.- El costo del producto.

Dmo_bodega.- Es el código de la bodega que fue extraída de la tabla de bodega.

Dmo_total.- El total de lo pedido o facturado la cantidad por el precio.

Dmo_debcre.- Identifica 1 para débito (ingreso), 2 para crédito (egreso).

Tabla Total.- Esta tabla contiene información de los datos finales del pedido y factura, como son los valores de subtotal, impuesto, total.

Tot_empresa.- Este campo identifica a la empresa en la que estamos trabajando.

Tot_cco_comproba.- Es el código del comprobante de la tabla ccomproba, un número secuencia único.

Tot_subtotal.- En éste campo va el subtotal del pedido o factura.

Tot_descuento1.- En este campo se guarda el valor de descuento que tiene el cliente dependiendo de su política.

Tot_timpuesto.- En este campo se guarda la información del impuesto iva.

Tot_total.- En este campo va el total de la factura el subtotal menos los descuentos y más el iva.

CAPITULO III

CODIFICACION DEL SOFTWARE

Introducción

Este capítulo explica las principales vistas y paquetes que sirvieron como integrador entre el aplicativo web y el ERP de la Comercializadora, además las formas desde las cuales se facturan dichos pedidos y notas de entrega, las pantallas del aplicativo; con el fin de entender el motor de esta integración y su relación, el detalle de todas las vistas, procedimientos, funciones, paquetes, formas y pantallas se describen en el Anexo A.

3.1 Antecedentes

El ERP de la comercializadora está basado en tecnología Oracle, desde su base de datos hasta sus interfaces, el aplicativo en ambiente web lee y graba directamente en la base de datos Oracle, por lo cual era necesario crear un integrador, el cual va a servir de vínculo para actualizar sus registros.

Para esto se construyó un paquete que contiene las funciones y procedimientos de actualización en la base de datos de la comercializadora y una serie de vistas para la extracción de la información de esta base de datos y compartirla con el aplicativo web.

3.2 Aplicativo Web.

El aplicativo web es una aplicación que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una Intranet mediante un navegador. En otras palabras es una aplicación software que se codifica en un lenguaje soportado por los navegadores web (en este caso por Visual Basic.Net), en la que se confía la ejecución al navegador.

Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para

actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales.

Es importante mencionar que una página Web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones, como por ejemplo acceder a gestores de base de datos de todo tipo.

3.3 Integrador entre el aplicativo Web y el ERP.

Para el desarrollo del paquete integrador, se realizó un estudio de las principales tablas del sistema ERP, las cuales tendrían afección en procedimientos de inserción, actualización o solamente consulta.

Una vez identificadas las tablas a las que tendría acceso el aplicativo en ambiente Web, se procedió a dar los permisos respectivos para que el sistema pueda realizar su tarea respectiva.

Con los permisos asignados, se desarrolló un paquete de base de datos llamado ast_sellermovil y ast_sellerweb, los cuales cuentan con todos los procesos y funciones que permiten la interacción entre el aplicativo WEB y el ERP. Dentro de este paquete están las inserciones, actualizaciones y eliminaciones que este aplicativo realiza en la base de datos Oracle.

Paquete ast_sellermovil.- Este paquete contiene los procedimientos de las inserciones a las tablas que se utilizaron para el desarrollo del aplicativo.

Función get_precio.- Esta función obtiene el precio del producto.

Procedimiento sp_movil_InsertCabeceraPedido.- Este proceso inserta la cabecera del pedido afectando a las tablas ccomproba y ccomfac.

Procedimiento sp_movil_InsertDetallePedido.- Este proceso inserta el detalle del pedido afectando a la tabla Dfactura.

Procedimiento sp_movil_InsertTotalPedido.- Este proceso inserta los valores totales del pedido como subtotal, iva, descuento, total, afectando a la tabla Total.

Procedimiento `sp_movil_InsertCliente`.- Este proceso inserta los datos del cliente afectando a la tabla `Cliente`.

Paquete `ast_sellerweb`.- Este paquete contiene los procedimientos de las modificaciones y consultas a las tablas que se utilizaron para el desarrollo del aplicativo.

Función `Sp_Web_Login`.- Esta función te extrae el código del cliente basado en el mail y en usuario que se ingresan en la parte de autenticación del usuario.

Función `Sp_Web_DatosCliente`.- Esta función extrae un registro con datos del cliente.

Función `sp_web_ListaPedidosClientes`.- Esta función te extrae un registro con datos de los pedidos de un cliente.

Procedimiento `Sp_Web_ModificaPedido`.- Este procedimiento permite actualizar los datos de un pedido modificado en el ERP.

Procedimiento `Sp_Web_ModificaCliente`.- Este proceso permite actualizar los datos del cliente en la tabla cliente cuando éstos son modificados.

Procedimiento `Sp_Web_EstadoPedidos`.- Este proceso actualiza el estado del pedido dependiendo de los procesos que hayan afectado al mismo.

Función `Sp_Web_GeneraNotaEntrega`.- Esta función devuelve el código del comprobante de inventarios generado en el momento de realizar el pedido.

Función `Sp_Web_ListarProductos`.- Esta función extrae un registro con datos del producto o ítem a facturarse.

Función `Sp_Web_CargarCabeceraPedido`.- Esta función extrae un registro con datos de la cabecera del pedido.

Función `Sp_Web_CargarDetallePedido`.- Esta función extrae un registro con datos del detalle del pedido.

Una vez desarrollado el paquete, este paso a las pruebas de integridad, se realizaron pruebas tanto a nivel de procedimientos como del aplicativo en conjunto para garantizar que la información sea confiable y que la integración sea horizontal entre todos los módulos del sistema.

3.4 Desarrollo de las vistas para extracción de información

De igual forma se desarrolló la extracción de la información de la base de datos Oracle hacia el aplicativo web, la cual se realizó básicamente a través de vistas; para el desarrollo de las vistas se procedió de igual manera que en el desarrollo del paquete integrador, es decir, se identificaron las tablas relacionadas y que no estuvieron en el proceso de desarrollo del paquete.

Se asignaron permisos de lectura a las tablas identificadas, procediendo al desarrollo de las vistas, en el desarrollo de estas vistas se tuvo especial cuidado en no generar muchos recursos del servidor, ya que al tener un número alto de usuarios accediendo a la información, este puede ser un cuello de botella.

3.5 Desarrollo de las formas para realizar la facturación

Se realizaron 5 formas 2 de las cuales listan los pedidos y notas de entrega pendientes para facturar, las otras 2 generan la facturación desde pedidos y desde notas de entrega y la última forma completa los datos del cliente para que se encuentre listo para realizar pedidos.

Resumen

En el desarrollo de un integrador se debe tener especial cuidado, ya que en el proceso podemos almacenar o entregar información errónea, incoherente o incompleta, de ahí que el proceso tiene que ser analizado, probado y depurado antes de que el software salga a producción.

CAPITULO IV

INSTALACION Y PRUEBAS

Introducción

Este capítulo explica la forma en la que se realiza la instalación del módulo de pedidos en ambiente Web en la comercializadora.

4.1 Características del equipo en que se alojará la página web

Las características que debe tener este equipo para el correcto funcionamiento de la página deben ser:

Procesador Core I3 2.4 Ghz

Memoria 3 Gb de RAM

Disco 250 Gb

Tarjeta de Red.

4.2 Instalación del Software en que se realizará la página web

Las herramientas que se utilizaron para el desarrollo fueron: Microsoft Visual Basic.net, Base de Datos Oracle (10.2.4), Forms y Reports 6i, Microsoft Office 2007 (Word, Excel, Visio, etc).

Se instalará el Sistema Operativo Windows 7

Se instalará Visual Studio V-2010

Instalación y configuración de la conexión con ORACLE:

Me baje el oracle data access components de la página web de oracle
http://www.oracle.com/technetwork/topics/dotnet/index_085163.html

Instrucciones de Instalación:

Descargar ODAC10202.exe o ODTwithODAC10202.exe en una carpeta temporal. Tenga en cuenta: No descargue este archivo en la "tmp".

Descomprimir su contenido mediante la ejecución del ejecutable.

Ejecute el Oracle Universal Installer (OUI) con el lanzamiento de el archivo setup.exe dentro de la "instalación" del directorio que se descomprimió.

OUI le llevará a través de la instalación en su máquina. Después de la instalación, es posible eliminar ODAC10202.exe y las carpetas y los archivos de instalación temporal.

Como nota informativa el FIREWALL que se está usando es el que viene por defecto en el sistema operativo Windows, cambiando ciertos parámetros para que nos deje acceder solo a nuestro sitio web y denegando el resto de conexiones.

4.3 Construcción de la cadena de conexión:

ODP.NET necesita la siguiente información para conectarse con la Base de Datos:

Dirección IP pública (a través del protocolo TCP/IP) del equipo que nos da el servicio de conexión con la base de datos
Puerto en el que escucha la Base de datos
Nombre de la Instancia de la Base de Datos
Usuario y Clave para la conexión

Usé esta información en la siguiente cadena de conexión.

```
Conexion="Data  
Source=(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)  
(HOST=200.25.212.146)(PORT=1521)))(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=Cristal));User  
Id=usuario;Password=pruebas;"
```

4.4 Pruebas.

Para realizar las pruebas para el aplicativo en web lo que se necesita es un computador con su sistema operativo y un navegador instalado con acceso a internet. Para facturar necesitamos tener acceso al ERP y el asistente encargado de la facturación procederá con la misma.

Prueba	Función	Procedimiento	Respuesta del Sistema	Aprobación
Registrarse	Registrarse	El cliente ingresa los datos de usuario y clave, si existe valida al cliente y pasa al siguiente Menú, caso contrario el cliente debe registrarse ingresando todos los datos requeridos.	El sistema valida datos y graba. Permite al usuario ingresar al menú principal.	
Nuevo Cliente	Confirmación Datos	Se listan los clientes nuevos en una opción del supervisor de ventas, este se comunica, los visita y llegan a algún acuerdo en precios, política etc. Con el acuerdo se completan datos del cliente.	Completan datos, graba y envían mail a cliente de confirmación de datos.	
	Modificación	Primero se realiza una consulta de los datos del usuario (cliente) y se elige el campo a modificar.	Habilita el campo y se procede a modificar, luego valida y graba datos, no se puede modificar campos como categoría, lista de precios, política, cedula.	
Realización del Pedido	Ingresar	Ingreso de datos, utilizando consultas para la búsqueda de productos	El sistema valida datos y graba; integra los datos ingresados a las tablas de la base de datos de oracle y a su vez genera un comprobante de inventario NET separando el producto que tenemos en stock en ese momento.	
	Modificar	Seleccionar el pedido que desea modificar	Solo puede modificar si el pedido está en estado de pendiente o sea no se ha despachado nada ni generado ninguna nota de entrega. Valida y graba las modificaciones.	
	Cerrar	Seleccionar el pedido que desea cerrar.	El sistema no permite eliminar pedidos que estén facturados o realizados una nota de entrega, valida y graba.	
	Consultar	Se elige criterio de búsqueda y se da click en buscar	Presenta datos que cumplan con criterio de búsqueda.	
Facturación	Facturar desde Pedidos	Se listan los pedidos pendientes de facturación, y se procede a facturar	Valida y graba datos.	
	Facturar desde Nota de Entrega	Se listan las notas de entrega pendientes a facturar y se procede a facturar	Valida y graba datos.	
Reportes		Se elige el reporte deseado.	El sistema presenta en pantalla y permite imprimir	

Gráfico 4.4. PRUEBAS

Resumen

El éxito o el fracaso de un sistema desarrollado, parte desde la cabeza del jefe de sistemas y el grado de compromiso y apoyo que este le dé al mismo, el compromiso y apoyo tiene que ir bajando estructuralmente, hacia sus actores, quienes son finalmente los que hacen grande al proyecto en su ejecución; o los que hacen que los proyectos fracasen, de ahí la importancia de una estrategia para conseguir que el sistema sea de todos los que componen la organización y no solamente de sus creadores o mentalizadores.

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones Generales.

Se realizó una breve descripción del tema específico y cuáles deben ser las consideraciones y las plataformas para proponer un sistema de pedidos en ambiente web dentro de una estructura comercial, así como, las funciones que esta herramienta debe tener como mínimo para que sea eficiente y productiva dentro de una organización, pudiendo ser tomado estas definiciones como punto de partida para elaborar dicho sistema.

Se ejecutó paso a paso los puntos necesarios para la elaboración del sistema, aplicándolos a la comercializadora, con la finalidad de conocer su problemática, su entorno, su situación actual y que esperan del nuevo sistema; como puede aportar a la organización un sistema de este tipo, cuáles son los requisitos que se tiene que cubrir y una estimación del trabajo a realizar.

Identificados los objetivos y limitaciones de la comercializadora y con un conocimiento profundo de la organización y las áreas afectadas, se procedió a realizar el desarrollo del sistema.

Se realizó el diseño de la aplicación, este diseño se lo realizó integrando a todo el personal involucrado, tanto a los niveles gerenciales, como a los niveles operativos; los primeros nos dan la idea clara de lo que quieren lograr en el aplicativo, es decir, los objetivos y su alcance, y, los segundos, nos brindaron el apoyo de cómo realizarlo y la mejor manera de hacerlo; para que este aplicativo sea funcional para ellos, en lugar de ser una carga de trabajo.

Como conclusión general se puede decir, que, un sistema informático tiene que seguir una estructura y una metodología correcta y realizar todos los pasos necesarios para que este sea parte de la organización e involucre a todos sus actores.

5.2 Recomendaciones.

Para el correcto uso de éste sistema se requiere tanto de parte del cliente como de la comercializadora, que el servicio de internet que se va a utilizar sea de condiciones estables; ya que un corte por cualquiera de las dos partes causaría pérdida de datos o duplicidad de los mismos.

Se deberá capacitar tanto al personal interno de la comercializadora como a los clientes de la misma para el correcto uso del aplicativo y evitarnos problemas posteriores.

Se recomienda entregar un manual del funcionamiento del aplicativo en web a los nuevos clientes (usuarios del mismo) el momento de realizar la visita y confirmar al cliente.

Con respecto a Visual Basic.Net recomiendo esta herramienta, por su facilidad de uso y aprendizaje, además es un lenguaje de cuarta generación lo que me ha permitido desarrollar la aplicación en 3 capas, aprovechando la orientación a objetos y la reutilización de código.

Glosario

TCP/IP: Protocolo de Control de Transmisión/Protocolo de Internet

Informática: Es la ciencia aplicada que abarca el estudio y aplicación del tratamiento automático de la información, utilizando sistemas computacionales, generalmente implementados como dispositivos electrónicos.

Internet: Es un conjunto descentralizado de redes de comunicación interconectadas que utilizan la familia de protocolos TCP/IP.

Intranet: Es una red de ordenadores privados que utiliza tecnología Internet para compartir dentro de una organización parte de sus sistemas de información y sistemas operacionales.

Web: World Wide Web (también conocida como «la Web»), el sistema de documentos (o páginas web) interconectados por enlaces de hipertexto, disponibles en Internet.

WI-FI: Wireless Fidelity

ERP: Planificación de recursos empresariales, Enterprise resource planning

Oracle: Sistema de gestión de base de datos relacional

ODP.NET: Componente de ORACLE para conexión a Base de Datos mediante Visual Studio.

Capa: Forma como una solución y es segmentada desde el punto de vista lógico.

Nivel: Forma en que las capas lógicas se encuentran distribuidas de forma física.

Firewall: Cortafuegos, es una parte de un sistema o una red que está diseñada para bloquear el acceso no autorizado, permitiendo al mismo tiempo comunicaciones autorizadas

Bibliografía

- EDUARDO CONTRERAS Y OTROS, Metodología de Informática de MIDEPLAN 1992
- PRESSMAN S, Roger: Ingeniería del Software, McGrawHill.
- LONEY, Kevin; THERIAULT, Marlene: Oracle9i Manual del Administrador, Primera Edición, McGraw-Hill, Madrid, 2002.
- Oracle Corporation, <http://metalink.oracle.com>.
- www.galeon.com/periodismo-digital/pagina_n.htm
- www.mef.gob.pe/dnpp/glosario.htm
- e-Book VB.NET 2010
[http://www.taringa.net/posts/ebooks-tutoriales/6988293/Manuales - Tutoriales-y-eBooks-Visual-Basic-2010.html](http://www.taringa.net/posts/ebooks-tutoriales/6988293/Manuales_-_Tutoriales-y-eBooks-Visual-Basic-2010.html) 22-12-2010
- Oracle Corporation, <http://opn.oracle.com>.
http://www.oracle.com/technology/global/lad-es/pub/articles/cook_dotnet.html 29-12-2010
- HALVORSON, Michael; "Microsoft Visual Basic 2010 Step by Step"; Impreso en Estados Unidos; Microsoft Press A Division of Microsoft Corporation; 2010.
- MACDONALD, Matthew; "Beginning ASP.NET 4 in VB 2010"; Impreso en Estados Unidos; Apres; 2010.
- TROELSEN Andrew; VRAT Vidya; "Pro VB 2010 and the .NET 4 Platform"; Impreso en Estados Unidos; Apress; 2010.
- VRAT Vidya, HUDDLESTON James; "Beginning VB 2008 Databases From Novice to Professional"; Impreso en Estados Unidos; Apres; 2010.
- ANALISIS DE SISTEMAS Y DISEÑO 6TA. EDICION
[KENDALL](#)
- ANALISIS DE SISTEMAS DISEÑO Y METODOS McGraw-Hill
Interamericana Editores S.A de C. V. (México D.F.)
Jeffrey Whitten

Anexo

Anexo A

“Introducción

La presente Política Informática ha sido elaborada por el departamento de sistemas de la Organización, y busca el establecimiento de las bases y acciones tendientes a:

- a) Optimizar los recursos humanos y materiales hacia la eficiencia de los sistemas informáticos.
- b) Mejorar la gestión de los procesos de información en la Organización de manera correcta, segura y eficiente.
- c) Fomentar el desarrollo de una estructura informática, compatible y convergente entre los diferentes departamentos y agencias de la Organización.
- d) Establecer normas específicas para el mejor desempeño de los Sistemas de Información y Comunicación.

a) Paquete

Procedimientos y Funciones para la integración de Web al ERP:

```
--=====
-- PROCEDIMIENTO PARA ALMACENAR EL PEDIDO (CABECERA)
-- LOS REGISTROS SE ALMACENARAN EN LAS TABLAS: CCOMPROBA Y
CCOMFAC
-- PARAMETROS QUE RECIBE:
-- REGISTROS PROVENIENTES DE LAS TABLAS:
--=====

PROCEDURE sp_movil_InsertCabeceraPedido( /******CAMPOS QUE SE
ALMACENARAN EN LA TABLA CCOMPROBA*/V_CCO_EMPRESA_SM
CCOMPROBA.CCO_EMPRESA%TYPE,
V_CCO_CODIGO_SM CCOMPROBA.CCO_CODIGO%TYPE,
V_CCO_PERIODO_SM CCOMPROBA.CCO_PERIODO%TYPE,
V_CCO_SIGLA_SM CCOMPROBA.CCO_SIGLA%TYPE,
V_CCO_ALMACEN_SM CCOMPROBA.CCO_ALMACEN%TYPE,
V_CCO_SERIE_SM CCOMPROBA.CCO_SERIE%TYPE,
--EL CAMPO NUMERO CCO_NUMERO_SM SE CALCULA

POSTERIORMENTE
--EL CAMPO CCO_DOCTRAN_SM SE CALCULA CUANDO SE GENERA
EL NEW ID DE CCOMPROBA
V_CCO_TIPODOC_SM CCOMPROBA.CCO_TIPODOC%TYPE,
--CCO_FECHA_SM FECHA DE LA BASE DE ORACLE
V_CCO_CONCEPTO_SM CCOMPROBA.CCO_CONCEPTO%TYPE,
V_CCO_MODULO_SM CCOMPROBA.CCO_MODULO%TYPE,
```

```

V_CCO_NOCONTABLE_SM
CCOMPROBA.CCO_NOCONTABLE%TYPE,
V_CCO_ESTADO_SM CCOMPROBA.CCO_ESTADO%TYPE,
V_CCO_DESCUADRE_SM CCOMPROBA.CCO_DESCUADRE%TYPE,
V_CCO_ADESTINO_SM CCOMPROBA.CCO_ADESTINO%TYPE,
V_CCO_PVENTA_SM CCOMPROBA.CCO_PVENTA%TYPE,
V_CCO_CENTRO_SM CCOMPROBA.CCO_CENTRO%TYPE,
V_CCO_TIPO_CAMBIO_SM CCOMPROBA.CCO_TIPO_CAMBIO%TYPE,
V_CCO_TCLIPRO_SM CCOMPROBA.CCO_TCLIPRO%TYPE,
V_CCO_CODCLIPRO_SM CCOMPROBA.CCO_CODCLIPRO%TYPE,
V_CCO_AGENTE_SM CCOMPROBA.CCO_AGENTE%TYPE,
V_CCO_TRANSACC_SM CCOMPROBA.CCO_TRANSACC%TYPE,
V_CCO_CIE_COMPROBA_SM
CCOMPROBA.CCO_CIE_COMPROBA%TYPE,
V_CCO_ANULADO_SM CCOMPROBA.CCO_ANULADO%TYPE,
V_CREA_USR_SM CCOMPROBA.CREA_USR%TYPE,
--CREA_FECHA SYSDATE
V_MOD_USR_SM CCOMPROBA.MOD_USR%TYPE,
--MOD_FECHA SYSDATE
V_CCO_BODEGA_SM CCOMPROBA.CCO_BODEGA%TYPE,
--CCO_FECHAFIN_SM SYSDATE
--CCO_DIA_SM DIA SYSDATE
--CCO_MES_SM MES SYSDATE
--CCO_ANIO_SM ANIO SYSDATE
V_CCO_RUTA_SM CCOMPROBA.CCO_RUTA%TYPE,

/*****CAMPOS QUE SE ALMACENARAN EN LA TABLA CCOMFAC*/

V_CFAC_AUTORIZA_SM CCOMFAC.CFAC_AUTORIZA%TYPE,
V_CFAC_POLITICA_SM CCOMFAC.CFAC_POLITICA%TYPE,
V_CFAC_LISTA_PRECIOS_SM
CCOMFAC.CFAC_LISTA_PRECIOS%TYPE,
V_CFAC_EST_ENTREGA_SM
CCOMFAC.CFAC_EST_ENTREGA%TYPE,
V_CFAC_PROC_FAC_SM CCOMFAC.CFAC_PROC_FAC%TYPE,
V_CFAC_PROCESO_SM CCOMFAC.CFAC_PROCESO%TYPE,
V_CFAC_NOMBRE_SM CCOMFAC.CFAC_NOMBRE%TYPE,
V_CFAC_DIRECCION_SM CCOMFAC.CFAC_DIRECCION%TYPE,
V_CFAC_TELEFONO_SM CCOMFAC.CFAC_TELEFONO%TYPE,
V_CFAC_CED_RUC_SM CCOMFAC.CFAC_CED_RUC%TYPE,
V_CFAC_CIUDAD_SM CCOMFAC.CFAC_CIUDAD%TYPE,
V_CFAC_TIPO_ACTPRO_SM CCOMFAC.CFAC_TIPO_ACTPRO%TYPE,
V_CFAC_SOL_COMPROBA_SM
CCOMFAC.CFAC_SOL_COMPROBA%TYPE,
V_CFAC_IMPUESTO_SM CCOMFAC.CFAC_IMPUESTO%TYPE,
V_CFAC_PORC_IMPUESTO_SM
CCOMFAC.CFAC_PORC_IMPUESTO%TYPE,
V_CFAC_TIPOPAGO_SM CCOMFAC.CFAC_TIPOPAGO%TYPE,
V_CFAC_COMISION_SM CCOMFAC.CFAC_COMISION%TYPE)

--DECLARACION DE VARIABLES
IS

V_NEW_NUMERO DTIPOCOM.DTI_NUMERO%TYPE;
V_ALM_ID ALMACEN.ALM_ID%TYPE; --CODIGO EXTERNO DEL ALMACEN
V_NEW_DOCTRAN CCOMPROBA.CCO_DOCTRAN%TYPE;
V_PVE_ID PUNTOVENTA.PVE_ID%TYPE;

BEGIN
--CALCULO EL NUEVO NUMERO DE LA TABLA DTIPOCOM
--SELECT DTI_NUMERO + 1 INTO V_NEW_NUMERO FROM DTIPOCOM
--WHERE DTI_CTI_CODIGO = V_CCO_SIGLA_SM
--AND DTI_EMPRESA = V_CCO_EMPRESA_SM
--AND DTI_ALMACEN = V_CCO_ALMACEN_SM
--AND DTI_SERIE = V_CCO_SERIE_SM;

```



```

--EDITO EL NUMERO DE LA TABLA DTIPOCOM

--UPDATE DTIPOCOM SET DTI_NUMERO = V_NEW_NUMERO
--WHERE DTI_CTI_CODIGO = V_CCO_SIGLA_SM
--AND DTI_EMPRESA = V_CCO_EMPRESA_SM
--AND DTI_ALMACEN = V_CCO_ALMACEN_SM
--AND DTI_SERIE = V_CCO_SERIE_SM;
--COMMIT;

V_NEW_NUMERO := ast_cnt.secuencia_ccomproba(nvl(V_CCO_EMPRESA_SM, 0),
                                             nvl(V_CCO_PERIODO_SM, 0),
                                             nvl(V_CCO_SIGLA_SM, 0),
                                             nvl(V_CCO_ALMACEN_SM, 0),
                                             nvl(V_CCO_SERIE_SM, 0));

--CALCULO EL CAMPO CCO_DOCTRAN

--OBTENGO EL ALM_ID (EJ.006)
SELECT ALM_ID
  INTO V_ALM_ID
  FROM ALMACEN AL
 WHERE AL.ALM_EMPRESA = V_CCO_EMPRESA_SM
    AND AL.ALM_CODIGO = V_CCO_ALMACEN_SM;
--OBTENGO EL PVE_ID (EJ.009)
SELECT PVE_ID
  INTO V_PVE_ID
  FROM PUNTOVENTA PV
 WHERE PV.PVE_ALMACEN = V_CCO_ALMACEN_SM
    AND PV.PVE_SECUENCIA = V_CCO_SERIE_SM
    AND PV.PVE_EMPRESA = V_CCO_EMPRESA_SM;

--DOCTRAN FORMADO POR ID_ALMACEN + PVE_ID(PUNTOVENTA) + NUEVO NUMERO
CALCULADO (V_NEW_NUMERO)
V_NEW_DOCTRAN := TO_CHAR(V_ALM_ID) || '-' || TO_CHAR(V_PVE_ID) || '-' ||
                LPAD(TO_CHAR(V_NEW_NUMERO), 9, '0');

/*****INSERTO LOS REGISTROS EN LA TABLA CCOMPROBA*****/
INSERT INTO CCOMPROBA
(CCO_EMPRESA,
 CCO_CODIGO,
 CCO_PERIODO,
 CCO_SIGLA,
 CCO_ALMACEN,
 CCO_SERIE,
 CCO_NUMERO,
 CCO_DOCTRAN,
 CCO_TIPODOC,
 CCO_FECHA,
 CCO_CONCEPTO,
 CCO_MODULO,
 CCO_NOCONTABLE,
 CCO_ESTADO,
 CCO_DESCUADRE,
 CCO_ADESTINO,
 CCO_PVENTA,
 CCO_CENTRO,
 CCO_TIPO_CAMBIO,
 CCO_TCLIPRO,
 CCO_CODCLIPRO,
 CCO_AGENTE,
 CCO_TRANSACC,
 CCO_CIE_COMPROBA,
 CCO_ANULADO,
 CREA_USR,
 CREA_FECHA,

```

```

MOD_USR,
MOD_FECHA,
CCO_BODEGA,
CCO_DIA,
CCO_MES,
CCO_ANIO,
CCO_RUTA)
VALUES
(V_CCO_EMPRESA_SM,
V_CCO_CODIGO_SM,
V_CCO_PERIODO_SM,
V_CCO_SIGLA_SM,
V_CCO_ALMACEN_SM,
V_CCO_SERIE_SM,
V_NEW_NUMERO,
V_NEW_DOCTRAN,
V_CCO_TIPODOC_SM,
TRUNC (SYSDATE),
V_CCO_CONCEPTO_SM,
V_CCO_MODULO_SM,
V_CCO_NOCONTABLE_SM,
V_CCO_ESTADO_SM,
V_CCO_DESCUADRE_SM,
V_CCO_ADESTINO_SM,
V_CCO_PVENTA_SM,
V_CCO_CENTRO_SM,
V_CCO_TIPO_CAMBIO_SM,
V_CCO_TCLIPRO_SM,
V_CCO_CODCLIPRO_SM,
V_CCO_AGENTE_SM,
V_CCO_TRANSACC_SM,
V_CCO_CIE_COMPROBA_SM,
V_CCO_ANULADO_SM,
V_CREA_USR_SM,
SYSDATE,
V_MOD_USR_SM,
SYSDATE,
V_CCO_BODEGA_SM,
TO_NUMBER(TO_CHAR(SYSDATE, 'DD')),
TO_NUMBER(TO_CHAR(SYSDATE, 'MM')),
TO_NUMBER(TO_CHAR(SYSDATE, 'YY')),
V_CCO_RUTA_SM);
COMMIT;

/*****INSERTO LOS REGISTROS EN LA TABLA CCOMFAC*****/
INSERT INTO CCOMFAC
(CFAC_EMPRESA,
CFAC_CCO_COMPROBA,
CFAC_AUTORIZA,
CFAC_POLITICA,
CFAC_LISTA_PRECIOS,
CFAC_EST_ENTREGA,
CFAC_PROC_FAC,
CFAC_PROCESO,
CFAC_NOMBRE,
CFAC_DIRECCION,
CFAC_TELEFONO,
CFAC_CED_RUC,
CFAC_CIUADAD,
CFAC_TIPO_ACTPRO,
CFAC_SOL_COMPROBA,
CFAC_IMPUESTO,
CFAC_PORC_IMPUESTO,
CFAC_TIPOPAGO,
CFAC_COMISION)
VALUES

```

```

(V_CCO_EMPRESA_SM,
V_CCO_CODIGO_SM,
V_CFAC_AUTORIZA_SM,
V_CFAC_POLITICA_SM,
V_CFAC_LISTA_PRECIOS_SM,
V_CFAC_EST_ENTREGA_SM,
V_CFAC_PROC_FAC_SM,
V_CFAC_PROCESO_SM,
V_CFAC_NOMBRE_SM,
V_CFAC_DIRECCION_SM,
V_CFAC_TELEFONO_SM,
V_CFAC_CED_RUC_SM,
V_CFAC_CIUADAD_SM,
V_CFAC_TIPO_ACTPRO_SM,
V_CFAC_SOL_COMPROBA_SM,
V_CFAC_IMPUESTO_SM,
V_CFAC_PORC_IMPUESTO_SM,
V_CFAC_TIPOPAGO_SM,
V_CFAC_COMISION_SM);
COMMIT;

END sp_movil_InsertCabeceraPedido;

-----
-- PROCEDIMIENTO PARA ALMACENAR EL DETALLE DEL PEDIDO (DFACTURA)
-- LOS REGISTROS SE ALMACENARAN EN LA DFACTURA
-- PARAMETROS QUE RECIBE:
-- REGSITROS PROVENIENTES DE LA TABLA:
-----

PROCEDURE sp_movil_InsertDetallePedido(DFAC_EMPRESA_SM
DFACTURA.DFAC_EMPRESA%TYPE,
        DFAC_CFAC_COMPROBA_SM
DFACTURA.DFAC_CFAC_COMPROBA%TYPE,
        DFAC_SECUENCIA_SM    DFACTURA.DFAC_SECUENCIA%TYPE,
        DFAC_PRODUCTO_SM    DFACTURA.DFAC_PRODUCTO%TYPE,
        DFAC_CATPRODUCTO_SM
DFACTURA.DFAC_CATPRODUCTO%TYPE,
        DFAC_CANTIDAD_SM    DFACTURA.DFAC_CANTIDAD%TYPE,
        DFAC_CANAPR_SM    DFACTURA.DFAC_CANAPR%TYPE,
        DFAC_PRECIO_SM    DFACTURA.DFAC_PRECIO%TYPE,
        DFAC_DESCUENTO_SM    DFACTURA.DFAC_DESCUENTO%TYPE,
        DFAC_BODEGA_SM    DFACTURA.DFAC_BODEGA%TYPE,
        DFAC_TOTAL_SM    DFACTURA.DFAC_TOTAL%TYPE,
        DFAC_CANENT_SM    DFACTURA.DFAC_CANENT%TYPE,
        DFAC_CANDEV_SM    DFACTURA.DFAC_CANDEV%TYPE,
        DFAC_CANRES_SM    DFACTURA.DFAC_CANRES%TYPE,
        DFAC_DSCITEM_SM    DFACTURA.DFAC_DSCITEM%TYPE,
        DFAC_TRAITEM_SM    DFACTURA.DFAC_TRAITEM%TYPE,
        DFAC_GRABAIVA_SM    DFACTURA.DFAC_GRABAIVA%TYPE,
        DFAC_UDIGITADA_SM    DFACTURA.DFAC_UDIGITADA%TYPE,
        DFAC_CDIGITADA_SM    DFACTURA.DFAC_CDIGITADA%TYPE,
        DFAC_PDIGITADO_SM    DFACTURA.DFAC_PDIGITADO%TYPE,
        DFAC_COMBO_SM    DFACTURA.DFAC_COMBO%TYPE)

IS
    DFAC_IVAITEM_SM DFACTURA.DFAC_IVAITEM%TYPE := 0;

BEGIN
    INSERT INTO DFACTURA
    (DFAC_EMPRESA,
    DFAC_CFAC_COMPROBA,
    DFAC_SECUENCIA,
    DFAC_PRODUCTO,
    DFAC_CATPRODUCTO,

```

```

DFAC_CANTIDAD,
DFAC_CANAPR,
DFAC_PRECIO,
DFAC_DESCUENTO,
DFAC_BODEGA,
DFAC_TOTAL,
DFAC_CANENT,
DFAC_CANDEV,
DFAC_CANRES,
DFAC_DSCITEM,
DFAC_TRAITEM,
DFAC_IVAITEM,
DFAC_GRABAIVA,
DFAC_UDIGITADA,
DFAC_CDIGITADA,
DFAC_PDIGITADO,
DFAC_COMBO)
VALUES
(DFAC_EMPRESA_SM,
DFAC_CFAC_COMPROBA_SM,
DFAC_SECUENCIA_SM,
DFAC_PRODUCTO_SM,
DFAC_CATPRODUCTO_SM,
DFAC_CANTIDAD_SM,
DFAC_CANAPR_SM,
DFAC_PRECIO_SM,
DFAC_DESCUENTO_SM,
DFAC_BODEGA_SM,
DFAC_TOTAL_SM,
DFAC_CANENT_SM,
DFAC_CANDEV_SM,
DFAC_CANRES_SM,
DFAC_DSCITEM_SM,
DFAC_TRAITEM_SM,
DFAC_IVAITEM_SM,
DFAC_GRABAIVA_SM,
DFAC_UDIGITADA_SM,
DFAC_CDIGITADA_SM,
DFAC_PDIGITADO_SM,
DFAC_COMBO_SM);
COMMIT;
END sp_movil_InsertDetallePedido;

```

```

--=====
--  PROCEDIMIENTO PARA ALMACENAR EL TOTAL DEL PEDIDO (TOTAL)
--  LOS REGISTROS SE ALMACENARAN EN LA TOTAL
--  PARAMETROS QUE RECIBE:
--  REGSITROS PROVENIENTES DE LA TABLA:
--  TOTAL DE SQL SERVER 2005 DE SELLERMOVIL
--=====

```

```

PROCEDURE sp_movil_InsertTotalPedido(
TOT_EMPRESA_SM          TOTAL.TOT_EMPRESA%TYPE,
TOT_CCO_COMPROBA_SM     TOTAL.TOT_CCO_COMPROBA%TYPE,
TOT_IMPUESTO_SM         TOTAL.TOT_IMPUESTO%TYPE,
TOT_PORC_DESC_SM        TOTAL.TOT_PORC_DESC%TYPE,
TOT_PORC_FINANC_SM      TOTAL.TOT_PORC_FINANC%TYPE,
TOT_PORC_PRO_PAGO_SM    TOTAL.TOT_PORC_PRO_PAGO%TYPE,
TOT_PORC_PAG_CONTA_SM   TOTAL.TOT_PORC_PAG_CONTA%TYPE,
TOT_LINEA_CREDITO_SM    TOTAL.TOT_LINEA_CREDITO%TYPE,
TOT_DIAS_PLAZO_SM       TOTAL.TOT_DIAS_PLAZO%TYPE,
TOT_NRO_PAGOS_SM        TOTAL.TOT_NRO_PAGOS%TYPE,
TOT_SUBTOTAL_SM         TOTAL.TOT_SUBTOTAL%TYPE,
TOT_DESCUENTO01_SM      TOTAL.TOT_DESCUENTO01%TYPE,
TOT_TIMPUESTO_SM        TOTAL.TOT_TIMPUESTO%TYPE,

```

```

TOT_TRANSPORTE_SM      TOTAL.TOT_TRANSPORTE%TYPE,
TOTAL.TOT_SEGURO_TRANS%TYPE,
TOT_AJUSTE_SM          TOTAL.TOT_AJUSTE%TYPE,
TOT_FINANCIA_SM        TOTAL.TOT_FINANCIA%TYPE,
TOT_TOTAL_SM           TOTAL.TOT_TOTAL%TYPE,
TOT_PORC_IMPUESTO_SM   TOTAL.TOT_PORC_IMPUESTO%TYPE,
TOT_ICE_SM             TOTAL.TOT_ICE%TYPE,
TOT_NETO_CLIENTE_SM    TOTAL.TOT_NETO_CLIENTE%TYPE,
TOT_DESC1_0_SM         TOTAL.TOT_DESC1_0%TYPE,
TOT_DESC2_0_SM         TOTAL.TOT_DESC2_0%TYPE,
TOT_SUBTOT_0_SM        TOTAL.TOT_SUBTOT_0%TYPE,
TOT_TIMPUESTO1_SM      TOTAL.TOT_TIMPUESTO1%TYPE,
TOT_ANTICIPO_SM        TOTAL.TOT_ANTICIPO%TYPE)

```

```

IS
BEGIN
  INSERT INTO TOTAL
    (TOT_EMPRESA,
     TOT_CCO_COMPROBA,
     TOT_IMPUESTO,
     TOT_PORC_DESC,
     TOT_PORC_FINANC,
     TOT_PORC_PRO_PAGO,
     TOT_PORC_PAG_CONTA,
     TOT_LINEA_CREDITO,
     TOT_DIAS_PLAZO,
     TOT_NRO_PAGOS,
     TOT_SUBTOTAL,
     TOT_DESCUENTO1,
     TOT_TIMPUESTO,
     TOT_TRANSPORTE,
     TOT_SEGURO_TRANS,
     TOT_AJUSTE,
     TOT_FINANCIA,
     TOT_TOTAL,
     TOT_PORC_IMPUESTO,
     TOT_ICE,
     TOT_NETO_CLIENTE,
     TOT_DESC1_0,
     TOT_DESC2_0,
     TOT_SUBTOT_0,
     TOT_TIMPUESTO1,
     TOT_ANTICIPO)
  VALUES
    (TOT_EMPRESA_SM,
     TOT_CCO_COMPROBA_SM,
     TOT_IMPUESTO_SM,
     TOT_PORC_DESC_SM,
     TOT_PORC_FINANC_SM,
     TOT_PORC_PRO_PAGO_SM,
     TOT_PORC_PAG_CONTA_SM,
     TOT_LINEA_CREDITO_SM,
     TOT_DIAS_PLAZO_SM,
     TOT_NRO_PAGOS_SM,
     TOT_SUBTOTAL_SM,
     TOT_DESCUENTO1_SM,
     TOT_TIMPUESTO_SM,
     TOT_TRANSPORTE_SM,
     TOT_SEGURO_TRANS_SM,
     TOT_AJUSTE_SM,
     TOT_FINANCIA_SM,
     TOT_TOTAL_SM,
     TOT_PORC_IMPUESTO_SM,
     TOT_ICE_SM,
     TOT_NETO_CLIENTE_SM,
     TOT_DESC1_0_SM,
     TOT_DESC2_0_SM,

```

```

        TOT_SUBTOT_0_SM,
        TOT_TIMPUESTO1_SM,
        TOT_ANTICIPO_SM);
    COMMIT;
END sp_movil_InsertTotalPedido;

--=====
-- PROCEDIMIENTO PARA ELIMINAR LA CABECERA DE UN PEDIDO EN PROCESO SQL TO
ORACLE
-- PARAMETRO: CCO_CODIGO
--=====

PROCEDURE sp_deletePedido(V_CCO_CODIGO CCOMPROBA.CCO_CODIGO%TYPE) IS
BEGIN
    DELETE CCOMFAC WHERE CFAC_CCO_COMPROBA = V_CCO_CODIGO;
    COMMIT;
    DELETE CCOMPROBA WHERE CCO_CODIGO = V_CCO_CODIGO;
    COMMIT;
END sp_deletePedido;

--=====
-- PROCEDIMIENTO PARA ANULAR UN PEDIDO QUE NO SE HAN FACTURADO TODAVIA
-- PARAMETROS: CCO_EMPRESA
-- CCO_CODIGO
-- USR_ID
--=====
PROCEDURE sp_movil_AnulaPedido(V_CCO_EMPRESA CCOMPROBA.CCO_EMPRESA%TYPE,
                                V_CCO_CODIGO CCOMPROBA.CCO_CODIGO%TYPE,
                                V_USR_ID USUARIO.USR_ID%TYPE) IS

    V_RESULTADO BOOLEAN;
BEGIN
    V_RESULTADO := ast_cnt.eliminar_pedido(V_CCO_EMPRESA,
                                            V_CCO_CODIGO,
                                            V_USR_ID);

END sp_movil_AnulaPedido;

--=====
-- FUNCION QUE DEVUELVE EL PRECIO DE UN PRODUCTO
-- PARAMETROS QUE RECIBE:
-- LISTAPRE
-- PRODUCTO
-- CATPRODUCTO
-- UMEDIDA
-- ALMACEN
-- EMPRESA
--=====

FUNCTION get_precio(v_plistapre listapre.lpr_codigo%TYPE,
                    v_pproducto dfactura.dfac_producto%TYPE,
                    v_pcatproducto dfactura.dfac_catproducto%TYPE,
                    v_pumedida umedida.umd_codigo%TYPE,
                    v_padestino ccomproba.cco_adeestino%TYPE,
                    v_pempresa ccomproba.cco_empresa%TYPE)
    return number is
    precio_item dlistapre.dlp_precio%TYPE;

    CURSOR c_dlistapre(v_plistapre listapre.lpr_codigo%TYPE,
v_pproducto dfactura.dfac_producto%TYPE, v_pcatproducto
dfactura.dfac_catproducto%TYPE, v_pumedida umedida.umd_codigo%TYPE,

```

```

v_padestino ccomproba.cco_adeestino%TYPE, v_pempresa
ccomproba.cco_empresa%TYPE) IS
    SELECT dlp_precio, dlp_almacen, dlp_descuento
    FROM dlistapre
    WHERE dlp_producto = v_producto
      AND dlp_listapre = v_plistapre
      AND dlp_empresa = v_pempresa
      AND nvl(dlp_catproducto, 0) = nvl(v_pcatproducto, 0)
      AND dlp_umedida = v_pumedida
      AND nvl(dlp_inactivo, 0) = 0
      AND (((sysdate BETWEEN (dlp_fecha_ini) AND
(dlp_fecha_fin+1)) OR
          (dlp_fecha_ini <= sysdate) AND (dlp_fecha_fin IS
NULL)))
    ORDER BY DLP_ALMACEN , DLP_FECHA_INI DESC;

```

```

BEGIN
    precio_item := NULL;
    FOR i IN c_dlistapre(v_plistapre,
                        v_producto,
                        v_pcatproducto,
                        v_pumedida,
                        v_padestino,
                        v_pempresa) LOOP
        IF i.dlp_almacen = v_padestino THEN
            precio_item := i.dlp_precio;
            EXIT;
        ELSIF i.dlp_almacen IS NULL AND precio_item is null THEN
            precio_item := i.dlp_precio;
            EXIT;
        END IF;
    END LOOP;
    RETURN precio_item;
END get_precio;

```

```

CREATE OR REPLACE
PACKAGE BODY "AST_SELLERWEB" AS
function Sp_Web_Login(
    V_EMPRESA CLIENTE.CLI_EMPRESA%TYPE,
    V_CLI_MAIL CLIENTE.CLI_MAIL%TYPE,
    V_CLI_PASSWORD CLIENTE.CLI_PASSWORD%TYPE
return NUMBER AS
    V_CLI_CODIGO NUMBER:=NULL;

    BEGIN
        select CLI_CODIGO
        INTO V_CLI_CODIGO
        from cliente
        where cliente.cli_mail=V_CLI_MAIL
        and cliente.cli_password=V_CLI_PASSWORD;

        return V_CLI_CODIGO;

    END Sp_Web_Login;

```

```

function Sp_Web_DatosCliente(V_EMPRESA CLIENTE.CLI_EMPRESA%TYPE,
                             V_CLIENTE CLIENTE.CLI_CODIGO%TYPE

```

```

return SYS_REFCURSOR
AS RS sys_refcursor;
BEGIN
    open RS FOR
    SELECT a.cli_agente as CodigoAgente,b.AGE_BODEGA as
CodigoBodega
    from CLIENTE a, AGENTE b
    where a.CLI_CODIGO=V_CLIENTE
    and a.CLI_EMPRESA=V_EMPRESA
    and a.CLI_EMPRESA=b.AGE_EMPRESA
    and a.CLI_AGENTE=b.AGE_CODIGO;

    return RS;

END Sp_Web_DatosCliente;

function Sp_Web_ListaPedidosCliente(V_EMPRESA
CLIENTE.CLI_EMPRESA%TYPE,V_CLIENTE CLIENTE.CLI_CODIGO%TYPE) return
SYS_REFCURSOR
AS RS sys_refcursor;
BEGIN
    OPEN RS FOR
    SELECT NUMERO_COMPROBANTE, FECHA, TOTAL
    from FAC_PEDIDOS_V
    where CODIGO=V_CLIENTE
    and EMPRESA=V_EMPRESA;

    return RS;

END Sp_Web_ListaPedidosCliente;

procedure Sp_Web_ModificaCliente(
pp_Cli_Ciudad          CLIENTE.Cli_Ciudad%Type,
pp_Cli_Direccion        CLIENTE.Cli_Direccion%Type,
pp_Cli_Telefono1        CLIENTE.Cli_Telefono1%Type,
pp_Cli_Mail             CLIENTE.Cli_Mail%Type,
pa_CodCliente           CLIENTE.CLI_CODIGO%Type,
pa_CodEmpresa           CLIENTE.CLI_EMPRESA%Type,
pp_NombreCliente        CLIENTE.cli_nombre%type,
                        ) is
begin
    update Cliente
    set Cli_Ciudad      = pp_Cli_Ciudad,
        Cli_Direccion   = pp_Cli_Direccion,
        Cli_Telefono1   = pp_Cli_Telefono1,
        Cli_mail        = pp_cli_mail,
        Cli_Nombre      = pp_NombreCliente,
    Where CLI_CODIGO = pa_CodCliente
    and CLI_EMPRESA = pa_CodEmpresa
    ;
    commit;
end Sp_Web_ModificaCliente;

-----
-- FUNCION QUE DEVUELVE EL CODIGO DEL COMPROBANTE DE INVENTARIO
-- REALIZA EL COMPROBANTE DE INVENTARIO PARA SEPARAR PRODUCTO
-- QUE EL CLIENTE REALIZO EL PEDIDO

```



```

--  PARAMETROS QUE RECIBE:
--  PN_EMPRESA
--  PN_COMPROBA    (Pedido del que se realiza el comprobante de
inventario)
=====
FUNCTION Sp_Web_GeneraNotaEntrega(
    Pn_empresa  ccomproba.cco_empresa%TYPE,
    Pn_comproba ccomproba.cco_codigo%TYPE) return number IS

    v_numero          ccomproba.cco_numero%TYPE := 0;
v_codigo_nota  ccomproba.cco_codigo%TYPE := 0;
v_periodo      ccomproba.cco_periodo%TYPE := 0;
v_sigla        ccomproba.cco_sigla%TYPE := 0;
v_almacen      ccomproba.cco_almacen%TYPE := 0;
v_serie        ccomproba.cco_serie%TYPE := 0;
v_transacc     ccomproba.cco_transacc%TYPE;
v_bodega_promo bodega.bod_codigo%TYPE;
v_bodega       bodega.bod_codigo%TYPE;
v_cantidad1    dmovinv.dmo_cantidad%TYPE;
--Cursor que recupera los campos de ccomproba de la factura
grabada
CURSOR c_comproba (v_codigo ccomproba.cco_codigo%TYPE) IS
SELECT *
FROM ccomproba
WHERE cco_empresa = pn_empresa
AND cco_codigo = v_codigo
AND cco_estado IN (2,3);

CURSOR c_bodega (v_bodega bodega.bod_codigo%TYPE) IS
SELECT b.bod_codigo
FROM ccomproba, bodega a, bodega b
WHERE cco_empresa = pn_empresa
AND cco_codigo = pn_comproba
AND cco_estado IN (2,3)
AND cco_bodega = a.bod_codigo
AND cco_empresa = a.bod_empresa
and a.bod_almacen = b.bod_almacen
and a.bod_empresa = b.bod_empresa
and nvl(b.bod_promo,0) = 1;

CURSOR c_nen (v_codigo ccomproba.cco_codigo%TYPE) IS
SELECT cco_codigo, cco_numero, cco_estado
FROM cmovinv, ccomproba
WHERE cmo_empresa = pn_empresa
AND cmo_cco_factura = v_codigo
AND cmo_empresa = cco_empresa
AND cmo_cco_comproba = cco_codigo;

--Cursor que recupera los campos de ccomfac de la factura
generada
CURSOR c_cfac (v_codigo ccomproba.cco_codigo%TYPE) IS
SELECT *
FROM ccomfac, ccomproba
WHERE cfac_empresa = pn_empresa
AND cfac_cco_comproba = v_codigo
AND cfac_empresa = cco_empresa
AND cfac_cco_comproba = cco_codigo
AND cco_estado IN (2,3);

CURSOR c_dfactura (v_comproba ccomproba.cco_codigo%TYPE) IS

```

```

SELECT a.*, pro_inventario, cco_fecha, cco_bodega
FROM dfactura a, producto, ccomproba
WHERE cco_empresa = pn_empresa
  AND cco_codigo = v_Comproba
  AND( pro_codigo = dfac_producto
  AND pro_empresa = dfac_empresa)
  AND (dfac_cfacs_comproba = cco_codigo
  AND dfac_empresa = cco_empresa)
  and nvl(dfacs_combo,0) = 0
  AND cco_estado IN (2,3)
ORDER BY dfacs_secuencia;

CURSOR c_dfacscombo (v_comproba ccomproba.cco_codigo%TYPE) IS
SELECT a.*, b.*, pro_inventario, cco_fecha, cco_bodega,
com_unidad, com_cantidad,
fac_factor, com_promocion,pro_tipo_combo
FROM dfacscombo b, producto, ccomproba, dfactura a, compone,
factor
WHERE cco_empresa = pn_empresa
  AND cco_codigo = v_Comproba
  AND (dfacs_cfacs_comproba = cco_codigo
  AND dfacs_empresa = cco_empresa)
  AND (dfc_dfacs_comproba = cco_codigo
  AND dfc_empresa = cco_empresa)
  AND( pro_codigo = dfc_producto
  AND pro_empresa = dfc_empresa)
-- and dfc_producto = dfacs_producto
  and dfc_dfacs_secuencia = dfacs_secuencia
  and nvl(dfacs_combo,0) = 1
  AND cco_estado IN (2,3)
  and dfc_producto = com_componente
  and a.dfacs_producto = com_producto
  and dfc_empresa = com_empresa
  and com_unidad = fac_unidad
  and com_componente = fac_producto
  and com_empresa = fac_empresa
ORDER BY dfc_dfacs_secuencia;

CURSOR c_inventario (v_comproba ccomproba.cco_codigo%TYPE) IS
SELECT max(nvl(pro_inventario,0)) pro_inventario
FROM dfactura a, producto
WHERE dfacs_empresa = pn_empresa
  AND dfacs_cfacs_comproba = v_Comproba
  AND (pro_codigo = dfacs_producto
  AND pro_empresa = dfacs_empresa);

--Cursor que recupera la sigla del documento de "Nota de Entrega"
CURSOR c_sigla IS
SELECT cti_codigo
FROM ctipocom, dtipodoc
WHERE cti_empresa = pn_empresa
  AND dtp_empresa = cti_empresa
  AND dtp_cti_codigo = cti_codigo
  AND dtp_tpd_codigo = 31
  AND nvl(cti_inactivo,0) != 1;
--Variables para la secuencia del detalle de DMOVINV y el control
del CCOMPROBA Y CMOVINV
  v_secuencia NUMBER := 0;
  v_bandera NUMBER(1) := 0;
  --Para el costo del producto que va en el DMOVINV

```

```

        v_dcosto NUMBER := 0;
        --Para extraer el comprobante de la factura recién grabada y
        ponerla en el concepto de la nota de entrega
        v_comprobante VARCHAR2(50);
        v_resultado BOOLEAN := TRUE;
        v_almacen_usuario almacen.alm_codigo%TYPE := 0;
        v_inventario number := 0;
        CURSOR c_acceso(p_usuario USUARIO.USR_id%TYPE) IS
            SELECT acc_almacen FROM ACCESO, USUARIO
            WHERE ACC_EMPRESA = Pn_empresa
            AND ACC_USUARIO = usr_codigo
            AND USR_ID = p_usuario;

BEGIN
    -- Si se esta modificando coge el numero de nota de entrega
    para mantener el mismo numero y elimina
    -- la nota de entrega para generar la nueva.
    FOR i IN c_inventario(pn_comproba) LOOP
        v_inventario := i.pro_inventario;
    END LOOP;
    FOR i IN c_nen(pn_COMPROBA) LOOP
        IF i.cco_estado != 9 THEN
            v_codigo_nota := i.cco_codigo;
            v_numero := i.cco_numero;
            IF i.cco_estado = 3 THEN
                BEGIN
                    UPDATE CCOMPROBA SET cco_estado = 2
                    WHERE Cco_codigo = i.cco_codigo
                    AND Cco_empresa = pn_empresa;
                EXCEPTION WHEN OTHERS THEN
                    NULL;
                END;
            END IF;
            v_resultado :=
ast_cnt.eliminar_comprobante(pn_empresa, i.cco_codigo, 0);
        END IF;
        IF v_resultado = TRUE THEN
            BEGIN
                DELETE DMOVINV
                WHERE dmo_cmo_comproba = i.cco_codigo
                AND dmo_empresa = pn_empresa;
            EXCEPTION WHEN OTHERS THEN
                null;
            END;
        END IF;
    END LOOP;
    IF v_resultado = TRUE AND v_inventario = 1 THEN
        --Extrae el comprobante de la factura actualmente
        grabada para ponerla en el concepto de la nota
        v_comprobante:=ast_gen.numero_comprobante(pn_empresa,
pn_comproba);
        --Extrae la sigla de la nota de entrega
        FOR i IN c_sigla LOOP
            v_sigla := i.cti_codigo;
        END LOOP;
        --Extrae los campos del pedido recién grabado
        FOR i IN c_comproba(pn_comproba) LOOP
            --
            v_fecha:=i.cco_fecha;
        For j In c_bodega (i.cco_bodega) LOOP
            v_bodega_promo:=j.bod_codigo;

```

```

END LOOP;
FOR j IN c_acceso(i.crea_usr) LOOP
    IF j.acc_almacen IS NOT NULL THEN
        v_almacen_usuario := j.acc_almacen;
    END IF;
END LOOP;

v_periodo := i.cco_periodo;
v_almacen := i.cco_almacen;
v_serie := i.cco_serie;
v_transacc := i.cco_transacc;

IF nvl(v_numero,0) = 0 THEN
    v_numero := ast_cnt.secuencia_ccomproba(i.cco_empresa,
i.cco_periodo,
v_sigla, 0, i.cco_bodega);
END IF;
IF nvl(v_codigo_nota,0) = 0 AND nvl(v_numero,0) > 0 THEN
    BEGIN
        SELECT ccomproba_s_codigo.NEXTVAL
            INTO v_codigo_nota FROM DUAL;
        EXCEPTION WHEN OTHERS THEN
            dbms_output.put_line('NO EXISTE SECUENCIA
CCOMPROBA_S_CODIGO');
            v_codigo_nota := 0;
        END;
        IF v_almacen_usuario > 0 THEN
            v_codigo_nota := v_codigo_nota +
(v_almacen_usuario * (POWER (10, 26)));
        END IF;
    END IF;
    IF nvl(v_codigo_nota,0) = 0 OR nvl(v_numero,0) = 0 THEN
        v_resultado := FALSE;
    END IF;
    IF v_resultado = TRUE THEN
        BEGIN
            INSERT INTO ccomproba FIELDS(CCO_EMPRESA,
CCO_CODIGO,CCO_PERIODO,
CCO_SIGLA,CCO_ALMACEN,CCO_SERIE, CCO_NUMERO,
CCO_DOCTRAN, CCO_TIPODOC, CCO_FECHA, CCO_CONCEPTO,
CCO_MODULO, CCO_NOCONTABLE, CCO_ESTADO, CCO_DESCUADRE,
CCO_ADESTINO, CCO_PVENTA, CCO_CENTRO, CCO_TIPO_CAMBIO,
CCO_TCLIPRO, CCO_CODCLIPRO , CCO_AGENTE, CCO_CUENTA,
CCO_TRANSACC, CCO_CODCLIPRO1, CCO_CIE_COMPROBA,
CCO_REF_COMPROBA,
CCO_ANULADO,CCO_ANU_COMPROBA, CCO_AUT_TIPO, CREA_USR,
CREA_FECHA,
MOD_USR,MOD_FECHA,CCO_NIVEL_APROB, CCO_BODEGA, CCO_FECHAFIN,
CCO_DIA,CCO_MES, CCO_ANIO, CCO_ESTADO_LIQ, CCO_CATCLIENTE)
VALUES (
pn_empresa,
v_codigo_nota, v_periodo,
v_sigla, null, i.cco_bodega, v_numero,
i.cco_doctran, 31, i.cco_fecha, 'NOTA DE ENTREGA DEL PEDIDO :
'||v_comprobante, 7, i.cco_nocontable, i.cco_estado,
i.cco_descuadre, i.cco_adeestino, i.cco_pventa,
i.cco_centro, i.cco_tipo_cambio, i.cco_tclipro,
i.cco_codclipro, i.cco_agente, i.cco_cuenta, 3,
i.cco_codclipro1, i.cco_cie_comproba, i.cco_ref_comproba,
i.cco_anulado, i.cco_anu_comproba, i.cco_aut_tipo, i.crea_usr,
i.crea_fecha, i.mod_usr, i.mod_fecha, i.cco_nivel_aprob,
i.cco_bodega, i.cco_fechafin,

```

```

        null,null,null,null,null );
    EXCEPTION WHEN DUP_VAL_ON_INDEX THEN
        UPDATE CCOMPROBA
            SET cco_fecha = i.cco_fecha,
                cco_estado = i.cco_estado,
                cco_tclipro = i.cco_tclipro,
                cco_codclipro = i.cco_codclipro,
                cco_agente = i.cco_agente,
                cco_cie_comproba = i.cco_cie_comproba,
                cco_bodega = i.cco_bodega
            WHERE Cco_codigo = v_codigo_nota
            AND Cco_empresa = pn_empresa;
    END;
END IF;
END LOOP;
--Graba en la cabecera adicional de la nota(s), tabla CMOVINV
IF v_resultado = TRUE THEN
    FOR i IN c_cfacs (pn_comproba) LOOP
        BEGIN
            INSERT INTO cmovinv FIELDS(
                CMO_EMPRESA,CMO_CCO_COMPROBA, CMO_TRANSACC,
                CMO_CCO_PEDIDO, CMO_CCO_FACTURA , CMO_CCO_REFERENCIA, CMO_BODINI,
                CMO_BODFIN, CMO_PROC_FAC, CMO_EST_ENTREGA, CMO_LIQUIDA,
                CMO_OTR_COMPROBA, CMO_PRODserie, CMO_REFERENCIA, CMO_CONTROL_TEMP,
                CMO_USR_LIQUIDA)
                VALUES (pn_empresa, v_codigo_nota, 3,
                    Pn_comproba, null, null,
                    i.cco_bodega, i.cco_bodega,
                    i.cfacs_proc_fac,
                    nvl(i.cfacs_est_entrega,0), 0,
                    i.cfacs_otr_comproba, 0,
                    '', 0, 0);
            EXCEPTION WHEN DUP_VAL_ON_INDEX THEN
                UPDATE cmovinv SET
                    CMO_CCO_PEDIDO = i.cfacs_cco_pedido,
                    CMO_CCO_FACTURA = i.cfacs_cco_comproba,
                    CMO_BODINI = i.cco_bodega,
                    CMO_BODFIN = i.cco_bodega,
                    CMO_EST_ENTREGA = i.cfacs_est_entrega
                WHERE CMO_CCO_COMPROBA = v_codigo_nota
                AND CMO_EMPRESA = pn_empresa;
            END;
        END LOOP;
    END IF;
    v_secuencia := 0;
    IF v_resultado = TRUE THEN
        FOR i IN c_dfactura(pn_comproba) LOOP
            -- anade el detalle
            v_secuencia := v_secuencia + 1;
            --Saca costo unitario del producto
            v_dcosto := ast_auto.costo_unitario(Pn_empresa,
                i.dfacs_producto, i.dfacs_catproducto, i.dfacs_bodega, i.cco_fecha);
            v_cantidad1:=0;
            v_cantidad1:=ast_saldos.control_saldo('a', 3, 2,
                'c',pn_empresa, null, i.dfacs_producto,
                i.dfacs_bodega, i.dfacs_catproducto, i.cco_fecha);
            If v_cantidad1 >= i.dfacs_cantidad Then
                v_cantidad1:=i.dfacs_cantidad;
            End If;
        END LOOP;
    END IF;
END IF;

```

```

        End If;
        --Graba en el detalle de la(s) nota(s) en
        la tabla DMOVINV
        IF nvl(i.pro_inventario,0) = 1 THEN
        BEGIN
            INSERT INTO dmovinv
            (
            DMO_EMPRESA,DMO_CMO_COMPROBA, DMO_SECUENCIA, DMO_PRODUCTO,
            DMO_CATPRODUCTO, DMO_TRANSACCION, DMO_DEBCRE, DMO_BODEGA,
            DMO_CANTIDAD, DMO_DEVUELTA, DMO_DESPACHADA, DMO_DEVDESPA,
            DMO_COSTO, DMO_TOTAL, DMO_CANT_FISICA,
            DMO_CANT_TRANSITO, DMO_CANT_CONSIGNA,DMO_CUENTA, CREA_USR,
            CREA_FECHA, MOD_USR, MOD_FECHA, DMO_CENTRO, DMO_FECHA,
            DMO_TEMPERATURA,DMO_GRADO,DMO_UDIGITADA,DMO_CDIGITADA,
            DMO_PDIGITADO, DMO_TIPODEV)
            VALUES ( Pn_empresa, v_codigo_nota, i.dfacs_secuencia,
            i.dfacs_producto, nvl(i.dfacs_catproducto,0),3, 2, i.cco_bodega,
            v_cantidad1, i.dfacs_candev, i.dfacs_canent, 0, v_dcosto, ROUND(
            NVL((v_dcosto * v_cantidad1),0),2), v_cantidad1 * (-1), 0, 0,
            i.dfacs_cuenta, i.crea_usr, i.crea_fecha, i.mod_usr,
            i.mod_fecha,null,SYSDATE, i.dfacs_temperatura,i.dfacs_grado,
            i.dfacs_udigitada, v_cantidad1, i.dfacs_pdigitado,
            null);
            UPDATE DFACTURA set dfacs_canent =
            nvl(dfacs_canent,0)+v_cantidad1
            where dfacs_cfacs_comproba = pn_comproba
            and dfacs_empresa = pn_empresa
            and dfacs_producto = i.dfacs_producto
            and dfacs_secuencia = i.dfacs_secuencia;
        EXCEPTION WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE(SQLCODE || ' ' || SQLERRM);
            v_resultado := FALSE;
        END;
        END IF;
        END LOOP;
    END IF;
END IF;
IF v_resultado = TRUE THEN
    COMMIT;
    IF v_inventario = 1 THEN
        ast_fac.asiento_egreso_bodega(pn_empresa,
        v_codigo_nota);
        COMMIT;
        RETURN v_codigo_nota;
    ELSE
        RETURN 0;
    END IF;
ELSE
    ROLLBACK;
    RETURN null;
END IF;
EXCEPTION WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE(SQLCODE || ' ' || SQLERRM);
    ROLLBACK;
    RETURN null;
END;

```

```

function Sp_Web_ListarProductos(V_EMPRESA CLIENTE.CLI_EMPRESA%TYPE,
                                V_BODEGA BODEGA.BOD_CODIGO%TYPE,
                                V_CRITERIOBUSQUEDA PRODUCTO.PRO_NOMBRE%TYPE
                                )

return SYS_REFCURSOR
AS   RS sys_refcursor;
BEGIN
    OPEN RS FOR
        SELECT PRO_CODIGO AS Codigo, PRO_NOMBRE AS Nombre
        FROM DATA_USR."web_producto_stock"
        WHERE BOD_EMPRESA = V_EMPRESA
        AND BOD_CODIGO = V_BODEGA
        AND (PRO_NOMBRE LIKE '%' || "UPPER"(V_CRITERIOBUSQUEDA) || '%'
        OR PRO_CODIGO LIKE '%' || "UPPER"(V_CRITERIOBUSQUEDA) || '%')
        AND ROWNUM <= 10;

    return RS;

END Sp_Web_ListarProductos;

function Sp_Web_CargarCabeceraPedido(V_EMPRESA
CCOMFAC.CFAC_EMPRESA%TYPE,
V_NUMCOMPROBANTE CCOMFAC.CFAC_CCO_PEDIDO%TYPE)
                                return SYS_REFCURSOR
AS   RS sys_refcursor;

BEGIN
    OPEN RS FOR
        SELECT CFAC_NOMBRE as nombre, CFAC_DIRECCION as direccion,
        CFAC_TELEFONO as telefono, CFAC_CED_RUC as ci
        FROM CCOMFAC
        WHERE CFAC_EMPRESA = V_EMPRESA
        AND CFAC_CCO_PEDIDO = V_NUMCOMPROBANTE;

    return RS;

END Sp_Web_CargarCabeceraPedido;

function Sp_Web_CargarDetallePedido(V_EMPRESA
DFACCOMBO.DFC_EMPRESA%TYPE, V_NUMCOMPROBANTE
DFACCOMBO.DFC_DFAC_COMPROBA%TYPE)
                                return SYS_REFCURSOR
AS   RS sys_refcursor;

BEGIN
    OPEN RS FOR
        SELECT DFC_PRODUCTO, DFC_CANTIDAD, DFC_PRECIO
        FROM DFACCOMBO
        WHERE DFC_EMPRESA = V_EMPRESA
        AND DFC_DFAC_COMPROBA = V_NUMCOMPROBANTE;

    return RS;

END Sp_Web_CargarDetallePedido;

function Sp_Web_EstadoPedido(V_EMPRESA CCOMPROBA.CCO_EMPRESA%TYPE,

```

```

                                V_CODIGO    CCOMPROBA.CCO_CODIGO%TYPE,
                                V_ESTADO    CCOMFAC.CFAC_EST_ENTREGA%TYPE)

return NUMBER AS

BEGIN

    Update ccomfac set cfac_est_entrega = v_estado
    where cfac_cco_comproba = v_codigo
    and cfac_empresa = v_empresa;

    commit;

    return 0;

END Sp_Web_EstadoPedido;

Function Sp_Web_InsertCliente(pcli_empresa
Cliente.cli_empresa%type,
                                pcli_nombre
Cliente.cli_nombre%type,
                                pcli_ced_ruc
Cliente.Cli_Ruc_Cedula%type,
                                pcli_ciudad
cliente.cli_ciudad%type,
                                pcli_direccion
Cliente.Cli_Direccion%type,
                                pcli_telefono
Cliente.Cli_Telefono1%type,
                                pcli_mail
Cliente.Cli_Mail%type,
                                pcli_password
Cliente.Cli_Password%type) return number is
    pcli_codigo    cliente.cli_codigo%type; -- VARIABLE PARA GENERAR
CLI_CODIGO
    pcli_id        cliente.cli_id%type; -- VARIABLE PARA GENERAR
CLI_IDE
    pcli_zona      cliente.cli_zona%type; ---- VARIABLE PARA GENERAR
CLI_ZONA
    pcli_fechaing  cliente.cli_fechaing%type; ---- VARIABLE PARA
GENERAR FECHA ACTUAL
    pcli_impuestos cliente.cli_impuestos%type; --
    pcli_inactivo  cliente.cli_inactivo%type; --
    pcli_cupo     cliente.cli_cupo%type; --
    pcli_ilimitado cliente.cli_ilimitado%type; --
    v_correcto     number:=0;
    v_tipoced     number:=0;

    pv_codigo     number; -- nuevo codigo para el nuevo cliente

begin
    If Length(pcli_ced_ruc) = 10 Then
        v_tipoced:=1;
    ElseIf Length(pcli_ced_ruc) = 13 Then
        v_tipoced:=2;
    End If;
    If v_tipoced != 0 Or
        (pcli_ced_ruc) <> '' Then
        v_correcto:=ast_fac1.validar_cedula(pcli_ced_ruc);
    Else
        v_correcto:=1;

```



```

        null,
        0,
        0,
        1,
        10000296,
        1,
        v_tipoced,
        sysdate,
        0,
        null,
        null,
        'VAC',
        sysdate,
        'VAC',
        sysdate,
        pcli_password);

    commit;
End If;

        return v_correcto;

end Sp_Web_InsertCliente;

Function Sp_Web_ModificaCliente(pcli_empresa
Cliente.cli_empresa%type, pcli_codigo
Cliente.cli_codigo%type, pcli_nombre
Cliente.cli_nombre%type, pcli_ced_ruc
Cliente.Cli_Ruc_Cedula%type, pcli_ciudad
cliente.cli_ciudad%type, pcli_direccion
Cliente.Cli_Direccion%type, pcli_telefono
Cliente.Cli_Telefono1%type, pcli_mail
Cliente.Cli_Mail%type, pcli_password
Cliente.Cli_Password%type) return number is

begin

        Update cliente set cli_nombre = pcli_nombre,
                           cli_ruc_cedula = pcli_ced_ruc,
                           cli_direccion = pcli_direccion,
                           cli_ciudad = pcli_ciudad,
                           cli_telefono1= pcli_telefono,
                           cli_mail = pcli_mail,
                           cli_password = pcli_password
        where cli_codigo = pcli_codigo
          and cli_empresa = pcli_empresa;

    commit;
    return 0;

end Sp_Web_ModificaCliente;

END AST_SELLERWEB;
b) Vistas

CREATE OR REPLACE VIEW FAC_PEDIDOS_V AS

```

```

SELECT
--GENERALES DE LA FACTURA
EMPRESA, CCO_CODIGO, NUMERO_COMPROBANTE, FECHA, PERIODO, MES,
CCO_CIE_COMPROBA, LIQUIDACION,
-- lista de precios con la que se realizo la venta
LISTA_PRECIOS_VENTA,
-- agencia
CCO_ADESTINO, AGENCIA,
-- DATOS DEL CLIENTE
CODIGO , NOMBRE_RAZON, RUC_CEDULA, RUTA,
-- DATOS DE LA FACTURA
-- producto
MERCADERIA, ID, PRODUCTO,
--z.fac_unidad,
UNIDAD_P_COMERCIAL, UNIDAD_INDIVISIBLE_COMERCIAL,
-- Datos de la cadena
CCO_CADAGENTE, CCO_AGENTE,
--TOTALES DE LA FACTURA
sum(SUBTOTAL) SUBTOTAL,
sum(DESCUENTO) DESCUENTO,
sum(IVA) IVA,
sum(NETO_SIN_IVA) NETO_SIN_IVA,
sum(TOTAL) TOTAL,
-- CANTIDADES
sum(CANTIDAD_DIGITADA) CANTIDAD_DIGITADA,
sum(CANTIDAD_P_COMERCIAL) CANTIDAD_P_COMERCIAL,
sum(CANTIDAD_INDIVISIBLE_COMERCIAL) CANTIDAD_INDIVISIBLE_COMERCIAL
from
(SELECT
--GENERALES DE LA FACTURA
a.cco_empresa      EMPRESA,
a.cco_codigo       CCO_CODIGO,
ast_gen.numero_comprobante(a.cco_empresa,a.cco_codigo)
NUMERO_COMPROBANTE,
a.cco_fecha        FECHA,
a.cco_periodo      PERIODO,
a.cco_mes          MES,
a.cco_cie_comproba,
ast_gen.numero_comprobante(LIQ.cco_empresa,LIQ.cco_codigo)
LIQUIDACION,
-- lista de precios con la que se realizo la venta
lpv.lpr_nombre LISTA_PRECIOS_VENTA,
-- agencia
a.cco_adeestino, g.alm_nombre      AGENCIA,
-- DATOS DEL CLIENTE
k.cli_codigo       CODIGO,
k.cli_nombre       NOMBRE_RAZON,
k.cli_ruc_cedula   RUC_CEDULA,
RUT_ID            RUTA,
-- DATOS DE LA FACTURA
-- producto
GPR_NOMBRE         MERCADERIA,
C.PRO_ID           ID,
C.PRO_NOMBRE       PRODUCTO,
--z.fac_unidad,
I.umd_id           UNIDAD_P_COMERCIAL,
y.umd_id           UNIDAD_INDIVISIBLE_COMERCIAL,
LIQ.CCO_CADAGENTE  CCO_CADAGENTE,
A.CCO_AGENTE,
sum((B.DFAC_TOTAL)) * DECODE(A.CCO_TRANSACC, 6, -1, 1) SUBTOTAL,

```

```

sum(decode((NVL(D.TOT_SUBTOTAL,0)+NVL(D.TOT_SUBTOT_0,0)),0,0,(nvl(B.
DFAC_DSCITEM,0)))) * DECODE(A.CCO_TRANSACC, 6, -1, 1) DESCUENTO,
sum(decode(NVL(B.dfacs_grabaiva,0),0,0,((NVL(B.DFAC_TOTAL,0)*NVL(D.TO
T_TIMPUESTO,0))/NVL(D.TOT_SUBTOTAL,0)))) * DECODE(A.CCO_TRANSACC, 6,
-1, 1) IVA,
sum((
(B.DFAC_TOTAL) -
(decode((NVL(D.TOT_SUBTOTAL,0)+NVL(D.TOT_SUBTOT_0,0)),0,0,(nvl(B.DFA
C_DSCITEM,0)))))) * DECODE(A.CCO_TRANSACC, 6, -1, 1)
NETO_SIN_IVA,
sum((
(B.DFAC_TOTAL) -
(decode((NVL(D.TOT_SUBTOTAL,0)+NVL(D.TOT_SUBTOT_0,0)),0,0,(nvl(B.DFA
C_DSCITEM,0)))))) * DECODE(A.CCO_TRANSACC, 6, -1, 1) +
sum(decode(NVL(B.dfacs_grabaiva,0),0,0,((NVL(B.DFAC_TOTAL,0)*NVL(D.TO
T_TIMPUESTO,0))/NVL(D.TOT_SUBTOTAL,0)))) * DECODE(A.CCO_TRANSACC, 6,
-1, 1)
TOTAL,
-- CANTIDADES
sum(B.DFAC_CDIGITADA) * DECODE(A.CCO_TRANSACC, 6, -1, 1)
CANTIDAD_DIGITADA,
sum((B.DFAC_CANTIDAD / z.facs_factor)) * DECODE(A.CCO_TRANSACC, 6, -
1, 1) CANTIDAD_P_COMERCIAL,
sum(B.DFAC_CANTIDAD) * DECODE(A.CCO_TRANSACC, 6, -1, 1)
CANTIDAD_INDIVISIBLE_COMERCIAL
FROM CCOMPROBA A, ccomproba liq, DFACTURA B, PRODUCTO C, TOTAL D, FACTOR
E, almacen g, umedida i, cliente k,
factor x, umedida y, factor z, RUTA, GPRODUCTO, LISTAPRE LPV,
CCOMFAC
WHERE
a.cco_estado in (2,3)
AND A.CCO_TIPODOC = 26
AND B.DFAC_EMPRESA=A.CCO_EMPRESA
AND B.DFAC_CFAC_COMPROBA=A.CCO_CODIGO
-- CCOMFAC
AND A.CCO_CODIGO = CFAC_CCO_COMPROBA(+)
AND A.CCO_EMPRESA = CFAC_EMPRESA(+)
-- PRODUCTO
AND C.PRO_EMPRESA=B.DFAC_EMPRESA
AND C.PRO_CODIGO=B.DFAC_PRODUCTO
AND NVL(B.DFAC_COMBO,0) != 1
-- TOTAL
AND D.TOT_CCO_COMPROBA=A.CCO_CODIGO
AND D.TOT_EMPRESA=A.CCO_EMPRESA
-- FACTOR EN LA UNIDAD DIGITADA
AND E.FAC_UNIDAD(+)=B.DFAC_UDIGITADA
AND E.FAC_PRODUCTO(+)=B.DFAC_PRODUCTO
AND E.FAC_EMPRESA(+)=B.DFAC_EMPRESA
-- AGENCIA
and g.alm_codigo=decode(a.cco_adeestino,268, -- PARA LA AGENCIA MACAS
LE UNE A CUENCA
1,
DECODE(A.CCO_CODCLIPRO, 39423, 270, a.cco_adeestino))
-- ARPI AVILA LE CARGA A SANTO DOMINGO
and g.alm_empresa=a.cco_empresa
-- FACTOR UNIDAD DEL PRODUCTO
and x.facs_unidad(+)=c.pro_unidad
and x.facs_producto(+)=pro_codigo
and x.facs_empresa(+)=pro_empresa
-- UNIDAD EL PRODUCTO

```

```

and y.umd_codigo=c.pro_unidad
and y.umd_empresa=c.pro_empresa
-- FACTOR UNIDAD POR OMISION PARA LA VENTA
and z.fac_default(+) = 1
and z.fac_producto(+) = c.pro_codigo
and z.fac_empresa(+) = c.pro_empresa
-- UNIDAD POR OMISION PARA LA VENTA
and i.umd_codigo(+) = z.fac_unidad
and i.umd_empresa(+) = z.fac_empresa
-- CLIENTE
and k.cli_codigo=a.cco_codclipro
and k.cli_empresa=a.cco_empresa
-- LISTA PRECIO VENTA
and cfac_lista_precios = lpv.lpr_codigo(+)
and cfac_empresa = lpv.lpr_empresa(+)
-- RUTA
and cli_ruta = rut_codigo(+)
and cli_empresa = rut_empresa(+)
-- GRUPO PRODUCTO
AND PRO_GPRODUCTO = GPR_CODIGO
AND PRO_EMPRESA = GPR_EMPRESA
-- CON LA LIQUIDACION
AND A.CCO_CIE_COMPROBA = LIQ.CCO_CODIGO(+)
AND A.CCO_EMPRESA = LIQ.CCO_EMPRESA(+)
group by
a.cco_empresa,a.cco_codigo,a.cco_fecha,a.cco_periodo,a.cco_mes,a.cco
_transacc,
ast_gen.numero_comprobante(a.cco_empresa,a.cco_codigo),
ast_gen.numero_comprobante(LIQ.cco_empresa,LIQ.cco_codigo),
A.CCO_CIE_COMPROBA,
lpv.lpr_nombre, a.cco_adeestino, g.alm_nombre,
k.cli_codigo,k.cli_nombre,k.cli_ruc_cedula,
RUT_ID,
GPR_NOMBRE,
C.PRO_ID,C.PRO_NOMBRE,
I.umd_id,y.umd_id,
liq.cco_cadagente,
a.cco_agente

union

SELECT
-- GENERALES DE LA FACTURA
a.cco_empresa      EMPRESA,
a.cco_codigo       CCO_CODIGO,
ast_gen.numero_comprobante(a.cco_empresa,a.cco_codigo)
NUMERO_COMPROBANTE,
a.cco_fecha        FECHA,
a.cco_periodo      PERIODO,
a.cco_mes          MES,
a.cco_cie_comproba,
ast_gen.numero_comprobante(LIQ.cco_empresa,LIQ.cco_codigo)
LIQUIDACION,
-- lista de precios con la que se realizo la venta
lpv.lpr_nombre LISTA_PRECIOS_VENTA,
-- agencia
a.cco_adeestino,
g.alm_nombre      AGENCIA,
-- DATOS DEL CLIENTE
k.cli_codigo      CODIGO ,

```

```

k.cli_nombre      NOMBRE_RAZON,
k.cli_ruc_cedula  RUC_CEDULA,
RUT_ID           RUTA,
-- DATOS DE LA FACTURA
-- producto
GPR_NOMBRE       MERCADERIA,
P.PRO_ID         ID,
P.PRO_NOMBRE     PRODUCTO,
--z.fac_unidad,
I.umd_id         UNIDAD_P_COMERCIAL,
y.umd_id         UNIDAD_INDIVISIBLE_COMERCIAL,
LIQ.CCO_CADAGENTE CCO_CADAGENTE,
A.CCO_AGENTE,
--TOTALES DE LA FACTURA
sum(DECODE(NVL(P.PRO_TIPO_COMBO,0),2,(DFAC_CANTIDAD*COM_CANTIDAD*FAC
.FAC_FACTOR*0.01),
((DFC_PDIGITADO*DFAC_CDIGITADA)-NVL(DFC_DESCUENTO,0)))) *
DECODE(A.CCO_TRANSACC, 6, -1, 1) SUBTOTAL,
sum(DECODE((NVL(D.TOT_DESCUENTO1,0)+NVL(D.TOT_DESCUENTO2,0)),0,0,DEC
ODE(NVL(P.PRO_TIPO_COMBO,0),2,(DFAC_CANTIDAD*COM_CANTIDAD*FAC.FAC_FAC
TOR*0.01),
(((DFC_PDIGITADO*DFAC_CDIGITADA)-
NVL(DFC_DESCUENTO,0))*((NVL(DFAC_DSCITEM,0)-
(NVL(dfcdscitem,0)*dfac_cdigitada))/DFAC_TOTAL)))) *
DECODE(A.CCO_TRANSACC, 6, -1, 1) DESCUENTO,
sum(decode(nvl(p.pro_tipo_combo,0),2,0,decode(NVL(B.dfacsgrabaiva,0)
,0,0,((NVL(((DFC_PDIGITADO*DFAC_CDIGITADA)-
NVL(DFC_DESCUENTO,0)))+(NVL(DFC_DSCITEM,0)*DFAC_CDIGITADA),0)*NVL(D.T
OT_TIMPUESTO,0))/NVL(D.TOT_SUBTOTAL,0)))) * DECODE(A.CCO_TRANSACC,
6, -1, 1) IVA,
sum((
DECODE(NVL(P.PRO_TIPO_COMBO,0),2,(DFAC_CANTIDAD*COM_CANTIDAD*FAC.FAC
_FACTOR*0.01),(((DFC_PDIGITADO*DFAC_CDIGITADA)-
NVL(DFC_DESCUENTO,0)))) -
(DECODE((NVL(D.TOT_DESCUENTO1,0)+NVL(D.TOT_DESCUENTO2,0)),0,0,DECODE
(NVL(P.PRO_TIPO_COMBO,0),2,(DFAC_CANTIDAD*COM_CANTIDAD*FAC.FAC_FAC
TOR*0.01),
(((DFC_PDIGITADO*DFAC_CDIGITADA)-
NVL(DFC_DESCUENTO,0))*((NVL(DFAC_DSCITEM,0)-
(NVL(DFC_DSCITEM,0)*DFAC_CDIGITADA))/DFAC_TOTAL)))))) *
DECODE(A.CCO_TRANSACC, 6, -1, 1)
NETO_SIN_IVA,
sum((
DECODE(NVL(P.PRO_TIPO_COMBO,0),2,(DFAC_CANTIDAD*COM_CANTIDAD*FAC.FAC
_FACTOR*0.01),(((DFC_PDIGITADO*DFAC_CDIGITADA)-
NVL(DFC_DESCUENTO,0)))) -
(DECODE((NVL(D.TOT_DESCUENTO1,0)+NVL(D.TOT_DESCUENTO2,0)),0,0,DECODE
(NVL(P.PRO_TIPO_COMBO,0),2,(DFAC_CANTIDAD*COM_CANTIDAD*FAC.FAC_FAC
TOR*0.01),
(((DFC_PDIGITADO*DFAC_CDIGITADA)-
NVL(DFC_DESCUENTO,0))*((NVL(DFAC_DSCITEM,0)-
(NVL(DFC_DSCITEM,0)*DFAC_CDIGITADA))/DFAC_TOTAL)))))) *
DECODE(A.CCO_TRANSACC, 6, -1, 1) +
sum(decode(nvl(p.pro_tipo_combo,0),2,0,decode(NVL(B.dfacsgrabaiva,0)
,0,0,((NVL(((DFC_PDIGITADO*DFAC_CDIGITADA)-
NVL(DFC_DESCUENTO,0)))+(NVL(DFC_DSCITEM,0)*DFAC_CDIGITADA),0)*NVL(D.T
OT_TIMPUESTO,0))/NVL(D.TOT_SUBTOTAL,0)))) * DECODE(A.CCO_TRANSACC,
6, -1, 1)
TOTAL,
-- CANTIDADES

```

```

sum((B.DFAC_CDIGITADA*COM_CANTIDAD)) * DECODE(A.CCO_TRANSACC, 6, -1,
1) CANTIDAD_DIGITADA,
sum(((B.DFAC_CANTIDAD*COM_CANTIDAD*FAC.FAC_FACTOR) / z.fac_factor))
* DECODE(A.CCO_TRANSACC, 6, -1, 1) CANTIDAD_P_COMERCIAL,
sum((B.DFAC_CANTIDAD*COM_CANTIDAD*FAC.FAC_FACTOR)) *
DECODE(A.CCO_TRANSACC, 6, -1, 1) CANTIDAD_INDIVISIBLE_COMERCIAL
FROM CCOMPROBA A, ccomproba liq, DFACTURA B, PRODUCTO C, TOTAL D, FACTOR
E, almacen g, umedida i, cliente k,
factor x, umedida y, factor z, RUTA,
GPRODUCTO, LISTAPRE LPV, CCOMFAC, PRODUCTO P, COMPONE, factor
fac, dfaccombo
WHERE a.cco_estado in (2,3)
AND A.CCO_TIPODOC = 26
AND B.DFAC_CFAC_COMPROBA=A.CCO_CODIGO
AND B.DFAC_EMPRESA=A.CCO_EMPRESA
-- CCOMFAC
AND A.CCO_EMPRESA = CFAC_EMPRESA(+)
AND A.CCO_CODIGO = CFAC_CCO_COMPROBA(+)
-- PRODUCTO
AND C.PRO_CODIGO=B.DFAC_PRODUCTO
AND C.PRO_EMPRESA=B.DFAC_EMPRESA
-- COMPONENTE
AND NVL(B.DFAC_COMBO,0) = 1
AND C.PRO_CODIGO = COM_PRODUCTO
AND C.PRO_EMPRESA = COM_EMPRESA
AND COM_COMPONENTE = P.PRO_CODIGO
AND COM_EMPRESA = P.PRO_EMPRESA
AND COM_COMPONENTE = FAC.FAC_PRODUCTO
AND COM_UNIDAD = FAC.FAC_UNIDAD
AND COM_EMPRESA = FAC.FAC_EMPRESA
AND B.DFAC_CFAC_COMPROBA=dfc_dfac_comproba
and b.dfac_empresa = dfc_empresa
and b.dfac_secuencia = dfc_dfac_secuencia
and com_componente = dfc_producto
-- TOTAL
AND D.TOT_CCO_COMPROBA=A.CCO_CODIGO
AND D.TOT_EMPRESA=A.CCO_EMPRESA
-- FACTOR EN LA UNIDAD DIGITADA
AND E.FAC_UNIDAD(+)=B.DFAC_UDIGITADA
AND E.FAC_PRODUCTO(+)=B.DFAC_PRODUCTO
AND E.FAC_EMPRESA(+)=B.DFAC_EMPRESA
-- AGENCIA
and g.alm_codigo=decode(a.cco_adeestino,268, -- PARA LA AGENCIA MACAS
LE UNE A CUENCA
1,
DECODE(A.CCO_CODCLIPRO, 39423, 270, a.cco_adeestino))
-- ARPI AVILA LE CARGA A SANTO DOMINGO
and g.alm_empresa=a.cco_empresa
-- FACTOR UNIDAD DEL PRODUCTO
and x.fac_unidad(+)=p.pro_unidad
and x.fac_producto(+)=p.pro_codigo
and x.fac_empresa(+)=p.pro_empresa
-- UNIDAD EL PRODUCTO
and y.umd_codigo=c.pro_unidad
and y.umd_empresa=c.pro_empresa
-- FACTOR UNIDAD POR OMISION PARA LA VENTA
and z.fac_default(+) = 1
and z.fac_producto(+)=p.pro_codigo
and z.fac_empresa(+)=p.pro_empresa
-- UNIDAD POR OMISION PARA LA VENTA

```

```

and i.umd_codigo(+) = z.fac_unidad
and i.umd_empresa(+) = z.fac_empresa
-- CLIENTE
and k.cli_codigo = a.cco_codclipro
and k.cli_empresa = a.cco_empresa
-- LISTA PRECIO VENTA
and cfac_lista_precios = lpv.lpr_codigo(+)
and cfac_empresa = lpv.lpr_empresa(+)
-- RUTA
and cli_ruta = rut_codigo(+)
and cli_empresa = rut_empresa(+)
-- GRUPO PRODUCTO
AND P.PRO_GPRODUCTO = GPR_CODIGO
AND P.PRO_EMPRESA = GPR_EMPRESA
-- CON LA LIQUIDACION
AND A.CCO_CIE_COMPROBA = LIQ.CCO_CODIGO(+)
AND A.CCO_EMPRESA = LIQ.CCO_EMPRESA(+)
group by
a.cco_empresa, a.cco_codigo, a.cco_fecha, a.cco_periodo, a.cco_mes, a.cco
_transacc,
ast_gen.numero_comprobante(a.cco_empresa, a.cco_codigo),
ast_gen.numero_comprobante(LIQ.cco_empresa, LIQ.cco_codigo),
A.CCO_CIE_COMPROBA,
lpv.lpr_nombre, a.cco_adeestino, g.alm_nombre,
k.cli_codigo, k.cli_nombre, k.cli_ruc_cedula, RUT_ID,
GPR_NOMBRE, P.PRO_ID, P.PRO_NOMBRE, p.pro_tipo_combo,
I.umd_id, y.umd_id,
liq.cco_cadagente, a.cco_agente)
group by EMPRESA, CCO_CODIGO, FECHA, PERIODO, MES,
NUMERO_COMPROBANTE, LIQUIDACION, CCO_CIE_COMPROBA,
LISTA_PRECIOS_VENTA,
CCO_ADESTINO,
AGENCIA,
CODIGO , NOMBRE_RAZON, RUC_CEDULA, RUTA,
MERCADERIA,
ID, PRODUCTO,
UNIDAD_P_COMERCIAL, UNIDAD_INDIVISIBLE_COMERCIAL,
CCO_CADAGENTE, CCO_AGENTE;

create or replace view web_agente_almacen as
select age_empresa, age_codigo, age_nombre,
alm_codigo, alm_id,
(select max(pve_id) from PUNTOVENTA
where alm_codigo = pve_almacen
and alm_empresa = pve_empresa
and nvl(pve_autoimpreso, 0) = 1) as pve_id,
bod_codigo
from agente, almacen, bodega
where age_empresa = 2
and nvl(age_inactivo, 0) = 0
and age_almacen = alm_codigo
and age_empresa = alm_empresa
and alm_bodega = bod_codigo
and alm_empresa = bod_empresa;

create or replace view web_producto_stock as
select bod_empresa, bod_codigo, pro_codigo, pro_nombre,
ast_saldos.producto_pedidos(bod_empresa, pro_codigo, bod_codigo,

```



```
                                0, sysdate,0) as saldo  
from bodega, producto  
where bod_empresa = 2  
and nvl(bod_inactivo,0) = 0  
and bod_empresa = pro_empresa  
and nvl(pro_inactivo,0) = 0;
```

c) Formas

Lista las Notas de Entrega pendientes por facturar

A S S I S T Software® Versión EMPRESARIAL - [LISTADO DE NOTAS DE ENTREGA REALIZADAS EN MAPA FACTURAR]

Acción Edición Búsquedas Bloque Registro Campo Window

Programa : INV_LISTA_NET_V **CORPORACION ECUATORIANA DE LIC** Fecha : 05-MAY-2011 15:39 Prg: 3
Módulo : COMERCIALIZACION AGENCIA CUENCA Usuario : DATA_USR Usr: 1

CRITERIOS DE CONSULTA

Concepto	Número Comprobante	Bodega	Transacción	Fecha Desde	Fecha Hasta	
			VENTA	01-JAN-2011	31-DEC-2011	Mostrar todo

LISTADO DE COMPROBANTES

	Comprobante	Referencia	Fecha	Concepto	Estado	Estado Despacho
☒	Ver ...					
☐	Ver ...					
☐	Ver ...					
☐	Ver ...					
☐	Ver ...					
☐	Ver ...					
☐	Ver ...					
☐	Ver ...					
☐	Ver ...					
☐	Ver ...					
☐	Ver ...					
☐	Ver ...					
☐	Ver ...					
☐	Ver ...					
☐	Ver ...					
☐	Ver ...					
☐	Ver ...					
☐	Ver ...					
☐	Ver ...					
☐	Ver ...					
☐	Ver ...					
☐	Ver ...					
☐	Ver ...					
☐	Ver ...					

Facturar

Creación:
Modificación:

ASIST Software® Versión EMPRESARIAL - [FACTURACION DESDE NOTAS DE ENTREGA REALIZADAS EN WEB]

Acción Edición Búsquedas Bloque Registro Campo Window

Programa: FAC_EGFRFACTURA_T **EMBOTELLADORA AZUAYA S.A.** **Fecha:** 05-MAY-2011 15:50 **Prgr:** 3
Módulo: COMERCIALIZACION **EASA CUENCA** **Usuario:** DATA_USR **Usr:** 1

Sigla: FA FACTURAS Fecha: 01-05-2011
Almacén: 001 EASA CUENCA Código: 0
Comprobante: -0-0- **Secuencia:** 36298
Autorización SRI: 1109199227

Vendedor: VAC_MACAS VACANTE MACAS Bodega: 28 MAC- PRODUCTO TERMINADO
Cliente: CUE00007 CELYASA Liquidación No: Ver Evento

L.Precios: 1 DISTRIBUIDOR Impuesto: 01 I.V.A. VENTAS Estado Cta: CELYASA
Política: 002 90 DIAS Dcto %: 0.00 Pag.Contado%: 0.00 Nro. Pagos: 1 Saldo Cliente: 672,128.15
Días Plazo: 90 Ch.Protestados: 0.00

Línea	Producto	Cantidad	Unidad	Dcto%	ICE	Precio	Total	IVA
1	1113 CRISTAL MELON 15° 375E	165	U	0.00	1467	0.7779	128.35	
2	1030 CRISTAL DURAZNO 20° 375C	114	U	0.00	1954	0.8496	96.85	
3	1029 WHISKY "PIPER'S CLAN"	85	U	0.00	0	5.8550	497.68	
4	1018 CRISTAL SECO 34° 375C	2,608	U	0.00	3321	1.0100	2,634.08	
5	1121 CRISTAL DURAZNO 20° 375E	1,947	U	0.00	1954	0.8496	1,654.17	
6	1119 CRISTAL SECO 34° 375E	1,552	U	0.00	3321	1.0100	1,567.52	
7	1101 CRISTAL DURAZNO 20° 750E	982	U	0.00	3908	1.5692	1,540.95	
8	1118 CRISTAL AGUARDIENTE 1960 375E	600	U	0.00	3321	1.0521	631.26	
9	1099 CRISTAL SECO 34° 750E	560	U	0.00	6642	1.8908	1,058.85	
10	1107 CRISTAL AGUARDIENTE 1960 375C	309	U	0.00	3321	1.0521	325.10	

Producto: CRISTAL MELON 15° 375E Subtotal 0%: 0.00 Subtotal 12%: 10,857.33
Factor Conversión: 1 Desc1: Descuento1: 0.00
Cantidad convertida: 165.0000 Unidad: U Desc2: Descuento2: 0.00
Precio convertido: 0.7779 Saldo Disponible: IVA 12.00: 1,302.88
Saldo Disponible del Producto Actual Transporte: 0.00
Total: 12,160.21

Anular Secuencia Dejar en Proceso

Suman: DOCE MIL CIENTO SESENTA 21/100

Creación: Estado: EN PROCESO Tipo: CONTABLE Anulado:
Modificación: Módulo: FAC Documento: FACTNE Almacén: 001

Completar datos Clientes:

