INSTITUTO TECNOLÓGICO SUPERIOR “CORDILLERA"

ESCUELA DE SISTEMAS

Proyecto de Grado, previa obtención del título de:
Tecnólogo Analista de Sistemas

TEMA:

DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA DE INVENTARIO Y FACTURACIÓN PARA LA FERRETERÍA GORDÓN

AUTOR:

Báez Diego

TUTOR:

Ing. Rodrigo Cobos

2011

QUITO – ECUADOR

APROBACIÓN DEL TRIBUNAL DE GRADO

Los miembros del Tribunal de Grado designado por el Honorable Concejo de la Escuela de Sistemas, aprueban el trabajo de investigación de acuerdo con las disposiciones reglamentarias emitidas por el Centro de Investigaciones Tecnológicas y Proyectos del Instituto Tecnológico Superior Cordillera” para proyectos de grado de Tecnólogos Analistas de Sistemas: del Sr: DIEGO BAEZ HUERA.

Quito, 21 Septiembre2011

Para constancia firman:

PRESIDENTE

VOCAL1 VOCAL2

APROBACIÓN DEL DIRECTOR DE ESCUELA DE SISTEMAS

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del trabajo sobre el tema: “SISTEMA DE INVENTARIO Y FACTURACION PARA LA FERRETERÍA GORDON”, presentado por el ciudadano: DIEGO BAEZ HUERA, estudiante de la Escuela de Sistemas, considero que dicho informe reúne los requisitos y méritos suficientes para ser sometido a la evaluación por parte del Tribunal de Grado, que el Honorable Consejo de Escuela designe, para su correspondiente estudio y calificación.

Quito, 21 Septiembre del 2011

Ing. Rodrigo Cobos

TUTOR

DECLARACIÓN DE AUTENTICIDAD

El abajo firmante, declara que los contenidos y los resultados obtenidos en el presente proyecto, como requerimiento previo para la obtención del Título de Tecnólogo Analista de Sistemas, son absolutamente originales, auténticos y personales y de exclusiva responsabilidad legal y académica del autor.

Diego Báez Huera

1723467187

AGRADECIMIENTO

Mi gratitud imperecedera al Instituto Tecnológico “Cordillera” Y a sus distinguidos profesores por haber Compartido sus valiosos saberes en Beneficio de mi formación Profesional.

Quito, 21 Septiembre del 2011

Diego Báez Huera

DEDICATORIA

A MIS PADRES

Les dedico este trabajo con todo mi amor y les agradezco todos los sacrificios que hicieron para que pudiera llegar hasta aquí, gracias por confiar en mí, por cuidarme en tantas y tantas ocasiones y sobre todo por darme un ejemplo de vida, los dos son dos son admirables para mí.

Diego Báez Huera

PENSAMIENTO

“Hay que creer que cada uno fue creado para tener éxitos. Hay un método para conseguir el éxito: es la consagración total a nuestro trabajo…”

San Juan Bosco

ÍNDICE GENERAL

PORTADA.....i
APROBACIÓN DEL TRIBUNAL DE GRADO..... ii
APROBACIÓN DEL DIRECTOR DE ESCUELA DE SISTEMAS..... iii
APROBACIÓN DEL TUTOR..... iv
DECLARACIÓN DE AUTENTICIDAD..... v
AGRADECIMIENTO..... vi
DEDICATORIA..... vii
PENSAMIENTO..... viii
ÍNDICE GENERAL..... ix
ÍNDICE DE GRÁFICOS..... xii
ÍNDICE DE TABLAS..... xiv
RESUMEN EJECUTIVO..... xv
CAPÍTULO I..... 1
EL PROBLEMA..... 1
1.1 Planteamiento Del Problema..... 1
1.2 Formulación Del Problema..... 2
1.3 Delimitación Del Problema..... 2
1.4 Objetivos..... 3
1.4.1 Objetivo general..... 3
1.4.2 Objetivos específicos..... 3
1.5 Justificación e Importancia..... 4
1.6 Alcance..... 5
CAPÍTULO II..... 8
MARCO TEÓRICO..... 8

2.1 Antecedentes 8

2.2 Reseña histórica 8

2.3 Marco referencial 11

2.4 Marco legal 25

CAPÍTULO III 29

INVESTIGACIÓN CIENTIFICA 29

3.1 Tipos de Investigación 29

3.2 Métodos de investigación 31

3.3 Herramientas de recolección de información 33

CAPÍTULO IV 40

DESARROLLO DE LA PROPUESTA 40

4.1 Diagnóstico Situacional 40

4.2 Estructura Organizacional 41

4.3 Infraestructura Informática 45

4.4 Descripción de alternativas 47

4.5 Evaluación de la Propuesta 55

4.6 Factibilidad Técnica 59

4.7 Descripción de los Procesos 59

4.8 Descripcion de metodología de desarrollo 70

4.9 Modelo Físico 73

4.10 Diccionario de Datos 77

4.11 Estándares de Programación 86

4.12 Prototipo 98

4.13 Pruebas y Depuración 101

4.14 Instalación 105

4.15 Recopilación y Carga de Datos 111

4.16 Pruebas y Depuración Final en Funcionamiento 111

4.17 Puesta en Marcha del Sistema 112

4.18 Capacitación al Usuario Final 112

4.19 Capacitación al Personal Técnico 113

CAPÍTULO V 114

PRINCIPALES IMPACTOS 114

5.1 Científico. 114

5.2 Educativo. 114

5.3 Técnico. 114

5.4 Tecnológico. 115

5.5 Empresarial. 115

5.6 Social. 115

5.7 Económico. 115

5.8 Conclusiones. 116

5.9 Recomendaciones. 117

CAPÍTULO VI 118

Bibliografía 118

Netgrafía 119

Glosario de Términos 120

ANEXOS 122

ÍNDICE DE GRÁFICOS

Ilustración 1 3
Ilustración 2 18
Ilustración 3 20
Ilustración 4 41
Ilustración 5 42
Ilustración 6 45
Ilustración 7 61
Ilustración 8 62
Ilustración 9 63
Ilustración 10 63
Ilustración 11 64
Ilustración 12 64
Ilustración 13 65
Ilustración 14 66
Ilustración 15 67
Ilustración 16 68
Ilustración 17 69
Ilustración 18 70
Ilustración 19 71
Ilustración 20 76
Ilustración 21 98
Ilustración 22 98
Ilustración 23 99
Ilustración 24 99
Ilustración 25 100
Ilustración 26 101

Ilustración 27 106
Ilustración 28 107
Ilustración 29 107
Ilustración 30 108
Ilustración 31 109
Ilustración 32 109
Ilustración 33 110
Ilustración 34 111
Ilustración 35 123
Ilustración 36 124
Ilustración 37 125
Ilustración 38 126
Ilustración 39 127
Ilustración 40 127
Ilustración 41 128
Ilustración 42 128
Ilustración 43 129
Ilustración 44 129
Ilustración 45 130
Ilustración 46 131
Ilustración 47 132
Ilustración 48 132
Ilustración 49 133
Ilustración 50 133
Ilustración 51 134
Ilustración 52 135
Ilustración 53 135

ÍNDICE DE TABLAS

Tabla 1 13
Tabla 2 46
Tabla 3 46
Tabla 4 47
Tabla 5 49
Tabla 6 51
Tabla 7 53
Tabla 8 55
Tabla 9 56
Tabla 10 57
Tabla 11 57
Tabla 12 58
Tabla 13 81
Tabla 14 83
Tabla 15 85
Tabla 16 86
Tabla 17 90
Tabla 18 92
Tabla 19 105
Tabla 20 105
Tabla 21 113

RESUMEN EJECUTIVO

CAPÍTULO I

Hace referencia al principal problema de la empresa para saber el objeto de estudio, planteando objetivos generales y objetivos específicos que se llevarán a cabo, según eso se analizará el alcance del proyecto.

CAPÍTULO II

Describe en su totalidad a la empresa, como se encuentra actualmente constituida, examina los antecedentes de la Ferretería Gordon, para el proceso de toda la información, siendo esta técnica u operativa, realizando diagramas de toda las actividades y un marco conceptual del vocabulario utilizado en la tesis.

CAPÍTULO III

Describe las principales metodologías de investigación, así como también las técnicas de recolección de información.

CAPÍTULO IV

Está enfocado al diagnóstico situacional de la empresa, su infraestructura de software y hardware, se plantea una alternativa conveniente para la empresa enfocándose en la problemática de la misma.

CAPÍTULO V

Describe los principales impactos que han tenido el desarrollo de proyecto, las conclusiones y recomendaciones para el mejor funcionamiento de la empresa, así como también el recurso humano.

CAPÍTULO I

EL PROBLEMA

1.1 Planteamiento Del Problema

En la ferretería GORDÓN, no existe un control en el inventario, las pérdidas de capital son grandes; y el descontrol que genera el manejo manual de facturación ocasiona la perdida de estabilidad económica, social, cultural, etc. de la organización. Además no se tiene un manual de procedimientos para llevar a cabo la realización de los pedidos a proveedores, lo que ocasiona la compra de productos de mala calidad.

El constante crecimiento en el volumen de información manejada y el número de operaciones y procedimientos para la realización de las compras y ventas, realizadas principalmente en forma manual; ha dificultado la gestión administrativa, ya que no se posee información completa, oportuna y actualizada.

En la visita realizada a las instalaciones de la empresa se observó que los trabajadores de la misma no cuentan con ningún sistema para el desempeño de su trabajo, todos los procesos que se realizan se hacen manualmente, lo cual resulta engorroso e ineficiente debido a que no se controlan los flujos de datos con detalles ni los procesos que se realizan sobre ellos, esto conlleva a la probabilidad de que se pueda modificar y/o perder alguna información para el negocio, además de esto no se controla la cantidad de materiales, clientes con los que se trabaja, ni los proveedores, es decir, no se tiene registro alguno de las transacciones.

El inventario de mercancía de la empresa es realizado manualmente mediante el uso de tarjetas donde se escriben las entradas y salidas de las facturas de clientes, y también de aquella mercancía comprada por el negocio, estas tarjetas sufren de deterioro por el paso del tiempo y el mismo uso.

No cuentan con un sistema de inventario automatizado, ni tampoco con un inventario detallado de los productos, lo cual afecta al crecimiento y desarrollo económico de la empresa ya que hace falta más control en lo que se tiene en existencia en el local.

Este sistema administrativo no controla la facturación tanto de las ventas como de las compras, además de no mantener actualizado el inventario de producto, no establece criterios de inventario para la adquisición de productos cuya cantidad en existencia este por agotarse.

La base de toda empresa comercial es la compra y ventas de bienes y servicios; de aquí viene la importancia del manejo de inventario por parte de la misma. El manejo manual del inventario no permite a la empresa mantener el control oportuno, así como también de no conocer al final del periodo de su actividad, un estado confiable de la situación económica de la empresa.

1.2 Formulación Del Problema

¿Con la implementación de un sistema de facturación e inventario mejoraran los procesos de despacho y venta de mercaderías en la Ferretería Gordon?

1.3 Delimitación Del Problema

El local Ferretería Gordon con un aproximado de 5000 clientes, está ubicado en la Calle García Moreno N° 14, en la Parroquia él Quinche, Cantón Quito, Provincia de Pichincha, República del Ecuador.

Ubicación Ferretería Gordon
Ilustración 1

Fuente: Internet

1.4 Objetivos

1.4.1 Objetivo general

Desarrollar e implementar un sistema de facturación e inventario para la Ferretería Gordona fin de mejorar la solvencia y eficacia de sus procesos.

1.4.2 Objetivos específicos

- Analizar los procesos y reglas del negocio.
- Modelar los procesos y reglas del negocio.
- Programar el código fuente.
- Realizar pruebas de banco y de campo.
- Implementar y capacitar el manejo de aplicativo.

1.5 Justificación e Importancia

Con la automatización del inventario de mercancía de la Ferretería Gordon se mejoraran los procesos de entradas y salidas de las facturas de clientes, y también de aquella mercancía comprada por el negocio, esto ayudará a que la información este siempre actualizada y sin alteraciones para la toma de decisiones.

Con un sistema de inventario automatizado, se obtendrá un inventario detallado de los productos, lo cual ayuda al crecimiento y desarrollo económico de la empresa ya que tendrá mayor control de lo que se tiene en existencia en el local.

De acuerdo a lo anteriormente expuesto se puede deducir claramente la imperiosa necesidad de implementar un sistema; el cual implica una total atención al manejo de las reglas del negocio y procesos de la empresa, la agilidad control y pertinencia de la información se verá reflejada en la velocidad de respuesta que se obtenga a través de las interfaces de salidas.

Cabe recalcar que todo este proceso de automatización ayudará a que toda la operatividad de la entidad se halle enmarcado dentro de estándares, rapidez, seguridad solides del manejo de información que sobre todas las cosas debemos dar soluciones y mantener la satisfacción del cliente que es la razón de ser en el desarrollo e implementación del proyecto SINYFAC.

1.6 Alcance

Este Sistema Administrativo de Facturación e Inventario (SINYFAC) controla la facturación tanto de las ventas como de las compras, además de mantener actualizado el inventario de producto, estableciendo criterios de inventario para la adquisición de productos cuya cantidad en existencia este por debajo de los valores mínimos permitidos.

- El control y elaboración de facturación para ventas e ingreso a inventario.
- El control y elaboración de facturas para devoluciones de ventas e ingresos a inventario.
- Reportes de estado de inventarios, ventas, compras, Clientes, proveedores y existencias.
- Acceso rápido a la información sobre los productos, proveedores, cliente, ventas y compras.
- El manejo del precio en función de márgenes de utilidad o valores absolutos.
- El manejo de porcentajes de descuentos independientes en cada producto a facturar
- El manejo del impuesto cargado a la facturación tanto de compras como de ventas.
- El control de códigos de productos, manteniendo la unicidad de los mismos.

El sistema de facturación e inventario, además de lo anterior, cuenta con la conexión a otro módulo de trabajo como inicio y cierre de caja, mantenimiento y respaldos de la base de datos.

Elaboración de reportes:

Los reportes pueden hacerse por:

Compras: Muestra las compras en un periodo cualquiera indicando los totales por factura. Se deben introducir las fechas de inicio y fin de periodo. Reporte de gestión de compras.

Factura: Muestra el detalle de una factura de compra: renglones, total, impuesto, cliente, etc. Se debe tener el número de factura.

Existencia: Muestra la existencia de inventarios según criterio de selección: existencias totales, existencias críticas, existencias por código, existencias por descripción del producto

Proveedores: Muestra información sobre proveedores según criterio de búsqueda: Nro., nombre, todos.

Clientes: Muestra información sobre clientes según criterio de búsqueda: Nro., nombre, todos.

- Administración de Proveedores.

  Esto permitirá crear, modificar y borrar proveedores. Si el proveedor ya ha sido asignado a productos, dicho proveedor no podrá ser borrado.

- Administración de Líneas de Producto.

  Esto permite ingresar nuevas líneas de productos, modificar y borrar líneas. Si la línea de producto ya ha sido asignada a productos, dicho línea no podrá ser borrada.

- Administración de Productos.

  Esto permitirá crear, modificar y borrar productos. El programa debe controlar que cuando el usuario ingresa un nuevo producto y va asignar el código de línea y de proveedor, de lo contrario debe indicar un mensaje al usuario a través de la pantalla.

- Administración de Compras de Productos.

  Este proceso permitirá hacer entrada a inventarios de productos. Cuando se realiza la entrada esta debe quedar registra en un arreglo, a su vez debe aumentar el campo de existencia en el arreglo de productos.

- Administración de Ventas de Productos.

  Esto permite realizar una salida de inventario, en casos de que el producto se dañe o se deba de devolver al proveedor por estar defectuoso, este proceso deberá de restar el campo de existencia en el arreglo de productos. Estas salidas deben quedar almacenadas en un arreglo (array).

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes

El desarrollo y análisis de la investigación enriquecen de un modo u otro el logro de muchos objetivos, los cuales representan aportes significativos a la empresa, la comunidad y a uno mismo como analista o persona. Aportes tales como la automatización, la cual contempla diversos aspectos como lo son la rapidez, eficiencia y mejor administración de los recursos disponibles esto conlleva sin duda al logro de muchas metas establecidas y a un mejor posicionamiento a nivel geográfico, de modo así que el usuario se sienta satisfecho por una mejor atención, lo cual es sin duda un aporte del sistema a la empresa.

Como analista o persona el ver que un sistema analizado, desarrollado e implementado ofrezca frutos provechosos a una empresa como a la sociedad es sin duda un logro, el cual equilibrado con otros beneficios o aportes logran crear en el analista una mentalidad más abierta, concreta, siempre encaminada a ayudar y no a sacar beneficios lucrativos por una profesión determinando así el cumplimiento de muchas metas o actividades, de modo que tanto las sociedad como la empresa den el 100 % de sus desempeños.

2.2 Reseña histórica

Ferretería Gordon surge como una Sucursal de Lavadora Gordon, empresa que funciona desde 1986 en la Calle 75, Vía principal del Sector de el Quinche, populoso sector del Este de Quito, donde, producto de una intensa expansión urbana, se han venido levantando

urbanizaciones y barrios populares, conformados por casas y edificios de apartamentos de baja altura.

El local de Ferretería Gordon tenía pocas posibilidades de expandirse, por lo que sus accionistas adquirieron en la misma calle 75, después del Colegio Cardenal de la Torre en el Quinche, un terreno donde construyeron una casa que les permitió almacenar materiales de construcción.

La compañía fue registrada como tal en 1988, pero comenzó operaciones en Agosto 1996, en su propio Local de un piso, construido sobre el mencionado terreno. Sucesivas ampliaciones han dado como resultado el edificio actual, donde tienen su aposento, no solo las oficinas de venta y el área de almacenamiento y despacho, sino incluso las oficinas administrativas del Grupo.

En un principio, la Ferretería Gordon comenzó vendiendo solo materiales de construcción, pero actualmente, venden todas las líneas que puede ofrecer una moderna ferretería en Ecuador, incluyendo artículos de plomería y electricidad.

Dentro de los productos más vendidos por nuestra compañía, se pueden destacar los siguientes:

- Arena blanca, roja y frijolito.
- Bloques rojos y blancos.
- Cabillas estriadas.
- Cal.
- Cemento Gris.
- Conexiones de Aguas Blancas.
- Láminas de Zinc.
- Láminas pulidas.

- Lámparas.
- Materiales de Electricidad.
- Materiales de Plomería.
- Pego.
- Pintura.
- Tubos pulidos.

Durante todo este tiempo, la empresa ha podido crecer debido al apoyo que le han dado las familias que han levantado o ampliado sus casas en los barrios y urbanizaciones, además de las Granjas cercanas que ofrecen recreación deportiva durante los fines de semana.

A lo largo de estos años, se ha ido especializado en la venta de productos ferreteros de todo tipo, habiéndose constituido en la principal alternativa para todos esos barrios de la zona Este de la ciudad de Quito, agrupados bajo las parroquias Ascázubi, Checa, Yaruquí y Pifo, cuyos habitantes ven en la empresa un proveedor seguro con precios solidarios.

Misión

La comercialización de todo tipo de productos del ramo de ferretería, así como la compra y venta de toda clase de materiales para la construcción de viviendas, herramientas y materiales de electricidad destinados a satisfacer las necesidades básicas dentro de la ciudad de Quito, a través del cumplimiento de las normas establecidas, disponiendo de recursos humanos altamente calificados y comprometidos con la organización y la comunidad en general.

Visión

Convertirnos en una compañía ubicada entre las primeras del mercado donde operamos en el ramo de la comercialización de productos ferreteros y de materiales de construcción,

dirigidos a organizaciones y público en general, alcanzando altos estándares de calidad y manteniendo un alto grado de competitividad y orientación de servicio hacia la comunidad, con posibilidades de proyección a otras regiones.

2.3 Marco referencial

Inventarios.

Desde de la antigüedad, se acostumbraba almacenar grandes cantidades de alimentos para ser utilizados en los tiempos de sequía o de calamidades. Es así como surge o nace la necesidad de los inventarios, como una forma de hacer frente a los periodos de escasez. Que le aseguraran la subsistencia de la vida y el desarrollo de sus actividades normales. Esta forma de almacenamiento de todos los bienes y alimentos necesarios para sobrevivir motivó la existencia de los inventarios. Los inventarios son importantes para comerciantes y fabricantes en general, y varían ampliamente entre los distintos grupos. La composición de esta parte del activo es una gran variedad de artículos, y es por eso que se han clasificado de acuerdo a su utilización en los siguientes tipos:

- Inventarios de materia prima.
- Inventarios de producción en proceso.
- Inventarios de productos terminados.
- Inventarios de materiales y suministros.

Para el caso de este sistema, se trabaja en base a inventarios de productos terminados exclusivamente ya que los otros tipos son de uso para empresas de producción.

Los inventarios de productos terminados, comprenden los artículos transferidos por el proveedor hacia el almacén o punto de venta de productos terminados por haber alcanzado su grado de terminación total y que a la hora de la toma física de inventario se encuentren

aun en los almacenes, es decir; los que todavía no han sido vendidos. El nivel de inventario de productos terminados va a depender directamente de las ventas, es decir; su nivel está dado por la demanda.

Gestor de Base de Datos y Herramientas de Desarrollo

Base de Datos

SQL es una herramienta para organizar, gestionar y recuperar datos almacenados en una base de datos informática. El nombre "SQL" es una abreviatura de Structured Query Languaje (Lenguaje de consultas estructurado). Como su propio nombre indica, SQL es un lenguaje informático que se puede utilizar para interaccionar con una base de datos y más concretamente con un tipo específico llamado base de datos relacional. SQL es a la vez un lenguaje fácil de aprender y una herramienta completa para gestionar datos. Las peticiones sobre los datos se expresan mediante sentencias, que deben escribirse de acuerdo con unas reglas sintácticas y semánticas de este lenguaje. Su aprendizaje no solo sirve para esta aplicación sino, también, para todas las existentes en el mercado que soporten este lenguaje ya que es un lenguaje estándar por haberse visto consolidado por el Instituto Americano de Normas (ANSI) y por la Organización de Estándares Internacional (ISO).

MySQL es una de las bases de datos más populares desarrolladas bajo la filosofía de código abierto. La desarrolla y mantiene la empresa MySql AB pero puede utilizarse gratuitamente y su código fuente está disponible.

Características (versión 4.0 en adelante)

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas web con contenido dinámico, justamente por su simplicidad; aquellos elementos faltantes fueron llenados por la vía de las aplicaciones que la utilizan.

Poco a poco los elementos faltantes en MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar:

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

SQL ofrece muchas ventajas comparado con MySQL por ejemplo la capacidad de ejecutar procedimientos almacenados así como también la compatibilidad con base de datos.

Comparación de Microsoft SQL 2000 con otros gestores de base de datos

Características	SQL 2000	MySQL	Oracle
Compatibilidad con bases de datos para dispositivos móviles	✓	×	✓
Servicios de Transferencia de datos(DTS)	✓	×	✓
Soporte de procedimientos almacenados, triggers, etc.	✓	×	✓
Conectividad con diferentes lenguajes de programación.	✓	×	✓
Interface gráfica para el usuario	✓	×	✓
Precio	×	✓	×
Multiplataforma	×	✓	✓

Comparación de Microsoft SQL 2000 con otros gestores de base de datos
Tabla 1
Fuente: Diego Báez

¿Qué es un Sistema de Base de datos?

Un Sistema de Base de Datos es básicamente un sistema computarizado para llevar registros. Es posible considerar a la propia Base de Datos como una especie de armario electrónico para archivar, es decir, un depósito o un contenedor de una colección de archivos de datos computarizados.

Principales componentes de un Sistema de Bases de Datos.

A continuación se mencionarán los cuatro principales componentes en un sistema de base de datos:

- La información
- El equipo
- Los usuarios
- Los programas

La información. En general, la información en la base de datos estará integrada y además será compartida.

El equipo. Se considera que son componentes del equipo del sistema:

Medios de almacenamiento secundario: Dispositivos E/S asociados, Drives, Canal es de E/S, etcétera, Procesador o procesadores y memoria principal.

Los usuarios: Es todo el personal del departamento que requiera usar el sistema de base de datos para implementar, consultar o realizar sus reportes. Se tienen diferentes tipos de usuarios, entre los cuales tenemos a los programadores de aplicaciones; los cuales son los responsables de escribir los programas de aplicación; los usuarios finales, quienes

interactúan con el sistema desde estaciones de trabajo o terminales; y finalmente el Administrador de la Base de Datos (DBA), y es quien administra la base de datos. Los programas. Existe una capa de programas entre la base de datos física misma y los usuarios del sistema: el Sistema de Administración de Base de Datos(DBMS, Data Base Managment System).

El DBMS maneja todas las solicitudes de acceso a la base de datos formuladas por los usuarios.

El Sistema de Administración de Bases de Datos (DBMS)

El software que permite a una o más personas el usar y/o modificar los datos de una base de datos se denomina Administrador de Base de Datos (DBMS).

Maneja todas las solicitudes de acceso a la base de datos formuladas por los usuarios.

Seguridad: no todos los usuarios tienen acceso a todos los datos.

Integridad: cierto tipo de “consistencia” deberá realizarse sobre los atributos y valores de los datos, para evitar la inconsistencia en los datos.

Sincronización: Cuando varios usuarios corren programas que accedan a la base de datos al mismo tiempo, el DBMS deberá dar protección de inconsistencias que puedan resultar de dos operaciones simultáneas a un mismo grupo de datos.

Protección de rupturas y recuperación: facilidades para realizar copias regulares de la base de datos y reconstruirla después de un error de hardware o software.

Uno de sus objetivos más importantes es proporcionar a los usuarios una visión abstracta de los datos, es decir, el sistema esconde ciertos detalles de cómo se almacenan y mantienen los datos, pero sin embargo se deben extraer eficientemente.

El Modelo Relacional

El modelo relacional de las bases de datos representa la tendencia dominante en el mercado actual debido al avance que ha tenido en el campo de las bases de datos principalmente en el modelo relacional, por esta razón la tecnología de bases de datos se orienta hacia los sistemas relacionales, por ende, en ésta parte del capítulo se explicará de manera general los conceptos fundamentales para los sistemas relacionales de bases de datos.

Una base de datos relacional es una base de datos percibida por el usuario como una colección de relaciones normalizadas [Boone 2000].

El modelo relacional se ocupa de tres aspectos:

1. Estructura de datos: El usuario percibe la información de la base de datos como tablas y nada más que tablas. La estructura de datos es la terminología que se utiliza como notación y forma de representar la información en el modelo relacional.

En particular una relación corresponde a lo que comúnmente se le llama Tabla.

Entidad: es cualquier objeto distinguible que ha de representarse en la base de datos.

Una Tupla corresponde a una fila de esa tabla y a un atributo a una columna.

La llave primaria es un identificador único para la tabla; es decir una columna o combinación de columnas con la siguiente propiedad: Nunca existen dos filas de la tabla con el mismo valor en esa columna o combinación de columnas.

Un Dominio es una colección de valores escalares, de los cuales uno o más atributos (columnas) son del mismo tipo.

Un atributo es una característica o valor que se toma (las columnas) dentro de la naturaleza de su dominio.

2. Integridad. Las tablas deben satisfacer ciertas restricciones de integridad, la integridad se basa en dos reglas en el modelo relacional:

Regla de las Entidades: Ningún componente de una relación base puede tener valores nulos.

La base de datos no debe contener valores de llave ajena sin concordancia.

3. Manipulación. Los operadores disponibles para que el usuario manipule estas tablas son operadores que derivan tablas a partir de tablas. En particular, tres de estos operadores son importantes: restringir, proyectar, juntar.

La operación restringir extrae las filas especificadas de una tabla.

La operación proyectar extrae las columnas especificadas de las tablas.

La operación juntar reúne dos tablas con base en valores comunes de al menos una columna en común.

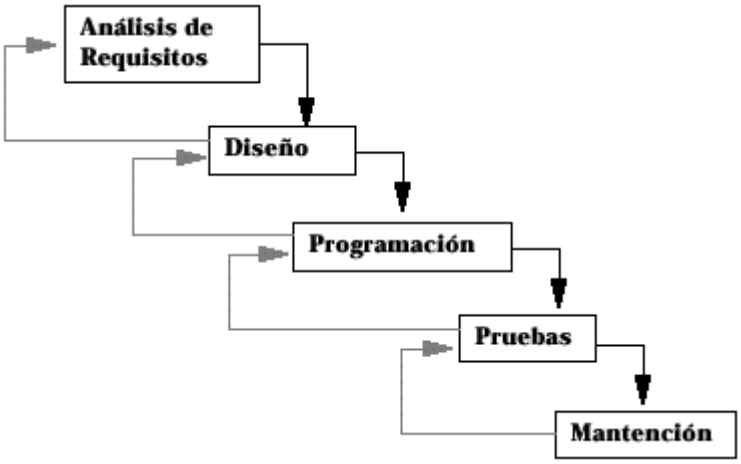
Ingeniería de Software

La Ingeniería del Software se podría definir como el establecimiento y aplicación de principios de la Ingeniería para obtener software. Teniendo en cuenta factores tan importantes como el coste económico, la fiabilidad del sistema y un funcionamiento eficiente que satisfaga las necesidades del usuario.

El modelo del ciclo de vida es un factor principal para conseguir los objetivos buscados, una mala elección del modelo de ciclo de vida puede hacer que se no se retrase el trabajo enormemente o que tengamos una planificación perfecta del trabajo.

Modelo de cascada

En un modelo en cascada un proyecto progresa a través de una secuencia ordenada de pasos que son:



Modelo Cascada
Ilustración 2
Fuente: Internet

- Concepto del software.
- Análisis de requerimientos.
- Diseño global.
- Diseño detallado.
- Codificación y depuración.
- Prueba del sistema.

El modelo contiene una serie de etapas que no se solapan, y el proyecto se va revisando tras cada una de las etapas. Para poder pasar a la siguiente etapa se tiene que haber conseguido todos los objetivos de la etapa anterior, es un proceso secuencial.

Tiene una buena aplicación cuando el problema es estable y cuando se trabaja con metodologías técnicas conocidas. Este modelo será apropiado para la migración de una aplicación o a una versión de mantenimiento bien definida.

Con este modelo se tiene un seguimiento de todas las fases del proyecto y del cumplimiento de todos los objetivos marcados en cada etapa tanto de costes, fechas de entrega y lo más importante que pueden comprobar al final de cada etapa si el proyecto cumple todas las necesidades del usuario.

A su vez esto es un problema ya que si el usuario se da cuenta de que falta una tarea de la empresa en el proyecto una vez pasada esta etapa, el trabajo que hay que realizar se retrasa en fechas de entrega y el coste es mayor. Por lo tanto esto produce un fracaso en la industria ya que es reacio a las modificaciones de última hora.

Por este motivo se puede modificar el modelo en cascada pudiendo pasar de una etapa a la anterior, pero suele ser difícil ya que hay que rehacer la etapa anterior, este modelo es el ciclo de vida del salmón. Por lo tanto este es un modelo poco apropiado para proyectos con fecha de entrega corta, pero su rendimiento puede mejorar notablemente variando el modelo de la cascada pura.

Espiral

Modelo Espiral

Ilustración 3
Fuente: Internet

El modelo de la espiral es un modelo orientado a riesgo que divide el proyecto software en mini proyectos. Cada proyecto se encargará de resolver uno o varios riesgos hasta que estén todos controlados. Una vez que estén los riesgos más importantes controlados se finaliza igual que el ciclo de vida en cascada.

En el ciclo de vida en espiral localizan los riesgos, genera un plan para manejar los y se establece una aproximación a la siguiente iteración. Con cada iteración se produce una aproximación al producto final.

En el modelo en espiral se comienza con una parte pequeña del proyecto y se expande tras reducir los riesgos para la siguiente iteración.

En cada iteración seguimos los siguientes pasos:

- Determinar objetivos, alternativas y límites.

- Identificar y resolver riesgos.
- Evaluar las alternativas.
- Generar entregas de esta iteración, y comprobar que son correctas.
- Planificar la siguiente iteración.
- Si se decide ejecutar la siguiente iteración, hay que establecer un enfoque para ella.

En este modelo las primeras iteraciones son menos costosas y a medida que se avanza aumenta el coste.

Las ventajas de este modelo son:

- Se disminuyen los riesgos.
- Al final de cada iteración se obtienen los puntos de verificación.
- Se obtienen con anterioridad indicaciones de cualquier riesgo insuperable.

Las desventajas de este modelo son:

- Un aumento de costes.
- Es un modelo complicado de llevar a cabo porque exige una gestión concienzuda, atenta y unos conocimientos profundos.

Modelo del desarrollo por prototipos

Los sistemas pueden desarrollarse con métodos y lenguajes de programación convencionales, aunque no tengan todas las características y toques finales de un sistema terminado. Quizás los informes no tengan encabezados, logos, etc., falten controles de entradas y procesamiento. Lo importante es el ensayo, y hallar los requerimientos.

En algunos casos donde el sistema no será utilizado frecuentemente, puede convertirse el prototipo en el sistema terminado, o bien, cuando no son muchos los beneficios que se obtienen.

Razones para desarrollar prototipos de sistemas

Los requerimientos de información no siempre están bien definidos, pueden ser demasiados vagos aún al formular el diseño. En otros casos, es probable que una investigación de sistemas bien llevada, de como resultado un conjunto muy amplio de requerimientos de sistemas, pero construir un sistema que satisfaga a todos ellos quizás necesite del desarrollo de nueva tecnología.

Los prototipos permiten evaluar situaciones extraordinarias donde los encargados de diseñar e implantar sistemas no tienen información ni experiencia, o también donde existen situaciones de riesgo y costos elevados, y aquellas donde el diseño propuesto es novedoso y aún no ha sido probada.

La información obtenida con su uso se aplica en un nuevo diseño que se emplea, otra vez, como prototipo y que revela más información valiosa sobre diseño.

El proceso se repite las veces que sea necesario para revelar los requerimientos esenciales del diseño.

Maquetas. Cuando se comienza el desarrollo, tiene por objetivo presentar a los usuarios y/o clientes la apariencia del sistema final. Los usuarios pueden manifestar su opinión.

Ambos métodos son muy útiles para establecer la viabilidad del proyecto y definir acuerdos sobre los objetivos y resultados esperados.

Etapas del método de prototipos

1) Identificación de requerimientos conocido.

La determinación de los requerimientos de una aplicación es tan importante para el método de desarrollo de prototipo como lo es para los métodos del ciclo clásico de desarrollo de sistemas o análisis estructurado (aunque las tácticas son diferentes). Por consiguiente, antes de crear el prototipo, los analistas y usuarios deben trabajar juntos para identificar los requerimientos conocidos que tiene que satisfacerse. Para hacerlo determinan los fines para lo que servirá el sistema y el alcance de sus capacidades.

2) Desarrollo de un modelo de trabajo

Es útil comenzar el proceso de construcción del prototipo con el desarrollo de un plan general que permita a las personas conocer lo que se espera de ellas y del proceso de desarrollo. Es difícil, y en ocasiones imposibles, fijar una fecha tentativa de terminación. La experiencia con el sistema es la que determina eventualmente cuando el sistema ésta terminado.

Para comenzar la primera iteración, usuarios y analistas identifican de manera conjunta los datos que son necesarios para el sistema y especifican la salida que debe producir la aplicación.

Las decisiones de diseño necesarias para desarrollar la salida del sistema cambian muy poco en relación con las tomadas en otros métodos de desarrollo. Sin embargo, con un prototipo, se espera que las especificaciones iníciales estén incompletas.

En el desarrollo de un prototipo se preparan los siguientes componentes:

- El lenguaje para el diálogo o conversación entre el usuario y el sistema
- Pantallas y formato para la entrada de datos
- Módulos esenciales de procesamiento
- Salida del sistema

Al construir el prototipo se deben seguir los estándares para datos que emplea la organización.

En esta etapa es más importante la rapidez con que se construye el prototipoque la eficiencia de operación. Es por esto que el analista no intenta optimizar lavelocidad de operación del sistema.

Durante la evaluación los analistas de sistemas desean capturar

3) El prototipo y el usuario

Es responsabilidad del usuario trabajar con prototipo y evaluar su característica y operación. La experiencia con el sistema bajo condiciones permite obtenerla familiaridad indispensable para determinar los cambios o mejoras que sean necesarios así como la eliminación de características inadecuadas o innecesarias.

4) Revisión del prototipo

Información sobre los que les gusta y los que les desagrada a los usuarios.

La información obtenida tendrá influencia sobre las características de la siguiente versión de la aplicación.

Los cambios al prototipo son planificados con los usuarios antes de llevarlos a cabo. El analista es el responsable de realizar las modificaciones.

5) Repetición del proceso las veces que sea necesario.

El proceso finaliza cuando los usuarios y analistas están de acuerdo en que el sistema ha evolucionado lo suficiente como para incluir todas las características necesarias o cuando ya es evidente que no se obtendrá mayor beneficio.

6) El abandono o dejarlo como esta:

Cuando se verifica que no es posible desarrollar el sistema para satisfacerlos objetivos deseados, ya sea por la tecnología existente o por el factor económico.

2.4 Marco legal

Compañía Anónima

Art. 143.- La compañía anónima es una sociedad cuyo capital, dividido en acciones negociables, está formado por la aportación de los accionistas que responden únicamente por el monto de sus acciones. Las sociedades o compañías civiles anónimas están sujetas a todas las reglas de las sociedades o compañías mercantiles anónimas.

Art. 144.- Se administra por mandatarios amovibles, socios o no.

La denominación de esta compañía deberá contener la indicación de "compañía anónima" o "sociedad anónima", o las correspondientes siglas. No podrá adoptar una denominación que pueda confundirse con la de una compañía preexistente. Los términos comunes y aquellos con los cuales se determine la clase de empresa, como "comercial", "industrial", "agrícola", "constructora", etc., no serán de uso exclusive e irán acompañadas de una expresión peculiar.

DE LOS BALANCES

Art. 289.- Los administradores de la compañía están obligados a elaborar, en el plazo máximo de tres meses contados desde el cierre del ejercicio económico anual, el balance general, el estado de la cuenta de pérdidas y ganancias y la propuesta de distribución de beneficios, y presentarlos a consideración de la junta general con la memoria explicativa de la gestión y situación económica y financiera de la compañía.

Ley Orgánica de Régimen Tributario Interno

- Esta Ley regula el impuesto a la renta, el impuesto al valor agregado y el impuesto a los consumos especiales.
- Obligaciones Tributarias:
- Declaración de impuesto a la renta
- Declaración de retenciones en la fuente
- Declaración mensual del IVA
- Impuesto a la propiedad de vehículos motorizados
- Entrega de anexos transaccionales

Código de Comercio

Es la norma que rige las obligaciones de los comerciantes en sus operaciones mercantiles, y los actos y contratos de comercio, aunque se lleven a cabo por no comerciantes.

Porque se dedica a la comercialización; la Ferretería Gordon se acoge a esta ley porque se dedica a la compra venta de artículos de construcción.

Código de Trabajo

Regula las relaciones entre empleadores y trabajadores, aplicando las diferentes modalidades y condiciones de trabajo. Estas leyes cumple la Ferretería Gordon; ya que para realizar sus actividades necesita recurso humano.

Ley de Seguridad Social

Regula la aplicación del seguro general obligatorio, y norma al organismo de control y aplicación IESS. La Ferretería Gordon, tiene a sus empleados asegurados y con sus debidos beneficios de Ley

LEY DE PROPIEDAD INTELECTUAL

Art. 28. Los programas de ordenador se consideran obras literarias y se protegen como tales. Dicha protección se otorga independientemente de que hayan sido incorporados en un ordenador y cualquiera sea la forma en que estén expresados, ya sea en forma legible por el hombre (código fuente) o en forma legible por máquina (código objeto), ya sean programas operativos y programas aplicativos, incluyendo diagramas de flujo, planos, manuales de uso, y en general, aquellos elementos que conformen la estructura, secuencia y organización del programa.

Art. 29. Es titular de un programa de ordenador, el productor, esto es la persona natural o jurídica que toma la iniciativa y responsabilidad de la realización de la obra. Se considerará titular, salvo prueba en contrario, a la persona cuyo nombre conste en la obra o sus copias de la forma usual.

Dicho titular está además legitimado para ejercer en nombre propio los derechos morales sobre la obra, incluyendo la facultad para decidir sobre su divulgación.

El productor tendrá el derecho exclusivo de realizar, autorizar o prohibir la realización de modificaciones o versiones sucesivas del programa, y de programas derivados del mismo.

Las disposiciones del presente artículo podrán ser modificadas mediante acuerdo entre los autores y el productor.

Ordenanzas Municipales

Art. 1.- Objeto.- La presente ordenanza establece las normas mínimas para diseño y construcción de espacios que permitan habitar suelo o edificación garantizando su funcionalidad, seguridad y estabilidad.

La conservación, consolidación y mejora de los inmuebles declarados de interés cultural se realizará según lo dispuesto en la Ley de Patrimonio Cultural y su Reglamento, en las disposiciones y normas de la Ordenanza de Patrimonio Cultural y en las disposiciones pertinentes de la presente normativa y aquellas especiales que para el efecto dicten los organismos pertinentes.

Art. 2.- Ámbito y sujeción.- Toda obra de habilitación del suelo y edificación que se desarrolle en el Distrito Metropolitano de Quito por personas naturales o jurídicas, públicas o privadas, se sujetará a lo dispuesto en esta normativa, a las regulaciones establecidas por el INEN que son referidas en este instrumento y a las regulaciones vinculadas.

La Municipalidad del Distrito Metropolitano de Quito, a través de sus entidades y Administraciones Zonales, hará cumplir lo dispuesto en esta Ordenanza. La Dirección Metropolitana de Planificación Territorial y Servicios Públicos se encargará de absolver consultas aclaratorias a esta norma.

Art. 3.- Modificaciones.- Corresponde a la Dirección Metropolitana de Planificación Territorial y de Servicios Públicos evaluar y actualizar permanentemente las normas constantes en esta Ordenanza. Para el efecto, propondrá al Concejo Metropolitano de Quito, por medio de la Comisión de Suelo y Ordenamiento Territorial, las modificaciones que san del caso, mediante un informe en el que se documente su alcance o naturaleza, previa consulta pública y coordinación con las administraciones zonales.

CAPÍTULO III

INVESTIGACIÓN CIENTIFICA

3.1 Tipos de Investigación

Investigación Cuantitativa Bibliográfica-Documental

La investigación documental permite analizar datos e información del tema objeto a investigación.

El tipo de investigación documental se aplica en su totalidad al proyecto de inventario y facturación para la Ferretería Gordon, ya que con la finalidad de extraer datos de información del mismo tengo que recurrir a fuentes de investigación que por tratarse de un asunto técnico están almacenados en una categoría especial, la recopilación de información es necesaria de libros técnicos, manuales, revistas, catálogos, me permitirá analizar los procesos y procedimientos propios del sistema que tengo que desarrollar, igualmente me permitirá conocer el origen de la información el flujo que sigue la misma y donde necesita ser almacenada, mediante un diseño lógico y coherente, materializare la información recopilada de los documentos antes mencionados.

Igual tratamiento se realizará con la información que obtenga a través de internet, Cd y material magnético que pueden aportar al desarrollo y construcción de mi proyecto. Cabe recalcar que toda la información referente a este tema será profundamente antes de ser considerado como válido, ya que la estructuración de un sistema informático depende mucho de la toma de datos en tiempo real, la estructuración de la misma a través de

modelos lógicos que son de entendimiento para los usuarios programadores como para la generación de base de datos que finalmente dará como resultado sistemas informáticos consistentes, rápidos y confiables.

La documentación anteriormente analizada permitirá conocer, interpretar, comparar y enfocar criterios, opiniones, conceptualizaciones que irán determinando las conclusiones o recomendaciones que tenga que enfocar como parte de mi proyecto de grado.

Investigación de Campo Descriptiva

La investigación descriptiva reseña rasgos o atributos de la población.

Este tipo de investigación se aplica al proyecto de investigación ya que con el mismo me permite analizar y describir todos los procesos y procedimientos que son parte del sistema a desarrollar igualmente permite determinar los rasgos y cualidades del recurso humano que forma parte de estos proceso; poniendo especial énfasis en el comportamiento del recurso humano pero anteponiendo lo que nos interesa sacar de resultados de las tareas que se realizan en cada uno de los procedimientos estructurales de la organización.

Es muy importante la aplicación de este tipo de investigación porque permitirá ir analizando los tiempos de ejecución de cada tarea por parte del recurso humano, y al mismo tiempo el aporte que significa estos tipos de información para el proyecto presentado.

Investigación de Campo Explicativa

Este tipo de investigación se aplicará en el estudio del planteamiento del problema ya que permitió determinar las causas que generaron la problemática por perdida de la información, la no facturación y de igual manera el efecto que tuvieron los mismos con el desarrollo del proyecto Sistema de Facturación e Inventario; normalmente pudiendo contestarlas

interrogantes porque ocurrió el hecho o fenómeno como la pérdida de la información y a su vez la misma relacionándola con los motivos de quien lo genero a fin de poder descubrir las causas y efectos que motivaron la ejecución de los mismos. Igualmente en la aplicación de este tipo de investigación que se fundamenta en la observación directa permitiendo determinar los conocimientos adquiridos en forma teórica y poderlos materializar en una forma práctica a través de las soluciones planteadas y de soluciones informáticas; aportes a solucionar el problema planteado.

3.2 Métodos de investigación

Histórico-Lógico

El método histórico lógico se aplicó al proyecto inventario y facturación para la ferretería Gordon ya que con la ayuda del mismo me ha permitido validar todos los procesos y procedimientos que se encuentran materializados en un plan y que me permitieron reproducir cronológicamente toda la sucesión de tareas que se tienen que ejecutar en un macro proceso y proceso al mismo tiempo que permite definir claramente él comportamiento de cada uno de los procesos es decir verificando el alcance que tiene cada uno de ellos en la realización de determinada tarea y al mismo tiempo definido en un tiempo de ejecución. Por consiguiente este método ayuda a delimitar las áreas automatizables y al mismo tiempo las actividades que tiene que cumplir cada uno de los actores involucrados.

Analítico-Sintético

El método analítico-sintético se aplica al proyecto ya que con el mismo permite realizar un análisis a profundidad de cada uno de los elementos que intervienen en el desarrollo de un software informático es decir una metodología, un diseño, estructuración en una base de datos etc. Lo que llevado a la incorporación de un lenguaje de programación van a permitir la sistematización en la organización; por otro lado de realizar una síntesis de la información

teórica y conceptual me permitirá reunir todos los documentos válidos a fin de poder reestructurar todo la lógica didáctica y conceptual de todo elemento que intervienen en el desarrollo de un tema inventario y facturación para la ferretería Gordon pudiendo encontrar el mayor esfuerzo en la búsqueda de información y análisis de la misma pero únicamente con la salvedad de que sea aplicable estrictamente al tema de ejecución. Por consiguiente el método analítico-sintético tiene su fundamentación de aplicación en cómo vamos a tratar la información teórica necesaria para la estructuración del capítulo II para este proyecto.

Inductivo

Este método realiza un análisis de la información desde su inicio hasta llegar a obtener información definida y estructurada; por lo tanto este método, permitirá ir analizando las tareas y procedimientos que originan los sistemas informáticos lo que desencadenara una serie de información valida en la obtención de datos esperados a su vez los mismos irán incrementando su valor a medida que reciba más información lo que permitirá consolidar en sistemas diseñados y estructurados y con una aplicación en una base de datos por lo tanto este método si realiza un análisis de información partiendo desde donde se origina la misma determinará los flujos necesarios de información a ser receptados por las interfaces graficas de usuarios y a su vez esta información valida será almacenada en una base de datos estructurada que en la parte construida viene a constituir lo general de la información informática.

Deductivo

Este método permite analizar la información de lo general a lo particular entendiéndose sobre el tema denominado facturación e inventario para la Ferretería Gordon, los general constituye el almacenamiento de la información en una base de datos estructurada la misma que para su estructuración siguió un proceso de análisis, diseño y se definirán ordenadamente los tipos de datos a ser utilizados al igual la cantidad de información a ser

remplazada por lo tanto en el tema de facturación e inventario tendrá los siguientes tipos de datos:

Producto Tipo: id_productoVarchar(10), nombre varchar(40), precioufloat, preciocfloat, caracteristicasvarchar(100), existencia float.

Ventas Tipo: id_ventasInt, Monto Flota, Fecha Date, Hora Time

lineaproducto Tipo: id_lineaint, descuento Int, cantidad float, id_ventasint, id_producto, varchar(10), importe float.

Sistémica

Este método nos conduce a la estructuración lógica del proyecto inventario y facturación para la ferretería Gordon ya que tendrá que seguir un orden definida en la estructuración del capítulo que permitirá a cualquier tutor ir verificando paso a paso el análisis de la problemática que objetivos se persigue como voy aplicar mi investigación finalmente un desarrollo de un software informático y que será la vinculación de las partes constitutivas del proyecto y al mismo tiempo el objetivo general que se persigue al automatizar todos los procesos y procedimientos de la entidad.

3.3 Herramientas de recolección de información

ENTREVISTA.

Esta herramienta consiste básicamente en reunirse una o varias personas y cuestionarlas en forma adecuada para obtener información.

Utilizaremos esta técnica para obtener datos consistentes mediante el diálogo o conversación para conseguir datos que nos ayuden al mejor manejo de información veraz y concisa que nos proporcionará la dueña del establecimiento.

SELECCIÓN DE LOS ELEMENTOS DE LA ENTREVISTA.

Los elementos que me ayudarán a indagar y buscar una información más allá que nos ayude a la realización de mi proyecto tomando datos en la entrevista con el Sr. Marco Gordon son:

- Historia,
- Cantidad de colaboradores,
- Jornadas de trabajo,
- Productos,
- Servicios,
- Visión,
- Planes futuros.

PROCEDIMIENTO PARA SU REALIZACIÓN.

La entrevista será dirigida al Dueño del local, en donde buscamos pues que el mismo nos haga un relato de cómo empezó la idea de una empresa, los servicios con que cuenta, los planes a futuros, qué le gustaría mejorar de la Ferretería Gordon, qué ha sido lo más importante y como ha ayudado a la comunidad donde se encuentra ubicada.

Entrevista

PREGUNTA

1. ¿Cuénteme una reseña de su negocio desde sus inicios hasta hoy día?

RESPUESTA

Bueno, a partir de 1,989 emprendí una pequeña empresa, la cual era un negocio de pinturas el cual no tenía que ver con artículos de construcción.

Luego gracias a mi Papá el cual tenía un lote aquí al frente de la Panamericana decidí hacer un edificio de dos pisos y con la ayuda de la familia y con información de nuestros hijos decidimos abrir un local comercial aprovechando la fluidez tanto de instituciones públicas como privadas y público en general, dejando así el negocio de las pinturas.

ANÁLISIS

Específicamente la respuesta de esta pregunta ayuda grandemente a obtener información relevante sobre los antecedentes de la empresa, lo cual es muy importante para el desarrollo del Sistema que se está proponiendo a la empresa.

PREGUNTA

2. ¿Con cuántos empleados cuenta en su negocio?

RESPUESTA

Actualmente solo somos seis personas laborando aquí en mi empresa, los cuales son dos Vendedores, un técnico, dos choferes y mi persona.

ANÁLISIS

Ayuda a determinar los factores por los que la empresa debe de ser automatizada para controlar y facilitar el normal desarrollo de sus actividades evitando el estrés, debido a la falta de personal requerido para este tipo de trabajo.

PREGUNTA

3. ¿Cómo y cada qué tiempo surte su negocio?

RESPUESTA

Lo hago personalmente ya que me gusta escoger la mercancía que pondré a la venta, surto calculando a no tener faltante de mercaderías para un pedido en el momento de facturar.

ANÁLISIS

Determina cómo se realiza el registro de la mercadería recién llegada a la empresa y el tiempo en que se demoran.

PREGUNTA

4. ¿Qué tipo de control o como usted lleva el inventario de la Mercancía que usted tiene en su local?

RESPUESTA

Bueno, el Sistema de Inventario que utilizo a veces no es el más adecuado. Actualmente utilizo un Sistema llamado Kardex el cual viene siendo como las tarjetas o fichas en el que uno lleva los archivos de lo que se compra y al final del periodo que son de 6 meses en que lo hago. Periódicamente realizo un conteo de lo que tengo y ahí me doy cuenta de lo que he vendido y así calculo lo que puedo vender en los próximos seis meses.

ANÁLISIS

La respuesta ayuda a determinar el tipo de control que necesita las mercaderías de la empresa para un mejor detalle de la información y así poder establecer el stock de las mismas.

PREGUNTA

5. ¿Si usted tuviera que cambiar algo de su negocio que sería?

RESPUESTA

Me gustaría mejorar muchas cosas, lo primero sería contar con una bodega adicional, ya que donde estoy es muy pequeña.

ANÁLISIS

Ayuda a prevenir factores que pueden llegar a afectar al sistema a desarrollar.

PREGUNTA

6. ¿Lleva un control de las ventas diarias?

RESPUESTA

Actualmente su llevo un control de las ventas diarias, pero me cuesta mucho trabajo ya que tengo que sumar todas las facturas del día.

ANÁLISIS

La respuesta es muy factible para establecer el punto de venta que va a necesitar para que llegue a ser muy eficiente el sistema, y así poder mejorar la atención al cliente.

OBSERVACIÓN.

Es un instrumento muy utilizado en la actualidad para la obtención de información, donde uno se sumerge en el problema para observar detenidamente el proceso, anomalía más de cerca.

Con la utilización de esta técnica de investigación nos ponemos en contacto con el fenómeno o hecho que analizamos para sacar información, tratando de forma discreta realizar la observación para que la información no se contamine ya que cuando las personas suelen saber que se les observan hacen cosas que normalmente no realizan, es por ello que hay que hacerla de forma cuidadosa y evitar que la persona que se observa no se entere para que la observación de mejores resultados que nos ayuden a reconocer el problema o las causas del mismo.

MÉTODOS DE OBSERVACIÓN.

La observación que realizare es una combinación de la observación directa e indirecta ya que estaremos en contacto con el problema o situación a mejorar, será participante porque nos adentraremos para conocer y tratar desde cerca con el objeto investigado y la observación la realizaremos como un equipo de trabajo que somos y como nos trasladaremos al lugar pues la observación será de campo.

Cuando realicemos la observación utilizaremos una ficha de observación para así poder obtener datos que nos ayuden en la confección del análisis y mejoramiento del inventario de mercancía de Ferretería Gordon.

ASPECTOS O CONDUCTAS A OBSERVAR.

Las conductas o aspectos que observe es la realización y tipo de inventario o control que se aplica para saber con qué mercancía se cuenta, cómo la manejan o llevan, además ver las debilidades y fortalezas con que cuentan para el mejor servicio y atención a los clientes que asistan a Ferretería Gordon, empresa de servicios de la localidad.

CAPÍTULO IV

DESARROLLO DE LA PROPUESTA

4.1 Diagnóstico Situacional

La necesidad creciente de comercializar cada día mayor cantidad de productos y servicios, requiere de técnicas y elementos que faciliten su desplazamiento hacia los mercados potenciales de clientes.

En este sentido, la situación que presenta la Ferretería Gordon, se debe a la falta de un Sistema de Inventario y Facturación que le permita optimizar las funciones y desarrollar con eficiencia las actividades, según sus cargos desempeñados.

El control de la cartera de clientes se realiza en un archivo de Excel en el cual se anota el nombre, teléfono, dirección y pedidos del cliente, con la dificultad de que la información continuaba dispersa y para poder surtir un producto se tiene que consultar a varias fuentes, de las cuales el acceso no es sencillo, ya que ésta computadora se encuentra en la planta alta del local de la empresa.

- La situación actual de Software y Hardware es: sistema operativo XP service Pack 3, aplicativo para controlar los inventarios de la mercadería así como también lo es el proceso de facturación. Una caja registradora, Fax, 4 líneas telefónicas, Un conmutador, tres equipo de cómputo, una impresora.

- La situación actual de la comunicación: la Ferretería Gordon su infraestructura del local donde funciona no dispone de una área amplia para que pueda

funcionar, como es la comunicación con los diferentes dispositivos es muy monótona.

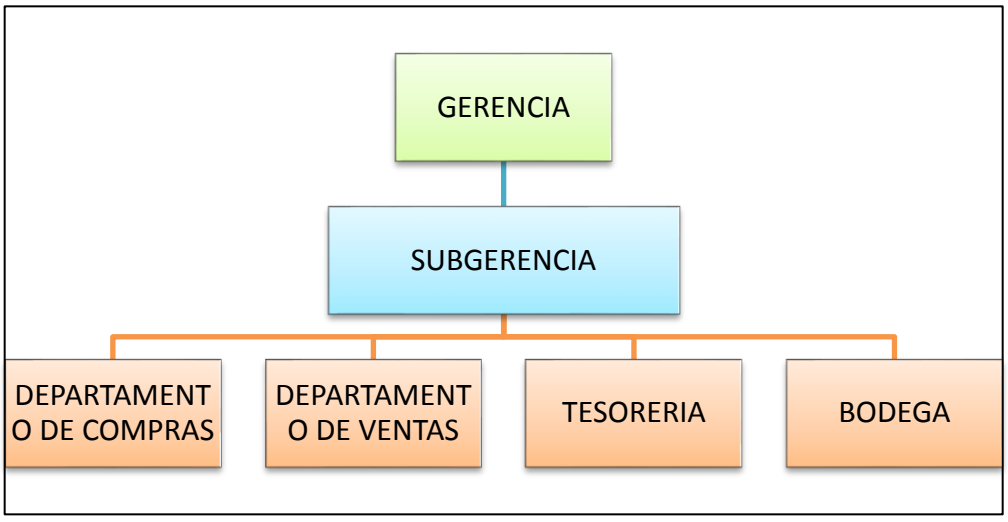
4.2 Estructura Organizacional

La Ferretería Gordon cuenta con el rol organizacional, porque tienen unos objetivos claros como lo es la comodidad y satisfacción del cliente, se tienen proyectos concisos para aumentar las ventas con márgenes muy superiores a los presentes.

Esta ferretería tiene una organización formal, es cierto con pocas divisiones, es una pequeña empresa, está debidamente departa mentalizada.

Organigrama Estructural

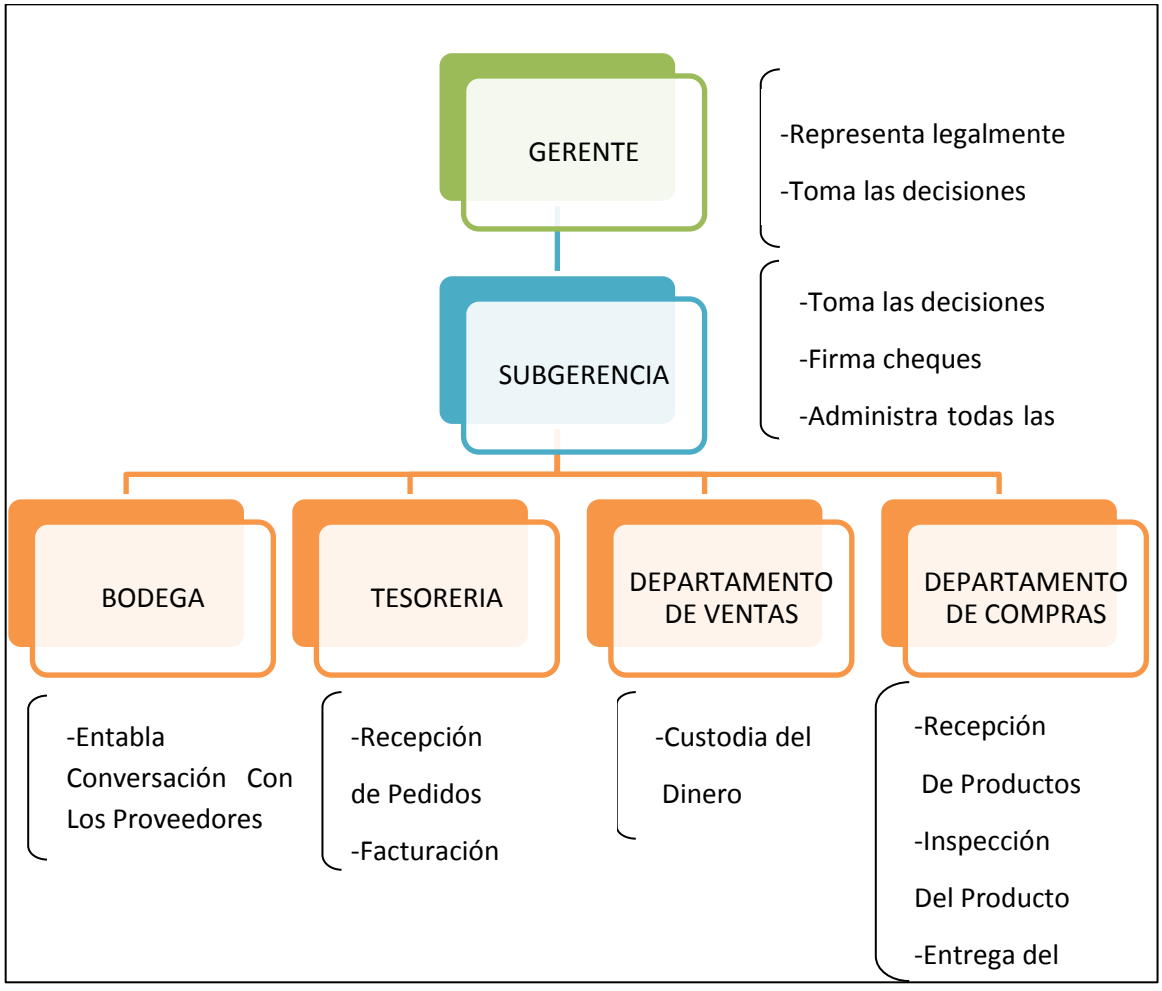
El organigrama estructural es la representación gráfica de la estructura orgánica de la Ferretería Gordon que refleja, en forma esquemática, la posición de las áreas que la integran, sus niveles jerárquicos, líneas de autoridad y de asesoría.



Organigrama Estructural
Ilustración 4
Fuente: Ferretería Gordon

Organigrama Funcional

Teniendo en cuenta, que el organigrama es una representación gráfica de la Ferretería Gordon que expresa en términos concretos y accesibles la estructura, jerarquía e interrelación de las distintas áreas que componen dicha empresa, resulta muy conveniente que todos los que la componen conozcan cuál es su definición, para que de esa manera, tengan un conocimiento básico pero fundamental, acerca de lo que es este sencillo pero valioso recurso.



Organigrama Funcional
Ilustración 5
Fuente: Ferretería Gordon

Los empleados de la empresa tienen que desarrollar las siguientes funciones; las cuales se detalla a continuación:

Gerencia

- Representar legalmente a la empresa
- Tomar las decisiones
- Firmar cheques
- Firmar de los formularios tributarios
- Autorizar la compras

Subgerencia

- Tomar las decisiones
- Firmar cheques
- Administrar todas las áreas

Departamento de Ventas

- Proformar
- Receptar los pedidos
- Facturar
- Servicio al cliente
- Conocer del stock de productos
- Mantener limpio y ordenados los mostradores
- Realizar el trámite correspondiente a devolución de productos
- Realizar guías de remisión
- Ingreso de facturas manuales al sistema
- Cuadre diario de cemento

Departamento de Compras

- Entablar comunicación con los proveedores
- Hacer pedidos
- Ver descuentos y las políticas de compras
- Hacer las debidas retenciones
- Realizar las liquidaciones de compras en el que amerité el caso

Departamento de Tesorería

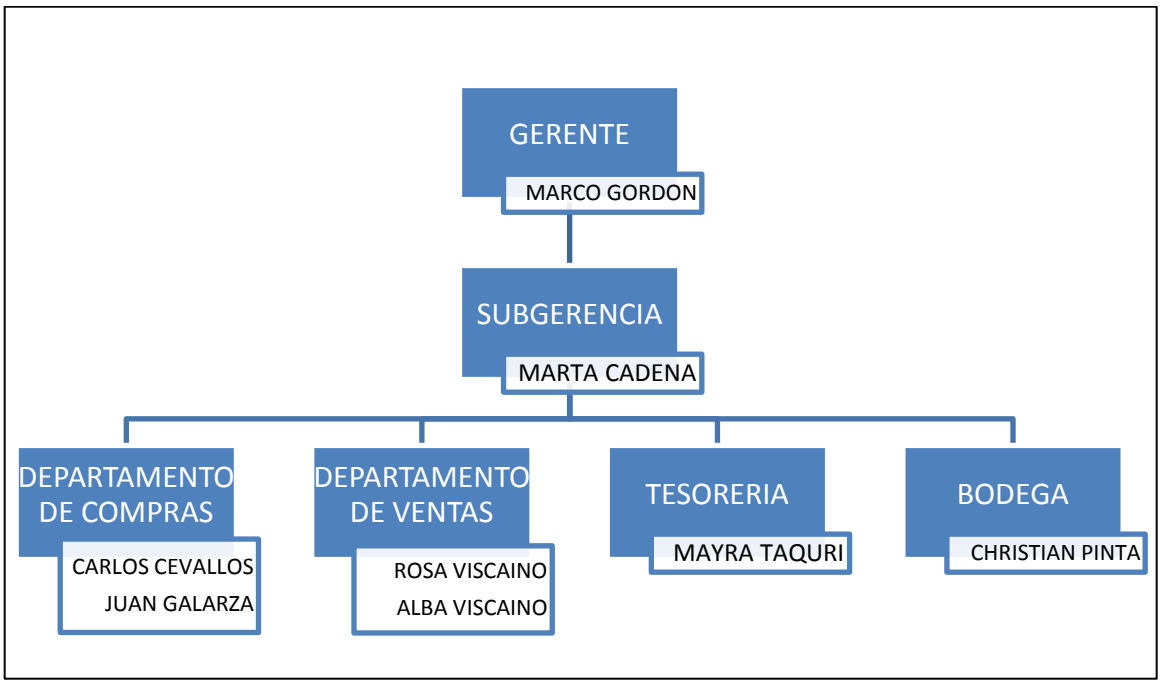
- Depositar a cada cuenta bancaria
- Entregar diariamente las ventas realizadas
- Realizar los debidos pagos a los proveedores
- Desembolso de gastos de administración, ventas, y otros

Bodega

- Inspección de que el producto este en perfectas condiciones y no defectuoso
- Tener los productos ordenados
- Realizar la entrega del producto al cliente

Organigrama Posicional

Incluyen las principales funciones que tienen asignadas, además de las unidades y sus interrelaciones. Este tipo de organigrama es de gran utilidad para capacitar al personal y presentar a la organización en forma general.



Organigrama Posicional
Ilustración 6

Fuente: Ferretería Gordon

4.3 Infraestructura Informática

HARDWARE

La empresa cuenta con equipos de las siguientes características:

Características	***Ubicación***
Intel Corei7	*Desarrollo*
Intel ZEON	*Servidor*

Intel Coreduo	*Homologación*

Cuadro de hardware
Tabla 2
Fuente: Diego Báez

4.3.2 SOFTWARE

Las máquinas cuentan con un sistema operativo:

Software	Ubicación
Windows seven, Office 2007,	Equipo desarrollo
Windows sever SQL 2008	Servidor
Windows XP	Homologación de equipos

Cuadro de Software
Tabla 3
Fuente: Diego Báez

4.3.3 RECURSO HUMANO TÉCNICO

En el siguiente cuadro se muestra el personal con el cual se cuenta actualmente en la empresa:

Nombre	Descripción
Diego Báez	Alumno

Ing. Rodrigo Cobos	Tutor
Ing. Marco Andrade	Técnico

Cuadro Recurso Humano
Tabla 4

Fuente: Diego Báez

4.4 Descripción de alternativas

Con la elaboración de este sistema para la Ferretería Gordon se llenara todos los requerimientos necesarios para un mejor manejo del inventario de mercancía de dicha empresa. Además he comprendido con la realización de este trabajo mucho de los procesos que conlleva realizar un buen análisis e implementar un sistema automatizado, por ello he realizado un estudio de tres alternativas para desarrollar y determinar la viabilidad del proyecto.

ACTIVIDAD	CUMPLE	NO CUMPLE
Back End		
MySQL		
Modelo Negocio		
Casos de Uso		
Clases		
Dominio		
Etc.		
Plataforma		
Netbeans		
Windows 7		
Implementación		
Secuencia		

Estado		
Colaboración		
Capacitación Usuario		
Manual de Usuario		
Metodología		
RUP		
FrondEnd		
Java		
Manuales Técnicos		
Diccionario de Datos		
Pruebas de Funcionamiento		
Unidad		
Integridad		
Validación		
Sistema (Hardware, personas)		
Seguridad		
Interfaces Gráficas		
sistema(resistencia, consistencia)		
Capacitación Técnica		
Metodologías (Visual, Práctico)		
En las instalaciones de la empresa		
Una semana de Capacitación		
Manuales		

Estándares Programación		
Variables		
Constantes		
Controles Visuales		
Código Fuente Comentado		
Tiempo de Desarrollo		
Garantía Técnica		
Soporte Técnico		
Costo		
Código Fuente (Si/No)		

Cuadro de alternativas
Tabla 5

Fuente: Diego Báez

Empresa Soluciones Computacionales Alternativa 1

Esta alternativa no cumple con las condiciones esperadas para el dueño de la empresa puesto que varias de las características exigidas por el dueño de la empresa no llegan a cumplirse en su totalidad y más aun su costo.

ACTIVIDAD	CUMPLE	NO CUMPLE
Back End	X	
MySQL		
Modelo Negocio	X	
Casos de Uso		
Clases		
Dominio		
Etc		
Plataforma	X	
Netbeans		
Implementación		X

Secuencia		
Estado		
Colaboración		
Capacitación Usuario	X	
Manual de Usuario		
Metodología	X	
RUP		
FrondEnd		X
JAVA		
Manuales Técnicos	X	
Diccionario de Datos		
Pruebas de Funcionamiento		X
Unidad		
Integridad		
Validación		
Sistema (Hardware, personas)		
Seguridad		
Interfaces Gráficas		
sistema(resistencia, consistencia)		
Capacitación Técnica		X
Metodologías (Visual, Práctico)		
En las instalaciones de la empresa		
Una semana de Capacitación		
Manuales		

Estándares Programación	X	
Variables		
Constantes		
Controles Visuales		
Código Fuente Comentado		
Tiempo de Desarrollo	6 MESES	
Garantía Técnica	6 MESES	
Soporte Técnico	Cada mes	
Costo	3000	
Código Fuente (Si/No)	SI	

Cuadro de alternativa 1
Tabla 6
Fuente: Diego Báez

Empresa CompuCronix SA Alternativa 2

En esta alternativa las condiciones y características no son viables para el dueño, ya que el costo es bajo pero no se ajusta a las necesidades que el usuario final espera con la automatización de los procesos que sigue.

ACTIVIDAD	CUMPLE	NO CUMPLE
Back End		x
MySQL		
Modelo Negocio	x	
Casos de Uso		
Clases		
Dominio		
Etc.		
Plataforma		x
Netbeans		
Windows 7		

Implementación		x
Secuencia		
Estado		
Colaboración		
Capacitación Usuario	x	
Manual de Usuario		
Metodología	x	
RUP		
FrondEnd	x	
JAVA		
Manuales Técnicos		x
Diccionario de Datos		
Pruebas de Funcionamiento		x
Unidad		
Integridad		
Validación		
Sistema (Hardware, personas)		
Seguridad		
Interfaces Gráficas		
sistema(resistencia, consistencia)		
Capacitación Técnica	x	
Metodologías (Visual, Práctico)		
En las instalaciones de la empresa		
Una semana de Capacitación		
Manuales		

Estándares Programación	x	
Variables		
Constantes		
Controles Visuales		
Código Fuente Comentado		

Tiempo de Desarrollo	6 MESES
Garantía Técnica	1 AÑO
Soporte Técnico	Cada 2 Meses
Costo	2450
Código Fuente Si/No	NO

Cuadro de alternativa 2
Tabla 7
Fuente: Diego Báez

Alumno Alternativa 3

Esta alternativa es la más viable para el dueño de la empresa puesto que no tiene costo y se ajusta con todas las características esperadas por el propietario.

ACTIVIDAD	CUMPLE	NO CUMPLE
Back End	X	
MySQL		
Modelo Negocio	X	
Casos de Uso		
Clases		
Dominio		
etc.		
Plataforma	X	
Netbeans		
Windows 7		

Implementación	X	
Secuencia		
Estado		
Colaboración		
Capacitación Usuario	X	
Manual de Usuario		
Metodología	X	
RUP		
FrondEnd	X	
JAVA		
Manuales Técnicos	X	
Diccionario de Datos		
Pruebas de Funcionamiento	X	
Unidad		
Integridad		
Validación		
Sistema (Hardware, personas)		
Seguridad		
Interfaces Gráficas		
sistema(resistencia, consistencia)		
Capacitación Técnica	X	
Metodologías (Visual, Práctico)		
En las instalaciones de la empresa		
Una semana de Capacitación		
Manuales		

Estándares Programación	X	
Variables		
Constantes		
Controles Visuales		
Código Fuente Comentado		
Tiempo de Desarrollo	6 MESES	
Garantía Técnica	18 MESES	
Soporte Técnico	Cada 3 Meses	
Costo	0	
Código Fuente	si	

Cuadro de alternativa 3
Tabla 8
Fuente: Diego Báez

4.5 Evaluación de la Propuesta

Con la finalidad de poder determinar y seleccionar la alternativa que más conviene a los intereses empresariales se ha definido un sistema de calificación con coeficientes matemáticos de la siguiente manera:

1. Evaluación técnicas 70 %
2. Evaluación Económica 20 %
3. Garantía técnica 5 %
4. Soporte técnico 5 %

Descripción	Valoración
No importante	1 – 4
Importante	5 – 7
Muy Importante	8 – 10

Valoración en Pesos
Tabla 9
Fuente: Diego Báez

Actividades	PESO	Software de Mercado	Empresa de Desarrollo	Alumno
Plataforma Java	7	7	0	7
Generación de Base de Datos MySQL	8	8	0	8
Análisis de las reglas del negocio de los requerimientos	7	7	7	0
Diseño del modelo del sistema	7	0	7	7
Implementación del sistema	7	O	0	7
Estructuración de las líneas de código	7	7	7	7
Facturación de compras y ventas	7	7	7	7
Manejo de Caja diariamente	7	0	7	7
Manejo de Inventario de Mercadería	7	7	7	7
Manejo de Seguridad de acceso al sistema	7	7	7	7
Pruebas de funcionamiento del Sistema	7	7	7	7
Corrección de líneas de código del Sistema	7	7	7	7
Pruebas definitivas de funcionamiento	7	7	0	7
Puesta en marcha del Sistema	8	8	7	7
Total	**100**	**79**	**70**	**93**

Evaluación de Alternativas
Tabla 10
Fuente: Diego Báez

Tabla de costos

Empresa	Costo	Porcentaje
Soluciones Computacionales	$3000	15%
CompuCronix SA	$2450	20%
Alumno	$0	10%

Cuadro de costos alternativas
Tabla 11
Fuente: Diego Báez

Tabla de garantía

Empresa	Tiempo	Porcentaje
Soluciones Computacionales	6 meses	20%
CompuCronix SA	1 año	10%
Alumno	18 meses	15%

Cuadro de Garantías
Tabla 12
Fuente: Diego Báez

Tabla de soporte técnico

Empresa	**Tiempo**	**Porcentaje**
Soluciones Computacionales	Cada mes	20%
CompuCronix SA	Cada dos meses	15%
Alumno	Cada 3 meses	10%

Cuadro de Soporte Técnico
Tabla 13
Fuente: Diego Báez

Realizando la evaluación definitiva de las propuestas se puede concluir claramente que para el software de mercado alcanzo los siguientes resultados, en la parte técnica sumo 79 puntos lo por lo que se le asigna un puntaje de 33 puntos, en la evaluación económica, soporte y garantía técnica un valor de 35 puntos por consiguiente obtiene un resultado general de **68 puntos**.

Para la empresa desarrolladora de software alcanzo los siguientes resultados, en la parte técnica sumo 70 puntos lo por lo que se le asigna un puntaje de 30 puntos, en la evaluación económica, soporte y garantía técnica un valor de 30 puntos por consiguiente obtiene un resultado general de **60 puntos**.

La propuesta a desarrollar alcanzo los siguientes resultados, en la parte técnica sumo 93 puntos lo por lo que se le asigna un puntaje de 40 puntos, en la evaluación económica, soporte y garantía técnica un valor de 35 puntos por consiguiente obtiene un resultado general de **75 puntos**.

4.6 Factibilidad Técnica

De lo expuesto anteriormente se desprende claramente que la opción de desarrollo propuesto como proyecto de tesis de grado, es la más conveniente para realizar el desarrollo del sistema, tanto en lo técnico como el económico.

La misma establece que el aspecto técnico es la principal para poder realizar una calificación coherente y acertada, por otro lado es muy conveniente por el aspecto económico, lo que se traduce en la conveniencia de la realización con la segunda opción que es la más adecuada en todos los aspectos, estableciendo claramente su factibilidad en tiempo y aspecto técnico de acuerdo a las evaluaciones realizadas en el ítem anteriormente expuesto.

4.7 Descripción de los Procesos

Los procesos involucrados en el sistema en desarrollo, son el ingreso, la salida de la mercadería y administración del sistema, dichos procesos reflejarán el estado del inventario, así como llevarán un control sobre la facturación de cada venta.

Una vez analizado cada uno de los procesos, tanto de inventarios como de facturación, se los ha dividido en los siguientes módulos y funciones generales en el sistema:

Módulo inventarios.

Descripción.- Módulo que permite la administración de existencias (ingreso, salidas y reportes) de productos en la empresa.

Entradas.- Información correspondientes a los productos catalogados en las posibles existencias de bodega.

Procesos.- Ingreso, actualización y eliminación de información de productos

Salidas.- Información almacenada en la base de datos correspondiente a las existencias y movimientos de cada uno de los productos almacenados.

Módulo ventas.

Descripción.- Módulo que permite la gestión de ventas de productos

Entradas.- Datos de cliente y productos a ser vendidos.

Procesos.- Facturación, anulación y devolución de productos.

Salidas.- Facturas emitidas, facturas anuladas, información de clientes y afectación al stock de productos.

Módulo administración.

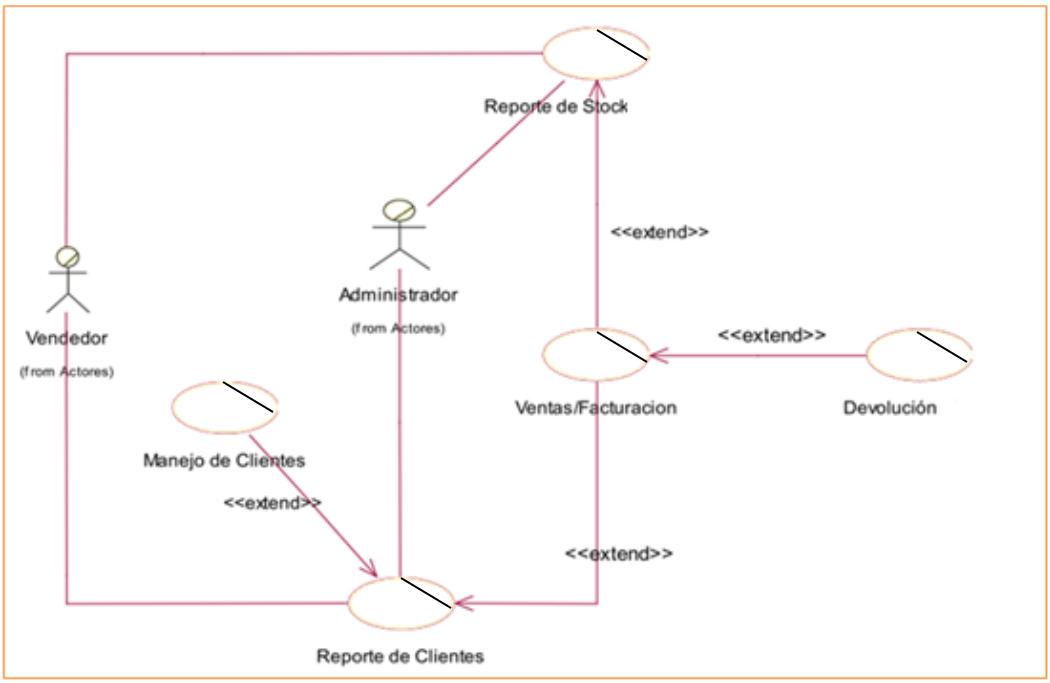
Descripción.- Administración de usuarios, perfiles, países, categorías y proveedores.

Entradas.- Información relativas a las entidades involucradas, usuarios, perfiles, países, categorías y proveedores.

Procesos.- Ingreso, modificación y eliminación de información correspondientes a usuarios, perfiles, categorías, países y proveedores.

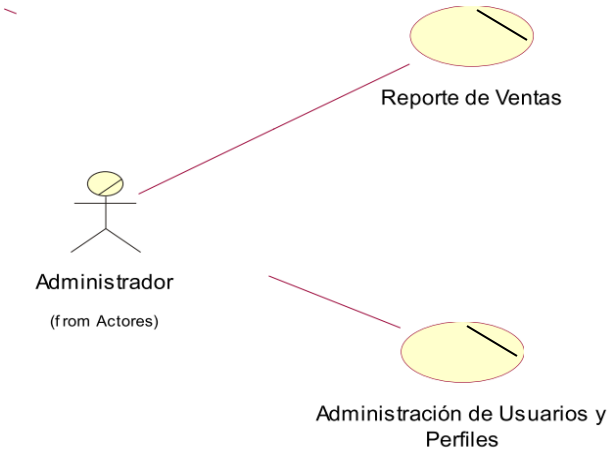
Salidas- Información almacenada en la base de datos relacionada a las entidades antes mencionadas

Casos de Uso



Modelo Casos de uso Ventas
Ilustración 7
Fuente: Diego Báez

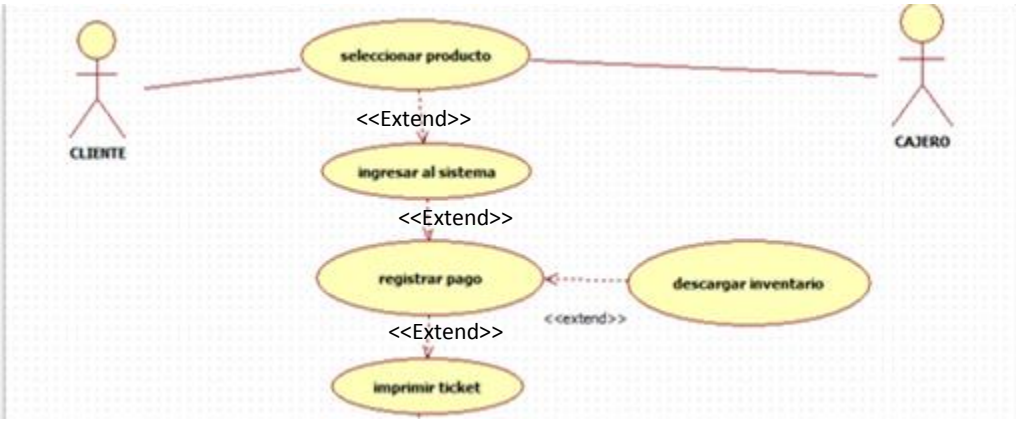
El vendedor captura la clave de los productos solicitados por el cliente, obteniendo el Inventario que existe en el almacén, en caso de haber existencia de los productos se despliega en la pantalla y se actualiza el inventario, en el caso de no existir el producto la venta se dará por terminada y pedirá el ingreso de producto.



Modelo casos de uso Administrador
Ilustración 8
Fuente: Diego Báez

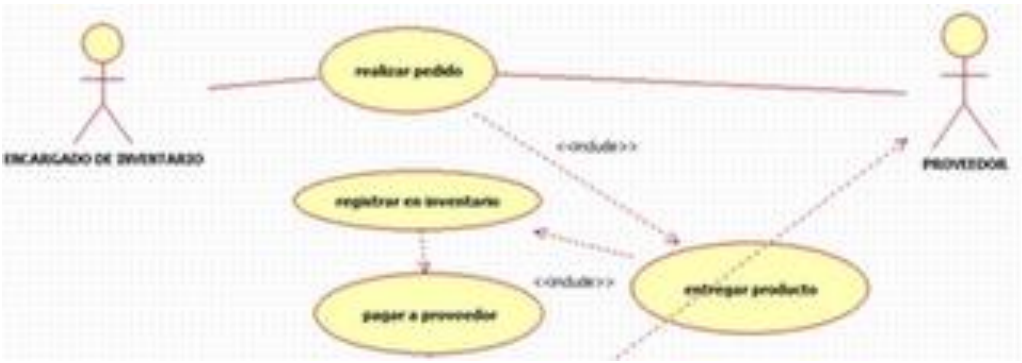
El Administrador introduce los nuevos datos para ser agregados o modificados del almacén, Nómina, los resultados se muestran en la pantalla.

VENTAS DE PRODUCTOS: el cliente selecciona el producto, el cajero toma el producto e ingresa al sistema, el sistema hace la suma de los productos que lleva el cliente, y hace la descarga en el inventario, el cliente paga el cajero le hace el registro, se imprime la factura al cliente y sale del local.



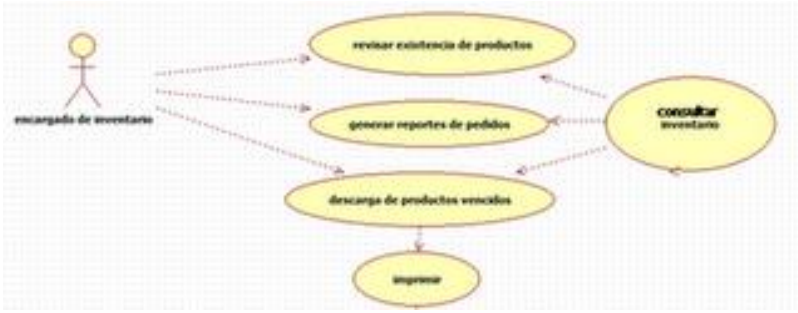
Modelo casos de Ventas de Productos
Ilustración 9
Fuente: Diego Báez

PRODUCTOS A RECIBIR DEL PROVEEDOR: el encargado de inventario realiza el pedido de los productos, proveedor realiza la entrega del producto, se registra el producto y se paga al proveedor.



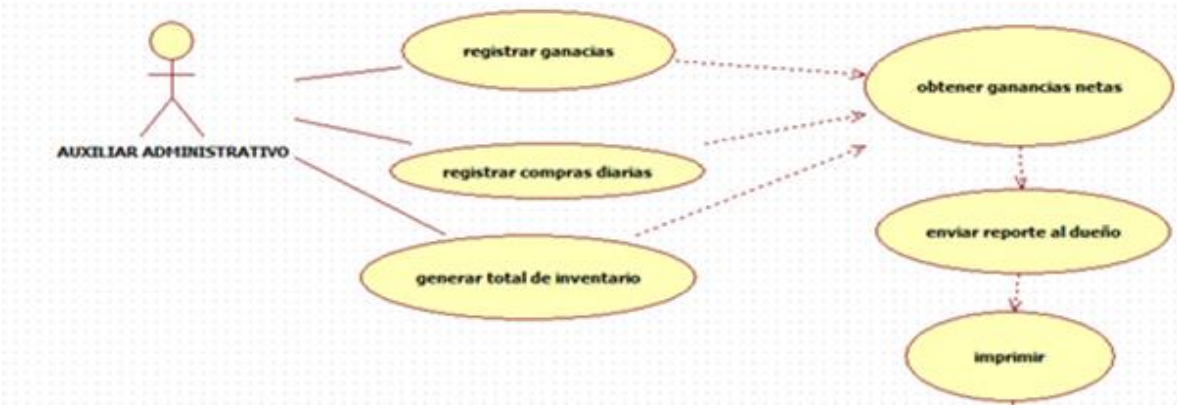
Modelo casos Recepción de productos
Ilustración 10
Fuente: Diego Báez

LLEVAR EL CONTROL DE INVENTARIO: el encargado de inventario revisa las existencias de los productos, la fecha de vencimiento, cambio de precio de productos, pedidos y descargas de productos vencidos.



Modelo casos del Control del Inventario
Ilustración 11
Fuente: Diego Báez

TAREAS ADMINISTRATIVAS: el auxiliar del registro administrativo de las ganancias del día, los totales de inventarios, el total del registro de compras del día, y obtener ganancias netas, enviar al dueño reporte por medio de correo electrónico.



Modelo casos de Tareas Administrativas
Ilustración 12
Fuente: Diego Báez

VENTAS DE PRODUCTOS: El cliente selecciona el producto, el cajero toma el producto y lo ingresa al sistema, el sistema hace la suma de los productos que lleva el cliente, y hace la descarga en el inventario, el cliente paga, el cajero lo registra, se imprime ticket, el cliente se va.

DIAGRAMA DE SECUENCIA

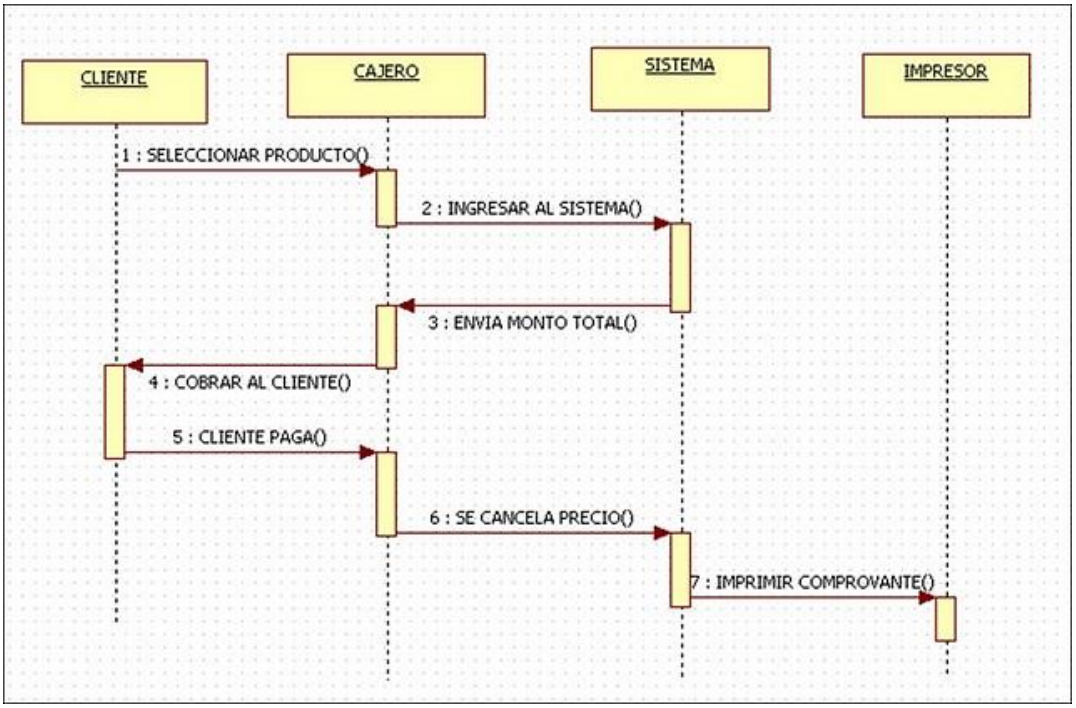


Diagrama de secuencia

Ilustración 13

Fuente: Diego Báez

DIAGRAMA DE COLABORACION

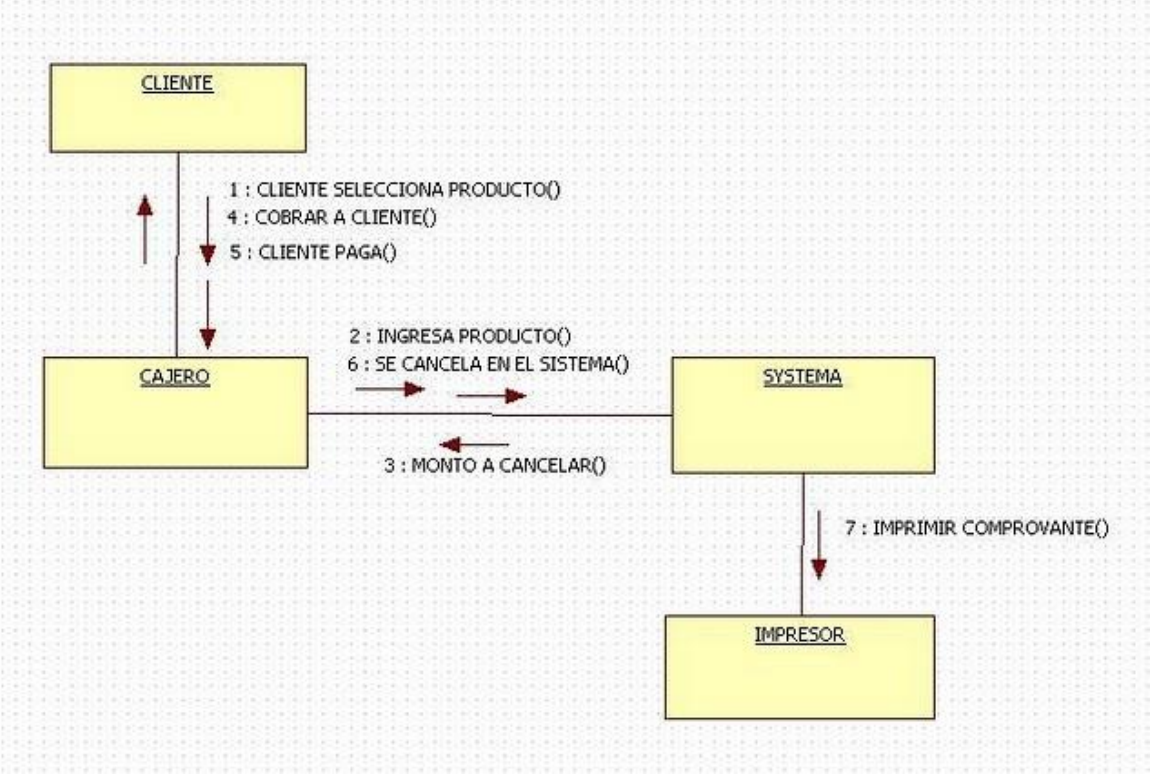


Diagrama de Colaboración
Ilustración 14
Fuente: Diego Báez

RECIBIR PRODUCTOS DE PROVEEDORES: El encargado de inventario realiza el pedido de producto, el proveedor hace la entrega del producto, se registra en el inventario y se le paga al proveedor.

DIAGRAMA DE SECUENCIA

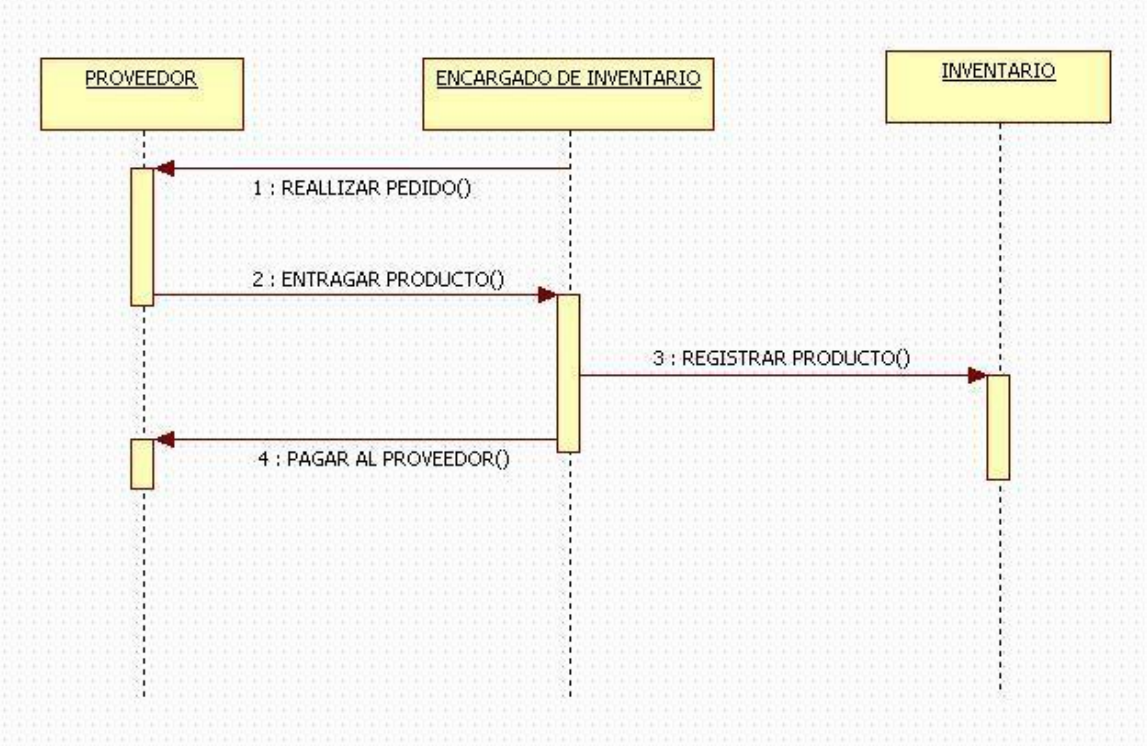


Diagrama de secuencia
Ilustración 15
Fuente: Diego Báez

DIAGRAMA DE COLABORACION

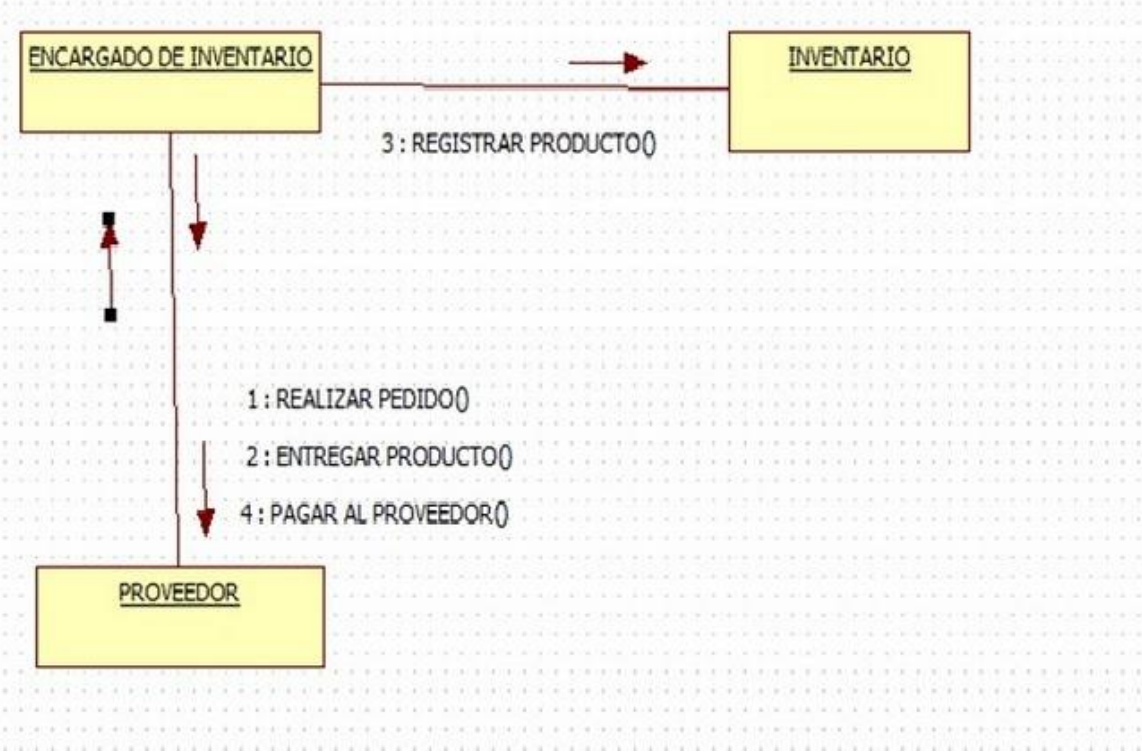


Diagrama de Colaboración
Ilustración 16
Fuente: Diego Báez

LLEVAR CONTROL DE INVENTARIO: El encargado de inventario revisa las existencias de los productos, elimina productos vencidos, actualiza dato de existencia de productos, hace pedido de producto, imprime comprobante y reportes.

DIAGRAMA DE SECUENCIA

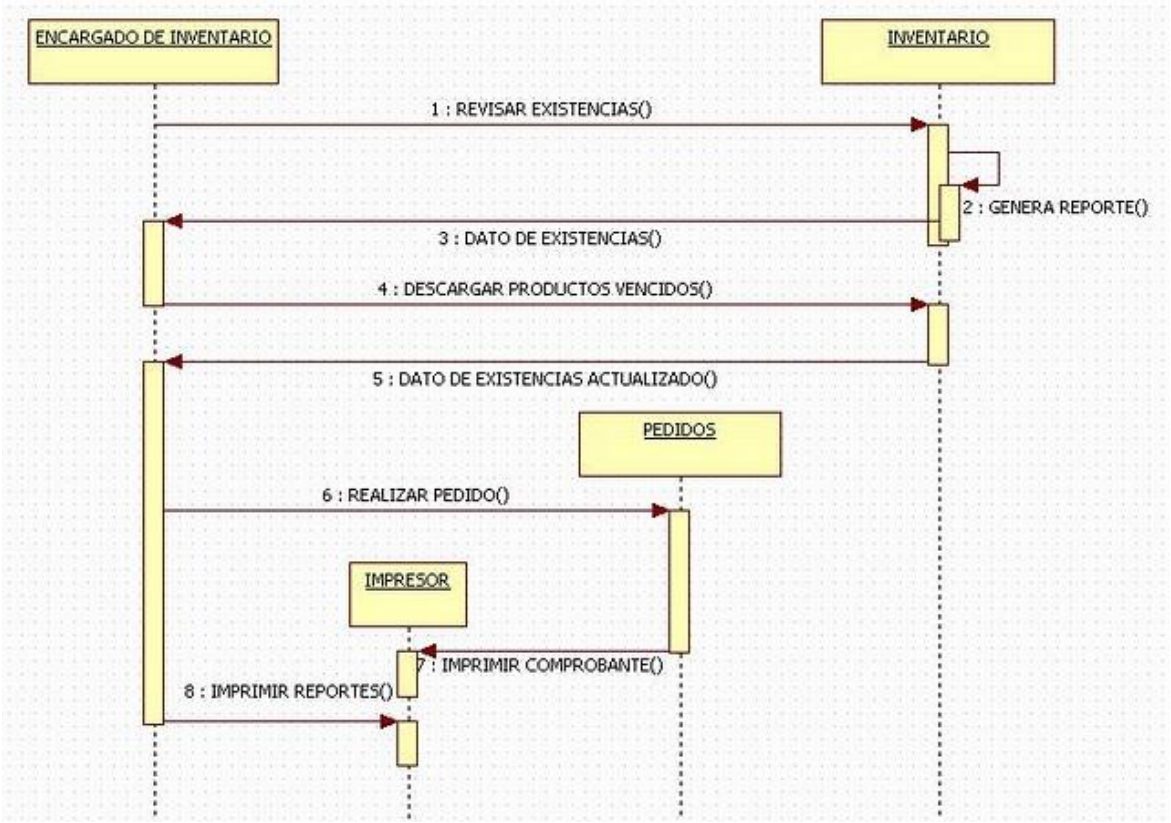


Diagrama de secuencia

Ilustración 17

Fuente: Diego Báez

DIAGRAMA DE COLABORACION

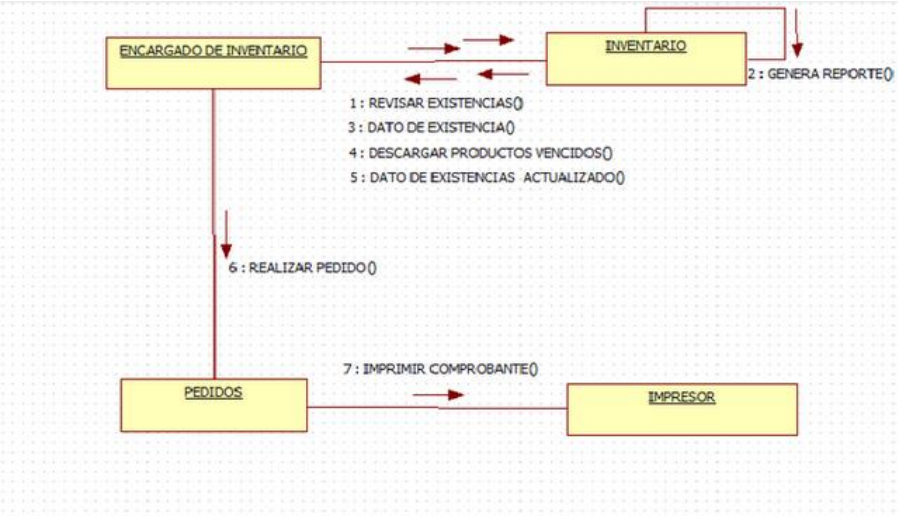


Diagrama de Colaboración
Ilustración 18
Fuente: Diego Báez

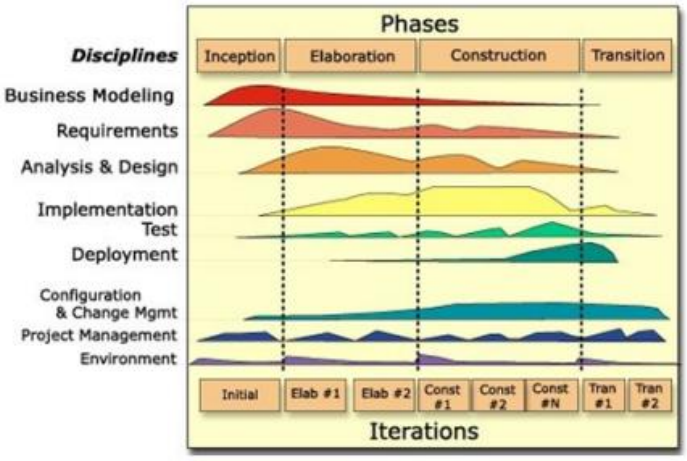
4.8 Descripcion de metodología de desarrollo

Una metodología define formalmente el proceso que se va a utilizar para recolectar los requerimientos, analizarlos, y diseñar una aplicación que cumpla con ellos en cada uno de sus puntos. Hay muchas metodologías, cada una difiere de alguna manera o maneras de las otras.

Esta es la organización dinámica del proceso a lo largo del tiempo.

El ciclo de vida del software se divide en ciclos, cada ciclo trabaja en una nueva generación del producto.

El modelo iterativo del gráfico muestra cómo se estructura el proceso en dos dimensiones.



Metodología de desarrollo
Ilustración 19
Fuente: Diego Báez

El RUP divide un ciclo de desarrollo en cuatro fases consecutivas:

1) Fase de Inicio (InceptionPhase)

2) Fase de elaboración (ElaborationPhase)

3) Fase de construcción (ConstructionPhase)

4) Fase de transición (TransitionPhase)

Fase de Inicio:

- Documento de visión del proyecto: Es una visión general de los requerimientos principales del mismo, características principales y restricciones principales.
- Modelo de casos de uso inicial: (Completado en un 10% – 20%)

- Caso de negocio inicial: Debe incluir el contexto del negocio, criterios de éxito (proyecciones de renta, reconocimiento del mercado, etc.), y proyección financiera. Evaluación de riesgo inicial.
- Planificación del proyecto: Que muestre las fases e iteraciones.
- Modelo de negocio: Si es necesario.
- Uno o varios prototipos

Fase de Elaboración:

- Modelo de casos de uso: Completo al menos en un 80%
- Todos los actores y casos de uso se han identificado, y la mayoría de las descripciones de los casos de uso se han desarrollado.
- Requerimientos suplementarios
  - Se han capturando los requerimientos no funcionales y cualquier requerimiento que no esté asociado con un caso de uso específico.
  - Descripción de la arquitectura del software
  - Prototipo arquitectónico ejecutable del prototipo.
  - Lista de riesgos revisada y un caso de negocios revisado.
  - Planificación del desarrollo para el proyecto global
  - Planificación de grano grueso del proyecto, que muestre las interacciones y el criterio de volución para cada iteración.
  - Una actualización del evento de desarrollo
- Manual de usuario preliminar (opcional).

Fase de Construcción:

- El producto software se ha integrado en las plataformas adecuadas.
- Los manuales de usuario.
- Una descripción de la versión actual.

- Producto listo para ponerlo en las manos del usuario final.

Fase de transición:

- “Beta Testing”, Para validar el nuevo sistema contra las expectativas del usuario.
- Entrenamiento de los usuarios y administradores de mantención.
- Primera aparición pública del producto al mercado, distribución y equipos de venta.

4.9 Modelo Físico

El paso de un modelo lógico a uno físico requiere un profundo entendimiento del manejador de bases de datos que se desea emplear, incluyendo características como:

Conocimiento a fondo de los tipos de objetos (elementos) soportados

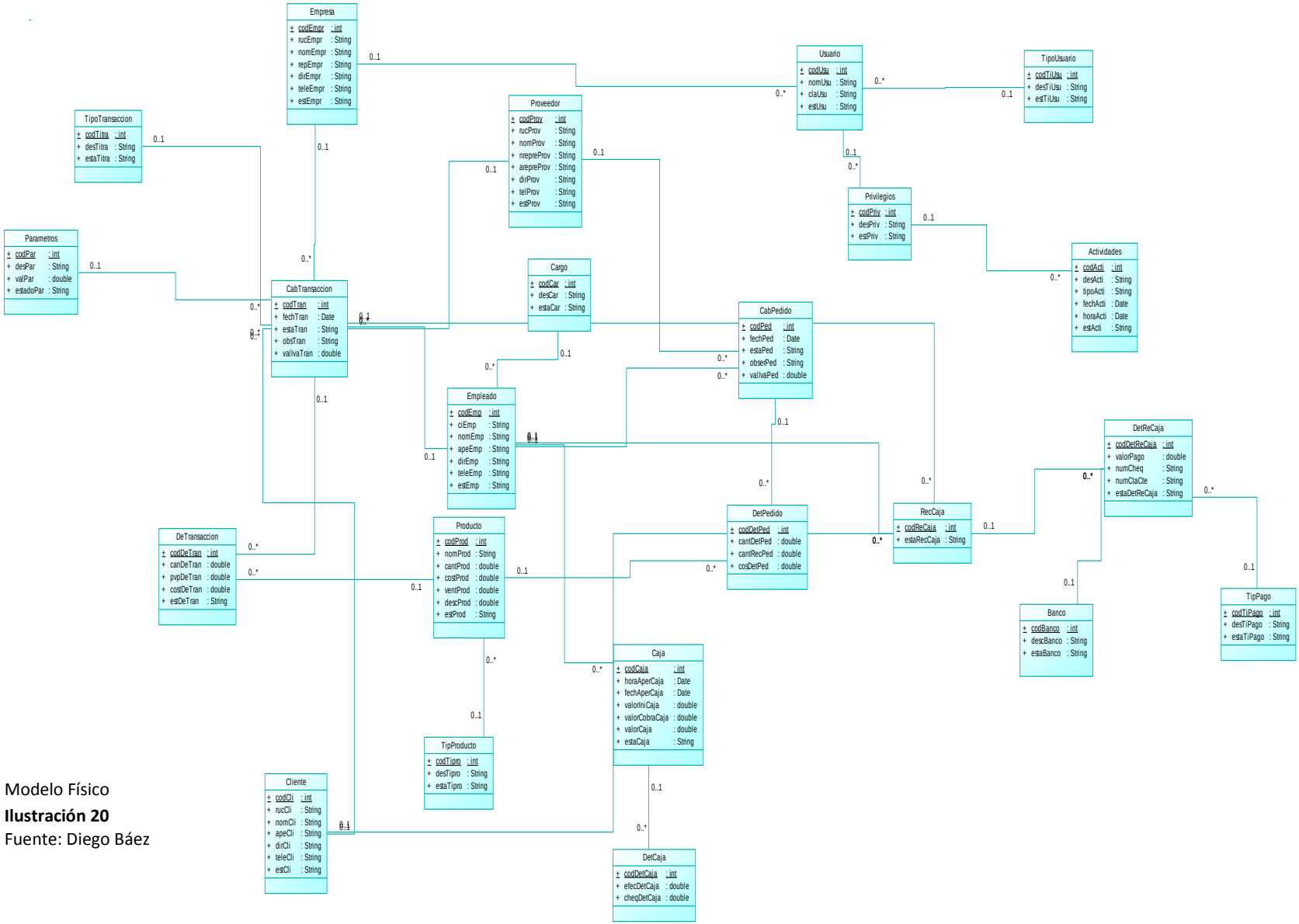
Detalles acerca del indexamiento, integridad referencial, restricciones, tipos de datos, etc

Detalles y variaciones de las versiones

Parámetros de configuración

Data Definition Language (DDL)

Como se comentó en el modelado lógico el paso de convertir el modelo a tablas hace que las entidades pasen a ser tablas (más las derivadas de las relaciones) y los atributos se convierten en las columnas de dichas tablas.



Modelo Físico

Ilustración 20

Fuente: Diego Báez

4.10 Diccionario de Datos

Un diccionario de datos es un conjunto de metadatos que contiene las características lógicas y puntuales de los datos que se van a utilizar en el sistema que se programa, incluyendo nombre, descripción, alias, contenido y organización.

En un diccionario de datos se encuentra la lista de todos los elementos que forman parte del flujo de datos de todo el sistema. Los elementos más importantes son flujos de datos, almacenes de datos y procesos. El diccionario de datos guarda los detalles y descripción de todos estos elementos.

Lista de Campos de las Tablas

NOMBRE	CODIGO
codPar	CODPAR
desPar	DESPAR
valPar	VALPAR
estadoPar	ESTADOPAR
codProv	CODPROV
rucProv	RUCPROV
nomProv	NOMPROV
nrepreProv	NREPREPROV

arepreProv	AREPREPROV
dirProv	DIRPROV
telProv	TELPROV
estProv	ESTPROV
codCli	CODCLI
rucCli	RUCCLI
nomCli	NOMCLI
apeCli	APECLI
dirCli	DIRCLI

teleCli	TELECLI
estCli	ESTCLI
codEmpr	CODEMPR
rucEmpr	RUCEMPR
nomEmpr	NOMEMPR
repEmpr	REPEMPR
dirEmpr	DIREMPR
teleEmpr	TELEEMPR
estEmpr	ESTEMPR
codEmp	CODEMP
codCar	CODCAR
ciEmp	CIEMP
nomEmp	NOMEMP
apeEmp	APEEMP

dirEmp	DIREMP
teleEmp	TELEEMP
estEmp	ESTEMP
codTran	CODTRAN
codTitra	CODTITRA
codProv	CODPROV
codEmpr	CODEMPR
codEmp	CODEMP
codCli	CODCLI
codPar	CODPAR
fechTran	FECHTRAN
estaTran	ESTATRAN
obsTran	OBSTRAN
valIvaTran	VALIVATRAN

codDeTran	CODDETRAN
codTran	CODTRAN
codProd	CODPROD
canDeTran	CANDETRAN
pvpDeTran	PVPDETRAN
costDeTran	COSTDETRAN
estDeTran	ESTDETRAN
codTitra	CODTITRA
desTitra	DESTITRA
estaTitra	ESTATITRA
codCar	CODCAR
desCar	DESCAR
estaCar	ESTACAR
codProd	CODPROD

codTipro	CODTIPRO
nomProd	NOMPROD
cantProd	CANTPROD
costProd	COSTPROD
ventProd	VENTPROD
descProd	DESCPROD
estProd	ESTPROD
codTipro	CODTIPRO
desTipro	DESTIPRO
estaTipro	ESTATIPRO
codPed	CODPED
codProv	CODPROV
codEmp	CODEMP
fechPed	FECHPED

estaPed	ESTAPED
obserPed	OBSERPED
valIvaPed	VALIVAPED
codDetPed	CODDETPED
codProd	CODPROD
codPed	CODPED
cantDetPed	CANTDETPED
cantRecPed	CANTRECPED
cosDetPed	COSDETPED
codUsu	CODUSU
codTiUsu	CODTIUSU
codEmpr	CODEMPR
nomUsu	NOMUSU
claUsu	CLAUSU

estUsu	ESTUSU
codTiUsu	CODTIUSU
desTiUsu	DESTIUSU
estTiUsu	ESTTIUSU
codPriv	CODPRIV
codUsu	CODUSU
desPriv	DESPRIV
estPriv	ESTPRIV
codActi	CODACTI
codPriv	CODPRIV
desActi	DESACTI
tipoActi	TIPOACTI
fechActi	FECHACTI
horaActi	HORAACTI

estActi	ESTACTI
codCaja	CODCAJA
codEmp	CODEMP
horaAperCaja	HORAAPERCAJA
fechAperCaja	FECHAPERCAJA
valorIniCaja	VALORINICAJA
valorCobraCaja	VALORCOBRACAJA
valorCaja	VALORCAJA
estaCaja	ESTACAJA
codDetCaja	CODDETCAJA
codCaja	CODCAJA
efecDetCaja	EFECDETCAJA
cheqDetCaja	CHEQDETCAJA
codReCaja	CODRECAJA

codTran	CODTRAN
codCli	CODCLI
codEmp	CODEMP
estaRecCaja	ESTARECCAJA
codDetReCaja	CODDETRECAJA
codTiPago	CODTIPAGO
codReCaja	CODRECAJA
codBanco	CODBANCO
estaTiPago	ESTATIPAGO
codBanco	CODBANCO
descBanco	DESCBANCO
estaBanco	ESTABANCO

Lista de Campos
Tabla 14
Fuente: Diego Báez

Lista De Índices De Las Tablas

NOMBRE	CODIGO	UNIQUE	PRIMARY	FOREIGN KEY	TABLA
PARAMETROS_PK	PARAMETROS_PK	X	X		Parametros
PROVEEDOR_PK	PROVEEDOR_PK	X	X		Proveedor
CLIENTE_PK	CLIENTE_PK	X	X		Cliente
EMPRESA_PK	EMPRESA_PK	X	X		Empresa
EMPLEADO_PK	EMPLEADO_PK	X	X		Empleado
CARGO_EMPLEADO_FK	CARGO_EMPLEADO_FK			X	Empleado
CABTRANSACCION_PK	CABTRANSACCION_PK	X	X		CabTransaccion
EMPRESA_TRANSACCION_FK	EMPRESA_TRANSACCION_FK			X	CabTransaccion
PARAMETROS_TRANSACCION_FK	PARAMETROS_TRANSACCION_FK			X	CabTransaccion
PROVEEDOR_TRANSACCION_FK	PROVEEDOR_TRANSACCION_FK			X	CabTransaccion
CLIENTE_TRANSACCION_FK	CLIENTE_TRANSACCION_FK			X	CabTransaccion
EMPLEADO_TRANSACCION_FK	EMPLEADO_TRANSACCION_FK			X	CabTransaccion
TIPOTRANSACCION_CABTRANSACCION_FK	TIPOTRANSACCION_CABTRANSACCION_FK			X	CabTransaccion
DETRANSACCION_PK	DETRANSACCION_PK	X	X		DeTransaccion
TRANSACCION_DETTRANSACCION_FK	TRANSACCION_DETTRANSACCION_FK			X	DeTransaccion
PRODUCTO_DETTRANSACCION_FK	PRODUCTO_DETTRANSACCION_FK			X	DeTransaccion
TIPOTRANSACCION_PK	TIPOTRANSACCION_PK	X	X		TipoTransaccion
CARGO_PK	CARGO_PK	X	X		Cargo
PRODUCTO_PK	PRODUCTO_PK	X	X		Producto
TIPOPRODUCTO_PRODUCTO_FK	TIPOPRODUCTO_PRODUCTO_FK			X	Producto

TIPPRODUCTO_PK	TIPPRODUCTO_PK	X	X		TipProducto
CABPEDIDO_PK	CABPEDIDO_PK	X	X		CabPedido
PROVEEDOR_CABPEDIDO_FK	PROVEEDOR_CABPEDIDO_FK			X	CabPedido
EMPLEADO_CABPEDIDO_FK	EMPLEADO_CABPEDIDO_FK			X	CabPedido
DETPEDIDO_PK	DETPEDIDO_PK	X	X		DetPedido
CABPEDIDO_DETPEDIDO_FK	CABPEDIDO_DETPEDIDO_FK			X	DetPedido
PRODUCTO_DETPEDIDO_FK	PRODUCTO_DETPEDIDO_FK			X	DetPedido
USUARIO_PK	USUARIO_PK	X	X		Usuario
TIPOUSUARIO_USUARIO_FK	TIPOUSUARIO_USUARIO_FK			X	Usuario
EMPRESA_USUARIO_FK	EMPRESA_USUARIO_FK			X	Usuario
TIPOUSUARIO_PK	TIPOUSUARIO_PK	X	X		TipoUsuario
PRIVILEGIOS_PK	PRIVILEGIOS_PK	X	X		Privilegios
USUSARIO_PRIVILEGIOS_FK	USUSARIO_PRIVILEGIOS_FK			X	Privilegios
ACTIVIDADES_PK	ACTIVIDADES_PK	X	X		Actividades
PRIVILEGIOS_ACTIVIDADES_FK	PRIVILEGIOS_ACTIVIDADES_FK			X	Actividades
CAJA_PK	CAJA_PK	X	X		Caja
EMPLEADO_CAJA_FK	EMPLEADO_CAJA_FK			X	Caja
DETCAJA_PK	DETCAJA_PK	X	X		DetCaja
CAJA_DETCAJA_FK	CAJA_DETCAJA_FK			X	DetCaja
RECCAJA_PK	RECCAJA_PK	X	X		RecCaja
EMPLEADO_RECCAJA_FK	EMPLEADO_RECCAJA_FK			X	RecCaja
TRANSACCION_CAJA_FK	TRANSACCION_CAJA_FK			X	RecCaja
CLIENTE_RECCAJA_FK	CLIENTE_RECCAJA_FK			X	RecCaja
DETRECAJA_PK	DETRECAJA_PK	X	X		DetReCaja
TIPOPAGO_DETRECCAJA_FK	TIPOPAGO_DETRECCAJA_FK			X	DetReCaja
BANCO_DETRECCAJA_FK	BANCO_DETRECCAJA_FK			X	DetReCaja
TIPPAGO_PK	TIPPAGO_PK	X	X		TipPago
BANCO_PK	BANCO_PK	X	X		Banco

Índice de tablas

Tabla 15

Fuente: Diego Báez

Lista de referencias

Nombre	Código	Tabla	Tabla Hija
Banco_DetRecCaja	BANCO_DETRECCAJA	Banco	DetReCaja
CabPedido_DetPedido	CABPEDIDO_DETPEDIDO	CabPedido	DetPedido
Caja_DetCaja	CAJA_DETCAJA	Caja	DetCaja
Cargo_Empleado	CARGO_EMPLEADO	Cargo	Empleado
Cliente_RecCaja	CLIENTE_RECCAJA	Cliente	RecCaja
Cliente_Transaccion	CLIENTE_TRANSACCION	Cliente	CabTransaccion
Empleado_CabPedido	EMPLEADO_CABPEDIDO	Empleado	CabPedido
Empleado_Caja	EMPLEADO_CAJA	Empleado	Caja
Empleado_RecCaja	EMPLEADO_RECCAJA	Empleado	RecCaja
Empleado_Transaccion	EMPLEADO_TRANSACCION	Empleado	CabTransaccion
Empresa_Transaccion	EMPRESA_TRANSACCION	Empresa	CabTransaccion
Empresa_Usuario	EMPRESA_USUARIO	Empresa	Usuario
Parametros_Transaccion	PARAMETROS_TRANSACCION	Parametros	CabTransaccion
Privilegios_Actividades	PRIVILEGIOS_ACTIVIDADES	Privilegios	Actividades
Producto_DetPedido	PRODUCTO_DETPEDIDO	Producto	DetPedido
Producto_DetTransaccion	PRODUCTO_DETTRANSACCION	Producto	DeTransaccion
Proveedor_CabPedido	PROVEEDOR_CABPEDIDO	Proveedor	CabPedido

Proveedor_Transaccion	PROVEEDOR_TRANSACCION	Proveedor	CabTransaccion
RecCaja_DetReCaja	RECCAJA_DETRECAJA	RecCaja	DetReCaja
TipoPago_DetRecCaja	TIPOPAGO_DETRECCAJA	TipPago	DetReCaja
TipoProducto_Producto	TIPOPRODUCTO_PRODUCTO	TipProducto	Producto
TipoTransaccion_CabTransaccion	TIPOTRANSACCION_CABTRANSACCION	TipoTransaccion	CabTransaccion
TipoUsuario_Usuario	TIPOUSUARIO_USUARIO	TipoUsuario	Usuario
Transaccion_Caja	TRANSACCION_CAJA	CabTransaccion	RecCaja
Transaccion_DetTransaccion	TRANSACCION_DETTRANSACCION	CabTransaccion	DeTransaccion
Ususario_Privilegios	USUSARIO_PRIVILEGIOS	Usuario	Privilegios

Índice de tablas
Tabla 16
Fuente: Diego Báez

Lista de tablas

Name	Code
Actividades	ACTIVIDADES
Banco	BANCO
CabPedido	CABPEDIDO
CabTransaccion	CABTRANSACCION
Caja	CAJA
Cargo	CARGO
Cliente	CLIENTE

DetCaja	DETCAJA
DetPedido	DETPEDIDO
DeTransaccion	DETRANSACCION
DetReCaja	DETRECAJA
Empleado	EMPLEADO
Empresa	EMPRESA
Parametros	PARAMETROS
Privilegios	PRIVILEGIOS
Producto	PRODUCTO
Proveedor	PROVEEDOR
RecCaja	RECCAJA
TipoTransaccion	TIPOTRANSACCION
TipoUsuario	TIPOUSUARIO
TipPago	TIPPAGO
TipProducto	TIPPRODUCTO
Usuario	USUARIO

Lista de tablas
Tabla 17
Fuente: Diego Báez

4.11 Estándares de Programación

Dentro de la aplicación ya sea Windows o Web, vamos a tener distintos tipos de objetos, pueden ser: variables, constantes, procedimientos, etc.; los nombres de los mismos estarán compuestos de un prefijo seguido del nombre del objeto.

Clases

Los nombres de las clases deben de poseer un prefijo en conjunto con el nombre propio de la clase.

<nombre de la capa><Nombre>

El nombre de la clase debe ser especificado por el consultor a cargo de su creación, debe de poseer un nombre significativo. El nombre debe de estar en singular.

Nombre de la Capa:

Por ejemplo:

Objeto (cl)

Multi-Objeto (mo)

Base Datos(bd)

Webforms (wf)

Por ejemplo:

Si queremos crear una clase que contenga los procedimientos necesarios para traer los datos de la tabla de usuarios, el nombre debe ser algo similar a:

clUsuarios

Procedimientos

Los nombres de los procedimientos deben de contener un conjunto de prefijos seguido del nombre del mismo.

<Nombre>

Los nombres de los procedimientos deben de llevar la primera letra de cada palabra en mayúscula y unidas.

Constantes

Las constantes se dividen en globales y locales, el nombre de la constante posee un prefijo seguido del nombre de la constante definido por el consultor.

Los nombres de constantes siempre se escribirán en MAYUSCULA

Los nombres de constantes deben iniciar con un prefijo que encapsule el módulo o área del programa, por ejemplo FILEMASTER – FM-

Como el nombre de la constante lo va a definir el consultor, se debe de ser lo más descriptivo posible, dejando a entender el valor del mismo. Si el nombre de la constante ésta compuesta de varias palabras, debemos de digitar las palabras unidas como nombre de la constante, las palabras deben tener su primera letra en mayúscula, excepto la primera palabra.

Por ejemplo:

Si vamos a necesitar una constante que va a contener un primer nombre (dos palabras) de una persona, debe ser como el siguiente ejemplo:

PRIMER_NOMBRE

FM_AGREGAR_REGISTRO

FM_FICHA_CAMPO_NUMERICO

FM_ESTADO_ACTIVO = ‘A’

Variables

Las variables dependiendo de su alcance se dividen en globales y locales, el nombre de la variable consta de un prefijo seguido del nombre de la variable definido por el consultor.

El nombre debe de tener la siguiente sintaxis:

<ubicación><tipo de dato>_<nombre de la variable>

Los prefijos son los siguientes:

Alcance:

Global (g)

Local (l), todas las variables locales pueden omitir el uso de este prefijo

Tipo de Dato

Nombre	**Prefijo**	**Ejemplo**
String	s	sNombre, sApellido
Char	c	cGenero, cEstado
Numerico	n	nNumFact
Integer	i	iCodCli
List	lst	lstClientes, lstEmpleado
DateTime	dt	dtActividad, dtFactura

Date	dt	dtFactura
Time	dt	dtFactura

Tipos de datos
Tabla 18
Fuente: Diego Báez

Como el nombre de la variable lo va a definir el consultor, debe ser lo más claro posible, dejando a entender el valor del mismo. Si el nombre de la variable consta de varias palabras, debemos de digitar las palabras unidas como nombre de la variable, las palabras deben tener su primera letra en mayúscula, excepto la primera palabra.

Por ejemplo:

Si vamos a necesitar una variable que va a contener un primer nombre (dos palabras) de una persona, debe ser como el siguiente ejemplo:

lsNombreEmpleado

Si necesitamos crear una variable que va a contener un estado (una palabra), debe ser como el siguiente ejemplo:

lsEstado o sEstado

Posición dentro del Código Fuente

Todas las variables sin distinción deben quedar declaradas en al cabecera o header del procedimiento, función o evento. A pesar que el código fuente permita otra acción se requiere que todas queden declaradas en la posición indicada.

Ejemplo

Function XYZ

DimnIAsInteger = 1

DimnAnchoAsInteger = 500

DimnAltoAsInteger = 400

Un mal ejemplo

Function XYZ

DimnIAsInteger = 1

DimnAnchoAsInteger = 500

IfllamarDatos() then

endif

DimnAltoAsInteger = 400

Controles Visuales

Los controles visuales deben poseer un nombre estándar, el cual se identificara con un prefijo todo en minúscula, seguido por el nombre del control.

Nombre	**Prefijo**	**Ejemplo**
Label	Lb	lbNombre, lbApellido
TextBox	Txt	TxtNombre, txtDirección

ComboBox	Cb	cbCargo, cbDepartamento
ListBox	Lst	lstProveedor
DialogBox	dlg	dlgTerminar
Checkbox	Ch	chEstado, chGenero
CommandButton	cmd	cmdNuevo, cmbEliminar

Controles Visuales
Tabla 19
Fuente: Diego Báez

Documentación interna

Procedimientos y Funciones

Para la declaración de los procedimientos debemos de seguir el siguiente estándar para los mismos.

NOMBRE:

FECHA Y CREADOR:

DESCIPCION

DETALLE:

MODIFICACION

Este encabezado debe ser agregado una línea antes de iniciar el procedimiento o la función.

Código fuente comentado

Se entiende como código fuente comentado aquellas líneas de código fuente que fueron puestas entre comentarios por fines de reemplazo de código. Este código comentado no tiene funcionalidad alguna dentro del sistema por lo tanto debe ser eliminado por completo del mismo, solo en casos excepcionales

Por ejemplo

```
'cargarSegundoCampo()

'‘' Me.cmdCondicionVariable.Enabled = False

' cmdSegCampos.Enabled = False
```

Las anterior líneas comentadas no deben pertenecer al código fuente estable que se usará para realizar una compilación.

Esta virtud de programación será altamente evaluada durante el proceso de auditoria.

Nombre de las funciones y procedimientos

El nombre de una función o procedimiento debe contener las siguientes especificaciones.

Inicia en mayúscula.

Identifica claramente el contenido.

Es en español.

Inicia con un verbo.

Ejemplo

```
ObtenerCodigoFicha ()
```

ImprimirDocumento ()

VerificarCodigosAceptados ()

Estándares de diseño de bases de datos: 18/10/06 11:46:46

Estándares de diseño de Bases de datos

El uso de estos estándares tiene innumerables ventajas, entre ellas:

Asegurar la legibilidad del modelo de datos, inclusive para personas que no están relacionadas con el ambiente informático, en etapas de análisis y diseño; Facilitar la portabilidad entre motores de bases de datos, plataformas y aplicaciones; Facilitar la tarea de los programadores en el desarrollo de los sistemas.

Es por esto que la codificación de las tablas de las bases de datos a desarrollar debe cumplir ciertos requisitos, detallados en el presente documento. Estos requisitos pueden aplicarse a cualquier motor de bases de datos.

Brevemente se resume en los siguientes puntos:

Reglas generales

Los nombres de tablas y campos deben especificarse bajo el estándar camelCase. Este estándar especifica escribir las palabras compuestas eliminando los espacios y poniendo en mayúscula la primera letra de cada palabra. En este ámbito se utilizará la variante lowerCamelCase (la primer letra del nombre, en minúscula).

Únicamente se utilizarán caracteres alfabéticos, salvo que por la naturaleza del nombre se necesiten dígitos numéricos. Se prohíbe el uso de caracteres de puntuación o símbolos.

Ejemplo: localidadesCenso2003.

Las letras acentuadas se reemplazarán con las equivalentes no acentuadas, y en lugar de la letra eñe (ñ) se utilizará (ni).

Ejemplos: anioExpediente, montoSenia.

El nombre elegido debe ser lo más descriptivo posible, evitando términos ambiguos o que se presten a distintas interpretaciones.

Ejemplo: tiposMunicipios =>categoriasMunicipios.

El nombre no debe abreviarse, salvo que por necesidad específica deban especificarse más de una palabra en el mismo.

Ejemplo: ido =>idOrganismo, freg =>fechaRegistro

Agregar comentarios a las bases de datos y los campos, sobre todo a los booleanos.

Tablas

Reglas generales

Los nombres deben especificarse en plural, y de acuerdo a las reglas generales.

Ejemplos: departamentos, facturas, monedas.

En el caso de tablas que se relacionan específicamente con otra tabla (ej. tablas tipo, nomencladores, entidades débiles), esta relación debe quedar expresada en el nombre.

Ejemplos: domiciliosPersonas, categoriasMunicipios.

Las tablas de relación (objetos asociativos, representan relaciones de N a M) deben nombrarse utilizando los nombres de las tablas intervinientes, siguiendo un orden lógico de frase.

Ejemplos: localidadesMunicipios, facturasNotas

Campos clave (Identificadores de tabla)

Toda tabla debe poseer uno o más campos clave.

Toda relación entre tablas debe implementarse mediante constraints (claves foráneas) con integridad referencial, de acuerdo al motor de base de datos utilizado.

La integridad referencial deberá actualizar en cascada en todos los casos, y restringir el borrado salvo para las entidades débiles.

Ejemplos: no se podrá eliminar un registro de la tabla localidades que tenga ocurrencias en otras tablas; para este caso deberá implementarse el borrado lógico. Por el contrario, sí podrá habilitarse el borrado en cascada si la relación fuera entre las tablas facturas y renglones

El nombre del campo clave debe estar compuesto por “id” + nombre de la tabla en singular (para claves no compuestas). Dependiendo de la naturaleza de la entidad, el nombre de la tabla a usar es el de la misma tabla, o el de la relacionada.

Ejemplos: tabla localidades =>idLocalidad.

Las claves compuestas sólo deben utilizarse en casos específicos, por ejemplo, tablas de relación o entidades débiles. Si una tabla X con clave compuesta necesita ser referenciada desde otra tabla Y, deberá generarse un campo clave en X al inicio de la misma como “idX”, y generar un índice único en los campos que la identificaban.

Otros campos

Todo campo que represente un nombre o descripción, se colocará inmediatamente después de los campos clave, y se nombrará como a la tabla a la que pertenece, en singular.

Ejemplos: tabla localidades =>idLocalidad, localidad.

tablasucursalesEmpresas =>idEmpresa, idSucursal, sucursal

Algunos campos que representan datos, de acuerdo a su representación conceptual en el ámbito del negocio, deberán prefijarse de la siguiente manera:

– Números: num (ejemplo: Número de factura =>numFactura)

– Fechas: fecha (ejemplo: Fecha de inscripción =>fechaInscripcion)

– Códigos: codigo (ejemplo: Código de producto: codigoProducto)

Los campos booleanos deberán nombrarse de acuerdo al estado correspondiente al valor

1/Verdadero/True de los mismos.

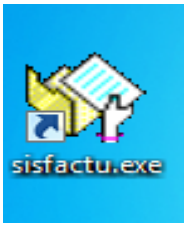
Ejemplos: autorizado, oculto, vigente.

Los campos de relación (foreignkeys, claves foráneas) deben nombrarse de la misma manera que los campos clave (usando el nombre de la tabla a la que hacen referencia).

Ejemplos: tabla personas =>idTipoDocumento, idEstadoCivil

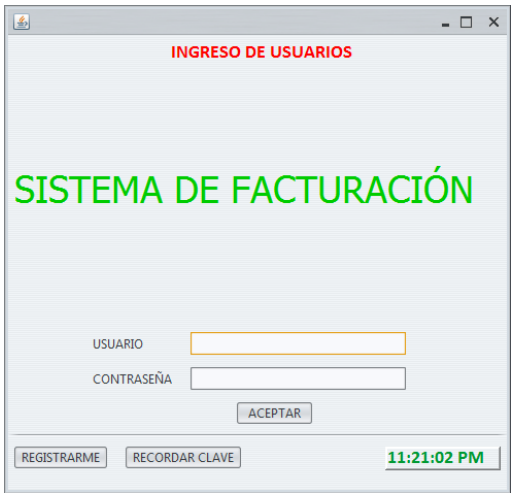
4.12 Prototipo

Para ingresara al sistemas dar doble clic en el icono del escritorio.



Icono Sistema
Ilustración 21
Fuente: Diego Báez

En la primera pantalla muestra la autentificación del usuario.

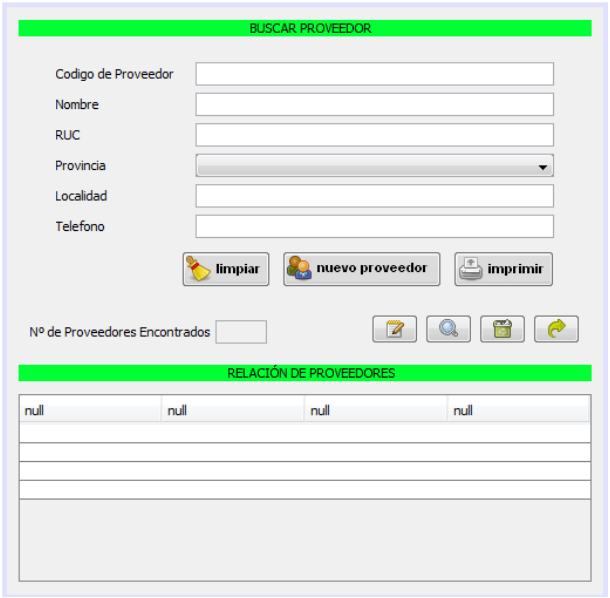


Pantalla inicio de Sesión
Ilustración 22
Fuente: Diego Báez

En la pantalla principal se encuentra el menú con el cual se podrá acceder a las diferentes configuraciones por ejemplo mantenimiento.

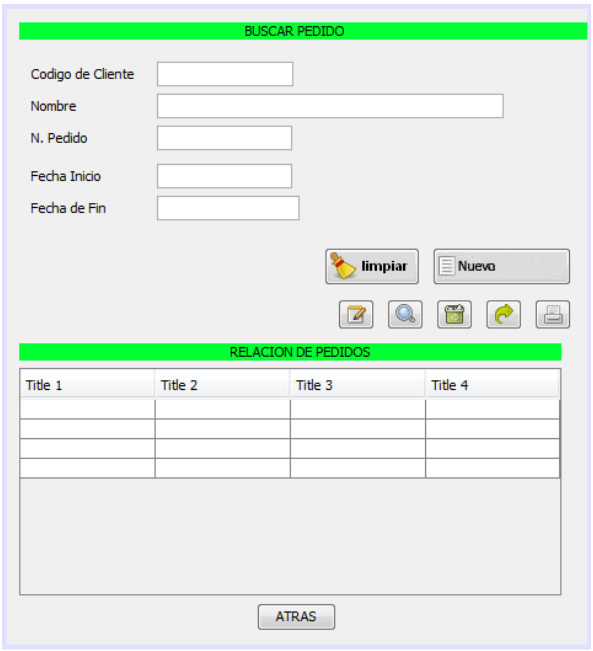
Menú Mantenimiento
Ilustración 23
Fuente: Diego Báez

La pantalla de mantenimiento contiene un menú con el cual se podrá acceder a crear clientes, proveedores, cargos, entre otros.



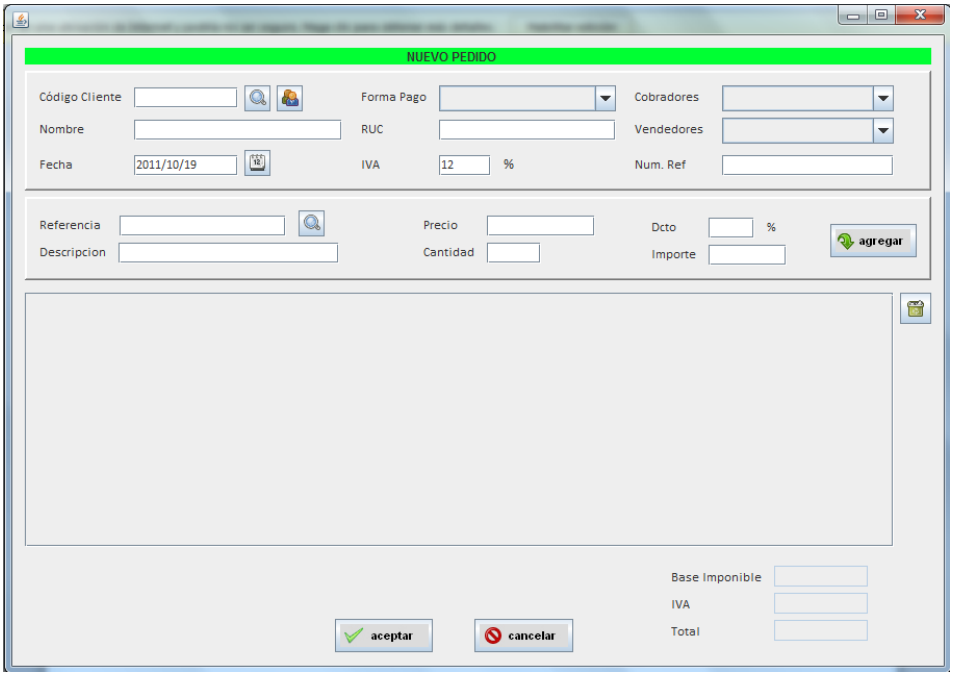
Buscar Proveedor
Ilustración 24
Fuente: Diego Báez

Para registrar un nuevo pedido ingresa al formulario dando clic en icono o imagen que muestra el menú una vez en el formulario se ingresara todos los datos para generar un nuevo registro.



Buscar Pedido
Ilustración 25
Fuente: Diego Báez

Para generar un nuevo pedido dar un clic en el botón nuevo en la cual nos mostrara la siguiente pantalla para ingresar los datos del pedido:



Buscar Pedido
Ilustración 26
Fuente: Diego Báez

4.13 Pruebas y Depuración

1. Pruebas de Unidad

- Se concentra en la verificación de la unidad más pequeña del diseño del software: el componente o módulo del software.

  La verificación en el diseño se realizó determinando que exista las relaciones entre cada una de las tablas, poniendo especial énfasis en las tablas que contienen la información que almacena los datos de los productos, así como también la información de clientes y proveedores.

- Las pruebas de unidad se concentran en la lógica del negocio los procesos internos que se relacionan las acciones de Comando y Control que se consideren con la finalidad que la relación entre las

tablas y almacenamientos de información pueda mantener una estructura adecuada y lógica.

- Este tipo de prueba se la realizo y el sistema mantiene su lógica, se consideró además que se mantenga la integridad de la información y sus datos, poniendo especial énfasis en las relaciones claves primarias y foráneas del modelo lógico del negocio.

2. Pruebas de Integración

- Esta prueba la realice individualmente verificando que todo funciona bien individualmente, por lo tanto las relaciones existentes en el modelo establecido se verificaron la consistencia del modelo indicando las claves primarias y foráneas establecidas, a fin de poder verificar la indexación de las búsquedas de las direcciones de los clientes.
- La prueba de integración es una técnica sistemática para construir la arquitectura del software, mientras, al mismo tiempo, se aplican las pruebas para descubrir errores asociados con la interfaz. La integración de los datos en la tabla de los clientes, determina la robustez del modelo colocando especial énfasis en las asociaciones de las tablas que tienen relación con la información que maneja los datos de transacciones.

3. Pruebas de Validación

- Las pruebas de validación empiezan tras la culminación de la prueba de integración, cuando se han ejercitado los componentes individuales. Se ha terminado de ensamblar el software como paquete y se han descubierto y corregido los errores de interfaz.

  Este tipo de pruebas se la realizo en el módulo de seguridad establecido para poder controlar el acceso de los usuarios al

sistema. También se estableció para poder verificar la integración que existe de los datos transaccionales con los datos operacionales.

- La prueba se concentra en las acciones visibles para el usuario y en la salida del sistema que éste puede reconocer.

  Este tipo de prueba se realizó con el módulo de seguridad poniendo énfasis en los tipos de usuarios que tiene que manejar el sistema

- Criterios de la prueba de validación

  Se verifico que existan mensajes de verificación de usuarios al sistema, a través de ventanas de alertas y de precaución. La validación del software se logra mediante una serie de pruebas que demuestren que se cumple los requisitos.

  Las pruebas realizadas determinan que cumple con los requisitos ya que los mismos ejecutan filtros y criterios de búsqueda razonables y efectivos a fin mejorar la atención al cliente.

4. Pruebas del Sistema

- Durante el diseño y la prueba del software se mejora en gran medida la probabilidad de tener éxito en la integración del software del sistema mayor.

  La integración de los datos y las relaciones entre cada uno de las tablas se verifico la consistencia de los datos y la integridad de los mismos.

5. Prueba de seguridad

- La interrupción abarca un amplio rango de actividades:

Se colocó interrupciones en la ejecución de los programas para comprobar que su ejecución sea idónea, igualmente se colocó en el código interrupciones para verificar sus errores, en las ubicación de los puntos de referencia, también como se están integrando los datosingresados al sistema ya que se tiene que mantener una estabilidad de información.

6. Prueba de Interfaces Gráficas de Usuario (GUI , GraphicalUser Interface)

- Uso de una lista de chequeo preestablecida:

  Se realizó una prueba con la ejecución del manejo de los iconos de acceso al sistema y la integración con la base de, la lista de productos en los reportes respectivos así como también lele stock existente, y la integración con su respectiva consola.

- Entrada de datos:

  El ingreso de los datos se realizara por medio de los datos del cliente como la cedula el mismo que es único para cada uno de los registros.

7. Pruebas del Sistema

- Prueba de resistencia y consistencia

  Se verifico que exista la relación y consistencia entre el sistema de información y la base de datos lo que significa que la integración con la base de datos cumplan con su requerimiento de integración de las dos.

4.14 Instalación

Para la instalación del sistema se deberá tener en cuenta los requisitos previos:

Requisitos ideales para PC Servidor.

Requisitos Mínimos	
Procesador	Intel Pentium IV a 3.4 GHz o superior
Sistema Operativo	▪ Microsoft Windows 2003, con el último Service Pack de Windows y las actualizaciones importantes disponibles en la página Web Microsoft Seguridad
Memoria RAM	2 GB
Disco Duro	120 GB
Monitor	Resolución: 1280 × 1024 pixeles, color verdadero
Dispositivo de entrada	Teclado y Mouse
Otros	▪ Microsoft Internet Explorer 5.01 o posterior ▪ Microsoft Data Access Components 2.6 ▪ IIS 5.0

Cuadro de hardware
Tabla 20
Fuente: Diego Báez

Requisitos ideales para PC Cliente.

Requisitos Mínimos	
Procesador	Intel Pentium IV a 3.4 GHz o superior
Sistema Operativo	▪ Microsoft Windows 2003, con el último Service Pack de Windows y las actualizaciones importantes disponibles en la página Web Microsoft Seguridad
Memoria RAM	2 GB
Disco Duro	120 GB
Monitor	Resolución: 1280 × 1024 pixeles, color verdadero
Dispositivo de entrada	Teclado y Mouse

Cuadro de hardware
Tabla 21
Fuente: Diego Báez

Configuración e implementación del servidor

Instalar AppServ

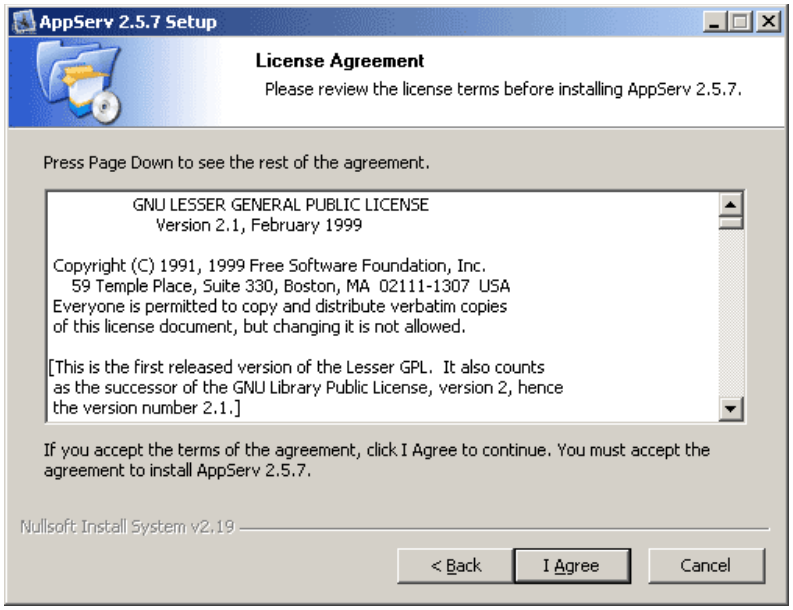
Eso es lo mejor, es muy fácil de hacer, basta con un doble click en el instalador que nos bajamos de la Web y luego damos siguiente, siguiente, varias veces hasta que queda instalado. Pero bueno este tutorial no tiene sentido si es tan fácil, por tanto para hacer como que trabajo, vamos a ver paso a paso esas pantallas donde solo basta con dar "siguiente".

1) Damos doble click en el icono del instalador que nos bajamos. Aparece la imagen que vemos debajo, recuerda que la versión puede ser más nueva cuando tú intentes ;), pero tranquilo se usara igual que como comento ahora. Debemos dar, siguiente (next).

Pantalla de instalación

Ilustración 27
Fuente: Diego Báez

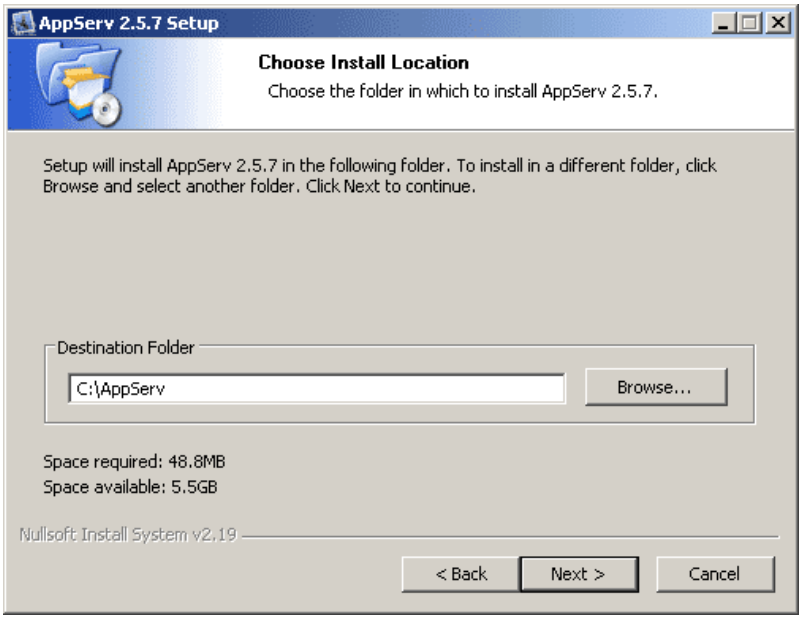
2) Aparece el acuerdo de licencia, ahí debemos dar conformidad (I Agree)



Licencia

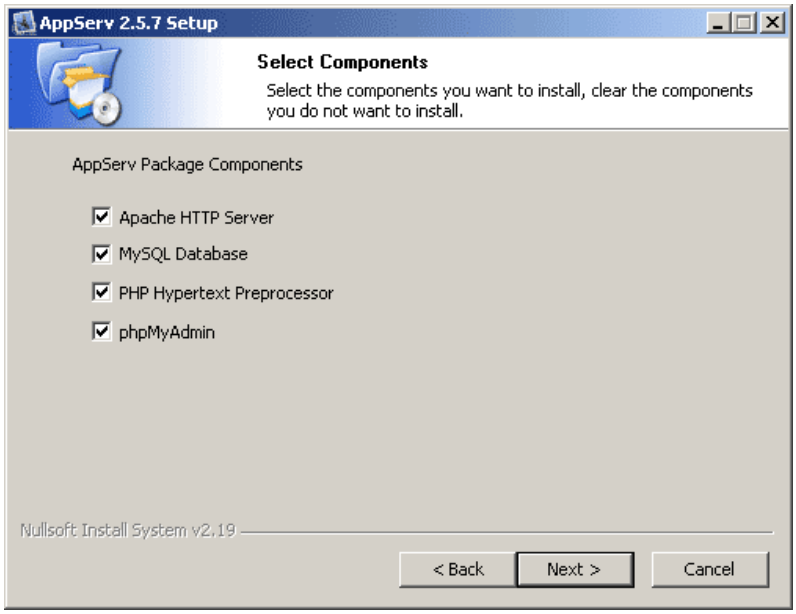
Ilustración 28
Fuente: Diego Báez

3) En este punto, debemos elegir en cual de nuestros discos duros y en que carpeta vamos a instalar el AppServ, directorio recomendado por defecto, o sea: c:\AppServ, como muestra la imagen, y le damos siguiente (Next)



Destino de la Instalación
Ilustración 29
Fuente: Diego Báez

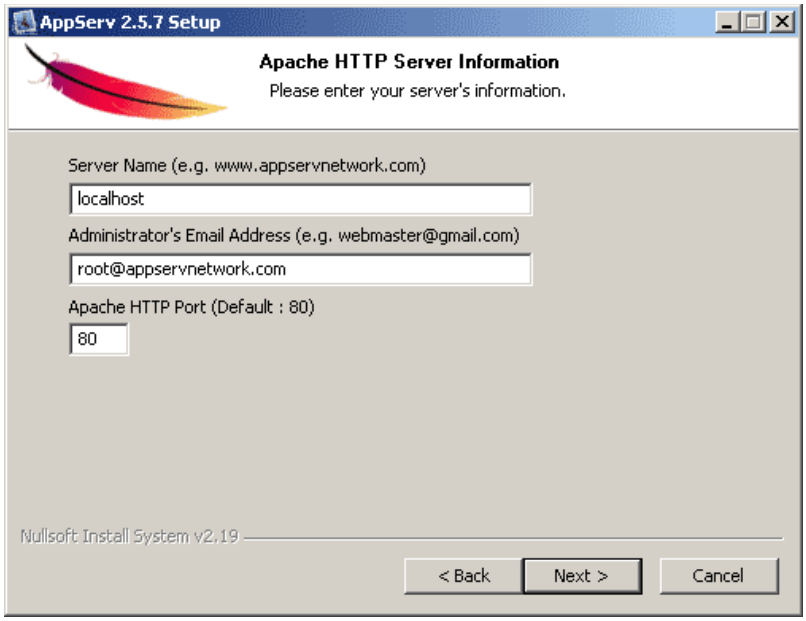
4) En este punto debemos elegir los componentes que vamos a instalar (recuerda que todos los pasos basta con dar siguiente siguiente), como ves aparecen marcados los cuatro elementos disponibles (apache, mysql, php, phpmyadmin) si bien puedes desmarcar alguno, es mejor instalar todos para tener completo tu equipo de testeo y desarrollo Web, le damos siguiente (Next)



Componentes a Instalar

Ilustración 30
Fuente: Diego Báez

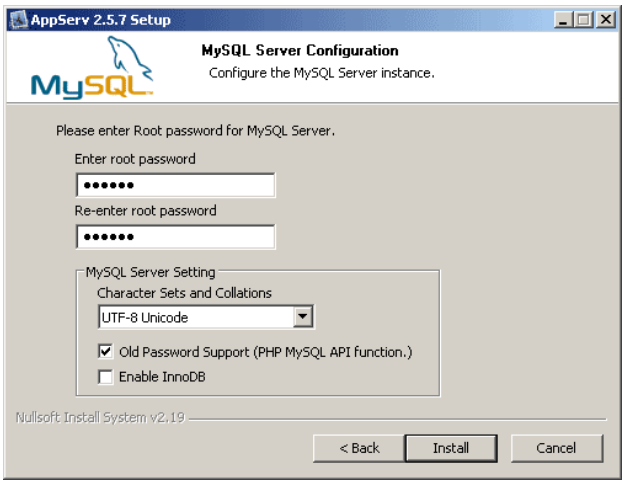
5) En este punto debes indicar el nombre del servidor y el email del administrador del sistema, en mi caso dejo localhost y email elquique@localhost.com, respectivamente, el puerto 80. Recuerda que puedes dar siguiente y dejar lo que sugiere sin problemas.



Punto de Inicio

Ilustración 31
Fuente: Diego Báez

6) Configuración MySQL, en este punto debemos prestar atencion, ya que debemos indicar el Password, para el usuario principal de la base de datos, lo escribimos 2 veces !! Y por favor lo recuerdan (no lo anoten bajo el teclado como mi tía Clotilde). Lo demás lo dejamos como esta en la imagen.

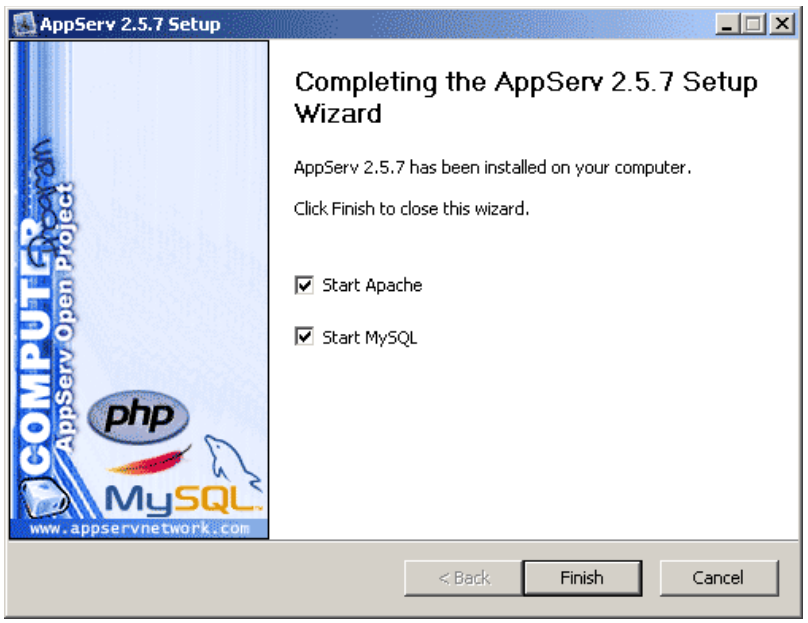


Configuración MySQL

Ilustración 32
Fuente: Diego Báez

Por tanto el usuario será root, que es por defecto (tu luego puedes crear mas, todos los que quieras) y tu Contraseña que acabas de escribir, eso lo usaras para abrir la base desde tus programas PHP, por eso es importante recordar ;)
Le damos siguiente (Next)

7) Para culminar la instalación, como indica la imagen, podemos elegir si al finalizar ya comenzara a funcionar el Server Apache y el servidor de base de datos MySQL, lo dejamos marcado y le damos finalizar (Finish), listo hemos terminado.



Iniciar los servicios

Ilustración 33
Fuente: Diego Báez

En este punto si vas a tu navegador (FireFox), puedes poner, http://localhost y debe mostrarse tal como muestra la imagen.

Servicios iniciados en el servidor

Ilustración 34
Fuente: Diego Báez

4.15 Recopilación y Carga de Datos

Para cargar los datos en el software se debe tomar en cuenta el lugar donde se encuentra la información de los cursos almacenada antes de implementar el sistema, se procederá a realizar la respectiva migración, pero en este caso lamentablemente no existen registros almacenados en una base de datos, esto implica que se debe ingresar toda la información de forma manual ya que estos datos se encuentran alojadas en hojas y cuadernos; cabe indicar que los datos no se pasarán en su totalidad siendo este responsabilidad de la empresa.

4.16 Pruebas y Depuración Final en Funcionamiento

La información que se encontraba en hojas y cuadernos ahora están alojados de manera segura en la base de datos del sistema, indicando que el software debe funcionar

correctamente; Un ejemplo seria que al momento de realizar la transacción tenga productos en stock. Otro de los casos es en las validaciones de los textos y números que se deben ingresar.

4.17 Puesta en Marcha del Sistema

Para que el sistema empiece a funcionar los usuarios deberán indicar su nombre y su clave en el caso de no tener deberá pedir al administrador sus datos para ingresar, dependiendo del tipo de usuario el sistema cargara lo que puede realizar como usuario.

4.18 Capacitación al Usuario Final

A través de una solución única y completa para crear, organizar, distribuir, accesar, mantener y compartir el conocimiento en la empresa.

Se entiende que el usuario es único, y es por ello que, basándome en las necesidades específicas defino la metodología de proyectos individuales de capacitación.

Para la capacitación utilizaré los materiales de equipo de implementación:

- Documentación
- Configuración
- Pruebas

Materiales de usuario final

- Tarjetas de ayuda
- Instrucciones de Trabajo
- Ejercicios en salón
- Simulaciones
- Exámenes

Materiales de soporte al desempeño

- Políticas
- Preguntas Más Frecuentes
- Flujos de Proceso
- Conceptos
- Formas
- Consejos y Trucos

Tutoriales multimedia.

4.19 Capacitación al Personal Técnico

La capacitación para el personal técnico se llevara a cabo según el cronograma de actividades, el cual tendrá como objetivo explicar la parte técnica del software.

DIA 1	DIA 6	DIA 10	DIA 20
Manejo del sistemas módulo de mantenimiento	Configuración del servidor	Manejo del sistema módulo de transacciones	Manejo del sistema 100%
Arquitectura del sistema	Manejo de la base de datos		

Cronograma de Capacitación
Tabla 22

Fuente: Diego Báez

CAPÍTULO V

PRINCIPALES IMPACTOS

5.1 Científico.

La investigación desarrollada por el proyecto tiene una característica especial ya que aplicativos y tecnologías implementadas en nuestro medio únicamente se lo realizaba con software extranjero de los cuales no se podía tener el soporte técnico necesario, por lo tanto al tratarse de una implementación local se podrá beneficiar los técnicos y especialistas, que se dedican a este tipo de negocios ya que cualquier inquietud y soporte será localmente.

5.2 Educativo.

El desarrollo de un sistema informático involucra el conocimiento del diseño y la programación, por lo tanto al realizar un proyecto de fusión de tecnologías se realizara un aporte muy importante para el conocimiento de los alumnos de la institución que se encuentran en los niveles inferiores los cuales se pueden motivar y desarrollar proyectos.

5.3 Técnico.

Lo referente a este tema se considera las herramientas que se utilizaron para el modelamiento y desarrollo del Sistema de Inventario y Facturación , considerando que las mismas fueron estudiadas en las aulas de la Institución, en consecuencia su aplicación no tuvo mayor dificultad.

5.4 Tecnológico.

El ejecutar temas de innovación tecnológica aplicando los conocimientos recibidos en las aulas hace que cualquier proyecto sea considerado como un aporte institucional hacia la sociedad, mejor aún si lo que se está realizando es la unión de varias tecnologías en un solo aplicativo, que es producto de la investigación tecnológica.

5.5 Empresarial.

Siendo la visión emprendedora la realización en un futuro de una empresa, se ha cristalizado el primer paso que consiste en tener un aplicativo que maneje las reglas del negocio, por lo tanto a medida que se perfecciona el mismo se continuara con una mentalidad empresarial.

5.6 Social.

La implementación de este sistema y la estructuración de una empresa en lo futuro para brindar este tipo de servicio, lograra crear fuentes de trabajo tanto en el ámbito administrativo como técnico, por consiguiente se estará brindando un aporte significativo en el ámbito social.

5.7 Económico.

La generación de impuestos por el servicio brindado con este tipo de empresa beneficiara a la economía nacional ya que su aporte será con todos lo que establece la ley en pagos de aranceles e impuestos en general.

5.8 Conclusiones.

1. La metodología de investigación científica aplicada al proyecto me permitió determinar la utilización de tipos de investigación, métodos y herramientas que facilitaron la obtención de resultados cuantitativos y cualitativos, que facilitaron la obtención de procedimientos, tareas y procesos propios de la toma de decisiones del Sistema de Inventario y Facturación.
2. La aplicabilidad de los pasos que se siguen en ingeniería de software fueron determinantes ya que mediante el análisis me permitieron discernir todo el flujo de información que se ejecuta en el proceso de compra y venta de los productos, al mismo tiempo poder diseñar adecuadamente los procesos sus relaciones, las bases de datos y sus objetos, a fin de poder obtener una organización metódica y bien estructurada del Sistema de Inventario y Facturación.
3. Los conocimientos adquiridos en el Instituto tecnológico Superior Cordillera, me permitieron consolidar las teorías pragmáticas en herramientas de ejecución prácticas utilizando lenguajes de programación, y bases de datos materializadas en conocimientos de programación informática. Lo que definió en una concatenación de ideas lógicamente estructuradas y enlazadas hacia la consecución de un objetivo general “Analizar, desarrollar e implementar un Sistema de Inventario y Facturación.
4. En un entorno de desarrollo de software se debe realizar muchas pruebas de campo como validaciones de sistema, validaciones de usuarios, para poder decir que el software está terminado y en funcionamiento, más aun si consideramos que en nuestro caso también se tiene que mantener un especial atención por la fusión de tecnologías ejecutadas en el proyecto, lo que permitirá la consolidación de la información estructural.

5.9 Recomendaciones.

- Todo trabajo de investigación debe sustentarse con una metodología métodos y herramientas de la investigación científica por lo tanto es recomendable que las personas se preparen en este campo para realizar cualquier investigación, más aun considerando que esto es la base para poder realizar el proyecto de grado previo a la obtención del título de Tecnólogo Analista de Sistemas.
- Seguir la misma secuencia de desarrollo de software a fin de poder llegar a obtener un sistema completo, aplicable y amigable, es el resultado de haber seguido una metodología de desarrollo y técnicas de programación que permita interactuar los datos alfanuméricos (información), a fin de poder estructurar un sistema real de Toma de decisiones.
- Realizar este proyecto me ha permitido conocer aún más las reglas del negocio de una Ferretería, y he podido constatar del negocio que se brinda en este tipo de servicio, igualmente me ha permitido conocer las que he tenido que investigar y que puedan ayudar en el desarrollo e integración de este tipo de soluciones informáticas.

CAPÍTULO VI

Bibliografía

- Cervantes Martínez, J.D. (2000) Nueva Ley de concursos mercantiles (comentada), 2° ed., México, Cárdenas.

- Hernández López, Aarón. (1999) El procedimiento mercantil, 2° ed., México, Porrúa.

- Marín R, Francisco. (2006) Derecho Mercantil, 5° ed., Ecuador, Quito.

- Rodríguez Rodríguez, Joaquín. (1999) Ley de quiebras y suspensión de pagos, 15° ed., México, Porrúa.

- Jorge Serrano (2009) Manual Avanzado de Visual Basic 2008, editorial Anaya Multimedia, edición 2.

- Jorge Serrano (2010) Manual Avanzado de Visual Basic 2010, editorial Anaya Multimedia, edición 1.

Netgrafía

Conexión de MySQL con Visual Studio 2008

http://es.wikipedia.org/wiki/MySQL

El derecho mercantil

http://en.wikipedia.org/wiki/Derecho_Mercantil

Entorno de desarrollo en Microsoft Visual Studio 2008

http://es.wikipedia.org/wiki/Microsoft_Visual_Studio

Modelamiento UML

http://en.wikipedia.org/wiki/PowerDesigner

Glosario de Términos

Acreedor: Una persona o empresa con quien se tiene un adeudo.

Activo: Una propiedad o recurso económico poseído por un individuo o empresa.

Activo Circulante: El efectivo o cualquier activo que se espera, se va a realizar en efectivo o se va a consumir dentro de un año o dentro de un ciclo de operación del negocio (el periodo más largo de los dos).

Activo Fijo: Los activos tangibles que tienen vida relativamente larga y que se utilizan en la producción o venta de otros activos o servicios.

Activo Intangible: El activo que no tiene existencia física, cuyo valor radica en lo derechos conferidos como resultado de la titularidad y propiedad de los mismos.

Balance General: Un estado financiero que muestra los activos, los pasivos y el capital contable de una empresa a una fecha específica.

Cheques en Circulación: Cheques que se han expedido, registrado y enviado o entregado a sus beneficiarios, pero que no han sido recibidos por el banco, ni pagados, ni devueltos.

Cheques Pagados Cancelados: Cheques pagados por el banco y que fueron cancelados mediante perforación o estampado de sello.

Cliente: El comprador de algo.

Cuenta: El medio contable utilizado para registrar y resumir los aumentos y disminuciones.

Cuenta de Retiros: La cuenta utilizada para registrar los retiros del negocio que hace su propietario, sea en efectivo o en otros activos, para su uso personal.

Globalización: Proceso económico, tecnológico, social y cultural a gran escala, que consiste en la creciente comunicación e interdependencia entre los distintos países del mundo unificando sus mercados, sociedades y culturas, a través de una serie de transformaciones sociales, económicas y políticas que les dan un carácter global

Franquiciadora: Es un sistema comercial que permite explotar comercialmente una marca, servicio o producto con una imagen ya asentada, dentro de una red local, nacional o internacional.

Ddl:Permite a los usuarios de la misma llevar a cabo las tareas de definición de las estructuras que almacenarán los datos así como de los procedimientos o funciones que permitan consultarlos.

Triggers: Un disparador define una acción que la base de datos debe llevar a cabo cuando se produce algún suceso relacionado con la misma, pueden utilizarse para completar la integridad referencial, también para imponer reglas de negocio complejas o para auditar cambios en los datos.

Corba: El protocolo de comunicaciones y los mecanismos necesarios para permitir la interoperabilidad entre diferentes aplicaciones escritas en diferentes lenguajes y ejecutadas en diferentes plataformas, lo que es fundamental en computación distribuida.

ANEXOS

Anexo 1.- Manual de Usuario

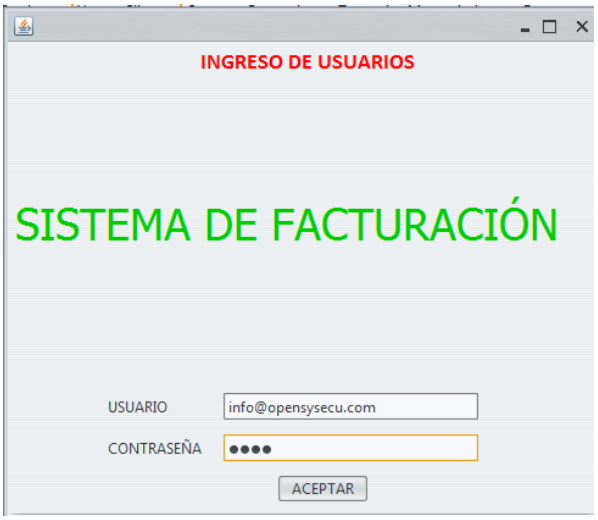
El sistema es una aplicación para controlar la facturación y gestionar el almacén la Ferretería Gordon. Sus principales funciones son:

- Gestión de Interlocutores comerciales [Clientes y proveedores]
- Gestión de Artículos y Familias
- Gestión de Facturas de los clientes
- Gestión de Facturas de los proveedores
- Ventas en mostrador
- Reportes

El funcionamiento a través de entorno java permite su uso multiplataforma, tanto en sistemas operativos Windows como Linux. El software ha sido desarrollado en lenguaje JAVA y utilizando como motor de base de datos MySQL.

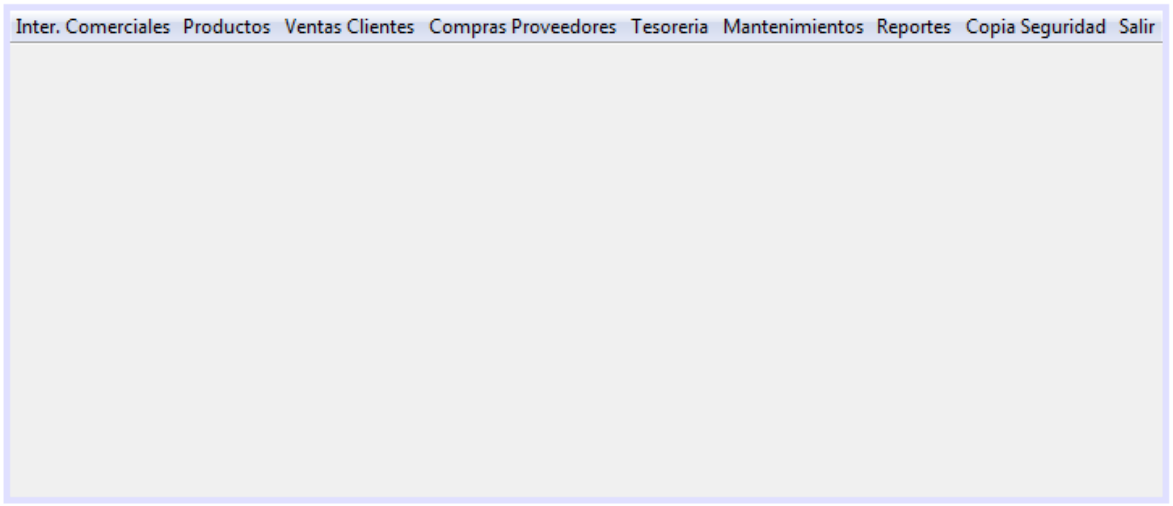
Inicio de sesión

La primera pantalla en la cual se identificara con los datos de usuario: info@opensysecu.com, y la contraseña: demo.



Identificación de usuario

Ilustración 35
Fuente: Diego Báez



Pantalla de inicio

Ilustración 36
Fuente: Diego Báez

Para poder iniciar a utilizar el sistema requiere de algunas tareas de instalación y configuración básicas, estas tareas están completamente explicadas con lujo de detalles y pantallas incluidas en el manual de instalación de AppServ.

A continuación se indica las actividades iniciales para utilizar SYSFAC, para comenzar a utilizar el sistema, se requiere ingresar la siguiente información en la secuencia que se especifica:

1. Impuestos
2. Entidades Bancarias
3. Ubicaciones
4. Embalajes
5. Formas de Pago
6. Clientes

7. Proveedores

8. Familia de Artículos

9. Artículos

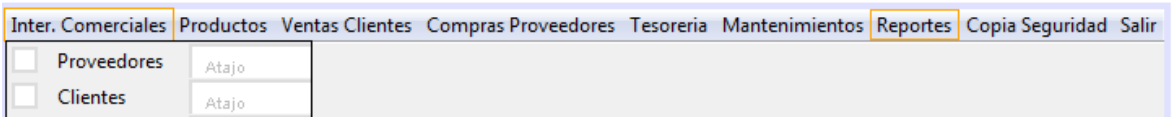
El detalle de cómo ingresar esta información la encuentras en este documento en los capítulos correspondientes, favor referirse a ellos para el detalle.

INTERLOCUTORES COMERCIALES

El sistema permite administrar a los interlocutores comerciales como proveedores y clientes.

PROVEEDORES

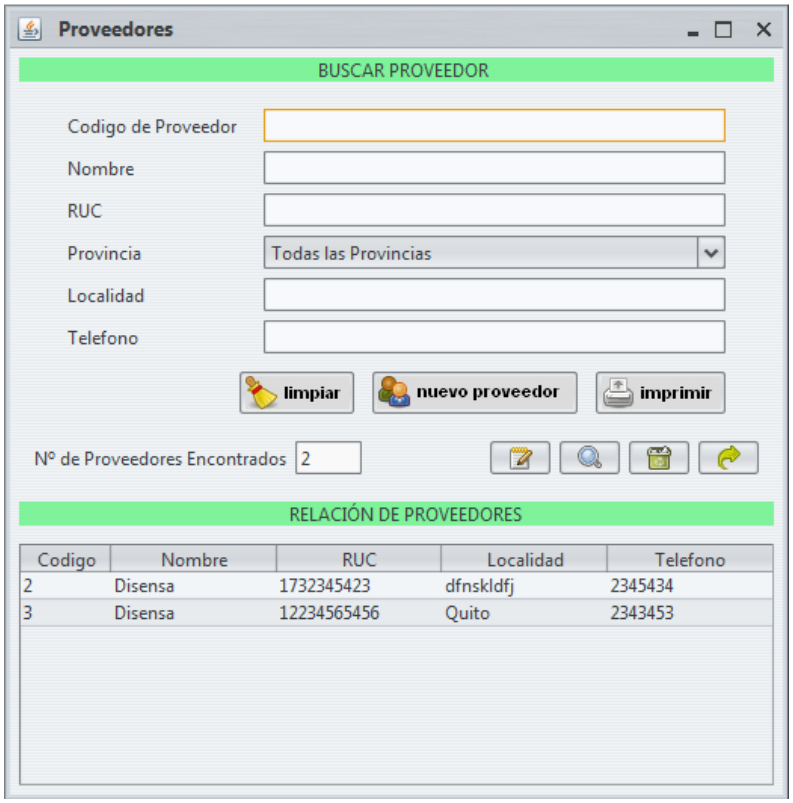
El sistema permite administrar una nómina de proveedores, para lo cual debes ingresar al menú “Proveedores” del menú Inter. Comerciales:



Menú de Interlocutores Comerciales

Ilustración 37
Fuente: Diego Báez

Se desplegará la siguiente pantalla con las opciones de imprimir, buscar o ingresar un nuevo registro:



Buscar proveedor

Ilustración 38
Fuente: Diego Báez

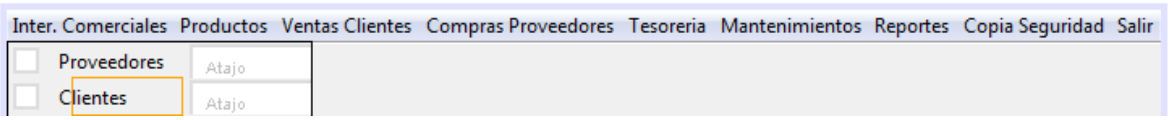
Al presionar “nuevo proveedor” se desplegará la siguiente pantalla para hacer el ingreso:

Nuevo Proveedor

Ilustración 39
Fuente: Diego Báez

CLIENTES

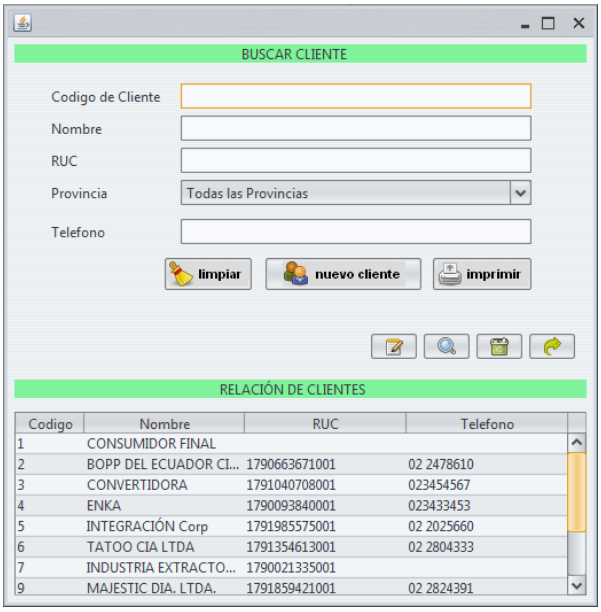
El sistema permite administrar una nómina de clientes, para lo cual debe ingresar a la opción “Clientes” del menú “Inter. Comerciales” como indica la siguiente pantalla:



Menú Opción Cliente

Ilustración 40
Fuente: Diego Báez

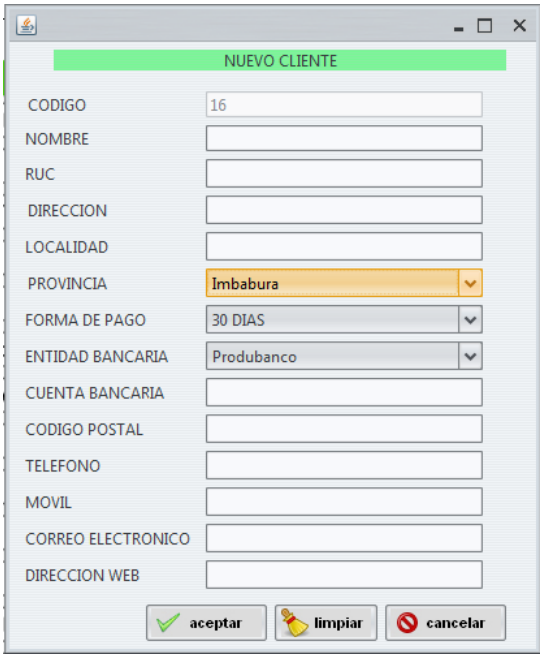
Se desplegará la siguiente pantalla con las opciones de imprimir, buscar o ingresar un nuevo registro:



Buscar Cliente

Ilustración 41
Fuente: Diego Báez

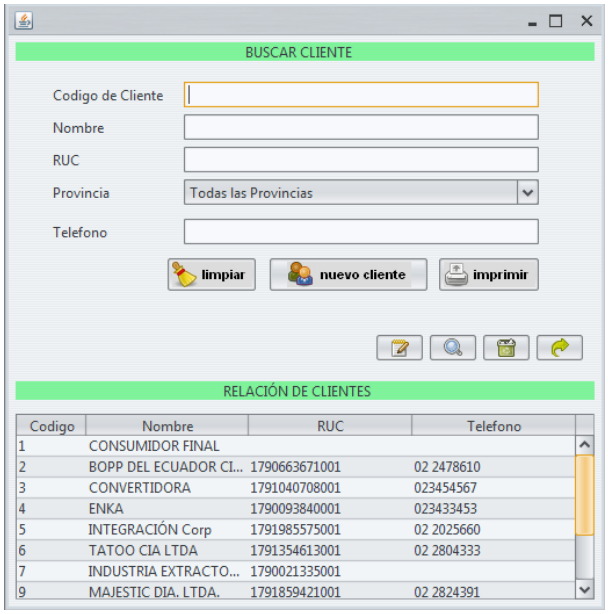
Al presionar “nuevo cliente” se desplegará la siguiente pantalla para hacer el ingreso:



Nuevo Cliente

Ilustración 42
Fuente: Diego Báez

Al presionar el botón aceptar se despliega la pantalla con la lista de todos los registros ingresados con la opción de buscar, imprimir o ingresar otro registro nuevo:



Registros Grabados

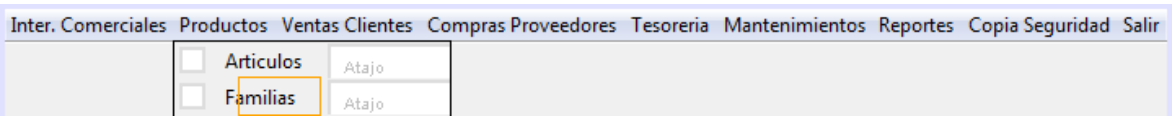
Ilustración 43
Fuente: Diego Báez

PRODUCTOS

El sistema permite administrar artículos y familias de artículos.

ARTÍCULOS

Para administrar artículos debes ingresar al menú Artículos del menú Productos:



Menú de artículos

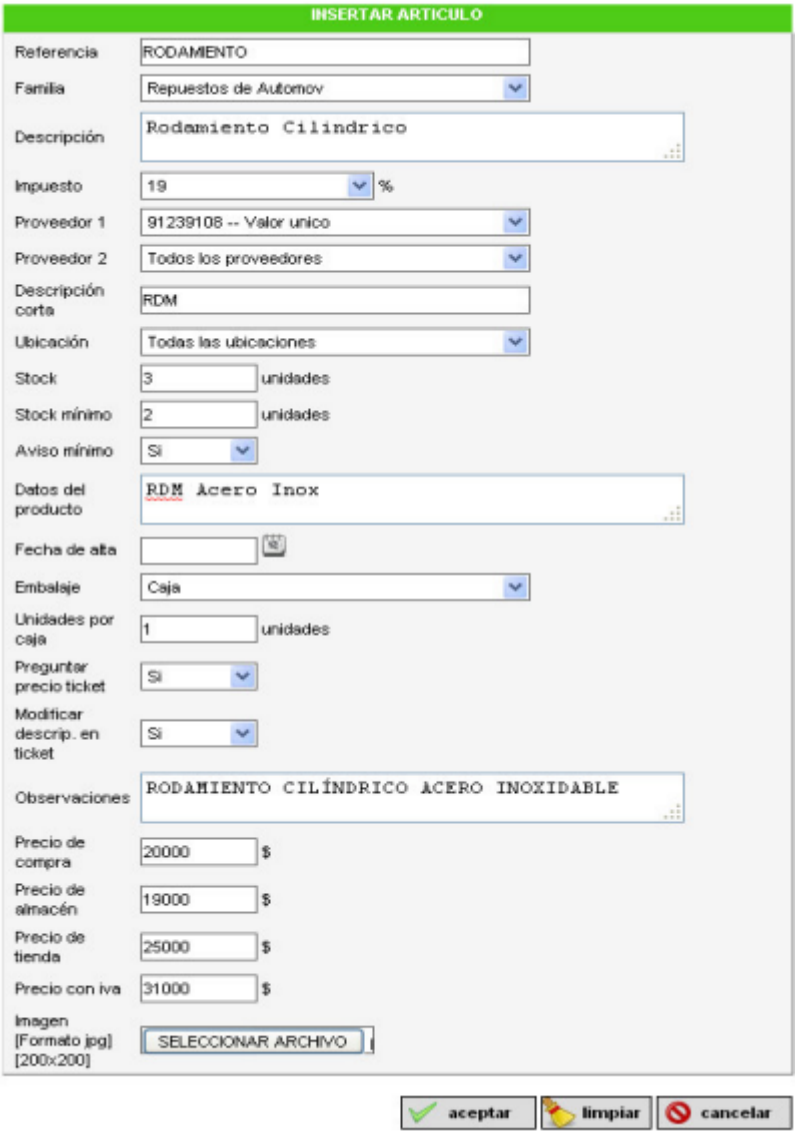
Ilustración 44
Fuente: Diego Báez

Se desplegará la siguiente pantalla con las opciones de imprimir, buscar o ingresar un nuevo registro:

Menú de artículos

Ilustración 45
Fuente: Diego Báez

Al presionar “nuevo artículo” se desplegará la siguiente pantalla para hacer el ingreso:

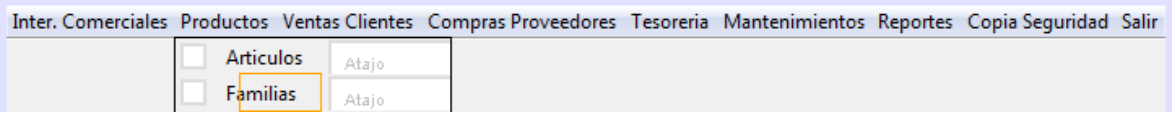


Menú de artículos

Ilustración 46
Fuente: Diego Báez

FAMILIAS

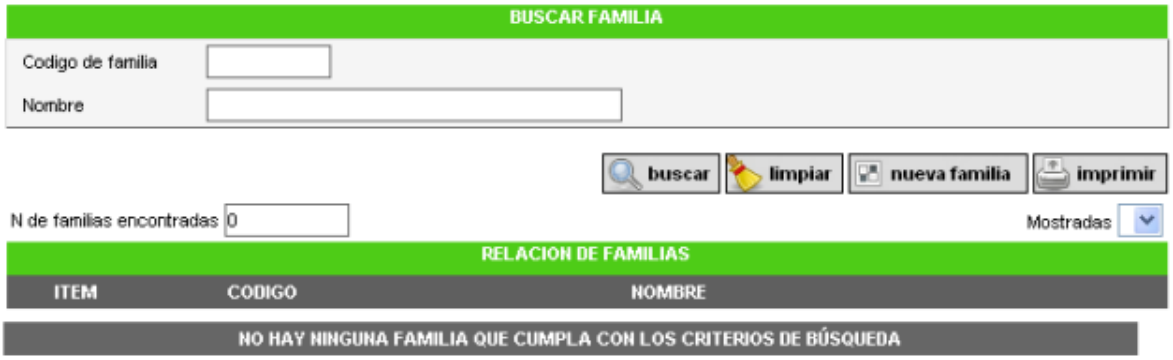
Para administrar familias de artículos debes ingresar al menú Familias del menú Productos:



Menú de familias

Ilustración 47
Fuente: Diego Báez

Se desplegará la siguiente pantalla con las opciones de imprimir, buscar o ingresar un nuevo registro:



Buscar Familia
Ilustración 48
Fuente: Diego Báez

Al presionar “nueva familia” se desplegará la siguiente pantalla para hacer el ingreso:

Confirmación de creación
Ilustración 49
Fuente: Diego Báez

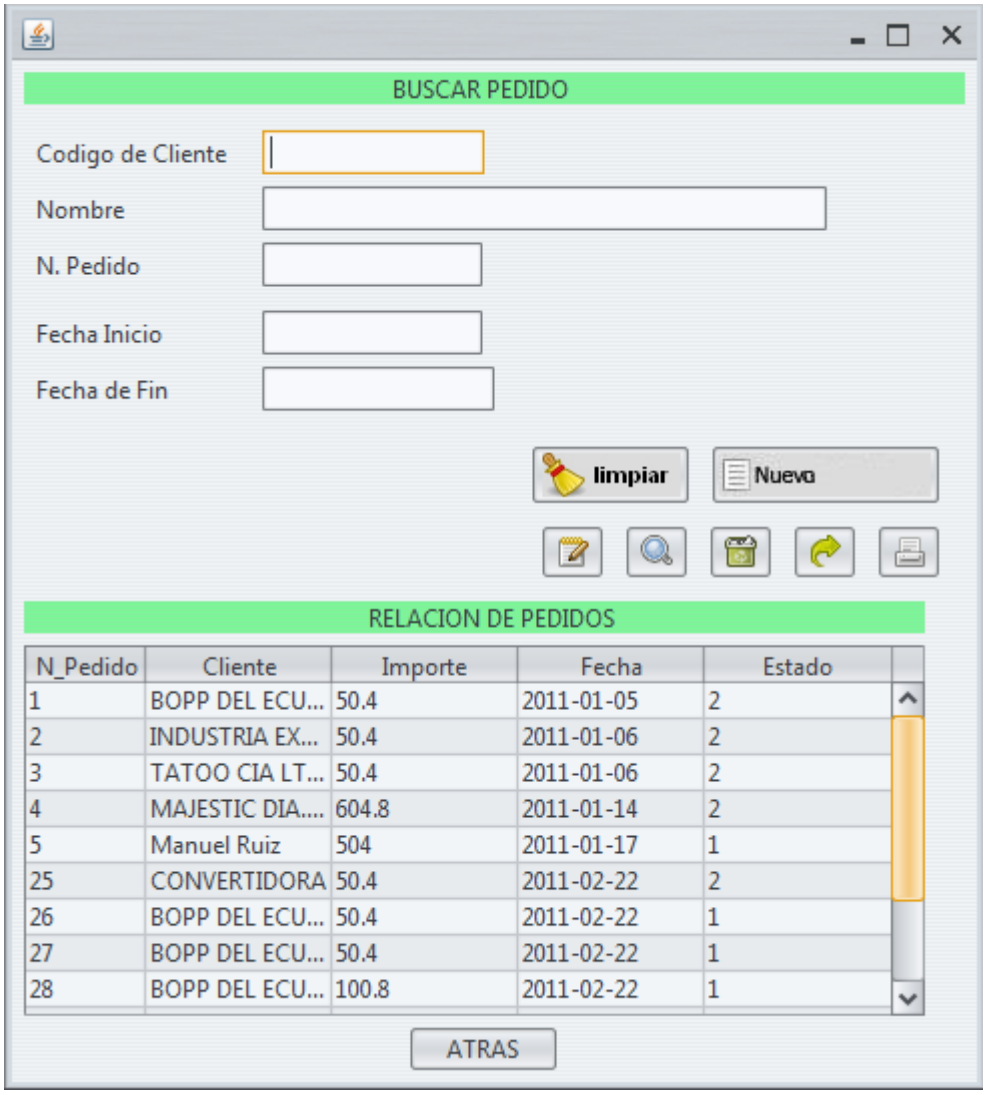
Al presionar el botón aceptar se despliega la pantalla con la lista de todos los registros ingresados con la opción de buscar e imprimir:

Buscar Familia
Ilustración 50
Fuente: Diego Báez

PEDIDOS

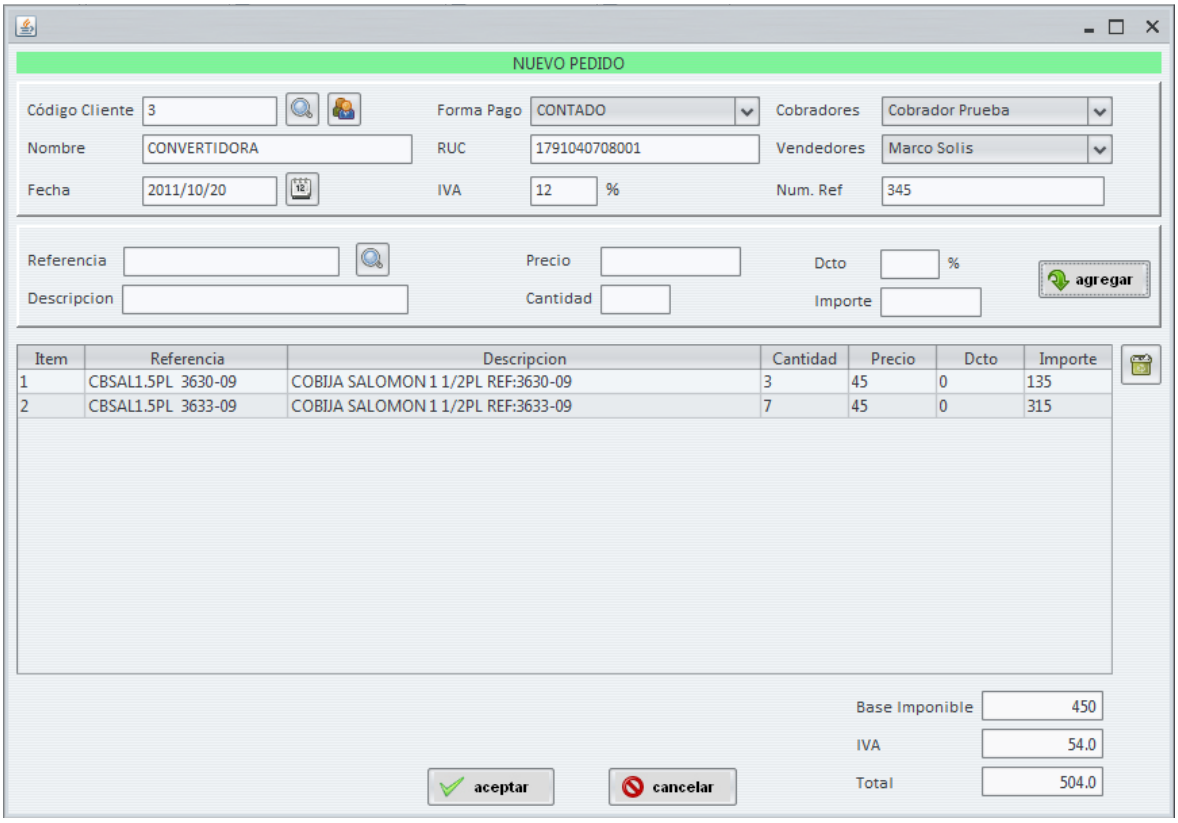
La opción albaranes o guías de despacho permite administrarlas, tal como buscar, crear, modificar, eliminar, ver, o imprimir etiquetas de códigos de barras.

Para acceder a la información de Guías de Despacho, debes ingresar a la opción Albaranes del menú Ventas Clientes:



Pedidos
Ilustración 51
Fuente: Diego Báez

Al presionar el botón nuevo se despliega la siguiente pantalla:



Nuevo pedido
Ilustración 52
Fuente: Diego Báez

Imprimir la factura:

Señor(es): Jhonny Melaslabo Today **NIT :** 9876543
Dirección: Calle Chacaltaya #655 **Teléfono:** 800-70088874

ID Producto	Detalle	Cantidad	Precio/Unidad	Total
PCA0LLKG0	Bolsos para Portatil Targus	4	11.0	44.0
PC0000000A	Computadora de marca HP Compaq Pavilion	2	255.8	511.6

Vendedor: jc Mouse
Observaciones:
Total Factura: Bs. 555.6
Son: QUINIENTOS CINCUENTA Y CINCO 6/100 BOLIVIANOS.

Nuevo pedido
Ilustración 53
Fuente: Diego Báez

Anexos 2.- Manual Técnico

Conexión

```
package Conexion;
import java.sql.*;
import java.util.Vector;
/**
 *
 * @author Fercho
 */
public class conexion {

      public Connection con;

          public Statement st;
          public ResultSet rs;
          //metodo para conectar a la base de datos
          //metodo para conectar a la base de datos
          public void conectar(){
          try{
          String userName="root";
          String password="root";
          String url="jdbc:mysql://localhost:3306/codeka";
                    Class.forName("com.mysql.jdbc.Driver").newInstance();
          con = DriverManager.getConnection(url,userName, password);
          //System.out.println("Conexión a la base de datos:" + con.getCatalog());
          }catch (Exception e){
          //System.out.print("Error en la conexion a la base de datos:" );
          }
```

```
	}
	public void desconectar(){
		try {
			if(con!=null){
			con.close();
			//System.out.println("Base de datos cerrada:" + con.getCatalog());
			}
		}catch (Exception e) {

			//System.out.println("Error al cerrar conexión"+e.getMessage());
			}
	}
public Vector cargarDatos (String spConsulta, String camposTabla[]){
	Vector filas=new Vector();
	try{
		conectar();
		st=con.createStatement();
		rs=st.executeQuery(spConsulta);
		while(rs.next()){
			Vector fila=new Vector();
			for(int counter=0; counter<camposTabla.length;counter++)
				fila.addElement(rs.getString(camposTabla[counter]));
			filas.addElement(fila);
			}
		rs.close();
		st.close();
	}catch(Exception e){
		e.printStackTrace();
	}
	return filas;
```

```
}
public boolean ejecutarSQL (String Sql){
        conectar ();
        boolean res=false;
        try {
                st=con.createStatement();
                if(st.executeUpdate(Sql)>0){
                        res= true;
                        st.close();
                }
        }
                catch (Exception e)
                {
                        e.printStackTrace();
                }
                return (res);
        }
public void recoletorbasura(){
        Runtime runtime=Runtime.getRuntime();
        runtime.gc();
        runtime.runFinalization();
}
}
```

Login

```
package Login;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.IOException;
import java.sql.*;
```

```
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.Vector;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author Fercho
 */
public class Inicio extends javax.swing.JFrame {
    public Connection con;
    public Statement st;
    public ResultSet rs;
    private Timer tiempo;
    private Date reloj1;
    private Calendar reloj2;
    private SimpleDateFormat formato;
    String usuario, elPassword;

    /** Creates new form Inicio */
    public Inicio() {
        initComponents();
        setLocationRelativeTo(this);
        RelojReal();
    }

    public void RelojReal(){
        formato = new SimpleDateFormat("hh:mm:ss aa");
```

```
    tiempo = new Timer(1000, new ActionListener(){
      public void actionPerformed(ActionEvent evt){
        reloj1 = new Date();
        reloj.setText(formato.format(reloj1));
      }
    });
    tiempo.start();
  }
   boolean validarUsuario(String elUsr, String elPw)  throws IOException
          {
            try
            {
              Connection unaConexion = DriverManager.getConnection
("jdbc:mysql://localhost/codeka","root", "root");
              // Preparamos la consulta
              Statement instruccionSQL = unaConexion.createStatement();
              ResultSet resultadosConsulta = instruccionSQL.executeQuery ("SELECT *
FROM users WHERE user_email='"+elUsr+"' AND user_name='"+ elPw+"'");

              if( resultadosConsulta.first() )        // si es valido el primer reg. hay una fila,
tons el usuario y su pw existen
                return true;       //usuario validado correctamente
              else
                return false;       //usuario validado incorrectamente

            } catch (Exception e)
            {
              e.printStackTrace();
              return false;
            }
```

```
            }
    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
        pack();
    }// </editor-fold>

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        try
                        {
                            //chekar si el usuario escrbio el nombre de usuario y pw
                            if (txtUser.getText().length() > 0 && txtPass.getText().length() > 0 )
                            {
                                // Si el usuario si fue validado correctamente
                                if( validarUsuario( txtUser.getText(), txtPass.getText() ) )      //enviar
datos a validar
                                {
                        // Codigo para mostrar la ventana principal
                                    setVisible(false);
                                JOptionPane.showMessageDialog(null, "BIENVENIDO AL SISTEMA ");
                                    new Principal().setVisible(true);
                                }
                                else
                                {
                                    JOptionPane.showMessageDialog(null, "El nombre de usuario y/o
contrasenia no son validos.");
```

```
                    JOptionPane.showMessageDialog(null, txtUser.getText()+" "
+txtPass.getText() );
                    txtUser.setText("");  //limpiar campos
                    txtPass.setText("");
                    txtUser.requestFocusInWindow();
                  }
                }
                else
                {
                  JOptionPane.showMessageDialog(null, "Debe escribir nombre de
usuario y contrasenia.\n" +
                    "NO puede dejar ningun campo vacio");
                }

              } catch (Exception e)
              {
                e.printStackTrace();
              }
  }

  /**
  * @param args the command line arguments
  */
  public static void main(String args[]) throws UnsupportedLookAndFeelException {
    UIManager.setLookAndFeel(new
org.jvnet.substance.skin.SubstanceOfficeSilver2007LookAndFeel());
    JFrame.setDefaultLookAndFeelDecorated(true);
    SwingUtilities.invokeLater(new Runnable() {
      public void run() {
        new Inicio().setVisible(true);
```

```
        }
    });
  }
```

Principal

```
package Login;

import Proveedores.Proveedores;

/**
 *
 * @author Fercho
 */
public class Principal extends javax.swing.JFrame {

  /** Creates new form Principal */
  public Principal() {
    initComponents();
    setLocationRelativeTo(this);
    }
    );

    pack();
  }// </editor-fold>

  private void jMenu9MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    System.exit(0);
  }
```

```
    private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        new Proveedores().setVisible(true);
        this.setVisible(false);
    }

    private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
       new Clientes.Clientes().setVisible(true);
       this.setVisible(false);
    }

    private void jMenuItem9ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        new Pedidos.Pedido().setVisible(true);
        this.setVisible(false);
    }

    /**
    * @param args the command line arguments

    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Principal().setVisible(true);
            }
        });
    }*/

}
```

Proveedores

```
package Proveedores;

import Conexion.conexion;
import java.sql.*;
import java.util.Vector;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperReport;
import net.sf.jasperreports.view.JasperViewer;



/**
 *
 * @author Fercho
 */
public class Proveedores extends javax.swing.JFrame {
conexion cn = new conexion();
    /** Creates new form Proveedores */
    public Proveedores() {
        initComponents();
        setLocationRelativeTo(this);
        cargarCombo();
        Cargar_Datos();
        Num_Registros();

    }
```

```
public void cargarCombo() {
    try {
    Connection con = DriverManager.getConnection
("jdbc:mysql://localhost:3306/codeka", "root", "root");
    Statement stmt = con.createStatement();
    ResultSet rs = stmt.executeQuery ("SELECT nombreprovincia FROM provincias
WHERE borrado=0 ORDER BY codprovincia ASC");
    cboprovincia.removeAllItems();
    cboprovincia.addItem("Todas las Provincias");
    while(rs.next())
    cboprovincia.addItem(rs.getObject(1));
    rs.close();
    stmt.close();
    con.close();
    } catch (SQLException e) {
    e.printStackTrace();
    }
    pack();
    setLocationRelativeTo (null);
  }

  public void Cargar_Datos(){
            Vector  filas = new Vector();

            String CamposTabla[] =new String [5];
            CamposTabla[0]="Codigo";
            CamposTabla[1]="Nombre";
        CamposTabla[2]="RUC";
        CamposTabla[3]="Localidad";
        CamposTabla[4]="Telefono";
```

```
        //CamposTabla[4]="cli_email";
                filas=cn.cargarDatos("SELECT codproveedor as Codigo," +
                        "nombre as Nombre, nif as RUC, " +
                        "localidad as Localidad, telefono as Telefono " +
                        "FROM proveedores WHERE borrado=0", CamposTabla);
                Vector columnas= new Vector();
                for(int counter=0;counter<CamposTabla.length;counter++){
                        columnas.addElement(CamposTabla[counter]);
                        }
                DefaultTableModel modelo= new DefaultTableModel (filas,columnas)
    {
    @Override
    public boolean isCellEditable (int row,int column){
                return true;
    }
    };
    tabla.setModel(modelo);
 tabla.getColumnModel().getColumn(0).setMaxWidth(60);
    }

public void Consulta_like(){
                Vector  filas = new Vector();

                String CamposTabla[] =new String [5];
                CamposTabla[0]="Codigo";
                CamposTabla[1]="Nombre";
        CamposTabla[2]="RUC";
        CamposTabla[3]="Localidad";
        CamposTabla[4]="Telefono";
                String Sql=
```

```
                "Select codproveedor as Codigo, nombre as Nombre, nif as RUC, localidad as Localidad, telefono as Telefono" +
                                " From proveedores " +
                                "Where codproveedor like " +
                                "'" + txtcodigo.getText().trim() + "%'" +
                                "and nombre like " +
                                "'" + txtnombre.getText().trim() + "%'" +
                        "and nif like " +
                                "'" + txtruc.getText().trim() + "%'" +
                        "and localidad like " +
                                "'" + txtlocalidad.getText().trim() + "%'"+
                        "and telefono like " +
                                "'" + txttelefono.getText().trim() + "%'" +
                        "and borrado=0";
                filas=cn.cargarDatos(Sql,CamposTabla);
                Vector columnas= new Vector();
                for(int counter=0;counter<CamposTabla.length;counter++){
                        columnas.addElement(CamposTabla[counter]);
                        }
                DefaultTableModel modelo= new DefaultTableModel (filas,columnas)
          {
      btneliminar.setIcon(new javax.swing.ImageIcon(getClass().getResource("/Img/eliminar.png"))); // NOI18N
      btneliminar.setFocusable(false);
      btneliminar.addActionListener(new java.awt.event.ActionListener() {
         public void actionPerformed(java.awt.event.ActionEvent evt) {
            btneliminarActionPerformed(evt);
         }
      });
    .addContainerGap())
```

```
        );

        pack();
    }// </editor-fold>

    private void btnlimpiarActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        cargarCombo();
        Cargar_Datos();
        Num_Registros();
        txtcodigo.setText("");
        txtnombre.setText("");
        txtruc.setText("");
        //cboprovincia;
        txtlocalidad.setText("");
        txttelefono.setText("");
        txtcodigo.enable(true);
        txtcodigo.requestFocus();
    }

    private void btneliminarActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
         if(!txtcodigo.getText().equals("")){
        new Eliminar_Proveedor().setVisible(true);
        this.setVisible(false);
        }
 else{
     JOptionPane.showMessageDialog(null, "DEBE SELECCIONAR UN PROVEEDOR DE LA
TABLA ", "ERROR ", JOptionPane.OK_OPTION);
 }
```

```
}

private void btnregresarActionPerformed(java.awt.event.ActionEvent evt) {
   // TODO add your handling code here:
   new Login.Principal().setVisible(true);
   this.setVisible(false);
}
private void tablaMouseReleased(java.awt.event.MouseEvent evt) {
   // TODO add your handling code here:
   txtcodigo.setText((String)tabla.getValueAt(tabla.getSelectedRow(),0));
   txtnombre.setText((String)tabla.getValueAt(tabla.getSelectedRow(),1));
   txtruc.setText((String)tabla.getValueAt(tabla.getSelectedRow(),2));
   txtlocalidad.setText((String)tabla.getValueAt(tabla.getSelectedRow(),3));
   txttelefono.setText((String)tabla.getValueAt(tabla.getSelectedRow(),4));
   Carga_Provincia_provincias();
}

private void btnnuevoActionPerformed(java.awt.event.ActionEvent evt) {
   // TODO add your handling code here:
   new Nuevo_Proveedor().setVisible(true);
   this.setVisible(false);
}

private void txtcodigoKeyReleased(java.awt.event.KeyEvent evt) {
   // TODO add your handling code here:
   Consulta_like();
}

private void txtnombreKeyReleased(java.awt.event.KeyEvent evt) {
   // TODO add your handling code here:
```

```
    Consulta_like();
  }

  private void txtrucKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    Consulta_like();
  }

  private void txttelefonoKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    Consulta_like();
  }

  private void btnmodificarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if(!txtcodigo.getText().equals("")){
    new Modificar_Proveedor().setVisible(true);
    this.setVisible(false);
    }
 else{
    JOptionPane.showMessageDialog(null, "DEBE SELECCIONAR UN PROVEEDOR DE LA TABLA ", "ERROR ", JOptionPane.OK_OPTION);
 }
  }

  private void btnimprimirActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try{

      //variable de tipo JasperReport
```

```
        JasperReport jaspertReport;

        //ensamblamos el path
        String fullPath="Reportes/Proveedores.jasper";

        //rellenar el reporte
        JasperPrint JPrint=JasperFillManager.fillReport(fullPath,null,cn.con);
        //ver el reporte
        JasperViewer.viewReport(JPrint,false);
      }catch(Exception ed){
      System.out.println("Error");
      }
   }

   private void btnvisualizarActionPerformed(java.awt.event.ActionEvent evt) {
      // TODO add your handling code here:
      if(!txtcodigo.getText().equals("")){
      new Ver_Proveedor().setVisible(true);
      this.setVisible(false);
      }
 else{
    JOptionPane.showMessageDialog(null, "DEBE SELECCIONAR UN PROVEEDOR DE LA
TABLA ", "ERROR ", JOptionPane.OK_OPTION);
 }
   }

   /**
   * @param args the command line arguments

   public static void main(String args[]) {
```

```
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Proveedores().setVisible(true);
            }
        });
    }*/
}
```

Nuevo Proveedor

```
package Proveedores;
import Conexion.conexion;
import java.sql.*;
import javax.swing.JOptionPane;
public class Nuevo_Proveedor extends javax.swing.JFrame {
conexion cn = new conexion();
    /** Creates new form Nuevo_Proveedor */
    public Nuevo_Proveedor() {
        initComponents();
        Cod_NuevoProveedor();
        cargarComboprovincia();
        cargarComboEntBancaria();
        txtnombre.requestFocus();

    }

    public void cargarComboprovincia() {
        try {
        Connection con = DriverManager.getConnection
("jdbc:mysql://localhost:3306/codeka", "root", "root");
        Statement stmt = con.createStatement();
```

```
      ResultSet rs = stmt.executeQuery ("SELECT * FROM provincias WHERE borrado=0
ORDER BY codprovincia ASC");
      cboprovincia.removeAllItems();
      cboprovincia.addItem("Seleccione una Provincia");
      while(rs.next())
      cboprovincia.addItem(rs.getObject(2));
      rs.close();
      stmt.close();
      con.close();
      } catch (SQLException e) {
      e.printStackTrace();
      }
      pack();
      setLocationRelativeTo (null);
   }

   public void cargarComboEntBancaria() {
      try {
      Connection con = DriverManager.getConnection
("jdbc:mysql://localhost:3306/codeka", "root", "root");
      Statement stmt = con.createStatement();
      ResultSet rs = stmt.executeQuery ("SELECT * FROM entidades WHERE borrado=0
ORDER BY codentidad ASC");
      cboentbancaria.removeAllItems();
      cboentbancaria.addItem("Seleccione una Entidad Bancaria");
      while(rs.next())
      cboentbancaria.addItem(rs.getObject(2));
      rs.close();
      stmt.close();
      con.close();
```

```
    } catch (SQLException e) {
    e.printStackTrace();
    }
    pack();
    setLocationRelativeTo (null);
}

public void Limpiar(){
    txtcodigo.setText("");
    txtcodigopos.setText("");
    txtcorreo.setText("");
    txtcuentabancaria.setText("");
    txtdireccion.setText("");
    txtlocalidad.setText("");
    txtmovil.setText("");
    txtnombre.setText("");
    txtruc.setText("");
    txttelefono.setText("");
    txtweb.setText("");
    cargarComboprovincia();
    cargarComboEntBancaria();
    Cod_NuevoProveedor();
    txtnombre.requestFocus();
}

public void Cod_NuevoProveedor() {
    Connection conn = null;
        Statement sta;
        ResultSet rsl;
```

```
String codmax;
   int a;
   String n = "";
   String b = "";
    try {
   String userName="root";
   String password="root";
   String url="jdbc:mysql://localhost:3306/codeka";
   Class.forName("com.mysql.jdbc.Driver").newInstance();
   conn = DriverManager.getConnection(url,userName, password);
   //toma el valor de la base
            sta =conn.createStatement();
            rsl = sta.executeQuery("select codproveedor from proveedores order by
codproveedor desc limit 0,1");
      while(rsl.next()){
            //System.out.println("\n Codigo:"+rs.getString("cli_codigo"));
      codmax=rsl.getString("codproveedor");
      a=Integer.parseInt(codmax);
      a=a+1;
      b = b.valueOf(a);
      txtcodigo.setText(b);
      txtcodigo.enable(false);
      }
      if(txtcodigo.getText().equals("")){
         txtcodigo.setText("1");
         txtcodigo.enable(false);
            }
            rsl.close();
    } catch (Exception e) {
   txtcodigo.requestFocus();
```

```
                System.out.println("Error al Abrir tabla ");
                }

}

 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Limpiar();
}

 private void GuardarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int resp;
resp=JOptionPane.showConfirmDialog(null,"¿Desea agregar el registro?","Pregunta",0);
    String comando;
if(resp==0){
    if (//!txtcodigo.getText().equals("")&&
        !txtnombre.getText().equals("")&&
        !txtruc.getText().equals("")&&
        !txtdireccion.getText().equals("")&&
        !txtlocalidad.getText().equals("")&&
        !txtcuentabancaria.getText().equals("")&&
        !txtcodigopos.getText().equals("")&&
        !txttelefono.getText().equals("")&&
        !txtmovil.getText().equals("")&&
        !txtcorreo.getText().equals("")&&
        !txtweb.getText().equals("")){
          try {
                String sql = ("INSERT INTO proveedores "
```

```
                +
/*"(codproveedor,nombre,nif,direccion,codprovincia,localidad,codentidad,cuentabancari
a,codpostal,telefono,movil,email,web,borrado) " +*/
                "VALUES ('" + txtcodigo.getText().trim() + "'," +
                "'" + txtnombre.getText().trim() + "'," +
                "'" + txtruc.getText().trim() + "'," +
                "'" + txtdireccion.getText().trim() + "'," +
                "'" + cboprovincia.getSelectedIndex() + "'," +
                "'" + txtlocalidad.getText().trim() + "'," +
                "'" + cboentbancaria.getSelectedIndex() + "'," +
                "'" + txtcuentabancaria.getText().trim() + "'," +
                "'" + txtcodigopos.getText().trim() + "'," +
                "'" + txttelefono.getText().trim() + "'," +
                "'" + txtmovil.getText().trim() + "'," +
                "'" + txtcorreo.getText().trim() + "'," +
                "'" + txtweb.getText().trim() + "'," + "'0'" + ")");
                System.out.println(sql);
                if (cn.ejecutarSQL(sql)== true)
                {
                    JOptionPane.showMessageDialog(null,"Registro Grabado", "Tabla
Proveedores", JOptionPane.OK_OPTION);
                    Limpiar();
                    new Proveedores().setVisible(true);
                    this.setVisible(false);
                }


            }
            catch (Exception sqle){
```

```
                JOptionPane.showMessageDialog(null,"ERROR", "Tabla Proveedores ",
JOptionPane.OK_OPTION);
            }
        }
        else{
            JOptionPane.showMessageDialog(null, " UN CASILLERO ESTA VACIO ", "ERROR
", JOptionPane.OK_OPTION);
        }
  }
   }

   private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
      // TODO add your handling code here:
      new Proveedores().setVisible(true);
      this.setVisible(false);
   }

}
```

Ver Proveedor

```
package Proveedores;

import java.sql.*;
import javax.swing.JOptionPane;

/**
 *
 * @author Fercho
 */
public class Ver_Proveedor extends javax.swing.JFrame {
```

```
/** Creates new form Ver_Proveedor */
public Ver_Proveedor() {
    initComponents();
    txtcodigo.setText(Proveedores.txtcodigo.getText());
    Cargar_Registros();
    Cargar_Registro_Provincia();
    Cargar_Registro_EntBancaria();
}

public void Cargar_Registro_Provincia(){
    try {
    Connection con = DriverManager.getConnection
("jdbc:mysql://localhost:3306/codeka", "root", "root");
    Statement stmt = con.createStatement();
    ResultSet rs = stmt.executeQuery ("SELECT pro.nombreprovincia FROM proveedores
as p, provincias as pro WHERE p.codproveedor =" + "'" + txtcodigo.getText().trim() + "'" +
" AND p.codprovincia=pro.codprovincia");
    while(rs.next())
    txtprovincia.setText(rs.getString("pro.nombreprovincia"));
    rs.close();
    stmt.close();
    con.close();

    } catch (SQLException e) {
    e.printStackTrace();
    }
    pack();
    setLocationRelativeTo (null);
}
```

```
   public void Cargar_Registro_EntBancaria(){
      try {
      Connection con = DriverManager.getConnection
("jdbc:mysql://localhost:3306/codeka", "root", "root");
      Statement stmt = con.createStatement();
      ResultSet rs = stmt.executeQuery ("SELECT e.nombreentidad FROM proveedores as
p, entidades as e WHERE p.codproveedor =" + "'" + txtcodigo.getText().trim() + "'" + " AND
p.codentidad=e.codentidad");
      while(rs.next())
      txtentidad.setText(rs.getString("e.nombreentidad"));
      rs.close();
      stmt.close();
      con.close();

      } catch (SQLException e) {
      e.printStackTrace();
      }
      pack();
      setLocationRelativeTo (null);
   }

   public void Cargar_Registros(){
      String b = txtcodigo.getText();
      if (!txtcodigo.getText().equals("")){
         try {
        Connection con = DriverManager.getConnection
("jdbc:mysql://localhost:3306/codeka", "root", "root");
      Statement stmt = con.createStatement();
      ResultSet rs = stmt.executeQuery ("SELECT * FROM proveedores");
```

```
                while(rs.next()){
            if(b.equals(rs.getString("codproveedor"))){
            txtnombre.setText(rs.getString("nombre"));
            txtruc.setText(rs.getString("nif"));
            txtdireccion.setText(rs.getString("direccion"));
            txtlocalidad.setText(rs.getString("localidad"));
            txtcuentabancaria.setText(rs.getString("cuentabancaria"));
            txtcodigopos.setText(rs.getString("codpostal"));
            txttelefono.setText(rs.getString("telefono"));
            txtmovil.setText(rs.getString("movil"));
            txtcorreo.setText(rs.getString("email"));
            txtweb.setText(rs.getString("web"));
            }
          }
                rs.close();

        } catch (Exception e) {
                System.out.println("Error al Abrir tabla ");
                }
  }
     else{
       JOptionPane.showMessageDialog(null, " NO EXISTE PROVEEDOR ", "ERROR ",
JOptionPane.OK_OPTION);
       }
  }


  public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
      public void run() {
```

```
            new Ver_Proveedor().setVisible(true);
        }
    });
}*/

// Variables declaration - do not modify
}
```

Cliente

```
package Clientes;

import Conexion.conexion;
import java.sql.*;
import java.util.Vector;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperReport;
import net.sf.jasperreports.view.JasperViewer;
    public Clientes() {
        initComponents();
        setLocationRelativeTo(this);
        cargarCombo();
        Cargar_Datos();
    }

    public void cargarCombo() {
        try {
```

```
Connection con = DriverManager.getConnection ("jdbc:mysql://localhost:3306/codeka", "root", "root");
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery ("SELECT nombreprovincia FROM provincias WHERE borrado=0 ORDER BY codprovincia ASC");
cboprovincia.removeAllItems();
cboprovincia.addItem("Todas las Provincias");
while(rs.next())
cboprovincia.addItem(rs.getObject(1));
rs.close();
stmt.close();
con.close();
} catch (SQLException e) {
e.printStackTrace();
}
pack();
setLocationRelativeTo (null);
}

public void Cargar_Datos(){
        Vector  filas = new Vector();

        String CamposTabla[] =new String [4];
        CamposTabla[0]="Codigo";
        CamposTabla[1]="Nombre";
    CamposTabla[2]="RUC";
    CamposTabla[3]="Telefono";
    //CamposTabla[4]="cli_email";
        filas=cn.cargarDatos("SELECT codcliente as Codigo," +
```

```
                "nombre as Nombre, nif as RUC, telefono as Telefono FROM clientes WHERE
borrado=0 ORDER BY codcliente ASC", CamposTabla);
                Vector columnas= new Vector();
                for(int counter=0;counter<CamposTabla.length;counter++){
                        columnas.addElement(CamposTabla[counter]);
                        }
                DefaultTableModel modelo= new DefaultTableModel (filas,columnas)
        {
        @Override
        public boolean isCellEditable (int row,int column){
                return true;
        }
        };
        tabla.setModel(modelo);
      tabla.getColumnModel().getColumn(0).setMaxWidth(60);
        }

    public void Consulta_like(){
                Vector  filas = new Vector();

                String CamposTabla[] =new String [4];
                CamposTabla[0]="Codigo";
                CamposTabla[1]="Nombre";
          CamposTabla[2]="RUC";
          CamposTabla[3]="Telefono";
                String Sql=
                "SELECT codcliente as Codigo," +
               "nombre as Nombre, " +
               "nif as RUC, " +
               "telefono as Telefono " +
```

```
            "FROM clientes " +
            "WHERE codcliente like " +
                        "'" + txtcodigo.getText().trim() + "%'" +
                        "and nombre like " +
                        "'" + txtnombre.getText().trim() + "%'" +
                "and nif like " +
                        "'" + txtruc.getText().trim() + "%'" +
                "and telefono like " +
                        "'" + txttelefono.getText().trim() + "%'" +
                "and borrado=0";
             filas=cn.cargarDatos(Sql,CamposTabla);
             Vector columnas= new Vector();
             for(int counter=0;counter<CamposTabla.length;counter++){
                    columnas.addElement(CamposTabla[counter]);
                    }
             DefaultTableModel modelo= new DefaultTableModel (filas,columnas)
    {
    btnregresar.setIcon(new
   public void mouseReleased(java.awt.event.MouseEvent evt) {
      tablaMouseReleased(evt);
   }
});
txttelefono.addKeyListener(new java.awt.event.KeyAdapter() {
   public void keyReleased(java.awt.event.KeyEvent evt) {
      txttelefonoKeyReleased(evt);
   }
});

txtnombre.addKeyListener(new java.awt.event.KeyAdapter() {
   public void keyReleased(java.awt.event.KeyEvent evt) {
```

```
            txtnombreKeyReleased(evt);
        }
    });

    txtruc.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyReleased(java.awt.event.KeyEvent evt) {
            txtrucKeyReleased(evt);
        }
    });

    jLabel7.setText("Telefono");

    txtcodigo.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyReleased(java.awt.event.KeyEvent evt) {
            txtcodigoKeyReleased(evt);
        }
    });

  private void btnmodificarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if(!txtcodigo.getText().equals("")){
        new Modificar_Cliente().setVisible(true);
        this.setVisible(false);
    } else{
        JOptionPane.showMessageDialog(null, "DEBE SELECCIONAR UN PROVEEDOR DE LA
TABLA ", "ERROR ", JOptionPane.OK_OPTION);
    }
}

  private void btnvisualizarActionPerformed(java.awt.event.ActionEvent evt) {
```

```
        // TODO add your handling code here:
        if(!txtcodigo.getText().equals("")){
            new Ver_Cliente().setVisible(true);
            this.setVisible(false);
        } else{
            JOptionPane.showMessageDialog(null, "DEBE SELECCIONAR UN PROVEEDOR DE LA
TABLA ", "ERROR ", JOptionPane.OK_OPTION);
        }
}

    private void btnlimpiarActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        cargarCombo();
        Cargar_Datos();
        txtcodigo.setText("");
        txtnombre.setText("");
        txtruc.setText("");
        //cboprovincia;
        txttelefono.setText("");
        txtcodigo.enable(true);
        txtcodigo.requestFocus();
}

    private void btnnuevoActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        new Nuevo_Cliente().setVisible(true);
        this.setVisible(false);
}

    private void btnimprimirActionPerformed(java.awt.event.ActionEvent evt) {
```

```
        // TODO add your handling code here:
        try{

            //variable de tipo JasperReport
            JasperReport jaspertReport;

            //ensamblamos el path
            String fullPath="Reportes/Proveedores.jasper";

            //rellenar el reporte
            JasperPrint JPrint=JasperFillManager.fillReport(fullPath,null,cn.con);
            //ver el reporte
            JasperViewer.viewReport(JPrint,false);
        }catch(Exception ed){
            System.out.println("Error");
        }
    }

    private void txttelefonoKeyReleased(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
        Consulta_like();
    }

    private void txtnombreKeyReleased(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
        Consulta_like();
    }

    private void txtrucKeyReleased(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
```

```
        Consulta_like();
    }

    private void txtcodigoKeyReleased(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
        Consulta_like();
    }

    private void btneliminarActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        if(!txtcodigo.getText().equals("")){
            new Eliminar_Cliente().setVisible(true);
            this.setVisible(false);
        } else{
            JOptionPane.showMessageDialog(null, "DEBE SELECCIONAR UN PROVEEDOR DE LA
TABLA ", "ERROR ", JOptionPane.OK_OPTION);
        }
    }
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Clientes().setVisible(true);
            }
        });
    }
}
```

Nuevo Cliente

```
package Clientes;
```

```
import Conexion.conexion;
import java.sql.*;
import javax.swing.JOptionPane;

/**
 *
 * @author Diego
 */
public class Nuevo_Cliente extends javax.swing.JFrame {
conexion cn = new conexion();
   /** Creates new form Nuevo_Cliente */
   public Nuevo_Cliente() {
      initComponents();
      Cod_NuevoProveedor();
      cargarComboprovincia();
      cargarComboformapago();
      cargarComboEntBancaria();
      txtnombre.requestFocus();
   }

   public void cargarComboprovincia() {
      try {
      Connection con = DriverManager.getConnection
("jdbc:mysql://localhost:3306/codeka", "root", "root");
      Statement stmt = con.createStatement();
      ResultSet rs = stmt.executeQuery ("SELECT * FROM provincias WHERE borrado=0
ORDER BY codprovincia ASC");
      cboprovincia.removeAllItems();
      cboprovincia.addItem("Seleccione una Provincia");
      while(rs.next())
```

```
        cboprovincia.addItem(rs.getObject(2));
        rs.close();
        stmt.close();
        con.close();
        } catch (SQLException e) {
        e.printStackTrace();
        }
        pack();
        setLocationRelativeTo (null);
    }

        public void cargarComboformapago() {
        try {
        Connection con = DriverManager.getConnection
("jdbc:mysql://localhost:3306/codeka", "root", "root");
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery ("SELECT * FROM formapago WHERE borrado=0
ORDER BY codformapago ASC");
        cboformapago.removeAllItems();
        cboformapago.addItem("Seleccione una Forma de Pago");
        while(rs.next())
        cboformapago.addItem(rs.getObject(2));
        rs.close();
        stmt.close();
        con.close();

        } catch (SQLException e) {
        e.printStackTrace();
        }
        pack();
```

```
        setLocationRelativeTo (null);
    }

    public void cargarComboEntBancaria() {
        try {
        Connection con = DriverManager.getConnection
("jdbc:mysql://localhost:3306/codeka", "root", "root");
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery ("SELECT * FROM entidades WHERE borrado=0
ORDER BY codentidad ASC");
        cboentbancaria.removeAllItems();
        cboentbancaria.addItem("Seleccione una Entidad Bancaria");
        while(rs.next())
        cboentbancaria.addItem(rs.getObject(2));
        rs.close();
        stmt.close();
        con.close();

        } catch (SQLException e) {
        e.printStackTrace();
        }
        pack();
        setLocationRelativeTo (null);
    }

    public void Limpiar(){
        txtcodigo.setText("");
        txtcodigopos.setText("");
        txtcorreo.setText("");
        txtcuentabancaria.setText("");
```

```
    txtdireccion.setText("");
    txtlocalidad.setText("");
    txtmovil.setText("");
    txtnombre.setText("");
    txtruc.setText("");
    txttelefono.setText("");
    txtweb.setText("");
    cargarComboprovincia();
    cargarComboEntBancaria();
    cargarComboformapago();
    Cod_NuevoProveedor();
    txtnombre.requestFocus();
}

public void Cod_NuevoProveedor() {
    Connection conn = null;
        Statement sta;
        ResultSet rsl;
    String codmax;
       int a;
       String n = "";
       String b = "";
        try {
       String userName="root";
       String password="root";
       String url="jdbc:mysql://localhost:3306/codeka";
       Class.forName("com.mysql.jdbc.Driver").newInstance();
       conn = DriverManager.getConnection(url,userName, password);
       //toma el valor de la base
                sta =conn.createStatement();
```

```
                rsl = sta.executeQuery("select codcliente from clientes order by codcliente
desc limit 0,1");
            while(rsl.next()){
                //System.out.println("\n Codigo:"+rs.getString("cli_codigo"));
            codmax=rsl.getString("codcliente");
            a=Integer.parseInt(codmax);
            a=a+1;
            b = b.valueOf(a);
            txtcodigo.setText(b);
            txtcodigo.enable(false);
            }
            if(txtcodigo.getText().equals("")){
               txtcodigo.setText("1");
               txtcodigo.enable(false);
                   }
                   rsl.close();
         } catch (Exception e) {
        txtcodigo.requestFocus();
                   System.out.println("Error al Abrir tabla ");
                   }

   }
      pack();
   }// </editor-fold>

   private void GuardarActionPerformed(java.awt.event.ActionEvent evt) {
      // TODO add your handling code here:
      int resp;
      resp=JOptionPane.showConfirmDialog(null,"¿Desea agregar el
registro?","Pregunta",0);
```

```
String comando;
if(resp==0){
   if (//!txtcodigo.getText().equals("")&&
        !txtnombre.getText().equals("")&&
        !txtruc.getText().equals("")&&
        !txtdireccion.getText().equals("")&&
        !txtlocalidad.getText().equals("")&&
        !txtcuentabancaria.getText().equals("")&&
        !txtcodigopos.getText().equals("")&&
        !txttelefono.getText().equals("")&&
        !txtmovil.getText().equals("")&&
        !txtcorreo.getText().equals("")&&
        !txtweb.getText().equals("")){
      try {
        String sql = ("INSERT INTO clientes "
            +
/*"(codproveedor,nombre,nif,direccion,codprovincia,localidad,codformapago,codentidad
,cuentabancaria,codpostal,telefono,movil,email,web,borrado) " +*/
            "VALUES ('" + txtcodigo.getText().trim() + "'," +
            "'" + txtnombre.getText().trim() + "'," +
            "'" + txtruc.getText().trim() + "'," +
            "'" + txtdireccion.getText().trim() + "'," +
            "'" + cboprovincia.getSelectedIndex() + "'," +
            "'" + txtlocalidad.getText().trim() + "'," +
            "'" + cboformapago.getSelectedIndex() + "'," +
            "'" + cboentbancaria.getSelectedIndex() + "'," +
            "'" + txtcuentabancaria.getText().trim() + "'," +
            "'" + txtcodigopos.getText().trim() + "'," +
            "'" + txttelefono.getText().trim() + "'," +
            "'" + txtmovil.getText().trim() + "'," +
```

```
                    """ + txtcorreo.getText().trim() + "'," +
                    """ + txtweb.getText().trim() + "'," + "'0'" + ")");
                System.out.println(sql);
                if (cn.ejecutarSQL(sql)== true) {
                   JOptionPane.showMessageDialog(null,"Registro Grabado", "Tabla Clientes",
JOptionPane.OK_OPTION);
                   Limpiar();
                   new Clientes().setVisible(true);
                   this.setVisible(false);
                }


             } catch (Exception sqle){
                JOptionPane.showMessageDialog(null,"ERROR", "Tabla Clientes ",
JOptionPane.OK_OPTION);
             }
          } else{
             JOptionPane.showMessageDialog(null, " UN CASILLERO ESTA VACIO ", "ERROR ",
JOptionPane.OK_OPTION);
          }
       }
}

   private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
      // TODO add your handling code here:
      Limpiar();
}

   private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
      // TODO add your handling code here:
```

```
        new Clientes().setVisible(true);
        this.setVisible(false);
}
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Nuevo_Cliente().setVisible(true);
            }
        });
    }
 }
```

Ver Cliente

```
package Clientes;

import java.sql.*;
import javax.swing.JOptionPane;
public class Ver_Cliente extends javax.swing.JFrame {

    /** Creates new form Ver_Cliente */
    public Ver_Cliente() {
        initComponents();
        txtcodigo.setText(Clientes.txtcodigo.getText());
        Cargar_Registros();
        Cargar_Registro_Provincia();
        Cargar_Registro_EntBancaria();
        Cargar_Registro_FormaPago();
    }

    public void Cargar_Registro_Provincia(){
```

```
try {
Connection con = DriverManager.getConnection ("jdbc:mysql://localhost:3306/codeka", "root", "root");
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery ("SELECT pro.nombreprovincia FROM clientes as p, provincias as pro WHERE p.codcliente =" + "'" + txtcodigo.getText().trim() + "'" + " AND p.codprovincia=pro.codprovincia");
while(rs.next())
txtprovincia.setText(rs.getString("pro.nombreprovincia"));
rs.close();
stmt.close();
con.close();

} catch (SQLException e) {
e.printStackTrace();
}
pack();
setLocationRelativeTo (null);
}
public void Cargar_Registro_EntBancaria(){
try {
Connection con = DriverManager.getConnection ("jdbc:mysql://localhost:3306/codeka", "root", "root");
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery ("SELECT e.nombreentidad FROM clientes as p, entidades as e WHERE p.codcliente =" + "'" + txtcodigo.getText().trim() + "'" + " AND p.codentidad=e.codentidad");
while(rs.next())
txtentidad.setText(rs.getString("e.nombreentidad"));
rs.close();
```

```
        stmt.close();
        con.close();

        } catch (SQLException e) {
        e.printStackTrace();
        }
        pack();
        setLocationRelativeTo (null);
    }

        public void Cargar_Registro_FormaPago(){
        try {
        Connection con = DriverManager.getConnection ("jdbc:mysql://localhost:3306/codeka", "root", "root");
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery ("SELECT e.nombrefp FROM clientes as p, formapago as e WHERE p.codcliente =" + "'" + txtcodigo.getText().trim() + "'" + " AND p.codformapago=e.codformapago");
        while(rs.next())
        txtformapago.setText(rs.getString("e.nombrefp"));
        rs.close();
        stmt.close();
        con.close();

        } catch (SQLException e) {
        e.printStackTrace();
        }
        pack();
        setLocationRelativeTo (null);
    }
```

```
public void Cargar_Registros(){
   String b = txtcodigo.getText();
   if (!txtcodigo.getText().equals("")){
      try {
      Connection con = DriverManager.getConnection
("jdbc:mysql://localhost:3306/codeka", "root", "root");
   Statement stmt = con.createStatement();
   ResultSet rs = stmt.executeQuery ("SELECT * FROM clientes");
            while(rs.next()){
         if(b.equals(rs.getString("codcliente"))){
         txtnombre.setText(rs.getString("nombre"));
         txtruc.setText(rs.getString("nif"));
         txtdireccion.setText(rs.getString("direccion"));
         txtlocalidad.setText(rs.getString("localidad"));
         txtcuentabancaria.setText(rs.getString("cuentabancaria"));
         txtcodigopos.setText(rs.getString("codpostal"));
         txttelefono.setText(rs.getString("telefono"));
         txtmovil.setText(rs.getString("movil"));
         txtcorreo.setText(rs.getString("email"));
         txtweb.setText(rs.getString("web"));
         }
       }
            rs.close();

      } catch (Exception e) {
            System.out.println("Error al Abrir tabla ");
            }
  }
   else{
```

```
        JOptionPane.showMessageDialog(null, " NO EXISTE CLIENTE ", "ERROR ",
JOptionPane.OK_OPTION);
        }
    }
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Ver_Cliente().setVisible(true);
            }
        });
    }
}
```

Pedido

```
package Pedidos;

import Conexion.conexion;
import java.util.Vector;
import javax.swing.table.DefaultTableModel;
public class Pedido extends javax.swing.JFrame {
conexion cn = new conexion();
    /** Creates new form Pedido */
    public Pedido() {
        initComponents();
        setLocationRelativeTo(this);
        Cargar_Datos();
    }

    public void Cargar_Datos(){
                    Vector  filas = new Vector();
```

```
            String CamposTabla[] =new String [5];
            CamposTabla[0]="N_Pedido";
            CamposTabla[1]="Cliente";
      CamposTabla[2]="Importe";
      CamposTabla[3]="Fecha";
      CamposTabla[4]="Estado";
      //CamposTabla[4]="cli_email";
            filas=cn.cargarDatos("SELECT codalbaran as N_Pedido," +
           "clientes.nombre as Cliente," +
           "albaranes.fecha as Fecha," +
           "totalalbaran as Importe," +
           "estado as Estado" +
           " FROM albaranes,clientes " +
           "WHERE albaranes.borrado=0 " +
           "AND albaranes.codcliente=clientes.codcliente", CamposTabla);
            Vector columnas= new Vector();
            for(int counter=0;counter<CamposTabla.length;counter++){
                  columnas.addElement(CamposTabla[counter]);
                  }
            DefaultTableModel modelo= new DefaultTableModel (filas,columnas)
   {
   @Override
   public boolean isCellEditable (int row,int column){
            return true;
   }
   };
   tabla.setModel(modelo);
tabla.getColumnModel().getColumn(0).setMaxWidth(60);
   }
```

```
public void Consulta_like(){
            Vector  filas = new Vector();

            String CamposTabla[] =new String [5];
            CamposTabla[0]="N_Pedido";
            CamposTabla[1]="Cliente";
      CamposTabla[2]="Importe";
      CamposTabla[3]="Fecha";
      CamposTabla[4]="Estado";
            String Sql="SELECT albaranes.codalbaran as N_Pedido," +
           "clientes.nombre as Cliente," +
           "albaranes.fecha as Fecha," +
           "totalalbaran as Importe," +
           "estado as Estado" +
           " FROM albaranes,clientes " +
           "WHERE albaranes.codalbaran like " +
                        "'" + txtnumpedido.getText().trim() + "%'" +
               "and clientes.nombre like " +
                        "'" + txtnombre.getText().trim() + "%'" +
               "and albaranes.borrado=0 " +
               "and albaranes.codcliente = clientes.codcliente";
            filas=cn.cargarDatos(Sql,CamposTabla);
            Vector columnas= new Vector();
            for(int counter=0;counter<CamposTabla.length;counter++){
                  columnas.addElement(CamposTabla[counter]);
                  }
            DefaultTableModel modelo= new DefaultTableModel (filas,columnas)
      {
      @Override
```

```
        public boolean isCellEditable (int row,int column){
                return true;
        }
        };
        tabla.setModel(modelo);
    tabla.getColumnModel().getColumn(0).setMaxWidth(70);
        }
    );

    pack();
}// </editor-fold>

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new Nuevo_Pedido().setVisible(true);
    this.setVisible(false);
}

private void tablaMouseReleased(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
}

private void txtnombreKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    Consulta_like();
}

private void txtnumpedidoKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    Consulta_like();
```

```
    }

    private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        new Login.Principal().setVisible(true);
        this.setVisible(false);
    }
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Pedido().setVisible(true);
            }
        });
    }
}
```

Nuevo Pedido

```
package Pedidos;

import Conexion.conexion;
import java.sql.*;
import java.util.Date;
import java.util.Vector;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
public class Nuevo_Pedido extends javax.swing.JFrame {
conexion cn = new conexion();
Date d = new Date();
    /** Creates new form Nuevo_Pedido */
    public Nuevo_Pedido() {
```

```
        initComponents();
        setLocationRelativeTo(this);
        txtfecha.setText((d.getYear() + 1900)+"/"+(d.getMonth()+1)+"/"+d.getDate());
        txtiva.setText("12");
        cargarComboFP();
        cargarComboCob();
        cargarComboVen();
        Generar_Cod_Albaran();
        Cod_Nuevo();
    }

    public void cargarComboFP() {
        try {
        Connection con = DriverManager.getConnection
("jdbc:mysql://localhost:3306/codeka", "root", "root");
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery ("SELECT nombrefp FROM formapago WHERE
borrado=0 ORDER BY codformapago ASC");
        cboformapago.removeAllItems();
        cboformapago.addItem("Seleccione Forma de Pago");
        while(rs.next())
        cboformapago.addItem(rs.getObject(1));
        rs.close();
        stmt.close();
        con.close();
        } catch (SQLException e) {
        e.printStackTrace();
        }
        pack();
        setLocationRelativeTo (null);
```

```
    }

    public void cargarComboCob() {
        try {
        Connection con = DriverManager.getConnection
("jdbc:mysql://localhost:3306/codeka", "root", "root");
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery ("SELECT nombre FROM cobradores WHERE
borrado=0 ORDER BY codcobrador ASC");
        cbocobradores.removeAllItems();
        cbocobradores.addItem("Seleccione un Cobrador");
        while(rs.next())
        cbocobradores.addItem(rs.getObject(1));
        rs.close();
        stmt.close();
        con.close();
        } catch (SQLException e) {
        e.printStackTrace();
        }
        pack();
        setLocationRelativeTo (null);
    }

    public void cargarComboVen() {
        try {
        Connection con = DriverManager.getConnection
("jdbc:mysql://localhost:3306/codeka", "root", "root");
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery ("SELECT nombre FROM vendedores WHERE
borrado=0 ORDER BY codvendedor ASC");
```

```
    cbovendedores.removeAllItems();
    cbovendedores.addItem("Seleccione un Vendedor");
    while(rs.next())
    cbovendedores.addItem(rs.getObject(1));
    rs.close();
    stmt.close();
    con.close();
    } catch (SQLException e) {
    e.printStackTrace();
    }
    pack();
    setLocationRelativeTo (null);
  }

  public void Cargar_Datos(){
                Vector  filas = new Vector();

                String CamposTabla[] =new String [7];
                CamposTabla[0]="Item";
                CamposTabla[1]="Referencia";
          CamposTabla[2]="Descripcion";
          CamposTabla[3]="Cantidad";
          CamposTabla[4]="Precio";
          CamposTabla[5]="Dcto";
          CamposTabla[6]="Importe";
          filas=cn.cargarDatos("SELECT albalineatmp.numlinea as Item,articulos.referencia
as Referencia, articulos.descripcion as Descripcion," +
                "albalineatmp.cantidad as Cantidad,albalineatmp.precio as
Precio,albalineatmp.dcto as Dcto,albalineatmp.importe as Importe " +
                "FROM albalineatmp,articulos,familias " +
```

```
                "WHERE albalineatmp.codalbaran='" + txtcodalbarantmp.getText() + "' AND
albalineatmp.codigo=articulos.codarticulo " +
                "AND albalineatmp.codfamilia=articulos.codfamilia AND
articulos.codfamilia=familias.codfamilia " +
                "ORDER BY albalineatmp.numlinea ASC", CamposTabla);
                 Vector columnas= new Vector();
                 for(int counter=0;counter<CamposTabla.length;counter++){
                         columnas.addElement(CamposTabla[counter]);
                         }
                 DefaultTableModel modelo= new DefaultTableModel (filas,columnas)
        {
        @Override
        public boolean isCellEditable (int row,int column){
                 return true;
        }
        };
        tabla_pedidos.setModel(modelo);
     tabla_pedidos.getColumnModel().getColumn(0).setMinWidth(50);
     tabla_pedidos.getColumnModel().getColumn(0).setMaxWidth(50);
     tabla_pedidos.getColumnModel().getColumn(1).setMinWidth(150);
     tabla_pedidos.getColumnModel().getColumn(1).setMaxWidth(150);
     tabla_pedidos.getColumnModel().getColumn(2).setMinWidth(350);
     tabla_pedidos.getColumnModel().getColumn(2).setMaxWidth(350);
        }

  public void Cod_Nuevo() {
     Connection conn = null;
        Statement sta;
        ResultSet rsl;
     String codmax;
```

```
        int a;
        String b = null;
         try {
        String userName="root";
        String password="root";
        String url="jdbc:mysql://localhost:3306/codeka";
        Class.forName("com.mysql.jdbc.Driver").newInstance();
        conn = DriverManager.getConnection(url,userName, password);
        //toma el valor de la base
                sta =conn.createStatement();
                rsl = sta.executeQuery("select codalbaran from albaranestmp " +
               "order by codalbaran desc limit 0,1");
                while(rsl.next()){
          txtcodalbarantmp.setText(rsl.getString("codalbaran"));
                }
          if(txtcodalbarantmp.getText().equals("")){
             txtcodalbarantmp.setText("1");
                }
                rsl.close();
         } catch (Exception e) {
                System.out.println("Error al Abrir tabla ");
                }
}

   public static void Cod_articulo() {
      Connection conn = null;
         Statement sta;
         ResultSet rsl;
      String codmax;
        int a;
```

```
        String b = null;
         try {
        String userName="root";
        String password="root";
        String url="jdbc:mysql://localhost:3306/codeka";
        Class.forName("com.mysql.jdbc.Driver").newInstance();
        conn = DriverManager.getConnection(url,userName, password);
        //toma el valor de la base
                sta =conn.createStatement();
                rsl = sta.executeQuery("select codarticulo from articulos " +
               "where descripcion='" + txtdes.getText() + "'");
                while(rsl.next()){
                //System.out.println("\n Codigo:"+rs.getString("cli_codigo"));
          codmax=rsl.getString("codarticulo");
          txtcodproducto.setText(codmax);
                }
                rsl.close();
         } catch (Exception e) {
                System.out.println("Error al Abrir tabla ");
                }
}

   public static void Cod_Familia() {
      Connection conn = null;
         Statement sta;
         ResultSet rsl;
      String codmax;
        int a;
        String b = null;
         try {
```

```
        String userName="root";
        String password="root";
        String url="jdbc:mysql://localhost:3306/codeka";
        Class.forName("com.mysql.jdbc.Driver").newInstance();
        conn = DriverManager.getConnection(url,userName, password);
        //toma el valor de la base
                sta =conn.createStatement();
                rsl = sta.executeQuery("select codfamilia from familias " +
               "where nombre='" + txtcodfamilia.getText() + "'");
                while(rsl.next()){
                //System.out.println("\n Codigo:"+rs.getString("cli_codigo"));
          codmax=rsl.getString("codfamilia");
          txtcodfamilia.setText(codmax);
                }
                rsl.close();
        } catch (Exception e) {
                System.out.println("Error al Abrir tabla ");
                }
}

   public void Suma_Importe() {
      Connection conn = null;
         Statement sta;
         ResultSet rsl;
      String codmax;
        int a;
        String b = null;
         try {
        String userName="root";
        String password="root";
```

```
        String url="jdbc:mysql://localhost:3306/codeka";
        Class.forName("com.mysql.jdbc.Driver").newInstance();
        conn = DriverManager.getConnection(url,userName, password);
        //toma el valor de la base
        if(!txtcodalbarantmp.getText().equals("")){
                sta =conn.createStatement();
                rsl = sta.executeQuery("SELECT SUM(importe) as imp FROM
`albalineatmp`WHERE `codalbaran` ='" + txtcodalbarantmp.getText() + "'");
                while(rsl.next()){
                //System.out.println(rsl.getString("imp"));
          txtsubtotal.setText(rsl.getString("imp"));
          Float subto = Float.valueOf(txtsubtotal.getText());
          Float subiva = Float.valueOf(txtiva.getText());
          Float multiplica = subto * (subiva/100);
          txtivasubtotal.setText(Float.toString(multiplica));
          Float aux = Float.valueOf(txtivasubtotal.getText());
          Float suma = subto + aux ;
          txttotal.setText(Float.toString(suma));
                }
                rsl.close();
        }
         } catch (Exception e) {
                System.out.println("Error al Abrir tabla ");
                }
}

   public void Generar_Cod_Albaran(){
      try {
        String sql = "INSERT INTO albaranestmp (codalbaran,fecha) " +
             "VALUE ('0',"+
```

```
        """ + txtfecha.getText().trim()+ "')";
        if (cn.ejecutarSQL(sql)== true)
        {
        }
        }
        catch (Exception sqle){
           JOptionPane.showMessageDialog(null,"ERROR", "", JOptionPane.OK_OPTION);
        }
}
   pack();
}// </editor-fold>

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
   // TODO add your handling code here:
   new Buscar_Cliente().setVisible(true);
}

private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {
   // TODO add your handling code here:
   Connection conn = null;
        Statement sta;
        ResultSet rsl;
        try {
       String userName="root";
       String password="root";
       String url="jdbc:mysql://localhost:3306/codeka";
       Class.forName("com.mysql.jdbc.Driver").newInstance();
       conn = DriverManager.getConnection(url,userName, password);
       //toma el valor de la base
                sta =conn.createStatement();
```

```
                rsl = sta.executeQuery("select * from clientes where borrado = 0 " +
               "and codcliente=" + txtcodigo.getText() );
                while(rsl.next()){
                //System.out.println("\n Codigo:"+rsl.getString("nombre"));
          txtnombre.setText(rsl.getString("nombre"));
          txtruc.setText(rsl.getString("nif"));
                }
                rsl.close();
       } catch (Exception e) {
                System.out.println("Error al Abrir tabla ");
                }
}

private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {
   // TODO add your handling code here:
   txtref.setText("");
   txtdes.setText("");
   txtprecio.setText("");
   txtcant.setText("");
   txtimporte.setText("");
   txtdcto.setText("");
   if(!txtcodigo.getText().equals("")){
      new Buscar_Articulo().setVisible(true);
   }
   else{
      JOptionPane.showMessageDialog(rootPane, "Debe Seleccionar un Cliente");
   }
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    if (!txtref.getText().equals("")&& !txtdes.getText().equals("")&&
!txtprecio.getText().equals("")&&
            !txtcant.getText().equals("")&& !txtimporte.getText().equals("")&&
!txtdcto.getText().equals("")){
        try {
            String sql = "INSERT INTO albalineatmp
(codalbaran,numlinea,codigo,codfamilia,cantidad,precio,importe,dcto) " +
                  "VALUES(" +
            "'" + txtcodalbarantmp.getText().trim()+ "', '0' ,"+
            "'" + txtcodproducto.getText().trim()+ "',"+
            "'" + txtcodfamilia.getText().trim()+ "',"+
            "'" + txtcant.getText().trim()+ "',"+
            "'" + txtprecio.getText().trim()+ "',"+
            "'" + txtimporte.getText().trim()+ "',"+
            "'" + txtdcto.getText().trim()+ "')";
            if (cn.ejecutarSQL(sql)== true)
            {
            }
            Cargar_Datos();
            Suma_Importe();
            txtref.setText("");
            txtdes.setText("");
            txtprecio.setText("");
            txtcant.setText("");
            txtimporte.setText("");
            txtdcto.setText("");
        }
        catch (Exception sqle){
            JOptionPane.showMessageDialog(null,"ERROR", "", JOptionPane.OK_OPTION);
```

```
        }
      }
      else{
        JOptionPane.showMessageDialog(null, " UN CASILLERO ESTA VACIO ", "ERROR ",
JOptionPane.OK_OPTION);
      }
   }

   private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
      // TODO add your handling code here:
      if (!txtlinea.getText().equals("")){
      int resp;
  resp=JOptionPane.showConfirmDialog(null,"¿Desea eliminar el registro?","Pregunta",0);
      String comando;
  if(resp==0){
      try {
        String sql = " DELETE FROM albalineatmp " +
        "WHERE numlinea= " + "'" + txtlinea.getText().trim() + "'";
        if (cn.ejecutarSQL(sql)== true)
        {
              JOptionPane.showMessageDialog(null,"Registro Borrado", "",
JOptionPane.OK_OPTION);
        }
        Cargar_Datos();
        txtref.setText("");
        txtdes.setText("");
        txtprecio.setText("");
        txtcant.setText("");
        txtimporte.setText("");
        txtdcto.setText("");
```

```
        }
        catch (Exception sqle){
              JOptionPane.showMessageDialog(null,"ERROR", "", JOptionPane.OK_OPTION);
        }
  }
  }
      else {
              JOptionPane.showMessageDialog(null, "DEBE SELECCIONAR EL ITEM DE LA
TABLA", "ERROR ", JOptionPane.OK_OPTION);
        }
   }

   private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
      // TODO add your handling code here:
      new Guardar_Pedido().setVisible(true);
      this.setVisible(false);
      /*
if (!txtcodigo.getText().equals("")&& !txtnombre.getText().equals("")&&
   !txtfecha.getText().equals("")&&
   !cboformapago.getSelectedItem().equals("Seleccione Forma de Pago") &&
   !cbocobradores.getSelectedItem().equals("Seleccione un Cobrador")&&
   !cbovendedores.getSelectedItem().equals("Seleccione un Vendedor")&&
   !txtruc.getText().equals("")&& !txtiva.getText().equals("")){
   try {
        String sql = "INSERT INTO albaranes (codalbaran, codfactura, fecha, iva, " +
        "codcliente, estado,totalalbaran, borrado, nref, codvendedor, codformapago,
codcobrador) " +
        "VALUES ('0', '0'," +
        "'" + txtfecha.getText().trim()+ "',"+
        "'" + txtiva.getText().trim()+ "',"+
```

```
        "'" + txtcodigo.getText().trim()+ "','1',"+
        "'" + txttotal.getText().trim()+ "','0',"+
        "'" + txtnref.getText().trim()+ "',"+
        "'" + cbovendedores.getSelectedIndex()+ "',"+
        "'" + cboformapago.getSelectedIndex()+ "',"+
        "'" + cbocobradores.getSelectedIndex() + "')";
        if (cn.ejecutarSQL(sql)== true)
        {
           new Alta_Pedido().setVisible(true);
           this.setVisible(false);
        }

        Cargar_Datos();
   }
   catch (Exception sqle){
        JOptionPane.showMessageDialog(null,"ERROR", "Tabla Clientes ",
JOptionPane.OK_OPTION);
   }
}
else{
   JOptionPane.showMessageDialog(null, " UN CASILLERO ESTA VACIO ", "ERROR ",
JOptionPane.OK_OPTION);
}
   }

   private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
      // TODO add your handling code here:
      if(!txtcodalbarantmp.getText().equals("")){
      try {
         String sql = " DELETE FROM albalineatmp " +
```

```
        "WHERE codalbaran= " + "'" + txtcodalbarantmp.getText().trim() + "'";
        if (cn.ejecutarSQL(sql)== true)
        {
           new Pedido().setVisible(true);
           this.setVisible(false);
        }
        else{
           new Pedido().setVisible(true);
           this.setVisible(false);
        }
        }
        catch (Exception sqle){
           JOptionPane.showMessageDialog(null,"ERROR", "", JOptionPane.OK_OPTION);
        }
}
 }

  private void txtcantKeyReleased(java.awt.event.KeyEvent evt) {
     // TODO add your handling code here:
     float importe;
     String precio = txtprecio.getText();
     String cantidad = txtcant.getText();
     importe = Float.valueOf(precio)* Float.valueOf(cantidad);
     txtimporte.setText(String.valueOf(importe));
  }
  public static void main(String args[]) {
     java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
           new Nuevo_Pedido().setVisible(true);
        }
```

```
        });
    }
}
```

Anexo 3.- Cronograma de Actividades

Anexo 4.- Presupuesto

PRESUPUESTO			
NOMBRE	DESCRIPCIÓN	VALOR UNITARIO	VALOR TOTAL
MAQUINARÍA Y HERRAMIENTAS			
Computadora	HP DM4-1160 US CORE 5	780, 00	780, 00
Impresora	Lexmark X1185	100, 00	100, 00
Internet	ISP PUNTO NET	360, 00	360, 00
MATERIALES Y SUMINISTROS			
Resma de Papel Bond	NORMA	4,00	4,00
Flash Memory	Kingston de 8GB	18,00	18,00
Carpetas	Perfiles	2, 00	10, 00
SERVICIOS BÁSICOS		120,00	120,00
VARIOS		3, 50	7, 00
Transporte		0,25	150,00
Celular	Nokia/Movistar		50,00
Impresiones		0,03	100
		TOTAL	1669,00