



INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

INSTITUTO TECNOLÓGICO SUPERIOR
CORDILLERA

ESCUELA DE SISTEMAS

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

TEMA:

SISTEMA DE FACTURACIÓN Y CONTROL DE INVENTARIOS
PARA LA EMPRESA FFHV EN EL CENTRO DE QUITO.

AUTOR:

Gina Cumbicus

TUTOR:

Ing. Rodrigo Cobos

2011

QUITO – ECUADOR



CAPÍTULO I

EL PROBLEMA

1.1. Planteamiento del Problema

El Sistema de Facturación y Control de Inventarios para la empresa FFHV surge por la necesidad de agilizar los procesos de compra y venta, además de llevar de una manera más segura y organizada la información, obteniendo datos precisos de los materiales existentes en bodega. Cabe recalcar que la implementación del Sistema se lo realizará solo en la bodega matriz.

Actualmente el proceso de facturación e inventario se lo realiza en hojas electrónicas de cálculo, lo que causa una gran pérdida de tiempo e información errónea, además debido a que la empresa ha crecido rápidamente, la información ha aumentado significativamente por lo que el usuario esta propenso a cometer una gran cantidad de errores.

Ya que no se manejan datos reales de los materiales existentes en bodega esto hace que se produzcan grandes pérdidas económicas debido al desconocimiento de esta información.

1.2. Formulación del Problema

¿Será posible la elaboración de un Sistema de Facturación y control de inventarios que agilite los procesos dentro de la empresa?

1.3. Delimitación del problema.

Para la elaboración del referido proyecto se contará con un periodo de tiempo de 5 meses, el mismo que se desarrollara en la oficina central ubicada en la avenida 24 de



INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

Mayo y Cuenca en el centro de Quito. El sistema no tendrá costo alguno para la empresa, todos los gastos corren por cuenta del desarrollador.

A continuación se presenta de forma gráfica el lugar exacto donde se encuentra ubicada la empresa.



UBICACIÓN BODEGA PRINCIPAL 24 DE MAYO
FIGURA N° 1.3
FUENTE: Internet (Google Maps)

1.4. Objetivos

1.4.1 Objetivo General

Diseñar, desarrollar e implementar un Sistema de Facturación y Control de Inventarios que permita controlar todo el proceso de ventas y compras en la empresa FFHV.

1.4.2 Objetivos específicos

- Analizar los procesos y reglas del negocio.
- Diseñar los procesos y reglas del negocio.
- Desarrollar el código fuente.



- Realizar los procesos.
- Implementar El Sistema.

1.5. Justificación e Importancia

FFHV llevará a cabo un sistema de facturación e inventario que agilite los procesos dentro de la empresa, logrando ahorrar tiempo y de esta manera brindar una mejor atención a sus clientes.

Con la implementación del Sistema se evitará que el empleado cometa la menor cantidad de errores posibles, además se tendrá información real en el momento que sea requerida y así tanto el cliente como el empleado se sentirán satisfechos, esto mejorará los ingresos económicos de la empresa.

El referido proyecto permitirá que la empresa tenga un manejo controlado y ordenado de la información así como la fácil ubicación de la misma en la menor cantidad de tiempo.

A través de los privilegios que se le conceda al usuario para ingresar al sistema, se garantiza total seguridad de la información y se podrá sacar respaldos de la misma en archivos magnéticos.

1.6. Alcance

El sistema de acuerdo a los módulos requeridos por la empresa es extenso, por lo tanto se desarrollará un prototipo funcional hasta la fecha de la defensa y posteriormente, en un tiempo establecido de cuatro meses se completará el sistema de acuerdo a los requerimientos de la empresa.

Por lo tanto a continuación detallo el alcance del prototipo al igual que el alcance del sistema completo.



1.6.1 Alcance del Prototipo

Compras:

- Recepción de Compras
- Devolución de Compras
- Elaboración de la Factura de Compra.

Ventas:

- Factura de Venta
- Devolución de Ventas
- Nuevos Clientes

Inventario de Material:

- Módulo de Existencias

Caja:

- Ingresos:
 - Por Venta
 - Por Devolución de Material del Proveedor
- Cierre de Caja

Mantenimiento:

- Empleados:
 - Nuevos Empleados
 - Actualización de Empleados
- Materiales:
 - Nuevos Materiales
 - Actualización de Materiales
- Seguridad del Sistema:
 - Control de Usuarios

Reportes:



INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

- Factura Compra
- Factura Venta
- Clientes
- Proveedores
- Kardex:
 - Saldo de Materiales
 - Movimiento de Materiales
- Empleados.
- Caja
- Inventario

1.6.2. Alcance del Sistema Completo:

Compras:

- Actualización del Inventario:
 - Ingreso por Compra
 - Egreso por Devolución
 - Anulación Factura de Compra

Ventas:

- Descuentos:
 - Normales
 - Especiales
 - Anulación Factura de Venta
- Actualización del Inventario:
 - Egreso por Venta
 - Ingreso por Devolución

Inventario de Material:

- Transferencias:
 - Egresos de Bodega



INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

- Ingresos de Bodega
- Devolución por Transferencia
- Módulo de Existencias
 - Conteo Físico
 - Ajustes de Inventario

Caja:

- Egresos:
 - Por transferencia de Dinero
- Ingresos:
 - Por Transferencia

Reportes:

- Transferencias
- Reportes de Auditoria

Varios:

- Consolidación Materiales
- Consolidación Dineros
- Reportes



CAPITULO II

MARCO TEÓRICO

2.1. Antecedentes

Ubicación.- FFHV es una empresa que está legalmente constituida, se encuentra ubicada en la avenida 24 de mayo al centro de Quito.

Actividad.- FFHV es una empresa que se dedica a la compra y venta de material metalúrgico.

Visión.- Brindar la mejor atención a los clientes, ofreciendo un precio justo, y generando un buen margen de rentabilidad con sus ventas.

Misión.- Satisfacer las necesidades económicas de los clientes a través de un servicio eficiente y precios competitivos.

2.2. Reseña histórica

FFHV es una empresa que se dedica a la compra y venta de material metalúrgico en la Av. 24 de Mayo ubicado en el Centro Histórico de la ciudad de Quito, tiene 10 años de permanencia en el mercado y mantiene convenios con algunas Empresas. Su objetivo es brindar la mejor atención posible a sus clientes, ofertando un precio y peso justo en la venta de los materiales. Entre los servicios que brinda están compra y venta de los siguientes materiales:

- Cobre
- Aluminio
- Perfil



- Bronce
- Hierro Fundido
- Chatarra
- Baterías
- Acero Inoxidable

La empresa FFHV siempre ha llevado la facturación en forma manual, la cual causa que se desperdicie mucho tiempo en procesos de la empresa que podrían ser más eficientes al ser automatizados. FFHV desea un Sistema de Facturación y Control de Inventarios con el cual se pueda ahorrar tiempo y brinde un mejor servicio.

Esta será la primera aplicación que utilizará la empresa para poder llevar a cabo la facturación por medio de una computadora. Anteriormente se hacía la facturación a mano, pero ahora que están en crecimiento quieren agilizar un poco más el proceso de compra-venta y ser más eficientes.

2.3. Marco Referencial

Factura. Una factura es un documento que detalla los bienes o servicios vendidos o prestados por una parte a la otra, con indicación de cantidades y precios. En las operaciones mercantiles se emiten facturas comerciales: por las compras y por las ventas.

Factura de compra. Cuando una empresa compra productos, su proveedor le envía una factura. En la factura de compra aparecen los detalles completos relativos a los productos y a su precio. Cuando la compra es al contado, la factura lleva la mención de haber sido pagada y cumple la función de un recibo.

Factura de venta. Cuando una empresa vende sus productos, envía una factura al cliente. En dicho documento se dan los detalles completos de los productos, precio y otros datos por ejemplo, fecha y número del pedido del cliente, procedimiento de entrega, descuentos, recargos por impuestos, etc.



Una vez que se procede al pago de la orden de pedido, es imposible modificar la factura. Por ello le aconsejamos que compruebe y verifique los datos de su factura antes de realizar el pago. Una vez que la orden de pedido se ha emitido correctamente (puede emitir tantas como quiera), puede proceder al pago de ésta de manera totalmente segura. Por esto la empresa llevara un prototipo de facturación automatizado.

Control de Inventarios. En la empresa el inventario es por lo general, el activo mayor en los balances a si también los gastos por inventarios, llamados costos de mercancías vendidas, son usualmente los gastos mayores en el estado de resultado. A aquellas empresas dedicadas a la compra y venta de mercancías, por ser esta su principal función y la que da origen a todas las restantes operaciones, necesitan de una constante información resumida y analizadas sobre sus inventarios, lo cual obliga a la apertura de unas series de cuentas principales y auxiliares relacionadas con estos controles. Entre las cuentas podemos mencionar las siguientes:

- Inventarios (inicial)
- Compras
- Ventas
- Ingreso de material
- Egreso de material
- Inventarios (final)

Los programas a utilizar son:

Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el bytecode es normalmente interpretado o compilado a código nativo para la ejecución,



aunque la ejecución directa por hardware del bytecode por un procesador Java también es posible.

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit y el compilador que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse).

Eclipse dispone de un Editor de texto con resaltado de sintaxis. La compilación es en tiempo real.

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

MySQL es un software de fuente abierta. Fuente abierta significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar, es patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

Power Designer es una suite de aplicaciones de Powersoft para la construcción, diseño y modelado de datos a través de diversas aplicaciones. Es una herramienta para el análisis, diseño inteligente y construcción sólida de una base de datos y un desarrollo orientado a modelos de datos a nivel físico y conceptual.

PowerDesigner ofrece las posibilidades de:

- Soporte para tipos de datos abstractos: PowerDesigner soporta la identificación de tipos de datos abstractos con ingeniería inversa de aplicaciones para Oracle.



- Soporte para usuarios de bases de datos: Los usuarios de bases de datos pueden ser recogidos de una base de datos existente y luego almacenados en un modelo físico de datos. Ahora, es posible añadir nuevos usuarios y también asignar usuarios como propietarios y vistas.
- Mayor selectividad en ingeniería inversa: PowerDesigner permite seleccionar no sólo las tablas que se desean cargar, sino todo tipo de objetos de la base de datos.
- Cálculo del tamaño de las bases de datos: Puede calcular y definir el tamaño definitivo de bases de datos de nuevo diseño y construcción, incluyendo tamaños detallados de índices y tablas.

2.4. Marco Legal

Para la realización de cualquier clase de proyecto dentro del territorio ecuatoriano, este debe cumplir con todas las normas, leyes y reglamentos, para que de esta manera pueda desarrollarse sin inconvenientes. El presente plan de proyecto se rige de acuerdo a todo el Marco Legal, Político y Administrativo que norma nuestro deber como ciudadanos, el cual se describe a continuación:

- **LEY DE LA PROPIEDAD INTELECTUAL**

Registro Oficial No 320 **Ley de Propiedad Intelectual**

TITULO PRELIMINAR

- **LEY DE COMPANIAS**

Codificación No. 000. RO/ 312 de 5 de noviembre de 1999.

TITULO PRELIMINAR

Para el desarrollo de la aplicación se usará el lenguaje de programación Java y para la implementación de la base de datos se utilizara MySql, los cuales se rigen por la licencia de Software Libre GNU (Licencia Pública General).



GNU es una licencia creada por la Free Software Fundación en 1989, y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

Para el diseño de la base de datos tanto conceptual como físicamente se utilizará el aplicativo Power Designer el cual tiene una licencia educativa que será proporcionada por el Instituto Tecnológico Cordillera.

2.5. Marco Conceptual

Bienes.- son aquellos que se consiguen previo pago de un precio en el mercado en el cual se los comercializa. Estos a su vez se subdividen en bienes materiales o inmateriales, pero cualquiera sea el caso poseerán un valor y serán susceptibles de ser valuados en términos económicos.

Recibo.- es una constancia que sirve para certificar que se ha pagado por un servicio o producto.

Automatizar.- aplicar procedimientos automáticos a un aparato proceso o sistema.

Activo.- un bien de una entidad que pueden convertirse de forma directa en tesorería.

Balance.- informe financiero que refleja la situación del patrimonio de una entidad en un momento determinado.

Mercancía.- es todo aquello que se puede vender o comprar, usualmente el término se aplica a bienes económicos

Lenguaje de Programación.- es un idioma artificial diseñado para expresar



computaciones que pueden ser llevadas a cabo por máquinas como las computadoras.

Puntero.- es una variable que referencia una región de memoria; en otras palabras es una variable cuyo valor es una dirección de memoria.

Bytecode.- es un código intermedio más abstracto que el código máquina.

Compilador es un programa informático que traduce un programa escrito en un lenguaje de programación a otro lenguaje de programación, generando un programa equivalente que la máquina será capaz de interpretar.

Hardware.- corresponde a todas las partes físicas y tangibles¹ de una computadora.

Entorno.- se trata de un software que permite que un usuario pueda utilizar una computadora de manera sencilla y amigable gracias a una adecuada interfaz gráfica.

Aplicación.- es un tipo de programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de trabajo.

Navegador.- es un programa que permite ver la información que contiene una página web.

Plataforma.- se refiere al sistema operativo o a sistemas complejos que a su vez sirven para crear programas, como las plataformas de desarrollo.

IDE.- Un entorno de desarrollo informático (en inglés integrated development environment) es un programa informático compuesto por un conjunto de herramientas de programación.

Clase.- es una construcción que se utiliza como un modelo (o plantilla) para crear objetos



de esa clase.

Base de Datos.- (en ocasiones abreviada B.D.D.) es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

Multihilo.- son aquellas que dividen su carga de trabajo en diversos Hilos o unidades de ejecución.

Multiusuario.- se refiere a un concepto de sistemas operativos, pero en ocasiones también puede aplicarse a programas de ordenador de otro tipo (ej. aplicaciones de base de datos).

Oracle.- es un sistema de gestión de base de datos relacional desarrollado por Oracle Corporation.

Ingeniería Inversa.- es un método de resolución. Aplicar ingeniería inversa a algo supone profundizar en el estudio de su funcionamiento, hasta el punto de que podamos llegar a entender, modificar y mejorar dicho modo de funcionamiento.

CAPÍTULO III



METODOLOGÍA

3.1. Métodos de Investigación

Los métodos de Investigación que se utilizará para el debido desarrollo del proyecto son: Deductivo e Inductivo.

3.1.1. Método Deductivo

La deducción va de lo general a lo particular. El método deductivo es aquél que parte de datos generales aceptados como válidos, para deducir por medio del razonamiento lógico, varias suposiciones, es decir, parte de verdades previamente establecidas como principios generales, para luego aplicarlo a casos individuales y comprobar así su validez.

El método deductivo estará dirigido a adquirir un conocimiento más amplio de cómo se manejan los procesos de compra y venta dentro de la empresa. Se analizará paso a paso cada tema, determinando los objetivos que se vayan a plantear.

El proceso de facturación e inventario se lo realiza en forma manual, motivo por el cual se comete una gran cantidad de errores y por tanto una mala atención a los clientes, además debido a que la empresa ha crecido rápidamente se genera una gran cantidad de información, la misma que no cuenta con ningún tipo de respaldo ni de seguridad, razón por la cual todos estos procesos serán automatizados.

3.1.2. Método Inductivo

La inducción va de lo particular a lo general. Empleamos el método inductivo



cuando de la observación de los hechos particulares obtenemos proposiciones generales, es decir, es aquél que establece un principio general una vez realizado el estudio y análisis de hechos y fenómenos en particular.

El método inductivo se basará en preguntar al cliente qué es lo que más le molesta al momento de ser atendido, para conocer en que está fallando la empresa, ya que se puede estar produciendo demoras o errores innecesarios que pueden ser corregidos a tiempo.

Se tomará nota de todos los aspectos que no se los esté haciendo de la manera adecuada y en qué lugar se forma un cuello de botella al momento de atender al cliente y despacharlo rápido luego de recibir su factura.

Debido a que los precios de los materiales cambian constantemente, esto hace que el vendedor cometa errores a la hora de emitir la factura, ya que no coloca el precio real del material y por ende se tiene que repetir una nueva factura, lo que causa molestias tanto al cliente como al vendedor, además que es pérdida para la empresa. La automatización de estos procesos evitará que se cometa este tipo de errores, debido a que el gerente será el encargado de cambiar el precio de los materiales cuando él considere necesario, ya que estos se cargarán automáticamente y de esta manera tanto el cliente como el vendedor se sentirán satisfechos.

3.2. Tipos de Investigación

Para alcanzar los objetivos planteados y generar un prototipo del Sistema de Facturación y Control de Inventarios que entregue una información veraz y a tiempo, la investigación estará enmarcada con los siguientes tipos de investigación:

3.2.1. Investigación Documental



Para la respectiva creación del prototipo, la empresa comercializadora de metales FFHV, nos proporciona la documentación necesaria con datos reales de los procesos que día a día se manejan dentro de cada bodega.

3.2.2. Investigación Histórica

Actualmente todos los procesos que se realizan dentro de la empresa se los hace manualmente, motivo por el cual no hay un manejo controlado y ordenado de la información, además de no tener ningún tipo de respaldo ni seguridad de la misma, lo que la empresa desea es en un futuro no muy lejano automatizar los procesos de compra y venta, cierre de caja, inventarios, reportes, stock de material entre otros y así brindar una mejor atención a sus clientes.

3.3. Técnicas de Recolección de la Información

Para la creación del prototipo se utilizó las siguientes técnicas para recolectar la información:

3.3.1. Observación

Para entender de mejor manera cómo funcionan los procesos de facturación y control de inventarios en la Empresa FFHV, se ha visto la necesidad de visitar el local matriz y varias sucursales, los mismos que se detallan a continuación:

- **Compra de Materiales:**

- El cliente ingresa a la bodega a vender el material o los materiales.
- La persona encargada de recibir el material, realiza el respectivo pesaje en caso de ser algunos materiales, primero los clasifica y luego se pesan de acuerdo al tipo de material. Cabe recalcar que por políticas de la empresa el pesaje solo se lo hace en libras, en caso de que hubieran onzas se procede a redondear los valores. Además el precio de los materiales los



INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

pone el gerente de la empresa, el mismo que se basa en la competencia que existe en el mercado, por esta razón los precios de los materiales pueden estar variando constantemente incluso en un mismo día puede variar algunas veces.

- Otro empleado se encarga de anotar los datos, simplemente toma nota del nombre, peso y precio por cada material a ser comprado, luego multiplica la cantidad por el valor unitario con el cual obtiene un valor total, en caso de ser varios materiales se sigue el mismo proceso y se hace una suma de todos los valores totales para obtener así un subtotal, que es el precio que se le tiene que pagar al proveedor.
- Este proceso se repite para todos los proveedores que llegan a vender el material, hay ocasiones en que se tienen clientes especiales (Estos se clasifican de acuerdo a la cantidad de material que van a vender o que constantemente acuden a cualquier bodega de la empresa a vender), por tanto el gerente da la orden de que se le compre el material o los materiales a un mayor precio, el proceso es el mismo solo cambia el valor del producto.
- Al finalizar el día el administrador de bodega, recoge todas las facturas que se han realizado en ese día y procede a sumar los pesos de todos los materiales clasificándolos de acuerdo al nombre, para así obtener un total en peso por cada material.
- Además suma todo el dinero que se ha invertido en la compra de materiales.
- Todos estos datos son anotados en una hoja de cuaderno.
- Todos estos pasos se lo realiza en todas las bodegas sin excepción.
- Al finalizar el mes y solo en la bodega matriz se procede a ingresar todos los datos obtenidos en el transcurso de ese periodo en una hoja de Excel ya que es la única bodega que cuenta con un computador.

- **Devolución por Compras:**



INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

En caso de que un proveedor no esté de acuerdo con lo que se le pago por su venta simplemente se le devuelve el material con el peso respectivo, luego nos entrega el dinero que se le dio por la compra del mismo y se borran los datos de la hoja con un corrector.

- **Venta de Materiales:**

- El cliente se contacta ya sea con el gerente general o el administrador de cualquier bodega para solicitar una cierta cantidad de uno o varios materiales.
- Se procede a verificar si la cantidad y el material solicitado existe en la bodega donde se hace el pedido, puede darse que no se tenga el material o que simplemente falte una cierta cantidad para completar el pedido, en ese caso se contactan con las otras bodegas para solicitar la cantidad o el material que hace falta, si existe tanto la cantidad como el material se procede a realizar la venta respectiva, en caso de que el administrador de bodega sea el que hace la venta, tiene que ser con la debida autorización del gerente caso contrario no se realiza la venta.
- El administrador de bodega o el gerente toma nota de los datos en una factura de venta, en la cabecera se anota: nombre del cliente, dirección, RUC, ciudad y la fecha, en el detalle el peso y descripción por cada material, luego se multiplica la cantidad por el valor unitario para tener el valor total, se suman todos estos y se obtiene un subtotal que después se multiplica por el IVA para así dar un total que es el valor que el cliente tiene que cancelar.
- Los clientes especiales son aquellos que constantemente compran el material a la empresa, por esta razón tienen un descuento adicional, lo que se hace es cobrarle por menos libras de la solicitada, dándole así un menor valor a pagar pero obviamente al cliente si se le hace llegar la cantidad de material que solicitó.



INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

- El cliente llega a un acuerdo con el gerente si el dinero le paga directamente en efectivo en la bodega que se va a realizar la venta o lo deposita en una cuenta bancaria. Luego de que se haya cancelado el dinero o se verifique que se ha depositado la cantidad establecida en la cuenta, se procede hacer la entrega del material o los materiales.
 - Para esto el cliente llega a la bodega que se le haya establecido, luego se procede a pesar el material para que el cliente verifique que se le entrega tanto la cantidad como el material solicitado.
 - Luego de esto el cliente recibe su factura de venta como constancia de que se le ha entregado correctamente su pedido mientras que el gerente o el administrador se queda con la copia para tener respaldos de la información.
 - Al finalizar el día el administrador de bodega, recoge todas las facturas de venta que se han realizado en ese transcurso y procede a sumar los pesos de todos los materiales que se han vendido clasificándolos de acuerdo al nombre, para así obtener un total en peso por cada uno.
 - Además suma solamente el dinero que ha ingresado a bodega por las ventas.
 - Todos estos datos son anotados en una hoja de cuaderno.
 - Se sigue el mismo procedimiento en todas las bodegas sin excepción.
 - Al finalizar el mes y solo en la bodega matriz, se procede a ingresar toda la información obtenida en el transcurso de ese periodo en una hoja de Excel, ya que es la única bodega que cuenta con un computador.
- **Caja:**
 - Se procede a verificar que la cantidad de dinero que ha quedado en ese día coincida con el dinero que se ha realizado por las ventas menos las compras de material.
 - Estos datos se los almacena en una hoja de cuaderno.



INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

- El dinero que sobra se entrega al día siguiente a la persona encargada de realizar la compra de los materiales.
- **Inventario de Material:**
 - Al finalizar el mes o cuando el gerente lo requiera se procede hacer un inventario de materiales para verificar el peso real existente por cada material en todas las bodegas.
 - Para esto se cuenta con un peso inicial que se tenía del mes anterior.
 - Entre los ingresos están, el peso inicial, las compras y recepción de material realizadas en el transcurso de ese mes
 - Luego se tiene los egresos, que son las ventas y entrega de material.
 - Las compras menos las ventas dan un sobrante total en pesos por cada material.
 - En cada bodega se pesa todo el material que ha quedado en el transcurso de ese mes, y se obtiene el pesaje real por cada material existente en bodega.
 - Los valores positivos en el sobrante del material es pérdida para la empresa por tanto se convierten en valores negativos y se suma con el peso real de cada material.
 - Los valores negativos es ganancia para la empresa por tanto se convierten en valores positivos para la empresa y se suman con el peso real de cada material.
 - Estos son los valores reales de los pesajes de cada material que se coloca en el inicial del siguiente inventario.
 - Se escribe la fecha en la que se hace el inventario y el nombre de la persona que lo realiza.
 - Solamente en la bodega matriz se ingresan estos datos en una hoja de Excel.



INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

- El gerente es el encargado de reunir los informes de todas las bodegas y hacer una suma total tanto del ingreso como egreso de material.

3.3.2. Entrevista

Primero me comuniqué con el gerente y propietario de la empresa, el Sr. Freddy Haro, con el cual llegamos al acuerdo de reunirnos en la bodega matriz para realizar las respectivas entrevistas.

Me dirigí a la bodega principal ubicada en la Av. 24 de Mayo en el Centro histórico de Quito, donde el gerente me atendió muy amablemente, le realice algunas preguntas las cuales se detallan a continuación con sus respectivas respuesta y análisis de cada una.

Pregunta 1: ¿A qué se dedica la empresa?

Respuesta: FFHV es una empresa comercializadora de metales. Sus principales ingresos los obtiene en la venta del material a las empresas mayoristas.

Análisis: Al ser su principal fuente de ingresos las ventas, lo más conveniente y oportuno para la optimización de los procesos es necesario la elaboración de un sistema de facturación y control de inventarios.

Pregunta 2: ¿Estaría la empresa dispuesta a colaborar para la implementación de un software?

Respuesta: Nosotros nos sentimos gustosos de apoyar en este proyecto y estamos dispuestos a proporcionar todo tipo de información que sea necesaria.

Análisis: La empresa está totalmente de acuerdo en la implementación del



software por lo que no habrá ningún inconveniente en el desarrollo del mismo.

Pregunta 3: ¿Actualmente en la empresa se cuenta con un recurso informático para la implantación de un software?

Respuesta: Si, tenemos tres computadoras, una en el área administrativa y dos en el área operativa.

Análisis: Por el momento se puede realizar la implementación del prototipo, ya que por tratarse de un modulo no tan extenso se puede instalar en cualquier computador personal, pero es necesario que lo más pronto posible se compre un servidor exclusivo para el manejo de la información de la empresa.

Pregunta 4: ¿Será posible la adquisición de un servidor para el manejo de la información?

Respuesta: La información que se maneja en la empresa es de suma importancia, por lo que no habría ningún problema en comprar un servidor lo más pronto posible.

Análisis: Con la adquisición de un servidor se garantiza que la información se mantenga segura ya que se puede configurar el servidor para que tenga varios discos duros y tener un doble respaldo de la información, lo que no es posible con un computador normal.

Pregunta 5 ¿Cree usted que los empleados se sentirán satisfechos con la implantación del software?

Respuesta: Pienso que si, pues ya no tendrán que hacer los procesos de forma manual, aunque la mayoría de los trabajadores no tienen conocimientos de



INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

computación por lo que creo que se les hará un poco difícil adaptarse a la nueva forma de trabajo.

Análisis: Los usuarios están dispuestos a colaborar, por lo que dándoles la capacitación necesaria para el uso del software, se irán adaptando y observando todas la ventajas que les brindará el mismo

Pregunta 6: ¿Cómo se llevan los procesos dentro de la empresa?

Respuesta: Primeramente la empresa compra diariamente el material a pequeños proveedores, se va clasificando el material comprado para después venderlo a grandes empresas.

Análisis: Se estudiara a profundidad cada proceso para separarlo en módulos y realizar un sistema de acuerdo a las necesidades de la empresa.

Pregunta 7: ¿Cómo se controla el stock de los materiales?

Respuesta: Al finalizar el mes se procede a pesar el material que se ha comprado en ese transcurso de tiempo y luego se ingresa esta información en una hoja de Excel.

Análisis: Es necesario que la empresa controle el stock de materiales diariamente, pues en caso de que un cliente solicite algún pedido, se le proporcione la información rápidamente, con la implementación del software esto será posible

Pregunta 7: ¿Dónde se guarda la información que diariamente genera la empresa?

Respuesta: La información se la registra en un cuaderno y al finalizar el mes se la ingresa en una hoja de Excel.



Análisis: Es necesario que la información esté totalmente respaldada, con la implementación del software se evitará la acumulación de documentos físicos y se tendrá la información de una manera ordenada.

Pregunta 8: ¿Se controla que la información sea totalmente confidencial y que personas ajenas no tengan acceso a ella?

Respuesta: El único control que se tiene es el inicio de sesión de la máquina y para tener acceso a los documentos de Excel se ingresa por medio de una clave.

Análisis: La información puede estar siendo respaldada pero en caso de tener acceso a ella se puede estar haciendo alteraciones en los documentos, lo que no ocurriría con el sistema a implantar, ya que para realizar algún cambio, el usuario tendría que ingresar a la base de datos lo que le resultará completamente difícil, por lo que se garantiza que la información estará completamente segura.

Pregunta 9: ¿Qué tiempo aproximadamente se demora el vendedor en atender el pedido al cliente?

Respuesta: Dependiendo de la cantidad de material que solicite el cliente, se procede a verificar existencia y luego se le hace la factura de venta, este proceso dura alrededor de unos 15 a 20 minutos .por cliente

Análisis: Con el uso del software se agilizarán tanto los procesos de venta como de compra ya que se le brindará al cliente información real, emitiéndole la factura rápidamente y por ende habrá un rotación mayor de clientes.



INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

Pregunta 10: ¿La empresa cuenta con varias bodegas, estaría dispuesto a adquirir equipos para cada una de ellas y que en todas las bodegas se manejen los procesos de forma automatizada?

Respuesta: Si, el objetivo es tener en cada bodega por lo menos un computador para que la información sea almacenada en el nuevo sistema.

Análisis: Al automatizar los procesos en todas las bodegas se puede tener un control total de la información que se maneja en la empresa, de esta manera se le puede dar mejor información al cliente, en caso de que un material no exista en una bodega infórmale rápidamente que puede acudir a otra sucursal.



CAPITULO IV

DESARROLLO DE LA PROPUESTA

4.1 Diagnóstico Situacional

Actualmente la empresa se encuentra dividida en cinco áreas principales como son: Gerencia General, Administración, Contabilidad, Caja y Operativo. Para el respectivo desarrollo del proyecto, se trabajará en todas estas áreas, donde se ha podido determinar que la empresa lleva actualmente la facturación y control de inventarios en forma manual por lo que se comete una gran cantidad de errores, además no se tiene ningún tipo de respaldo de la información, ni seguridad de la misma.

La empresa si cuenta con un recurso informático tanto en el área operativa como administrativa, no se han encontrado problemas de software y hardware existente, por lo que no habrá ningún inconveniente a la hora de implantar el software. El personal consultado ha mostrado un interés en su creación y piensan aportar con ideas para su estructura desde el punto de vista de usuario.

4.2 Estructura Organizacional

La estructura organizacional se refiere a la forma en que se dividen, agrupan y coordinan las actividades de la organización en cuanto a las relaciones entre los gerentes y los empleados, entre gerentes y gerentes y entre empleados y empleados. Los departamentos de una organización se pueden estructura formalmente en tres formas básicas: por función, por producto/mercado o en forma de matriz.

La Estructura Organizacional de la empresa FFHV actualmente tiene definidos puestos de trabajo, pero no cuenta en todas las áreas con los equipos necesarios para la implantación del software



4.2.1 Organigrama Estructural

Un diagrama estructural es aquel que está representado jerárquicamente, es decir por el cargo que ocupan en la compañía. En base a estudios realizados en la empresa se ha visto la necesidad de implementar el siguiente organigrama estructural. (VER ANEXO N°1 ↻)

4.2.2 Orgánico Posicional

El orgánico posicional, recoge los nombres de las personas que pertenecen a cada uno de los niveles estructurales y funcionales.

La empresa FFHV cuenta en forma implícita con cierto juego de jerarquías y atribuciones asignadas a los miembros o componentes de la misma. En consecuencia de esto se ha realizado la siguiente estructura posicional (VER ANEXO N°2 ↻)

4.2.3 Orgánico Funcional

Un orgánico funcional es aquel que está determinado según las funciones que realizan cada persona en la Empresa. La empresa se basa en el principio de la especialización y en la ruptura de la unidad de mando en la base operativa, es así que se ha decidido realizar la siguiente estructura funcional (VER ANEXO N°3 ↻)

4.3 Infraestructura Informática

Actualmente la empresa cuenta con un recurso informático tanto en área administrativa como en el área operativa, por lo que no existirá ningún inconveniente al momento de implantar el prototipo, a continuación detallaremos las características de cada uno:



4.3.1 Hardware

| ÁREA | EQUIPOS | HARDWARE |
|----------------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Administrativa | 1 | <ul style="list-style-type: none">• Computador Intel Dual Core• Procesador de 2.8 Ghz• Memoria 1 Gb• Monitor LC19• Cable Puerto serial Impresora• Teclado• Mouse |
| Operativa | 2 | <ul style="list-style-type: none">• 2 Computadores Intel Pentium IV• Procesador de 2.5 Ghz• Memora 512 Mb• Cable Puerto serial Impresora• Teclado - Mouse• Monitor LG 14" |

HARDWARE EXISTENTE ÁREA ADMINISTRATIVA Y OPERATIVA

TABLA N° 4.3.1

FUENTE: Información Entregada por la empresa

4.3.2 Software

A continuación se detalla el software que se encuentra instalado en lo equipos que se describen en la parte 4.3.1.

| AREA | EQUIPOS | SOFTWARE | LICE NCIA |
|----------------|---------|--------------------------------------------------------------------------------------------------------|--------------|
| Administrativa | 1 | <ul style="list-style-type: none">• Windows Xp Professional• Power Designer 6.0 | Si No |



| | | | |
|------------------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| | | <ul style="list-style-type: none">• MySQL 5.0• Microsoft Office• Antivirus• Internet Security | No Si No No |
| Operativa | 2 | <ul style="list-style-type: none">• Windows Xp Professional• Microsoft Office• Antivirus• Internet Security | Si Si No No |

SOFTWARE EXISTENTE ÁREA ADMINISTRATIVA Y OPERATIVA

TABLA N° 4.3.2

FUENTE: Información Entregada por la empresa

4.3.3 Comunicaciones

La empresa dispone de la red de cableado estructurado, está realizada con cable UTP categoría 5, la Topología de la red es de tipo estrella, Grupo de Trabajo, como “FFHV”, donde se comparte carpetas, archivos e impresoras de todos los usuarios.

Afortunadamente la empresa trabaja con una buena topología, ya que ha diferencia de otras como son: Bus y Ring, es más tolerante esto quiere decir que si una computadora se desconecta o si se le rompe el cable solo esa computadora es afectada y el resto de la red mantiene su comunicación normalmente además es fácil de reconfigurar, añadir o remover una computadora es tan simple como conectar o desconectar el cable

4.3.4 Recurso Humano Técnico

Dado que la empresa no cuenta actualmente con un recurso humano técnico, se ha visto la necesidad de implantar una persona dirigida al campo informático, la



misma que dará el respectivo mantenimiento y control del prototipo para que funcione correctamente y así poder atender las necesidades de la empresa en el momento en que se presenten inconvenientes.

Hay que destacar que el Recurso Humano Técnico, deberá estar dirigido a la parte informática, dará la capacitación a los usuarios y su debido mantenimiento tanto del hardware como software.

4.4 Descripción de Alternativas

Después del estudio realizado en la empresa FFHV, se ha determinado la necesidad de contar con un prototipo que maneje de forma rápida y confiable los procesos. A continuación se plantearán alternativas que se tiene como posibles soluciones a las necesidades de la empresa:

4.4.1 Alternativa 1. Adquisición de un software existente.

Esta es la alternativa que las pequeñas empresas elijen con mayor frecuencia a la hora de encontrarse con la necesidad de contar con un sistema que les ayude en los procesos que realizan, pero esta elección no siempre es la mejor y la más indicada ya que al adquirir un software genérico en el mercado, este no siempre se acopla correctamente a las necesidades de la empresa y es aquí donde se puede apreciar que la eminente solución no es más que un nuevo problema, ya que la empresa será en este caso la que tendrá que acoplarse al sistema adquirido y por ende tendrá que pasar por alto algunos de los procesos que generalmente realice.

4.4.2 Alternativa 2. Contratar una empresa dedicada al desarrollo de software.

Esta alternativa permite a la empresa contar con un software de acuerdo a sus necesidades. Actualmente en el mercado hay muchas empresas que se dedican a



la elaboración e implantación de sistemas, pero el problema radica en el costo-beneficio, ya que se contaría con un buen programa pero a un costo muy alto, es por ello que esta alternativa es considerada en empresas grandes multinacionales e internacionales, pero en medianas o pequeñas se hace difícil contratar sus servicios.

4.4.3 Alternativa 3. Desarrollar un nuevo sistema propio para la empresa.

La persona encargada del diseño, desarrollo e implantación del nuevo sistema, será yo misma, lo que resulta muy conveniente para la empresa ya que estará desarrollado en base a los requerimientos específicos de la misma y no tendrá costo alguno por tratarse de un proyecto de grado.

4.5 Evaluación y Selección de Alternativas

En este punto se realizará una evaluación de las alternativas expuestas anteriormente, para que la empresa FFHV pueda tener una mejor perspectiva de las opciones que se le presenta y escoger la alternativa que mejor llene sus expectativas.

Para esto se basará en tres parámetros fundamentales como son:

- Parámetro Técnico.
- Parámetro Operativo.
- Parámetro Económico.

4.5.1 Evaluación de las alternativas desde el punto de vista Técnico.

| PARÁMETROS DE EVALUACIÓN | ALT. 1 | ALT. 2 | ALT. 3 |
|--------------------------|--------|--------|--------|
|--------------------------|--------|--------|--------|



| | | | | |
|---------------------------|-----------|-----------|-----------|------------|
| Hardware que Utiliza | 1 | 2 | 2 | Para poder |
| Software de Base | 3 | 3 | 3 | |
| Software Utilitario | 1 | 2 | 3 | |
| Software Aplicación | 1 | 2 | 3 | |
| Integración Modular | 1 | 3 | 3 | |
| Documentación Técnica | 1 | 2 | 3 | |
| Documentación del Usuario | 1 | 2 | 3 | |
| Código Fuente | 1 | 1 | 3 | |
| Soporte Técnico | 1 | 3 | 3 | |
| Integridad de Datos | 2 | 3 | 3 | |
| TOTAL | 12 | 23 | 29 | |

determinar la mejor alternativa técnica se ha decidido calificar en escalas del 1 al 3, donde 1 representa malo, 2 bueno y 3 excelente, teniendo como la mejor alternativa la que obtenga el mejor puntaje.

EVALUACIÓN DE PARÁMETROS DESDE EL PUNTO DE VISTA TÉCNICOS.

TABLA N° 4.5.1

FUENTE: Gina Cumbicus



| PARÁMETROS DE EVALUACIÓN | ALT. 1 | ALT. 2 | ALT. 3 |
|-------------------------------|--------|--------|--------|
| Oportunidad de la Información | 2 | 2 | 3 |
| Confiabilidad de Resultados | 1 | 2 | 3 |
| Consultas más rápidas | 1 | 3 | 3 |
| Interfaz del Usuario | 1 | 3 | 3 |

EVALUACIÓN DE PARÁMETROS DESDE EL PUNTO DE VISTA TÉCNICOS.

FIGURA N° 4.5.1

FUENTE: Gina Cumbicus

4.5.2 Evaluación de las alternativas desde el punto de vista Operativo.

De acuerdo a un análisis realizado en la FFHV se ha logrado establecer los siguientes parámetros de evaluación con la misma manera que se califico en el punto 4.5.1.

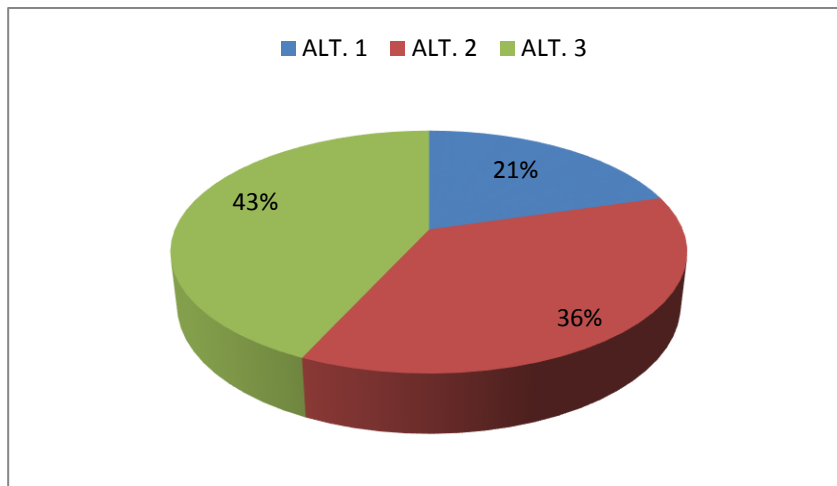


| | | | |
|---------------------------------------|----------|-----------|-----------|
| Minimización de Esfuerzos | 1 | 2 | 2 |
| Menor cantidad de errores | 1 | 2 | 2 |
| Uso del sistema por parte del usuario | 2 | 2 | 3 |
| TOTAL | 9 | 16 | 19 |

EVALUACIÓN DE PARÁMETROS DESDE EL PUNTO DE VISTA OPERATIVO.

TABLA N° 4.5.2

FUENTE: Gina Cumbicus



EVALUACIÓN DE PARÁMETROS DESDE EL PUNTO DE VISTA OPERATIVO.

FIGURA N° 4.5.2

FUENTE: Gina Cumbicus

4.5.3 Evaluación de las alternativas desde el punto de vista Económico.

Para poder detallar los parámetros económicos de la Empresa se ha decidido evaluar de una manera diferente es decir el valor de 3 significa menos costo y el valor 1 significa que es de mayor costo, para poder así mantener el criterio de la mejor alternativa, es la que tenga mayor puntaje.

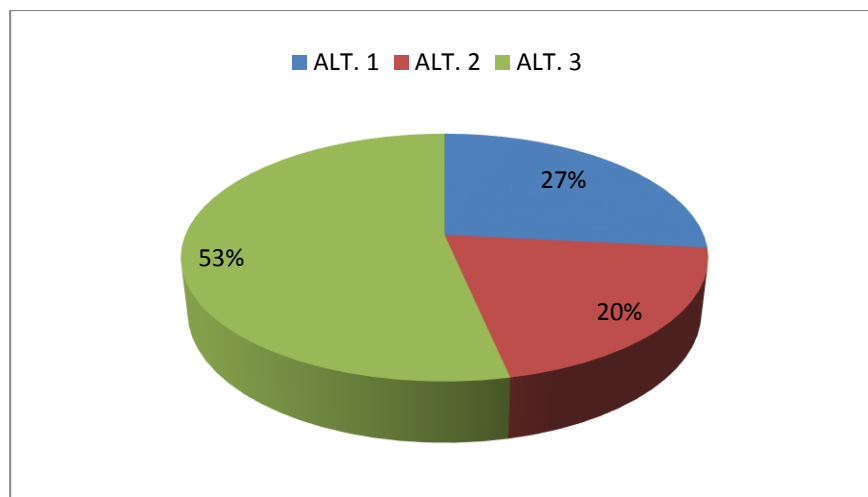


| PARÁMETROS DE EVALUACIÓN | ALT. 1 | ALT. 2 | ALT. 3 |
|---------------------------|----------|----------|-----------|
| Actualización de Hardware | 1 | 1 | 2 |
| Licencias | 1 | 1 | 2 |
| Software de Aplicación | 2 | 1 | 3 |
| Capacitación | 2 | 1 | 3 |
| Código Fuente | 1 | 1 | 3 |
| Mantenimiento | 1 | 1 | 3 |
| TOTAL | 8 | 6 | 16 |

EVALUACIÓN DE PARÁMETROS DESDE EL PUNTO DE VISTA ECONÓMICO.

TABLA N° 4.5.3

FUENTE: Gina Cumbicus



EVALUACIÓN DE PARÁMETROS DESDE EL PUNTO DE VISTA ECONÓMICO.

FIGURA N° 4.5.3

FUENTE: Gina Cumbicus

4.6 Factibilidad Técnica

De acuerdo a un análisis realizado de las tres alternativas desde la factibilidad técnica, operativa y económica se ha realizado los siguientes cuadros de datos comparativos con la finalidad de apreciar gráficamente la obtención de resultados.

| | | | |
|--|--|--|--|
| | | | |
|--|--|--|--|

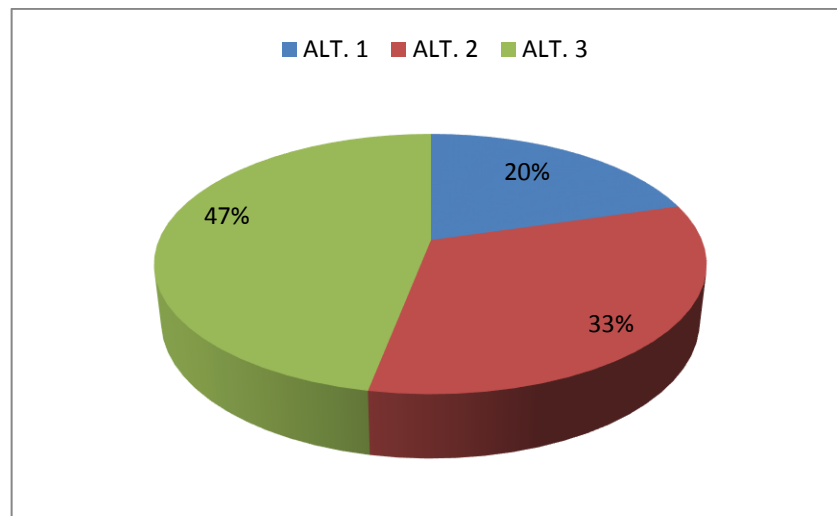


| FACTIBILIDAD | ALT. 1 | ALT. 2 | ALT. 3 |
|--------------|-----------|-----------|-----------|
| TÉCNICA | 12 | 23 | 29 |
| OPERATIVA | 9 | 16 | 19 |
| ECONÓMICA | 8 | 6 | 16 |
| TOTAL | 27 | 43 | 62 |

DETERMINACIÓN DE LA MEJOR ALTERNATIVA

TABLA N° 4.6

FUENTE: Gina Cumbicus



DETERMINACIÓN DE LA MEJOR ALTERNATIVA

FIGURA N° 4.6

FUENTE: Gina Cumbicus

Una vez analizado y expuesto los diferentes parámetros de evaluación respecto al problema se ha llegado a las siguientes conclusiones:

- Se deduce que la tercera opción es la más favorable, ya que al realizar un nuevo sistema y por tratarse de una empresa que no tiene automatizado sus procesos, se inicia desde cero lo que facilita el aprendizaje y adaptabilidad al sistema.
- Estará desarrollado en base a las especificaciones de los requerimientos propios de la empresa. Se podrá dar el respectivo mantenimiento debido a que se tiene código fuente.



- Como está desarrollado por módulos puede irse incrementando de acuerdo a las necesidades y exigencias de la Empresa.
- Resulta conveniente a la empresa inclusive en el sentido económico ya que por tratarse de un proyecto de grado los costos del desarrollo los asume el estudiante, además se utiliza software libre lo que resulta más favorable todavía.
- El software se encuentra desarrollado con tecnología de punta en uno de los lenguajes de programación orientado a objetos, más competitivos del mercado.
- Al ser desarrollado con criterios técnicos y profesionales, de una persona capacitada, se garantiza que funcione de la mejor manera posible, y se hagan mejoras.
- Muy amigable y de fácil utilización ya que se utilizan herramientas adecuadas para diseñar cada una de las partes.
- Cien por ciento participaciones de los usuarios en el desarrollo del prototipo.

4.7 Descripción de Procesos.

Los procesos detallados a continuación corresponden a la información relacionada con el prototipo del sistema.

Cada proceso detalla el manejo de información en la organización y sigue las diferentes secuencias de cada actividad, para la descripción de cada uno de ellos, se utilizarán diagramas para esto se ha visto la necesidad de utilizar la herramienta Buinessness Proccess Model (BMP) y casos de uso.



Acceso al Sistema: Este proceso permite que los usuarios ingresen al prototipo del sistema con su respectivo nombre de usuario y contraseña, asignándole el cargo que ocupa cada uno (proceso mantenimiento tipo de empleado) de esta manera se puede controlar que las personas que utilizan el software tengan acceso solamente a la información que el gerente considere necesario, esta consulta se la realiza en la tabla USUARIO de la base de datos BDFACTURACIONINVENTARIO. (VER ANEXO N°4 ↻)

Crear Cuenta de Usuario: Este proceso permite crear una cuenta a cada empleado que trabaje en la empresa, para que pueda ingresar a la aplicación, encriptando la contraseña lo que brindará mayor seguridad de los datos, dicha información está contenida en la tabla USUARIO de la base de datos BDFACTURACIONINVENTARIO. (VER ANEXO N°5 ↻)

Mantenimiento de Clientes: Este proceso sirve para ingresar, actualizar o eliminar un cliente, el botón actualizar solamente estará habilitado cuando el objeto se encuentre creado, dicha información está contenida en la tabla CLIENTE de la base de datos BDFACTURACIONINVENTARIO. (VER ANEXO N°6 ↻)

Mantenimiento de Proveedores: Este proceso sirve para ingresar, actualizar o eliminar un proveedor, el botón actualizar solamente estará habilitado cuando el objeto se encuentre creado dicha información está contenida en la tabla PROVEEDOR de la base de datos BDFACTURACIONINVENTARIO. (VER ANEXO N°7 ↻)

Mantenimiento de Materiales: Este proceso sirve para ingresar, actualizar o eliminar los materiales que se compran en la empresa, el botón actualizar solamente estará habilitado cuando el objeto se encuentre creado dicha información está contenida en la tabla MATERIAL de la base de datos BDFACTURACIONINVENTARIO. (VER ANEXO N°8 ↻)

Mantenimiento de Bodegas: Este proceso sirve para ingresar, actualizar o eliminar todas las bodegas con las que cuenta la empresa, además permite buscar las bodegas que se encuentran activas o inactivas, dicha información está contenida en la tabla BODEGA de



la base de datos BDFACTURACIONINVENTARIO. (VER ANEXO N°9 ↻)

Mantenimiento de Tipo de Empleados: Este proceso permite ingresar, actualizar o eliminar el cargo que ocupará cada empleado, el botón actualizar solamente estará habilitado cuando el objeto se encuentre creado, dicha información está contenida en la tabla TIPOEMPLEADO de la base de datos BDFACTURACIONINVENTARIO. (VER ANEXO N°10 ↻)

Mantenimiento de Empleados: Este proceso permite ingresar, actualizar o eliminar los empleados que trabajan en la empresa, el botón actualizar solamente estará habilitado cuando el objeto se encuentre creado además que se puede seleccionar el cargo que ocupa cada uno de ellos(proceso mantenimiento TIPOEMPLEADO), dicha información está contenida en la tabla EMPLEADO de la base de datos BDFACTURACIONINVENTARIO. (VER ANEXO N°11 ↻)

Compras: Este proceso permite al usuario crear la factura de compra, para lo cual se selecciona el proveedor en caso de existir o se procede a ingresar un nuevo proveedor en caso de no haberlo(proceso mantenimiento proveedor), luego escogemos el o los materiales a comprar ingresando el peso y el precio por cada uno, si el material no existe se procede q ingresar un nuevo material(proceso mantenimiento MATERIAL) el prototipo calcula automáticamente el valor total a pagar, cabe recalcar que el pago se lo hace en efectivo, la información del proveedor con el número de factura está contenida en la tabla CABECERAFACTURACOMPRA y los detalles se guardaran en la tabla DETALLEFACTURACOMPRA de la base de datos BDFACTURACIONINVENTARIO. (VER ANEXO N°12 ↻)

Ventas: Este proceso permite al usuario crear la factura de venta, para lo cual se selecciona el nombre del cliente en caso de existir o se procede a ingresar un nuevo cliente en caso de no haberlo(proceso mantenimiento cliente), luego se selecciona el o los materiales a ser vendidos, en caso de existir en bodega, automáticamente se cargará



INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

el precio de cada material, para lo cual solo se debe ingresar el peso del material, el prototipo calcula el valor total a pagar, el cliente realiza el pago en efectivo e inmediatamente se imprime la factura de venta, la información del cliente con el número de factura está contenida en la tabla CABECERAFACTURAVENTA y los detalles se guardaran en la tabla DETALLEFACTURAVENTA de la base de datos BDFACTURACIONINVENTARIO. (VER ANEXO N°13 ↻).

Devolución de Ventas: En caso de que algún cliente desee devolver material, se debe entregar la factura de venta con el estado en que se encuentre en ese momento, para hacer la entrega de un nuevo material o si fuera el caso de que el peso entregado no fue el correcto, reponer el faltante. (VER ANEXO N°14 ↻).

Devolución de Compras: En el momento en que el material ingresa a cualquiera de las bodegas se procede a revisar inmediatamente que el material comprado sea el correcto, en caso de encontrarse alguna anomalía, en ese instante es entregado al proveedor para su cambio inmediato o de no tener el material en ese momento no se realiza la compra. (VER ANEXO N°15 ↻).

Cierre de Caja: Como rutina de la empresa al finalizar el día se realiza el respectivo cierre de caja, la persona encargada de realizar dicho proceso, verifica que el dinero existente en caja coincida con la cantidad que muestra el prototipo. Dicha información será almacenada en la tabla CAJA de la base de datos BDFACTURACIONINVENTARIO. (VER ANEXO N°16 ↻).

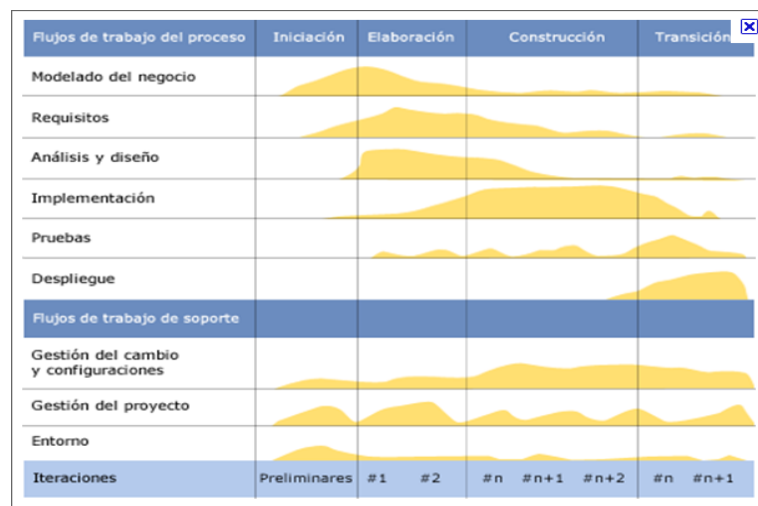
Inventario: Este proceso permite tener un control tanto del peso como del precio de cada material que se ha comprado, vendido o devuelto con la fecha respectiva en que se realizo dicha acción, además permite cerrar el inventario en el momento que sea necesario o que el gerente lo requiera. El número del inventario, la fecha inicial y final así como el peso inicial y final de cada material está contenido en la tabla INVENTARIOFISICO y los detalles por cada compra o venta realizada están almacenados en la tabla



DETALLE INVENTARIO FÍSICO de la base de datos BDFACTURACION INVENTARIO.

4.8 Descripción de La Metodología de Desarrollo

Para el desarrollo de proyecto se utilizará la metodología RUP (Rational Unified Process en inglés) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Esta metodología comprende cuatro fases como son: inicio, elaboración, construcción y transición. Gráficamente se puede apreciar de una mejor manera las fases de la metodología RUP:



METODOLOGÍA RUP

FIGURA N° 4.8

FUENTE: Internet

Para el desarrollo del prototipo analizaremos cada una de ella, las mismas que se detallan a continuación:

Fase de Inicio:

(Inspección y Concepción) realizamos un plan de fases donde identificamos los principales casos de uso concretando nuestra idea, como está enmarcado el negocio y el alcance del proyecto.



Fase de Elaboración:

Planificamos las actividades necesarias y los recursos requeridos, especificando las características y el diseño del proyecto.


Fase de Construcción:

En esta fase se va elaborando la aplicación y los manuales de usuario, realizamos todas las pruebas necesarias, simulando el trabajo de la empresa con el fin de dejar el prototipo del sistema listo para la siguiente fase.


Fase de Transición:

Se procede a realizar la instalación y la debida capacitación a los usuarios, con el fin de que el prototipo se encuentre listo para ser usado. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto

4.9 Modelo Conceptual

Para poder establecer el modelo conceptual de datos se presentará de forma gráfica el diseño de la base de datos elaborado en Power Designer, en donde se encuentran las diferentes entidades y atributos propios de cada una de ellas, además se establecerá las llaves primarias de cada tabla y las relaciones entre las mismas. (VER ANEXO N°17 ).

4.10 Modelo Físico

En el siguiente modelo físico de datos se observan las claves primarias, claves secundarios o foráneas, el tipo de datos que tiene cada atributo y el tipo de relaciones que constan en el diseño, el mismo que sirvió para generar la base de datos en MySQL. (VER ANEXO N°18 )

4.11 Diccionario de Datos



En el diccionario de datos se mostrará el nombre y el código de los elementos que intervienen en el diseño de la base de datos en donde se podrá constar la aplicación de los estándares de programación utilizados.

Listado de tablas

| NOMBRE | | CODIGO | |
|-------------------------|-------------------------|-------------------------|-------------------------|
| AUDITORIA | AUDITORIA | AUDITORIA | AUDITORIA |
| BODEGA | BODEGA | BODEGA | BODEGA |
| BODEGAMATERIAL | BODEGAMATERIAL | BODEGAMATERIAL | BODEGAMATERIAL |
| CABECERAFATURACOMPRA | CABECERAFATURACOMPRA | CABECERAFATURACOMPRA | CABECERAFATURACOMPRA |
| CABECERAFATURAVENTA | CABECERAFATURAVENTA | CABECERAFATURAVENTA | CABECERAFATURAVENTA |
| CIERRECAJA | CIERRECAJA | CIERRECAJA | CIERRECAJA |
| CLIENTE | CLIENTE | CLIENTE | CLIENTE |
| DETALLEFACTURAVENTA | DETALLEFACTURAVENTA | DETALLEFACTURAVENTA | DETALLEFACTURAVENTA |
| DETALLEFATURACOMPRA | DETALLEFATURACOMPRA | DETALLEFATURACOMPRA | DETALLEFATURACOMPRA |
| DETALLEINVENTARIOFISICO | DETALLEINVENTARIOFISICO | DETALLEINVENTARIOFISICO | DETALLEINVENTARIOFISICO |
| EMPLEADO | EMPLEADO | EMPLEADO | EMPLEADO |
| INVENTARIOFISICO | INVENTARIOFISICO | INVENTARIOFISICO | INVENTARIOFISICO |
| MATERIAL | MATERIAL | MATERIAL | MATERIAL |
| PROVEEDOR | PROVEEDOR | PROVEEDOR | PROVEEDOR |
| TIPOEMPLEADO | TIPOEMPLEADO | TIPOEMPLEADO | TIPOEMPLEADO |

Listado de relaciones

| NOMBRE | CODIGO DE REFERENCIA | ENTIDAD 1 | ENTIDAD 2 |
|------------------|----------------------|--------------|----------------------|
| bodega_bodmat | BODEGA_BODMAT | BODEGA | BODEGAMATERIAL |
| bodega_cabfaccom | BODEGA_CABFACCOM | BODEGA | CABECERAFATURACOMPRA |
| bodega_cabfacven | BODEGA_CABFACVEN | BODEGA | CABECERAFATURAVENTA |
| bodmat_detfacven | BODMAT_DETFACVEN | BODEGAMATERI | DETALLEFACTURAV |



| NOMBRE | CODIGO DE REFERENCIA | ENTIDAD 1 | ENTIDAD 2 |
|---------------------|----------------------|--------------------------|----------------------------|
| bodmat_detinvfis | BODMAT_DETINVFI | AL BODEGAMATERI | ENTA DETALLEINVENTAR |
| CabFacCom_DetFacCom | CABFACCOM_DETFA | AL CABECERAFAC | IOFISICO DETALLEFATURAC |
| cabfacven_detfacven | CABFACVEN_DETFA | URACOMPRA CABECERAFAC | OMPRA DETALLEFACTURAV |
| cliente_cabfacven | CLIENTE_CABFAC | URAVENTA CLIENTE | ENTA CABECERAFAC |
| detfaccom_bodmat | DETFAACCOM_BOD | COMPRA DETALLEFATURA | COMPRA BODEGAMATERIAL |
| Emp_CanFacCom | EMP_CANFACCOM | EMPLEADO | CABECERAFAC |
| empleado_cabfacven | EMPLEADO_CABFA | EMPLEADO | COMPRA CABECERAFAC |
| empleado_ciecaj | EMPLEADO_CIECA | EMPLEADO | URAVENTA CIERRECAJA |

Listado de atributos

| Nombre | Código | Tipo |
|----------------------|----------------------|-----------------------------|
| Aju_Codigo | AJU_CODIGO | Integer |
| Aju_Egreso | AJU_EGRESO | Float (10) |
| Aju_Estado | AJU_ESTADO | Boolean |
| Aju_Fecha | AJU_FECHA | Date |
| Aju_NumCompania | AJU_NUMCOMPANIA | Integer |
| Aju_Tipo | AJU_TIPO | Variable characters (20) |
| Aud_CodEmpleado | AUD_CODEMPLEADO | Integer |
| Aud_Codigo | AUD_CODIGO | Integer |
| Aud_Estado | AUD_ESTADO | Boolean |
| Aud_Fecha | AUD_FECHA | Date |
| Aud_Hora | AUD_HORA | Time |
| Aud_NumCompania | AUD_NUMCOMPANIA | Integer |
| Aud_NumTransferencia | AUD_NUMTRANSFERENCIA | Integer |
| Aud_TipTransferencia | AUD_TIPTRANSFERENCIA | Variable characters (30) |
| Bma_Cantidad | BMA_CANTIDAD | Float (10) |
| Bma_Estado | BMA_ESTADO | Boolean |
| Bma_Fecha | BMA_FECHA | Date |
| Bma_Secuencial | BMA_SECUENCIAL | Integer |
| Bod_Estado | BOD_ESTADO | Boolean |
| Bod_Nombre | BOD_NOMBRE | Variable |



| Nombre | Código | Tipo |
|-------------------|-------------------|-----------------|
| Bod_NumCompania | BOD_NUMCOMPANIA | characters (20) |
| Bod_Numero | BOD_NUMERO | Integer |
| Bod_Ubicacion | BOD_UBICACION | Integer |
| | | Variable |
| | | characters (30) |
| Cca_Codigo | CCA_CODIGO | Integer |
| Cca_EgresoDinero | CCA_EGRESODINERO | Float (10) |
| Cca_Estado | CCA_ESTADO | Boolean |
| Cca_Fecha | CCA_FECHA | Date |
| Cca_IngresoDinero | CCA_INGRESODINERO | Float (10) |
| Cca_NumCompania | CCA_NUMCOMPANIA | Integer |
| Cca_ValorFinal | CCA_VALORFINAL | Float (10) |
| Cca_ValorInicial | CCA_VALORINICIAL | Float (10) |
| Cfc_Codigo | CFC_CODIGO | Integer |
| Cfc_Descuento | CFC_DESCUENTO | Float (10) |
| Cfc_Estado | CFC_ESTADO | Boolean |
| Cfc_FechaEmision | CFC_FECHAEMISION | Date |
| Cfc_ID | CFC_ID | Integer |
| Cfc_Iva | CFC_IVA | Float (10) |
| Cfc_NumCfc | CFC_NUMCFC | Variable |
| | | characters (10) |
| Cfc_NumCompania | CFC_NUMCOMPANIA | Integer |
| Cfc_Subtotal | CFC_SUBTOTAL | Float (10) |
| Cfc_Total | CFC_TOTAL | Float (10) |
| Cfv_Descuento | CFV_DESCUENTO | Float (10) |
| Cfv_Estado | CFV_ESTADO | Boolean |
| Cfv_FechaEmision | CFV_FECHAEMISION | Date |
| CFV_ID | CFV_ID | Integer |
| Cfv_Iva | CFV_IVA | Float (6) |
| Cfv_NumCfv | CFV_NUMCFV | Integer |
| Cfv_NumCompania | CFV_NUMCOMPANIA | Integer |
| Cfv_Subtotal | CFV_SUBTOTAL | Float (10) |
| Cfv_Total | CFV_TOTAL | Float (10) |
| Cli_Apellido | CLI_APELLIDO | Variable |
| | | characters (30) |
| Cli_CedRuc | CLI_CEDRUC | Variable |
| | | characters (30) |
| Cli_Celular | CLI_CELULAR | Variable |
| | | characters (30) |
| Cli_Ciudad | CLI_CIUDAD | Variable |
| | | characters (20) |
| Cli_Descuento | CLI_DESCUENTO | Boolean |
| Cli_Direccion | CLI_DIRECCION | Variable |
| | | characters (30) |



| Nombre | Código | Tipo |
|------------|------------|--------------------------|
| Cli_Email | CLI_EMAIL | Variable characters (30) |
| Cli_Estado | CLI_ESTADO | Boolean |

Listado de identificadores

4.12 Estándares de Programación

| NOMBRE | TABLA | CODIGO |
|-------------------------|-------------------------|----------------|
| AUDITORIA | AUDITORIA | AUD_CODIGO |
| BODEGA | BODEGA | BOD_NUMERO |
| BODEGAMATERIAL | BODEGAMATERIAL | BMA_SECUENCIAL |
| CABECERAFATURACOMPRA | CABECERAFATURACOMPRA | CFC_ID |
| CABECERAFATURAVENTA | CABECERAFATURAVENTA | CFV_ID |
| CIERRECAJA | CIERRECAJA | CCA_CODIGO |
| CLIENTE | CLIENTE | CLI_CEDRUC |
| DETALLEFACTURAVENTA | DETALLEFACTURAVENTA | DFV_SECUENCIAL |
| DETALLEFATURACOMPRA | DETALLEFATURACOMPRA | DFC_SECUENCIAL |
| DETALLEINVENTARIOFISICO | DETALLEINVENTARIOFISICO | DIF_SECUENCIAL |
| EMPLEADO | EMPLEADO | EMP_CEDULA |
| INVENTARIOFISICO | INVENTARIOFISICO | IFI_NUMINV |
| MATERIAL | MATERIAL | MAT_CODIGO |
| PROVEEDOR | PROVEEDOR | PRO_CEDRUC |
| TIPOEMPLEADO | TIPOEMPLEADO | TEM_CODIGO |

Para la realización del proyecto se ha visto la necesidad de implantar los siguientes estándares para el desarrollo:

- Los formularios tendrán nombres específicos donde la primera letra de la palabra estará en mayúscula y las demás minúsculas, seguido del prefijo "Frm" separados con un guión bajo "_". En caso de que se tenga dos palabras o más, se escribirá la inicial de cada palabra en mayúscula y las demás en minúscula unidas las dos palabras y lo demás igual que lo redactado anteriormente. Ejemplo:



| FORMULARIO | NOMENCLATURA |
|------------------------------|-------------------|
| Formulario cliente | Cliente_Frm |
| Formulario factura de compra | FacturaCompra_Frm |

NOMENCLATURA DE FORMULARIOS

TABLA N° 4.12.1

FUENTE: Gina Cumbicus

- Las variables públicas y privadas tendrán el mismo formato como se encuentra en la base de datos. En caso de que se necesiten variables extras se escribirá el nombre de cada una de ellas en minúscula, si son dos palabras o más, de la misma manera se escribirán en minúscula seguida de la otra palabra y separadas por un guión bajo “_”. Ejemplo:

| VARIABLE | NOMENCLATURA |
|---------------------|--------------------|
| Nombre de la bodega | bod_nombre |
| Secuencial cliente | Secuencial_cliente |

NOMENCLATURA DE LAS VARIABLES

TABLA N° 4.12.2

FUENTE: Gina Cumbicus

- Las funciones estarán en infinitivo singular lo que se desea hacer con la primera inicial de cada palabra en mayúscula y las demás en minúscula, seguido por un guión bajo “_” con el nombre de la entidad. Ejemplo:

| FUNCIÓN | NOMENCLATURA |
|---------------------|---------------------|
| Insertar Cliente | Insertar_Cliente |
| Actualizar Material | Actualizar_Material |

NOMENCLATURA DE FUNCIONES

TABLA N° 4.12.3

FUENTE: Gina Cumbicus

- La estandarización de los objetos más utilizados en los formularios serán los siguientes:

| OBJETOS | ESTANDARIZACIÓN |
|---------|-----------------|
|---------|-----------------|



| | |
|----------------|-----|
| ETIQUETAS | lbl |
| CAJAS DE TEXTO | txt |
| BOTONES | btn |
| COMBO | cbo |
| LIST | lst |
| TABLAS | tbl |
| FORMULARIOS | frm |
| IMAGENES | img |
| SCROLL PANEL | scr |

ESTANDARIZACIÓN DE LOS OBJETOS

FIGURA N° 4.12.4

FUENTE: Gina Cumbicus

- Para usar cada uno de ellos, primero se escribe la inicial de cada objeto seguida de las tres primeras iniciales por cada palabra que se desee utilizar. La primera inicial en mayúscula y las demás en minúscula. Ejemplo:

| OBJETO | NOMENCLATURA |
|---------------------------|---------------------|
| Label cliente | lblCli |
| Texto nombre del material | txtNomMat |

NOMENCLATURA PARA OBJETOS

TABLA N° 4.12.5

FUENTE: Gina Cumbicus

- Para el modelo conceptual las tablas deberán ser nominadas en su estado original acompañada de un prefijo que identifique los procesos de la empresa. El nombre no debe exceder de los 20 caracteres, deberán estar en mayúsculas y sin ningún tipo de separación. Por ejemplo:

| TABLA | NOMENCLATURA |
|--------------------------|---------------------|
| Cliente | CLIENTE |
| Detalle Factura de Venta | DETALLEFACTURAVENTA |

NOMENCLATURA

TABLA N° 4.12.6

PARA TABLAS



FUENTE: Gina Cumbicus

- En caso de los atributos o campos se los nombrará con las tres primeras iniciales del nombre de la tabla y con la palabra en su estado original de lo que se desea detallar, separados por un guión bajo “_”. En caso de que sea una tabla con dos palabras distintas, se tomará en cuenta la primera inicial de la primera palabra y las dos primeras iniciales de la segunda con lo que se desea detallar y si tiene tres palabras se toma en cuenta las tres primeras iniciales de cada una y lo demás es igual que las anteriores. Por ejemplo:

| TABLA | NOMBRE | NOMENCLATURA |
|---------------------|--------------------------------------|--------------|
| PROVEEDOR | Código del Proveedor | PRO_CODIGO |
| INVENTARIOFISICO | Nombre del Inventario Físico | IFI_NOMBRE |
| DETALLEFACTURAVENTA | Fecha detalle de la factura de venta | DFV_FECHA |

NOMENCLATURA PARA ATRIBUTOS

TABLA N° 4.12.7

FUENTE: Gina Cumbicus

- En cuanto se refiere a las relaciones se realizará con el nombre de la tabla completo, en minúscula separado con un guión bajo “_”, con la segunda tabla que se quiera relacionar, en caso de que el nombre de la tabla tenga más de dos palabras solo se tomara en cuenta las tres primeras iniciales de cada palabra como se puede ver a continuación:

| TABLA 1 | TABLA 2 | RELACIÓN | NOMENCLATURA |
|------------------|----------|----------------------------|-----------------|
| TIPO EMPLEADO | EMPLEADO | Tipo empleado - empleado | tipemp_empleado |
| MATERIAL | BODEGA | Material – bodega material | material_bodmat |

NOMENCLATURA PARA RELACIONES

TABLA N° 4.12.8

FUENTE: Gina Cumbicus

4.13 Pantallas Y Reportes

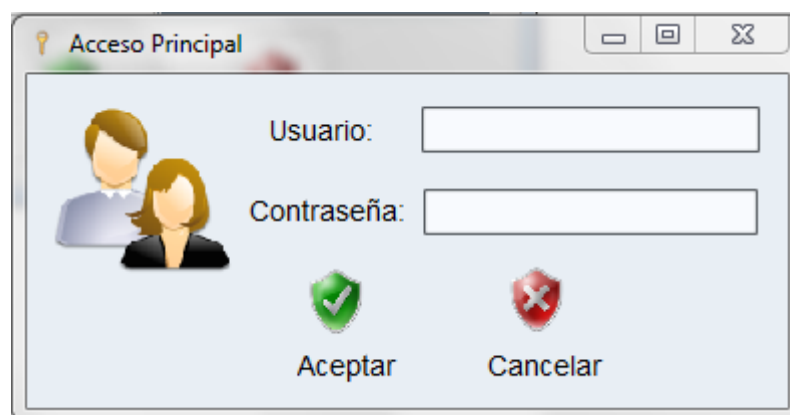


- Las pantallas del prototipo irán apareciendo según el diagrama estructurado, primero aparecerá el Splash indicando el nombre de la empresa con su logo respectivo y el nombre del autor. Para este diseño se han incluido colores de acuerdo al logo de la empresa.



PANTALLA SPLASH
FIGURA N° 4.13.1
FUENTE: Gina Cumbicus

- En el formulario acceso al sistema, el usuario deberá digitar su nombre y contraseña, que será verificado en la base de datos comprobando si el usuario ingresado tiene acceso a la aplicación según el cargo que ocupa, además si está en estado activo o inactivo.



PANTALLA ACCESO AL SISTEMA
FIGURA N° 4.13.2
FUENTE: Gina Cumbicus

- En el menú principal estarán todos los procesos clasificados en submenús, como son: Archivo, Mantenimiento, Procesos, Reportes y Usuarios.



MENÚ PRINCIPAL

FIGURA N° 4.13.3

FUENTE: Gina Cumbicus

- Para los formularios de mantenimiento de tablas tendrán un mismo estándar de diseño, es decir tendrán botones de (nuevo, guardar y eliminar), una tabla donde se mostrara toda la información ingresada y compuesto por diferentes cajas de texto, así como diferentes etiquetas que acompañaran a cada caja de texto, en el respectivo formulario.

| Ced/Ruc | Nombre | Apellido | Dirección | Teléfono | Estado |
|------------|--------|----------|------------|----------|--------|
| 1723456784 | Andres | Garcia | La Marin | 2334343 | A |
| 1715521454 | July | Cumbicus | Carapungo | 2678931 | I |
| 1102264395 | Marco | Castillo | Calderon | 2344453 | I |
| 1792826713 | Rosa | Campos | La Florida | 2342434 | A |

PANTALLA MANTENIMIENTO PROVEEDORES

FIGURA N° 4.13.4

FUENTE: Gina Cumbicus

- Para las consultas los formularios no tendrán estándares de diseño puesto que



INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

es de acorde a las necesidad de la empresa, estarán conformadas en algunos casos de combos, botones, cajas de texto y una tabla en donde se desplegará la toda la información.

PANTALLA CONSULTA INVENTARIO

FIGURA N° 4.13.5

FUENTE: Gina Cumbicus

- Para los procesos de compra, venta, devolución en compra y devolución en ventas todos tendrán un formato similar, los cuales estarán conformados de botones, combos, cajas de texto y una tabla donde se desplegará los detalles.

PANTALLA FACTURA DE COMPRA

FIGURA N° 4.13.6

FUENTE: Gina Cumbicus

- El formulario de Stock de Materiales se podrá seleccionar el nombre de la bodega



INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

y se desplegará la lista de materiales existentes en cada una con los respectivos detalles que se desplegarán en una tabla.

The screenshot shows a window titled "Stock" with a "Guardar" button and a dropdown menu for "Seleccione la bodega:" set to "BOD1". Below is a table with the following data:

| Cantidad | Material | Costo | Costo Venta | Iva | Estado | Costos | Detalle Stock |
|----------|----------|-------|-------------|-------------------------------------|-------------------------------------|---------|-----------------------------|
| 65.0 | Lata | 3.0 | 2.0 | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | NINGUNO | Ver Detalle |
| 59.0 | Limalla | 1.0 | 0.0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | NINGUNO | Ver Detalle |
| 90.0 | Aerosol | 3.0 | 0.0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | NINGUNO | Ver Detalle |
| 45.0 | Aluminio | 2.0 | 0.0 | <input type="checkbox"/> | <input checked="" type="checkbox"/> | NINGUNO | Ver Detalle |
| 3.0 | cobre | 3.0 | 0.0 | <input type="checkbox"/> | <input type="checkbox"/> | NINGUNO | Ver Detalle |

PANTALLA STOCK DE MATERIALES

FIGURA N° 4.13.7

FUENTE: Gina Cumbicus

- Los reportes tendrán un formulario distinto a los anteriores, puesto que el usuario podrá ver todos los registros guardados de antemano en una pantalla blanca lista para imprimir, los reportes de cada tabla de mantenimiento así como las tablas de los procesos.



REPORTE MODELO

| MATERIAL: | PESO(libras): |
|-----------|---------------|
| COBRE | 50 |
| BRONCE | 120 |
| ALUMINIO | 420 |
| PERFIL | 65 |
| HIERRO | 180 |

PANTALLA REPORTE MODELO
FIGURA N° 4.13.8
FUENTE: Gina Cumbicus

4.14 Pruebas y Depuración

Una vez que se ha obtenido el programa ejecutable, este es sometido a prueba a fin de determinar si resuelve o no el problema planteado en forma satisfactoria. El proceso de depuración consiste en localizar los errores y corregirlos en caso de que estos existan.

De acuerdo a este criterio utilizaremos diferentes tipos de pruebas en el prototipo del Sistema de Facturación y Control de Inventarios. Las pruebas que se le aplican son de diversa índole y generalmente dependen del tipo de problema que se está resolviendo.

- **Prueba Funcional**

La primera prueba comienza por introducir datos válidos, inválidos e incongruentes y observar cómo reacciona el prototipo en cada ocasión. Se verifica que se aplique apropiadamente cada regla de negocio, además que los resultados esperados ocurran cuando se usen datos válidos, y que se desplieguen los



mensajes de error cuando sean inválidos.

- **Prueba de Validación.**

Se realiza una prueba de todo el prototipo, realizando todos los procesos que ejecuta el software con el fin de comprobar si cumple los requisitos funcionales y de rendimiento, facilidad de mantenimiento y recuperación de errores.

- **Prueba del prototipo del Sistema.**

Se realiza las pruebas con la intervención de los usuarios y se toma nota de todas las novedades que se presenten, para luego dar una solución a los problemas existentes.

- **Prueba de Seguridad.**

Verificar que las personas que utilicen el programa solo puedan acceder a las funciones y datos que se les han asignado de acuerdo al cargo que ocupan dentro de la empresa, para esto se crea pruebas para cada tipo de usuario y se verifica cada permiso, creando transacciones específicas para cada tipo de usuario, se hacen modificaciones y nuevamente se vuelve a ejecutar.

- **Pruebas de Resistencia.**

Verificar que la aplicación funcione adecuadamente, cuando se cargan muchos datos en la base y comprobar que soporte dicha información en un periodo de tiempo dado.

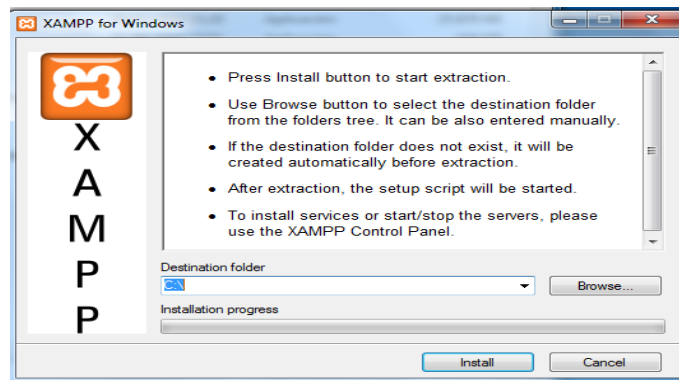
4.15 Instalación del Sistema

Para la instalación del prototipo se utilizara el Sistema Operativo XP con el cual trabajan en la empresa. Los pasos para la instalación del programa se detallan a



continuación:

- Primero se instala un servidor en este caso Xampp donde tendremos la Base de Datos MySQL y el servidor web Apache.

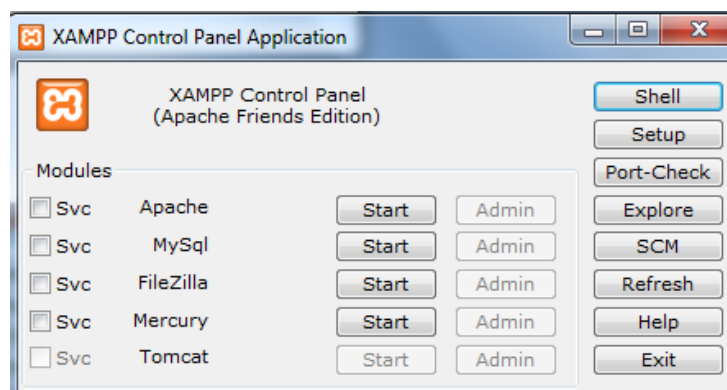


INSTALACIÓN DEL SERVIDOR XAMPP

FIGURA N° 4.15.1

FUENTE: Servidor Xampp

- Simplemente damos click en el botón Instalar y en lo demás solo ponemos siguiente. Una vez instalado, se activa el servidor Apache y la base de datos MySQL, haciendo click en los botones de Start como vemos a continuación:

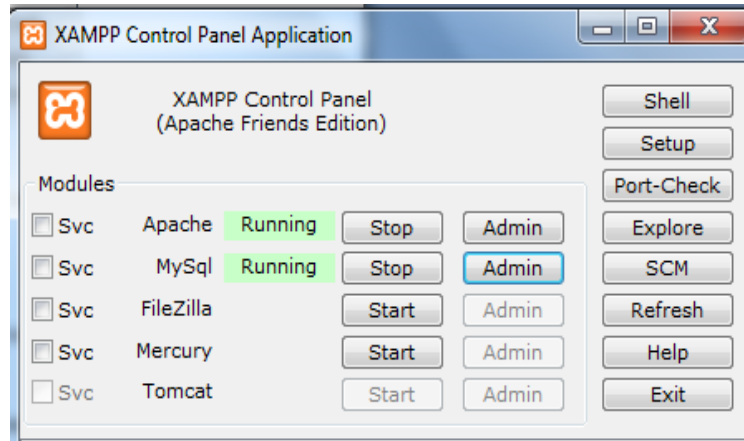


ENCENDER EL SERVIDOR XAMPP

FIGURA N° 4.15.2

FUENTE: Servidor Xampp

- Una vez encendido nos muestra una pantalla de la siguiente manera:

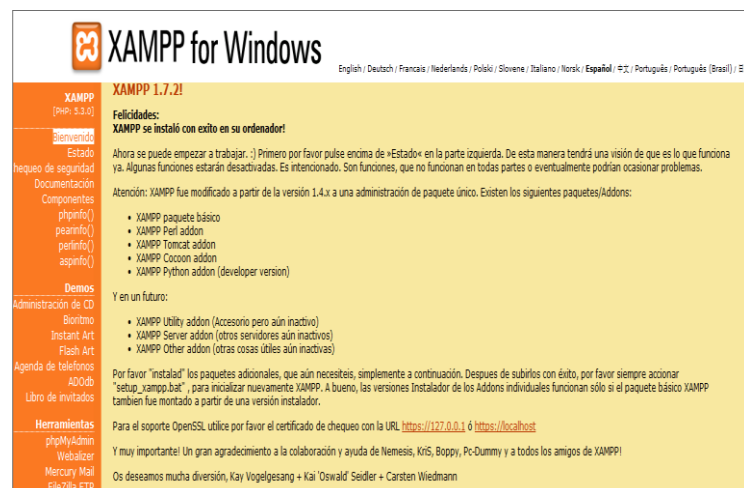


SERVIDOR XAMPP ENCENDIDO

FIGURA N° 4.15.3

FUENTE: Servidor Xampp

- Damos click en el botón Admin y se despliega la siguiente pantalla:

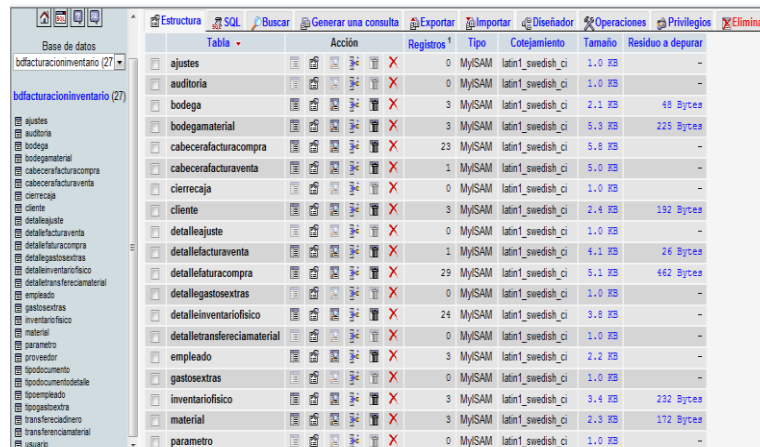


ADMINISTRADOR DEL SERVIDOR XAMPP

FIGURA N° 4.15.4

FUENTE: Servidor Xampp

- En la parte donde dice Herramientas damos click en phpMyAdmin que es donde crearemos la base de datos con el nombre “bdfacturacioninventario” y subiremos el script, ahí se cargaran todas las tablas que se crearon en PowerDesigner, obteniendo el siguiente resultado:



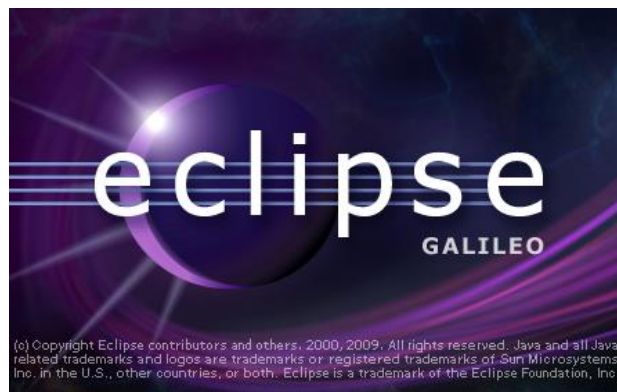
| Tabla | Acción | Registros ¹ | Tipo | Cotejamiento | Tamaño | Residuo a depurar |
|------------------------------|--------|------------------------|--------|-------------------|--------|-------------------|
| ajustes | | 0 | MyISAM | latin1_swedish_ci | 1.0 KB | - |
| auditoria | | 0 | MyISAM | latin1_swedish_ci | 1.0 KB | - |
| bodega | | 3 | MyISAM | latin1_swedish_ci | 2.1 KB | 48 Bytes |
| bodegamaterial | | 3 | MyISAM | latin1_swedish_ci | 5.3 KB | 225 Bytes |
| cabecerafacturacompra | | 23 | MyISAM | latin1_swedish_ci | 5.8 KB | - |
| cabecerafacturaventa | | 1 | MyISAM | latin1_swedish_ci | 5.0 KB | - |
| cierrecaja | | 0 | MyISAM | latin1_swedish_ci | 1.0 KB | - |
| cliente | | 3 | MyISAM | latin1_swedish_ci | 2.4 KB | 192 Bytes |
| detalleajuste | | 0 | MyISAM | latin1_swedish_ci | 1.0 KB | - |
| detallefacturaventa | | 1 | MyISAM | latin1_swedish_ci | 4.1 KB | 26 Bytes |
| detallefacturacompra | | 29 | MyISAM | latin1_swedish_ci | 5.1 KB | 462 Bytes |
| detallegastosextras | | 0 | MyISAM | latin1_swedish_ci | 1.0 KB | - |
| detalleinventariofisico | | 24 | MyISAM | latin1_swedish_ci | 3.8 KB | - |
| detalletransferenciamaterial | | 0 | MyISAM | latin1_swedish_ci | 1.0 KB | - |
| empleo | | 3 | MyISAM | latin1_swedish_ci | 2.2 KB | - |
| gastosextras | | 0 | MyISAM | latin1_swedish_ci | 1.0 KB | - |
| inventariofisico | | 3 | MyISAM | latin1_swedish_ci | 3.4 KB | 232 Bytes |
| material | | 3 | MyISAM | latin1_swedish_ci | 2.3 KB | 172 Bytes |
| parametro | | 0 | MyISAM | latin1_swedish_ci | 1.0 KB | - |

CREAR BASE DE DATOS

FIGURA N° 4.15.5

FUENTE: Servidor Xampp

- Se procede a instalar Java JDK 1.6 que es un software que provee herramientas de desarrollo para la creación de programas en java.
- Se ejecuta el aplicativo Eclipse Galileo y se procede a cargar el prototipo, con todas las librerías que sean necesarias.

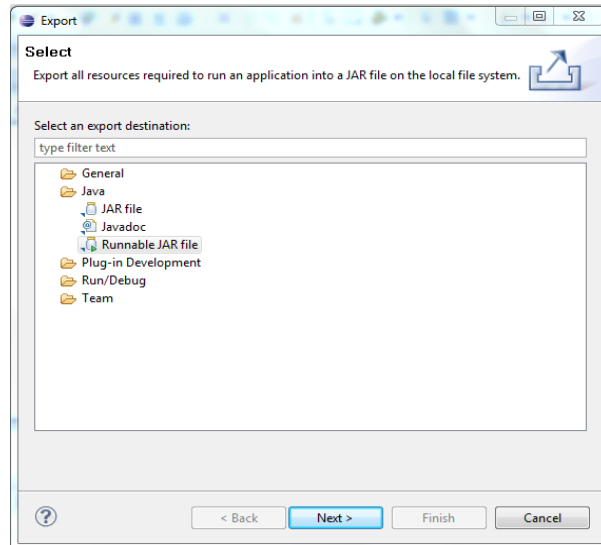


ECLIPSE

FIGURA N° 4.15.6

FUENTE: Aplicativo Eclipse Galileo

- En el entorno de desarrollo se generara un archivo ejecutable (“Runnable jar file”) de la siguiente manera: click en file/export donde se despliega la siguiente pantalla:

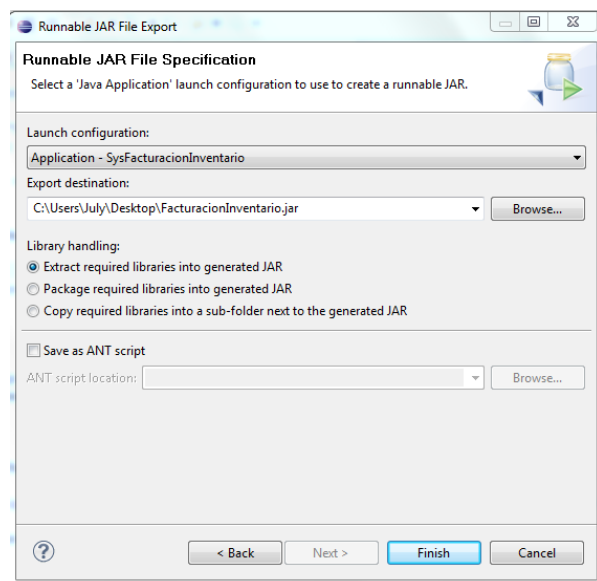


RUNNABLE JAR FILE

FIGURA N° 4.15.7

FUENTE: Aplicativo Eclipse Galileo

- Damos click en Java y seleccionamos Runnable JAR file luego aparece la siguiente pantalla:



ESPECIFICACIÓN RUNNABLE JAR FILE

FIGURA N° 4.15.8

FUENTE: Aplicativo Eclipse Galileo

- Aquí seleccionamos la opción Application –SysFacturacionInventario, en el botón Browser, escogemos la dirección donde queremos que aparezca el ejecutable de la aplicación, en este caso será en el escritorio, le damos el nombre de



INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

“Facturacion-Inventario”, hacemos click en next y ya tenemos el aplicativo listo para ser usado el programa. El ícono es de la siguiente forma:



EJECUTABLE DEL PROTOTIPO

FIGURA N° 4.15.9

FUENTE: Aplicativo Eclipse Galileo

- En todos los computadores en los que se desee utilizar la aplicación se realizaran los mismos pasos.

4.16 Recopilación y Carga de Datos

Para la recopilación de datos se tomó como base el inventario el mes de enero del 2011 y se comparó con los Kardex de cada material para poder ingresar la información referente a la mercadería, también se realizó un registro de todos los proveedores de la empresa, ya que era necesario ingresar los materiales al Stock, mediante una factura de compra, así como también el de especificar el precio de venta, y si el producto se encuentra en estado activo o inactivo. Se ingreso los movimientos o transacciones que llevará la empresa a la hora de realizar una factura de venta para poder registrar en el Kardex. Además se ingreso información de los usuarios, dependiendo del tipo de cargo que tienen en la empresa.

Los datos de los clientes, factura de venta y consultas se fueron generando según el transcurso de la prueba, por lo cual no fue necesario cargarlos en el prototipo.

Luego de todo esto se procedió a sacar una copia de todos los documentos utilizados para la carga de los datos como un respaldo escrito sobre la información que se ingresó en el



prototipo y como una referencia en la comparación de los resultados.

Cabe destacar también que en el transcurso de ingreso de datos, el Gerente de la empresa estuvo presente para poder determinar el precio de venta de cada producto, así como también los empleados de la empresa que ayudaron a la recopilación de datos.

4.17 Pruebas y Depuración Final en Funcionamiento

Una vez que el prototipo pasó la prueba con los datos reales, está listo para la prueba y depuración final, en donde se realizó, la facturación manualmente y en forma sistematizada. En el momento que finalizó el tiempo de prueba con los datos reales, se comparó los resultados generados por los dos procedimientos.

En la comparación de los resultados, la cual se realizó con presencia del gerente de la empresa para que abalice el correcto funcionamiento del prototipo a su vez determinar errores del programa para su inmediata corrección, de todo lo que ocurrió se hizo un reporte por escrito adjuntando los resultados de forma manual y del prototipo, se discutió la opción de implementar definitivamente el software que fue realizado por el alumno.

A continuación se detalla los resultados de las últimas pruebas realizadas:

| PRUEBA | RESULTADO | RESPONSABLE |
|-------------------------------|-----------|---------------|
| Acceder a la aplicación | Exitoso | Gerente |
| Ingresar nuevo cliente | Fallo | Administrador |
| Crear nueva factura de compra | Fallo | Cajero |
| Cierre de Caja | Exitoso | Cajero |
| Reporte Materiales | Exitoso | Administrador |

PRUEBA FINAL

FIGURA N° 4.17

FUENTE: Gina Cumbicus



Error al Ingresar nuevo Cliente: Cuando se hizo el ingreso de un nuevo cliente, el usuario por equivocación ingreso una letra en el número de cédula y no se presentó el mensaje de error, que solamente se ingresen números en esa caja de texto, inmediatamente se procedió validar dicho error.

Error al crear nueva factura de compra: En el momento de crear una nueva factura, no se elimino los detalles de la tabla de la factura de compra creada anteriormente, por lo que la nueva factura guardo también los datos de la anterior, se procedió a validar el error inmediatamente.

El resto de la pruebas fueron exitosas, aunque a medida que se siga utilizando el software podrían presentarse nuevos errores, para lo cual el usuario tiene que informar los inconvenientes al desarrollador.

4.18 Puesta en Marcha del Sistema

Para la entrega del prototipo se considera aspectos tanto para la parte técnica como la instalación, configuración y tareas o componentes adicionales como conexión a la base de datos entre el cliente y el servidor. La capacitación a usuarios finales se lo realizará de acuerdo al área de trabajo y el responsable de cada bodega.

La capacitación al personal técnico, constará de puntos tales como instalación del prototipo, conexión a base de datos, interfaces gráficas de usuarios, estructura de datos, parámetros iniciales, inicialización del prototipo y lógica de negocio.

4.19 Capacitación al Usuario Final

Los Usuarios finales del prototipo del Sistema Facturación y Control de Inventarios, han colaborado e interactuado en el desarrollo del mismo lo que facilita la operación y manejo de este.



La capacitación teórica y práctica para el manejo de las diferentes opciones de la aplicación, serán satisfechas primeramente por una charla general de aproximadamente dos horas y luego por la capacitación personalizada que se dará los usuarios. Esta capacitación se realizará en la bodega matriz ubicada en la Av. 24 de mayo y el Robo, en el Centro Histórico de Quito, el día lunes 21 y martes 22 de marzo a las 9:30 de la mañana.

4.20 Capacitación al Personal Técnico

Se ha considerado la capacitación personalizada para el personal técnico, quienes, estarán en capacidad de dar soporte y mantenimiento al prototipo y a los colaboradores de la empresa, para satisfacer los requerimientos presentes o futuros y la inclusión de nuevas aplicaciones que se integren a este módulo.

El Gerente general, recibirá el script, modelo Conceptual de Datos (CDM) y modelo Físico de datos (PDM) de la base de datos, con la documentación respectiva, programas fuentes del prototipo, interfaces de usuarios con la base de datos.

Para el personal técnico del prototipo se entregará el análisis de la lógica de negocio, interfaces de la aplicación, acceso y comunicación con la base de datos.



CAPÍTULO V

PRINCIPALES IMPACTOS

5.1 Científico

Gracias al desarrollo del proyecto se adquirieron muchos conocimientos que beneficiaran no solo a una sino a varias personas, la empresa se siente satisfecha con la implantación del prototipo ya que se han agilitado sus procesos y ahora será más competitiva en el mercado.

5.2 Educativo

El referido proyecto ha sido de gran utilidad para poder reforzar todo lo aprendido en el transcurso de la carrera, a su vez ampliar nuevos conocimientos gracias a la investigación realizada para el desarrollo del proyecto.

5.3 Técnico

En el ámbito técnico, la mayoría de las materias dictadas a lo largo de la carrera, fueron de gran ayuda para el desarrollo del proyecto, ya que se utilizó el conocimiento aprendido en las materias de análisis y diseño, desarrollo de software, redes, base de datos, organización y métodos entre otras, lo que permitió resolver todo tipo de problemas referentes al proyecto.

5.4 Tecnológico



El presente proyecto está desarrollado con todos los conocimientos adquiridos a lo largo de esta carrera, el mismo que será de gran utilidad para la empresa auspiciante ya que podrán estar al día con la tecnología.

5.5 Empresarial

El proyecto tiene el completo interés y apoyo de la empresa FFHV, ya que cumple con las expectativas propuestas.

5.6 Social

La implementación del prototipo permitirá que los usuarios cometan la menor cantidad de errores posibles y que puedan realizar procesos más rápidos, de esta manera se beneficiaran tanto los usuarios como los clientes y la empresa tendrá un mejor prestigio.

5.7 Económico

En vista de que el desarrollo del prototipo es un proyecto de grado, no tendrá costo alguno para la empresa FFHV lo que resulta atractivo para la misma.

CONCLUSIONES

- Los usuarios se sienten satisfechos con la implantación del prototipo ya que se agilizaran los procesos de compra y venta y se cometerá la menor cantidad de errores posibles, brindándole al cliente una mejor atención.
- Se tiene información real y oportuna en menos tiempo, lo que evita pérdidas a la empresa.



INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

- El personal tiene la capacidad para el manejo de un software con el debido entrenamiento.
- El software está orientado exactamente hacia los objetivos de la empresa en general, dado que el prototipo aportará con un seguimiento de los pedidos y reportes diarios a la gerencia, llevando un control más ordenado de la documentación de la empresa.
- El prototipo está reforzando la estructura de trabajo que tiene FFHV, haciéndola más rápida y confiable. De esta manera puede llegar a ser más atractiva para el mercado.
- Debido a que el prototipo está desarrollado acorde a los requerimientos del usuario, cuenta con una interfaz amigable, lo que facilita el uso del software para los usuarios.
- Con el uso del prototipo estarán a la vanguardia de la tecnología y la empresa será más competitiva.

RECOMENDACIONES

- Se recomienda la constante capacitación de los usuarios en el uso del prototipo, para que se cometan la menor cantidad de errores.
- En caso de que el prototipo presente algún tipo de error, se recomienda informar inmediatamente al desarrollador del software, para dar una solución la más pronto posible.
- Se recomienda capacitar a los usuarios con nuevas tecnologías, para que en un futuro aporten con nuevas ideas para la extensión de módulos.



- Se recomienda sacar respaldos de la información cada cierto periodo de tiempo, por cualquier inconveniente que pudiera presentarse.
- Se recomienda que solo los usuarios capacitados hagan uso del prototipo, para evitar que se cometan daños en el mismo.
- Se recomienda lo más pronto posible diseñar el módulo de Cartera de clientes.

ASPECTOS ADMINISTRATIVOS

- **Presupuesto**

Durante el desarrollo del proyecto se ha visto en la necesidad de invertir en diferentes aspectos, los cuales se detallan a continuación:

| Ingresos | | Egresos | |
|------------------|--------|--------------------|--------|
| Recursos propios | 400,00 | Movilización | 50,00 |
| | | Internet | 100,00 |
| | | Teléfono | 80,00 |
| | | Material didáctico | 100,00 |
| | | Alimentación | 50,00 |
| | | Otros Gastos | 20,00 |
| Total ingresos | 400,00 | Total egresos | 400,00 |

- **Cronograma**

Para detallar las actividades involucradas en el desarrollo del proyecto se ha elaborado un cronograma de actividades utilizando la herramienta de Project 2003.
(VER ANEXO N°19🔗)



ANEXOS

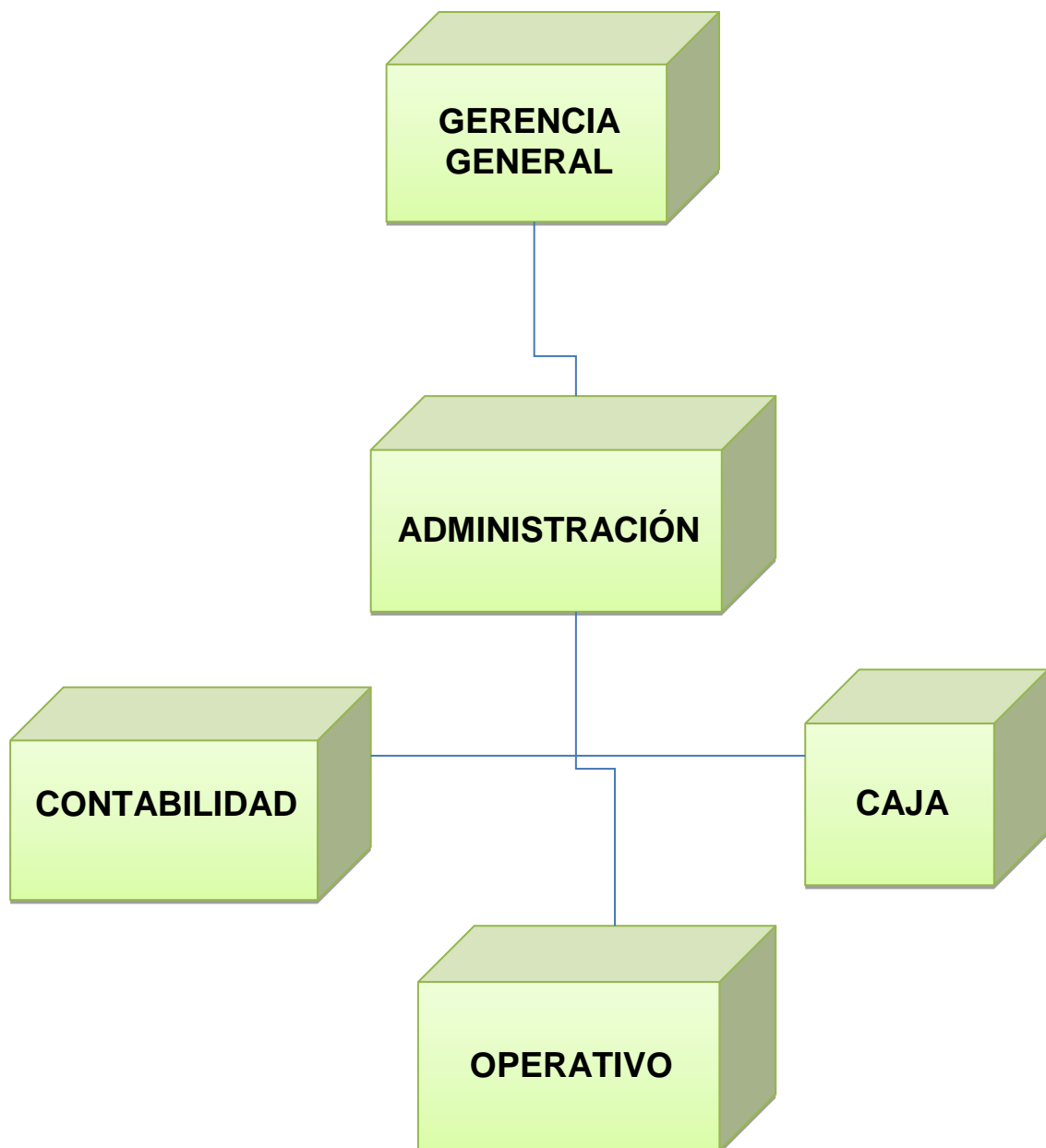


ANEXO N°1

ESTRUCTURA ORGÁNICA



ESTRUCTURA ORGÁNICA





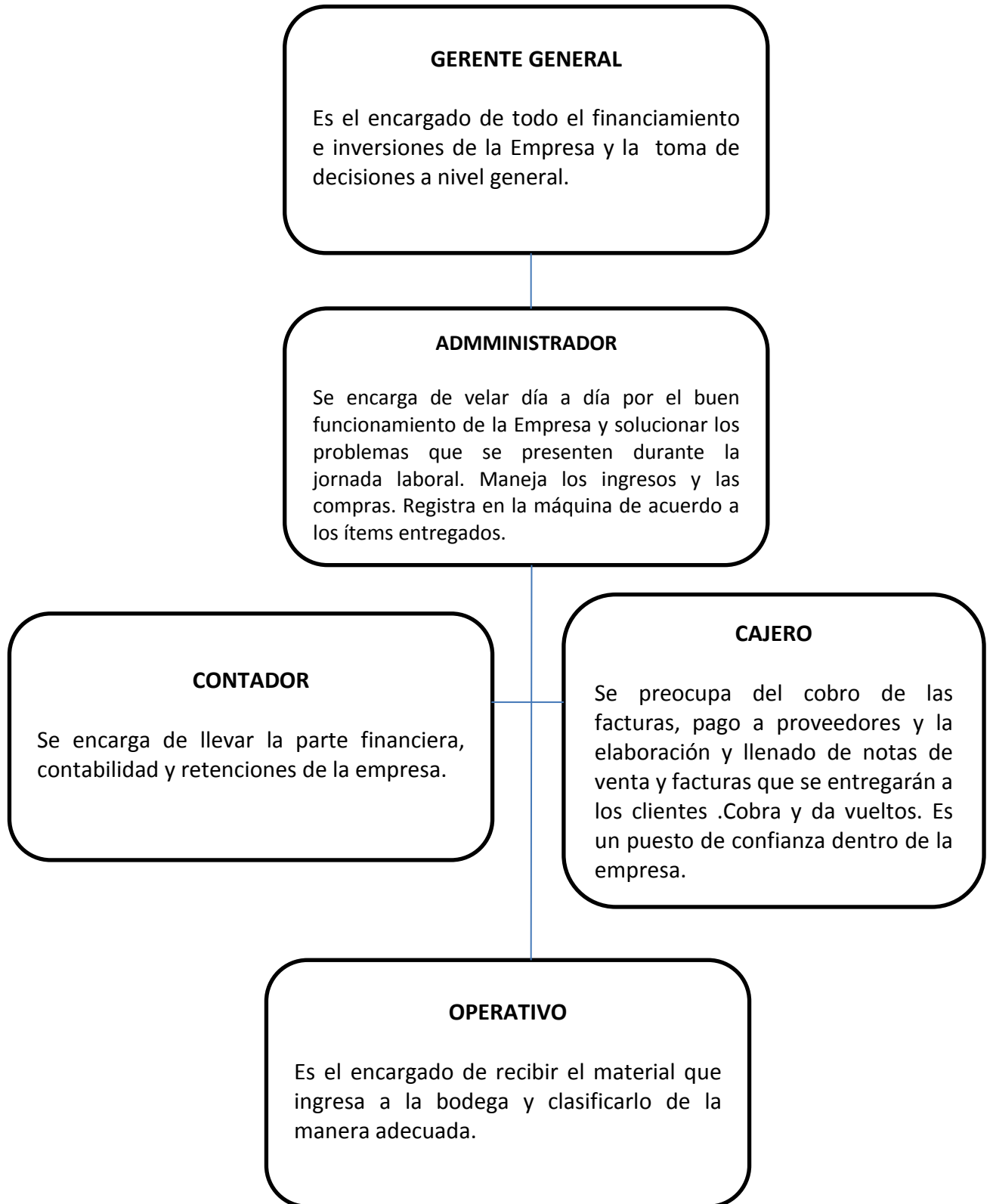
INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

ANEXO N°2

ORGÁNICO FUNCIONAL



ORGÁNICO FUNCIONAL



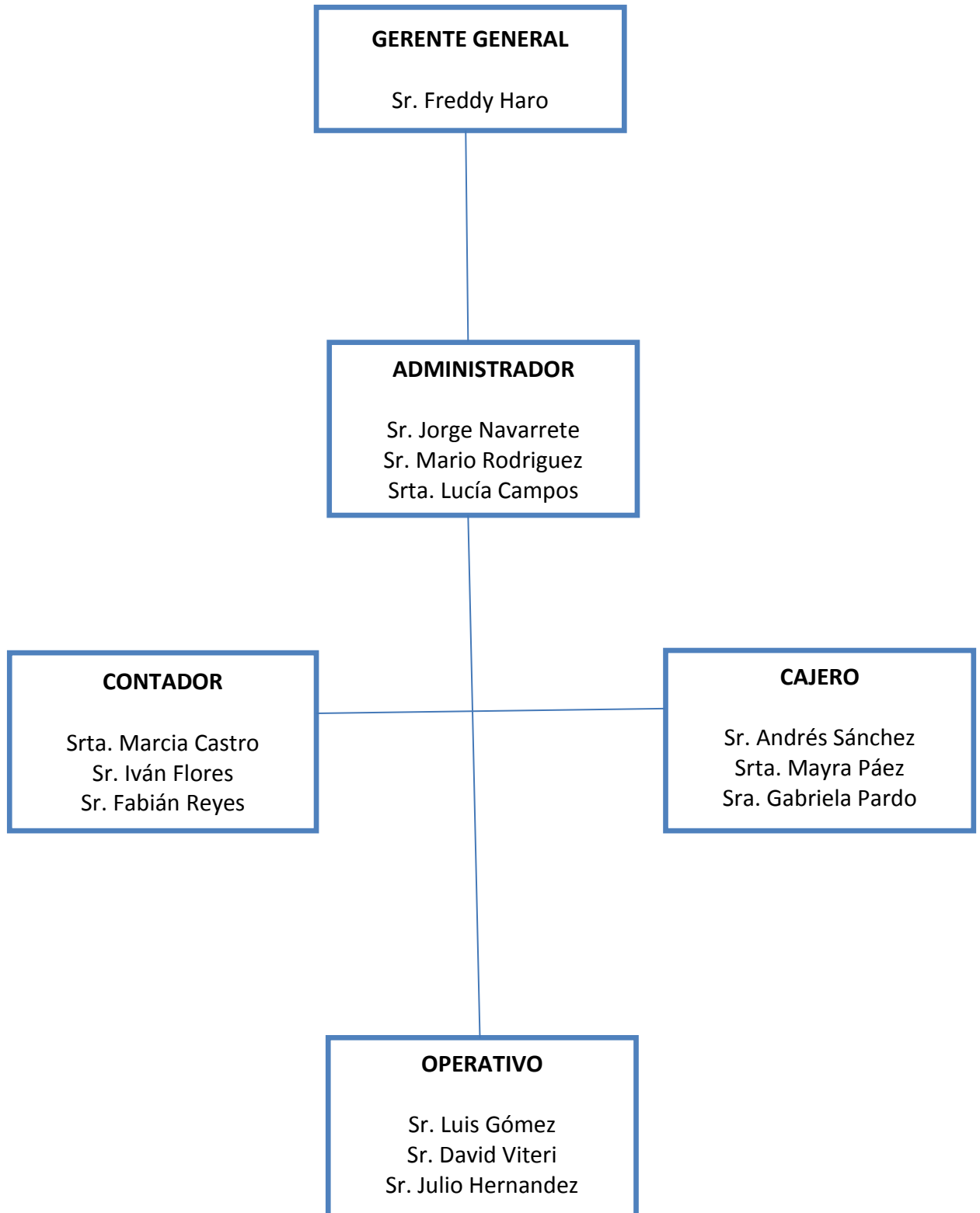


ANEXO N°3

ORGÁNICO POSICIONAL



ORGÁNICO POSICIONAL



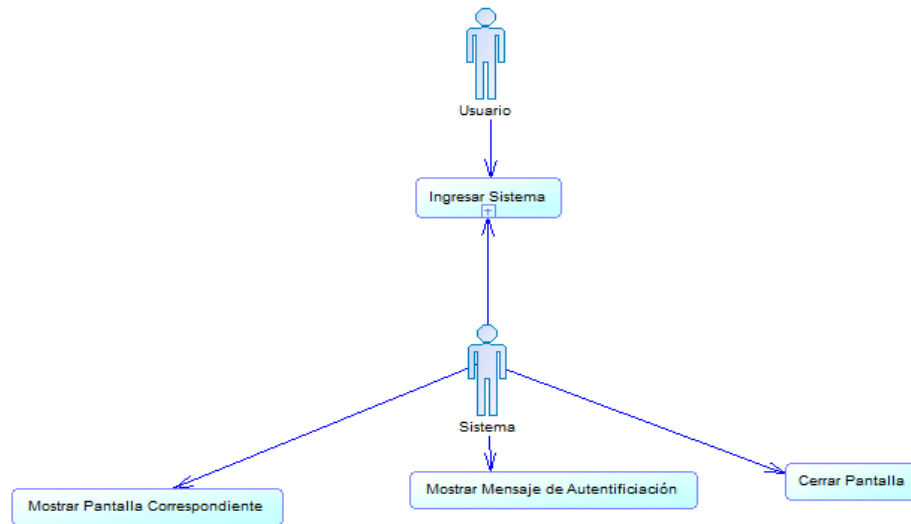


ANEXO N°4

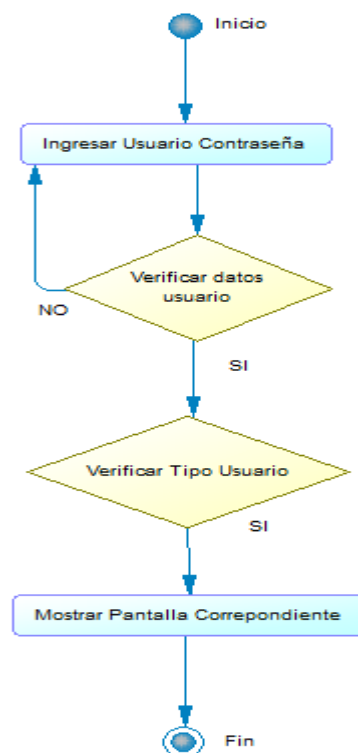
ACCESO AL SISTEMA

ACCESO AL SISTEMA

Caso de Uso



BPM





INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

ANEXO N°5

CREAR CUENTA DE USUARIO

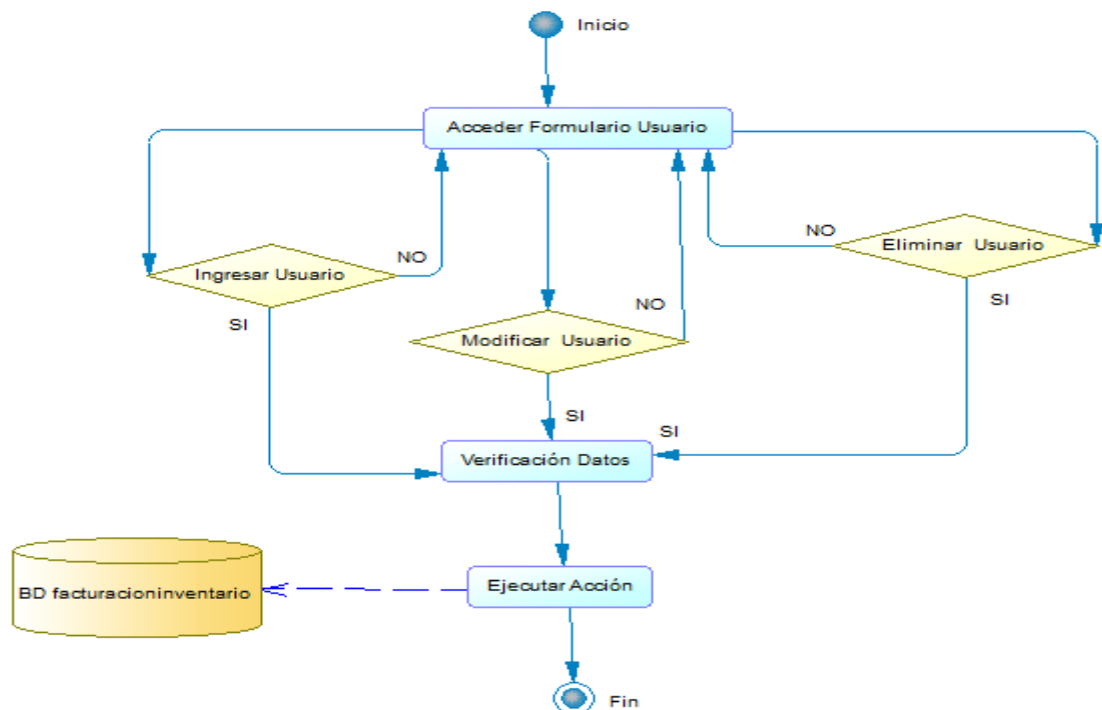


CREAR CUENTA DE USUARIO

Caso de Uso



BPM





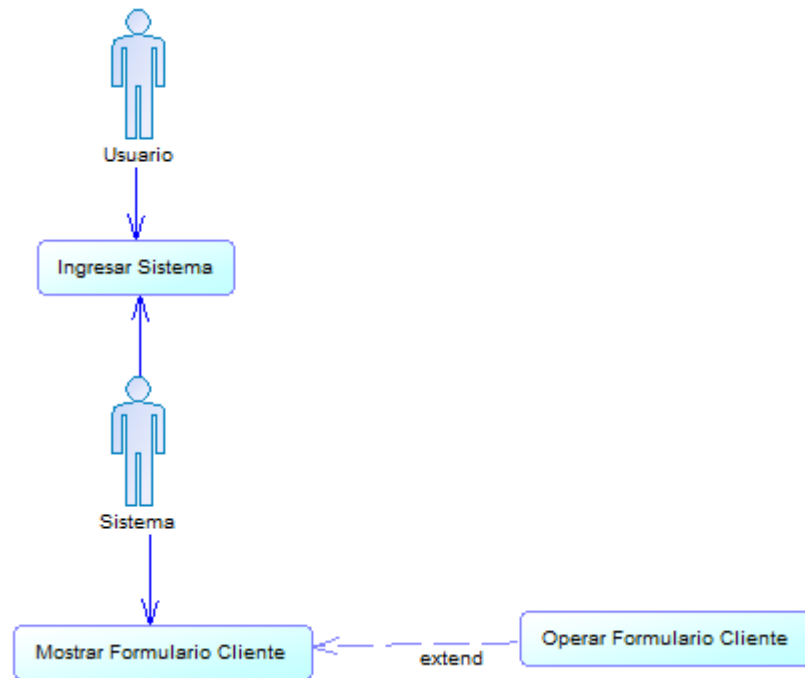
ANEXO N°6

CLIENTES

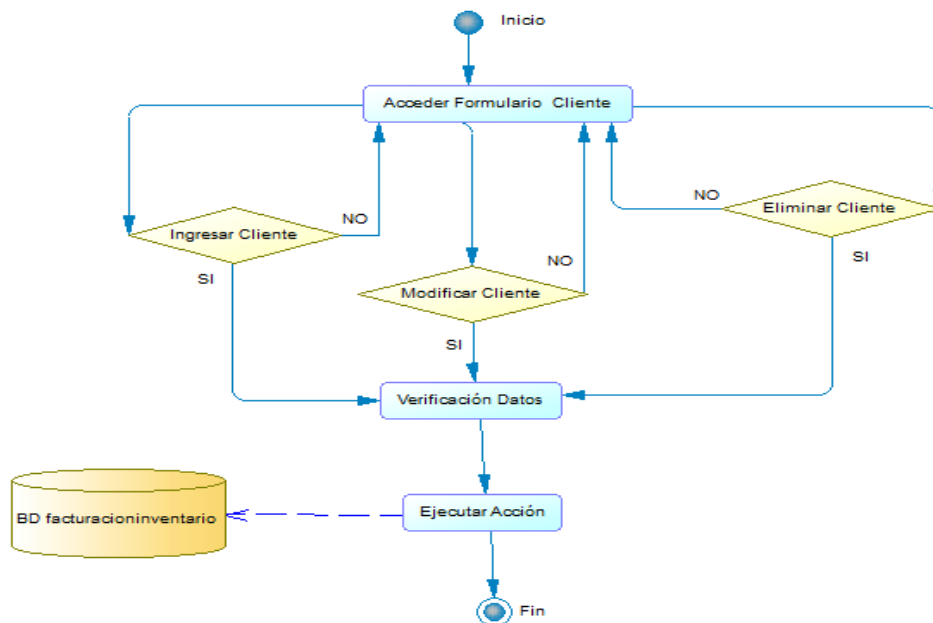


CLIENTES

Caso de Uso



BPM





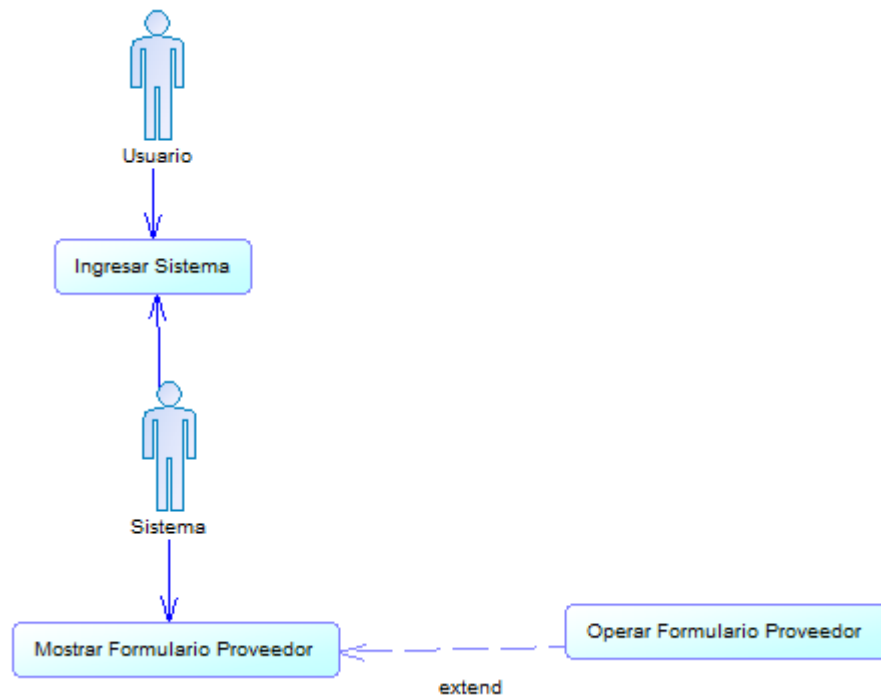
ANEXO N°7

PROVEEDORES

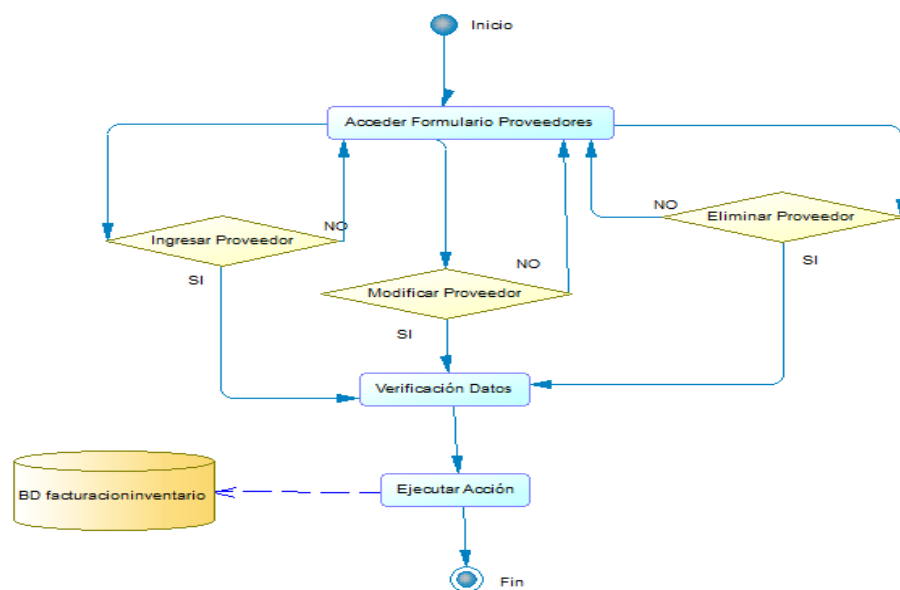


PROVEEDORES

Caso de Uso



BPM





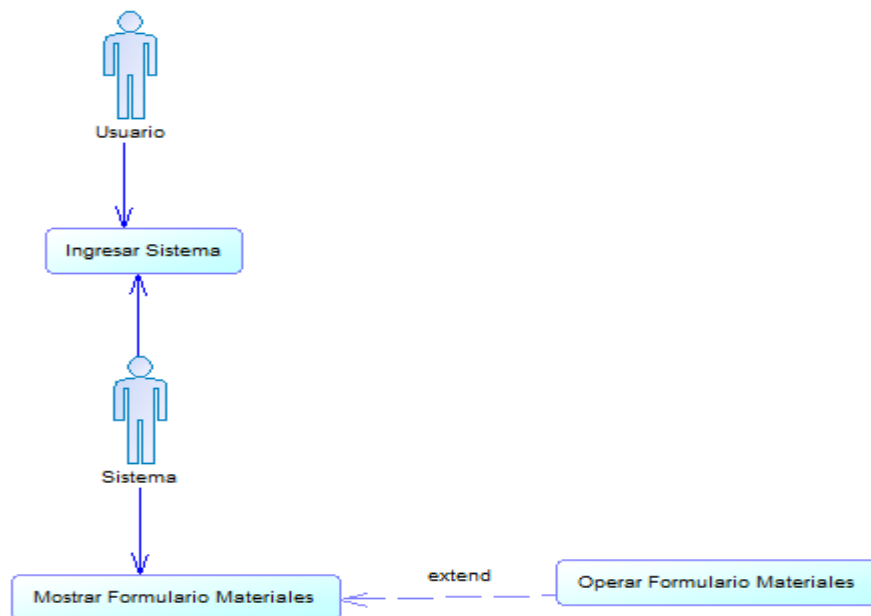
ANEXO N°8

MATERIALES

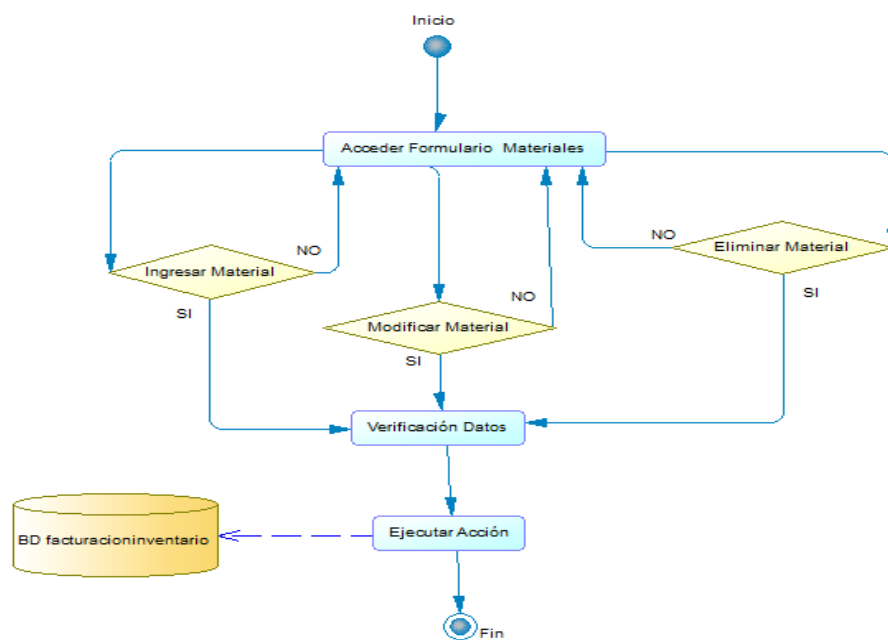


MATERIALES

Caso de Uso



BPM





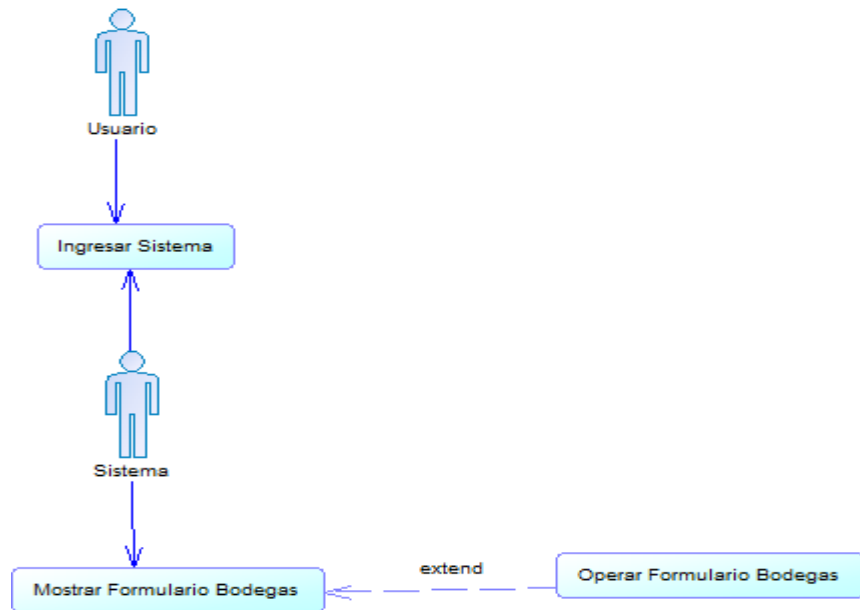
ANEXO N°9

BODEGAS

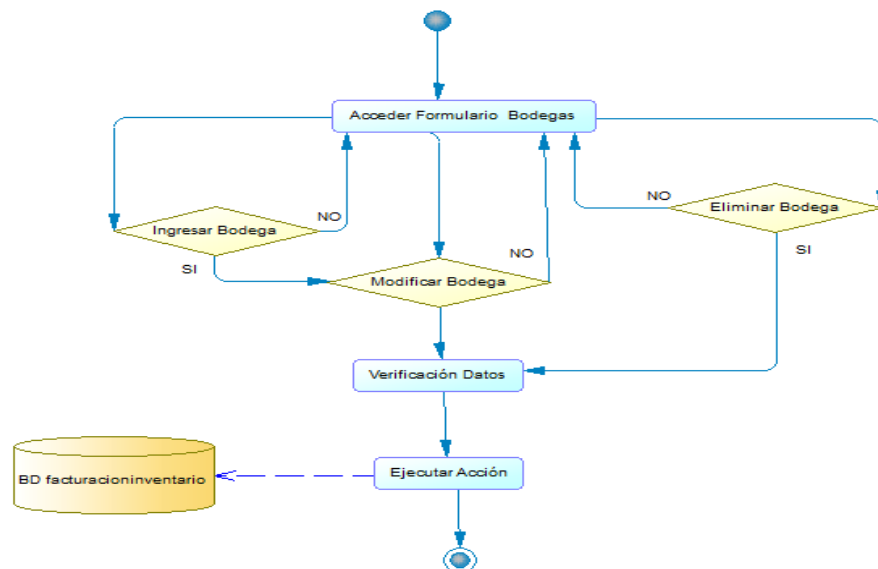


BODEGAS

Caso de Uso



BPM





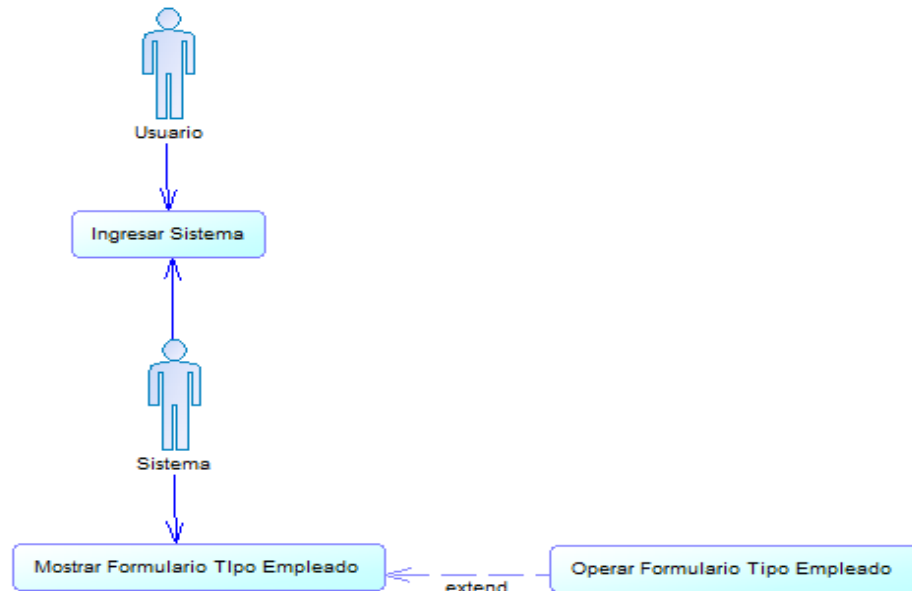
ANEXO N°10

TIPO DE EMPLEADOS

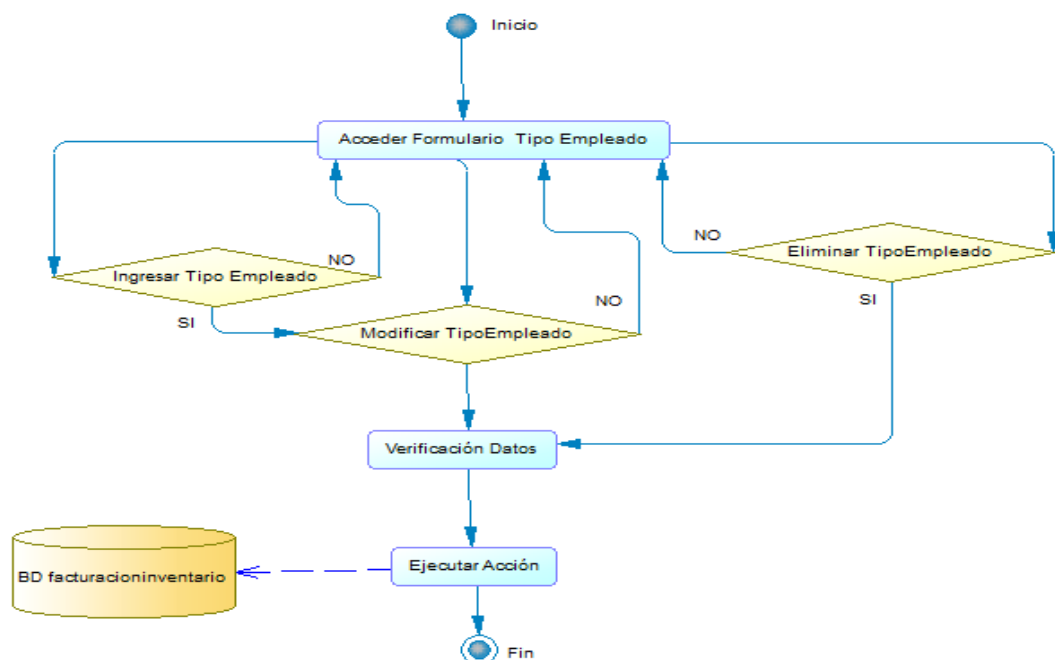


TIPO DE EMPLEADOS

Caso de Uso



BPM





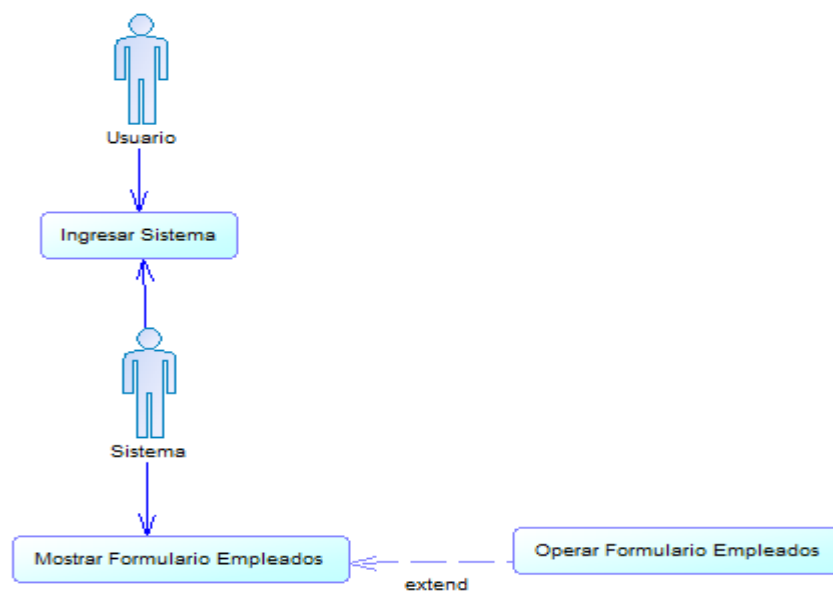
ANEXO N°11

EMPLEADOS

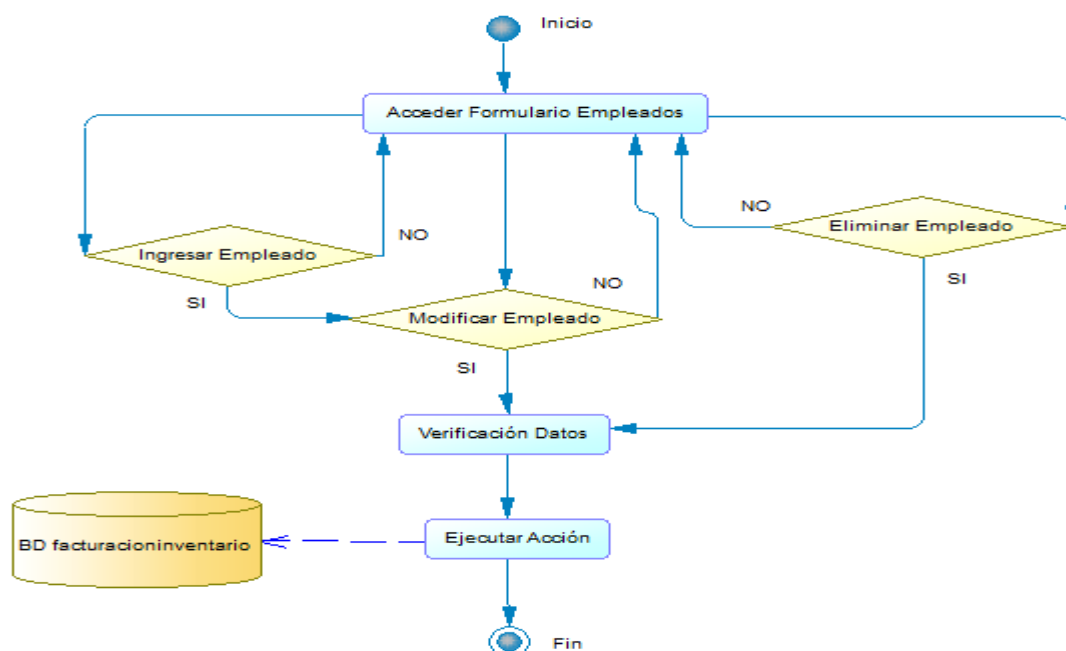


EMPLEADOS

Caso de Uso



BPM





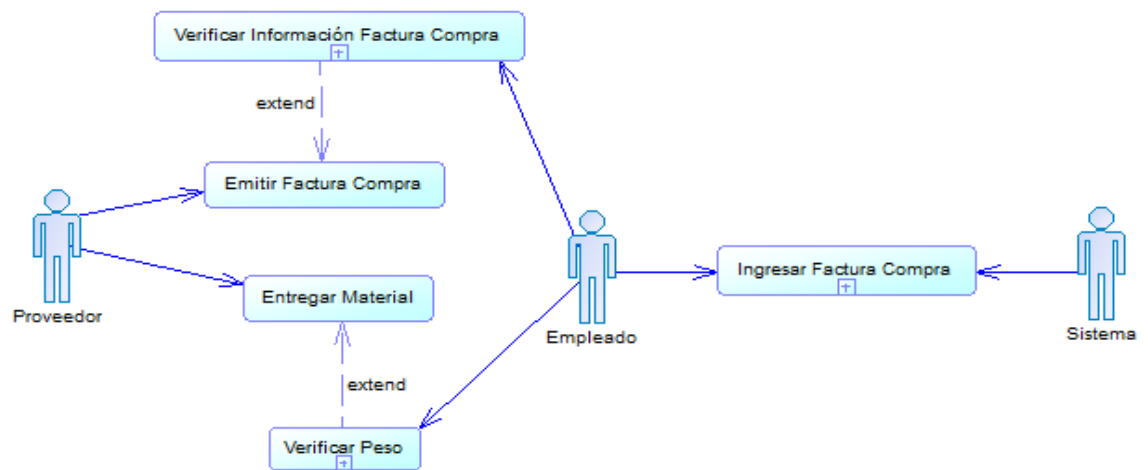
ANEXO N°12

COMPRAS



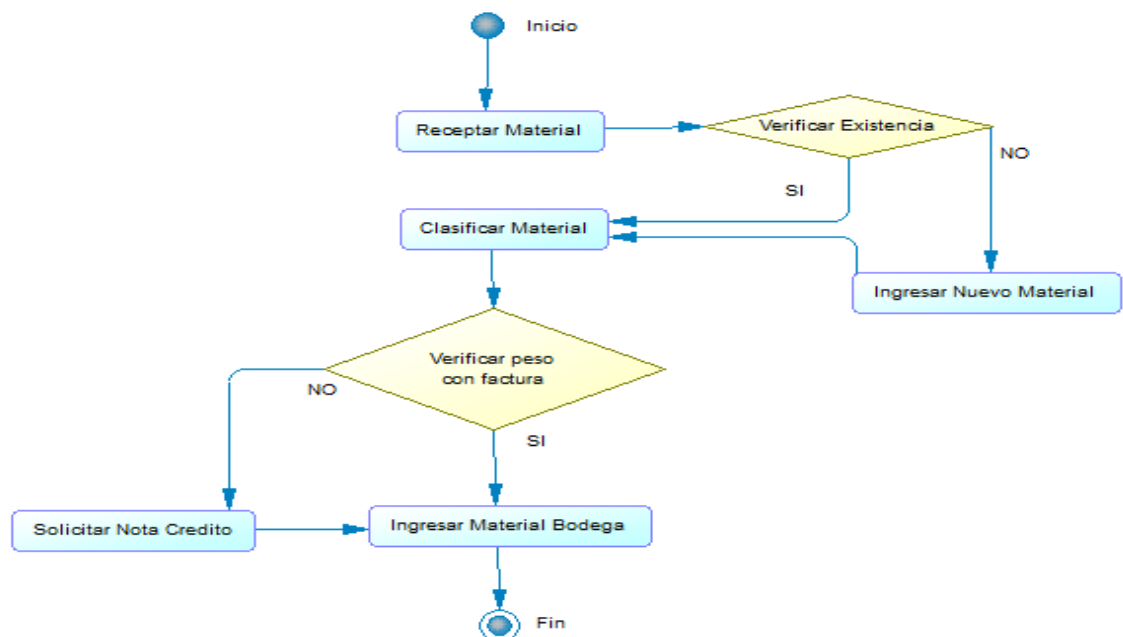
COMPRAS

Caso de Uso



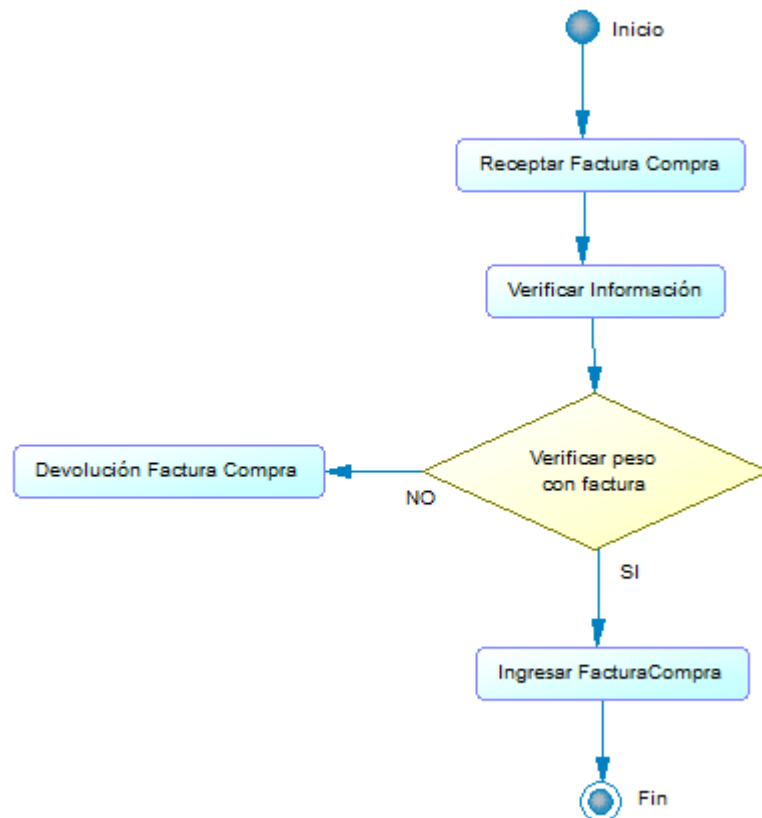
BPM

+Descripción Proceso (Verificación del Peso)



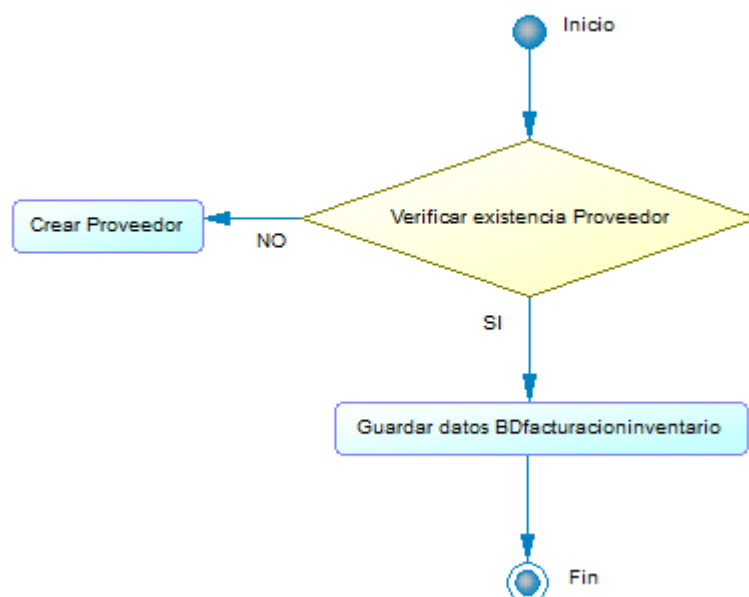


Descripción Proceso (Verificación de la Información)



Descripción Proceso

Ingresar Factura Compra





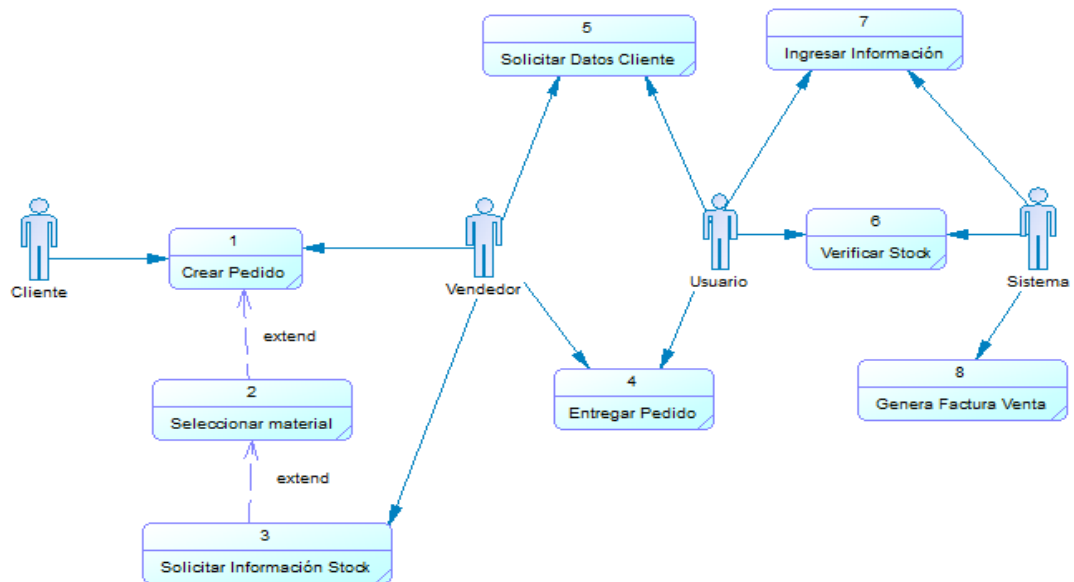
ANEXO N°13

VENTAS



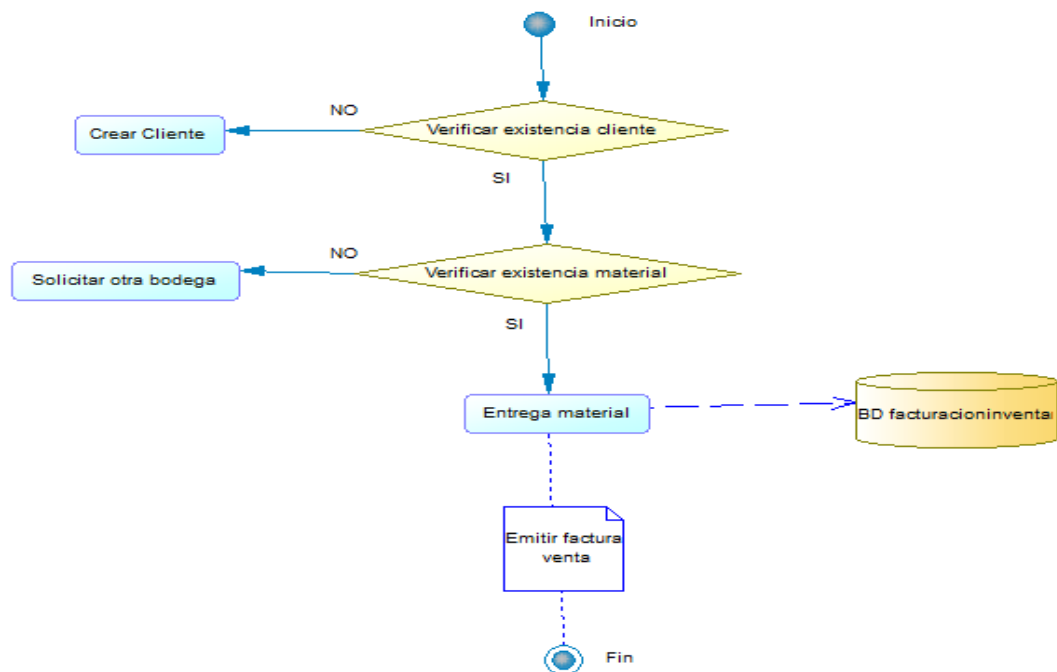
VENTAS

Caso de Uso



BPM

Descripción Proceso (Generar Factura Venta)





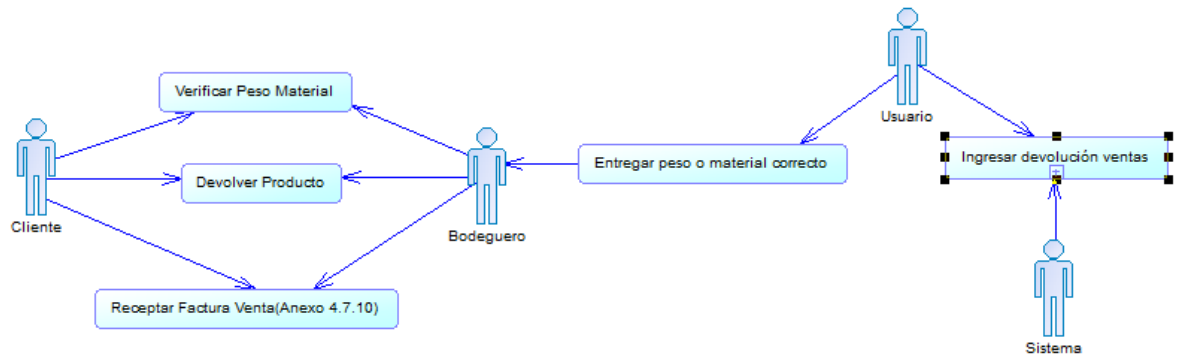
ANEXO N°14

DEVOLUCIÓN VENTAS



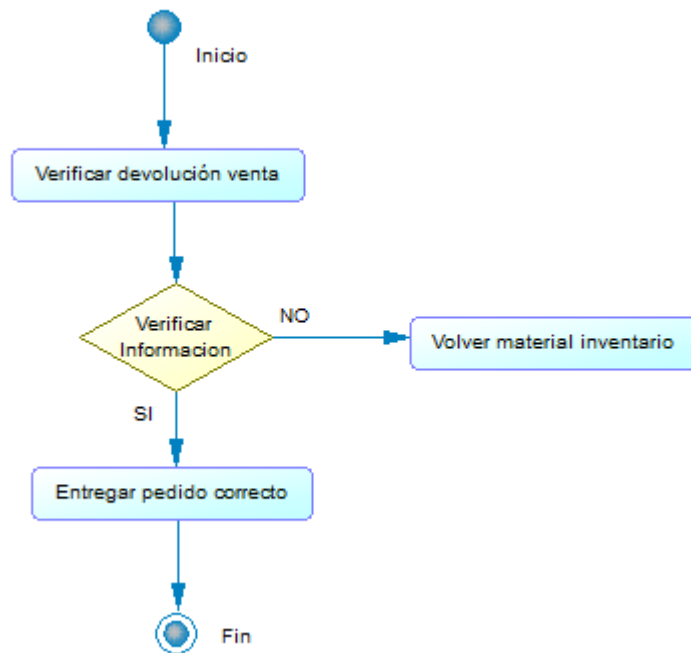
DEVOLUCIÓN VENTAS

Caso de Uso



BPM

Descripción Proceso (Ingresar Devolución en Ventas)





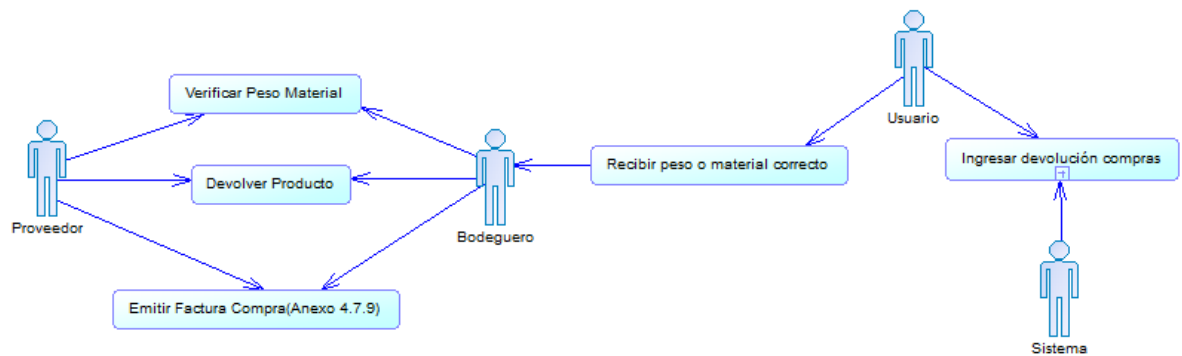
ANEXO N°15

DEVOLUCIÓN COMPRAS



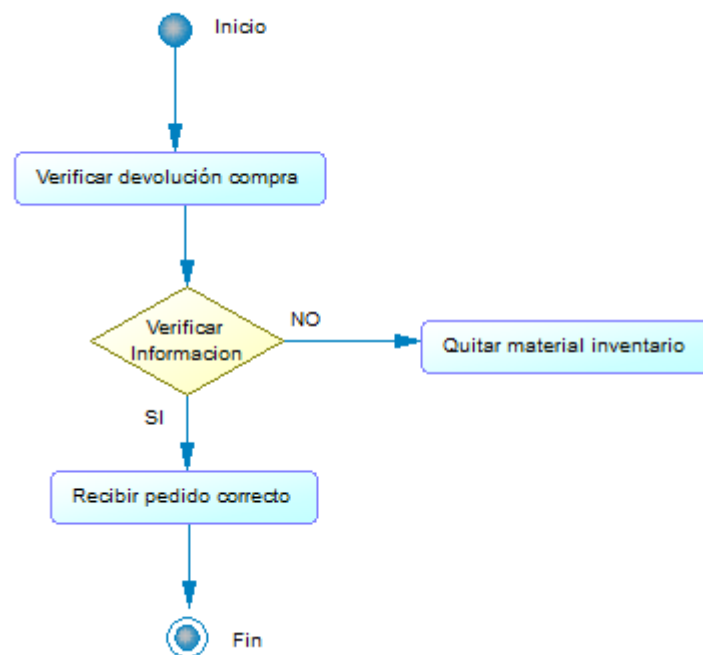
DEVOLUCIÓN COMPRAS

Caso de Uso



Descripción Proceso

Ingresar devolución por compras





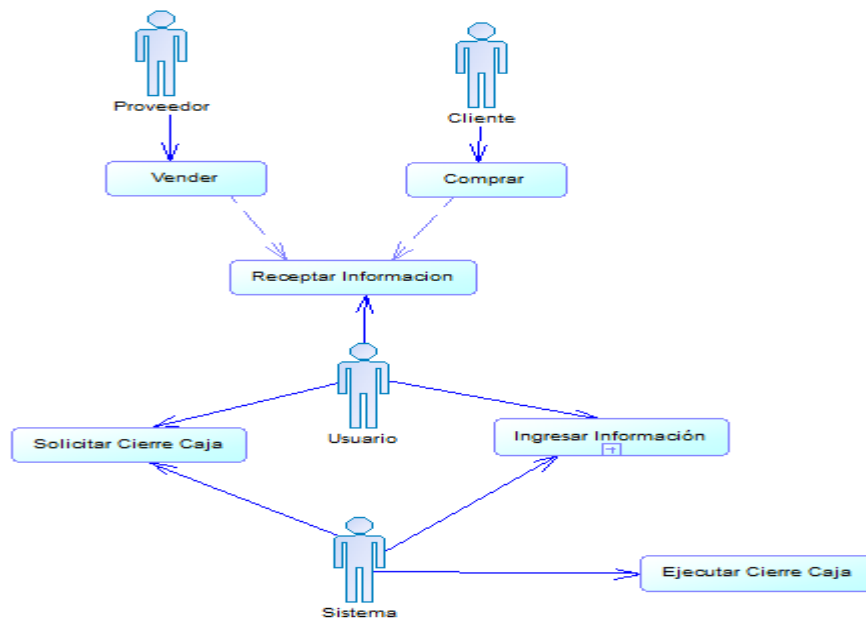
ANEXO N°16

CIERRE DE CAJA



CIERRE DE CAJA

Caso de Uso

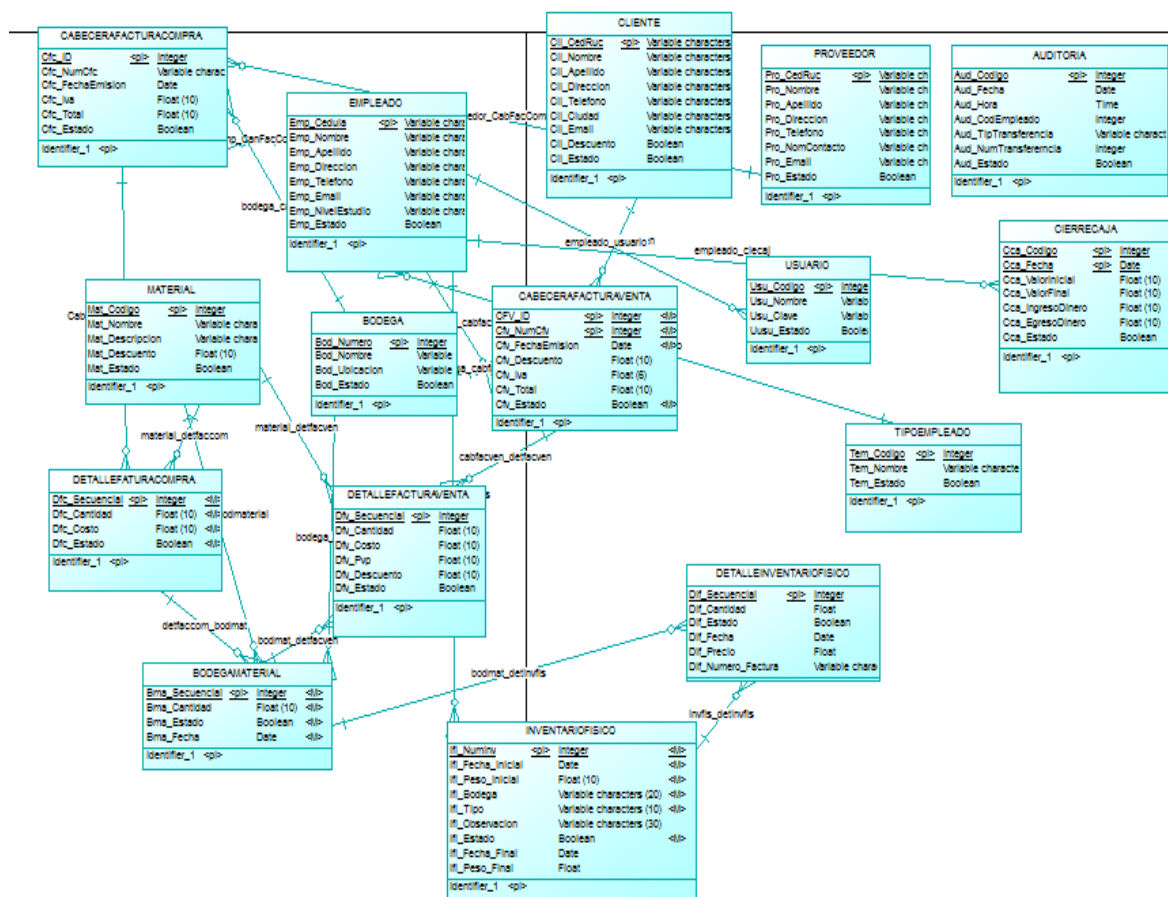




ANEXO N°17

MODELO CONCEPTUAL

MODELO CONCEPTUAL



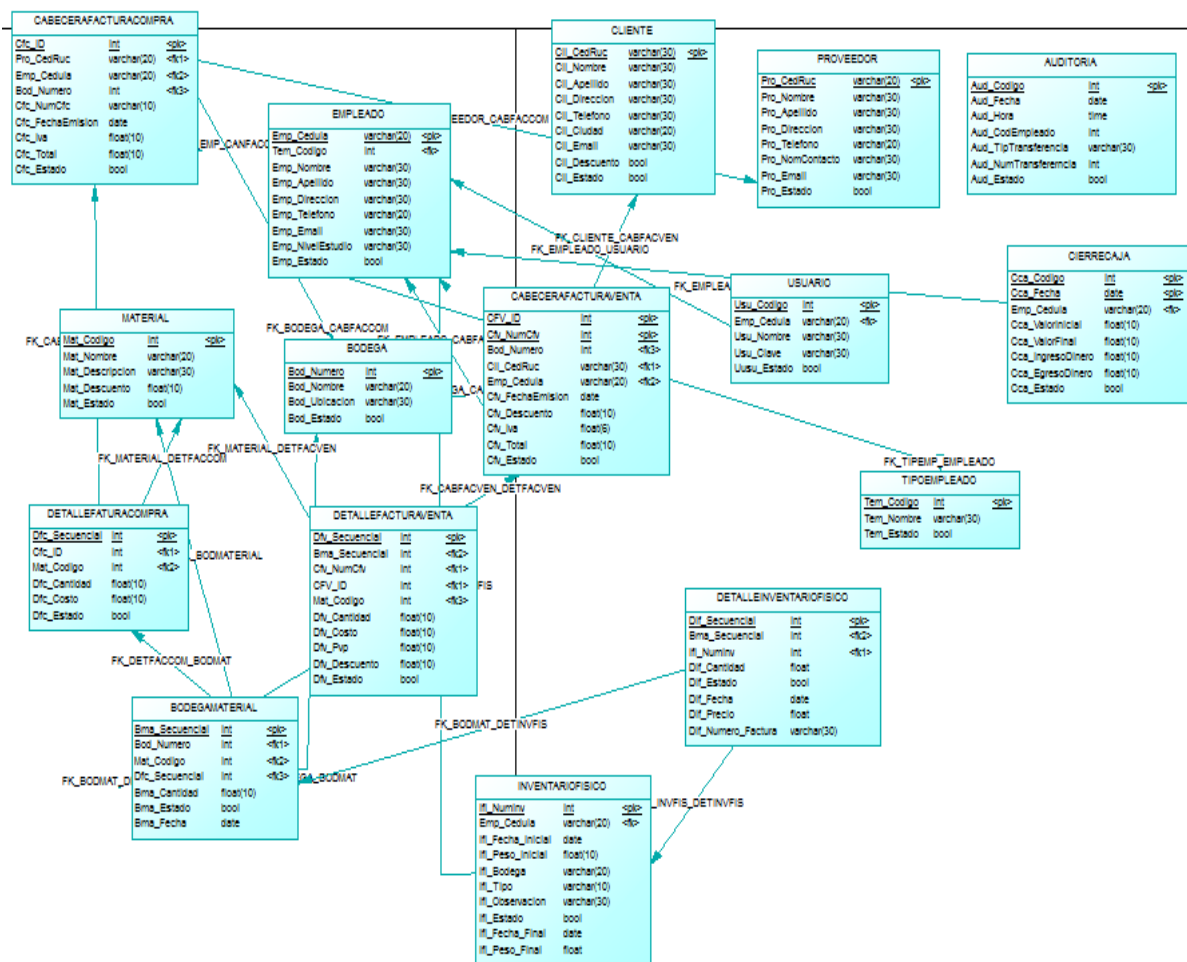


ANEXO N°18

MODELO FÍSICO



MODELO FÍSICO






















ANEXO N°19

CRONOGRAMA DE ACTIVIDADES



CRONOGRAMA DE ACTIVIDADES

| |  | Nombre de tarea | Comienzo | Fin |
|----|-------------------------------------------------------------------------------------|------------------------------------------|--------------------------------------------------------------------------------------------------|--------------|
| 1 |  | Propuesta del Proyecto a la Empresa FFHV | lun 25/10/10 | vie 29/10/10 |
| 2 |  | Recepcion de Información de la Empresa | lun 01/11/10 | mar 02/11/10 |
| 3 |  | Propuesta del Plan de Proyecto | mié 03/11/10 | vie 05/11/10 |
| 4 |  | Aprobación del Plan de Propuesta | mié 10/11/10 | mié 10/11/10 |
| 5 |  | Capítulo I | vie 26/11/10 | mié 29/12/10 |
| 6 |  | Capítulo II | vie 21/01/11 | vie 21/01/11 |
| 7 |  | Capítulo III | jue 03/02/11 | jue 03/02/11 |
| 8 |  | Pruebas y Depuración | lun 21/02/11 | mié 23/02/11 |
| 9 |  | Pruebas Finales | lun 28/02/11 | lun 28/02/11 |
| 10 |  | Implementación | mié 30/03/11 | mié 30/03/11 |
| 11 |  | Capítulo IV |  vie 04/03/11 | vie 04/03/11 |
| 12 |  | Capítulo V | jue 10/03/11 | jue 10/03/11 |
| 13 |  | Capítulo VI | dom 13/03/11 | dom 13/03/11 |
| 14 |  | Entrega definitiva del Proyecto | lun 04/04/11 | mié 06/04/11 |
| 15 |  | Sustentación Del Proyecto | lun 11/04/11 | mié 13/04/11 |







MANUAL DE USUARIO

INGRESO AL PROGRAMA

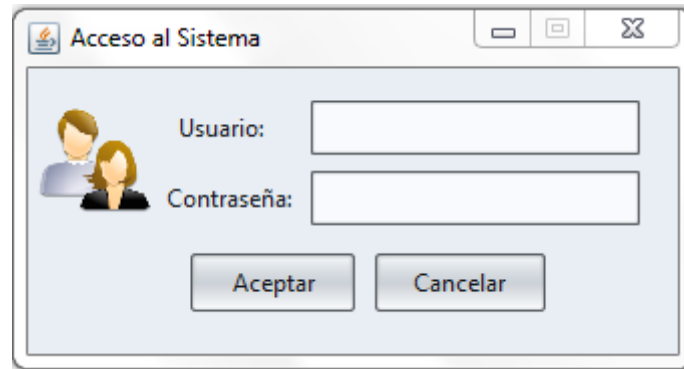
Para arrancar el programa, en el escritorio del computador se encontrará el siguiente ícono:



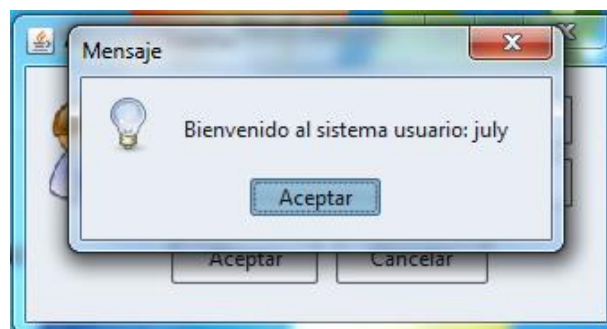
Haciendo doble click sobre él se despliega una pantalla automática, indicando el logo de la empresa, el nombre del programa y su respectivo autor como podemos observar en el siguiente gráfico:



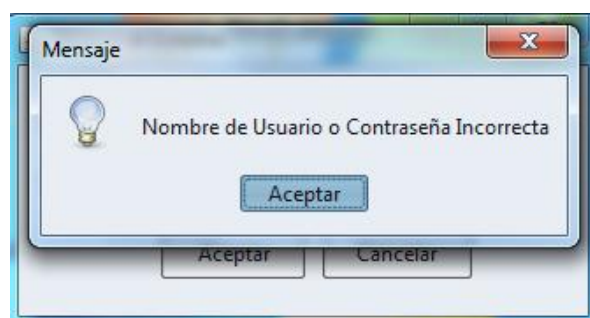
Luego de esto nos muestra la pantalla de acceso al sistema, como vemos a continuación:



En la cual tenemos que ingresar el nombre de usuario y la contraseña, si los datos ingresados son correctos aparece el mensaje de bienvenida con el nombre del usuario de la siguiente manera:



Caso contrario nos indicará que el nombre de usuario o contraseña es incorrecto, el mensaje de error es el siguiente:



Si fue posible el ingreso al programa, entonces aparece la pantalla del menú principal, donde podemos encontrar las principales funciones como son: Archivo, Mantenimiento,



Procesos, Reportes, Usuario.

Haciendo click en Archivo, se despliega la opción para salir de la aplicación en caso de que no sea requerida, como se puede apreciar en el gráfico:



Luego tenemos la opción de mantenimiento, donde podemos crear, guardar o eliminar un registro, haciendo click se despliegan diferentes opciones como son: Clientes, Proveedores, Bodegas, Materiales y Empleados como vemos en la figura:



Para dar mantenimiento a cualquiera de las opciones, damos un click sobre ellas e inmediatamente se desplegará la pantalla respectiva.

MANTENIMIENTO DE CLIENTES

A primera vista consta de un formulario activo en donde nos muestran todos los clientes



INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

que existen empresa detallada en una tabla en la parte inferior como se puede ver en el siguiente gráfico:

The screenshot shows a window titled 'Clientes'. At the top, there are three icons: a document, a green checkmark, and a red X. Below these are three buttons: 'Nuevo', 'Guardar', and 'Eliminar'. The form contains several input fields: 'Cédula/Ruc', 'Ciudad', 'Nombre', 'E-mail', 'Apellido', 'Teléfono', 'Dirección', and 'Estado'. Below the form is a table with the following data:

| Ced/Ruc | Nombre | Apellido | Dirección | Teléfono | Estado |
|------------|----------|----------|-------------|----------|--------|
| 1765739273 | Gabriela | Valdez | San Bartolo | 2345643 | A |
| 1715521454 | July | Cumbicus | Carapungo | 2222222 | A |
| 1134567894 | Luis | Montero | La Florida | 2345465 | I |

¿COMO CREAR UN NUEVO CLIENTE?

Para crear un nuevo cliente hacemos click en el botón Nuevo y procedemos a llenar los datos en las cajas de texto como se muestra a continuación:

The screenshot shows the same 'Clientes' window, but now the form is filled with data for a new client. The 'Nuevo' button is highlighted. The data entered is:

- Cédula/Ruc: 1728363922
- Ciudad: Quito
- Nombre: Daniel
- E-mail: danyLopez@hotmail.com
- Apellido: Lopez
- Teléfono: 2679323
- Dirección: El Inca
- Estado: ☒

The table below the form remains the same as in the previous screenshot.

En el campo Cédula/Ruc, se escribe el número de cédula del cliente o en caso de que se



trate de una empresa el número de RUC.

En el Nombre escribimos el nombre del cliente ya sea los dos o solo uno, como sea de preferencia.

En el Apellido, de la misma manera puede ser opcional escribir los dos apellidos.

En el campo Dirección anotamos el lugar donde vive el cliente.

El campo Teléfono es opcional ya que el cliente podría o no tener un número al cual localizarle.

En el campo Ciudad escribimos el lugar donde el cliente reside esta opción también es opcional.

En el campo E-mail se coloca una dirección de correo electrónico en caso de que el cliente la tuviera, no es obligatorio

En el estado se hace click en caso de que sea un cliente activo en la empresa o simplemente se deja inactivo si no lo es.

¿COMO GUARDAR UN CLIENTE?

Una vez que llenamos los datos del cliente, hacemos click en el botón guardar, nos aparecerá una pantalla indicando que los datos se han guardado correctamente o un mensaje de error en caso de que lo hubiera, ponemos aceptar y ya está creado nuestro registro, la información se desplegará inmediatamente en la tabla inferior.

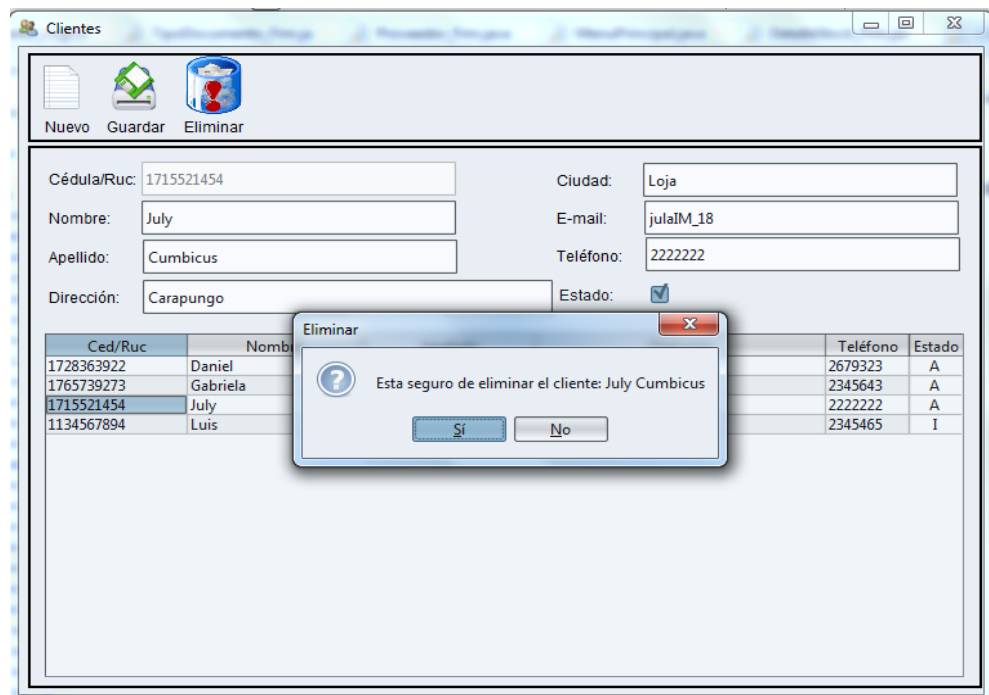
| Ced/Ruc | Nombre | Apellido | Dirección | Teléfono | Estado |
|------------|----------|----------|------------|----------|--------|
| 1728363922 | Daniel | Lopez | | 2679323 | A |
| 1765739273 | Gabriela | Valdez | | 2345643 | A |
| 1715521454 | July | Cumb | | 2222222 | A |
| 1134567894 | Luis | Montero | La Florida | 2345465 | I |



También haciendo click en el mismo botón guardar, se pueden actualizar los datos en el caso de que un registro ya exista ya sea porque no está escrito correctamente o por equivocación del usuario, o simplemente deseamos cambiar cierta información, hay que tomar en cuenta que cuando editamos un cliente, podemos hacerlo en todos los datos menos en la cédula.

¿COMO ELIMINAR UN CLIENTE?

Primero seleccionamos el cliente al cual deseamos eliminar y hacemos click en el botón eliminar. Seguido de esto nos aparecerá un cuadro de información en donde nos pregunta si realmente deseamos eliminar al registro, en caso de que sea SI hemos eliminado nuestro registro de la base de datos, caso contrario no se verá afectado.



🔗 **NOTA:** Hay que tomar en cuenta que no se puede eliminar un cliente que haga referencia de otro registro, el prototipo le informara que no se puede eliminar.



INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

En este caso para poder eliminar un cliente debemos primero eliminar los registros de los cuales depende antes de proceder a realizar tal acción.

MANTENIMIENTO DE PROVEEDORES

A primera vista consta de un formulario activo en donde nos muestran todos los proveedores que hay en la empresa detallada en una tabla en la parte inferior como se puede ver en el siguiente gráfico:

| Ced/Ruc | Nombre | Apellido | Dirección | Teléfono | Estado |
|------------|--------|----------|------------|----------|--------|
| 1723456784 | Andres | Garcia | La Marin | 2334343 | A |
| 1715521454 | July | Cumbicus | Carapungo | 2678931 | I |
| 1102264395 | Marco | Castillo | Calderon | 2344453 | I |
| 1792826713 | Rosa | Campos | La Florida | 2342434 | A |

¿COMO CREAR UN NUEVO PROVEEDOR?

Para crear un nuevo cliente hacemos click en el botón Nuevo y procedemos a llenar los datos en las cajas de texto como se muestra a continuación:



The screenshot shows a web application window titled "Proveedores". At the top, there are three icons: a document, a folder with a checkmark, and a trash can, labeled "Nuevo", "Guardar", and "Eliminar" respectively. Below these is a form with the following fields:

- Cédula/Ruc: 1725384922
- Teléfono: 2344552
- Nombre: Gabriel
- Contacto: (empty)
- Apellido: Mejia
- E-mail: (empty)
- Dirección: La planada
- Estado: ☒

Below the form is a table with the following data:

| Ced/Ruc | Nombre | Apellido | Dirección | Teléfono | Estado |
|------------|--------|----------|------------|----------|--------|
| 1723456784 | Andres | Garcia | La Marin | 2334343 | A |
| 1715521454 | July | Cumbicus | Carapungo | 2678931 | I |
| 1102264395 | Marco | Castillo | Calderon | 2344453 | I |
| 1792826713 | Rosa | Campos | La Florida | 2342434 | A |

En el campo Cédula/Ruc, se escribe el número de cédula del proveedor o en caso de que se trate de una empresa el número de RUC.

En el Nombre escribimos el nombre del cliente ya sea los dos o solo uno como sea de preferencia.

En el Apellido, de la misma manera puede ser opcional escribir los dos.

En el campo Dirección anotamos el lugar donde reside el proveedor.

El campo Teléfono es opcional ya que el proveedor podría o no tener un número al cual localizarle.

El campo Contacto un número de teléfono opcional donde se posible su localización.

En el E-mail se coloca una dirección de correo electrónico en caso de que el proveedor la tuviera, no es obligatorio.

En el estado se hace click en caso de que sea un proveedor activo en la empresa o simplemente se deja inactivo si no lo es.

¿COMO GUARDAR UN PROVEEDOR?

Una vez que llenamos los datos del proveedor, hacemos click en el botón guardar, nos



INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

aparecerá una pantalla indicando que los datos se han guardado correctamente o un mensaje de error en caso de que lo hubiera, ponemos aceptar y ya está creado nuestro registro, la información se desplegará inmediatamente en la tabla inferior.

The screenshot shows a window titled 'Proveedores'. At the top, there are three buttons: 'Nuevo' (with a document icon), 'Guardar' (with a floppy disk icon), and 'Eliminar' (with a trash can icon). Below these are input fields for 'Cédula/Ruc', 'Nombre', 'Apellido', 'Dirección', 'Teléfono', 'Contacto', 'E-mail', and 'Estado' (with a checkbox). A modal dialog box titled 'Proveedor' is centered on the screen, displaying a green checkmark icon and the text 'Datos guardados con éxito', with an 'Aceptar' button. Below the dialog, a table displays the following data:

| Ced/Ruc | Nombre | Dirección | Teléfono | Estado |
|------------|---------|-------------------|----------|--------|
| 1723456784 | Andres | | 2334343 | A |
| 1725384922 | Gabriel | | 2344552 | A |
| 1715521454 | July | | 2678931 | I |
| 1102264395 | Marco | | 2344453 | I |
| 1792826713 | Rosa | Campos La Florida | 2342434 | A |

También haciendo click en el mismo botón guardar, se pueden actualizar los datos en caso de que un registro ya exista, sea porque este no está escrito correctamente o por equivocación del usuario, o simplemente deseamos cambiar cierta información, hay que tomar en cuenta que cuando editamos un proveedor, podemos hacerlo en todos los datos menos en la cédula.

¿COMO ELIMINAR UN PROVEEDOR?

Primero seleccionamos el proveedor al cual deseamos eliminar y hacemos click en el botón eliminar. Seguido de esto nos aparecerá un cuadro de información en donde nos pregunta si realmente deseamos eliminar el proveedor, en caso de que sea SI hemos eliminado nuestro registro de la base de datos, caso contrario no se verá afectado.



Nuevo Guardar Eliminar

Cédula/Ruc: 1723456784 Teléfono: 2334343

Nombre: Andres Contacto: Julio Perez

Apellido: Garcia E-mail: ertwertwre

Dirección: La Marin

| Ced/Ruc | Nombre | Teléfono | Estado |
|------------|---------|----------|--------|
| 1723456784 | Andres | 2334343 | A |
| 1725384922 | Gabriel | 2344552 | A |
| 1715521454 | July | 2678931 | A |
| 1102264395 | Marco | 2344453 | A |
| 1792826713 | Rosario | 2342434 | A |

Eliminar

Esta seguro de eliminar el proveedor: Andres Garcia

Si No

🔔 **NOTA:** Hay que tomar en cuenta que no se puede eliminar un proveedor que haga referencia de otro registro, el prototipo le informara que no se puede eliminar. En este caso para poder eliminar un proveedor debemos primero eliminar los registros de los cuales depende antes de proceder a realizar tal acción.

MANTENIMIENTO DE BODEGAS

A primera vista consta de un formulario activo en donde nos muestran todas las bodegas que hay en la empresa detallada en una tabla en la parte inferior como se puede ver en el siguiente gráfico:



The screenshot shows the 'Bodegas' application window. At the top, there are three buttons: 'Nuevo' (New), 'Guardar' (Save), and 'Eliminar' (Delete). Below these buttons, there is a section labeled 'Bodega N°:' with three input fields: 'Nombre:', 'Ubicación:', and 'Estado:'. The 'Estado:' field has a checked checkbox. Below the input fields is a table with the following data:

| Nombre | Ubicación | Estado |
|------------|-----------|--------|
| BOD1 | Marin | A |
| BOD2 | Carolina | A |
| BOD3 | Guajalo | I |
| Cordillera | La Prensa | A |

¿COMO CREAR UNA NUEVA BODEGA?

Para crear una nueva bodega hacemos click en el botón Nuevo y procedemos a llenar los datos en las cajas de texto como se muestra a continuación:

The screenshot shows the 'Bodegas' application window with the 'Nuevo' button clicked. The input fields are now filled: 'Nombre:' contains 'BOD4' and 'Ubicación:' contains 'Av 24 de mayo'. The 'Estado:' field still has a checked checkbox. The table below the input fields remains the same as in the previous screenshot:

| Nombre | Ubicación | Estado |
|------------|-----------|--------|
| BOD1 | Marin | A |
| BOD2 | Carolina | A |
| BOD3 | Guajalo | I |
| Cordillera | La Prensa | A |

En el campo Bodega N° se cargará automáticamente, en el Nombre escribimos el nombre con el que se desea identificar a cada bodega, en el campo Ubicación el lugar donde se



encuentra actualmente y el estado se hace click en caso de que sea una bodega activa o simplemente se deja inactivo si no está en uso.

¿COMO GUARDAR UNA BODEGA?

Una vez que llenamos los datos de la bodega, hacemos click en el botón guardar, nos aparecerá una pantalla indicando que los datos se han guardado correctamente o un mensaje de error en caso de que lo hubiera, ponemos aceptar y ya está creado la nueva bodega la información se desplegará inmediatamente en la tabla inferior.

Bodegas

Nuevo Guardar Eliminar

Bodega N°:

Nombre:

Ubicación:

Estado: ☒

| Nombre | Estado |
|------------|--------|
| BOD1 | A |
| BOD2 | A |
| BOD3 | I |
| Cordillera | A |
| BOD4 | A |

La Prensa
Av 24 de mayo

Bodega

Datos guardados con éxito

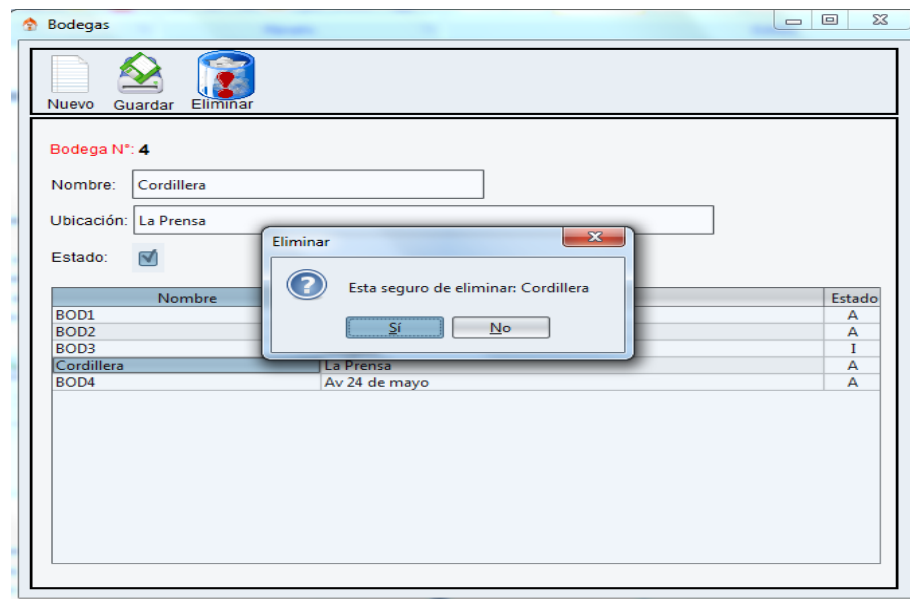
Aceptar

También haciendo click en el mismo botón guardar, se pueden actualizar los datos en caso de que un registro ya exista, sea porque este no está escrito correctamente o por equivocación del usuario, o simplemente deseamos cambiar cierta información, hay que tomar en cuenta que cuando editamos un proveedor, podemos hacerlo en todos los datos menos en la cédula.



¿COMO ELIMINAR UNA BODEGA?

Primero seleccionamos la bodega que deseamos eliminar y hacemos click en el botón eliminar. Seguido de esto nos aparecerá un cuadro de información en donde nos pregunta si realmente deseamos eliminar la bodega, en caso de que sea SI hemos eliminado nuestro registro de la base de datos, caso contrario no se verá afectado.



🔗 **NOTA:** Hay que tomar en cuenta que no se puede eliminar una bodega que haga referencia de otro registro, el prototipo le informara que no se puede eliminar.

En este caso para poder eliminar una bodega debemos primero eliminar los registros de los cuales depende antes de proceder a realizar dicha acción.

MANTENIMIENTO DE MATERIALES

A primera vista consta de un formulario activo en donde nos muestran todos los materiales que existen empresa detallados en una tabla en la parte inferior como se puede ver en el siguiente gráfico:



| Nombre | Descripción | Estado |
|------------------|---------------------------------------------------------------------------------|--------|
| Acero Inoxidable | material sólido | A |
| Aerosol | | A |
| Alambres | es un producto metalúrgico derivado de un proceso de laminación en caliente... | A |
| Aluminio | Mecánicamente es un material blando y maleable | A |
| Cobre con PVC | metal de transición blanco, brillante, blando, dúctil y maleable | A |
| Lata | envase opaco y resistente que resulta adecuado para envasar líquidos y produ... | A |
| Limalla | | A |

¿COMO CREAR UN NUEVO MATERIAL?

Para crear un nuevo material hacemos click en el botón Nuevo y procedemos a llenar los datos en las cajas de texto como se muestra a continuación:

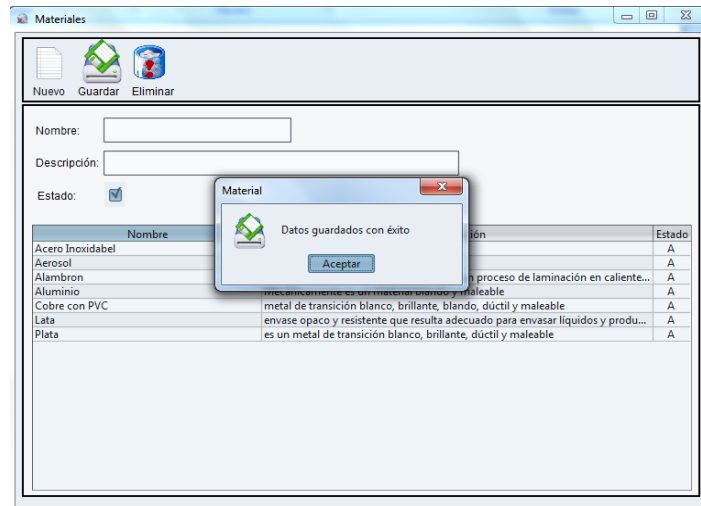
| Nombre | Descripción | Estado |
|------------------|---------------------------------------------------------------------------------|--------|
| Acero Inoxidable | material sólido | A |
| Aerosol | | A |
| Alambres | es un producto metalúrgico derivado de un proceso de laminación en caliente... | A |
| Aluminio | Mecánicamente es un material blando y maleable | A |
| Cobre con PVC | metal de transición blanco, brillante, blando, dúctil y maleable | A |
| Lata | envase opaco y resistente que resulta adecuado para envasar líquidos y produ... | A |
| Limalla | | A |

En el campo Nombre, se escribe el del material, en la Descripción ponemos algo característico que identifique a cada material, aunque el campo no es obligatorio y en el Estado se hace click en caso de que sea un material activo en la empresa o simplemente se deja inactivo si no se encuentra en el mercado.



¿COMO GUARDAR UN MATERIAL?

Una vez que llenamos los datos del material, hacemos click en el botón Guardar, nos aparecerá una pantalla indicando que los datos se han guardado correctamente o un mensaje de error en caso de que lo hubiera, ponemos aceptar y ya está creado nuestro registro, la información se desplegará inmediatamente en la tabla inferior.



También haciendo click en el botón Guardar se puede simplemente actualizar un material en caso de que el registro ya haya sido creado ya sea porque se ha ingresado mal algún dato, porque no está escrito correctamente o por equivocación del usuario, o simplemente deseamos cambiar cierta información.

Primero seleccionamos el cliente a actualizar haciendo click en cualquier fila de la tabla y automáticamente se cargaran los datos en las cajas de texto, que es donde podemos realizar cualquier cambio mas no se puede editar sobre la tabla, hacemos click en Guardar y ya tenemos actualizados los datos.

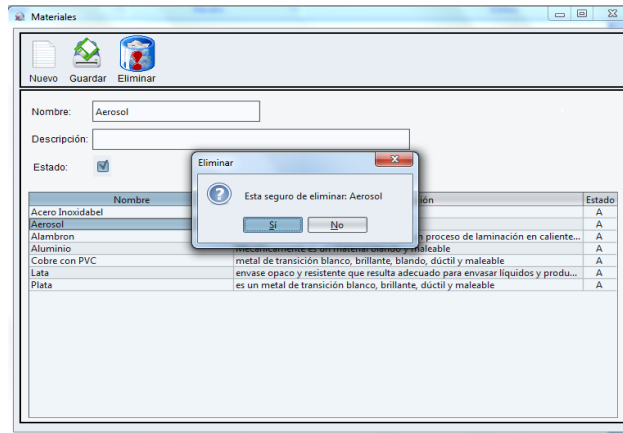
¿COMO ELIMINAR UN MATERIAL?

Primero seleccionamos el material al cual deseamos eliminar y hacemos click en el botón



INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

Eliminar. Seguido de esto nos aparecerá un cuadro de información en donde nos pregunta si realmente deseamos eliminar al registro, en caso de que sea SI hemos eliminado el registro de la base de datos, caso contrario no se verá afectado.



☞ **NOTA:** Hay que tomar en cuenta que no se puede eliminar un material que haga referencia de otro registro, el prototipo le informara que no se puede eliminar. En este caso para poder eliminar un material debemos primero eliminar los registros de los cuales depende antes de proceder a realizar dicha acción.

MANTENIMIENTO TIPO DE EMPLEADO

A primera vista consta de un formulario activo en donde nos muestran todos los tipos de empleados que hay en la empresa detallada en una tabla en la parte inferior como se puede ver en el siguiente gráfico:



The screenshot shows a window titled 'Tipo de Empleado'. At the top, there are three buttons: 'Nuevo', 'Guardar', and 'Eliminar'. Below these buttons, there is a text field labeled 'Nombre:' and a checkbox labeled 'Estado:' which is checked. Below the checkbox, there is a table with two columns: 'Nombre' and 'Estado'.

| Nombre | Estado |
|---------------|--------|
| Administrador | A |
| Bodeguero | A |
| Cajero | I |

¿COMO CREAR UN NUEVO TIPO DE EMPLEADO?

Para crear un nuevo tipo de empleado hacemos click en el botón Nuevo y procedemos a llenar los datos en las cajas de texto como se muestra a continuación:

The screenshot shows the same 'Tipo de Empleado' window. The 'Nombre' field now contains the text 'Secretaria'. The 'Estado' checkbox remains checked. The table below is the same as in the previous screenshot.

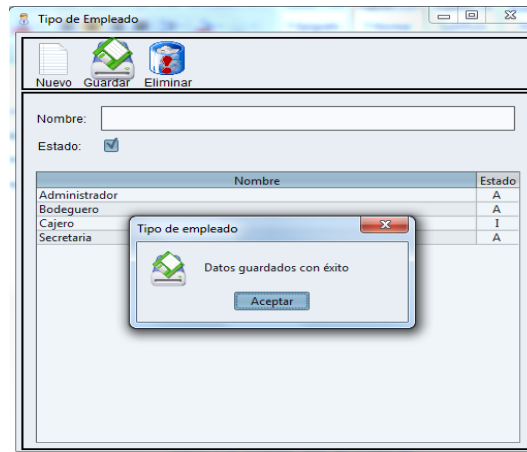
| Nombre | Estado |
|---------------|--------|
| Administrador | A |
| Bodeguero | A |
| Cajero | I |

En el campo Nombre escribimos el nuevo tipo de empleado que tendrá la empresa y en el Estado se hace click en caso de que sea un tipo de empleado que está activo o simplemente se deja inactivo si no lo es.

¿COMO GUARDAR UN TIPO DE EMPLEADO?



Una vez que llenamos los datos del tipo de empleado, hacemos click en el botón Guardar, donde aparece un mensaje indicando que los datos se han guardado correctamente o un mensaje de error en caso de que lo hubiera, ponemos aceptar y ya está creado nuestro registro, la información se desplegará inmediatamente en la tabla inferior.



También haciendo click en el botón Guardar se puede actualizar un tipo de empleado cuando por algún motivo hemos ingresado mal algún dato, porque no está escrito correctamente o por equivocación del usuario, o simplemente deseamos cambiar cierta información.

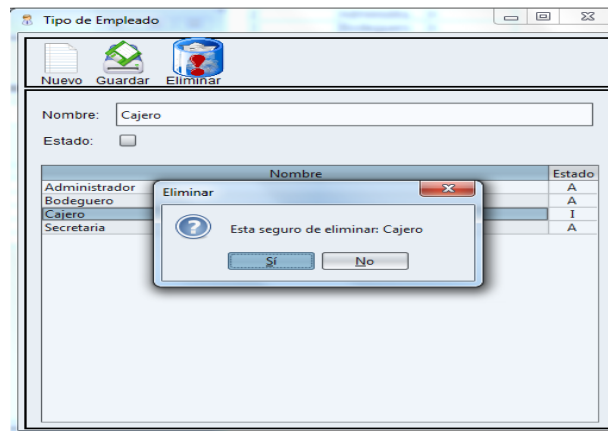
Primero seleccionamos el tipo de empleado a actualizar haciendo click en cualquier fila de la tabla y automáticamente se cargaran los datos en las cajas de texto, que es donde podemos realizar cualquier cambio, mas no se puede editar sobre la tabla, hacemos click en el botón Guardar y ya tenemos actualizados los datos.

¿COMO ELIMINAR UN TIPO DE EMPLEADO?

Primero seleccionamos el tipo de empleado que deseamos eliminar y hacemos click en el botón eliminar. Seguido de esto nos aparecerá un cuadro de información en donde nos pregunta si realmente deseamos eliminar el tipo de empleado, en caso de que sea SI



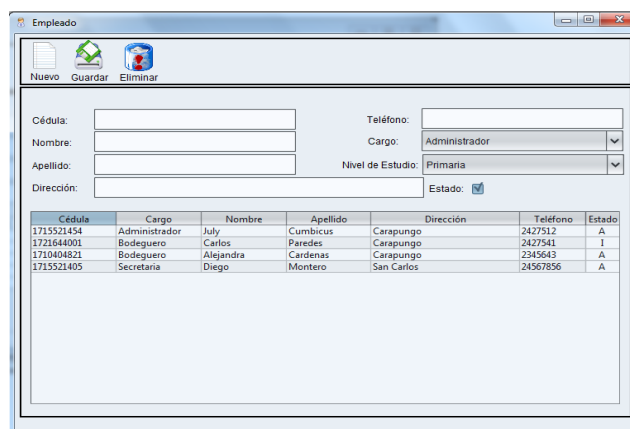
hemos eliminado nuestro registro de la base de datos, caso contrario no se verá afectado.



☞ **NOTA:** Hay que tomar en cuenta que no se puede eliminar un tipo de empleado que haga referencia de otro registro, el prototipo le informara que no se puede eliminar. En este caso para poder eliminar un tipo de empleado debemos primero eliminar los registros de los cuales depende antes de proceder a realizar tal acción.

MANTENIMIENTO DE EMPLEADOS

A primera vista consta de un formulario activo en donde nos muestran todos los empleados que trabajan en la empresa detallada en una tabla en la parte inferior como se puede ver en el siguiente gráfico:





¿COMO CREAR UN NUEVO EMPLEADO?

Para crear un nuevo empleado hacemos click en el botón Nuevo y procedemos a llenar los datos en las cajas de texto como se muestra a continuación:

The screenshot shows a window titled 'Empleado' with a toolbar containing 'Nuevo', 'Guardar', and 'Eliminar' buttons. The form contains the following fields:

- Cédula: 1102264395
- Nombre: Catalina
- Apellido: Montero
- Dirección: El Bosque
- Teléfono: 2456788
- Cargo: Secretaria
- Nivel de Estudio: Superior
- Estado: ☒

Below the form is a table with the following data:

| Cédula | Cargo | Nombre | Apellido | Dirección | Teléfono | Estado |
|------------|---------------|-----------|----------|------------|----------|--------|
| 1715521454 | Administrador | July | Cumbicus | Carapungo | 2427512 | A |
| 1721644001 | Bodeguero | Carlos | Paredes | Carapungo | 2427341 | I |
| 1710404821 | Bodeguero | Alejandra | Cardenas | Carapungo | 2345643 | A |
| 1715521405 | Secretaria | Diego | Montero | San Carlos | 24567856 | A |

En el campo Cédula, se escribe el número de cédula del empleado, en el Nombre escribimos el nombre del empleado ya sea los dos o solo uno, como sea de preferencia, en el Apellido, de la misma manera puede ser opcional escribir los dos apellidos, en el campo Dirección anotamos el lugar donde vive el empleado, en el campo Teléfono un número en el cual localizarle, en Nivel de Estudio se escoge el nivel hasta el cual llego dicha persona, luego en el Cargo se puede escoger el lugar de trabajo que ocupará el empleado dentro de la empresa y el Estado se deja inactivo en caso de que dicha persona ya no trabaje en la empresa caso contrario estará activa haciendo click en el botón.

¿COMO GUARDAR UN EMPLEADO?

Una vez que llenamos los datos del empleado, hacemos click en el botón guardar, nos aparecerá una pantalla indicando que los datos se han guardado correctamente o un mensaje de error en caso de que lo hubiera, ponemos aceptar y ya está creado nuestro registro, la información se desplegará inmediatamente en la tabla inferior.



The screenshot shows a web application window titled 'Empleado'. At the top, there are three buttons: 'Nuevo', 'Guardar', and 'Eliminar'. Below these are input fields for 'Cédula', 'Nombre', 'Apellido', 'Dirección', 'Teléfono', 'Cargo', and 'Estado'. A modal dialog box is open in the center with the message 'Datos guardados con éxito' and an 'Aceptar' button. Below the form is a table with the following data:

| Cédula | Cargo | Nombre | Apellido | Dirección | Teléfono | Estado |
|------------|---------------|----------|----------|------------|----------|--------|
| 1715521454 | Administrador | Julio | Montero | San Carlos | 2427512 | A |
| 1721644001 | Bodeguero | Calisto | Montero | El Bosque | 2427541 | I |
| 1710404821 | Bodeguero | Alfonso | Montero | San Carlos | 2345643 | A |
| 1715521405 | Secretaria | Diego | Montero | San Carlos | 24567856 | A |
| 1102264395 | Secretaria | Catalina | Montero | El Bosque | 2456788 | A |

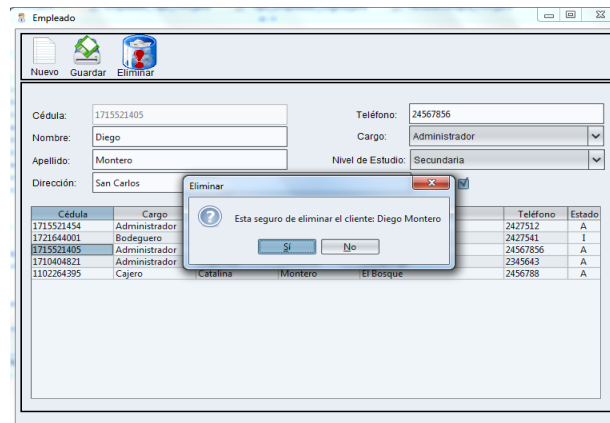
¿COMO ACTUALIZAR UN EMPLEADO?

También haciendo click en el botón guardar se puede actualizar los datos de un empleado ya sea porque se ha ingresado mal algún dato, porque no está escrito correctamente o por equivocación del usuario, o simplemente deseamos cambiar cierta información, hay que tomar en cuenta que cuando editamos un empleado, podemos hacer cambios en todos los datos menos en la cédula.

Primero seleccionamos el empleado a actualizar haciendo click en cualquier fila de la tabla y automáticamente se cargaran los datos en las cajas de texto, que es donde podemos realizar cualquier cambio, mas no se puede editar sobre la tabla, hacemos click en el botón Guardar y ya están realizados los cambios respectivos, aparece el mensaje correspondiente.

¿COMO ELIMINAR UN EMPLEADO?

Primero seleccionamos el empleado que deseamos eliminar y hacemos click en el botón eliminar. Seguido de esto nos aparecerá un cuadro de información en donde nos pregunta si realmente deseamos eliminar el empleado, en caso de que sea SI hemos eliminado nuestro registro de la base de datos, caso contrario no se verá afectado.



☞ **NOTA:** Hay que tomar en cuenta que no se puede eliminar un tipo de empleado que haga referencia de otro registro, el prototipo le informara que no se puede eliminar. En este caso para poder eliminar un empleado debemos primero eliminar los registros de los cuales depende antes de proceder a realizar dicha acción

PROCESOS

Para ingresar a los procesos que se manejan en el programa, vamos al Menú principal la opción Procesos y ahí se despliegan las diferentes funcionalidades que son: Compras, Ventas e Inventario, como podemos apreciar en el siguiente gráfico:



Haciendo un click en Compras se despliega la opción Factura de Compra, como vemos en el gráfico:



Haciendo click en Factura de Compra inmediatamente nos muestra la pantalla de Factura de Compra que es donde se ingresarán todas las compras realizadas, la figura se muestra de la siguiente manera:

| Eliminar | Peso | Material | Precio | Subtotal |
|----------|------|----------|--------|----------|
|----------|------|----------|--------|----------|

Haciendo un click en Ventas se despliega la opción Factura de Venta, como vemos en el gráfico:



Damos click en Factura de Venta e inmediatamente nos muestra la pantalla de Factura de



INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

Venta que es donde se ingresarán todas las ventas realizadas, la figura se muestra de la siguiente manera:

| Material | Precio de Compra | Peso | IVA | Subtotal |
|----------|------------------|------|-----|----------|
| Bronce | | | | |

FACTURA DE COMPRA

Dentro de esta pantalla tenemos los siguientes botones como son: Nuevo, Guardar y Nuevo Proveedor.

Haciendo click en Nuevo podemos crear una nueva factura de compra.

Cuando se han llenado los datos, tanto de la cabecera de la factura de compra como del detalle, se hace click en el botón Guardar.

En la cabecera de la factura de compra se selecciona el nombre del proveedor y se cargarán automáticamente todos sus datos, en caso de que el proveedor no exista se hace click en el botón Nuevo Proveedor e inmediatamente se carga la pantalla de mantenimiento del Proveedor que se explico anteriormente.

En la opción Bodega, podemos seleccionar el nombre de la bodega en la cual vamos a ingresar el material comprado.

En el detalle se selecciona el nombre del material, se ingresa tanto el precio como el peso y seleccionamos si se cobra IVA o no, hacemos click en ingresar y automáticamente se cargaran los datos en la tabla y se realizarán todos los cálculos automáticamente, obteniéndose así los subtotales, el IVA cobrado y el total a pagar, en caso de que se ingrese mal un detalle, se hace click en la tabla en el botón eliminar, si todo es correcto hacemos click en el botón Guardar y ya tenemos creada la factura de compra con su



INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

respectiva serie, en el grafico se puede apreciar una factura con todos los datos ingresados:

Factura de Compra

Nuevo Guardar Nuevo Proveedor

Proveedor: Carlos Andrade Bodega: BOD1 Tipo Documento: fadsv
Cedula o RUC: 1721544001 Fecha: 2011-03-23
Dirección: Calderón **Factura No: 001-001-001**

Material: Acero Inoxidable Precio de Compra: Peso: IVA: ☐ Insertar

| Eliminar | Peso | Material | Precio | Subtotal |
|-------------------------------------|------|------------------|--------|----------|
| <input checked="" type="checkbox"/> | 3.4 | Bronce | 1.2 | 4.08 |
| <input checked="" type="checkbox"/> | 4.0 | Cobre | 3.0 | 12.0 |
| <input checked="" type="checkbox"/> | 3.0 | Acero Inoxidable | 2.0 | 6.0 |

Subtotal: 22.08
Iva12%: 1.93
Total: 24.01

FACTURA DE VENTA

Dentro de esta pantalla tenemos los siguientes botones como son: Nuevo, Guardar y Nuevo Cliente.

Haciendo click en Nuevo podemos crear una nueva factura de venta.

Cuando se han llenado los datos, tanto de la cabecera de la factura de venta como del detalle, se hace click en el botón Guardar.

En la cabecera de la factura de venta se selecciona el nombre del cliente y se cargarán automáticamente todos sus datos, en caso de que el cliente no exista se hace click en el botón Nuevo Cliente e inmediatamente se carga la pantalla de mantenimiento del Cliente que se explico anteriormente.

En la opción Bodega, podemos seleccionar el nombre de la bodega en la cual vamos a ingresar el material a vender.

De acuerdo a la bodega que sea seleccionada se cargarán los materiales en la factura, al escoger cualquiera de ellos, automáticamente se cargará el precio de venta respectivo,



INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

para lo cual solo ingresamos el peso y seleccionamos si cobramos IVA o no por dicho material, todos los cálculos realizan automáticamente, obteniéndose así los subtotales, el IVA cobrado y el total a cancelar, en caso de que se ingrese mal un detalle, se hace click en la tabla en el botón eliminar, si todo es correcto hacemos click en el botón Guardar y ya tenemos creada nuestra factura con su respectiva serie, en el grafico se puede apreciar una factura de venta con todos los datos ingresados:

Factura de Venta

Nuevo Guardar Nuevo Cliente

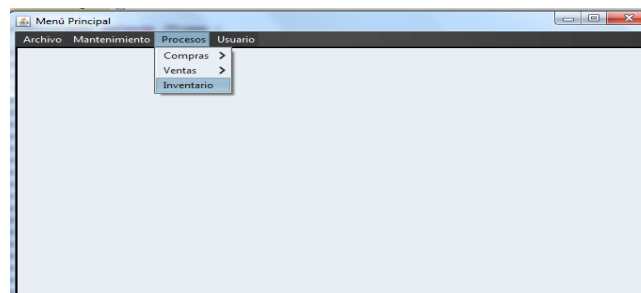
Cliente: Rosa Campos Bodega: BOD1 Tipo Documento: fadsv
Cedula o RUC: 1715521454 Fecha: 2011-03-23
Dirección: La Florida **Factura No: 001-001-001**

Material: Acero Inoxidable Precio de Compra: Peso: IVA: ☐ Insertar

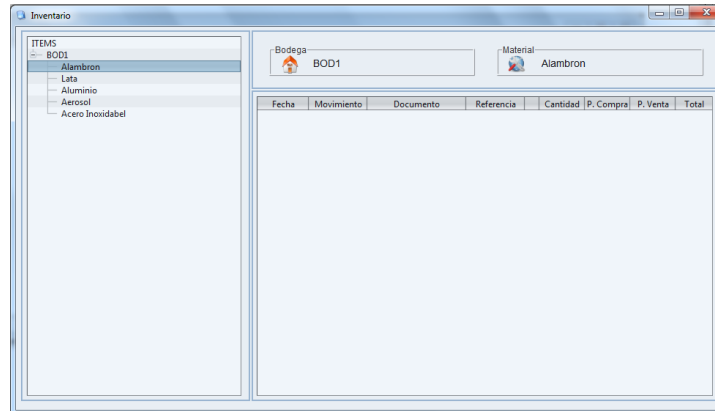
| Eliminar | Peso | Material | Precio | Subtotal |
|-------------------------------------|------|------------------|--------|----------|
| <input checked="" type="checkbox"/> | 2.5 | Bronce | 1.25 | 3.125 |
| <input checked="" type="checkbox"/> | 4.0 | Plata | 1.8 | 7.2 |
| <input checked="" type="checkbox"/> | 3.0 | Cobre | 1.5 | 4.5 |
| <input checked="" type="checkbox"/> | 5.0 | Acero Inoxidable | 2.0 | 10.0 |

Subtotal: 24.82
Iva12%: 1.4
Total: 26.23

Haciendo click en la pestaña procesos y seleccionando la opción Inventario como se puede apreciar en el gráfico:



Luego se despliega la siguiente la siguiente pantalla:



En esta ventana al lado izquierdo, haciendo click en ITEMS, podemos encontrar todas las bodegas que se encuentran activas en la empresa, haciendo click en cualquiera de ellas se despliega la lista de materiales que existe en cada una de ellas. Si Damos click en cualquiera de los materiales al lado derecho se cargarán los detalles de ese material, es decir todos los movimientos que se le ha dado.

REPORTES

Para realizar, los diferentes reportes hacemos click en la celda Reportes del Menú Principal y a continuación se despliega la lista de reportes que se podemos realizar.

CREAR UNA CUENTA

Haciendo click en la celda Usuario del Menú Principal, se puede crear una de usuario para poder tener acceso al programa, cabe recalcar que esta celda solo estará activa para el gerente, la pantalla que se muestra es la siguiente:



| Perfil | Empleado | Usuario | Clave | Estado |
|---------------|------------|---------|----------|--------|
| Gerente | 1715521454 | july | 8aãÖeÖiE | ACTIVO |
| Administrador | 1715521405 | diego | iiiÖÖ | ACTIVO |

Primero seleccionamos el nombre real con el que trabajamos en la empresa, en el Nombre de Usuario se escribe un identificativo cualquiera que sea fácil de recordar, también seleccionamos que perfil le daremos a este usuario, luego se ingresa la contraseña, cabe recalcar que solo se podrán ingresar letras mas no números, además se tiene que hacer una confirmación de la contraseña para saber que el usuario está escribiendo correctamente su password, se hace click en estado activo si es un usuario que hace uso del programa o inactivo si simplemente no lo utiliza. Si los datos son ingresados correctamente aparecerá el mensaje de confirmación, caso contrario aparecerán los mensajes de advertencia dependiendo del error que se cometa.





MANUAL TÉCNICO

Para la programación del software, se dividió en cuatro paquetes principales como son:

- Acceso Datos
- Entidades
- Lógica
- FacturacionInventario

Acceso Datos

En el paquete Acceso Datos se encuentra la clase Conexión, que nos servirá para conectar el aplicativo con la base de datos MySQL, a continuación escribo el código para esta clase:

```
package AccesoDatos;

import java.sql.*;

public class Conexion {

    // Objetos Tipo Sql

    public Statement st = null;
    public PreparedStatement ps = null;
    public ResultSet rs = null;

    public Connection con;

    public Connection recuperaConexion() {

        return (this.con);
    }

    public Connection abrirConexion() {

        try {

            String userName = "root";
```



```
String password = "";
String url = "jdbc:mysql://localhost:3306/bdfacturacioninventario";
Class.forName("com.mysql.jdbc.Driver").newInstance();
con = DriverManager.getConnection(url, userName, password);
// System.out.println("Conexion a la base de datos");
return con;

} catch (Exception e) {

    System.out.println(e.getMessage());

    System.out.println("Error en conexión ");
    return con;
}

}

public void cerrarConexion() {

    try {

        this.con.close();

    } catch (Exception e) {

        System.out.println("La conexion ya esta cerrada");

    }

}

public static void main(String[] args) {
    Conexion con = new Conexion()
    con.cerrarConexion();
}

}
```

Entidades

En el paquete entidades se crean las diferentes clases en donde se encuentran los atributos y funciones que tiene cada una de ellas. A continuación escribo el código por cada clase creada:



- **Clase:** BodegaMaterial.java

```
package Entidades;
import java.util.Date;

public class BodegaMaterial {

    int bma_secuencial =0;
    int bod_numero = 0;
    int mat_codigo = 0;
    int dfc_secuencial =0;
    double bma_cantidad =0;
    Date bma_fecha;
    boolean bma_estado =false;
    public int getBma_secuencial() {
        return bma_secuencial;
    }
    public void setBma_secuencial(int bmaSecuencial) {
        bma_secuencial = bmaSecuencial;
    }
    public int getBod_numero() {
        return bod_numero;
    }
    public void setBod_numero(int bodNumero) {
        bod_numero = bodNumero;
    }
    public int getMat_codigo() {
        return mat_codigo;
    }
    public void setMat_codigo(int matCodigo) {
        mat_codigo = matCodigo;
    }
    public int getDfc_secuencial() {
        return dfc_secuencial;
    }
    public void setDfc_secuencial(int dfcSecuencial) {
        dfc_secuencial = dfcSecuencial;
    }
    public double getBma_cantidad() {
        return bma_cantidad;
    }
    public void setBma_cantidad(double bmaCantidad) {
        bma_cantidad = bmaCantidad;
    }
    public Date getBma_fecha() {
```



```
        return bma_fecha;
    }
    public void setBma_fecha(Date bmaFecha) {
        bma_fecha = bmaFecha;
    }
    public boolean isBma_estado() {
        return bma_estado;
    }
    public void setBma_estado(boolean bmaEstado) {
        bma_estado = bmaEstado;
    }

    public BodegaMaterial() {
    }

    public BodegaMaterial(int bmaSecuencial, int bodNumero, int matCodigo,
        int dfcSecuencial, double bmaCantidad, Date bmaFecha,
        boolean bmaEstado) {
        super();
        bma_secuencial = bmaSecuencial;
        bod_numero = bodNumero;
        mat_codigo = matCodigo;
        dfc_secuencial = dfcSecuencial;
        bma_cantidad = bmaCantidad;
        bma_fecha = bmaFecha;
        bma_estado = bmaEstado;}}

```

- **Clase:** CabeceraFacturaCompra.java

```
package Entidades;

import java.util.Date;

public class CabeceraFacturaCompra {

    int cfc_id =0;
    String pro_CedRuc = "";
    String emp_cedula = "";
    int bod_numero = 0;
    String cfc_numCfc = "";
    Date cfc_fechaEmision;
    double cfc_iva =0;
    double cfc_total =0;
    boolean cfc_estado =false;
}

```




```
int tdo_secuencial=0;
```

```
public int getTdo_secuencial() {  
    return tdo_secuencial;  
}  
public void setTdo_secuencial(int tdoSecuencial) {  
    tdo_secuencial = tdoSecuencial;  
}  
public int getCfc_id() {  
    return cfc_id;  
}  
public void setCfc_id(int cfclId) {  
    cfc_id = cfclId;  
}  
public String getPro_CedRuc() {  
    return pro_CedRuc;  
}  
public void setPro_CedRuc(String proCedRuc) {  
    pro_CedRuc = proCedRuc;  
}  
public String getEmp_cedula() {  
    return emp_cedula;  
}  
public void setEmp_cedula(String empCedula) {  
    emp_cedula = empCedula;  
}  
public int getBod_numero() {  
    return bod_numero;  
}  
public void setBod_numero(int bodNumero) {  
    bod_numero = bodNumero;  
}  
public String getCfc_numCfc() {  
    return cfc_numCfc;  
}  
public void setCfc_numCfc(String cfcNumCfc) {  
    cfc_numCfc = cfcNumCfc;  
}  
public Date getCfc_fechaEmision() {  
    return cfc_fechaEmision;  
}  
public void setCfc_fechaEmision(Date cfcFechaEmision) {  
    cfc_fechaEmision = cfcFechaEmision;  
}
```



```
}  
public double getCfc_iva() {  
    return cfc_iva;  
}  
public void setCfc_iva(double cfcIva) {  
    cfc_iva = cfcIva;  
}  
public double getCfc_total() {  
    return cfc_total;  
}  
public void setCfc_total(double cfcTotal) {  
    cfc_total = cfcTotal;  
}  
public boolean isCfc_estado() {  
    return cfc_estado;  
}  
public void setCfc_estado(boolean cfcEstado) {  
    cfc_estado = cfcEstado;  
}  
  
public CabeceraFacturaCompra() {  
  
}  
  
public CabeceraFacturaCompra(int cfclId, String proCedRuc, String  
empCedula,  
                                int bodNumero, String cfcNumCfc, Date cfcFechaEmision,  
                                double cfcIva, double cfcTotal, boolean cfcEstado, int  
tdoSecuencial) {  
    super();  
    cfc_id = cfclId;  
    pro_CedRuc = proCedRuc;  
    emp_cedula = empCedula;  
    bod_numero = bodNumero;  
    cfc_numCfc = cfcNumCfc;  
    cfc_fechaEmision = cfcFechaEmision;  
    cfc_iva = cfcIva;  
    cfc_total = cfcTotal;  
    cfc_estado = cfcEstado;  
    tdo_secuencial = tdoSecuencial;  
}
```

- **Clase:** CabeceraFacturaVenta.java



```
package Entidades;

import java.util.Date;

public class CabeceraFacturaVenta {
    int cfv_id=0;
    String cfv_numcfv="";
    int bod_numero=0;
    String cli_cedruc="";
    String emp_cedula="";
    Date cfv_fechaemision;
    double cfv_iva=0;
    double cfv_total=0;
    boolean cfv_estado=false;
    public int getCfv_id() {
        return cfv_id;
    }
    public void setCfv_id(int cfvId) {
        cfv_id = cfvId;
    }
    public String getCfv_numcfv() {
        return cfv_numcfv;
    }
    public void setCfv_numcfv(String cfvNumcfv) {
        cfv_numcfv = cfvNumcfv;
    }
    public int getBod_numero() {
        return bod_numero;
    }
    public void setBod_numero(int bodNumero) {
        bod_numero = bodNumero;
    }
    public String getCli_cedruc() {
        return cli_cedruc;
    }
    public void setCli_cedruc(String cliCedruc) {
        cli_cedruc = cliCedruc;
    }
    public String getEmp_cedula() {
        return emp_cedula;
    }
    public void setEmp_cedula(String empCedula) {
        emp_cedula = empCedula;
    }
}
```



```
    public Date getCfv_fechaemision() {
        return cfv_fechaemision;
    }
    public void setCfv_fechaemision(Date cfvFechaemision) {
        cfv_fechaemision = cfvFechaemision;
    }
    public double getCfv_iva() {
        return cfv_iva;
    }
    public void setCfv_iva(double cfvIva) {
        cfv_iva = cfvIva;
    }
    public double getCfv_total() {
        return cfv_total;
    }
    public void setCfv_total(double cfvTotal) {
        cfv_total = cfvTotal;
    }
    public boolean isCfv_estado() {
        return cfv_estado;
    }
    public void setCfv_estado(boolean cfvEstado) {
        cfv_estado = cfvEstado;
    }
    public CabeceraFacturaVenta(){}
    public CabeceraFacturaVenta(int cfvId, String cfvNumcfv, int bodNumero,
        String cliCedruc, String empCedula, Date cfvFechaemision,
        double cfvIva, double cfvTotal, boolean cfvEstado) {
        super();
        cfv_id = cfvId;
        cfv_numcfv = cfvNumcfv;
        bod_numero = bodNumero;
        cli_cedruc = cliCedruc;
        emp_cedula = empCedula;
        cfv_fechaemision = cfvFechaemision;
        cfv_iva = cfvIva;
        cfv_total = cfvTotal;
        cfv_estado = cfvEstado;
    }
}
```

- **Clase:** CabeceralInventario.java

package Entidades;



```
import java.util.Date;

public class DetalleInventario {

    int dif_secuencial =0;
    int ifi_numinv =0;
    double dif_cantidad =0;
    boolean dif_estado = false;
    Date dif_fecha;
    double dif_precio =0;
    String dif_numero_fac ="";

    public DetalleInventario(){}

    public int getDif_secuencial() {
        return dif_secuencial;
    }

    public void setDif_secuencial(int difSecuencial) {
        dif_secuencial = difSecuencial;
    }

    public int getIfi_numinv() {
        return ifi_numinv;
    }

    public void setIfi_numinv(int ifiNuminv) {
        ifi_numinv = ifiNuminv;
    }

    public double getDif_cantidad() {
        return dif_cantidad;
    }

    public void setDif_cantidad(double difCantidad) {
        dif_cantidad = difCantidad;
    }
}
```



```
public boolean isDif_estado() {  
    return dif_estado;  
}
```

```
public void setDif_estado(boolean difEstado) {  
    dif_estado = difEstado;  
}
```

```
public Date getDif_fecha() {  
    return dif_fecha;  
}
```

```
public void setDif_fecha(Date difFecha) {  
    dif_fecha = difFecha;  
}
```

```
public double getDif_precio() {  
    return dif_precio;  
}
```

```
public void setDif_precio(double difPrecio) {  
    dif_precio = difPrecio;  
}
```

```
public String getDif_numero_fac() {  
    return dif_numero_fac;  
}
```

```
public void setDif_numero_fac(String difNumeroFac) {  
    dif_numero_fac = difNumeroFac;  
}
```

```
public DetalleInventario(int difSecuencial, int ifiNuminv, double difCantidad,  
    boolean difEstado, Date difFecha, double difPrecio, String  
    difNumeroFac) {
```



```
super();
dif_secuencial = difSecuencial;
ifi_numinv = ifiNuminv;
dif_cantidad = difCantidad;
dif_estado = difEstado;
dif_fecha = difFecha;
dif_precio = difPrecio;
dif_numero_fac = difNumeroFac;
}

}
```

Clase: DetalleFacturaCompra.java

```
package Entidades;
public class DetalleFacturaCompra {

    int dfc_secuencial =0;
    int cfc_id = 0;
    int mat_codigo = 0;
    double dfc_cantidad =0;
    double dfc_costo =0;
    boolean dfc_estado =false;

    public int getDfc_secuencial() {
        return dfc_secuencial;
    }
    public void setDfc_secuencial(int dfcSecuencial) {
        dfc_secuencial = dfcSecuencial;
    }
    public int getCfc_id() {
        return cfc_id;
    }
    public void setCfc_id(int cfclId) {
        cfc_id = cfclId;
    }
    public int getMat_codigo() {
        return mat_codigo;
    }
    public void setMat_codigo(int matCodigo) {
        mat_codigo = matCodigo;
    }
}
```



```
public double getDfc_cantidad() {  
    return dfc_cantidad;  
}  
public void setDfc_cantidad(double dfcCantidad) {  
    dfc_cantidad = dfcCantidad;  
}  
public double getDfc_costo() {  
    return dfc_costo;  
}  
public void setDfc_costo(double dfcCosto) {  
    dfc_costo = dfcCosto;  
}  
public boolean isDfc_estado() {  
    return dfc_estado;  
}  
public void setDfc_estado(boolean dfcEstado) {  
    dfc_estado = dfcEstado;  
}  
  
public DetalleFacturaCompra() {  
  
}  
  
public DetalleFacturaCompra(int dfcSecuencial, int cfclId, int matCodigo,  
    double dfcCantidad, double dfcCosto, boolean dfcEstado) {  
    super();  
    dfc_secuencial = dfcSecuencial;  
    cfc_id = cfclId;  
    mat_codigo = matCodigo;  
    dfc_cantidad = dfcCantidad;  
    dfc_costo = dfcCosto;  
    dfc_estado = dfcEstado;  
}  
  
}
```

- **Clase:** DetalleFacturaVenta.java

```
package Entidades;
```

```
public class DetalleFacturaVenta {  
    int dfv_secuencial=0;  
    int cfv_id=0;
```




```
int mat_codigo=0;
double dfv_cantidad=0;
double dfv_pvp=0;
double dfv_descuento=0;
boolean dfv_estado = false;
public int getDfv_secuencial() {
    return dfv_secuencial;
}
public void setDfv_secuencial(int dfvSecuencial) {
    dfv_secuencial = dfvSecuencial;
}
public int getCfv_id() {
    return cfv_id;
}
public void setCfv_id(int cfvId) {
    cfv_id = cfvId;
}
public int getMat_codigo() {
    return mat_codigo;
}
public void setMat_codigo(int matCodigo) {
    mat_codigo = matCodigo;
}
public double getDfv_cantidad() {
    return dfv_cantidad;
}
public void setDfv_cantidad(double dfvCantidad) {
    dfv_cantidad = dfvCantidad;
}
public double getDfv_pvp() {
    return dfv_pvp;
}
public void setDfv_pvp(double dfvPvp) {
    dfv_pvp = dfvPvp;
}
public double getDfv_descuento() {
    return dfv_descuento;
}
public void setDfv_descuento(double dfvDescuento) {
    dfv_descuento = dfvDescuento;
}
public boolean isDfv_estado() {
    return dfv_estado;
}
public void setDfv_estado(boolean dfvEstado) {
```



```
        dfv_estado = dfvEstado;
    }
    public DetalleFacturaVenta(){}

    public DetalleFacturaVenta(int dfvSecuencial, int cfvId,
        int matCodigo, double dfvCantidad, double dfvPvp,
        double dfvDescuento, boolean dfvEstado) {
        super();
        dfv_secuencial = dfvSecuencial;
        cfv_id = cfvId;
        mat_codigo = matCodigo;
        dfv_cantidad = dfvCantidad;
        dfv_pvp = dfvPvp;
        dfv_descuento = dfvDescuento;
        dfv_estado = dfvEstado;
    }

}
```

- **Clase:** CabeceraInventario.java

```
package Entidades;

import java.util.Date;

public class CabeceraInventario {

    int ifi_numinv =0;
    String emp_cedula = "";
    Date ifi_fecha_inicial;
    boolean ifi_estado = false;
    double ifi_inicial_peso =0;
    double ifi_final_peso =0;
    Date ifi_fecha_final;
    int bma_secuencial =0;

    public CabeceraInventario(){}

    public int getIfi_numinv() {
        return ifi_numinv;
    }

    public void setIfi_numinv(int ifiNuminv) {
```



```
        ifi_numinv = ifiNuminv;
    }

    public String getEmp_cedula() {
        return emp_cedula;
    }

    public void setEmp_cedula(String empCedula) {
        emp_cedula = empCedula;
    }

    public Date getIfi_fecha_inicial() {
        return ifi_fecha_inicial;
    }

    public void setIfi_fecha_inicial(Date ifiFechaInicial) {
        ifi_fecha_inicial = ifiFechaInicial;
    }

    public boolean isIfi_estado() {
        return ifi_estado;
    }

    public void setIfi_estado(boolean ifiEstado) {
        ifi_estado = ifiEstado;
    }

    public double getIfi_inicial_peso() {
        return ifi_inicial_peso;
    }

    public void setIfi_inicial_peso(double ifiInicialPeso) {
        ifi_inicial_peso = ifiInicialPeso;
    }

    public double getIfi_final_peso() {
        return ifi_final_peso;
    }

    public void setIfi_final_peso(double ifiFinalPeso) {
        ifi_final_peso = ifiFinalPeso;
    }

    public Date getIfi_fecha_final() {
        return ifi_fecha_final;
    }
```



```
}

public void setIfi_fecha_final(Date ifiFechaFinal) {
    ifi_fecha_final = ifiFechaFinal;
}

public int getBma_secuencial() {
    return bma_secuencial;
}

public void setBma_secuencial(int bmaSecuencial) {
    bma_secuencial = bmaSecuencial;
}

public CabeceraInventario(int ifiNuminv, String empCedula,
    Date ifiFechaInicial, boolean ifiEstado, double ifiInicialPeso,
    double ifiFinalPeso, Date ifiFechaFinal, int bmaSecuencial) {
    super();
    ifi_numinv = ifiNuminv;
    emp_cedula = empCedula;
    ifi_fecha_inicial = ifiFechaInicial;
    ifi_estado = ifiEstado;
    ifi_inicial_peso = ifiInicialPeso;
    ifi_final_peso = ifiFinalPeso;
    ifi_fecha_final = ifiFechaFinal;
    bma_secuencial = bmaSecuencial;
}

}
```

- **Clase:** DetallaStock.java

```
package Entidades;

import java.util.Date;

public class DetalleStock {

    int dst_secuencial = 0;
    int stk_secuencial = 0;
    Date dst_fecha;
    double dts_cantidad = 0;
    double dst_costo = 0;
    double dst_total = 0;
```



```
double dst_costoPromedio = 0;
double dst_costoLifo = 0;
double dst_costoPifo = 0;

public int getDst_secuencial() {
    return dst_secuencial;
}
public void setDst_secuencial(int dstSecuencial) {
    dst_secuencial = dstSecuencial;
}
public int getStk_secuencial() {
    return stk_secuencial;
}
public void setStk_secuencial(int stkSecuencial) {
    stk_secuencial = stkSecuencial;
}
public Date getDst_fecha() {
    return dst_fecha;
}
public void setDst_fecha(Date dstFecha) {
    dst_fecha = dstFecha;
}
public double getDts_cantidad() {
    return dts_cantidad;
}
public void setDts_cantidad(double dtsCantidad) {
    dts_cantidad = dtsCantidad;
}
public double getDst_costo() {
    return dst_costo;
}
public void setDst_costo(double dstCosto) {
    dst_costo = dstCosto;
}
public double getDst_total() {
    return dst_total;
}
public void setDst_total(double dstTotal) {
    dst_total = dstTotal;
}
public double getDst_costoPromedio() {
    return dst_costoPromedio;
}
public void setDst_costoPromedio(double dstCostoPromedio) {
```



```
        dst_costoPromedio = dstCostoPromedio;
    }
    public double getDst_costoLifo() {
        return dst_costoLifo;
    }
    public void setDst_costoLifo(double dstCostoLifo) {
        dst_costoLifo = dstCostoLifo;
    }
    public double getDst_costoPifo() {
        return dst_costoPifo;
    }
    public void setDst_costoPifo(double dstCostoPifo) {
        dst_costoPifo = dstCostoPifo;
    }
    public DetalleStock() {
        super();
    }
    public DetalleStock(int dstSecuencial, int stkSecuencial, Date dstFecha,
        double dtsCantidad, double dstCosto, double dstTotal,
        double dstCostoPromedio, double dstCostoLifo, double
dstCostoPifo) {
        dst_secuencial = dstSecuencial;
        stk_secuencial = stkSecuencial;
        dst_fecha = dstFecha;
        dts_cantidad = dtsCantidad;
        dst_costo = dstCosto;
        dst_total = dstTotal;
        dst_costoPromedio = dstCostoPromedio;
        dst_costoLifo = dstCostoLifo;
        dst_costoPifo = dstCostoPifo;
    }
}
}
```

- **Clase:** Stock.java

```
package Entidades;
```

```
public class Stock {
    int sec_stk = 0;
    int mat_cod = 0;
    int bod_numero=0;
    double cantidad;
    double costo = 0;
```



```
double costo_venta = 0;
boolean cobra_iva = false;
boolean estado = false;
boolean esPromedio = false;
boolean esPifo = false;
boolean esLifo = false;
public int getSec_stk() {
    return sec_stk;
}
public void setSec_stk(int secStk) {
    sec_stk = secStk;
}
public int getMat_cod() {
    return mat_cod;
}
public void setMat_cod(int matCod) {
    mat_cod = matCod;
}
public int getBod_numero() {
    return bod_numero;
}
public void setBod_numero(int bodNumero) {
    bod_numero = bodNumero;
}
public double getCantidad() {
    return cantidad;
}
public void setCantidad(double cantidad) {
    this.cantidad = cantidad;
}
public double getCosto() {
    return costo;
}
public void setCosto(double costo) {
    this.costo = costo;
}
public double getCosto_venta() {
    return costo_venta;
}
public void setCosto_venta(double costoVenta) {
    costo_venta = costoVenta;
}
public boolean isCobra_iva() {
    return cobra_iva;
}
```



```
public void setCobra_iva(boolean cobralva) {
    cobra_iva = cobralva;
}
public boolean isEstado() {
    return estado;
}
public void setEstado(boolean estado) {
    this.estado = estado;
}
public boolean isEsPromedio() {
    return esPromedio;
}
public void setEsPromedio(boolean esPromedio) {
    this.esPromedio = esPromedio;
}
public boolean isEsPifo() {
    return esPifo;
}
public void setEsPifo(boolean esPifo) {
    this.esPifo = esPifo;
}
public boolean isEsLifo() {
    return esLifo;
}
public void setEsLifo(boolean esLifo) {
    this.esLifo = esLifo;
}
public Stock() {
    super();
}
public Stock(int secStk, int matCod, int bodNumero, double cantidad,
    double costo, double costoVenta, boolean cobralva,
boolean estado,
    boolean esPromedio, boolean esPifo, boolean esLifo) {
    sec_stk = secStk;
    mat_cod = matCod;
    bod_numero = bodNumero;
    this.cantidad = cantidad;
    this.costo = costo;
    costo_venta = costoVenta;
    cobra_iva = cobralva;
    this.estado = estado;
    this.esPromedio = esPromedio;
    this.esPifo = esPifo;
    this.esLifo = esLifo;
}
```




```
}  
  
}
```

- **Clase:** Usuario.java

package Entidades;

```
public class Usuario {  
    int usu_codigo=0;  
    String emp_cedula="";  
    String usu_nombre="";  
    String usu_clave="";  
    boolean usu_estado = true;  
    public Usuario() {  
        // TODO Auto-generated constructor stub  
    }  
    public int getUsu_codigo() {  
        return usu_codigo;  
    }  
    public void setUsu_codigo(int usuCodigo) {  
        usu_codigo = usuCodigo;  
    }  
    public String getEmp_cedula() {  
        return emp_cedula;  
    }  
    public void setEmp_cedula(String empCedula) {  
        emp_cedula = empCedula;  
    }  
    public String getUsu_nombre() {  
        return usu_nombre;  
    }  
    public void setUsu_nombre(String usuNombre) {  
        usu_nombre = usuNombre;  
    }  
    public String getUsu_clave() {  
        return usu_clave;  
    }  
    public void setUsu_clave(String usuClave) {  
        usu_clave = usuClave;  
    }  
    public boolean isUsu_estado() {  
        return usu_estado;  
    }  
    public void setUsu_estado(boolean usuEstado) {
```



```
        usu_estado = usuEstado;
    }
    public Usuario(int usuCodigo, String empCedula, String usuNombre,
        String usuClave, boolean usuEstado) {
        super();
        usu_codigo = usuCodigo;
        emp_cedula = empCedula;
        usu_nombre = usuNombre;
        usu_clave = usuClave;
        usu_estado = usuEstado;
    }
}
```

Lógica

En el paquete Lógica se encuentran todos los métodos utilizados para realizar las distintas funciones que tiene el software, a continuación se escribe el código fuente:

- **Clase:** BodegaMaterial_Logica.java

```
package Logica;
```

```
import java.sql.ResultSet;
import javax.swing.JComboBox;
import javax.swing.JOptionPane;
import javax.swing.JTree;
import Util.BodegaCombo;
import Util.JBodegaMateriales;
import Util.MaterialCombo;
import Util.Utilitarios;
import AccesosDatos.Conexion;
import Entidades.BodegaMaterial;
```

```
public class BodegaMaterial_Logica extends Conexion{
    public BodegaMaterial_Logica() {}

    public ResultSet getBodegaMaterialDetalle() {
        abrirConexion();
        try {
            StringBuilder sql = new StringBuilder();
```



```
        sql.append("select ");
        sql.append("ma.mat_nombre,");
        sql.append("bo.bod_nombre,");
        sql.append("ma.mat_codigo,");
        sql.append("bo.bod_numero,");
        sql.append("bm.bma_secuencial");
        sql.append(" from bodega bo,");
        sql.append(" material ma,");
        sql.append(" bodegaMaterial bm");
        sql.append(" where ma.mat_codigo = bm.mat_codigo and");
        sql.append(" bo.bod_numero = bm.bod_numero");
        System.out.println(sql.toString());
        st = con.createStatement();
        rs = st.executeQuery(sql.toString());

    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
    return rs;
}

/*
 * Verifica si existe materiales en bodega en la tabla BodegaMaterial
 * //
 */
public int getExisteBodegaMaterial(int codigoMaterial, int codigoBodega) {
    abrirConexion();
    int secuencialBodegaMaterial = 0;
    try {
        st = con.createStatement();
        StringBuilder sql = new StringBuilder();
        sql.append(" SELECT bma_secuencial from bodegaMaterial");
        sql.append(" WHERE mat_codigo = ");
        sql.append(codigoMaterial);
        sql.append(" and ");
        sql.append("bod_numero =");
        sql.append(codigoBodega);
        rs = st
            .executeQuery(sql.toString());
        System.out.println(sql);
        rs = st.getResultSet();
        rs.next();
        secuencialBodegaMaterial = rs.getInt(1);
        cerrarConexion();
    }
```



```
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
        return secuencialBodegaMaterial;
    }

    public JTree getJtreeBodegaProducto() {
        JTree jt = null;
        try {
            jt = new JTree(JTreeBodegaMaterial.getDefaultTreeModel());
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
        return jt;
    }

    /*
    * Nombre:Insertar_BodegaMaterial
    * Tipo:void
    * Descripcion:Metodo encargado de insertar
    * un objeto BodegaMaterial en la base de datos
    */
    public int Insertar_BodegaMaterial(
        BodegaMaterial bodegamaterial) {
        int secuencialBodegaMaterial = 0;
        abrirConexion();
        try {
            StringBuilder sql = new StringBuilder();

            sql.append(" INSERT INTO BDFacturacionInventario.bodegamaterial
(");

            sql.append("BMA_SECUENCIAL ,");
            sql.append("BOD_NUMERO ,");
            sql.append("MAT_CODIGO ,");
            sql.append("DFC_SECUENCIAL ,");
            sql.append("BMA_CANTIDAD ,");
            sql.append("BMA_ESTADO ,");
            sql.append("BMA_FECHA ");
            sql.append(")");
            sql.append("VALUES (");
            sql.append(bodegamaterial.getBma_secuencial());
            sql.append(", ");
            sql.append(bodegamaterial.getBod_numero());
            sql.append(",");
            sql.append(bodegamaterial.getMat_codigo());
```



```
        sql.append(",");
        sql.append(bodegamaterial.getDfc_secuencial());
        sql.append(",");
        sql.append(bodegamaterial.getBma_cantidad());
        sql.append(",");
        sql.append(bodegamaterial.isBma_estado());
        sql.append(",");
        sql.append(Utilitarios.FechaHora());
        sql.append(")");
        System.out.println(sql.toString());
        ps = con.prepareStatement(sql.toString());
        ps.executeUpdate();
        cerrarConexion();
        secuencialBodegaMaterial = getSecuencialBodegaMaterial();

    } catch (Exception e) {

        System.out.println("No se ha podido insertar la bodega material");

    }
    return secuencialBodegaMaterial;
}

public int getSecuencialBodegaMaterial() {
    abrirConexion();
    int secuencialBodegaMaterial = 0;
    try {
        st = con.createStatement();
        rs = st
            .executeQuery("SELECT      MAX(bma_secuencial)      FROM
bodegaMaterial ");
        rs = st.getResultSet();
        rs.next();
        secuencialBodegaMaterial = rs.getInt(1);
        System.out.println(secuencialBodegaMaterial);
        cerrarConexion();

    } catch (Exception e) {
        System.out.print(e.getMessage());
    }

    return secuencialBodegaMaterial;
}
```



- **Clase:** CabeceraFacturaCompra_Logica.java

```
package Logica;

import AccesoDatos.Conexion;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.EmptyStackException;
import java.util.List;
import org.apache.poi.hpsf.Util;
import Util.Utilitarios;
import Entidades.BodegaMaterial;
import Entidades.CabeceraFacturaCompra;
import Entidades.CabeceraInventario;
import Entidades.DetalleFacturaCompra;
import Entidades.DetalleInventario;
import Entidades.DetalleStock;
import Entidades.Stock;
import Logica.DetalleStock_Logica.TIPOCOSTO;

public class CabeceraFacturaCompra_Logica extends Conexion {

    public CabeceraFacturaCompra_Logica() {

    }

    public void GuardarFacturaGeneralCompra(
        CabeceraFacturaCompra cabeceraFacturaCompra,
        List<DetalleFacturaCompra> listado,
        int numeroBodega) {
        try {

            // Guardar Cabecera Factura
            int secuencialIDFacturaCompra = 0;
            secuencialIDFacturaCompra =
Insertar_CabeceraFacturaCompra(cabeceraFacturaCompra);
            System.out.println("Grabo Cabecera..");
            System.out.println(secuencialIDFacturaCompra);
            // Guardar Detalle Factura
            int secuencialFacturaCompraDetalle = 0;
            for (DetalleFacturaCompra detalleFacturaCompraItem : listado) {

                detalleFacturaCompraItem.setCfc_id(secuencialIDFacturaCompra);
```



```
        secuencialFacturaCompraDetalle = (new
DetalleFacturaCompra_Logica())

        .Insertar_DetalleFacturaCompra(detalleFacturaCompraltem);

        Date fecha = new Date();
        // Guardar Bodega Material
        int secuencialBodegaMaterial = 0;
        int secuencialInventario = 0;

        secuencialBodegaMaterial = (new BodegaMaterial_Logica())

        .getExisteBodegaMaterial(detalleFacturaCompraltem
                                .getMat_codigo(),
numeroBodega);

        //Actualizo o/e Inserto en la tabla STOCK
        Stock stock = (new Stock_Logica()).getStock(
                                detalleFacturaCompraltem.getMat_codigo(),
numeroBodega);

        double totalCosto =0;
        java.util.Date utilDate = new java.util.Date();
        totalCosto = detalleFacturaCompraltem.getDfc_cantidad() *
detalleFacturaCompraltem.getDfc_costo();

        if (stock == null) {
            int secuencialStock = (new
Stock_Logica()).Insertar_Stock(new Stock(0,

                detalleFacturaCompraltem.getMat_codigo(),
                                numeroBodega,
detalleFacturaCompraltem

                                .getDfc_cantidad(),

                detalleFacturaCompraltem.getDfc_costo(), 0, false,

                                false,
false, false, false));

            (new DetalleStock_Logica())
                .Insertar_DetalleStock(new
DetalleStock(0,

                                secuencialStock,
                                utilDate,
```



```
detalleFacturaCompraltem

.getDfc_cantidad(),

detalleFacturaCompraltem.getDfc_costo(),
totalCosto,
detalleFacturaCompraltem

.getDfc_costo(),

detalleFacturaCompraltem.getDfc_costo(),

detalleFacturaCompraltem.getDfc_costo()));
// Insertar_DetalleStock

} else {
double cantidaActual = stock.getCantidad()
+
detalleFacturaCompraltem.getDfc_cantidad();
(new Stock_Logica())

.ActualizarCantidad_StockPorBodegaProducto(
cantidaActual,
detalleFacturaCompraltem

.getMat_codigo(), numeroBodega);

(new DetalleStock_Logica())
.setCostosProducto(new
DetalleStock(0,
stock.getSec_stk(),
utilDate,

detalleFacturaCompraltem.getDfc_cantidad(),

detalleFacturaCompraltem.getDfc_costo(),
totalCosto,

detalleFacturaCompraltem.getDfc_costo(),

detalleFacturaCompraltem.getDfc_costo(),

detalleFacturaCompraltem.getDfc_costo()));
```




INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

```
        if (stock.isEsPromedio() || stock.isEsLifo()
            || stock.isEsPifo()) {

            double costoActual = (new
DetalleStock_Logica()).getCostoProducto(
                                stock.getSec_stk(),

                                stock.isEsPromedio() ? TIPOCOSTO.PROMEDIO
                                                                :
stock.isEsLifo() ? TIPOCOSTO.LIFO : TIPOCOSTO.PIFO);

            (new
Stock_Logica()).ActualizarCostoPorBodegaProducto(
                                detalleFacturaCompraltem
                                costoActual,

                                .getMat_codigo(), numeroBodega);
        }

        if (secuencialBodegaMaterial == 0) {
            // Bodega Material
            BodegaMaterial bodegaMaterial = new
BodegaMaterial(0,
                                numeroBodega,
detalleFacturaCompraltem
                                .getMat_codigo(),
                                secuencialFacturaCompraDetalle,
                                secuencialFacturaCompraDetalle,
fecha, true);

            secuencialBodegaMaterial = (new
BodegaMaterial_Logica())

            .Insertar_BodegaMaterial(bodegaMaterial);

            // Cabecera Inventario
            CabeceraInventario cabeceraInventario = new
CabeceraInventario(
                                0, "ADMIN", fecha, true,
detalleFacturaCompraltem
                                .getDfc_cantidad(),

            detalleFacturaCompraltem.getDfc_cantidad(), fecha,
```



```
                                secuencialBodegaMaterial);

                                secuencialInventario =      (new
CabeceraInventario_Logica())

                                .InsertarCabeceraInventario(cabeceraInventario);

                                // Detalle Invetario
                                DetalleInventario detalleInventarioItem = new
DetalleInventario(
                                0, secuencialInventario,
detalleFacturaCompraltem
                                .getDfc_cantidad(), true,
fecha,

                                detalleFacturaCompraltem.getDfc_costo(),
                                cabeceraFacturaCompra.getCfc_numCfc());

                                (new DetalleInventario_Logica())

                                .InsertarDetalleInventario(detalleInventarioItem);

                                } else {
                                secuencialInventario = (new
CabeceraInventario_Logica())

                                .getSecuencialCabeceraInventarioPorBodegaMaterial(secuencialBodegaMaterial);

                                // Detalle Invetario
                                DetalleInventario detalleInventarioItem = new
DetalleInventario(
                                0, secuencialInventario,
detalleFacturaCompraltem
                                .getDfc_cantidad(), true,
fecha,

                                detalleFacturaCompraltem.getDfc_costo(),
                                cabeceraFacturaCompra.getCfc_numCfc());

                                (new DetalleInventario_Logica())

                                .InsertarDetalleInventario(detalleInventarioItem);
```



```
(new CabeceraInventario_Logica()).

ActualizarCabeceraInventario(detalleFacturaCompraltem
                                                                    .getDfc_cantidad(),
secuencialInventario, 2
                                                                    );

    }

    }

    } catch (Exception e) {
        // TODO: handle exception
    }

}

/*
 * Nombre:Insertar_CabeceraFacturaCompra
 * Tipo:int
 * Descripcion:Metodo encargado de insertar
 * todos los objetos en la tabla CabeceraFacturaCompra
 */
public int Insertar_CabeceraFacturaCompra(
    CabeceraFacturaCompra cabeceraFacturaCompra) {
    abrirConexion();
    int secuencial = 0;
    try {
        StringBuilder sql = new StringBuilder();

        sql
            .append(" INSERT INTO
BDFacturacionInventario.cabecerafacturacompra (");
        sql.append("CFC_ID ,");
        sql.append("PRO_CEDRUC ,");
        sql.append("EMP_CEDULA ,");
        sql.append("BOD_NUMERO ,");
        sql.append("CFC_NUMCFC ,");
        sql.append("CFC_FECHAEMISION ,");
        sql.append("CFC_IVA ,");
        sql.append("CFC_TOTAL ,");
        sql.append("CFC_ESTADO, ");
        sql.append("TDO_SECUENCIAL ");
        sql.append(")");
```



```
        sql.append("VALUES (");
        sql.append(cabeceraFacturaCompra.getCfc_id());
        sql.append(",");
        sql.append(cabeceraFacturaCompra.getPro_CedRuc());
        sql.append(",");
        sql.append(cabeceraFacturaCompra.getEmp_cedula());
        sql.append(",");
        sql.append(cabeceraFacturaCompra.getBod_numero());
        sql.append(",");
        sql.append(cabeceraFacturaCompra.getCfc_numCfc());
        sql.append(",");
        sql.append(Utilitarios.FechaHora());
        sql.append(",");
        sql.append(cabeceraFacturaCompra.getCfc_iva());
        sql.append(",");
        sql.append(cabeceraFacturaCompra.getCfc_total());
        sql.append(",");
        sql.append(cabeceraFacturaCompra.isCfc_estado());
        sql.append(",");
        sql.append(cabeceraFacturaCompra.getTdo_secuencial());
        sql.append(")");
        ps = con.prepareStatement(sql.toString());
        ps.executeUpdate();
        System.out.println("Prepara para pedir secuencial");
        secuencial = getSecuencialFacturaCompra();
        System.out.println(secuencial);

    } catch (Exception e) {
        // Siempre aqui imprime el posible mensaje de error que te emuite
        la
        // base de datos
        System.out.println(e.getMessage());
        System.out.println("No se ha podido insertar la
        cabeceraFacturaCompra");
        throw new EmptyStackException();

    }

    return secuencial;
    // conexion.cerrarConexion();
}

/*
 * Nombre:FechaFacturaCompra Tipo:String Descripcion:Metodo encargado de
 * devolver la fecha actual
 */
```



```
*/  
int siguiente;  
  
public String FechaFacturaCompra() {  
    Date hoy = new Date();  
    String patronFecha = "yyyy-MM-dd";  
    SimpleDateFormat sdtFecha = new SimpleDateFormat(patronFecha);  
    String datoFecha = sdtFecha.format(hoy);  
  
    return datoFecha;  
  
}  
  
public int getSecuencialFacturaCompra() {  
    abrirConexion();  
    try {  
        st = con.createStatement();  
        rs = st  
                                .executeQuery("SELECT MAX(CFC_ID) FROM  
cabecerafacturacompra ");  
        rs = st.getResultSet();  
        rs.next();  
        siguiente = rs.getInt(1);  
        cerrarConexion();  
  
    } catch (Exception e) {  
        System.out.print(e.getMessage());  
    }  
  
    return siguiente;  
}
```



BIBLIOGRAFÍA

- Uwe Rozanski, (2007), Javabeans 3.0 con eclipse, Editorial: MARCOMBO, S.A., Edición: 1,
- Harvey M, (2004), Cómo programar en java, Editorial: PEARSON EDUCATION, Edición: Berthold Daum, (2005), Eclipse 3 para desarrolladores java, Editorial: ADISSON, Edición: 1
- Paul DuBois, (2005), MySQL, Editorial: ANAYA, Edición: 1
- Baron Shuwartz, Meter Zaitsev, (2006), MySQL Avanzado, Editorial: ENI, Edición: 2
- Gema Cmpiña, Ma Jesus Fernández, (2008) Empresa y Administración, Editorial: EDITEX, Edición 1, Bogotá-Colombia.
- Max Muller, (2004), Fundamentos de Administración de Inventarios, Editorial: NORMA, Edición: 2, España-Barcelona
- McGraw-Hill, (2005), Contabilidad principios y aplicaciones, Editorial: REVERTÉ, Edición: 2, España-Barcelona



NETGRAFÍA

- <http://www.java.eclipse.video-tutorials-1-16-mp.html>
- <http://www.mtbase.com/productos/modelamientometadatos/powerdesigner>
- <http://es.wikipedia.org/wiki/MySQL>
- [http://es.wikipedia.org/wiki/Eclipse_\(software\).](http://es.wikipedia.org/wiki/Eclipse_(software))
- <http://eclipsetutorial.forge.os4os.org/programa.htm>
- <http://www.telecable.es/personales/retamar/articulos/eclipse.pdf>
- <http://www.monografias.com/trabajos60/control-inventarios/control-inventarios.shtml>
- <http://www.softonic.com/windows/facturacion-y-cobros>