



INSTITUTO TECNOLÓGICO
"CORDILLERA"

CARRERA DE ANÁLISIS DE SISTEMAS

CONTROL DE INVENTARIOS Y FACTURACIÓN MEDIANTE UNA
APLICACIÓN WEB PARA LA EMPRESA BRAVO POLO.

Proyecto de desarrollo de sistema previo a la obtención del título de Tecnólogo en
Análisis de Sistemas

Autor: Andagoya Guamán Luis Orlando

Tutora: Ing. Terán Diana

Quito, Abril de 2014

Declaración de aprobación de tutor

En mi calidad de Tutor del trabajo sobre el proyecto: **control de inventarios y facturación mediante una aplicación web para la empresa Bravo Polo** presentado por el ciudadano: Luis Orlando Andagoya Guamán, estudiante de la Escuela de Sistemas, considero que dicho informe reúne los requisitos y méritos suficientes para ser sometido a la evaluación por parte del Tribunal de Grado, que el Honorable Consejo de Escuela designe, para su correspondiente estudio y calificación.

Quito, Abril del 2014

Ing. Diana Terán

Director de Trabajo de Graduación

Declaración de Autoría del Estudiante

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

Luis Orlando Andagoya Guamán
CI: 172106440-8

CONTRATO DE CESIÓN SOBRE DERECHOS PROPIEDAD INTELECTUAL

Comparecen a la celebración del presente contrato de cesión y transferencia de derechos de propiedad intelectual, por una parte, el estudiante LUIS ORLANDO ANDAGOYA GUAMAN, por sus propios y personales derechos, a quien en lo posterior se le denominará el “CEDENTE”; y, por otra parte, el INSTITUTO SUPERIOR TECNOLÓGICO CORDILLERA, representado por su Rector el Ingeniero Ernesto Flores Córdova, a quien en lo posterior se lo denominará el “CESIONARIO”. Los comparecientes son mayores de edad, domiciliados en esta ciudad de Quito Distrito Metropolitano, hábiles y capaces para contraer derechos y obligaciones, quienes acuerdan al tenor de las siguientes cláusulas:

PRIMERA: ANTECEDENTE.- a) El Cedente dentro del pensum de estudio en la carrera de análisis de sistemas que imparte el Instituto Superior Tecnológico Cordillera, y con el objeto de obtener el título de Tecnólogo en Análisis de Sistemas, el estudiante participa en el proyecto de grado denominado “CONTROL DE INVENTARIOS Y FACTURACIÓN MEDIANTE UNA APLICACIÓN WEB PARA LA EMPRESA BRAVO POLO.”, el cual incluye la creación y desarrollo del programa de ordenador o software, para lo cual ha implementado los conocimientos adquiridos en su calidad de alumno. b) Por iniciativa y responsabilidad del Instituto Superior Tecnológico Cordillera se desarrolla la creación del programa de ordenador, motivo por el cual se regula de forma clara la cesión de los derechos de autor que genera la obra literaria y que es producto del proyecto de grado, el mismo que culminado es de plena aplicación técnica, administrativa y de reproducción.

SEGUNDA: CESIÓN Y TRANSFERENCIA.- Con el antecedente indicado, el Cedente libre y voluntariamente cede y transfiere de manera perpetua y gratuita todos los derechos patrimoniales del programa de ordenador descrito en la cláusula

anterior a favor del Cesionario, sin reservarse para sí ningún privilegio especial (código fuente, código objeto, diagramas de flujo, planos, manuales de uso, etc.). El Cesionario podrá explotar el programa de ordenador por cualquier medio o procedimiento tal cual lo establece el Artículo 20 de la Ley de Propiedad Intelectual, esto es, realizar, autorizar o prohibir, entre otros: a) La reproducción del programa de ordenador por cualquier forma o procedimiento; b) La comunicación pública del software; c) La distribución pública de ejemplares o copias, la comercialización, arrendamiento o alquiler del programa de ordenador; d) Cualquier transformación o modificación del programa de ordenador; e) La protección y registro en el IEPI el programa de ordenador a nombre del Cesionario; f) Ejercer la protección jurídica del programa de ordenador; g) Los demás derechos establecidos en la Ley de Propiedad Intelectual y otros cuerpos legales que normen sobre la cesión de derechos de autor y derechos patrimoniales.

TERCERA: OBLIGACIÓN DEL CEDENTE.- El cedente no podrá transferir a ningún tercero los derechos que conforman la estructura, secuencia y organización del programa de ordenador que es objeto del presente contrato, como tampoco emplearlo o utilizarlo a título personal, ya que siempre se deberá guardar la exclusividad del programa de ordenador a favor del Cesionario.

CUARTA: CUANTIA.- La cesión objeto del presente contrato, se realiza a título gratuito y por ende el Cesionario ni sus administradores deben cancelar valor alguno o regalías por este contrato y por los derechos que se derivan del mismo.

QUINTA: PLAZO.- La vigencia del presente contrato es indefinida.

SEXTA: DOMICILIO, JURISDICCIÓN Y COMPETENCIA.- Las partes fijan como su domicilio la ciudad de Quito. Toda controversia o diferencia derivada de

éste, será resuelta directamente entre las partes y, si esto no fuere factible, se solicitará la asistencia de un Mediador del Centro de Arbitraje y Mediación de la Cámara de Comercio de Quito. En el evento que el conflicto no fuere resuelto mediante este procedimiento, en el plazo de diez días calendario desde su inicio, pudiendo prorrogarse por mutuo acuerdo este plazo, las partes someterán sus controversias a la resolución de un árbitro, que se sujetará a lo dispuesto en la Ley de Arbitraje y Mediación, al Reglamento del Centro de Arbitraje y Mediación de la Cámara de comercio de Quito, y a las siguientes normas: a) El árbitro será seleccionado conforme a lo establecido en la Ley de Arbitraje y Mediación; b) Las partes renuncian a la jurisdicción ordinaria, se obligan a acatar el laudo arbitral y se comprometen a no interponer ningún tipo de recurso en contra del laudo arbitral; c) Para la ejecución de medidas cautelares, el árbitro está facultado para solicitar el auxilio de los funcionarios públicos, judiciales, policiales y administrativos, sin que sea necesario recurrir a juez ordinario alguno; d) El procedimiento será confidencial y en derecho; e) El lugar de arbitraje serán las instalaciones del centro de arbitraje y mediación de la Cámara de Comercio de Quito; f) El idioma del arbitraje será el español; y, g) La reconvenición, caso de haberla, seguirá los mismos procedimientos antes indicados para el juicio principal.

SÉPTIMA: ACEPTACIÓN.- Las partes contratantes aceptan el contenido del presente contrato, por ser hecho en seguridad de sus respectivos intereses.

En aceptación firman a los veintiún días del mes de abril del dos mil catorce.

f) _____

f) _____

C.C. N° 1721064408

Instituto Superior Tecnológico Cordillera

CEDENTE

CESIONARIO

AGRADECIMIENTOS

Primeramente me gustaría agradecer a DIOS, por ayudarme a culminar mi carrera ya que él ha sido quien me ha cuidado y dado fuerzas para no claudicar ante los problemas y circunstancias que han surgido.

Agradezco inmensamente al Instituto Tecnológico Superior Cordillera por darme la oportunidad de estudiar la Carrera de Análisis de Sistemas, a mis profesores que me han inculcado conocimientos referentes a la Informática así como valores morales y humanos.

También me gustaría agradecer a mi esposa, quien a pesar de todos los problemas que han surgido por el motivo de estar estudiando esta Carrera ha sabido comprenderme y apoyarme en todo lo que he necesitado.

Mi esposa, mi hijo y mis padres son la razón de seguir adelante.

Gracias.

DEDICATORIA

Con mucho amor dedico este trabajo a mi querido Dios porque su inmenso amor siempre me ha protegido, ayudándome a caminar día a día a pesar de los problemas y circunstancias adversas, mostrándome su inmenso amor para conmigo en cada momento.

A mi esposa por su extraordinario amor y comprensión que ha sido un gran apoyo durante toda la carrera.

Luis Andagoya.

Índice General

	Pág.
Declaración de aprobación de tutor	i
Declaración de Autoría del Estudiante	ii
Contrato de Cesión Sobre Derechos Propiedad Intelectual.....	iii
Agradecimientos.....	vi
Dedicatoria.....	vii
Índice General	viii
Índice de Tablas	xi
Índice de Figuras.....	xiii
Índice de Anexos	xix
Resumen Ejecutivo.....	xx
Capítulo I: Antecedentes	1
1.01 Contexto.....	1
1.02 Justificación.....	2
1.03 Definición del Problema Central.....	3
Capítulo II: Análisis de Involucrados	5
2.01 Mapeo de Involucrados	5
2.02 Matriz de Análisis de los Involucrados	6
Capítulo III: Problemas y Objetivos.....	9
3.01 Árbol de problemas.	9
3.02 Árbol de objetivos.	11

Capítulo IV: Análisis de Alternativas. 13

4.01 Matriz De Análisis De Alternativas 13

4.02 Matriz de Análisis de Impacto de los Objetivos 14

4.03 Diagrama de Estrategias 16

4.04 Matriz de Marco Lógico..... 16

Capítulo V: Justificación Propuesta..... 17

5.01 Justificación..... 17

5.02 Análisis y diseño 18

5.02.01 Diagrama de Casos de Uso18

5.02.02 Diagrama de Casos de Realización.....27

5.02.03 Diagramas de Secuencia30

5.02.04 Diagramas de Colaboración.....34

5.02.05 Diagrama de Componentes37

5.02.06 Diagrama de Clases39

5.03 Desarrollo42

5.03.01 Arquitectura del sistema42

5.03.02 Estándares de programación48

5.03.03 Estándares de Base de Datos61

5.03.04 Diseño de interfaces.....73

5.04 Pruebas88

5.04.01 Documento De Casos De Prueba.....88

Capítulo VI: Aspectos Administrativos	103
6.01 Recursos	103
6.02 Presupuesto.....	105
6.03 Cronograma.....	106
Capítulo VII: Conclusiones y Recomendaciones	107
7.01 Conclusiones.	107
7.02 Recomendaciones.....	107
Referencias Bibliográficas:	108
Anexos	109

Índice de Tablas

	Pág.
Tabla 1 Matriz de Análisis de fuerzas T	3
Tabla 2 Matriz de Análisis de los Involucrados	6
Tabla 3 Matriz de Análisis de Impactos de los Objetivos.....	14
Tabla 4 Proceso de Ingreso al Sistema.....	20
Tabla 5 Proceso para la Venta de Productos.....	22
Tabla 6 Proceso para la Generación de la Factura	23
Tabla 7 Proceso para la Entrega de Productos	24
Tabla 8 Proceso de Generación de Nota de Crédito	25
Tabla 9 Proceso de la Compra de Productos a Proveedores	26
Tabla 10 Proceso para la Compra de Productos	27
Tabla 11 Estándares para los Nombres de los Controles	50
Tabla 12 Prefijos de Elementos Html5	52
Tabla 13 Prefijos de Alcance de Variables	53
Tabla 14 Tipos de Datos de Variables	54
Tabla 15 Estándares de Generación de Texto para el Usuario.....	59
Tabla 16 Historial de Revisiones	88
Tabla 17 Venta de Producto.....	95
Tabla 18 Escenario de Venta de Producto	96
Tabla 19 Emisión de Factura.....	96
Tabla 20 Escenario de Emisión de Factura	97
Tabla 21 Entrega de Productos al Cliente	97
Tabla 22 Escenario de Entrega de Productos al Cliente	98
Tabla 23 Devolución de Productos	98
Tabla 24 Escenario de Devolución de Productos.....	99

Pág.

Tabla 25 Compra de Productos	99
Tabla 26 Escenario de Compra de Productos	100
Tabla 27 Generación de Retenciones.....	101
Tabla 28 Escenario de Generación de Retenciones	102
Tabla 29 Recurso Humano.....	103
Tabla 30 Recurso Material	104
Tabla 31 Presupuesto	105
Tabla 32 Matriz de Alternativas.....	110
Tabla 33 Análisis de Marco Lógico	111

Índice de Figuras

	Pág.
Figura 1 Mapeo de Involucrados.....	5
Figura 2 Árbol de Problemas	9
Figura 3 Árbol de Objetivos.....	11
Figura 4 Diagrama de Estrategias.	16
Figura 5 Caso de Uso: Ingreso al Sistema	20
Figura 6 Caso de Uso: Venta de Productos.....	21
Figura 7 Caso de Uso: Emisión de Facturas.	22
Figura 8 Caso de Uso: Entrega de Productos.....	23
Figura 9 Caso de Uso: Devolución de Productos	24
Figura 10 Caso de Uso: Compra de Productos	25
Figura 11 Caso de Uso: Generación de Retenciones.	26
Figura 12 Caso de Realización: Ingreso al Sistema.	27
Figura 13 Caso de Realización: Emisión de Facturas.....	28
Figura 14 Caso de Realización: Entrega de Productos.	28
Figura 15 Caso de Realización: Devolución de Productos	29
Figura 16 Caso de Realización: Entrega de Productos	29
Figura 17 Caso de Realización: Generación de Retenciones.....	30
Figura 18 Driagrama de Secuencia: Ingreso al Sistema.....	30
Figura 19 Driagrama de Secuencia: Venta de Productos.....	31
Figura 20 Driagrama de Secuencia: Emisión de Facturas.	31
Figura 21 Driagrama de Secuencia: Entrega de Productos.....	32
Figura 22 Driagrama de Secuencia: Devolución de Productos.....	32
Figura23 Driagrama de Secuencia: Entrega de Productos.....	33
Figura 24 Driagrama de Secuencia: Generación de Retenciones	33

Figura 25 Diagrama de Colaboración: Ingreso al Sistema.....	34
Figura 26 Diagrama de Colaboración: Venta de Productos.....	34
Figura 27 Diagrama de Colaboración: Emisión de Facturas	34
Figura 28 Diagrama de Colaboración: Entrega de Productos.....	35
Figura 29 Diagrama de Colaboración: Devolución de Productos.....	35
Figura 30 Diagrama de Colaboración: Entrega de Productos.....	35
Figura 31 Diagrama de Colaboración: Generación de Retenciones	36
Figura 32 Diagrama de Componentes.....	37
Figura 33 Diagrama de Clases	39
Figura 34 Diagrama de Base de Datos Lógico.....	40
Figura 35 Diagrama de Base de Datos Físico	41
Figura 36 Programación por Capas.....	42
Figura 37 Formato de los Casos de Uso.....	62
Figura 38 Formato de Actores.....	62
Figura 39 Formato para Include	63
Figura 40 Formato para Extend.....	63
Figura 41 Orden de los Diagramas Uml	64
Figura 42 Orden Incorrecta de Diagramas Uml.....	64
Figura 43 Elaboración de Diagrama de Secuencia	66
Figura 44 Línea de Vida de Diagrama de Secuencia	67
Figura 45 Símbolos de Diagrama de Secuencia.....	67
Figura 46 Mensajes de Diagrama de Secuencia.....	68
Figura 47 Representación de una Clase	70
Figura 48 Representación de una Clase Completa	71
Figura 49 Representación de la Relación de Herencia	71

Figura 50 Representación de Asociaciones en General	72
Figura 51 Diseño de Interfaces: Ingreso al Sistema.....	73
Figura 52 Diseño de Interfaces Creación de Empresas.....	73
Figura 53 Diseño de Interfaces: Creación de Administradores	74
Figura 54 Diseño de Interfaces: Menú Principal.....	74
Figura 55 Diseño de Interfaces: Insertar y Actualizar Usuario.....	75
Figura 56 Diseño de Interfaces: Lista de Usuarios	75
Figura 57 Diseño de Interfaces: Lista de Permisos de Acceso	76
Figura 58 Diseño de Interfaces: Modificar Categoría.....	76
Figura 59 Diseño de Interfaces: Lista de clientes	77
Figura 60 Diseño de Interfaces: Insertar y Modificar Cliente.....	77
Figura 61 Diseño de Interfaces: Lista de Proveedores.....	78
Figura 62 Diseño de Interfaces: Insertar y Modificar Proveedor.....	78
Figura 63 Diseño de Interfaces: Lista de productos.....	79
Figura 64 Diseño de Interfaces: Insertar y Modificar Productos	79
Figura 65 Diseño de Interfaces: Compras de Productos	80
Figura 66 Diseño de Interfaces: Buscar Proveedor.....	80
Figura 67 Diseño de Interfaces: Guardar Compras.....	81
Figura 68 Diseño de Interfaces: Factura	81
Figura 69 Diseño de Interfaces: Buscar Cliente.....	82
Figura 70 Diseño de Interfaces: Cobrar	82
Figura 71 Diseño de Interfaces: Imprimir Factura	83
Figura 72 Diseño de Interfaces: Lista de Facturas	83
Figura 73 Diseño de Interfaces: Nota de Crédito.....	84
Figura 74 Diseño de Interfaces: Imprimir Nota de Crédito	84

Figura 75 Diseño de Interfaces: Anular Factura	85
Figura 76 Diseño de Interfaces: Existencias de Productos	85
Figura 77 Diseño de Interfaces: Imprimir Existencias.....	86
Figura 78 Diseño de Interfaces: Lista de Detalle de Kardex	86
Figura 79 Diseño de Interfaces: Imprimir Detalle de Kardex.....	87
Figura 80 Diagrama General de Casos de Uso	113
Figura 81 Manual de Usuario: Funcionalidad del Usuario del Sistema	114
Figura 82 Manual de Usuario: Ingreso al Sistema	115
Figura 83 Manual de Usuario: Usuario y Contraseña.....	116
Figura 84 Manual de Usuario: Menú Principal.....	116
Figura 85 Manual de Usuario: Anular Facturas	117
Figura 86 Manual de Usuario: Listado de Clientes.....	117
Figura 87 Manual de Usuario: Devoluciones.....	118
Figura 88 Manual de Usuario: Nota de Credito	118
Figura 89 Manual de Usuario: Factura.....	119
Figura 90 Manual de Usuario: Imprimir Factura	119
Figura 91 Manual de Usuario: Listado de Facturas	120
Figura 92 Manual de Usuario: Listado de Notas de Credito.....	120
Figura 93 Manual de Usuario: Compras	121
Figura 94 Manual de Usuario: Marca de Productos.....	122
Figura 95 Manual de Usuario: Productos.....	122
Figura 96 Manual de Usuario: Listado de Proveedores	123
Figura 97 Manual de Usuario: Listado de Usuarios.....	123
Figura 98 Manual de Usuario: Listado de Categorías.....	124
Figura 99 Manual de Usuario: Estado de productos	124

Figura 100 Manual de Usuario: Kardex de Productos	125
Figura 101 Manual de Usuario: Ventas de Productos.....	125
Figura 102 Manual de Usuario: Cambio de Clave.....	126
Figura 103 Manual Tecnico: Pagina de Descarga de Netbeans	127
Figura 104 Manual Tecnico: Pagina de Opciones de Descarga.....	128
Figura 105 Manual Tecnico: Pagina de Recursos.....	129
Figura 106 Manual Tecnico: Pagina de Confirmación de Descarga.....	129
Figura 107 Manual Tecnico: Progreso de Instalación.....	130
Figura 108 Manual Tecnico: Opciones de Instalación.....	131
Figura 109 Manual Tecnico: Opciones de Servidor	131
Figura 110 Manual Tecnico: Aviso de Instalación	132
Figura 111 Manual Tecnico: Licencia	132
Figura 112 Manual Tecnico: Contrato de Licencia.....	133
Figura 113 Manual Tecnico: Ubicación de Instalación	134
Figura 114 Manual Tecnico: Ubicaciones de Componentes.....	134
Figura 115 Manual Tecnico: Ubicación de Servidor Apache.....	135
Figura 116 Manual Tecnico: Información de Productos.....	135
Figura 117 Manual Tecnico: Proceso de Instalación	136
Figura 118 Manual Tecnico: Finalizar Instalación	136
Figura 119 Manual Tecnico: Descarga del Servidor Wamp Server	137
Figura 120 Manual Tecnico: Pagina de Descargas	137
Figura 121 Manual Tecnico: Progreso de Descarga	137
Figura 122 Manual Tecnico: Instalación de Wamp Server	138
Figura 123 Manual Tecnico: Licencia de Uso	138
Figura 124 Manual Tecnico: Carpeta de Ubicación	139

Pág.

Figura 125 Manual Tecnico: Creación de Iconos	139
Figura 126 Manual Tecnico: Aceptar Licencia de Uso	140
Figura 127 Manual Tecnico: Instalar Wamp Server	140
Figura 128 Manual Tecnico: Progreso de la Instalación.....	140
Figura 129 Manual Tecnico: Opciones de Navegador.....	141
Figura 130 Manual Tecnico: Buscar Navegador	142
Figura 131 Manual Tecnico: Configurar Servidor.....	142
Figura 132 Manual Tecnico: Finalizar Instalación	143
Figura 133 Manual Tecnico: Ver Icono de Wamp Server	143
Figura 134 Manual Tecnico: Ver Funcionando el Servidor.....	144
Figura 135 Manual Tecnico: Obciones de Servidor	144
Figura 136 Manual Tecnico: Comprobar el Servidor	145
Figura 137 Manual Tecnico: Pagina Principal del Servidor	190

Índice de Anexos

	Pág.
A.01 Matriz de análisis de alternativas	110
A.03 Casos de Uso	113
A.04 Manual de Usuario	114
A.04.a Propósito del Documento	114
A.04.b Conceptos Importantes	115
A.05 Manual Técnico	127
A.05.a Instalación de Id de desarrollo	127
A.06.b Instalación del servidor	137
A.05.c Base de Datos	145
A.05.d Código Fuente	155
A.06 Cronograma de Actividades	190
A.07 Glosario de Términos	191

Resumen Ejecutivo

En los últimos años las empresas del Ecuador han crecido económicamente de una manera muy considerable lo que ha generado un mayor control en sus finanzas, ventas y adquisición de mercaderías, esto crea la necesidad de un sistema que maneje todos los mencionados procesos de una manera agradable y comprensible para el usuario.

La compañía Bravo Polo dedicada a la venta de productos adhesivos, no es indiferente ante tal crecimiento y como consecuencia surge el inconveniente de no controlar las ventas, compras y existencias de mercancía de una forma adecuada. Para dar solución a este problema se genera una aplicación web que permite registrar los procesos de facturación e inventarios de bodega de una forma automática, amigable y comprensible para los diferentes usuarios de dicho sistema.

Esta aplicación facilita el registro de compras y ventas de mercancías, genera inventarios de bodega con sus respectivos kardex de cada uno de los productos, así como emite facturas en la venta de mercaderías y realiza reportes de ventas, cabe destacar que este aplicativo está desarrollado conforme a los requerimientos previamente levantados en la empresa en cuestión.

Esta aplicación ayuda al crecimiento financiero y a la disminución de costos, permitiendo que la empresa abarque un mayor número de mercados y que sus productos sean reconocidos por su alta calidad y bajos precios, lo cual solo es posible gracias a una buena administración de bodega y un adecuado control de facturación, lo que mediante el software implementado se está realizando.

Abstract

In recent years companies have grown economically Ecuador a very large which has led to greater control over their finances , sales and purchase of goods way , this creates the need for a system that handles all these processes in a nice way and understandable to the user.

Bravo Company Polo is not indifferent to such growth so the drawback of not track revenue, expenses and sales of merchandise in an appropriate way arises. To solve this problem, a web application that allows billing processes and inventory hold of an automatic, nice and understandable to the various users of the system is generated.

This application facilitates the control of purchases and sales of products, warehouse inventory generated with their respective kardex for each of the products admitted to hold, and issue invoices when selling goods and makes sales reports, note that this application is developed according to the requirements previously raised in the company concerned.

Helping the financial growth and lower costs of goods, allowing the company to cover a greater number of markets and their products are recognized for their high quality and low prices, which is only possible through good management of warehouse and adequate control of billing, which means the implemented software is performing.

Capítulo I: Antecedentes

1.01 Contexto.

En los últimos años el sector empresarial del Ecuador ha tenido un gran índice de crecimiento, lo que ha demandado una adaptación en la forma de emitir facturas y llevar los controles de inventarios de bodega, estos procedimientos han ido mejorando con el pasar de los años y gracias al órgano regulatorio de gobierno, el Servicio de Rentas Internas todos, los contribuyentes ya sea naturales o jurídicos están obligados a facturar; de ahí la necesidad de manejar eficientemente la información obtenida en los procesos anteriormente mencionados.

Desde la creación de Bravo Polo Cía. Ltda., dedicada a la producción y comercialización de cintas y etiquetas adhesivas; su crecimiento ha sido notorio debido a la calidad de su producto y estrategias de ventas, esto demanda un mayor control de productos que ingresan a bodega a través de proveedores o egresan al realizar las ventas, dichos procesos son controlados por un kardex de bodega que es el documento encargado de llevar un historial de los productos, también se los puede dar de baja mediante el documento anteriormente mencionado, sin un control necesario se podrían generar errores que perjudicarían económicamente con la pérdida o extravío de productos.

Con respecto a facturación se puede decir que la desorganización y un inadecuado control han llevado a que el cliente pierda su tiempo en esperas innecesarias y al emitir los reportes de ventas contengan errores. El retraso al emitir una factura, la dificultad para generar los formularios para el pago de impuestos y los datos poco exactos o tardíos que se tiene en bodega ha generado dentro de la empresa un retraso en sus tareas diarias.

1.02 Justificación

Partiendo de que la facturación y control de inventarios de bodega es primordial para cualesquier empresa, dichos procesos ayudan a registrar las ventas que se realizan y las adquisiciones de productos, obteniendo así un mayor conocimiento acerca de la situación financiera de la misma, denotando que los dos procesos son primordiales para un adecuado manejo de ingresos y egresos de productos, lo que ayuda a un mayor crecimiento financiero de Bravo Polo Cía. Ltda., así llegando a la conclusión que al llevar un adecuado control de facturación y bodega se podrá tener información exacta acerca de los productos y sus ventas, disminuyendo al mínimo la dificultad de generar los formularios para el pago de impuestos y se tendrá un mejor conocimiento acerca de la situación económica de la empresa, lo cual ayudará a disminuir considerablemente los retrasos en las tareas diarias tanto en bodega como en facturación y así brindar una mejor atención al cliente, permitiendo que la empresa pueda fortalecerse económicamente día a día.

1.03 Definición del Problema Central

Tabla 1

Matriz de análisis de fuerzas T.

ANÁLISIS DE FUERZAS T					
Situación Empeorada	Situación Actual				Situación Mejorada
Pérdida de información y extravió de stock de productos.	Inadecuado control de inventarios de bodega y facturación.				Minimizar riesgos de fraude.
Fuerzas Impulsadoras	I	PC	I	PC	Fuerzas Bloqueadoras
Conocimiento de cantidad de stock de productos manera ambigua.	1	4	3	4	Poco énfasis en ordenar adecuadamente los archivos de facturación.
Control en los sistemas operativos para evitar copiar información en dispositivos de almacenamiento externo.	2	3	4	5	Errores al generar inventarios de bodega.
Implementación de circuito cerrado de cámaras de seguridad.	1	4	4	5	Demora en la respuesta por parte de bodega al generar un informe sobre la existencia de un producto.
Almacenamiento de los archivos físicos de facturación y bodega de una manera incorrecta.	1	4	4	5	Inadecuado control por parte de bodega al ingresar productos mediante compras realizadas a proveedores.
Honestidad de los empleados en el manejo de stock de productos.	1	4	3	1	Desconocimiento de los clientes y sus compras

Nota: En la tabla se explica las fuerzas impulsoras y bloqueadoras que existen con respecto a la problemática central del proyecto.

Análisis de Matriz de Fuerza T:

En esta tabla se puede observar que la situación empeorada de la matriz es Pérdida de información y extravió de stock de productos, lo que pasaría si no existieran las siguientes fuerzas impulsoras que evitan que se de dicha situación:

- Conocimiento de cantidad de stock de productos manera ambigua.
- Control en los sistemas operativos para evitar copiar información en dispositivos de almacenamiento externo.
- Implementación de circuito cerrado de cámaras de seguridad.
- Almacenamiento de los archivos físicos de facturación y bodega de una manera incorrecta.
- Honestidad de los empleados en el manejo de stock de productos.

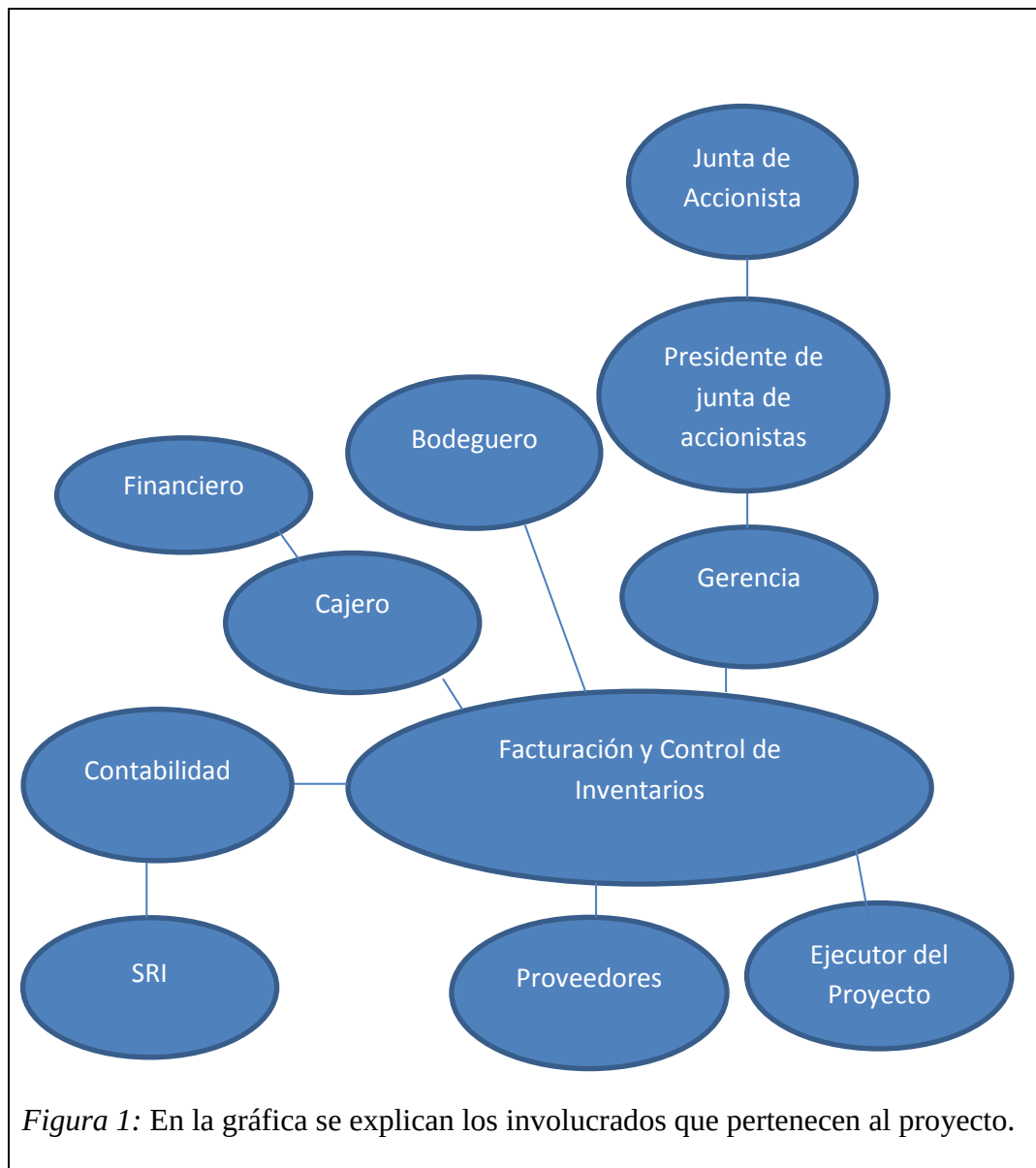
También se quiere llegar a minimizar riesgos de fraude pero las fuerzas bloqueadoras impiden que esta situación se dé:

- Poco énfasis en ordenar adecuadamente los archivos de facturación.
- Errores al generar inventarios de bodega.
- Demora en la respuesta por parte de bodega al generar un informe sobre la existencia de un producto.
- Inadecuado control por parte de bodega al ingresar productos mediante compras realizadas a proveedores.
- Desconocimiento de los clientes y sus compras

Para lo cual se tiene la situación actual que es un inadecuado control de bodega y facturación.

Capítulo II: Análisis de Involucrados

2.01 Mapeo de Involucrados



Análisis de los Involucrados:

En la gráfica de mapeo de involucrados que describe gráficamente los implicados dentro del proyecto, estos serán los que se benefician de su implantación, como podemos observar cada grafica esta ordenada según corresponda, para una mejor comprensión de la estructura de los beneficiarios y actores del proyecto.

2.02 Matriz de Análisis de los Involucrados

Tabla 2

Matriz de Análisis de los involucrados.

Actores Involucrados	Interés sobre el problema Central	Problemas Percibidos	Recursos, Mandos y Capacidades	Interés sobre el Proyecto	Conflictos Potenciales
Bodeguero	Llevar un adecuado control del inventario de los productos.	Demasiado stock de productos de venta mínima.	R1: Humano R2: Tecnológico C1: Comunicación Asertiva	Generar kardex en un tiempo óptimo y sin errores.	Errores al generar kardex de productos.
Gerente	Conocer el estado financiero de compra y venta de productos.	Dificultad al generar reporte de compras y ventas de productos.	R1: Humano R2: Tecnológico C1: Comunicación Asertiva C2: Liderazgo	Obtener reportes de los productos de mayor demanda.	Recopilación de datos.
SRI	Pago de impuestos a tiempo.	Retraso para generar formularios de declaración de impuestos.	M1: Ley Tributaria	Tener la información necesaria para generar los formularios de pago de impuestos.	Errores al generar facturas.

Cajero	Mayor organización al archivar facturas.	Lentitud al emitir facturas.	R1: Humano R2: Tecnológico C1: Comunicación Asertiva.	Disminuir el tiempo de espera de un cliente para recibir su factura.	
Ejecutor del Proyecto	Verificar los problemas existentes	Un inadecuado manejo de la información de facturación e inventarios de bodega	R1: Humano R2: Tecnológico C1: Comunicación Asertiva	Disminuir el tiempo al realizar la facturación aumentar la fiabilidad de la información obtenida del proceso anteriormente mencionado y de los productos almacenados en bodega	Los dueños de los procesos no entreguen la información verdadera.
Proveedores	Retenciones elaboradas correctamente.	Demora al generar las tenciones.	R1: Humano R2: Tecnológico C1: Comunicación Asertiva	Rapidez al generar pago de facturas.	

Contabilidad	Conocer reportes de las cuentas generadas en facturación.		R1: Humano R2: Tecnológico C1: Comunicación Asertiva M1: Mandato	Conocimiento de los movimientos financieros realizados en facturación.
Financiero	Conocer la cantidad de facturas emitidas y sus valores.	Dificultad al generar reportes de facturas emitidas y sus valores.	R1: Humano R2: Tecnológico C1: Comunicación Asertiva	Generar reportes de ventas de productos y cuentas por cobrar.

Nota: En la tabla se explica los involucrados que existen con respecto a la problemática central del proyecto.

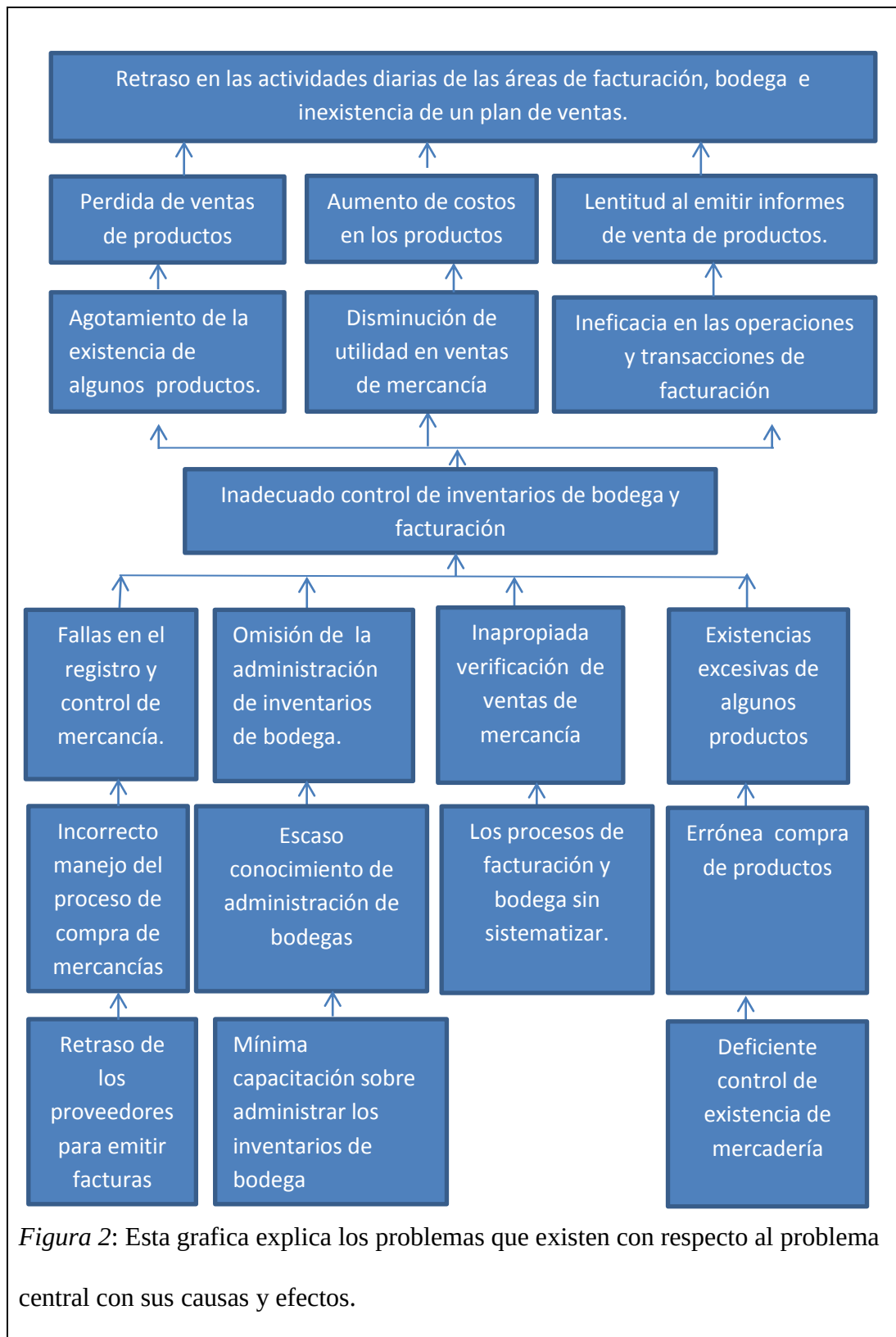
Análisis de la Matriz de Involucrados:

En esta tabla se muestra los diferentes aspectos de cada uno de los involucrados dentro del proyecto ya sea en el interés sobre el proyecto o sobre el problema.

Así como el cajero, bodeguero, gente, proveedor, contador tienen una gran expectativa sobre el problema del proyecto, ya que ayudara a dar solución a la problemática de la empresa permitiendo que esta surja y crezca a nivel económico.

Capítulo III: Problemas y Objetivos.

3.01 Árbol de problemas.



Análisis del Árbol de Problemas:

Esta grafica explica los problemas que existen con respecto al problema central con sus causas y efectos, las cuales parten de que existe un inadecuado control de inventarios de bodega y facturación dando como resultado el retraso en las labores diarias en las dos areas mencionadas anteriormente.

Los escasos conocimientos de administración de bodega, el mal manejo de ingresos de mercancía, el desconocimiento de cómo administrar una bodega de una manera adecuada, una inapropiada verificación de ventas de productos han dado como resultado un inadecuado control de inventarios de bodega y facturación.

3.02 Árbol de objetivos.

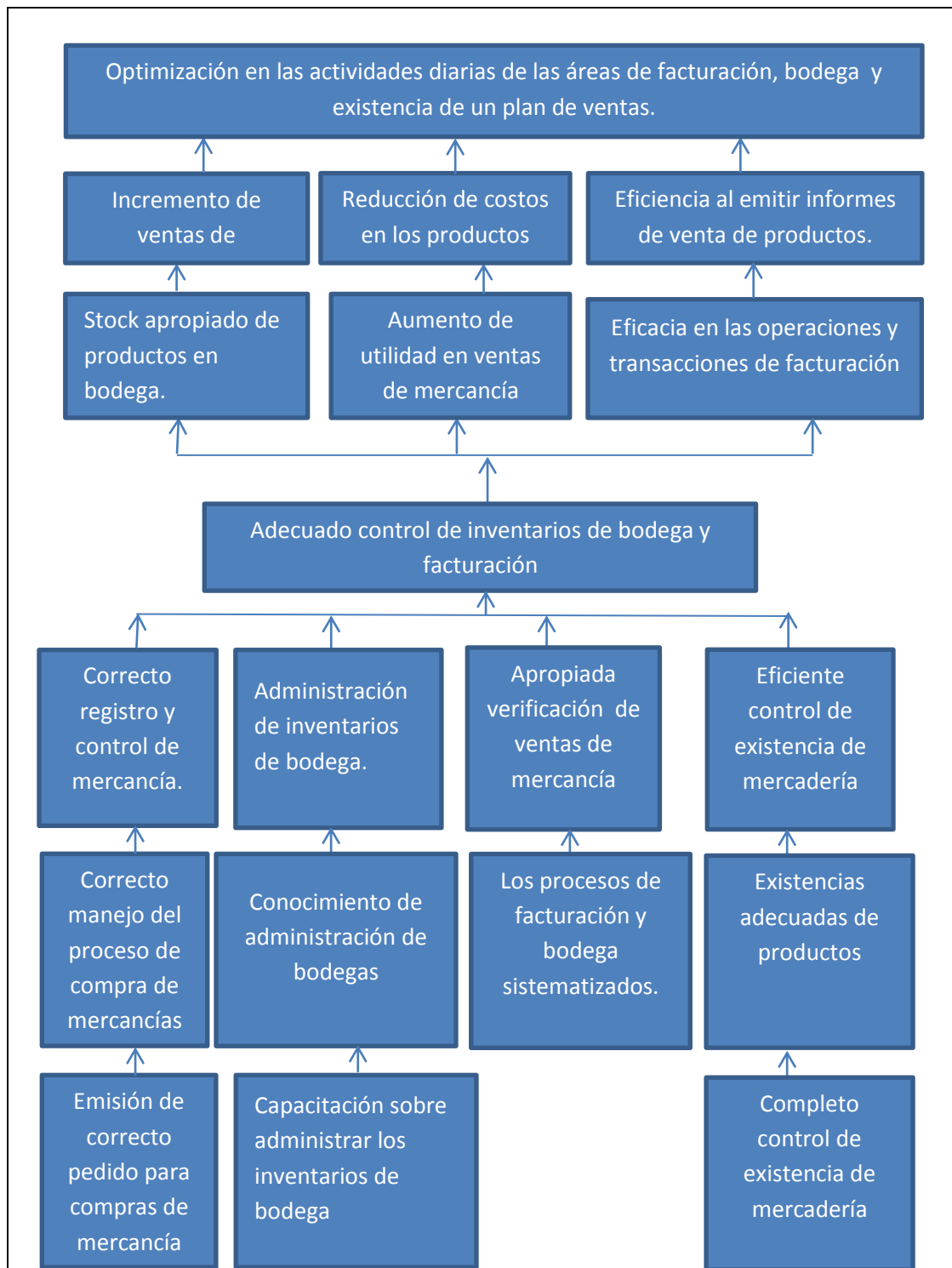


Figura 3: En esta grafica se muestran los objetivos del proyecto que desean alcanzarse para que se tenga un adecuado control de bodega y facturación lo que permitirá agilizar las actividades diarias dentro las dos áreas.

Análisis del Árbol de Objetivos:

Para este proyecto se ha delimitado cuatro objetivos principales los cuales se pretenden cumplir a cabalidad dentro de la implantación del proyecto:

- Tener un correcto registro y control de mercancía.
- Administrar adecuadamente los inventarios de bodega.
- Tener una apropiada verificación de ventas de mercancía.
- Ser eficientes al controlar la existencia de mercadería.

Estos mencionados objetivos ayudaran a llegar a un adecuado control de inventarios de bodega y facturación.

Además con los componentes y actividades que se generan en dicho árbol ayudaran a que el proyecto llegue a buen término.

Capítulo IV: Análisis de Alternativas.

4.01 Matriz De Análisis De Alternativas

Describe las alternativas que se tiene en el proyecto con sus objetivos.

Ver Anexo A.01

Conclusión de la matriz de análisis de alternativas

- El correcto registro y control de mercancía es un proceso que es un objetivo principal del proyecto y su calificación es alta, por su inferencia en un apropiado manejo de costos de productos.
- La Administración de inventarios de bodega es un objetivo principal que tiene una calificación alta ya que permitirá tener un stock apropiado de productos en bodega
- Tener Apropiada verificación de ventas de mercancía es un objetivo principal que tiene calificación alta, ya que esto ayudara al aumento de utilidad en ventas de mercancía.
- Eficiente control de existencia de mercadería es un objetivo principal que tiene calificación alta, generara una eficacia en las operaciones y transacciones del área facturación.

4.02 Matriz de Análisis de Impacto de los Objetivos

Tabla 3

Análisis de impacto de los objetivos.

Objetivo	Factibilidad de lograrse (alta-media-baja)			Impacto de Género (alta-media-baja)			Relevancia (alta-media-baja)			Sostenibilidad (alta-media-baja)			Total
	4	2	1	4	2	1	4	2	1	4	2	1	
Correcto registro y control de mercancía.	Es aceptable y conveniente para los beneficiarios.			Factibilidad de manejo para ambos géneros para los dos géneros.			Responde a las expectativas de los beneficiarios. Es una prioridad sentida por los beneficiarios.			Fortalece la participación de los beneficiarios. Fortalece a la Organización. Se incrementara métodos para un correcto control y registro de mercancía.			32 puntos 2 a 10 Baja 11 a 21 Media 22 a 32 Alta
	Existe y se conoce la tecnología adecuada para su ejecución.												
	Se cuenta con soporte institucional.												
Administración de inventarios de bodega.	Tener un adecuado stock de productos			Incremento de trabajo en equipo de la compañía.			Resolver conflictos presentados en el transcurso de las actividades. Tener una adecuada administración de bodega.			Fortalece una adecuada administración de bodega. Mejorar cada vez más la administración de bodega			28 puntos 2 a 10 Baja 11 a 21 Media 22 a 32 Alta
	Utilizar procesos adecuados para realizar las actividades.												
	Disminución de pérdidas de ventas y clientes												

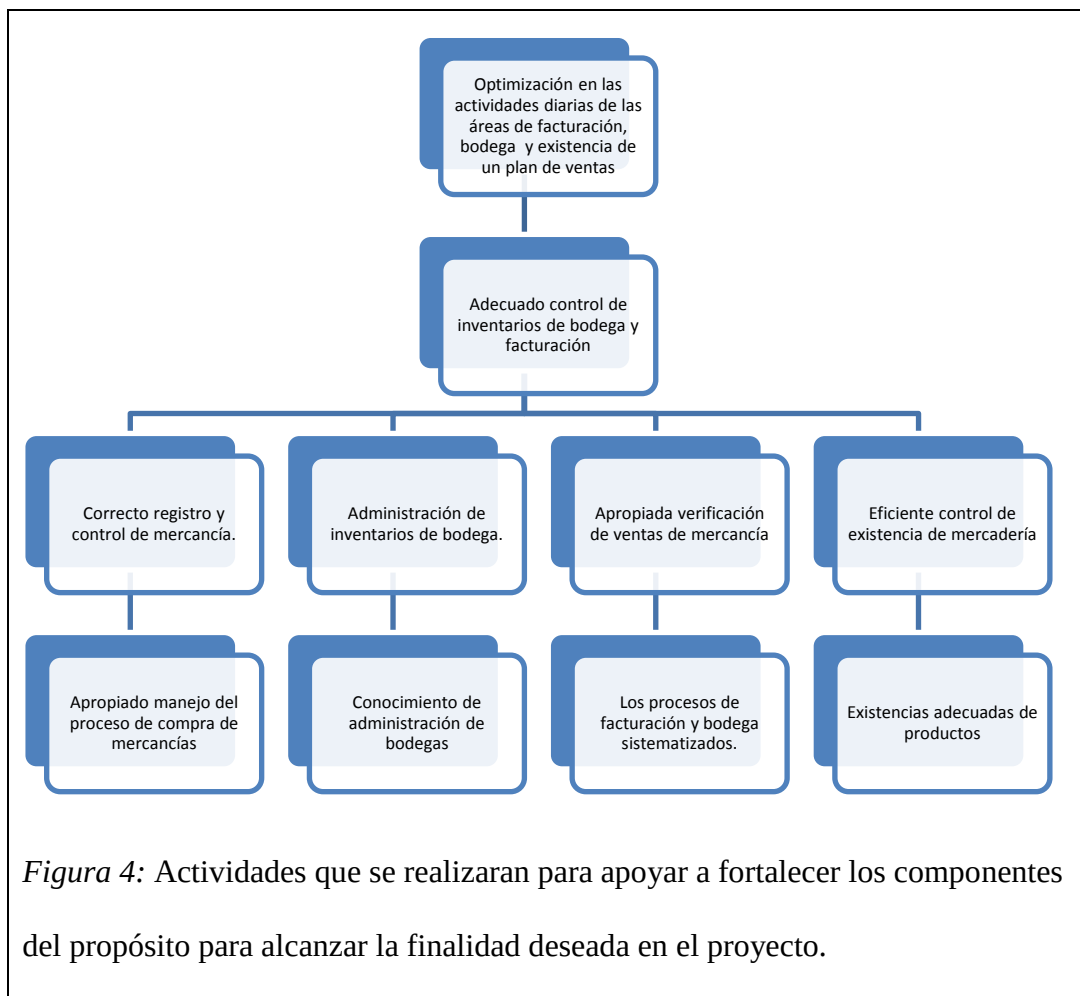


Objetivo	Factibilidad de lograrse (alta-media-baja)			Impacto de Género (alta-media-baja)			Relevancia (alta-media-baja)			Sostenibilidad (alta-media-baja)			Total
	4	2	1	4	2	1	4	2	1	4	2	1	
Apropiada verificación de ventas de mercancía	Aumentar la utilidad en ventas de la empresa.			Mayor interacción entre Jefe de bodega y Cajera.			Manifestar las posibles soluciones de un determinado inconveniente en el transcurso de las actividades.			Generar reportes de ventas de productos y verificar los precios.			28 puntos
	Generar procesos convenientes para la verificación de productos.						Incrementar métodos de verificación de ventas.			Implementar métodos de verificación de ventas.			2 a 10 Baja
													11 a 21 Media
							Sistematizar los procesos de facturación y bodega.						22 a 32 Alta
Eficiente control de mercancía	Realizar correcto ingreso de las compras de mercancía.			Minimizar desacuerdos entre el área de bodega y facturación al interactuar el cajero con el bodeguero.			Analizar la información y lo que se necesita para generar correcto reporte de existencias de productos.			Capacitar al personal sobre el manejo adecuado kardex de bodega.			32 puntos
	Verificar los datos obtenidos del reporte de stock de bodega						Tener existencia adecuada de productos.			Manifestar anomalías en el transcurso del registro de ingresos de productos			2 a 10 Baja
													11 a 21 Media
													22 a 32 Alta

Nota: Aquí se describe la capacidad de impacto de los objetivos.

Conclusión de la matriz de impacto de objetivos.- Después de un análisis de cada uno de los objetivos, se determina la factibilidad de lograrse, la relevancia, la sostenibilidad y se define la categoría de impacto que refleja cada objetivo en el proyecto.

4.03 Diagrama de Estrategias



4.04 Matriz de Marco Lógico

Se resume el análisis del marco lógico como es las actividades, los componentes, el propósito y el fin del proyecto.

Ver Anexo A.02

Capítulo V: Justificación Propuesta

5.01 Justificación

Los procesos automatizados, son factores de vital importancia en las compañías, en esta era se ha observado una gran innovación y desarrollo de tecnologías de la información que ha permitido la evolución de computadores que son capaces de producir sistemas de información a la sociedad con aplicaciones a los distintos sectores económicos, sociales, políticos, educativos, entre otros.

La base de toda empresa comercial es el control de los productos que entran y salen; de aquí la importancia del manejo del inventario por parte de la misma. Este manejo permitirá a la empresa mantener el control oportunamente, así como también conocer al final del periodo un estado confiable de la situación económica de la misma.

El control interno sobre los inventarios es importante, ya que los inventarios son el aparato circulatorio de una empresa de comercialización. Las compañías exitosas tienen gran cuidado de proteger sus inventarios.

Al igual que el proceso facturación es primordial para registrar las ventas realizadas y así conocer el estado financiero de la compañía.

Actualmente se puede afirmar que el proceso de contar y registrar datos financieros se desarrolla de una manera más simple y sencilla con el apoyo del contador, pero, es preciso aclarar que se siguen rigiendo por los principios establecidos para ejecutar la contabilidad empresarial.

La creación de este software que permita facturar y llevar el control del inventario de Bodega de la "Compañía Bravo Polo", lograra que los usuarios estén satisfechos por la rapidez al momento de solicitar información sobre la mercancía existente, ventas realizadas, compras, estados de los clientes y proveedores puesto que serán

revelados cuando lo desee, gracias a que existirá un adecuado control de facturación e inventarios de bodega por que los procesos anteriormente mencionados estarán sistematizados lo cual ayudara a un mejor desempeño diario por parte de los empleados de la empresa.

5.02 Análisis y diseño

5.02.01 Diagrama de Casos de Uso

Definición de actores

Cliente: Este actor es considerado como un actor externo debido a que es muy independiente del proyecto, es el que permite que la compañía crezca financieramente ya por medio de sus compras ingresa dinero a la empresa.

Bodeguero: Dicho actor cumple una de las funciones más importantes, ya depende de él para que los registros de ingreso y egreso de mercadería se manejen de manera correcta.

Cajero: Es aquel que cumple la función de emitir las facturas de venta a los clientes.

Proveedor: Es considerado como un actor externo debido a que es muy independiente del proyecto, es el encargado de proveer a la empresa de los productos que necesita para la venta.

Definición de casos de Uso

Ingresos al sistema: Caso de uso que permitirá el acceso de los actores al sistema, introduciendo su usuario y contraseña.



Emitir Factura: Mediante este caso de uso se describe el proceso de vender productos con la emisión de su factura respectiva.

Devolución de producto: En dicho caso de uso se describe el proceso para emitir notas de créditos cuando haya devolución de productos.

Generar Nota de Crédito: Caso de uso que permitirá que el cliente al devolver un producto pueda comprar otro por la cantidad devuelta del producto.

Compra de Producto: Caso de uso que describe el procedimiento que se realiza para comprar productos a proveedores e ingresarlos en bodega.

Entrega Productos: En este caso de uso se procede a entregar los productos por parte del bodeguero al cliente.

Generación de retención: Es importante para toda empresa ser agente de retenciones en este caso de uso se describe el proceso para generar retención.

Diagramas de Casos de Uso

Describe gráficamente el diagrama de casos de usos y las funciones que cumplen.

Ver Anexo.04

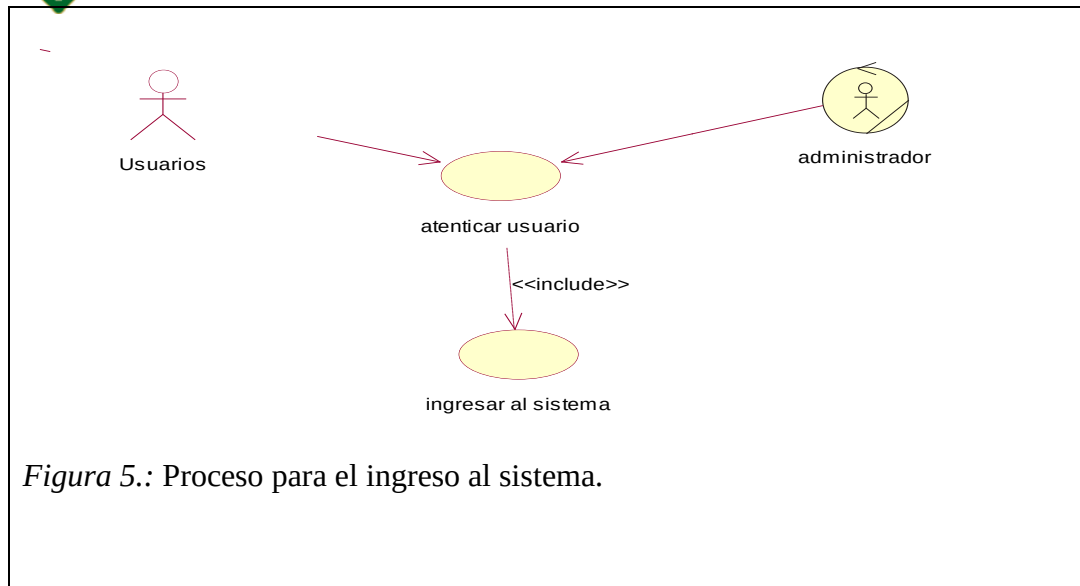


Figura 5.: Proceso para el ingreso al sistema.

Tabla 4

Proceso de ingreso al sistema. - CU01.

NOMBRE:	Ingresar al Sistema
ACTORES	Cajero
PRECONDICION:	Ejecutar el sistema
DESCRIPCION:	El usuario ejecuta el sistema en la ventana de inicio.
POST CONDICION:	Ingresar el nombre de usuario
NOMBRE:	Ingresar el nombre de usuario
ACTORES	Cajero
PRECONDICION:	Ingresar al sistema
El cajero luego de ingresar al sistema debe escribir:	
DESCRIPCION:	• Usuario
	• Contraseña
	En las cajas de texto correspondiente, una vez ingresado los datos debe hacer click en el botón

ingresar y pasa al siguiente caso de uso.

POST CONDICION: Verificar Usuario

NOMBRE: Verificar Usuario

ACTORES Administrador

PRECONDICION: Ingresa el nombre de usuario

DESCRIPCION: El administrador de la base de Datos verifica si el usuario se encuentra registrado Y su contraseña es la correcta.

Si los datos son correctos pasa al siguiente caso de uso, caso contrario envía un mensaje de error y vuelve al caso de uso ingreso del nombre de usuario.

POST CONDICION: Venta de productos

Nota: Proceso y condiciones para ingresar al sistema.

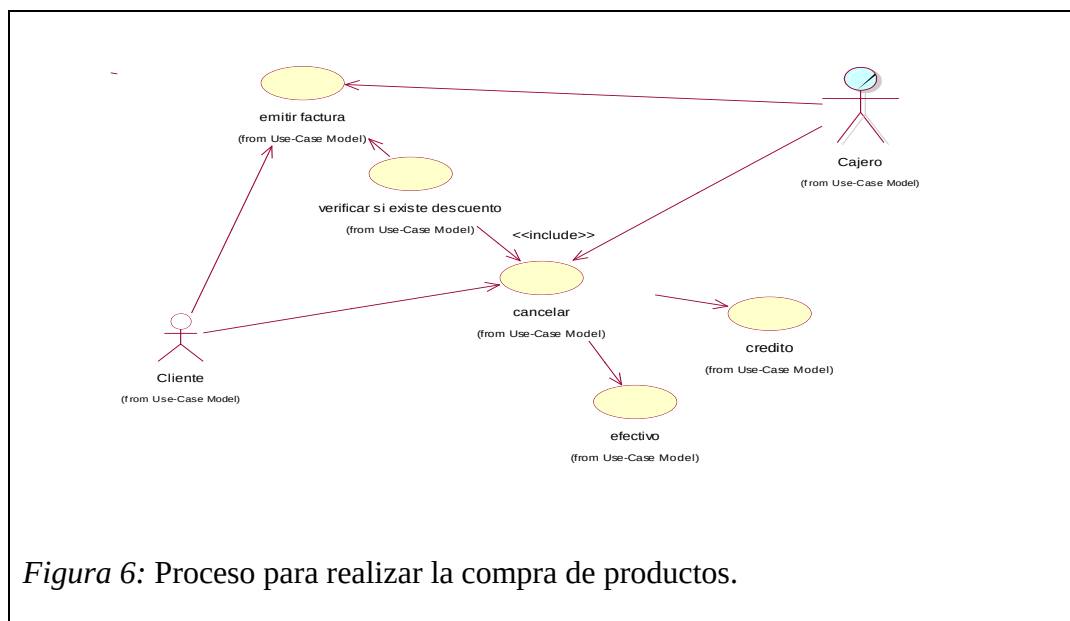


Tabla 5

Proceso para la venta de producto- CU02.

NOMBRE:	Venta de producto
ACTORES	Cajero
PRECONDICION:	Verificar Usuario
DESCRIPCION:	El cajero genera la venta
POST	Generación de la factura
CONDICIONES:	

Nota: Proceso y condiciones para la venta de productos.

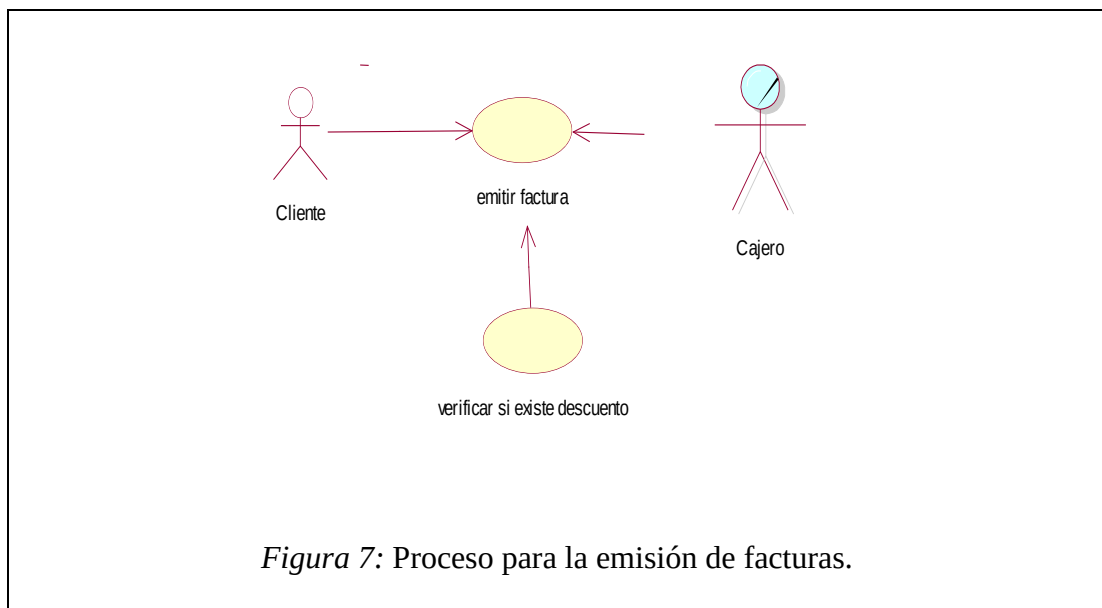


Tabla 6

Proceso para la generación de la factura- CU03.

NOMBRE:	Generación de la factura
ACTORES	Cajero
PRECONDICION:	Entrega de factura al cliente
DESCRIPCION:	El cajero emite la factura al cliente si se cumplen todas las condiciones necesarias para ello, caso contrario regresa al caso de uso anterior.
POST	Entrega del producto
CONDICIONES:	

Nota: Proceso y condiciones para la generación de factura.

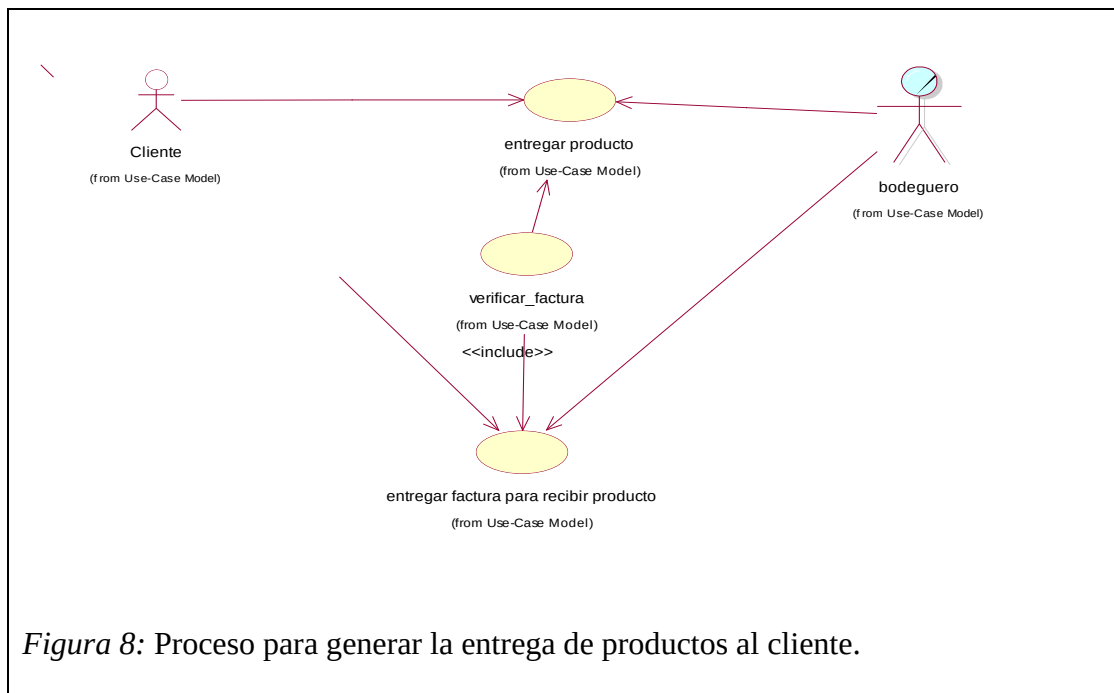


Figura 8: Proceso para generar la entrega de productos al cliente.

Tabla 7:

Proceso para la entrega del producto- CU04.

NOMBRE:	Entrega de producto
ACTORES	Bodeguero
PRECONDICION:	Entrega de producto
DESCRIPCION:	El jefe de Bodega realiza verificación correspondiente a la factura si las condiciones de la factura son correctas avanzan al siguiente caso de uso, caso contrario solicita que se verifique la factura.
POST	Genera factura
CONDICIONES:	

Nota: Proceso y condiciones para la entrega de producto.

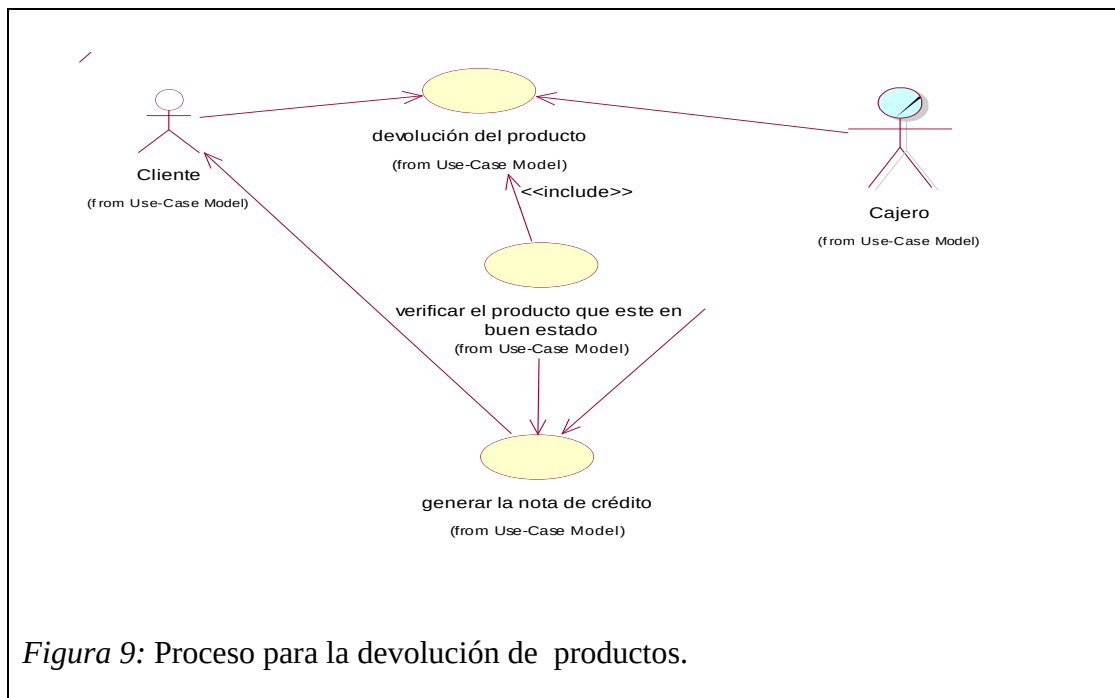


Tabla 8

Proceso de generación de nota de crédito - CU05.

NOMBRE:	Genera la nota de crédito
ACTORES	Cajero
PRECONDICION:	Genera nota de crédito
DESCRIPCION:	El cajero aprueba la devolución del producto y genera nota.
POST	Devolución de producto
CONDICIONES:	
NOMBRE:	Devolución de producto
ACTORES	Bodeguero
PRECONDICION:	Verificar Producto que este en buen estado
DESCRIPCION:	Bodeguero recibe el producto
POST	Compra de productos a proveedores
CONDICIONES:	

Nota: Proceso y condiciones para generar la nota de crédito.

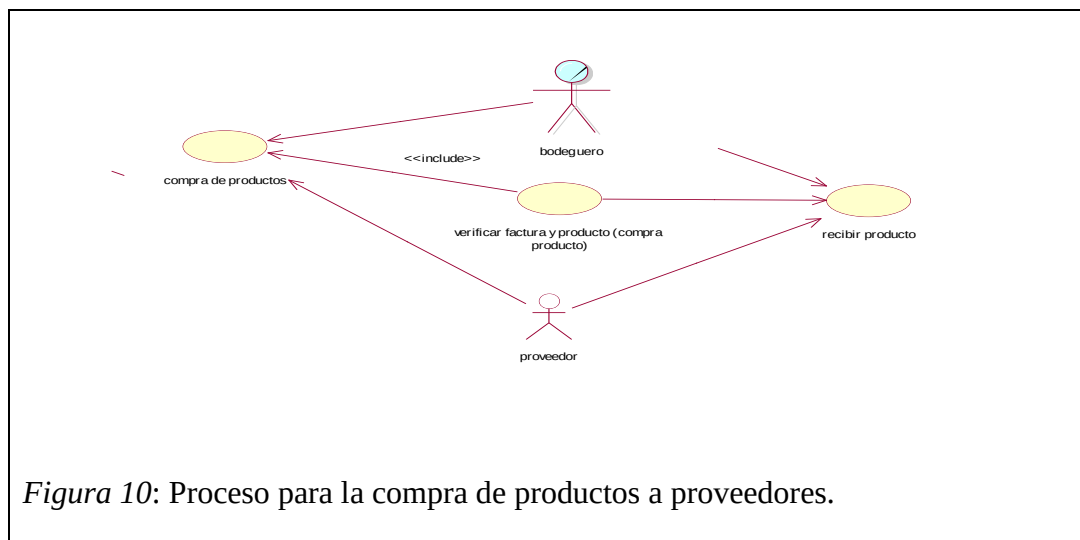


Tabla 9

Proceso de la compra de productos a proveedores. CU06.

NOMBRE:	Compra de productos a proveedores
ACTORES	Bodeguero
PRECONDICION:	Verificar Producto que este en buen estado y la factura correcta.
DESCRIPCION:	Bodeguero recibe el producto y la factura
POST CONDICIONES:	Generar retención.

Nota: Proceso y condiciones para la compra de productos a proveedores.

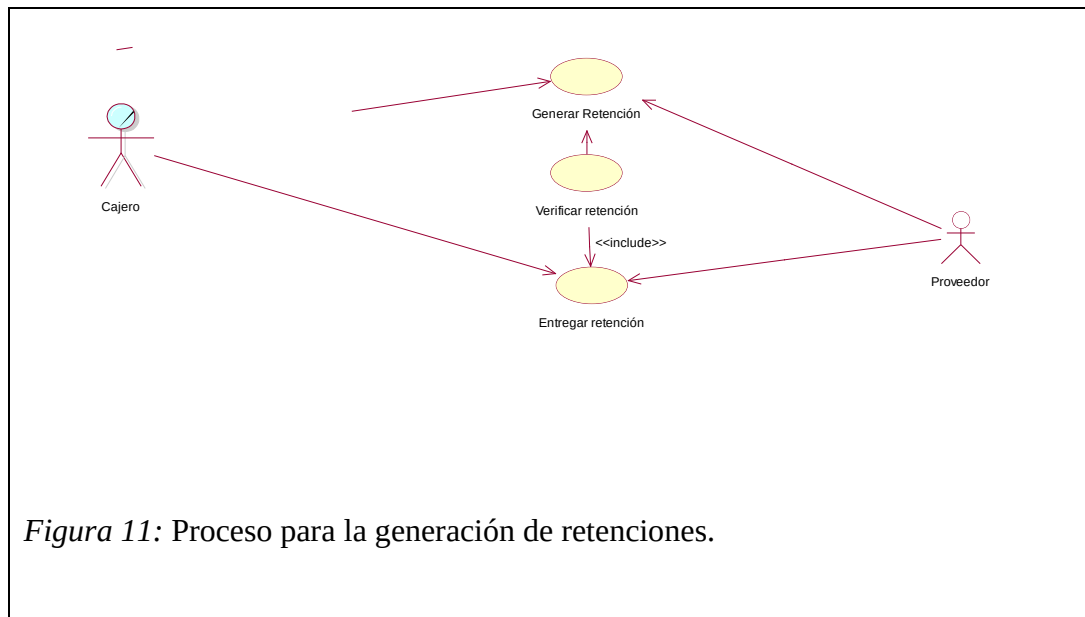


Tabla 10

Proceso para la compra de productos - CU07.

NOMBRE:	Generar retención.
ACTORES	Cajero
PRECONDICION:	Compra de productos
DESCRIPCION:	Se genera la retención con sus datos respectivos.
POST	Fin
CONDICIONES:	

Nota: Proceso y condiciones para la generación de retenciones.

5.02.02 Diagrama de Casos de Realización

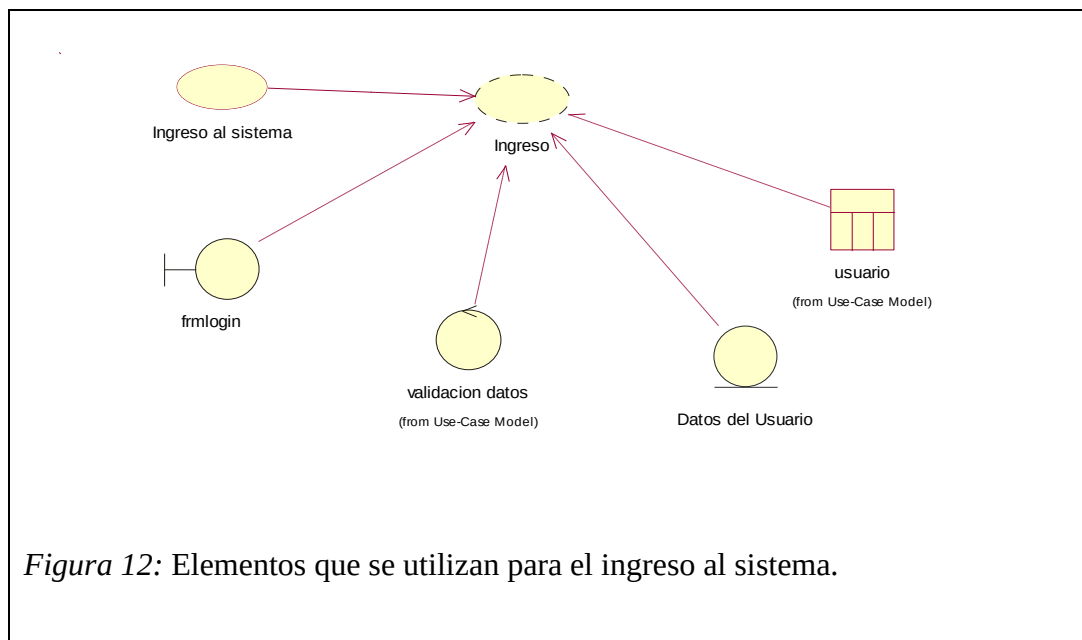


Figura 12: Elementos que se utilizan para el ingreso al sistema.

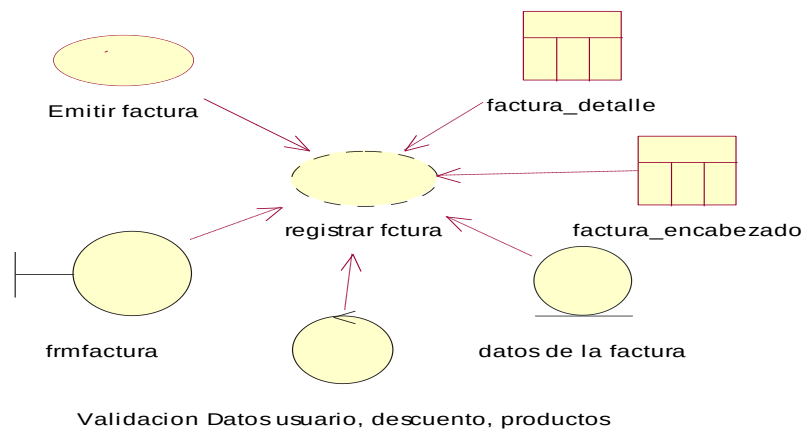


Figura 13: Elementos que se utilizan para la generación de factura.

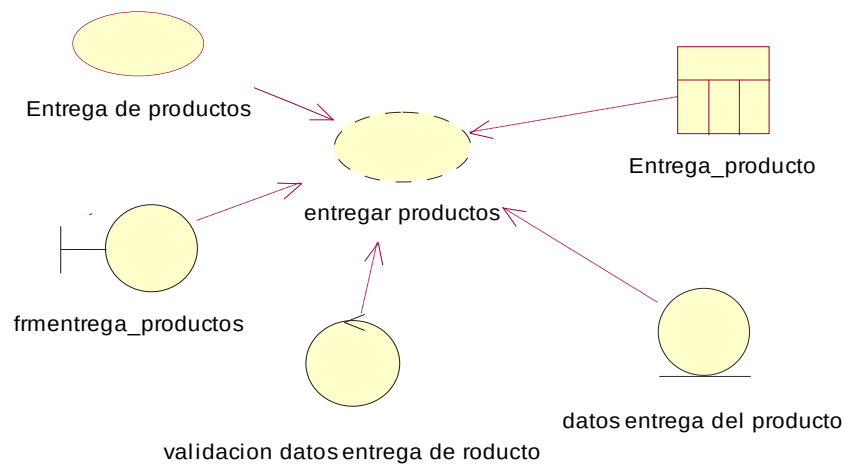


Figura 14: Elementos que se utilizan para la entrega de producto al cliente y generación de factura.

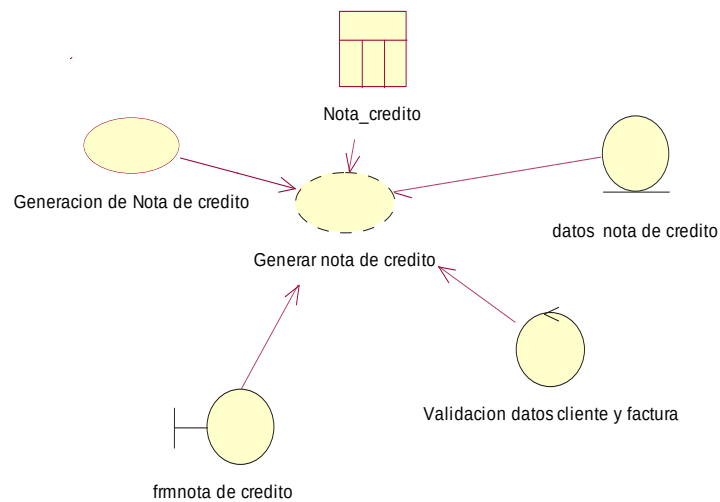


Figura 15: Elementos que se utilizan para realizar la generación de nota de crédito.

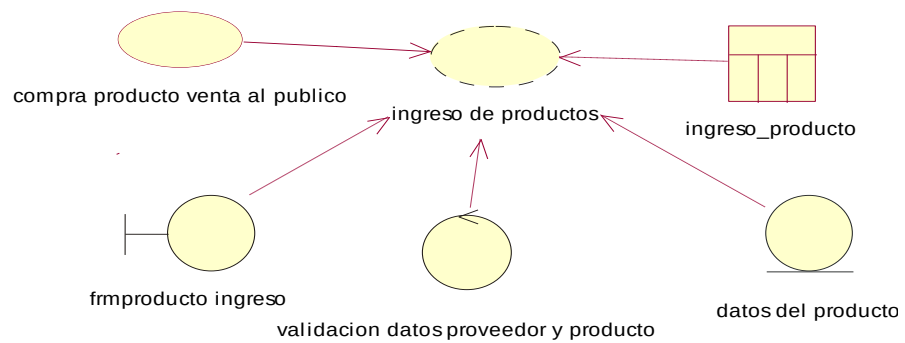


Figura 16: Elementos que se utilizan para la compra de productos a proveedores.

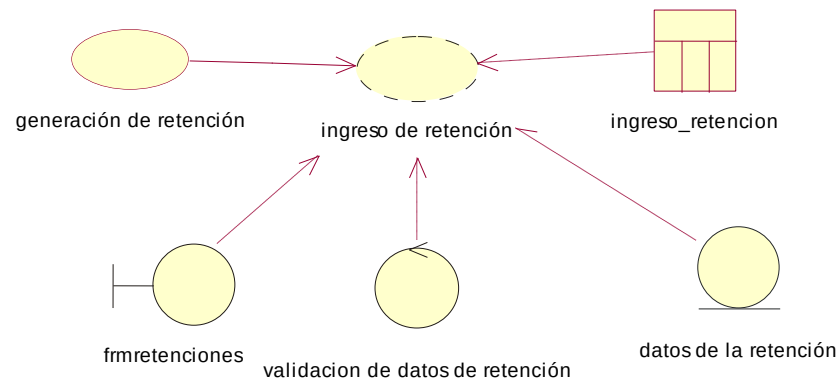


Figura 17: Elementos que se utilizan para la generación de retenciones.

5.02.03 Diagramas de Secuencia

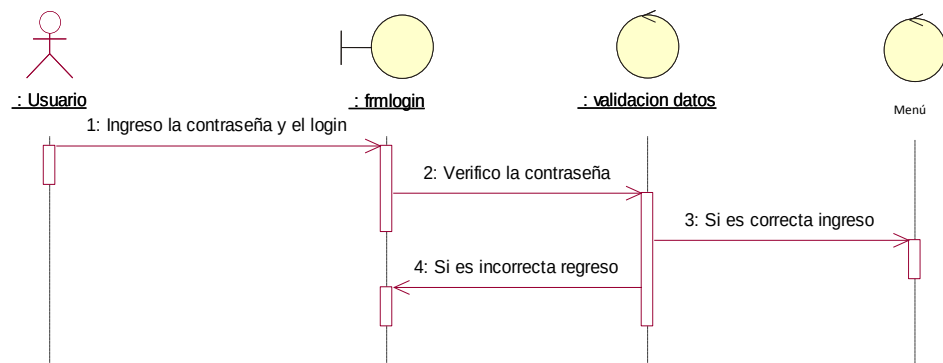


Figura 18: Proceso para ingresar al sistema.

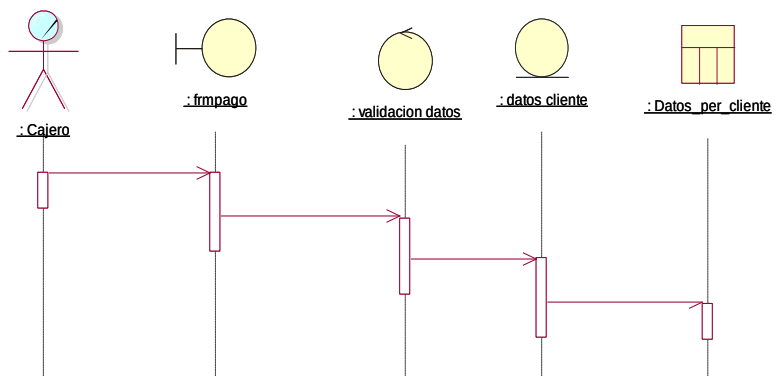


Figura 19:Proceso para realizar la venta de productos.

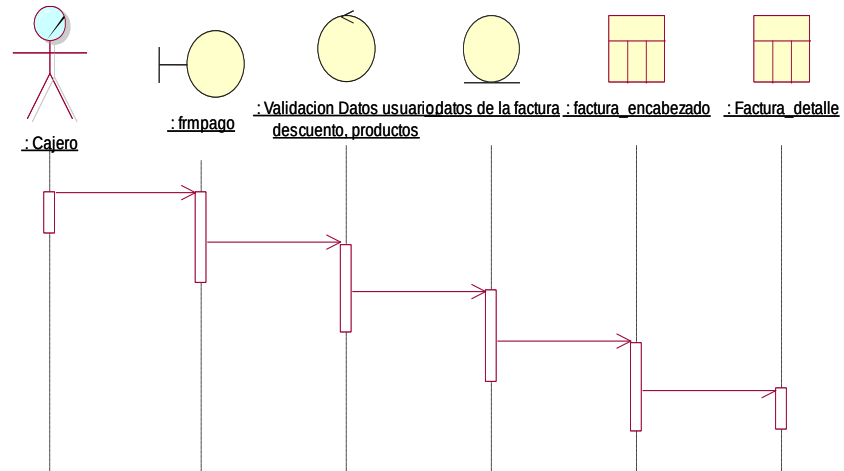


Figura 20:Proceso para la generación de facturas.

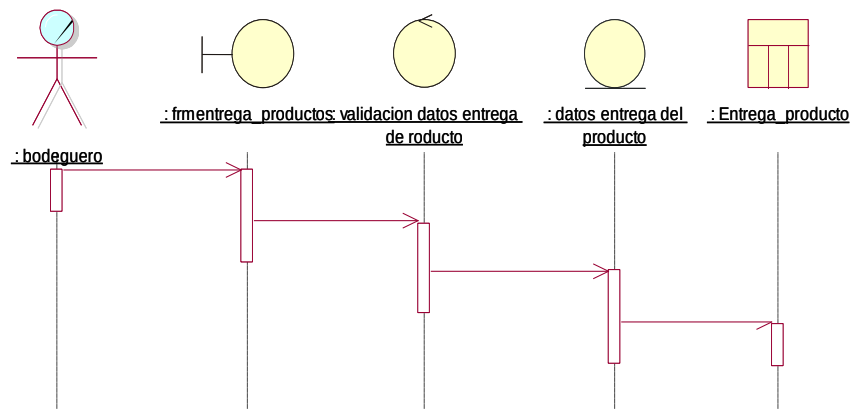


Figura 21: Proceso para realizar la entrega de producto a clientes.

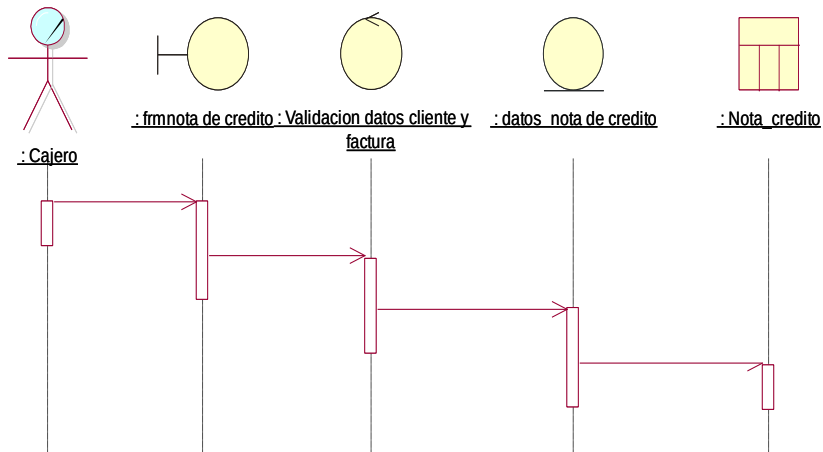


Figura 22: Proceso para la generación de notas de crédito.

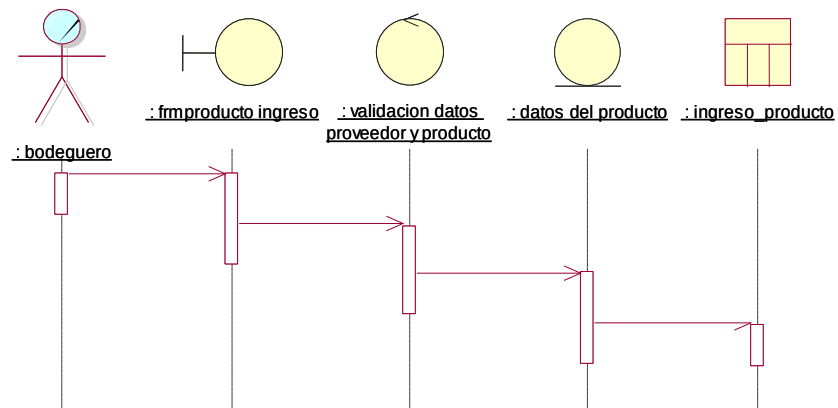


Figura23: Proceso para comprar de producto a proveedores.

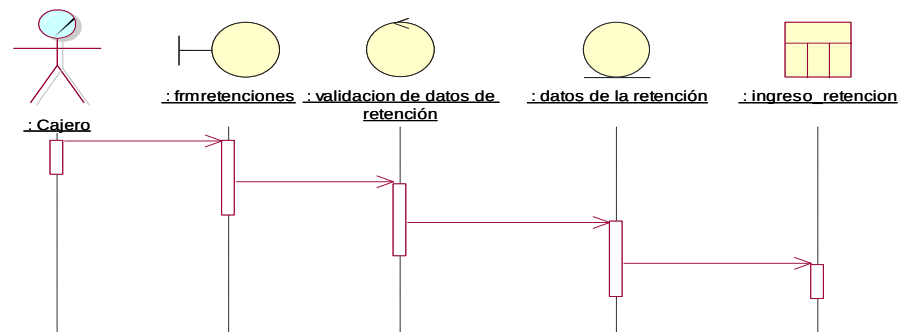


Figura 24: Proceso para la generación de retenciones.

Conclusión de los Diagramas de Secuencia.-Se realizan diagramas de secuencia para definir acciones que se pueden realizar en la aplicación; en las figuras anteriores se realiza la identificación de los eventos internos del sistema, es decir, la identificación de quien demanda que inicie un escenario, de esta forma podemos observar que el inicio lo marca el usuario ingresando al sistema.

5.02.04 Diagramas de Colaboración

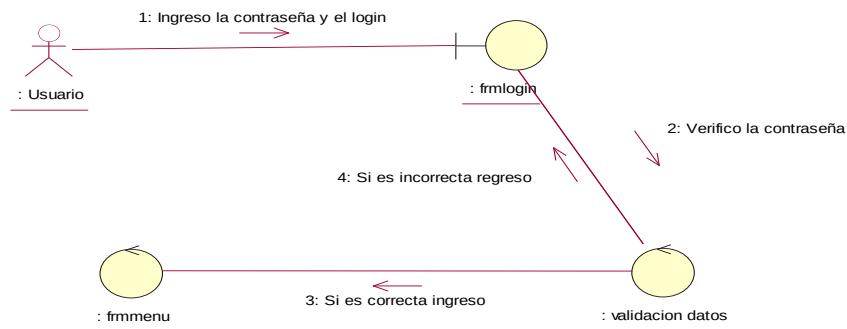


Figura 25: Proceso para ingresar al sistema.

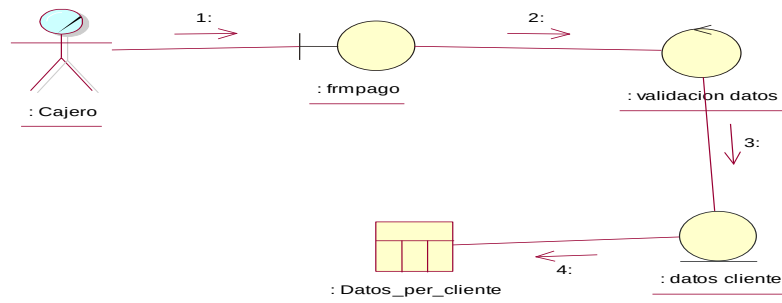


Figura 26: Proceso para realizar la venta de productos.

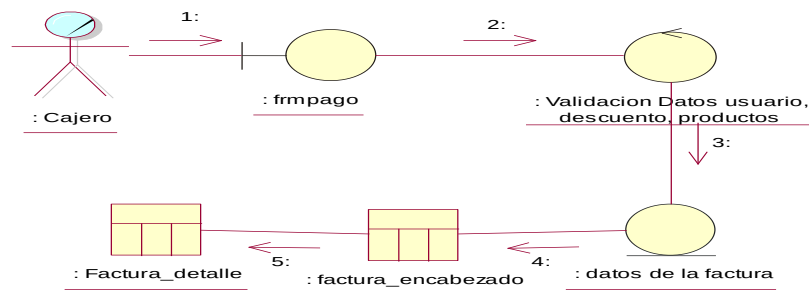


Figura 27: Proceso para generar facturas.

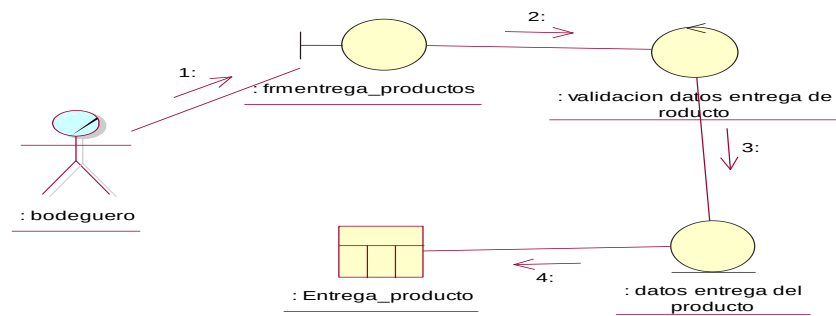


Figura 28: Proceso para realizar la entrega de productos a clientes.

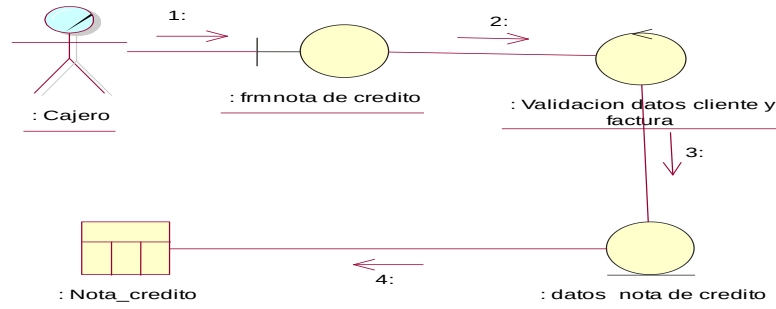


Figura 29: Proceso para generar las notas de crédito.

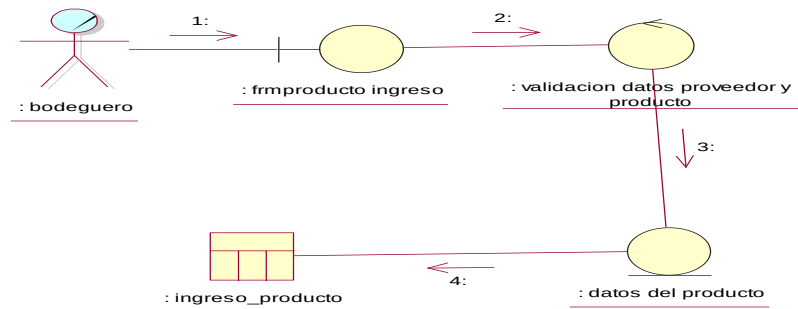


Figura 30: Proceso para realizar la compra de productos a proveedores.

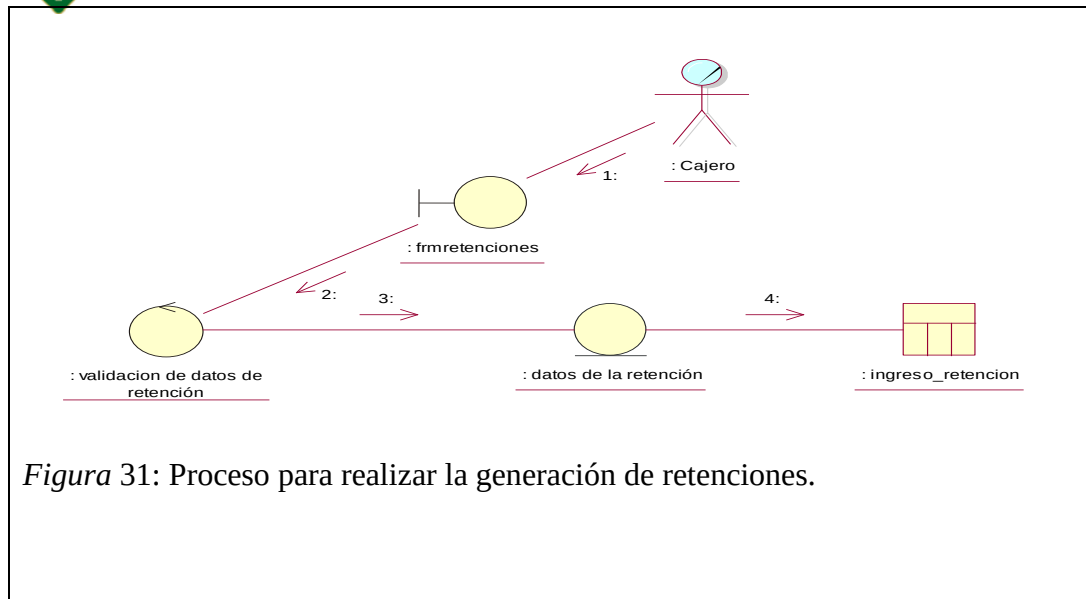
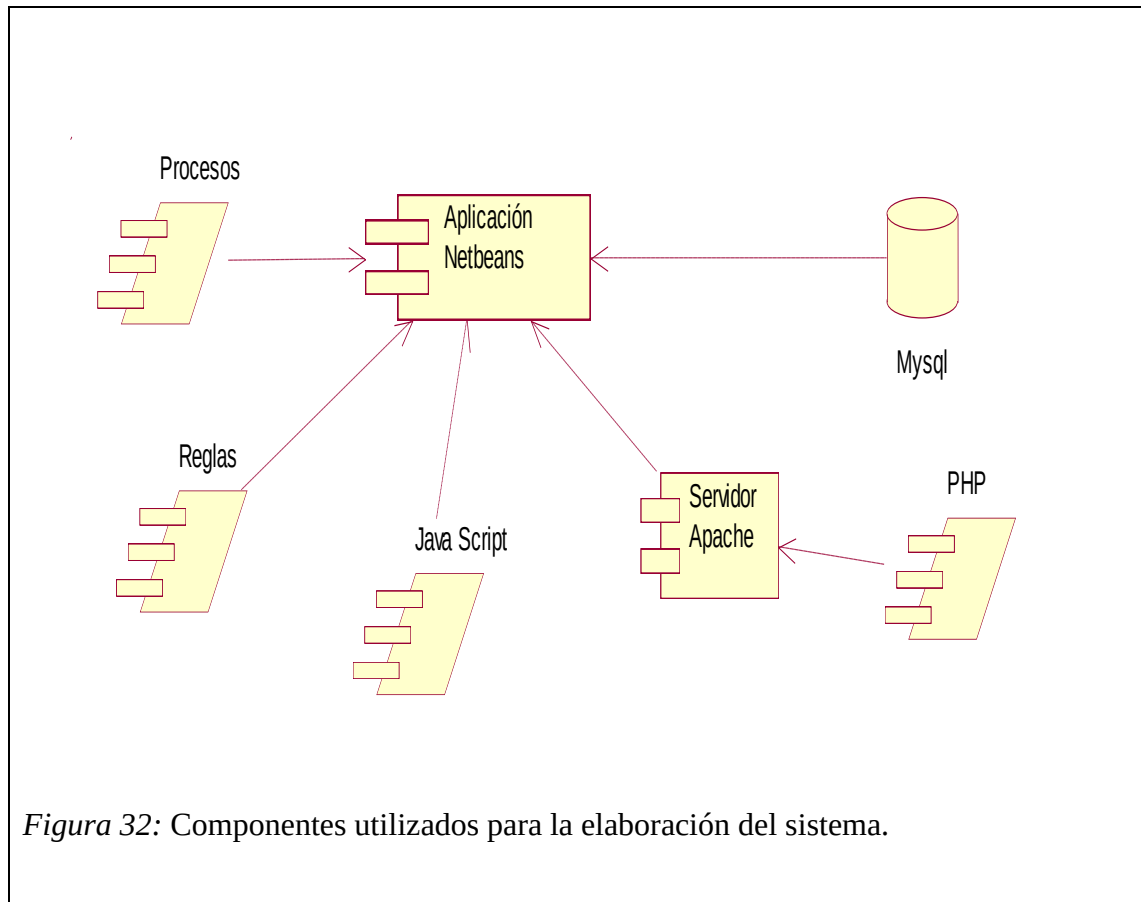


Figura 31: Proceso para realizar la generación de retenciones.

Conclusión de los Diagramas de Colaboración.-Se realizan diagramas de colaboración para definir las acciones que se pueden realizar en la aplicación; en las figuras anteriores se realiza la identificación de los eventos internos del sistema obteniendo un mejor entorno visual, es decir, la identificación de quien demanda que inicie un escenario, de esta forma podemos observar que el inicio lo marca el usuario ingresando al sistema.

5.02.05 Diagrama de Componentes



Conclusión del Diagrama de Componentes.- El diagrama de componentes representa la estructura física del código, los componentes del software, en este caso el propósito del diagrama de componentes es definir los módulos de software, hardware y las relaciones que existe entre sí; se observa en el grafico anterior que se tiene un componente que es el servidor apache el cual nos permite trabajar con el lenguaje de programación php ya que lo tiene incorporado, gracias a este lenguaje que en si es muy sencillo de aprender y no requiere de muchos requerimientos para trabajar en el gracias a esto se puede generar programas en muy corto tiempo y aplicando una estricta política de seguridad se pude obtener sistemas muy seguros, de la misma manera se trabaja para la generación de la interfaz gráfica con la aplicación de desarrollo netbeans la cual nos permite dar un acabado excelente a la

interfaz que el usuario manejara a más de eso nos permite trabajar con java script el cual permite generar aplicaciones amigables, atractivas y fáciles de manejar, para complementar se ha tomado en cuenta a la base de datos mysql la cual no necesita de requerimientos costosos y puede ser instalada en cualquier servidor sin que haya problema con el sistema operativo que utilice sin olvidar que es muy segura y no existe costo por utilizarla.

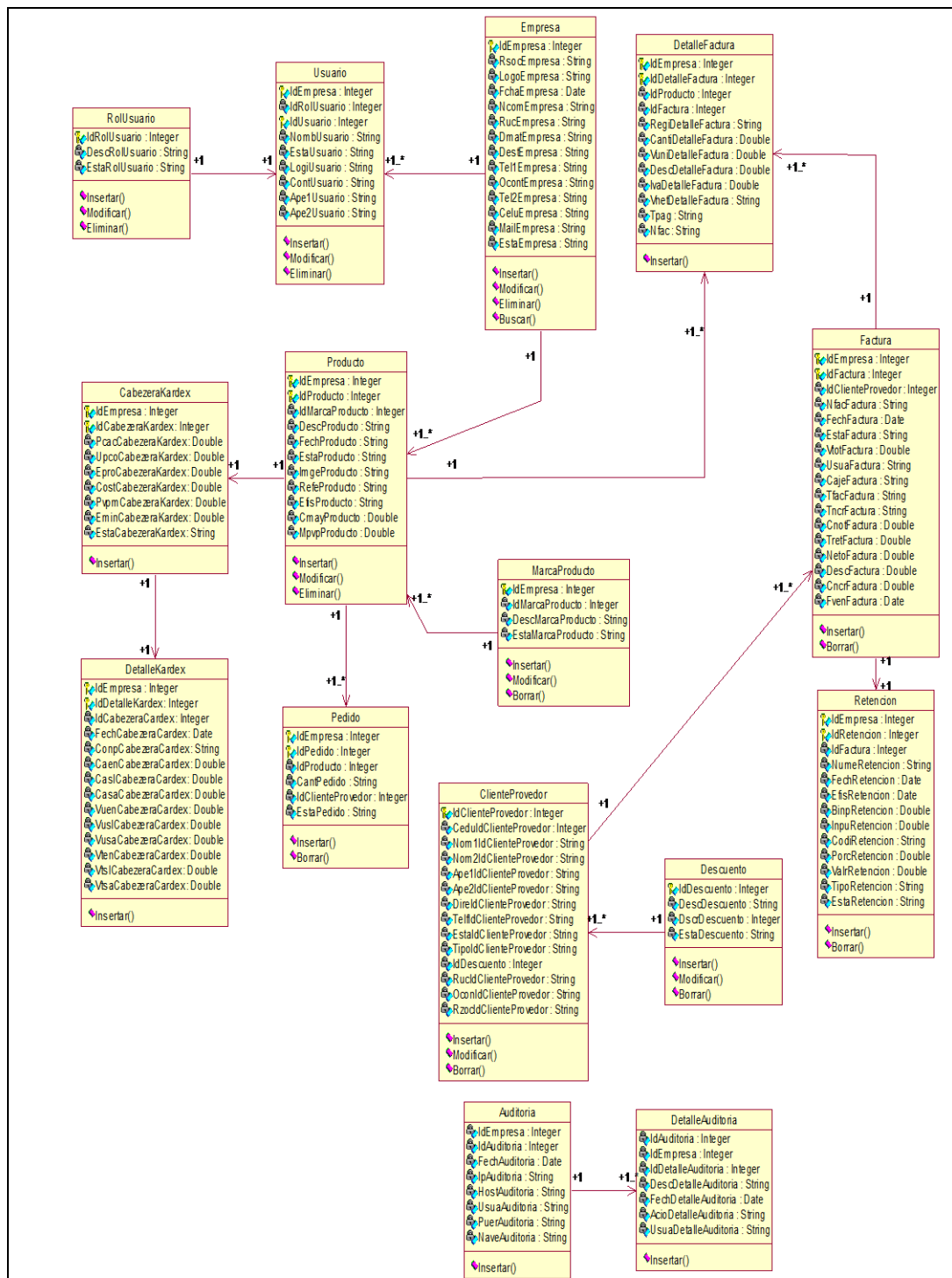
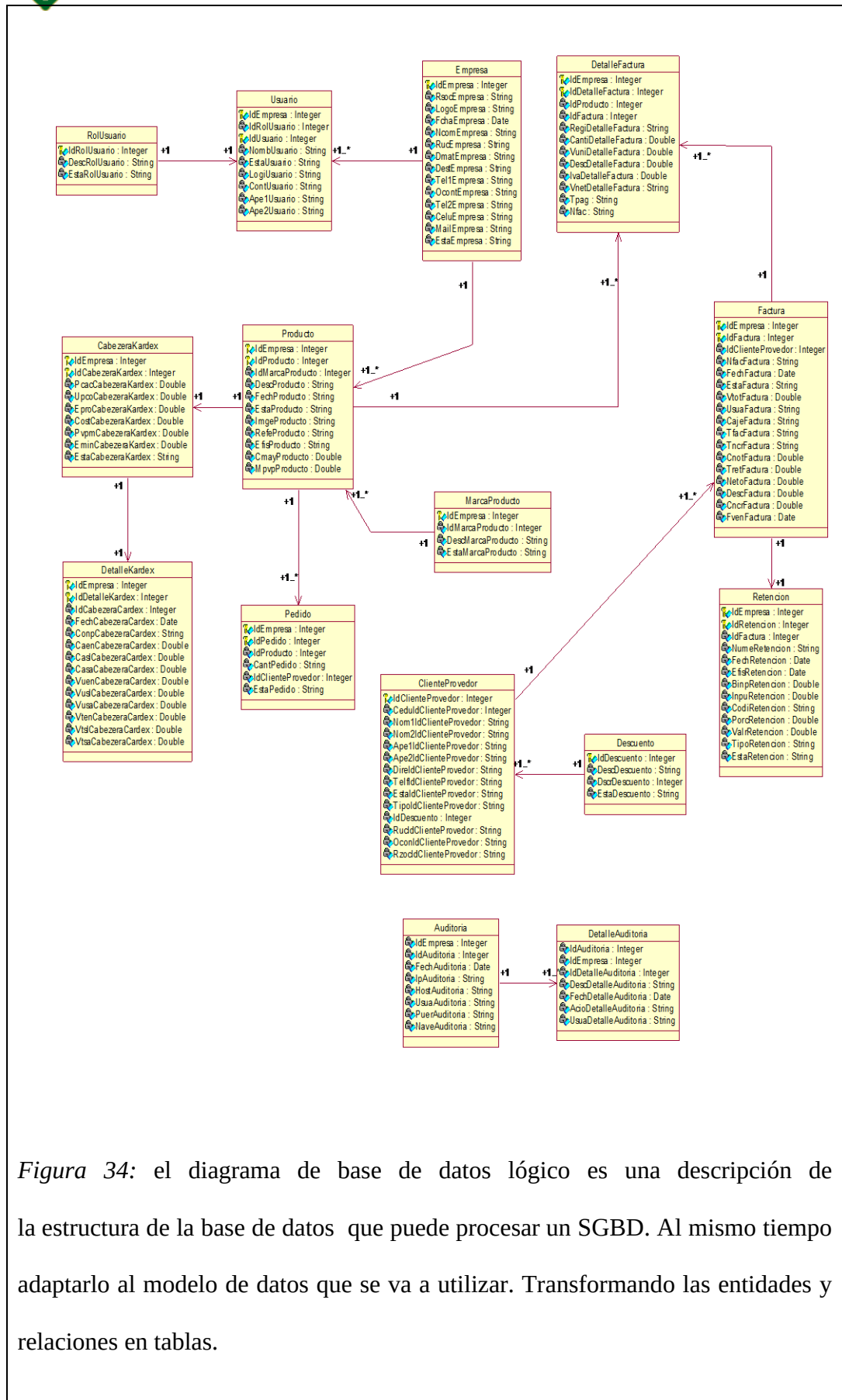


Figura 33: El diagrama de clases, muestra las diferentes clases que se crearan para la capa lógica del sistema a generar.



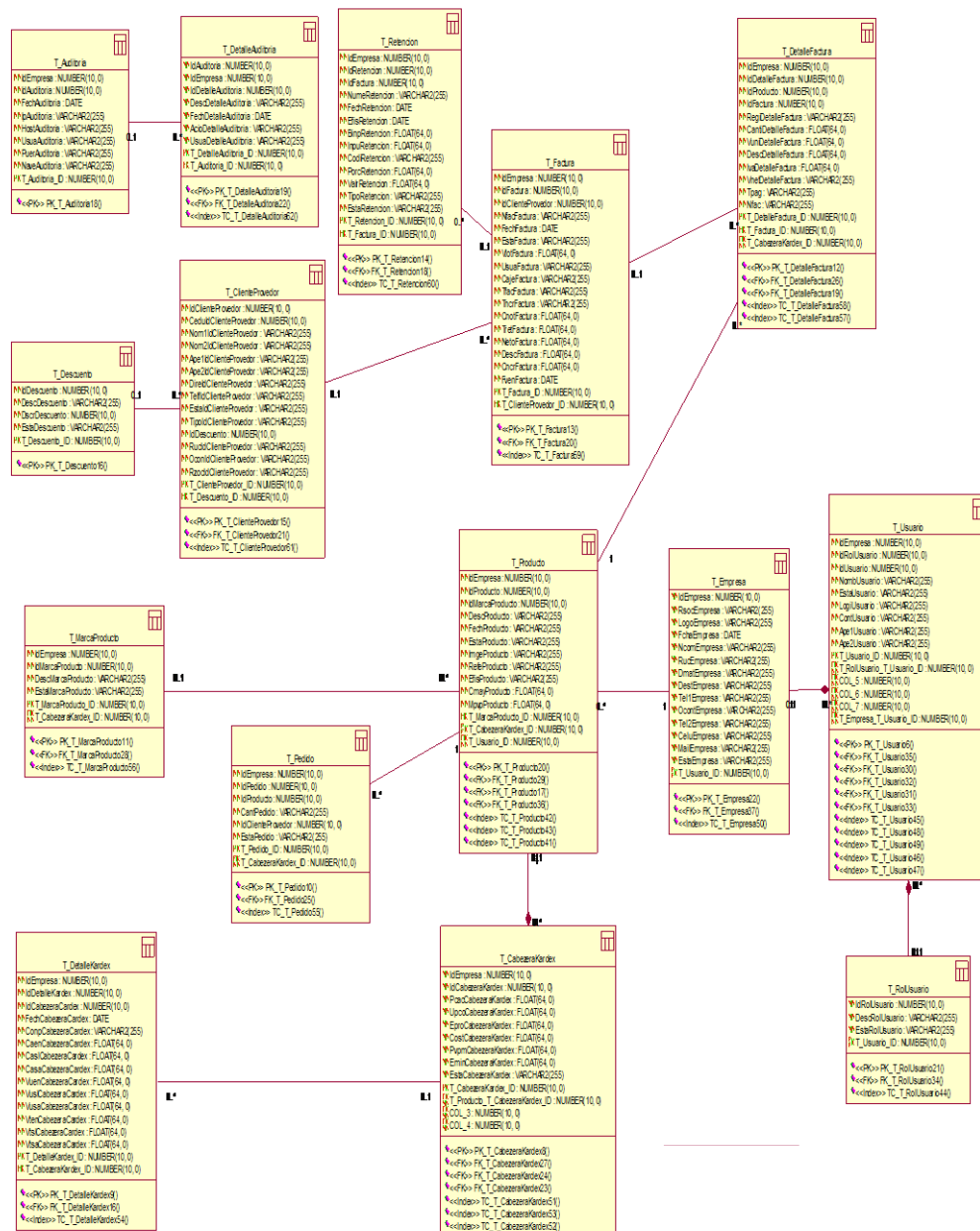


Figura 35: Este diagrama de base de datos físico tiene como objetivo conseguir una instrumentación, lo más eficiente posible, del esquema lógico, donde describe los componentes, de hardware y de software, que se desplegarán en el ambiente seleccionado. En este caso, empleando el gestor de la DDBB, se implementan las tablas de las DDBB con sus características, organización y estructuras de almacenamiento interno.

5.03 Desarrollo

5.03.01 Arquitectura del sistema

Arquitectura Lógica

Partiendo de que la Programación por Capas es una Arquitectura Cliente-Servidor, cuyo principal objetivo es separar La Lógica de Negocio de la Lógica de Diseño, como se muestra en la siguiente gráfica.

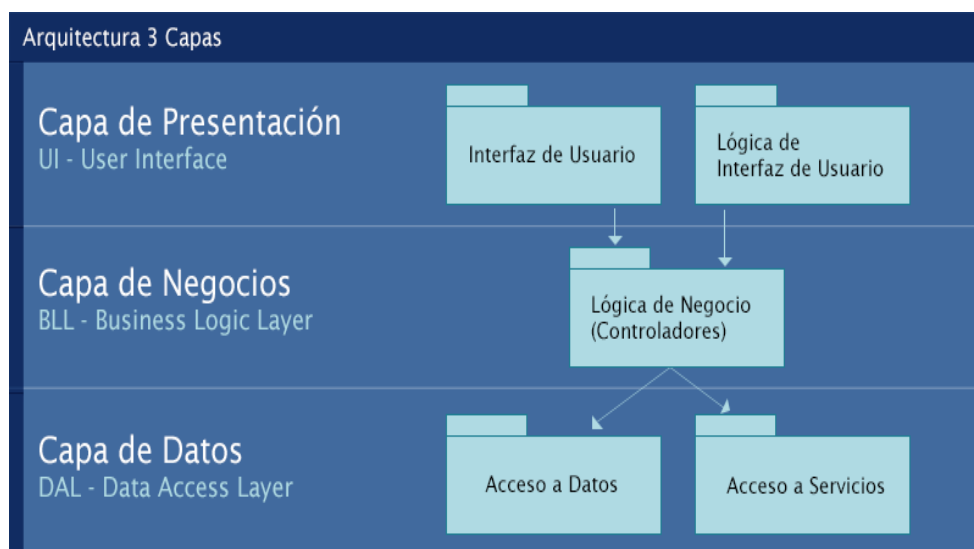


Figura 36: En la gráfica se describe la forma en la que se maneja la programación por capas.

La estructura de programación en capas nos permite entre muchas opciones manejar niveles de seguridad ya que al estar la capa de Presentación un nivel más arriba de la capa de acceso a datos se obtiene una mayor seguridad, al igual que el mantenimiento de la aplicación es mucho más sencillo al encontrarse ordenado el código se obtiene una mayor rapidez para modificar



o cambiar el código que tenga mal funcionamiento, la aplicación se vuelve escalable es decir que se puede ir implementando más módulos a la aplicación según se vaya requiriendo.

Se llega a la conclusión de que la manera más óptima de desarrollar la aplicación web es con la arquitectura tres capas, tomando en cuenta la estructura de elementos web como lo son:

- Manejar el contenido del Sitio como texto, imágenes, videos, sonidos, animaciones, archivos en sus carpetas específicas.
- Utilización de Hojas de estilo para todas las pantallas de la aplicación las cuales serán llamadas desde cada una de las páginas web según corresponda. Permitiendo la estandarización para definir como se presentará cada elemento del contenido, el tipo de letra a definir, el formato y la posición de cada elemento del documento, y la presentación de las páginas.
- Estructura de las paginas dinámicas php, con lenguaje de programación php separando la presentación de la interfaz gráfica de la capa de lógica de negocio por medio de clases.

Con respecto a la interfaz gráfica (páginas web) se maneja con la librería de código abierto bootstrap la cual contiene un sinnúmero de hojas de estilos CSS lo que permite generar una amigable interfaz de usuario también se maneja un iframe para la página principal que vendría hacer el menú de la aplicación la cual da un mejor dinamismo a nuestro sistema.

Utilización Iframe

Un iframe es un marco flotante que puede estar colocado en cualquier parte de la página. La ventaja que tiene sobre los frames o marcos tradicionales es que se puede colocar donde sea necesario, sin partir la página en varias partes y además puede ser dinámico.

`<iframe src="URL"></iframe>` Donde el url apunta a la localización de la página separada width y height es para definir su anchura y altura:

Ejemplo:

```
<iframe src="lista_clientes.php" width="200" height="200"></iframe>
```

Frameborder: es un atributo que define con un valor numérico el grado de grosor del iframe

Ejemplo:

```
<iframe src="demo_iframe.htm" frameborder="0"></iframe>
```

Ejemplo de un iframe:

```
<iframe src="demo_iframe.htm" name="iframe_a"></iframe>
```

```
<p><a href="lista_productos.php" target="iframe_a">listado de  
productos</a></p>
```

Manejo de Menús

Gracias a la utilización iframe se puede crear una única página para los menús y dentro de esta mandar a llamar a cada uno de las pantallas para los diferentes procesos lo cual optimiza en tiempo de carga de las páginas



dinámicas php, el manejo de la interfaz de menús se los hace por medio de hojas de estilo de la librería bootstrap la cual genera la presentación gráfica y la programación se lo hace por medio de php utilizando las etiquetas de hipervínculo.

La ubicación de los menús se recomiendan que sean: Horizontales.

La Capa Acceso Datos

Esta capa reúne todos los aspectos del software que tienen que ver con el manejo de los datos persistentes en la Base de Datos. La cual interactúa con el motor de base de datos MYSQL 5.6.12, por los beneficios que brinda esta herramienta, los que se describe a continuación:

- MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.
- Es lenguaje de programación muy utilizado en aplicaciones web, sobre plataformas Linux/Windows-Apache-MySQL-PHP/Perl/Python. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL.
- El servidor de bases de datos MySQL es la base de datos de fuente abierta más popular en el mundo. Su arquitectura lo hace extremadamente rápido y fácil de adaptar. Este servidor de bases de datos potente es usado por muchos programas de origen abierto incluyendo phpBB, osCommerce, y phpNuke.
- Plataforma estándar de Base de Datos en sistemas basados en Unix (Linux / BSD). PhpMyAdmin - Herramienta web para controlar y

manejar su base de datos MySQL. Corre bajo lenguaje de programación PHP.

- MySQL es el servidor de bases de datos relacionales más popular, desarrollado y proporcionado por MySQL AB.

MySQL es un sistema de administración de bases de datos (Data base Management System, DBMS) para bases de datos relacionales. Así, MySQL no es más que una aplicación que permite gestionar archivos llamados de bases de datos.

Existen muchos tipos de bases de datos, desde un simple archivo hasta sistemas relacionales orientados a objetos. MySQL, como base de datos relacional, utiliza múltiples tablas para almacenar y organizar la información.

Características de mysql:

Las principales características de este gestor de bases de datos son las siguientes:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.



- Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.

La Capa de Lógica

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

- Esta aplicación tiene código para implementar reglas de negocios.
- Se puede seleccionar almacenar la lógica de negocios sobre cada estación de cliente, u optar por ejecutar la lógica de negocios sobre un servidor de aplicaciones, en el caso de esta aplicación se ha optado por ejecutar la lógica de negocio sobre el servidor Apache.
- No toda la lógica de negocio es la misma algunas no requieren un frecuente acceso a los datos, pero en este sistema informático necesitara de la lógica de negocios para la validación en la entrada de campos, cálculos en tiempo real u otras interacciones de usuarios.

La Capa de la Presentación

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

"amigable" para el usuario generalmente se presentan como formularios.

5.03.02 Estándares de programación

Los estándares de programación son una parte muy importante en el desarrollo de software, para facilitar la identificación de cada uno de los componentes utilizados en el sistema, además de las siguientes razones por las cuales se deberá estandarizar el código fuente:

- El 70% al 80% del coste del código de un programa va a su mantenimiento.
- Las convenciones de código mejoran la lectura del software, permitiendo comprender el código nuevo más rápidamente y más a fondo.
- Si se distribuye el código fuente como un producto, necesitas asegurarte de que esté bien hecho.
- Cada persona piensa diferente, fijar estándares mejora el entendimiento mutuo.

Nombres básicos para las clases

Para una mejor compresión y fácil acceso a las diferentes clases creadas se ha implementara un estándar para los nombres de las clases partiendo de que es más fácil leer y diferenciar una clase llamada ListaCliente.php que una clase llamada LISTACLIENTE.php se llega a la conclusión que se deberá usar la primera letra con mayúscula de cada palabra del nombre de la clase:

Ejemplo:

ListaCliente.php



Y si se utilizara más de dos palabras se llevara el mismo estándar

Ejemplo:

ListaNuevosProductos.php

Nombres básicos para los controles

Para identificar mejor los controles que se utilizaran dentro de la aplicación ya sea para ingresar datos como son los (input tipo text) para mostrarlos como son las tablas dinámicas se implementara un estándar para todos estos controles como por ejemplo si se desea mostrar un mensaje de error en un (label) aplicando el estándar que se implementara quedaría así: Lbl_Error, también otro ejemplo sería si deseo mostrar los datos del cliente en un control tipo (table) quedaría de esta forma ya estandarizado Tbl_Cliente.

Nombres básicos para los procedimientos

Este es un punto importante en las tareas de definición de nombres. Sin embargo poco a poco se ha convertido en una buena norma el indicar una descripción precisa (y sin embargo corta) del procedimiento que se refiera. A estos nombres puede antecederseles un prefijo ps_ para distinguir a los procedimientos.

Convenciones de nombres para objetos

Para un mejor y más óptimo desarrollo se implementara un estándar general para los objetos que se usaran durante la creación de la aplicación con esto se pretende minimizar los errores por duplicidad de objetos.

La siguiente tabla contiene los nombres de los objetos a utilizar en la aplicación.

Tabla 11

Prefijos de los estándares que se van a utilizar en el desarrollo del sistema.

Tipo de control	Prefijo	Ejemplo
Input tipo text	Txt	Txt_Grupo
Input tipo password	Pasw	Pasw_Usuario
Input tipo radio	Rdo	Rdo_Respuesta
Input tipo checkbox	Cbx	Cbx_Producto
Input tipo image	Img	Img_Empresa
Input tipo file	Fle	Fle_Logo
Form	Frm	Frm_Venta
Input tipo hidden	Hdn	Hdn_Costo
Input tipo button	Btn	Btn_Guardar
Input tipo submit	Sbt	Sbt_Insertar
Input tipo reset	Rst	Rst_Usuario
textarea	Txa	Txa_Reporte
label	Lbl	Lbl_Error
table	Tbl	Tbl_Productos
select	Slct	Slct_Cedula
datalist	Dtl	Dtl_Ruc
Botón link	Btnlk	Btnlk_Menu
fieldset	Fdst	Fdst_Controles
legend	Lgd	Lgd_Ventas
optgroup	Optg	Optg_Preguntas
option	Opt	Opt_Ruc
keygen	Kygn	Kygn_Contraseña

Output	Oput	Oput_Iva
Progress	Prgs	Prgs_Ingreso
Command	Cmand	Cmand_Listado
Menú	Mnu	Mnu_Facturar
Iframe	Ifrm	Ifrm_Contenido

Nota: Estándares que se utilizan en el lenguaje de programación para realizar el sistema.

JavaScript

JavaScript (abreviado comúnmente "JS") es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a

funciones del servidor. JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Estándares HTML5

Html5 es una estandarización de las etiquetas html ya que es un estándar que todas las páginas web o sistemas orientados a la web deberían tener. En html5 se han incluido nuevos tipos de entradas de datos para los formularios, algunos de estos nuevos tipo de entrada tienen la capacidad de comprobar si el valor introducido coincide con su tipo de dato lo que nos ayuda bastante a la hora de solucionar la validación de los datos que el usuario introduce.

Con el uso de estas nuevas características en los formularios html5 nos evitamos tener que crear script para validar los datos del lado del cliente.

Lista de elementos Html5

Tabla 12

Prefijos de elementos htm5l.

Tipo de control	Prefijo	Ejemplo
Input tipo search	Sch	Sch_Retencion
Input tipo email	Eml	Eml_Cliente
Input tipo date	Dte	Dte_Fecha
Input tipo number	Nbr	Nbr_Costo

Nota: Elementos html5 utilizados en el sistema.

Convenciones de nombres de constantes y variables

Las variables se deben definir siempre con el menor alcance posible. Las variables globales (públicas) pueden crear máquinas de estado enormemente complejas y hacer la lógica de una aplicación muy difícil de entender. Las variables globales también hacen mucho más difícil mantener y volver a usar el código.

Prefijos de alcance de variables

A medida que aumenta el tamaño del proyecto, también aumenta la utilidad de reconocer rápidamente el alcance de las variables. Esto se consigue al escribir un prefijo de alcance de una letra delante del tipo de prefijo propio, sin aumentar demasiado la longitud del nombre de las variables.

Tabla 13

Prefijos de alcance de variables

Alcance	Prefijo	Ejemplo
Global	Gl	GlStrTipoCliente
Nivel de módulo	Md	MdNumrTotalCalculo
Local del procedimiento	Lc	LcDsmlIva
Público	Pb	PbNumrIdUsuario

Una variable tiene alcance global si se declara como Public en un módulo estándar o en un módulo de formulario. Una variable tiene alcance de nivel de módulo si se declara como Private en un módulo estándar o en un módulo de formulario, respectivamente.

Variables

Declarar todas las variables ahorra tiempo de programación porque reduce el número de errores debidos a erratas (por ejemplo, aNombreUsuario frente a sNombreUsuario y frente a sNombreUsuario). Las variables deben llevar un prefijo para indicar su tipo de dato. Opcionalmente, y en especial para programas largos, el prefijo se puede ampliar para indicar el alcance de la variable.

Tabla 14

Tipos de datos de variables.

Tipo de datos	Prefijo	Ejemplo
integer	Int	\$IntCosto
float	Flt	\$FLtPrecio
double	Dbl	\$DblIva
string	Str	\$StrCliente
Boolean	Bln	\$BlnRespuesta

Nombres descriptivos de variables y procedimientos

Se debe tener en cuenta las siguientes recomendaciones:

- Evitar (en lo posible) poner más de un return en una misma función.
- Todas las sentencias de control (if,for,while,if-else,etc.) deben poseer sus correspondientes llaves de apertura y de cierre ({}).
- Separar con un espacio en blanco las condiciones en una sentencia de control.

También diferenciar entre nombres de variables y funciones.

Funciones o métodos:

Se deberá escribir la primera letra de cada palabra con mayúscula Ejemplo:

MostrarNombre() {}

Variables:

Se diferenciara dentro del nombre de la variable de las partes por una mayúscula.

miVariable

Otros Aspectos:

Reglas de construcción en PHP

➤ **Función eval ():**

No utilizar la función eval()

Se prohíbe el uso de la función eval() cuando sea posible y cuando no lo sea, asegurar que los datos usados en la construcción no hayan podido ser manipulados.

➤ **Función phpinfo()**

Desactivar la función phpinfo()

Se debe desactivar la función phpinfo(), por motivos de seguridad, y restringir el acceso a cualquier recurso que utiliza esta función, ya que expone el contenido de la matriz \$_SERVER, lo que facilitaría información para proyectar ataques sobre la aplicación.

➤ **Función die()**

No usar die() para el manejo de errores

Si se produce un error, éste no reportará información concluyente, provocando un sensación poco agradable al usuario. Se deben utilizar funciones como trigger_error() junto con set_error_handler() para manejar los errores de una aplicación.

➤ **Funciones de conteo en bucles**

No usar funciones de conteo dentro de bucles

No se deben utilizar funciones tipo count() que se ejecuten dentro de la iteración, porque penalizan la velocidad del proceso. Si se calcula la función fuera del bucle, mejora la velocidad en torno a un 600% respecto a la codificación inicial.

➤ **Métodos estáticos**

Declarar los métodos estáticos cuando sea posible

Si es posible, hay que declarar los métodos como estáticos, siempre que vayan a ser tratados de esta manera. Está probado que se reduce su tiempo de ejecución hasta cuatro veces con respecto a los métodos que no están declarados como estáticos.

➤ **Función time()**

Utilizar la variable \$_SERVER ['REQUEST_TIME'] sobre la función time().



Cuando necesite calcular el tiempo actual dentro de la ejecución del algún script en una aplicación es mucho más eficiente utilizar `$_SERVER [ 'REQUEST_TIME' ]` que la función `time()`.

➤ **Función `foreach()`**

Utilizar `foreach()` en los bucles de colecciones y arrays

Cuando tenemos una estructura con un bucle destinado a la lectura de un array o una colección está demostrado que es mucho más rápido utilizar `foreach()` antes que estructuras del tipo `while` o `for`.

➤ **Función `strpos()`**

Utilizar `strpos()` para las búsquedas de subcadenas

Para realizar la búsqueda de subcadenas dentro de cadenas de texto, la mejor manera de realizarlo es utilizar la función `strpos()`, preferentemente sobre `preg_match()` o `ereg()`.

➤ **Pre incremento de variables**

Realizar pre incremento de las variables cuando sea posible

Realizar un pre incremento de una variable `++$i` es más rápido que el post incremento `$i++` de esa variable. Se deben cambiar las instrucciones para que se realice el pre incremento cuando se pueda y sobre todo en bucles críticos.



➤ **Depuración de código**

Realizar depuraciones de código

Se debe revisar cuidadosamente la salida que las páginas PHP producen con el fin de asegurarse de que no se están produciendo errores. Como alternativa, puede asegurarse que el error de registro esté activado en su archivo php.ini, y comprobarlo con regularidad. Existen funciones que aumentan la capacidad de depuración, permitiendo comprobar los scripts sin necesidad de mostrar datos en HTML.

➤ **Conexiones persistentes**

Usar conexiones persistentes

Se debe conectar a la base de datos mediante conexiones persistentes. Éstas permanecen aunque el script haya terminado, lo que agiliza la ejecución de otro script si conecta a base de datos porque la conexión sigue abierta. Esto ahorra tiempo negociando contraseñas y la ejecución de parte del código.

➤ **Inicialización de variables**

Inicializar las variables para reducir el tiempo de ejecución

Aunque no es necesario, se recomienda inicializar las variables en PHP para reducir el tiempo de ejecución. Las variables no inicializadas tienen un valor predeterminado de acuerdo a su tipo (las booleanas se asumen como FALSE, los enteros y flotantes como cero, las cadenas se establecen como una cadena vacía y las matrices se

convierten en un array vacío). En algunos casos, como los métodos GET, no será posible inicializar las variables.

Apoyo apropiado del teclado

Para cualquier aplicación que requiera entrada de los datos cabeza-abajo, ésta es probablemente una regla buena para tener en cuenta. Todos las órdenes del menú necesitan tener un teclado accesible (con llaves mnemónicas o atajos del teclado), y todos las órdenes de sistema deben estar disponibles en el menú.

Generación de texto al usuario

Mensajes del texto, sobre todo informativo y diálogos de advertencia, necesitan ser llevados de forma apropiada y consistente. Aquí están algunas reglas: Evite jerga técnica. Limite los mensajes a 2-3 líneas. Evite redacción que culpe al usuario. Evite el uso de abreviaturas. Los mensajes se alinearan a la Izquierda cuando sean de líneas múltiples. Nunca, incluya faltas de ortografía, errores de gramática o pronunciación incorrecta. Uso del conjunto de caracteres apropiado

Tabla15

Estándares de generación de texto para el usuario

Texto	Font (letra)
Title bars, menu text	Helvetica, Arial, sans-serif (14 point)
Controles, labels y captions	14 point
Campos de entrada de datos	14 point
Textos de los Status bar	14 point
Textos de los iconos	14 point



La interface con controles incoherentemente etiquetados (textos, combos, listas, option buttons, grupos, y otros) puede hacer tanto como cualquier otro factor para que el sistema sea rechazado por parte de los usuarios.

Ventanas y Diálogos

Se mostraran los mensajes de error y alerta utilizando la librería de estilos bootstrap la cual tiene incorporado todo tipo de mensajes para mostrar al usuario, también se utilizara los mensajes de alerta javascript para hacer más dinámica la aplicación.

Menús

Su forma y colores deben de ser siempre consistentes en todos los sistemas, los ítem a seleccionar deben tener teclas de acceso rápido, en caso de utilizar iconos o imágenes en los menús estos deben ser consistentes, en todo el sistema y serán agregados tras previa discusión y puesta en estándares por el grupo de desarrolladores de la institución. Para cualquiera de los tipos de menús, los textos deberán comenzar con un verbo, preferiblemente en infinitivo y que sea suficientemente descriptivo de la acción que realizara El formato básico para los menús pop-up (emergente o contextual), será como sigue: El 1er Ítem será el título del menú, este debe ser escrito en mayúscula
Un separador Ítems seleccionables (opciones)

5.03.03 Estándares de Base de Datos

Estándares de Caso de Uso

El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos.

La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como *modelo*. Recordemos que un *modelo* es una representación simplificada de la realidad; el modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema.

Por eso es necesario estandarizar la manera correcta para modelar los diferentes componentes de un diagrama de casos de uso en todo lo que respecta a su formato, organización y otros aspectos relevantes. Cabe mencionar que este diagrama de casos de uso está creado utilizando la herramienta de modelado Rational Rose, una herramienta de diseño basada en la especificación del lenguaje de modelado, permitiendo crear variados diagramas que apoyan a la etapa de diseño de software.

Formato de los Casos de Uso

Nombre.- La escritura de la etiqueta se realizara en tipo oración es decir la primera letra con mayúsculas.

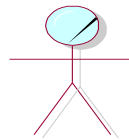


Entregar retención

Figura 37: Representación de Caso de Uso.

Formato de los Actores

En el nombre del actor se deberá escribir la primera letra con mayúscula.



Cajero

Figura 38: Representación de un actor.

Relaciones.

Las relaciones entre un actor y un caso de uso, se dibujan con una línea simple.

Para relaciones entre casos de uso, se utilizan flechas etiquetadas "incluir" o "extender."

Include.- Una relación "incluir" indica que un caso de uso es necesitado por otro para poder cumplir una tarea, además deberá ir incluida por la etiqueta <<include>> y utilizando una línea simple.

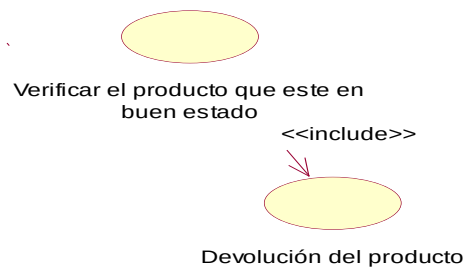


Figura 39: Representación de un caso de uso con la relación incluye.

Extend.- Una relación "extender" indica opciones alternativas para un cierto caso de uso, además deberá ir incluida por la etiqueta `<<extend>>` y utilizando una flecha simple.

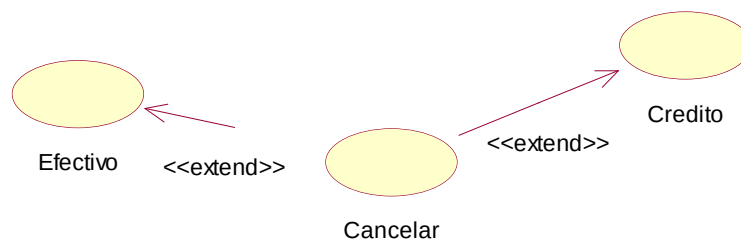


Figura 40: Representación de un caso de uso con la relación incluye.

Orden en los diagramas

Se debe alinear horizontalmente los casos de uso con respecto a su actor principal, como se muestra en el ejemplo siguiente.

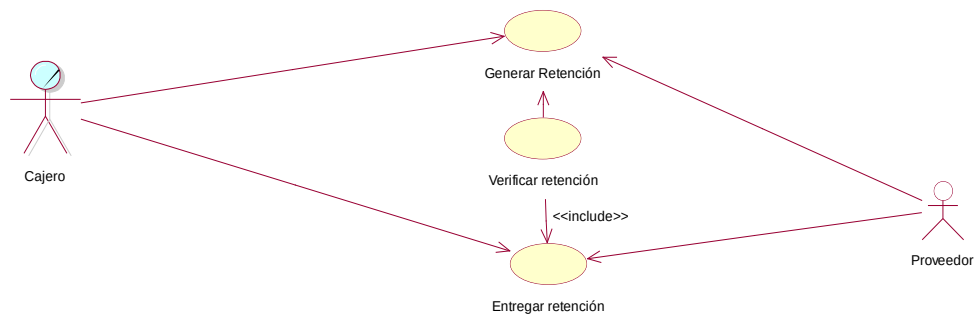


Figura 41: Representación del orden correcto de un diagrama de un caso de uso.

Se debe evitar que las líneas de relación se crucen, para que no se produzca confusión al momento de revisar los casos de uso.

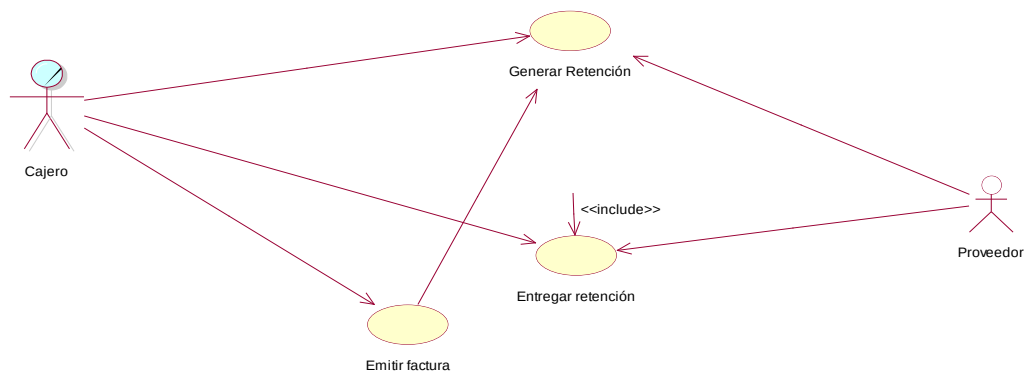


Figura 42: Representación del orden incorrecto de un diagrama de un caso de uso.

Estándares del Diagrama De Secuencia

Los diagramas de clases y los de objetos representan información estática. No obstante, en un sistema funcional, los objetos interactúan entre sí, y tales



interacciones suceden con el tiempo. El diagrama de secuencias UML muestra la mecánica de la interacción con base en tiempos.

Rol de la Clase.-El rol de la clase describe la manera en que un objeto se va a comportar en el contexto. No se listan los atributos del objeto.

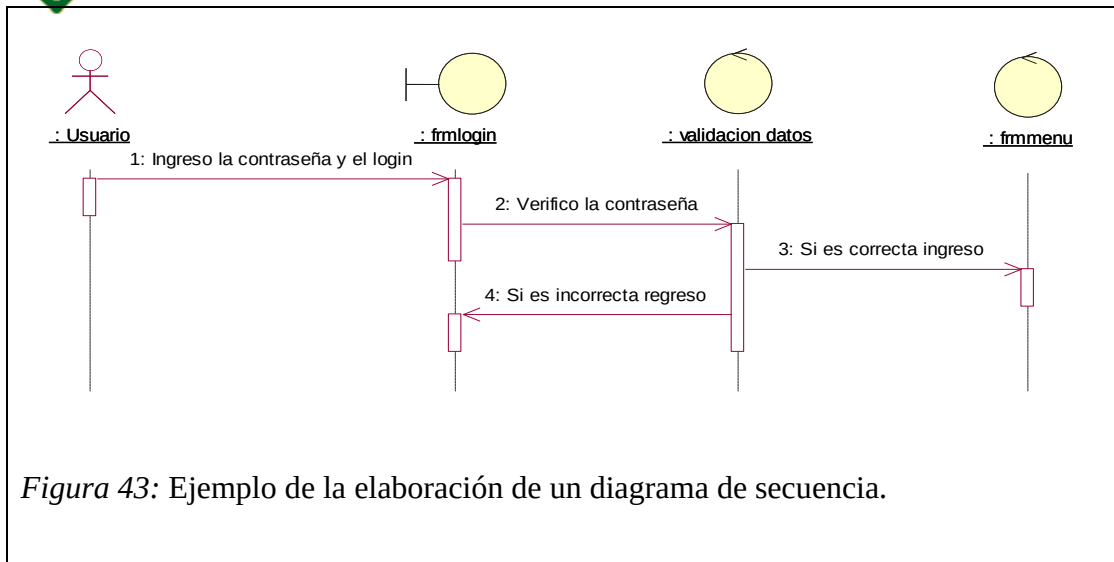
Activación.-Los cuadros de activación representan el tiempo que un objeto necesita para completar una tarea.

Mensajes.-Los mensajes son flechas que representan comunicaciones entre objetos. Las medias flechas representan mensajes asincrónicos. Los mensajes asincrónicos son enviados desde un objeto que no va a esperar una respuesta del receptor para continuar con sus tareas.

Líneas de Vida.- Las líneas de vida son verticales y en línea de puntos, ellas indican la presencia del objeto durante el tiempo.

Destrucción de Objetos.- Los objetos pueden ser eliminados tempranamente usando una flecha etiquetada "<<destruir>>" que apunta a una X.

Loops.- Una repetición o loop en un diagrama de secuencias, es representado como un rectángulo. La condición para abandonar el loop se coloca en la parte inferior entre corchetes [].



Formato de los diagramas de secuencia

Un diagrama de secuencia es una forma de diagrama de interacción que muestra los objetos como líneas de vida a lo largo de la página y con sus interacciones en el tiempo representadas como mensajes dibujados como flechas desde la línea de vida origen hasta la línea de vida destino. Los diagramas de secuencia son buenos para mostrar qué objetos se comunican con qué otros objetos y qué mensajes disparan esas comunicaciones. Los diagramas de secuencia no están pensados para mostrar lógicas de procedimientos complejos.

Línea de vida.- Una línea de vida representa un participante individual en un diagrama de secuencia. Una línea de vida usualmente tiene un rectángulo que contiene el nombre del objeto. Si el nombre es self entonces eso indica que la línea de vida representa el clasificador que posee el diagrama de secuencia. Lo anteriormente mencionado se puede ver en la siguiente imagen

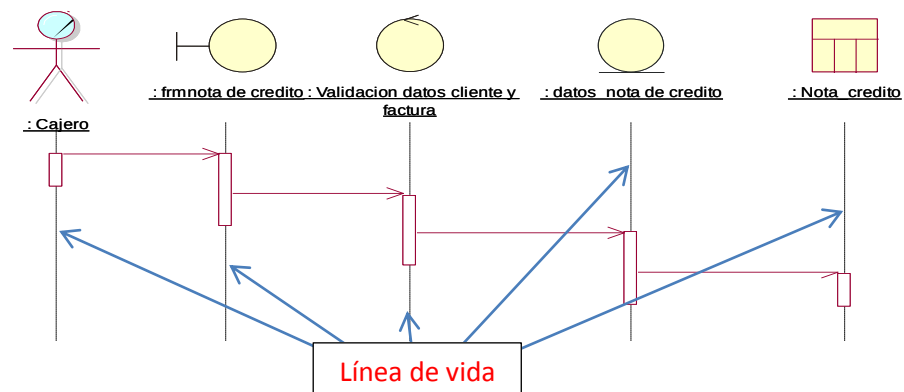


Figura 44: Línea de vida del proceso en un Diagrama de Secuencia con objetos.

Un diagrama de secuencia también puede tener una línea de vida con un símbolo del elemento actor en la parte superior. Este usualmente sería el caso si un diagrama de secuencia es contenido por un caso de uso. Los elementos entidad, control y límite de los diagramas de robustez también pueden contener líneas de vida, tal y como se puede ver en la imagen siguiente.

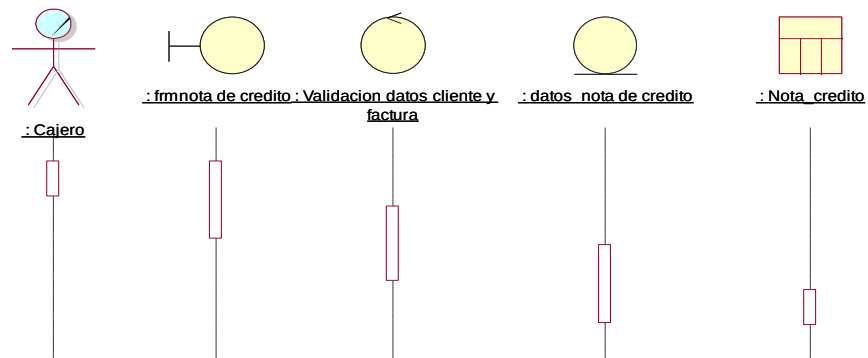
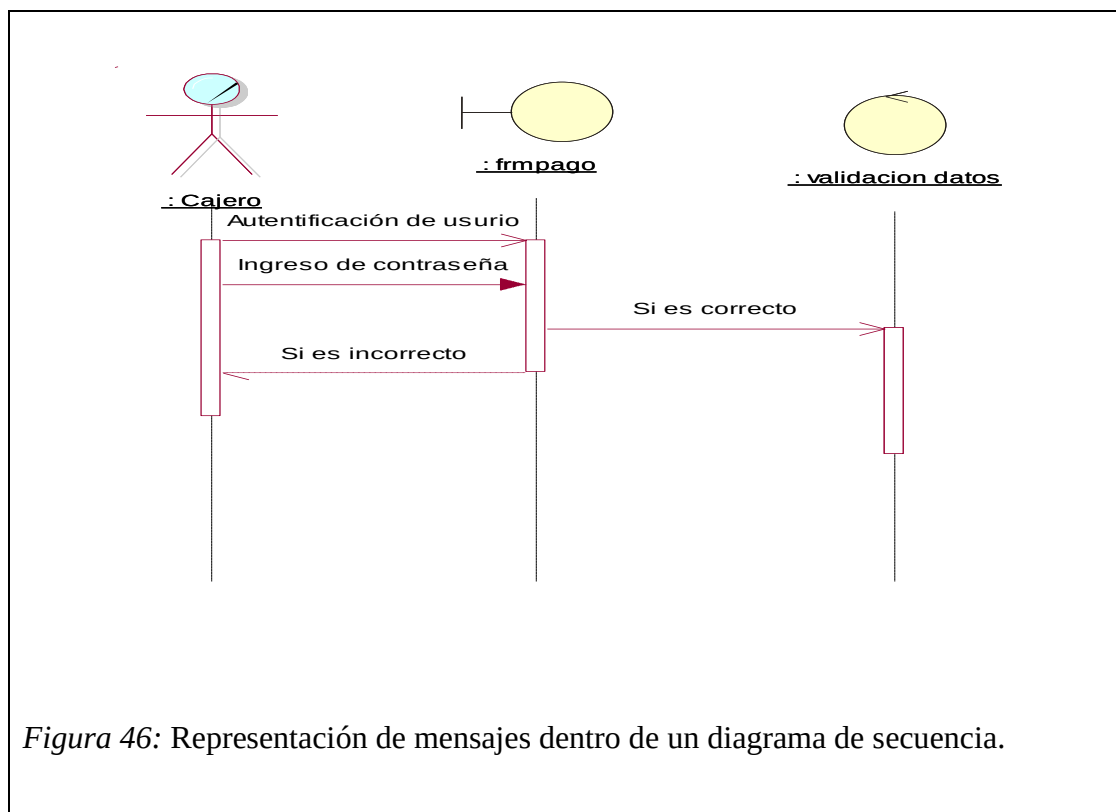


Figura 45: Línea de vida del proceso en un Diagrama de Secuencia con el uso de símbolos.

Al igual que en las otras líneas de vida, los nombres de cada una de ellas deben ser escritos utilizando mayúscula para la primera letra.

Si el nombre de la línea de vida está compuesto por más de una palabra, estas deben ser escritas con formato de frase, es decir, la primera letra de la primera palabra en mayúscula y las otras palabras escritas en minúscula.

Mensajes.- Los mensajes se muestran como flechas. Los mensajes pueden ser completos, perdidos o encontrados; síncronos o asíncronos: llamadas o señales. En el siguiente diagrama, el primer mensaje es un mensaje síncrono (denotado por una punta de flecha oscura), completo con un mensaje de retorno implícito; el segundo mensaje es asíncrono (denotado por una punta de flecha en línea) y el tercero es un mensaje de retorno asíncrono (denotado por una línea punteada).





Estándares Del Diagrama De Clases

Los *diagramas de clases* describen la estructura estática de un sistema.

Las cosas que existen y que nos rodean se agrupan naturalmente en categorías. Una clase es una categoría o grupo de cosas que tienen atributos (propiedades) y acciones similares. Un ejemplo puede ser la clase “Aviones” que tiene atributos como el “modelo de avión”, “la cantidad de motores”, “la velocidad de crucero” y “la capacidad de carga útil”. Entre las acciones de las cosas de esta clase se encuentran: “acelerar”, “elevarse”, “girar”, “descender”, “desacelerar”.

Un rectángulo es el símbolo que representa a la clase, y se divide en tres áreas. Un diagrama de clases está formado por varios rectángulos de este tipo conectados por líneas que representan las asociaciones o maneras en que las clases se relacionan entre sí.

Formato de la Clase

Clase Abstracta.- Las *clases* se representan con rectángulos divididos en tres áreas: la superior contiene el nombre de la clase, la central contiene los *atributos* y la inferior las *acciones*.

Nombre.- La primera letra de cada palabra deber ser escrita con letra mayúscula. Si el nombre consta más de una palabra, estas deben ir unidas como se muestra en la imagen siguiente.

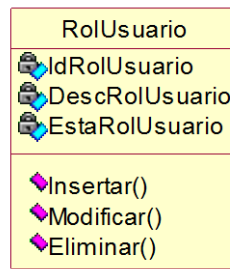


Figura 47: Representación de una clase.

Atributos.- Deben escribirse la primera palabra con mayúscula. Al poseer más de una palabra, las palabras deben comenzar con cada primera letra en mayúscula.

Detallar el tipo de dato de los atributos de las clases, detallar la visibilidad o el modo de acceso, es decir, si es pública (+), privada (-) o protegida (#). Al detallar un atributo "static", será representado por texto subrayado.

Operaciones.- Se cumple con la misma configuración detallada en el apartado "Atributos", dentro de las operaciones debe encontrarse el constructor de la clase.

Se debe especificar el tipo de dato, tanto de los parámetros como el retorno en la siguiente imagen se muestra un ejemplo.

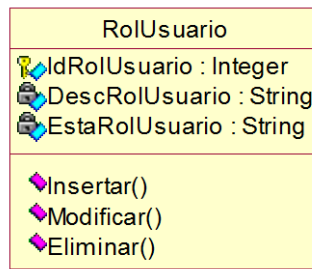


Figura 48: Representación de una clase.

Relaciones.

Herencia.- En cuanto a la organización de las líneas de generalización, estas deben mostrar características de ortogonalidad, como se muestra en el ejemplo.

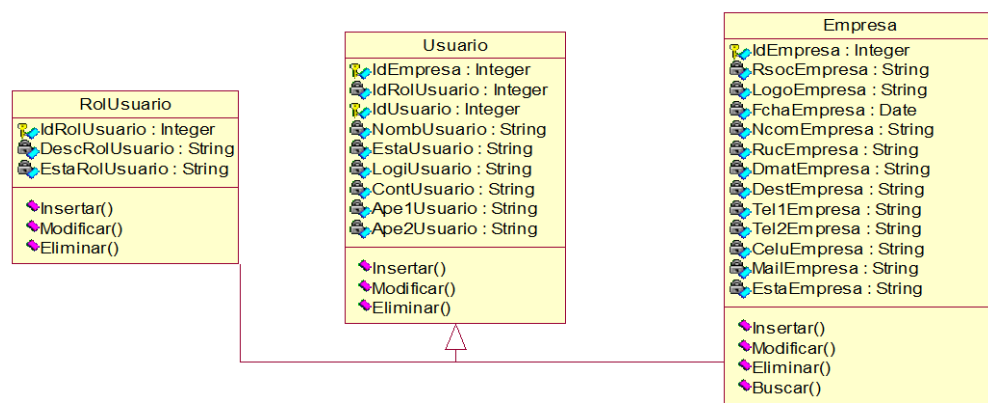


Figura 49: Representación de la relación de Herencia.

Asociaciones en general.- En cuanto a la realización de asociaciones simples, agregación, composición, dependencia, etc. Se debe evitar en gran parte la representación oblicua o diagonal de estas, haciéndolas más ortogonales.

También se debe lograr cierta alineación en el espacio de representación (mejor organización posible).

Respecto a los nombres de las asociaciones, estos deben ser presentados en letra minúscula y separados (si cuentan con más de una palabra).

Se debe detallar la multiplicidad de cada relación. Ejemplo: $1 \dots *$, $1 \dots n$, entre otros

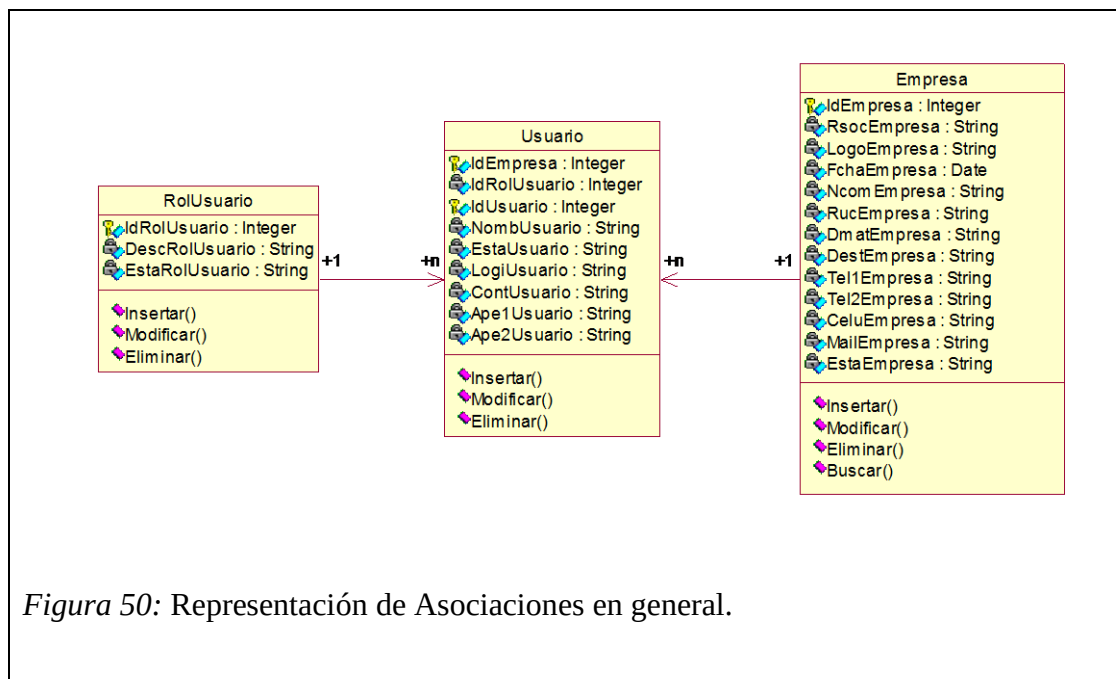


Figura 50: Representación de Asociaciones en general.

5.03.04 Diseño de interfaces

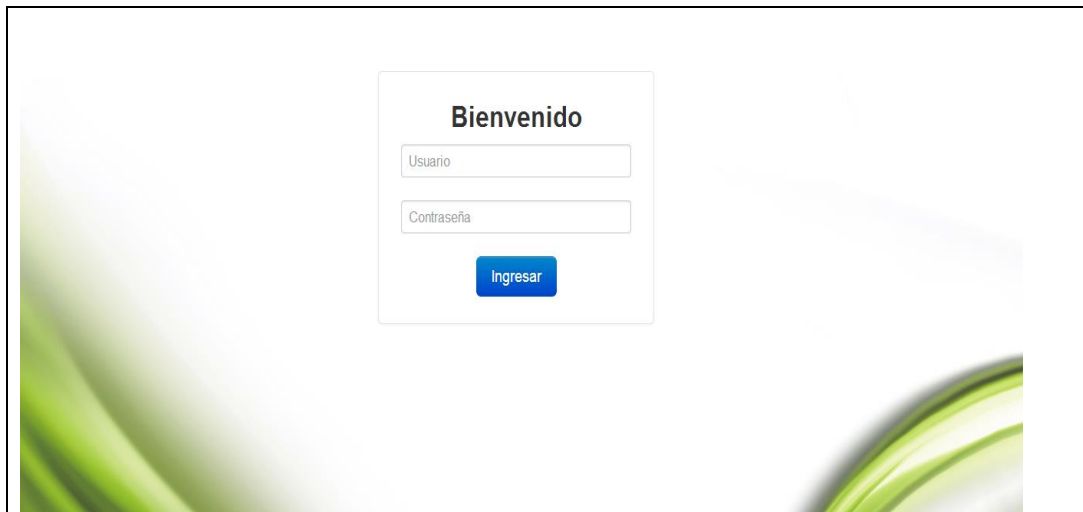


Figura 51: En esta página se muestra el ingreso a la aplicación el usuario se debe autenticar y dependiendo del tipo de usuario se mostrara un menú caso contrario regresara a la pantalla que se muestra en el gráfico.

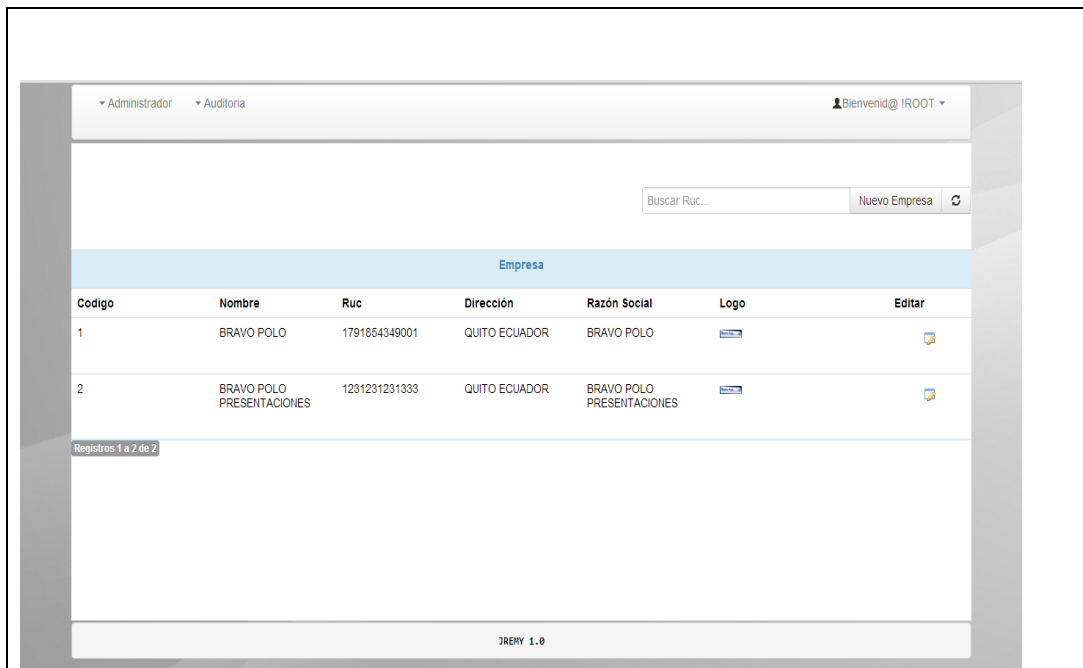
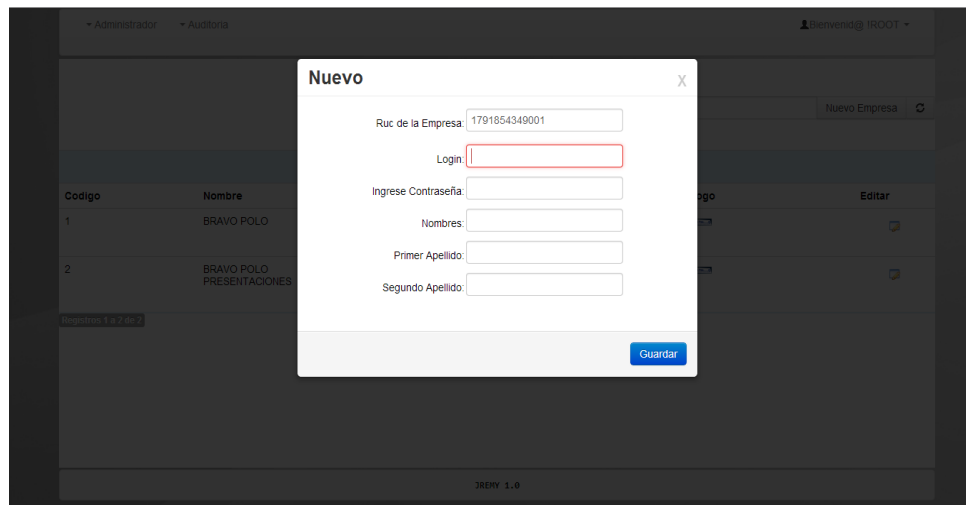


Figura 52: En esta página el usuario root puede ingresar una nueva empresa, actualizar además crear usuarios administradores para las diferentes empresas creadas.



Código	Nombre
1	BRAVO POLO
2	BRAVO POLO PRESENTACIONES

Figura 53: En esta página se puede crear administradores para las empresas creadas.

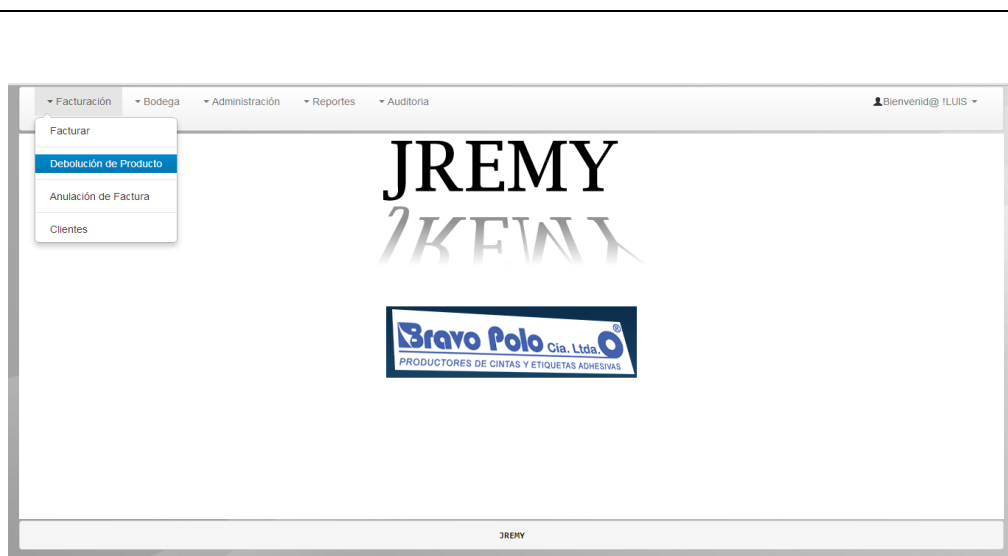


Figura 54: En esta página se muestra el menú principal del sistema.

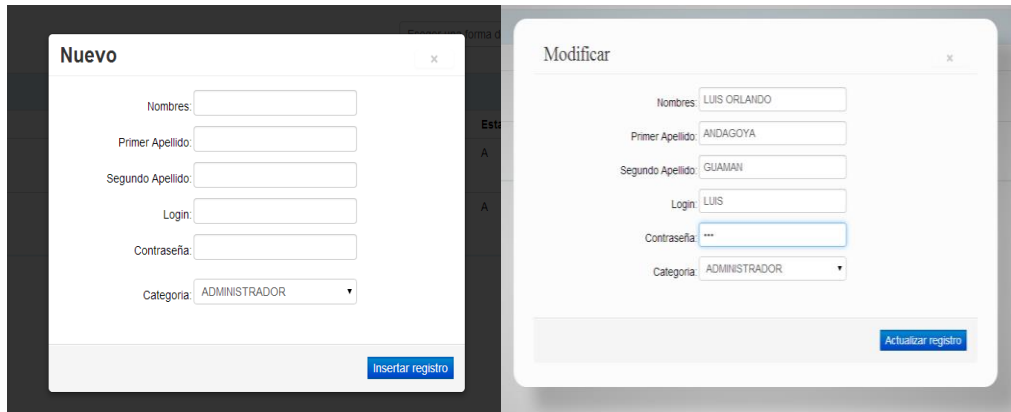
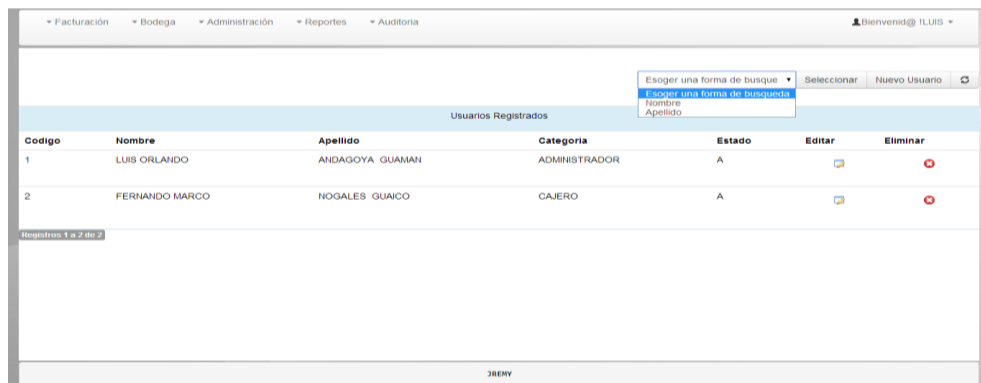
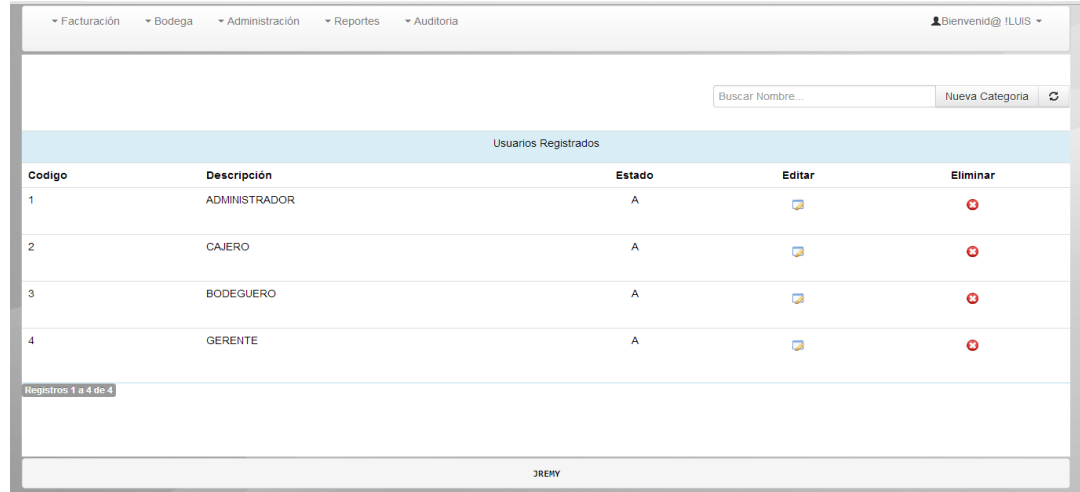


Figura 55: En estas ventanas modales se puede modificar y crear un nuevo usuario su categoría.



Código	Nombre	Apellido	Categoría	Estado	Editar	Eliminar
1	LUIS ORLANDO	ANDAGOYA GUAMAN	ADMINISTRADOR	A		
2	FERNANDO MARCO	NOGALES GUAICO	CAJERO	A		

Figura 56: Se visualizan todos los usuarios y sus categorías que pertenecen a la empresa y también se tiene la opción de eliminar, además de buscar usuarios específicos por nombre o apellido.



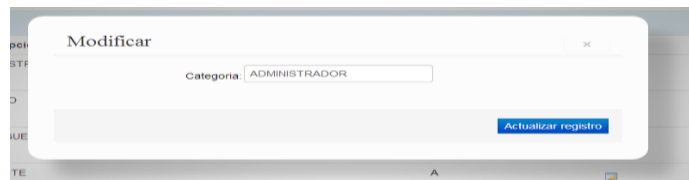
The screenshot shows a web application interface with a navigation menu at the top: Facturación, Bodega, Administración, Reportes, and Auditoría. The user is logged in as 'Bienvenid@ ILUIS'. Below the menu is a search bar labeled 'Buscar Nombre...' and a button 'Nueva Categoría'. The main content area displays a table titled 'Usuarios Registrados' with the following data:

Código	Descripción	Estado	Editar	Eliminar
1	ADMINISTRADOR	A		
2	CAJERO	A		
3	BODEGUERO	A		
4	GERENTE	A		

At the bottom of the table, it says 'Registros 1 a 4 de 4' and 'JREMY'.

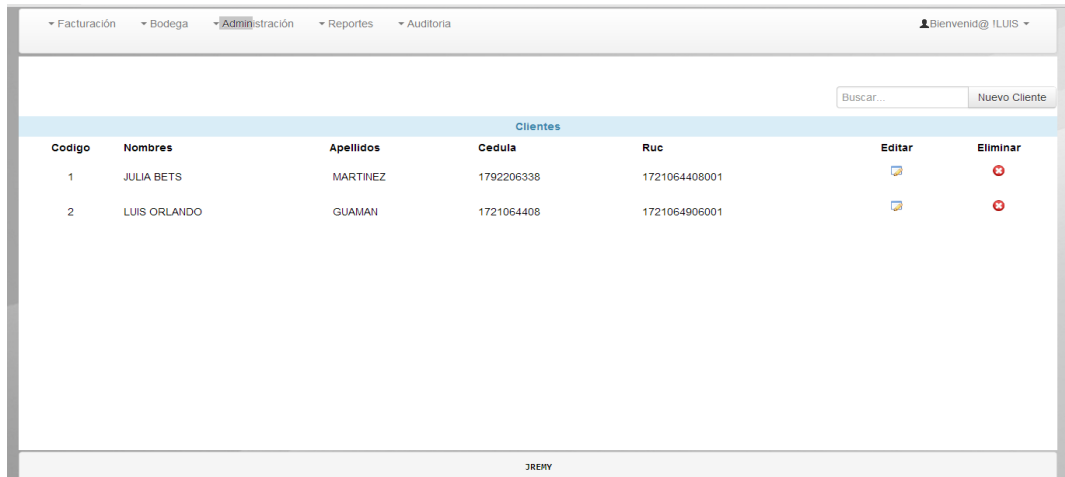
Figura 57: Se visualiza los las categorías para los permisos de acceso al sistema de cada empresa.

Ventana Modal para actualizar la categoría del cliente.



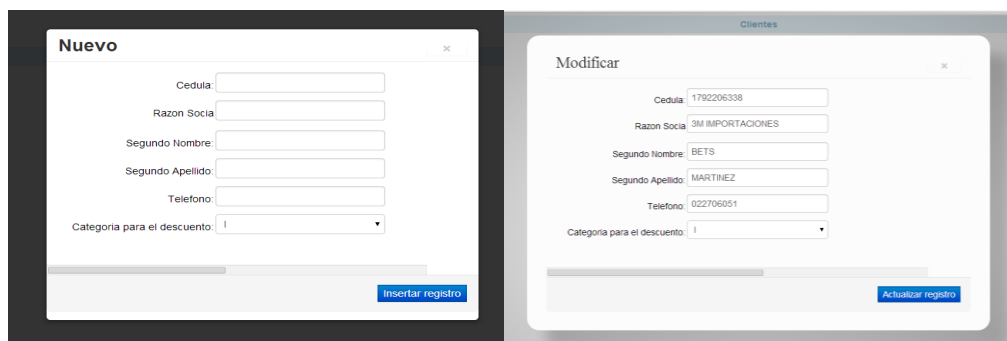
The screenshot shows a modal window titled 'Modificar' with a close button (X). Inside the modal, there is a label 'Categoría:' followed by a text input field containing the value 'ADMINISTRADOR'. At the bottom right of the modal, there is a blue button labeled 'Actualizar registro'.

Figura 58: En esta ventana el administrador podrá modificar la categoría para el ingreso al sistema de los usuarios.



Código	Nombres	Apellidos	Cedula	Ruc	Editar	Eliminar
1	JULIA BETS	MARTINEZ	1792206338	1721064408001		
2	LUIS ORLANDO	GUAMAN	1721064408	1721064906001		

Figura 59: En esta página se visualiza los clientes que tiene la empresa, solo el administrador podrá eliminar un cliente.



Nuevo

Cedula:

Razon Social:

Segundo Nombre:

Segundo Apellido:

Telefono:

Categoria para el descuento:

Insertar registro

Modificar

Cedula:

Razon Social:

Segundo Nombre:

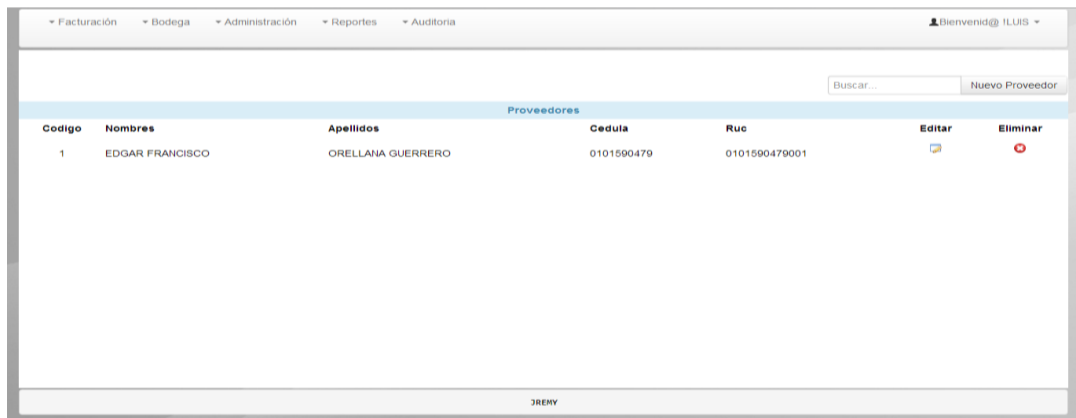
Segundo Apellido:

Telefono:

Categoria para el descuento:

Actualizar registro

Figura 60: En estas dos ventanas se podrá ingresar un nuevo cliente o modificar sus datos según se vea conveniente.



Código	Nombres	Apellidos	Cedula	Ruc	Editar	Eliminar
1	EDGAR FRANCISCO	ORELLANA GUERRERO	0101590479	0101590479001		

Figura 61: En esta página se podrá visualizar los datos personales del proveedor, solo el administrador podrá eliminar un proveedor.

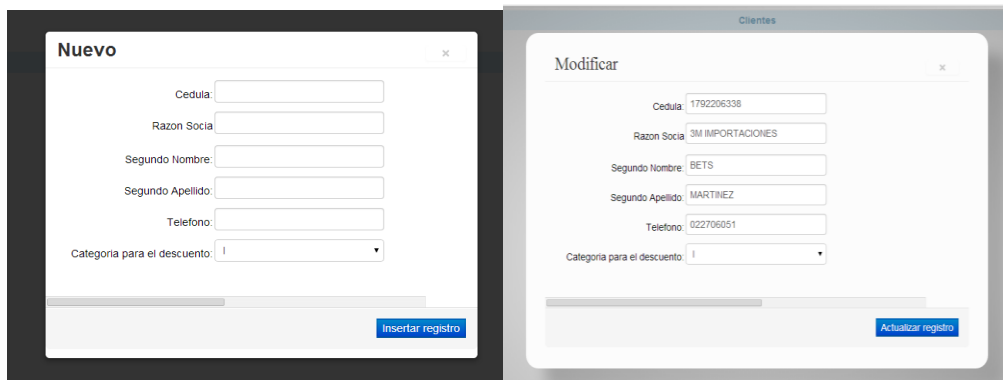
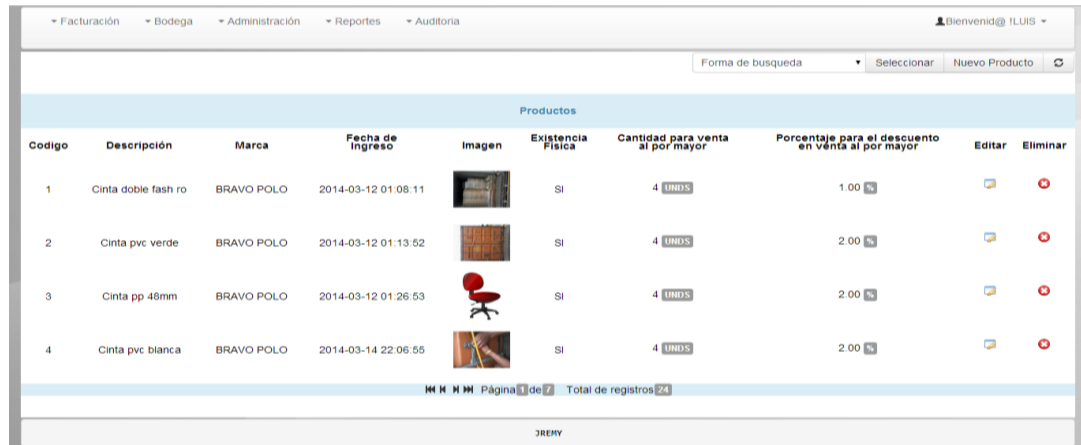


Figura 62: En esta página el jefe de bodega puede registrar o modificar el tipo de productos y dirigirse a la lista del tipo de productos.



Código	Descripción	Marca	Fecha de Ingreso	Imagen	Existencia Física	Cantidad para venta al por mayor	Porcentaje para el descuento en venta al por mayor	Editar	Eliminar
1	Cinta doble fash ro	BRAVO POLO	2014-03-12 01:08:11		SI	4 UNDS	1.00 %		
2	Cinta pvc verde	BRAVO POLO	2014-03-12 01:13:52		SI	4 UNDS	2.00 %		
3	Cinta pp 48mm	BRAVO POLO	2014-03-12 01:26:53		SI	4 UNDS	2.00 %		
4	Cinta pvc blanca	BRAVO POLO	2014-03-14 22:06:55		SI	4 UNDS	2.00 %		

Forma de búsqueda: [v] Seleccionar Nuevo Producto [x]

Página 1 de 2 Total de registros 24

Figura 63: Se puede visualizar el listado de los productos existentes en bodega, solo el administrador puede eliminar un producto.



Nuevo

Nombre:

Marca del Producto:

Cantidad de Venta al por Mayor:

Porcentaje de venta al por mayor:

Referencia:

Imagen: Ningún archivo seleccionado

Existe en Bodega:

Modificar

Nombre:

Marca del Producto:

Cantidad de Venta al por Mayor:

Porcentaje de venta al por mayor:

Referencia:

Imagen: P020111_1251.jpg

Existe en Bodega:

Figura 64: En esta página el administrador puede realizar el ingreso, modificación, eliminación y visualizar la categoría a la que pertenece cada producto.



Proveedor | ☒ Nuevo Proveedor

Razon Social: SUPER PROVEEDOR (SUMINISTROS DE OFICINA Y COMPUTAC) Cedula/Ruc: 0101590479 Direccion: AV. HUAYNA-C??PAC S/N Y ELOY ALFARO Telefono: 022786051

☒ Nuevo Producto

Codigo de barra o Nombre del producto CANTIDAD: 1 COSTO: Agregar

Numero	Descripcion del Producto	Valor Unitario	Cant.	Importe	
1	Cinta doble fash ro	\$ 1,00	5	\$ 5,00	Remover
2	Cinta pp 48mm	\$ 2,00	4	\$ 8,00	Remover

Total de Articulos: 9

[Guardar](#)

Subtotal: \$ 13,00
Iva 12%: \$ 1,56
Neto: \$ 14,56

Figura 65: Se visualiza en esta página los datos del proveedor y se puede ingresar los productos además de sus cantidades y costos.

Selecciona Proveedor

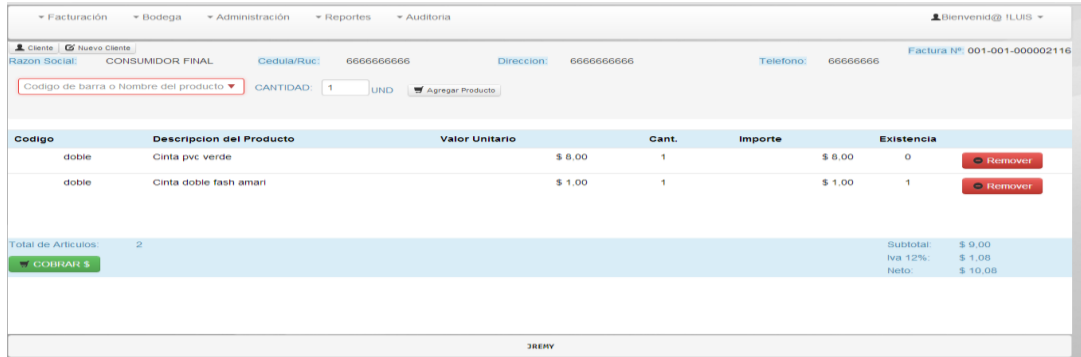
Escoga una opción de busqueda: ☒ Cedula ☐ Ruc

[Seleccionar](#) [Cerrar](#)

Figura 66: En esta ventana se puede escoger un proveedor ya sea por el número de la cedula o el ruc del proveedor.



Figura 67: En esta ventana el bodeguero puede guardar los productos ingresados y el número de factura al que pertenece así como el descuento se puede aplicarlo automáticamente solo con ingresar el porcentaje.



Codigo	Descripcion del Producto	Valor Unitario	Cant.	Importe	Existencia
doble	Cinta pvc verde	\$ 6.00	1	\$ 6.00	0
doble	Cinta doble fash amari	\$ 1.00	1	\$ 1.00	1

Total de Articulos: 2

Subtotal: \$ 9.00
Iva 12%: \$ 1.08
Neto: \$ 10.08

Figura 68: Esta página permite vender los productos mediante facturas.

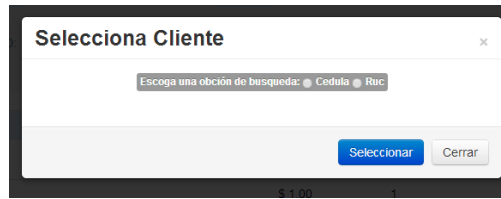


Figura 69: En esta ventana se puede escoger un cliente ya sea por el número de ruc o por el número de cedula.



Figura 70: Esta ventana permite guardar la factura también ingresar el descuento y la forma de pago.

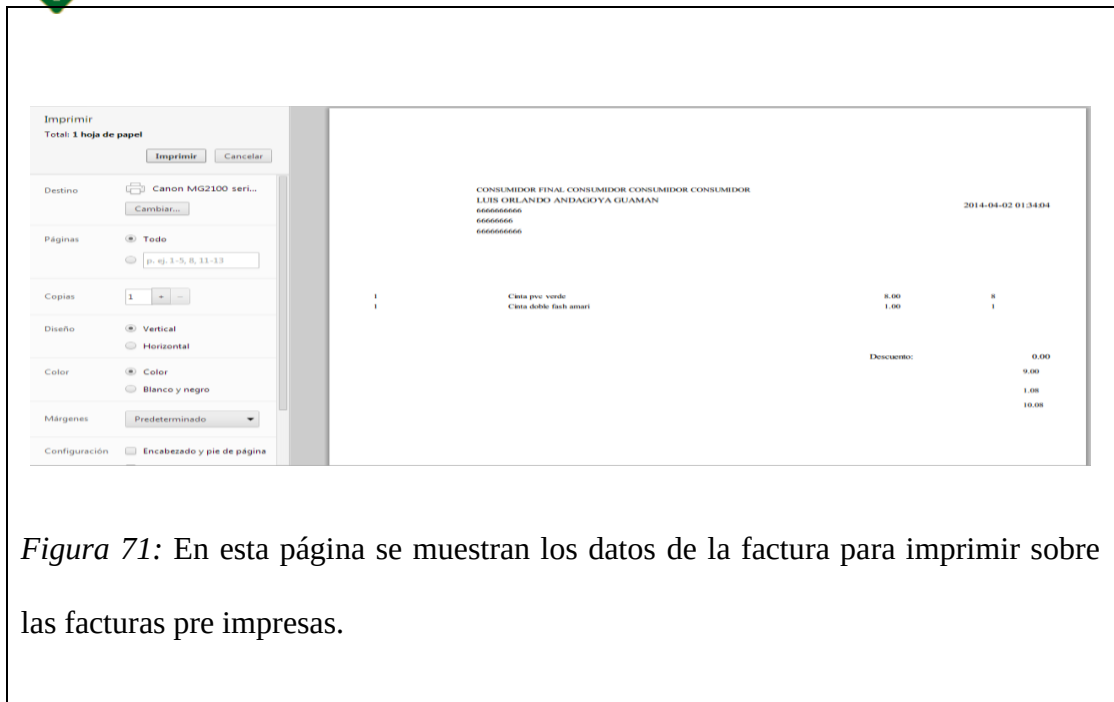


Figura 71: En esta página se muestran los datos de la factura para imprimir sobre las facturas pre impresas.

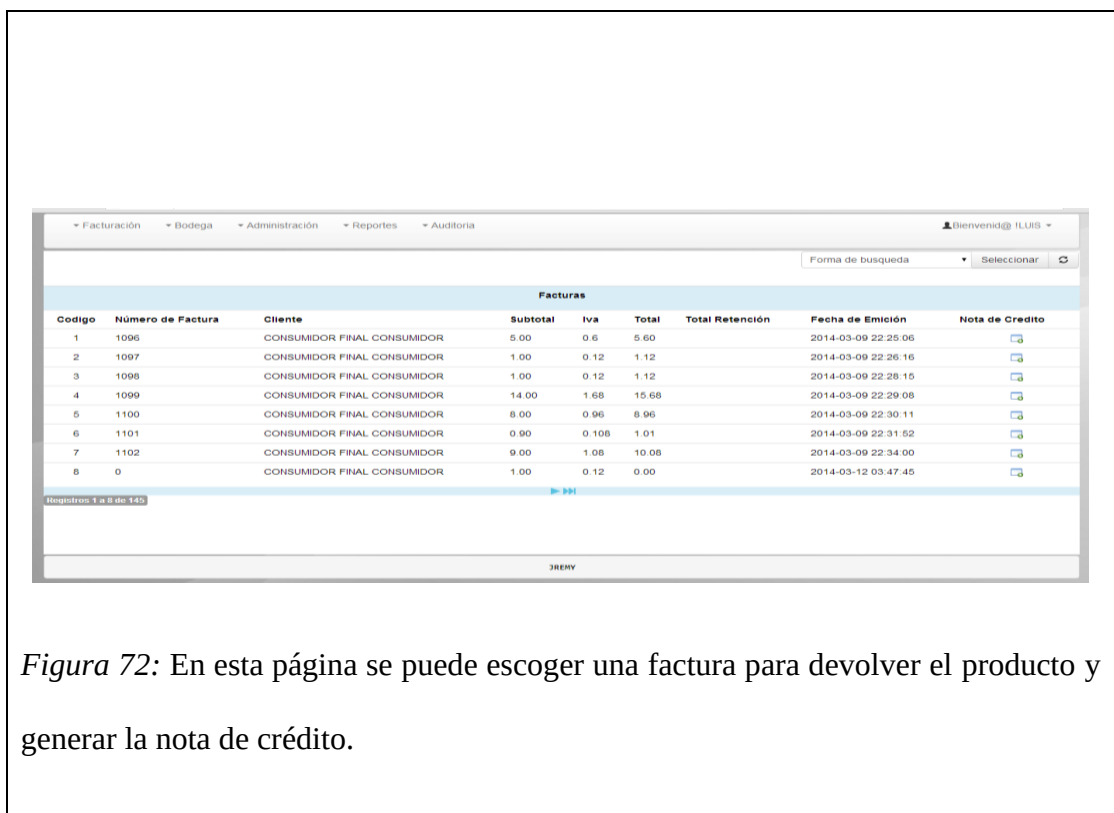


Figura 72: En esta página se puede escoger una factura para devolver el producto y generar la nota de crédito.

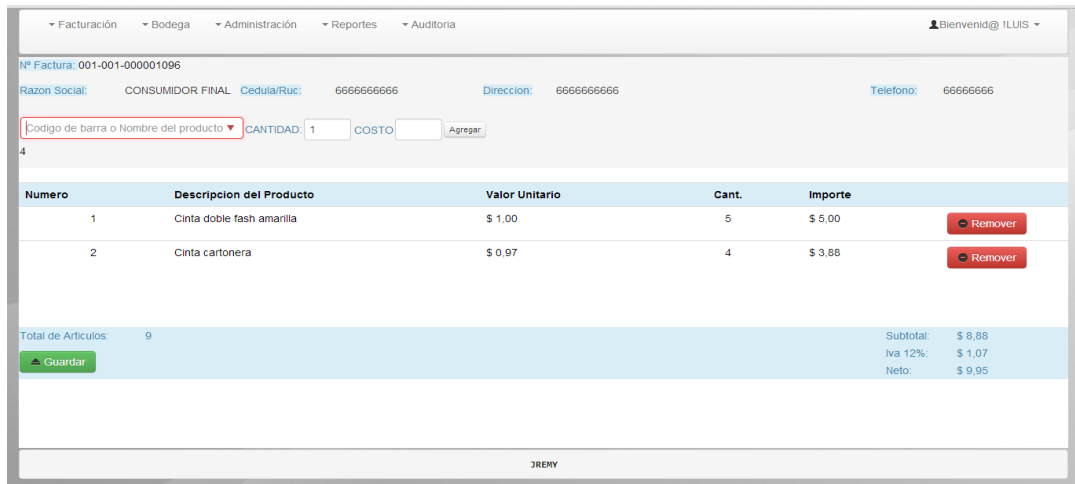


Figura 73: Se podrá realizar la generación de la nota de crédito por el costo del producto devuelto.

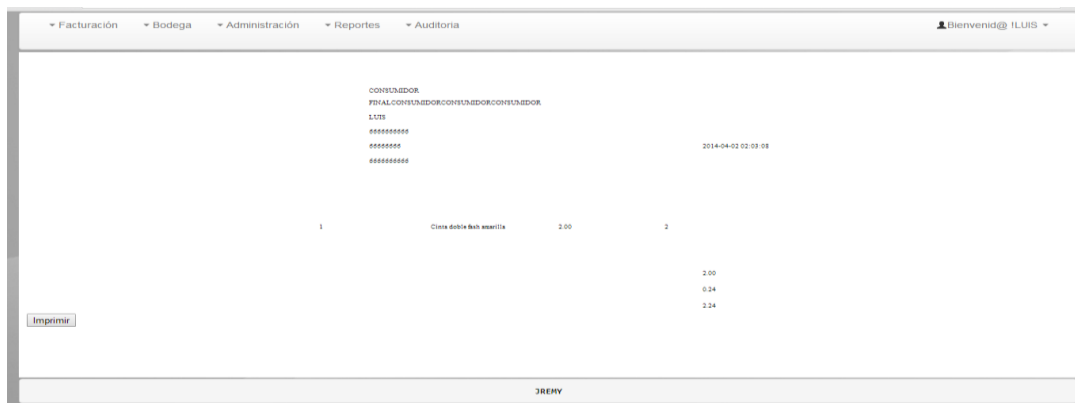


Figura 74: Aquí se podrá imprimir las notas de crédito sobre las notas de crédito pre impresas.

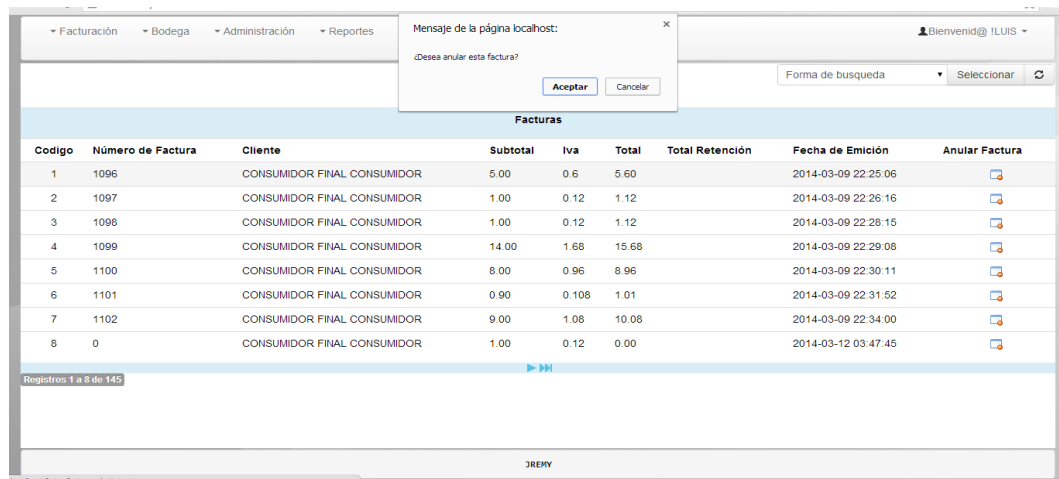


Figura 75: En esta página se podrá anular una factura cabe decir que solo el administrador podrá anular una factura con autorización del cliente.

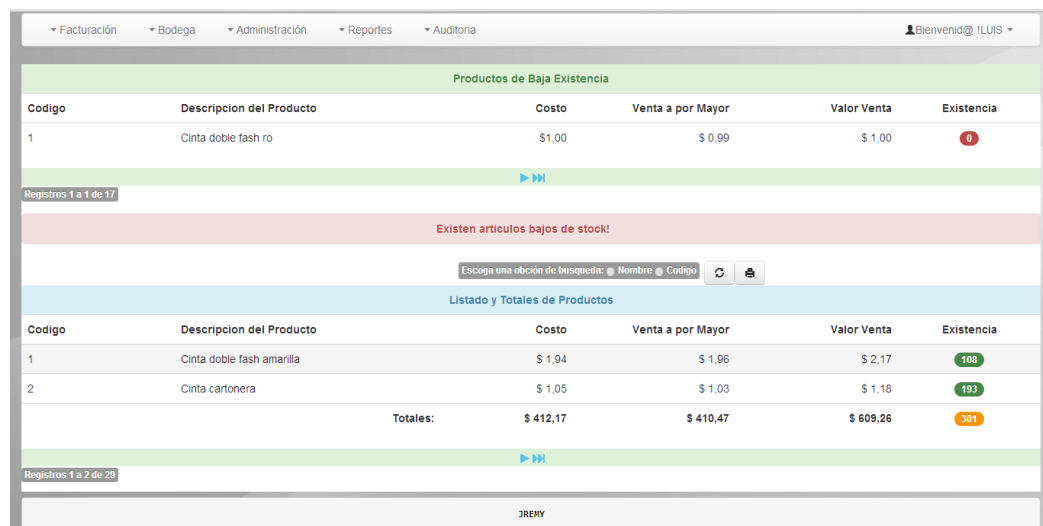
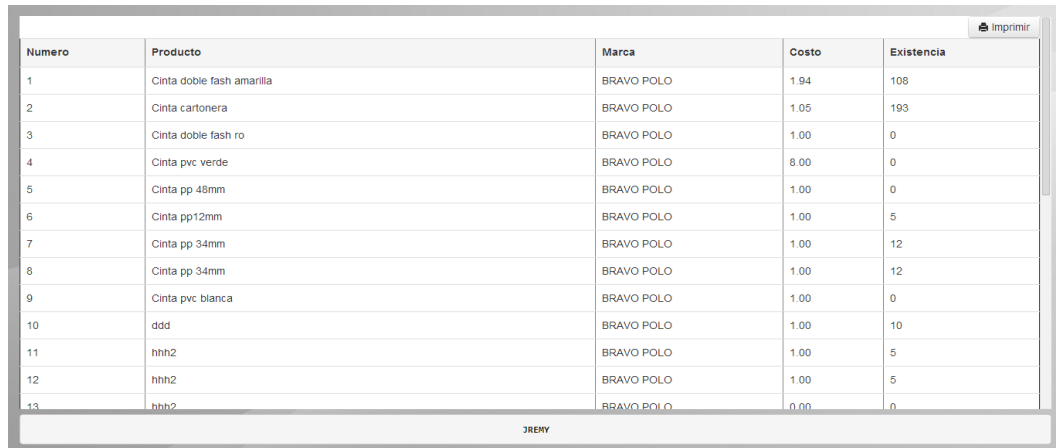
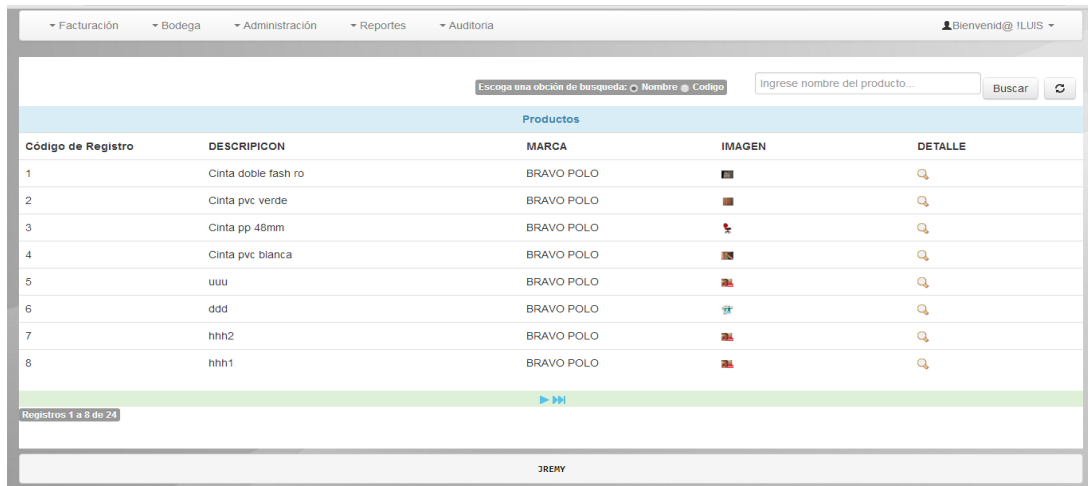


Figura 76: Se podrá visualizar las existencias de productos en bodega.



Numero	Producto	Marca	Costo	Existencia
1	Cinta doble fash amarilla	BRAVO POLO	1.94	108
2	Cinta cartonera	BRAVO POLO	1.05	193
3	Cinta doble fash ro	BRAVO POLO	1.00	0
4	Cinta pvc verde	BRAVO POLO	8.00	0
5	Cinta pp 48mm	BRAVO POLO	1.00	0
6	Cinta pp12mm	BRAVO POLO	1.00	5
7	Cinta pp 34mm	BRAVO POLO	1.00	12
8	Cinta pp 34mm	BRAVO POLO	1.00	12
9	Cinta pvc blanca	BRAVO POLO	1.00	0
10	ddd	BRAVO POLO	1.00	10
11	hhh2	BRAVO POLO	1.00	5
12	hhh2	BRAVO POLO	1.00	5
13	hhh2	BRAVO POLO	0.00	0

Figura 77: Se podrá imprimir todas las existencias de productos y sus costos.



Código de Registro	DESCRIPCION	MARCA	IMAGEN	DETALLE
1	Cinta doble fash ro	BRAVO POLO		
2	Cinta pvc verde	BRAVO POLO		
3	Cinta pp 48mm	BRAVO POLO		
4	Cinta pvc blanca	BRAVO POLO		
5	uuu	BRAVO POLO		
6	ddd	BRAVO POLO		
7	hhh2	BRAVO POLO		
8	hhh1	BRAVO POLO		

Figura 78: Se podrá escoger el producto para ver el detalle del kardex .

Facturación

Bodega

Administración

Reportes

Auditoria

Bienvenid@ ILUIS

Desde: dd/mm/aaaa

Hasta: dd/mm/aaaa

Buscar

MOVIMIENTOS DE KARDEX											
PRODUCTO: Cinta pvc verde			ENTRADAS			SALIDAS			SALDOS		
CÓDIGO	FECHA	CONCEPTO	CANTIDAD	VALOR UNITARIO	VALOR TOTAL	CANTIDAD	VALOR UNITARIO	VALOR TOTAL	CANTIDAD	VALOR UNITARIO	VALOR TOTAL
1	2014-03-28 19:25:46	COMPRAS	1	8.00	8.00	0	0.00	0.00	1	8.00	8.00
2	2014-04-02 01:34:04	VENTAS	0	0.00	0.00	1	8.00	8.00	0	8.00	0.00

JRENY

. *Figura 79:* En esta página se podrá imprimir el detalle de kardex de un producto específico así como buscar sus entradas y salidas por la fecha.

5.04 Pruebas

5.04.01 Documento De Casos De Prueba

Tabla 16

Historial de revisiones

Fecha	Descripción	Autor
24/03/2014	Inicio de pruebas	Luis Orlando Andagoya Guaman
24/03/2014	Pruebas de integridad de datos	Luis Orlando Andagoya Guaman
27/03/2014	Pruebas del sistema	Luis Orlando Andagoya Guaman
04/04/2014	Pruebas de la interfaz del usuario	Luis Orlando Andagoya Guaman
04/04/2014	Pruebas de desempeño	Luis Orlando Andagoya Guaman

Nota: Explica la fecha de inicio de cada prueba.

Plan de pruebas maestro

Propósito

A continuación se describe en este documento el plan de pruebas para el sistema de facturación y control de inventario de bodega para la empresa Bravo Polo Cía. Ltda., ubicada en la ciudad de Quito, el cual pretende probar las funcionalidades y características del mencionado sistema.



El cual está basado sobre los siguientes objetivos:

- Identificar que la información existente del proyecto y que los componentes de software sean probados.
- Hacer una lista de requerimientos recomendados de prueba (de alto nivel).
- Recomendar y describir las estrategias a ser empleadas.
- Identificar los recursos requeridos y estimar los esfuerzos de las pruebas.
- Listar los elementos a entregar de las actividades de pruebas.

Alcance

Se pretende realizar la implantación de la versión 1.0 del sistema de facturación y control de inventarios de bodega para la compañía Bravo Polo Cía. Ltda., ubicada en la ciudad de Quito así como realizar los test respectivos respecto a la funcionalidad del sistema.

Requerimientos de pruebas

Tenemos dos requerimientos funcionales y no funcionales los cuales son los objetivos de las pruebas.

Los detalles de cada prueba serán determinados posteriormente mientras los casos de prueba sean identificados y los scripts sean desarrollados.

Pruebas de integridad de datos y BD

- Verificar el acceso a la Base de Datos.
- Verificar el acceso simultáneo en la lectura de registro de las distintas tablas.



- Verificar el bloqueo realizado durante actualizaciones de registros de las tablas transaccionales.
- Verificar la correcta obtención de data actualizada.

Pruebas del sistema

- Verificar el ingreso al sistema - CU01.
- Verificar venta de producto – CU02.
- Verificar emisión de facturas – CU03.
- Verificar entrega de productos al cliente – CU04.
- Verificar devolución de producto – CU05
- Verificar compra de productos a proveedores – CU06.
- Verificar generación de retenciones – CU07.

Pruebas de desempeño

- Verificar el tiempo de respuesta para acceder remotamente a la aplicación.
- Verificar el tiempo de respuesta para registrar un usuario.
- Verificar el tiempo de respuesta para consultar solicitud de servicio.
- Verificar el tiempo de respuesta para generar orden de servicio.

Estrategia de pruebas

Se debe considerar que depende mucho la estrategia de prueba del sistema para que los resultados sean favorables y exactos así permitiendo una mejor depuración del sistema.

Tipos de pruebas



Las consideraciones principales para la estrategia de pruebas son las técnicas a usarse y los criterios para determinar si la prueba fue completada.

Además de las consideraciones provistas para cada prueba mencionada, las pruebas deberían ser únicamente ejecutadas usando bases de datos conocidas y controladas en entornos seguros.

Pruebas de integridad de datos y Base de Datos

La base de datos y los procesos de bases de datos deberían ser probadas en sistemas separados.

Revisión exhaustiva sobre el gestor de base de datos a usarse necesita ser realizada para identificar las herramientas y técnicas que puedan existir para soportar las pruebas a realizarse.

Objetivo

Verificar que el acceso a datos sea en tiempo óptimo, que los procesos internos como disparadores y procedimientos almacenados tengan un buen funcionamiento y que se mantenga la integridad de la información.

Técnicas

Ingresar datos correctos e incorrectos para probar que cada tabla solo guarde datos correctos definidos en el diseño de base de datos.

Realizar consultas a cada tabla paramétricamente de forma que se pueda denotar que solo se pueda ver los datos pedidos en la consulta.

Criterio de cumplimiento



La base de datos funciona correctamente, los datos que se guardan en cada tabla es correspondiente a aquellos asignados en el diseño y las consultas realizadas a la base de datos no contiene errores.

Pruebas del sistema

Las pruebas sobre la aplicación deberían enfocarse en requerimientos que puedan ser asociados directamente a casos de uso (o funciones de negocio), y reglas del negocio. Las metas de estas pruebas son verificar la aceptación, el procesamiento y obtención de data apropiada, así como la apropiada implementación de reglas del negocio. Este tipo de pruebas está basado en las técnicas de caja negra, utilizando para ello la GUI y analizando los resultados.

Objetivo

Comprobar el correcto funcionamiento de los procesos del sistema como son facturación, compras etc., y que cada proceso realice solo aquello para lo que fue creado.

También se pretende demostrar que la aplicación funciona correctamente para varios clientes conectados simultáneamente.

Técnicas

Ejecutar cada uno de los componentes de sistema ingresando datos validos e inválidos para denostar que:

- Las validaciones de datos sean correctas.
- Los mensajes de error se generen al momento de surgir inconvenientes en el sistema.



- Los reportes contengan solo los datos correctos.
- Los mensajes de alerta sean generados antes de realizar un proceso no valido.

Criterio de cumplimiento

- Todas las pruebas planificadas fueron ejecutadas
- Todos los mensajes de alerta, error, validación de datos y reportes se generaron correctamente y a tiempo.

Pruebas de la interfaz de usuario (IU)

Verifica la interacción del usuario con el software. La meta de las pruebas de IU es asegurar que la interfaz de usuario provea al usuario el acceso apropiado para acceder y navegar por las funciones de la aplicación. Además, las pruebas IU asegura que los objetivos dentro de la IU funcionen como se esperaba y conforme a los estándares de la compañía.

Objetivo

Verificar:

- a) Que la aplicación refleje los requerimientos del negocio.
- b) Que los objetos de ventanas y sus características, como menús medidas posición, estado y foco sea conforme a los estándares.

Técnicas

Ingresar a cada ventana y verificar que los estándares y requerimientos sean los pedidos por la compañía Bravo Polo Cía. Ltda.



Criterio de cumplimiento

Todas las ventanas, menús y reportes son los requeridos por el negocio.

Pruebas de desempeño

Realizar las pruebas que miden los tiempos de respuesta, las tasas de transacción y otros requerimientos sensibles al tiempo. La meta de las pruebas de desempeño es verificar y validar que los requerimientos de desempeño han sido alcanzados. Este tipo de pruebas es ejecutado muchas veces, y cada ejecución emplea una carga subrepticia (background load) en el sistema.

Objetivo

Validar el tiempo de respuesta para transacciones diseñadas o funciones de negocio bajo las siguientes condiciones:

- a) volumen normal anticipado.
- b) volumen de caso mal anticipado.

Técnicas

Usar scripts de prueba desarrollados por pruebas de modelo de negocio (pruebas de sistema).

Modificar archivos de datos (para incrementar el número de transacciones) o modificar los scripts para incrementar el número de iteraciones en que cada transacción ocurre.

Lo scripts deben correr en una sola máquina (en el mejor de los casos simular un usuario único, una única transacción) y ser repetido en múltiples clientes (virtuales o actuales).

Criterio de cumplimiento

Una transacción / un único usuario. El cumplimiento exitoso de estas pruebas, es cuando no se encuentran fallas en los tiempos esperados o requerido (en cada transacción).

Múltiples transacciones / múltiples usuarios. El cumplimiento exitoso de estas pruebas, es cuando no se encuentran fallas en los tiempos aceptables.

Casos de Prueba

Tabla 17

Venta de producto.

Caso de Uso	Venta de producto – CU02.
Caso de Prueba	Venta de producto – CU02.
Actor	Cajero.
Pre Condición	El cajero debe ingresar al sistema.
Propósito	Ingresar todos los datos de producto y cliente.

Nota: Requerimientos que el cajero debe seguir para generar la venta de productos.

Tabla 18

Escenario.

Sección	Actividad	Clase de Equivalencia	Resultado Esperado
1	El cajero ingresa los datos: Del cliente. La cantidad de producto. El tipo de producto. El costo del producto.	Valida	Visualiza los datos ingresados en una tabla la cual muestra ya calculado los totales, iva y subtotales.
2	Registrar la factura	Valida	Envía a la pantalla para imprimir factura

Tabla 19

Emisión de factura.

Caso de Uso	Emisión de factura – CU03.
Caso de Prueba	Emitir factura– CU03.
Actor	Cajero.
Pre Condición	El cajero debe ingresar los datos de la factura.
Propósito	Emitir una factura.

Nota: Requerimientos que el cajero debe seguir para la emisión de facturas.

Tabla 20

Escenario.

Sección	Actividad	Clase de Equivalencia	Resultado Esperado
1	El cajero debe presionar el botón imprimir.	Valida	Impresión de los datos de la factura sobre las facturas pre impresas.

Tabla 21

Entrega de productos al cliente.

Caso de Uso	Entrega de productos al cliente – CU04.
Caso de Prueba	Entregar los productos al cliente– CU04.
Actor	Bodeguero.
Pre Condición	El bodeguero debe ingresar al sistema y el cajero debe anteriormente generar la factura y entregarla al cliente.
Propósito	Entregar el producto al cliente.

Nota: Requerimientos que el bodeguero debe seguir para entregar el producto al cliente.

Tabla 22

Escenario.

Sección	Actividad	Clase de Equivalencia	Resultado Esperado
1	El bodeguero debe verificar la existencia de productos.	Valida	Visualizar las existencias de productos.

Tabla 23

Devolución de productos.

Caso de Uso	Devolución de productos – CU05.
Caso de Prueba	Generar de nota de crédito – CU05.
Actor	Cajero.
Pre Condición	El cajero debe ingresar al sistema.
Propósito	Emitir al cliente una nota de crédito.

Nota: Requerimientos que el cajero debe seguir para generar una nota de crédito.

Tabla 24

Escenario.

Sección	Actividad	Clase de Equivalencia	Resultado Esperado
1	El cajero ingresa los datos:	Valida	Visualiza la nota de crédito en pantalla.
	Cantidad de producto		
	Costo de producto		
	Descripción del producto		
2	Registra la nota de crédito	Valida	Imprime la nota de crédito.

Nota: El proceso que se sigue para generar una nota de crédito.

Tabla 25

Compra de productos.

Caso de Uso	Compra de productos – CU06.
Caso de Prueba	Comprar productos –CU06.
Actor	Bodeguero.
Pre Condición	El bodeguero debe ingresar al sistema.
Propósito	Ingresa los productos comprados y los datos de la factura.

Nota: Requerimientos que el cajero debe seguir para la compra de productos.

Tabla 26.

Escenario.

Sección	Actividad	Clase de Equivalencia	Resultado Esperado
1	El bodeguero ingresa los datos:	Valida	Generar el kardex, el costo del producto y actualizar las existencias.
	Cantidad de producto		
	Costo de producto		
	Descripción del producto		
	Descuento del total del producto		
	Numero de factura		
2	Registra los productos.	Valida	Imprime el ingreso de productos.

Nota: Describe el funcionamiento del sistema

Tabla 27.

Generación de retenciones.

Caso de Uso	Generación de retenciones – CU07.
Caso de Prueba	Generación de retenciones – CU07.
Actor	Cajero.
Pre Condición	El cajero debe ingresar al sistema y debe haberse generado una compra de producto.
Propósito	Emitir retención.

Nota: Requerimientos que el cajero debe seguir para generar retenciones.

Tabla 28.

Escenario.

Sección	Actividad	Clase de Equivalencia	Resultado o Esperado
1	El cajero ingresa los datos:	Valida	Visualiza
	Ejercicio fiscal		la
	Base imponible para la		retención
	retención		en
	Impuesto		pantalla.
	Código del impuesto		
2	Valor retenido	Valida	Imprime
	Total de la retención		la
	Registra retención	Valida	retención.

Nota: Describe el proceso para la retención.

Capítulo VI: Aspectos Administrativos

6.01 Recursos

Tabla 29

Recurso Humano.

RECURSOS HUMANOS			
Humano	Nombre	Actividad	Responsabilidad
Promotor del proyecto	Luis Orlando Andagoya Guaman	Desarrollo del proyecto.	Investigar, crear y desarrollar el sistema de facturación y control de inventarios de bodega para la empresa.
Tutor	Ingeniera Diana Teran	Guiar el desarrollo del proyecto.	Revisar el progreso del desarrollo del sistema y la documentación.
Lector	Ingeniero Jaime Basantes	Revisar el desarrollo del proyecto.	Asegurar que el desarrollo del proyecto elaborado cumple con las normas establecidas.

Nota: Muestra el recurso humano utilizado en el desarrollo del proyecto.

Tabla 30

Recurso Material.

RECURSOS MATERIALES			
Material	Cantidad	Valor Unitario	Valor Total
Computador	1	1000,00	1000,00
Hojas de papel bon	300	0,02	6,00
Esfero	2	0,35	0,70
Lápiz	1	0,40	0,40
Borrador	1	0,15	0,15
Grapas	15	0,01	0,15
Servicios Básicos	800 horas	0,20	160,00
Servicio de internet	500horas	0,080	40,00
Alimentación	100 días	2,00	200,00
Transporte	100 días	1,00	100,00
Total			1.507,40

Nota: Muestra el recurso material utilizado en el desarrollo del proyecto.

6.02 Presupuesto.

Tabla 31

Presupuesto

Nombre	Cantidad	Valor Unitario	Valor Total
Empastado	1	50,00	50,00
Anillado	5	10,00	50,00
Impresiones	600	0,10	60
Suministros de oficina	-----	-----	10,00
Servicios Básicos	800 horas	0,20	160,00
Servicio de internet	500horas	0,080	40,00
Alimentación	100 días	2,00	200,00
Transporte	100 días	1,00	100,00
Seminario de profesionalización	1	520,00	520,00
Tutoría	1	200,00	200,00
Computador	1	1000,00	1000,00
Total			2.390,00

Nota: Muestra el presupuesto con el que se cuenta para la elaboración del proyecto.

6.03 Cronograma.

Muestra las fechas del desarrollo del proyecto incluido el tiempo de la elaboración del sistema.

Ver Anexo A.06

Capítulo VII: Conclusiones y Recomendaciones

7.01 Conclusiones.

- Los registros de compra y venta de mercancía se realizan adecuadamente así permitiendo un óptimo desempeño al realizar de estos procesos.
- Ay una adecuada administración de bodega, lo que ayuda a mejorar el diario desempeño laboral en esta área.
- La compra de productos se realiza de una manera óptima, ya que solo se adquiere lo que la empresa necesita para su crecimiento financiero, la existencia de productos siempre está sobre el nivel mínimo que deberían haber en bodega y no existen errores al generar kardex de productos.
- Se tiene una adecuada verificación de ventas de mercancías, ayudando que los reportes de ventas sean correctos y no generen errores.

7.02 Recomendaciones.

- Para un mejor registro de mercancía se recomienda que en el instante que la mercancía llegue sea ingresada en inventarios de bodega.
- Realizar mensualmente la verificación de cantidades de existencia física de productos almacenados en bodega con las cantidades existentes en el sistema, así evitando fallas al momento de facturar o adquirir.
- Generar reporte de ventas diarias para mantener un control adecuado de ventas de productos.
- Antes de realizar un pedido de compra verificar las existencias de stock en el sistema y luego confirmar con la existencia física.

Referencias Bibliográficas:

Universidad politécnica. (Agosto-2012). Propuesta para la implementación del proceso de facturación electrónica. Recuperado el 15 enero de 2014, de <http://dspace.ups.edu.ec/handle/123456789/2931>

Librosweb. (2014). Programación básica. Recuperado el 10 de diciembre de 2013, de http://librosweb.es/javascript/capitulo_3.html.

DesarrolloWeb.com. (2009). Programación en PHP. Recuperado el 10 de diciembre de 2013, de <http://www.desarrolloweb.com/manuales/12/>

Betabeers.com. Aprender a programar PHP desde cero. Recuperado el 20 de enero de 2014, de <http://betabeers.com/curso/aprender-programar-php-desde-cero-402/>

Mysql. (2011). Crear y utilizar una base de datos. Recuperado 20 de noviembre de 2013, de <http://dev.mysql.com/doc/refman/5.0/es/database-use.html>

Microsoft Developer Network. (2013). Crear proyectos y diagramas de modelado UML. Recuperado 13 de febrero de 2014, de <http://msdn.microsoft.com/es-es/library/dd409445.aspx>

Anexos

A.01 Matriz de análisis de alternativas

Tabla 32.

Matriz de alternativas.

MATRIZ DE ANALISIS DE ALTERNATIVAS							
OBJETIVOS	Impacto sobre el propósito	Factibilidad Técnica	Factibilidad Financiera	Factibilidad Social	Factibilidad Política	Total	Categoría
Correcto registro y control de mercancía.	4	4	5	4	4	21	Alta
Administración de inventarios de bodega.	5	4	4	4	4	21	Alta
Apropiada verificación de ventas de mercancía.	5	5	5	4	5	24	Alta
Eficiente control de existencia de mercadería	4	2	3	2	3	14	Media
TOTAL	18	15	17	14	16	80	Alta

A.02 Matriz de Marco Lógico

Tabla 33.

Análisis del marco lógico.

RESUMEN NARRATIVO	INDICADORES	MEDIOS DE VERIFICACION	SUPUESTOS
FIN DEL PROYECTO			
➤ Adecuado control de inventarios de bodega y facturación	➤ Disminución de errores en labores diarias.	➤ Rapidez al emitir facturas. ➤ Agilidad al ingresar productos a las existencias de bodega.	➤ Tener clientes satisfechos y disminución de la carga de trabajo a las áreas de facturación y bodega.
PROPOSITO DEL PROYECTO			
➤ Correcto registro y control de mercancía.	➤ Un adecuado manejo de los procesos de facturación y bodega.	➤ Reporte ingresos y egresos de productos.	➤ Adecuado control de inventarios de bodega y facturación
➤ Administración de inventarios de bodega.			
➤ Apropiada verificación de ventas de mercancía			
➤ Eficiente control de existencia de mercadería			

RESUMEN NARRATIVO	INDICADORES	MEDIOS DE VERIFICACION	SUPUESTOS
COMPONENTE DEL PROYECTO			
➤ Tener un correcto manejo del proceso de compra de mercancías. ➤ Obtener conocimiento de administración de bodegas. ➤ Los procesos de facturación y bodega sistematizados	➤ Reporte de compras sin errores. ➤ Reporte de ventas correcto	➤ Sistema de control de inventarios y facturación. ➤ Aumento de rapidez en la toma de decisiones por porta de gerencia en la compra de productos	➤ Correcto registro y control de mercancía. ➤ Administración de inventarios de bodega. ➤ Apropiada verificación de ventas de mercancía ➤ Los procesos de facturación y bodega sistematizados
ACTIVIDADES DEL PROYECTO			
➤ Generar correctamente los pedidos de productos. ➤ Capacitación sobre administrar los inventarios de bodega. ➤ Controlar la existencia de mercadería.	➤ Ingreso de facturas ➤ Reportes de ventas y kardex de productos correctos. ➤ Reporte de existencias de productos ➤ Reportes de ventas y compras de productos diarias.	➤ Kardex de bodega ➤ Lista de productos existentes.	

A.03 Casos de Uso

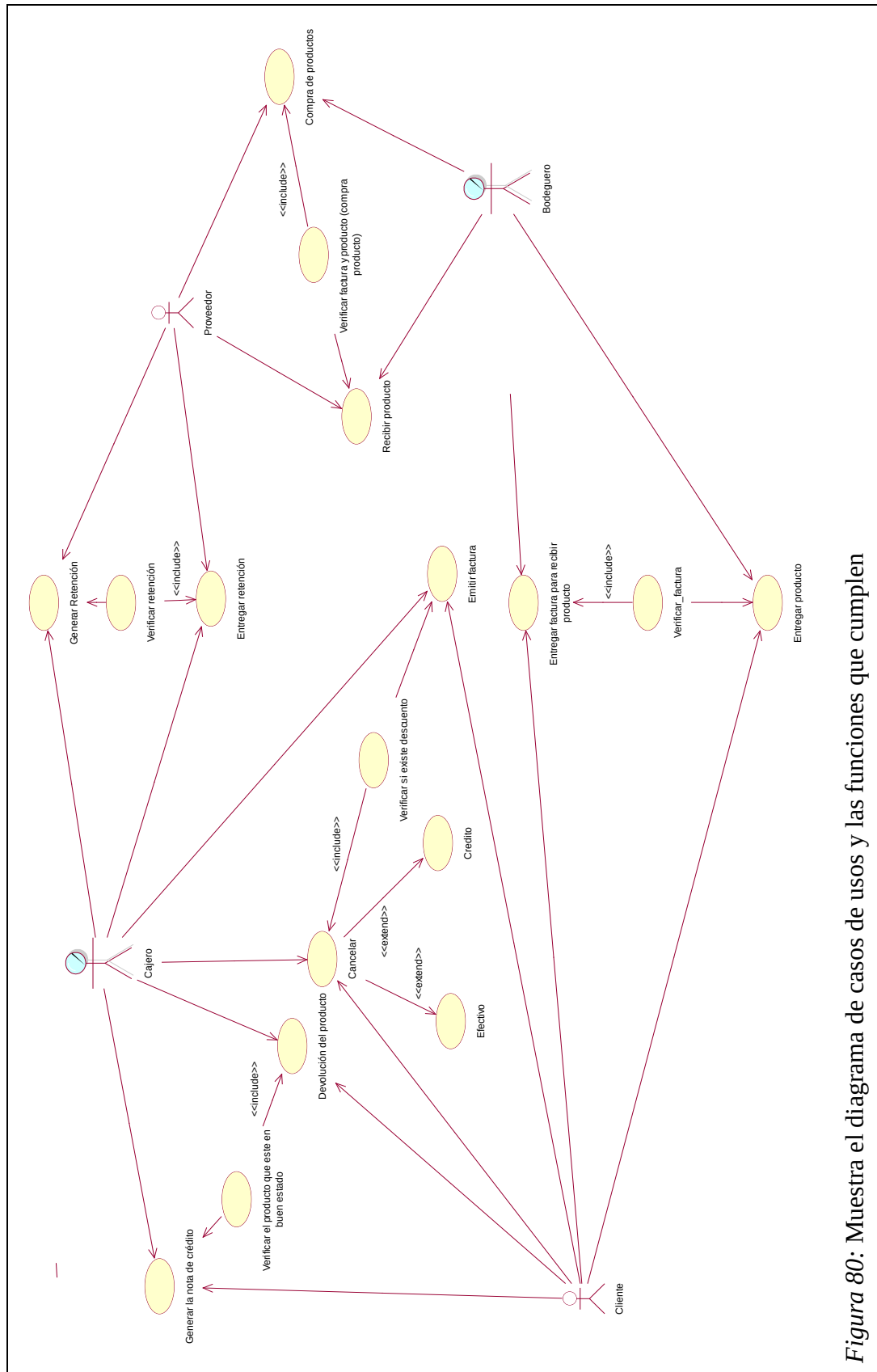


Figura 80: Muestra el diagrama de casos de usos y las funciones que cumplen

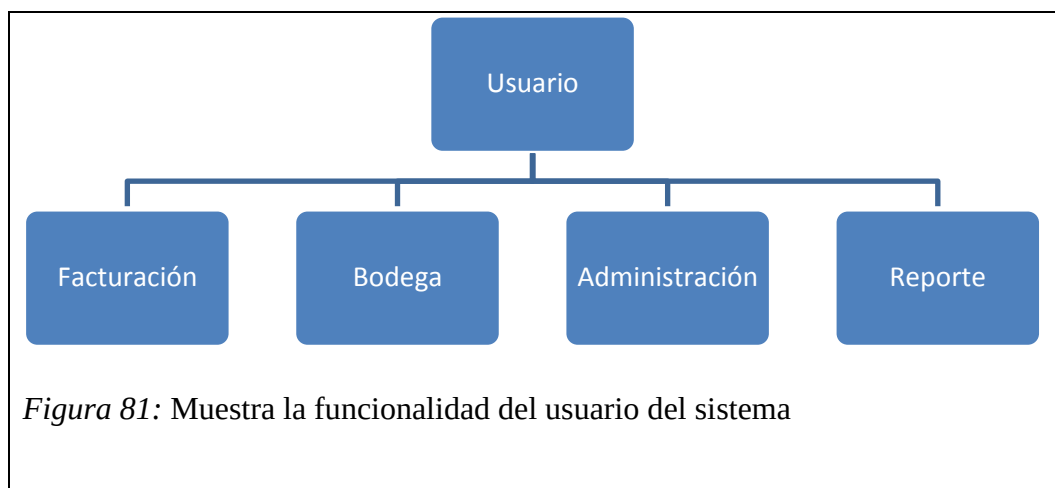
A.04 Manual de Usuario

A.04.a Propósito del Documento

El presente documento está dirigido a entregar las pautas de operación del Sistema de Facturación y control de inventarios de bodega. Este sistema permite el control de inventarios de bodega y la generación de ventas por medio de emisión de facturas.

El Jeremy es una aplicación Web que permitirá una interacción más transparente y eficaz tanto para el cajero como para el bodeguero y el gerente.

La siguiente figura muestra la funcionalidad del Usuario del Sistema de Facturación y control de inventarios de bodega



A.04.b Conceptos Importantes

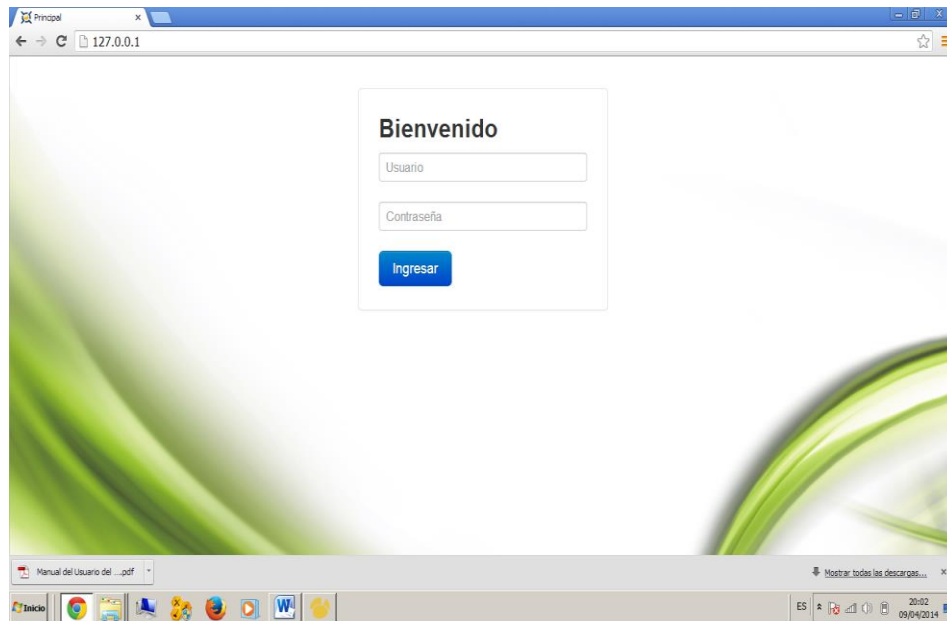


Figura 82: Se debe ingresar al navegador web google chrome, en la barra de dirección escribimos el ip del servidor donde se aloja la aplicación.

Para ingresar al sistema se debe ingresar el usuario y la contraseña y presionar el botón ingresar



Bienvenido

Usuario

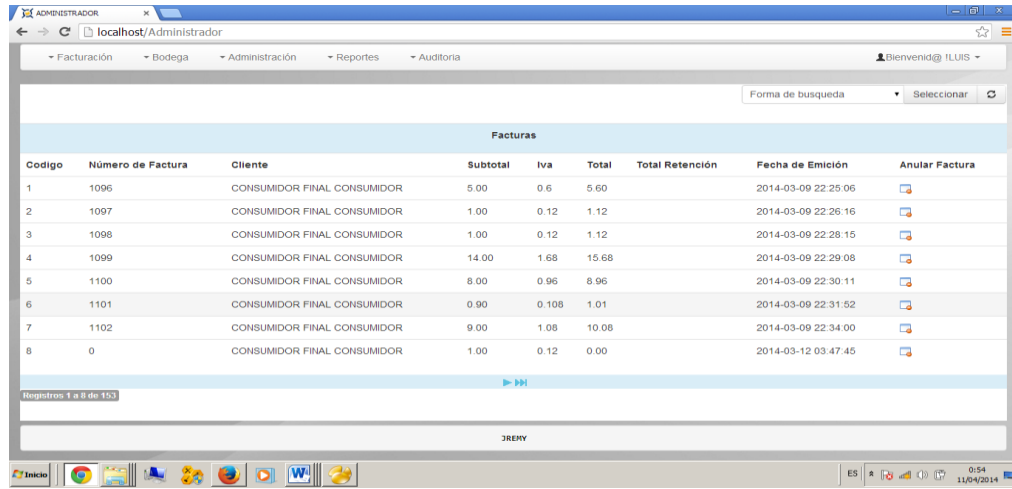
Contraseña

Figura 83: Ingresar usuario y contraseña



Figura 84: Ingresar al menú principal de la aplicación.

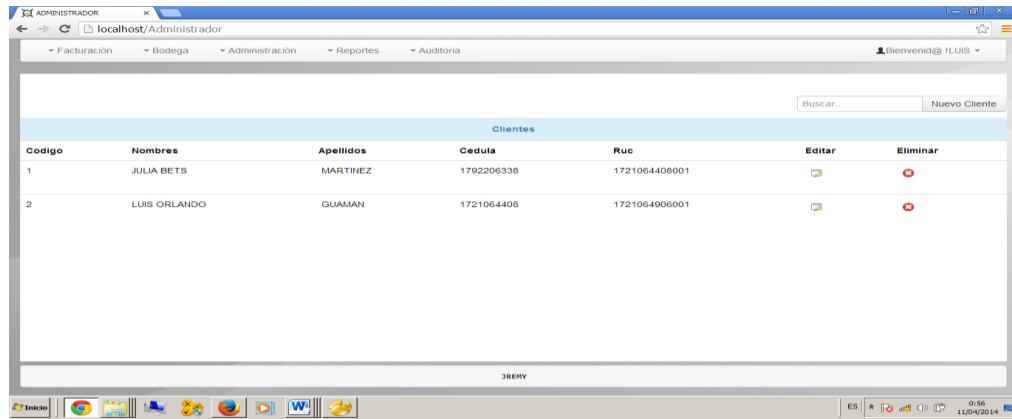
Una vez que se haya ingresado al sistema nos ubicaremos en el menú principal donde podremos escoger las acciones que vamos a realizar.



Código	Número de Factura	Cliente	Subtotal	Iva	Total	Total Retención	Fecha de Emisión	Anular Factura
1	1096	CONSUMIDOR FINAL CONSUMIDOR	5.00	0.6	5.60		2014-03-09 22:25:06	
2	1097	CONSUMIDOR FINAL CONSUMIDOR	1.00	0.12	1.12		2014-03-09 22:26:16	
3	1098	CONSUMIDOR FINAL CONSUMIDOR	1.00	0.12	1.12		2014-03-09 22:28:15	
4	1099	CONSUMIDOR FINAL CONSUMIDOR	14.00	1.68	15.68		2014-03-09 22:29:08	
5	1100	CONSUMIDOR FINAL CONSUMIDOR	8.00	0.96	8.96		2014-03-09 22:30:11	
6	1101	CONSUMIDOR FINAL CONSUMIDOR	0.00	0.108	1.01		2014-03-09 22:31:52	
7	1102	CONSUMIDOR FINAL CONSUMIDOR	9.00	1.08	10.08		2014-03-09 22:34:00	
8	0	CONSUMIDOR FINAL CONSUMIDOR	1.00	0.12	0.00		2014-03-12 03:47:45	

Figura 85: Se puede anular facturas.

Se podrá anular las facturas, realizando una búsqueda por fecha o por número.



Código	Nombres	Apellidos	Cedula	Rue	Editar	Eliminar
1	JULIA BETS	MARTINEZ	1792206336	1721064408001		
2	LUIS ORLANDO	GUAMAN	1721064408	1721064906001		

Figura 86: Listado de clientes.

En esta pantalla se podrá administrar todos los clientes, ya sea crear nuevos clientes, eliminar o actualizar los ya existentes.

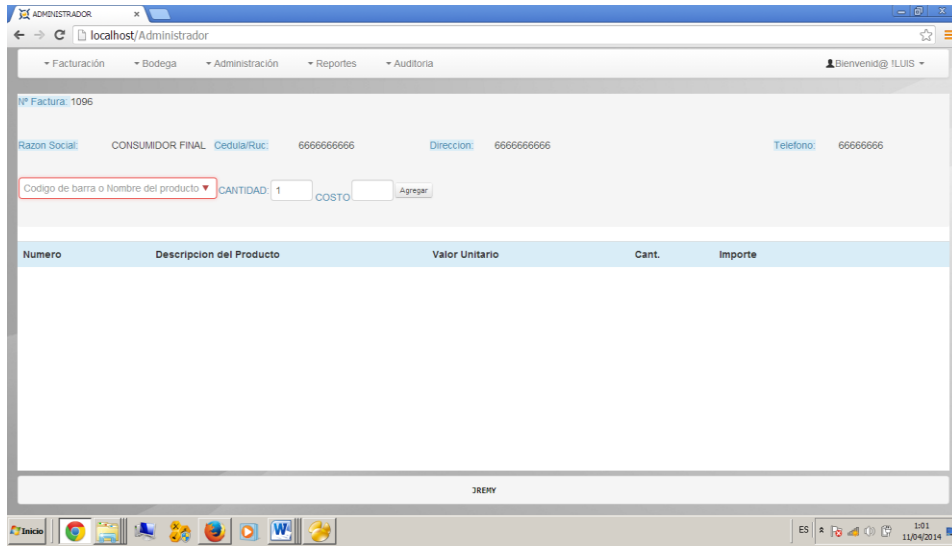
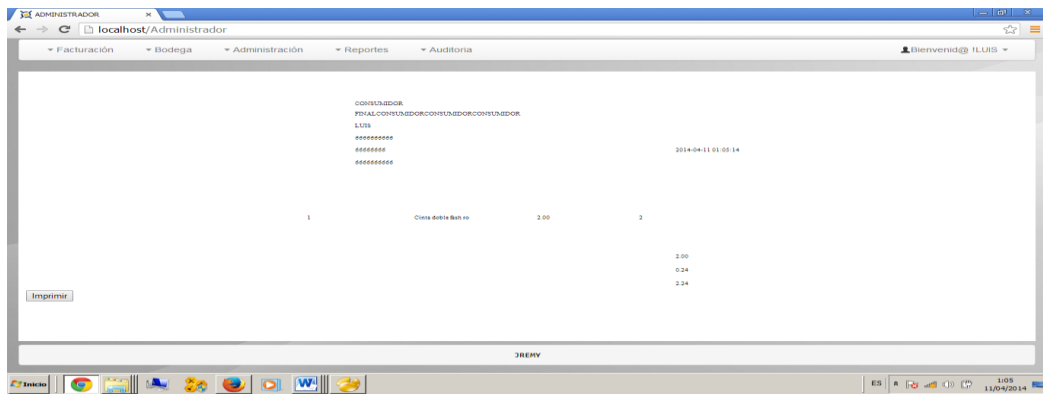


Figura 87: Ingreso de productos devueltos.

Se ingresara las cantidades de los productos devueltos.



Numero	Descripcion del Producto	Valor Unitario	Cant.	Importe
1	Cinta de cinta Bath ro	2.00	2	2.00
				0.24
				2.24

Figura 88: Impresión de nota de crédito.

Se podrá imprimir la nota de crédito para entregarla al cliente

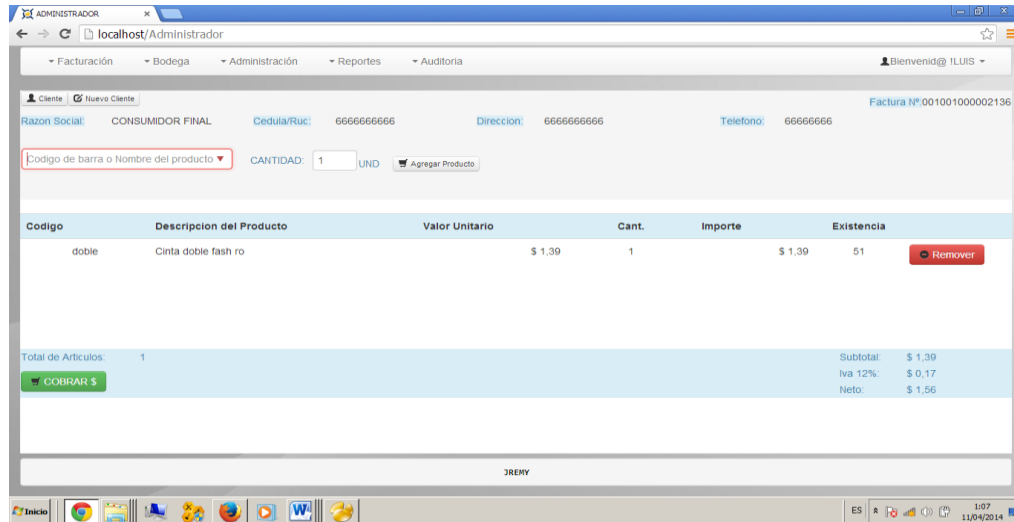


Figura 89: Pantalla para generar factura.

Se podrá generar una factura así como, ingresar un nuevo cliente o buscar un existente.

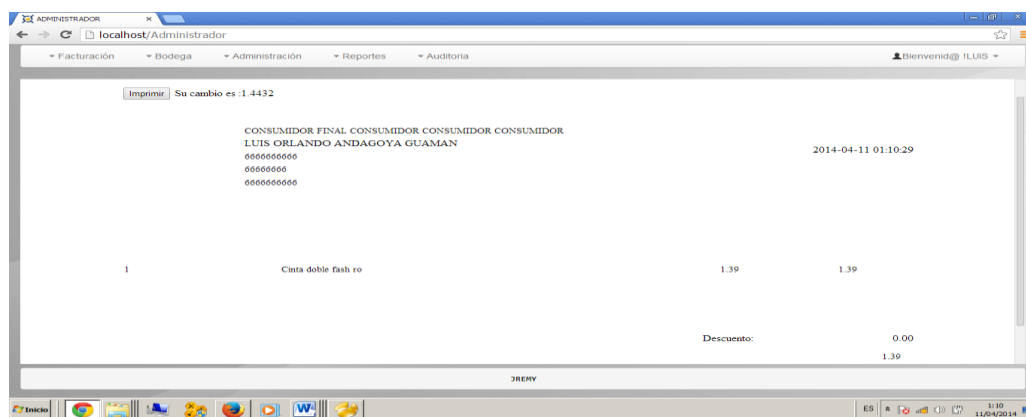
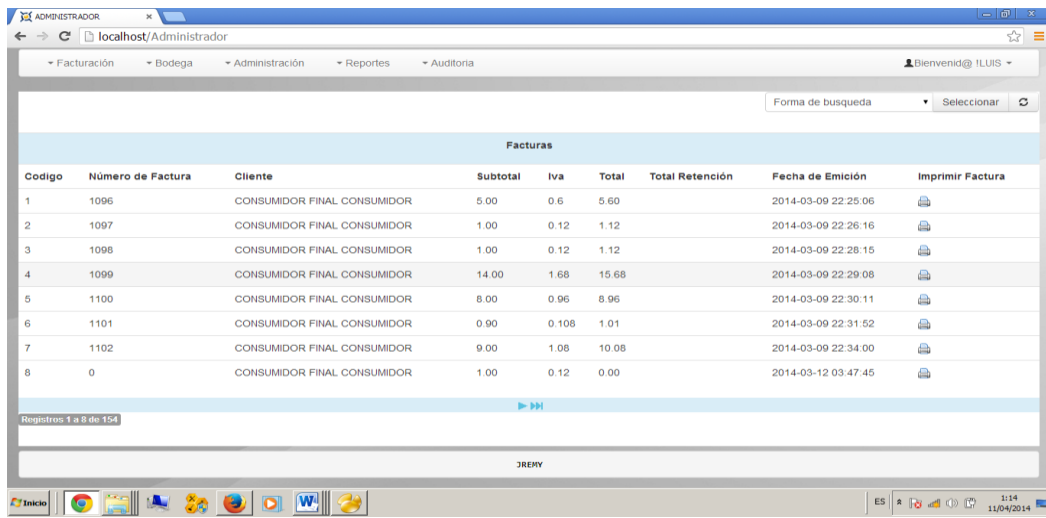


Figura 90: Pantalla para imprimir la factura.

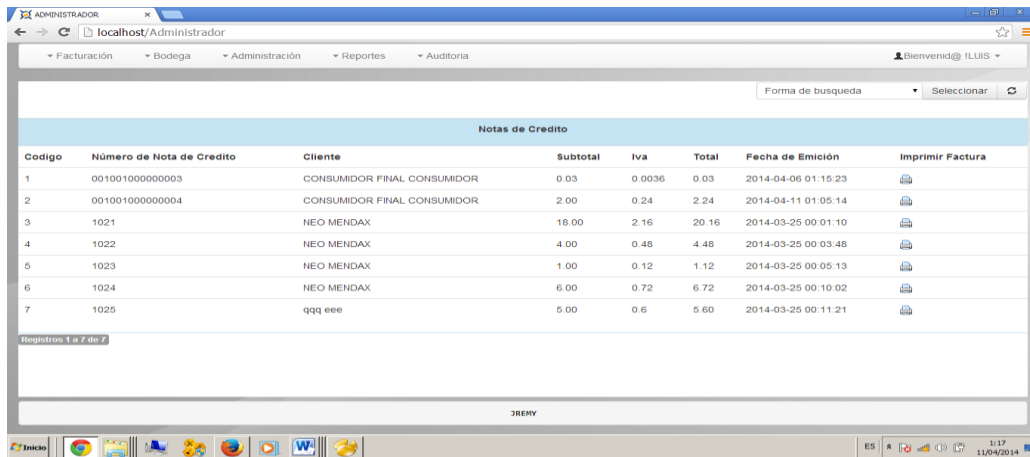
Se podrá imprimir la factura para entregarla al cliente.



Código	Número de Factura	Cliente	Subtotal	Iva	Total	Total Retención	Fecha de Emisión	Imprimir Factura
1	1096	CONSUMIDOR FINAL CONSUMIDOR	5.00	0.6	5.60		2014-03-09 22:25:06	
2	1097	CONSUMIDOR FINAL CONSUMIDOR	1.00	0.12	1.12		2014-03-09 22:26:16	
3	1098	CONSUMIDOR FINAL CONSUMIDOR	1.00	0.12	1.12		2014-03-09 22:28:15	
4	1099	CONSUMIDOR FINAL CONSUMIDOR	14.00	1.68	15.68		2014-03-09 22:29:08	
5	1100	CONSUMIDOR FINAL CONSUMIDOR	8.00	0.96	8.96		2014-03-09 22:30:11	
6	1101	CONSUMIDOR FINAL CONSUMIDOR	0.90	0.108	1.01		2014-03-09 22:31:52	
7	1102	CONSUMIDOR FINAL CONSUMIDOR	9.00	1.08	10.08		2014-03-09 22:34:00	
8	0	CONSUMIDOR FINAL CONSUMIDOR	1.00	0.12	0.00		2014-03-12 03:47:45	

Figura 91: Lista de facturas para imprimir.

En este listado se podrá imprimir las facturas ya generadas.



Código	Número de Nota de Credito	Cliente	Subtotal	Iva	Total	Fecha de Emisión	Imprimir Factura
1	001001000000003	CONSUMIDOR FINAL CONSUMIDOR	0.03	0.0036	0.03	2014-04-06 01:15:23	
2	001001000000004	CONSUMIDOR FINAL CONSUMIDOR	2.00	0.24	2.24	2014-04-11 01:05:14	
3	1021	NEO MENDAX	18.00	2.16	20.16	2014-03-25 00:01:10	
4	1022	NEO MENDAX	4.00	0.48	4.48	2014-03-25 00:03:48	
5	1023	NEO MENDAX	1.00	0.12	1.12	2014-03-25 00:05:13	
6	1024	NEO MENDAX	6.00	0.72	6.72	2014-03-25 00:10:02	
7	1025	qqq eee	5.00	0.6	5.60	2014-03-25 00:11:21	

Figura 92: Listado de notas de crédito para imprimir.

En este listado podremos escoger las notas de crédito ya generadas e imprimirlas.

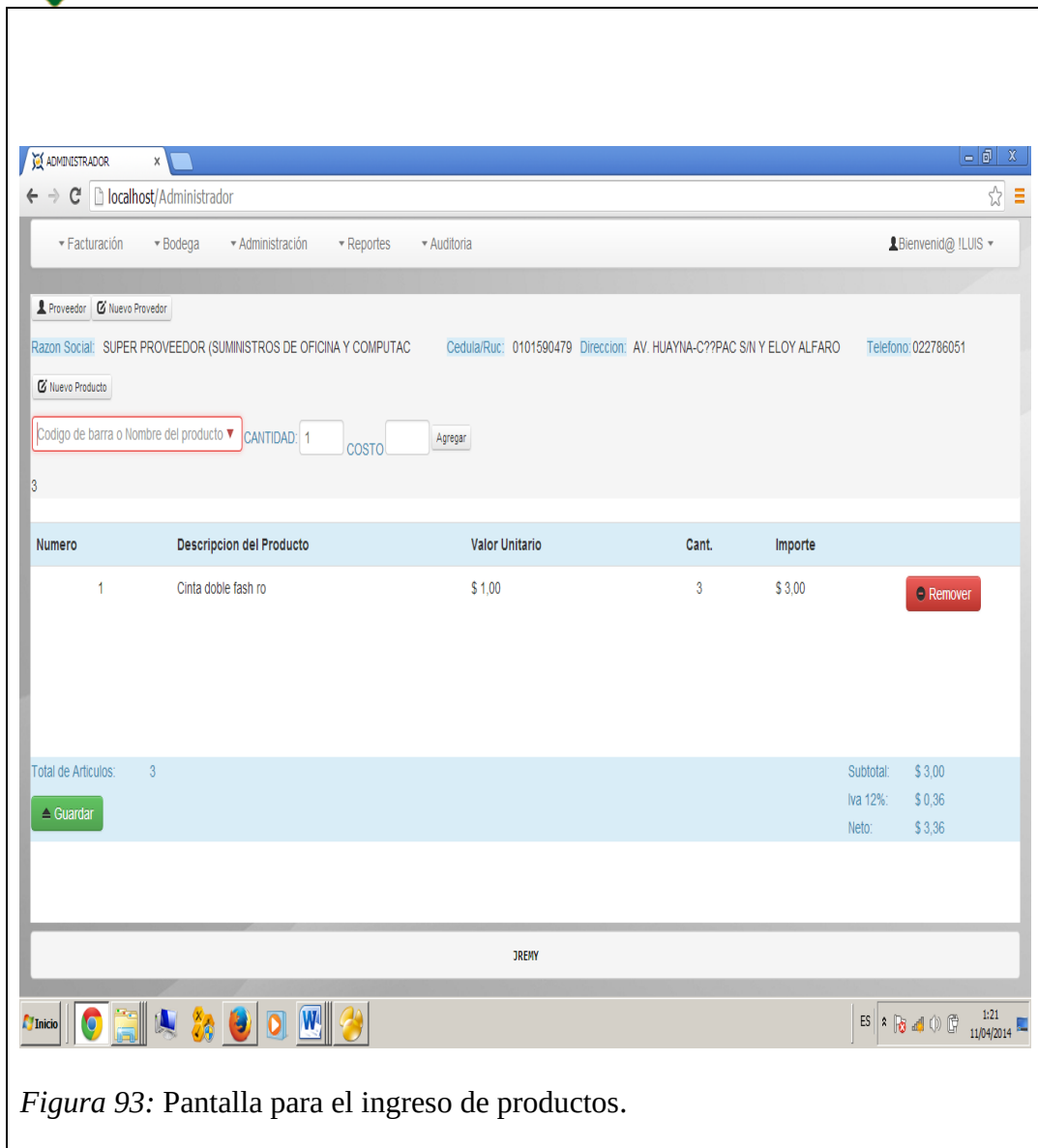


Figura 93: Pantalla para el ingreso de productos.

En esta pantalla se podrá ingresar las compras de productos nacionales, registrar los costos, las cantidades, también los proveedores y el número de factura.

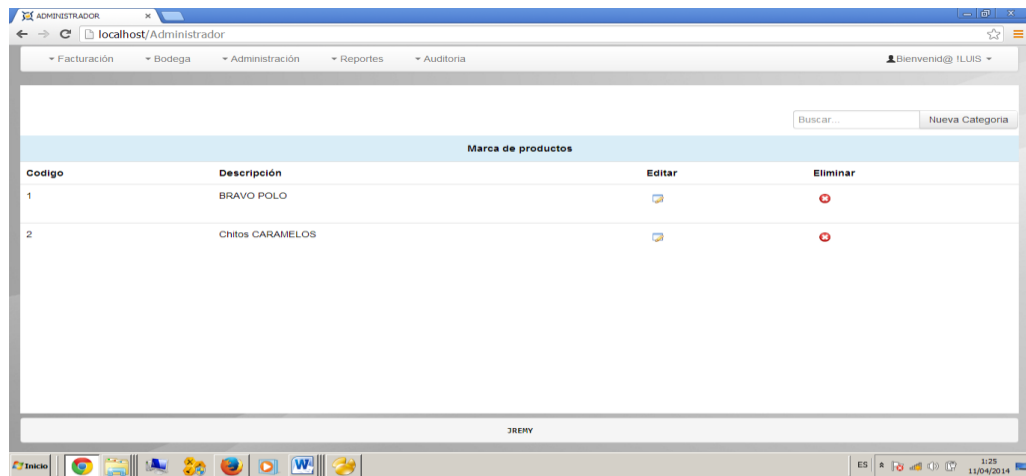


Figura 94: Pantalla para la administración de marca de productos.

Se podrá administrar las marcas de los productos ya sea crear, modificar o eliminar una marca de producto.

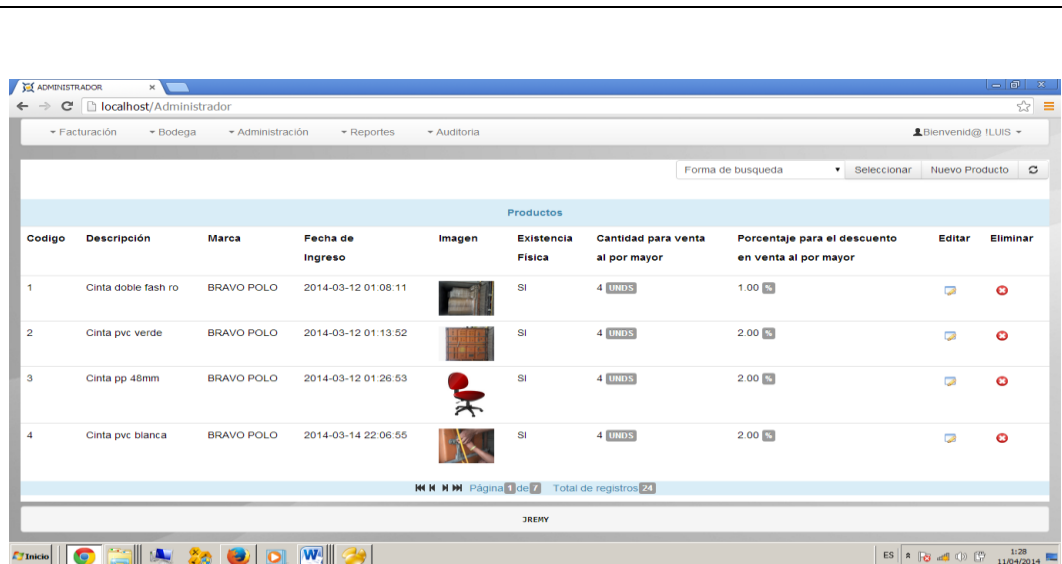
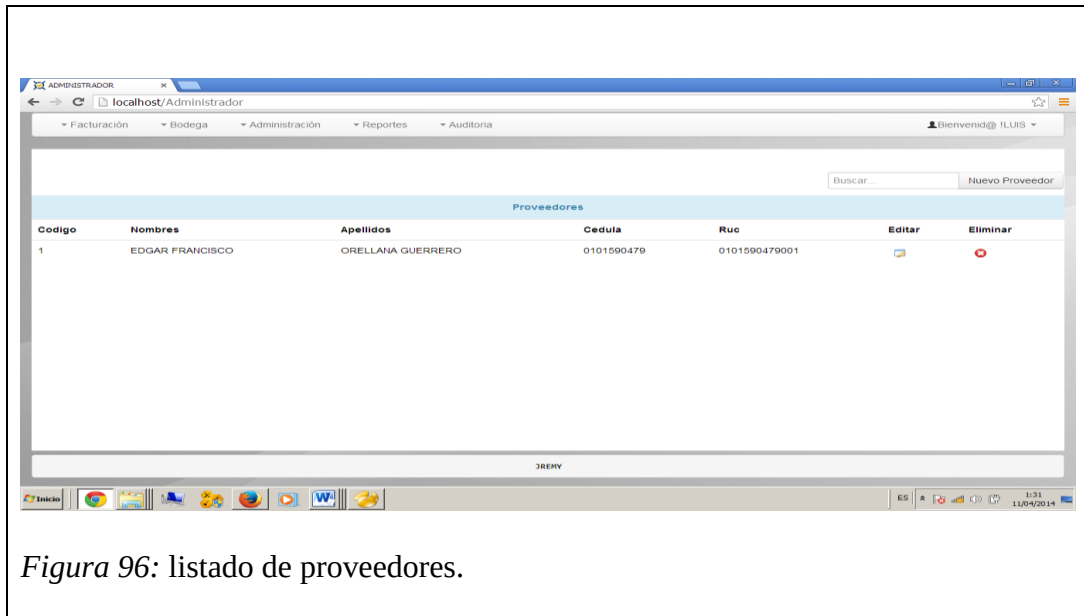
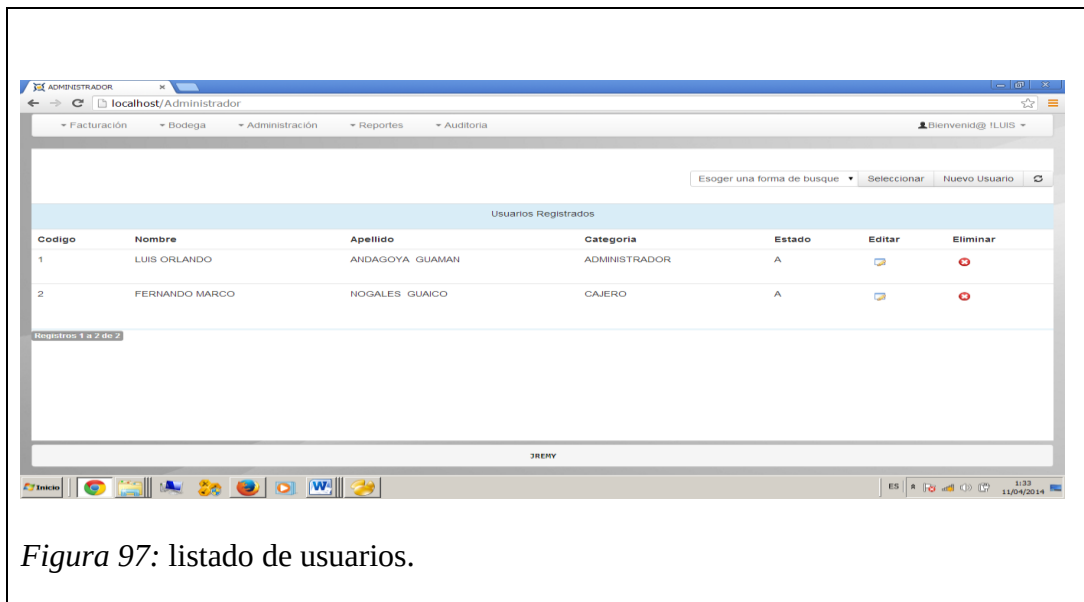


Figura 95: Pantalla para la administración de productos.

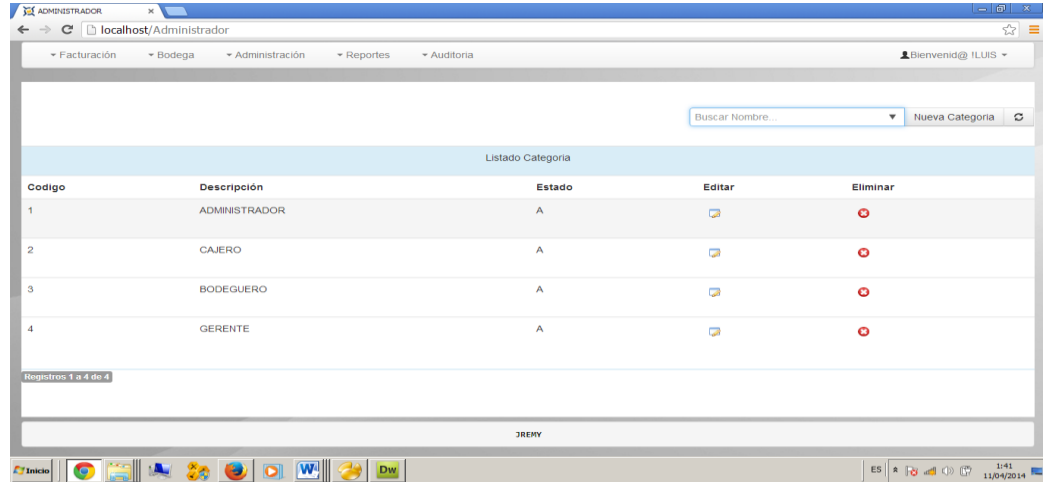
Se podrá administrar los productos existentes en bodega ya sea crear nuevos, eliminar o modificar.



Se podrá administrar los proveedores como lo es crear, eliminar y modificar un proveedor.



En esta pantalla se podrá administrar los usuarios ya sea cajero, bodeguero, gerente, administradores los cuales podrán ser creados, eliminados o actualizados.



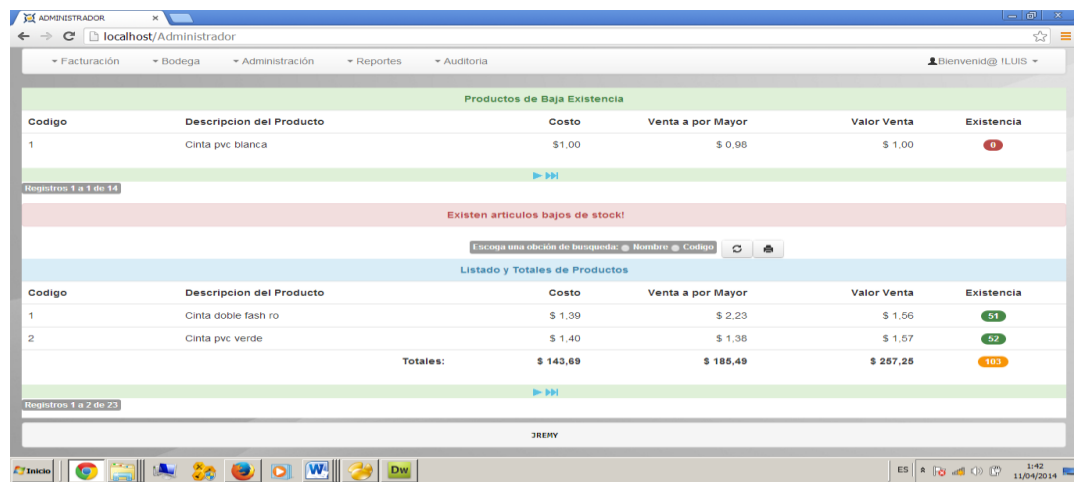
Código	Descripción	Estado	Editar	Eliminar
1	ADMINISTRADOR	A		
2	CAJERO	A		
3	BODEGUERO	A		
4	GERENTE	A		

Registros 1 a 4 de 4

JREHY

Figura 98: listado de categorías.

Se podrá administrar las categorías de usuario modificar o eliminar.



Código	Descripción del Producto	Costo	Venta a por Mayor	Valor Venta	Existencia
1	Cinta pvc blanca	\$ 1.00	\$ 0.98	\$ 1.00	

Registros 1 a 1 de 14

Existen artículos bajos de stock!

Escoge una opción de búsqueda: @ Nombre @ Código

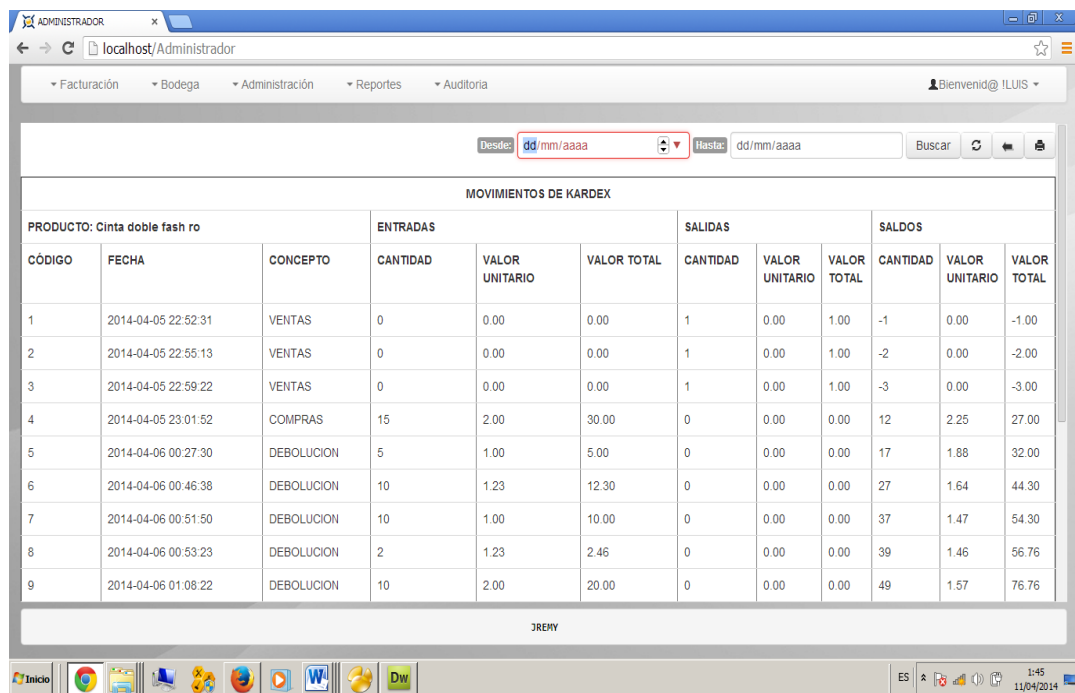
Código	Descripción del Producto	Costo	Venta a por Mayor	Valor Venta	Existencia
1	Cinta doble fash ro	\$ 1.39	\$ 2.23	\$ 1.56	
2	Cinta pvc verde	\$ 1.40	\$ 1.38	\$ 1.57	
Totales:		\$ 143.69	\$ 185.49	\$ 257.26	

Registros 1 a 2 de 23

JREHY

Figura 99: Estado de productos.

Se podrá visualizar las existencias de productos, mirar que productos tienen bajo stock e imprimir los estados de productos para inventarios físicos de bodega.



ADMINISTRADOR

localhost/Administrador

Facturación Bodega Administración Reportes Auditoria

Bienvenid@ ILUIS

Desde: dd/mm/aaaa Hasta: dd/mm/aaaa Buscar

MOVIMIENTOS DE KARDEX

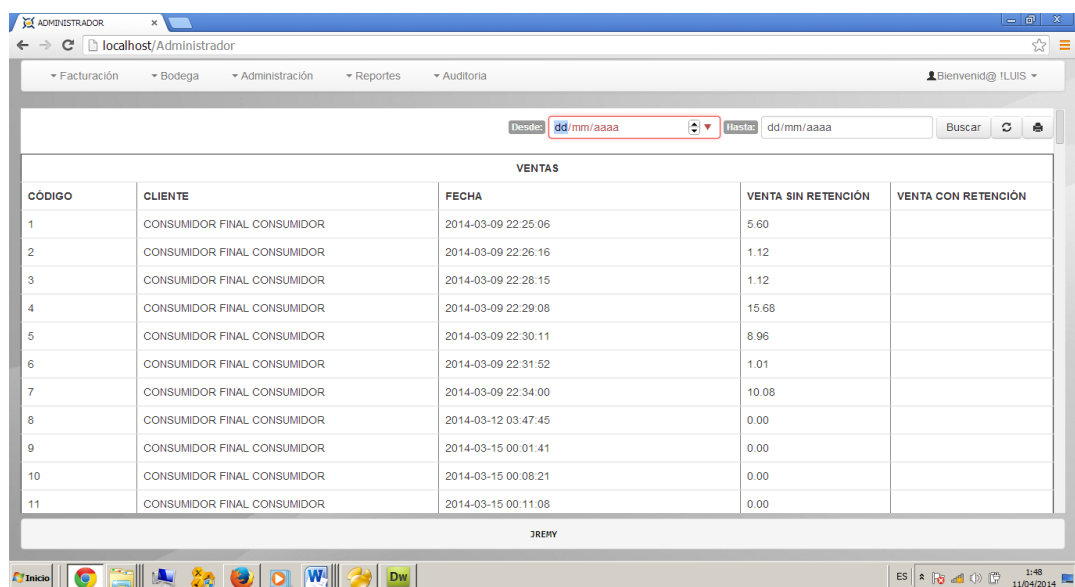
PRODUCTO: Cinta doble fash ro

CÓDIGO	FECHA	CONCEPTO	ENTRADAS			SALIDAS			SALDOS		
			CANTIDAD	VALOR UNITARIO	VALOR TOTAL	CANTIDAD	VALOR UNITARIO	VALOR TOTAL	CANTIDAD	VALOR UNITARIO	VALOR TOTAL
1	2014-04-05 22:52:31	VENTAS	0	0.00	0.00	1	0.00	1.00	-1	0.00	-1.00
2	2014-04-05 22:55:13	VENTAS	0	0.00	0.00	1	0.00	1.00	-2	0.00	-2.00
3	2014-04-05 22:59:22	VENTAS	0	0.00	0.00	1	0.00	1.00	-3	0.00	-3.00
4	2014-04-05 23:01:52	COMPRAS	15	2.00	30.00	0	0.00	0.00	12	2.25	27.00
5	2014-04-06 00:27:30	DEBOLUCION	5	1.00	5.00	0	0.00	0.00	17	1.88	32.00
6	2014-04-06 00:46:38	DEBOLUCION	10	1.23	12.30	0	0.00	0.00	27	1.64	44.30
7	2014-04-06 00:51:50	DEBOLUCION	10	1.00	10.00	0	0.00	0.00	37	1.47	54.30
8	2014-04-06 00:53:23	DEBOLUCION	2	1.23	2.46	0	0.00	0.00	39	1.46	56.76
9	2014-04-06 01:08:22	DEBOLUCION	10	2.00	20.00	0	0.00	0.00	49	1.57	76.76

JREMY

Figura 100: Kardex de producto.

Se podrá imprimir el kardex de producto de bodega.



ADMINISTRADOR

localhost/Administrador

Facturación Bodega Administración Reportes Auditoria

Bienvenid@ ILUIS

Desde: dd/mm/aaaa Hasta: dd/mm/aaaa Buscar

VENTAS

CÓDIGO	CLIENTE	FECHA	VENTA SIN RETENCIÓN	VENTA CON RETENCIÓN
1	CONSUMIDOR FINAL CONSUMIDOR	2014-03-09 22:25:06	5.60	
2	CONSUMIDOR FINAL CONSUMIDOR	2014-03-09 22:26:16	1.12	
3	CONSUMIDOR FINAL CONSUMIDOR	2014-03-09 22:28:15	1.12	
4	CONSUMIDOR FINAL CONSUMIDOR	2014-03-09 22:29:08	15.68	
5	CONSUMIDOR FINAL CONSUMIDOR	2014-03-09 22:30:11	8.96	
6	CONSUMIDOR FINAL CONSUMIDOR	2014-03-09 22:31:52	1.01	
7	CONSUMIDOR FINAL CONSUMIDOR	2014-03-09 22:34:00	10.08	
8	CONSUMIDOR FINAL CONSUMIDOR	2014-03-12 03:47:45	0.00	
9	CONSUMIDOR FINAL CONSUMIDOR	2014-03-15 00:01:41	0.00	
10	CONSUMIDOR FINAL CONSUMIDOR	2014-03-15 00:08:21	0.00	
11	CONSUMIDOR FINAL CONSUMIDOR	2014-03-15 00:11:08	0.00	

JREMY

Figura 101: Se visualizara las ventas de productos.

Se podrá ver los reportes de ventas de productos, además de podrá buscar ventas diarias semanales o anuales.

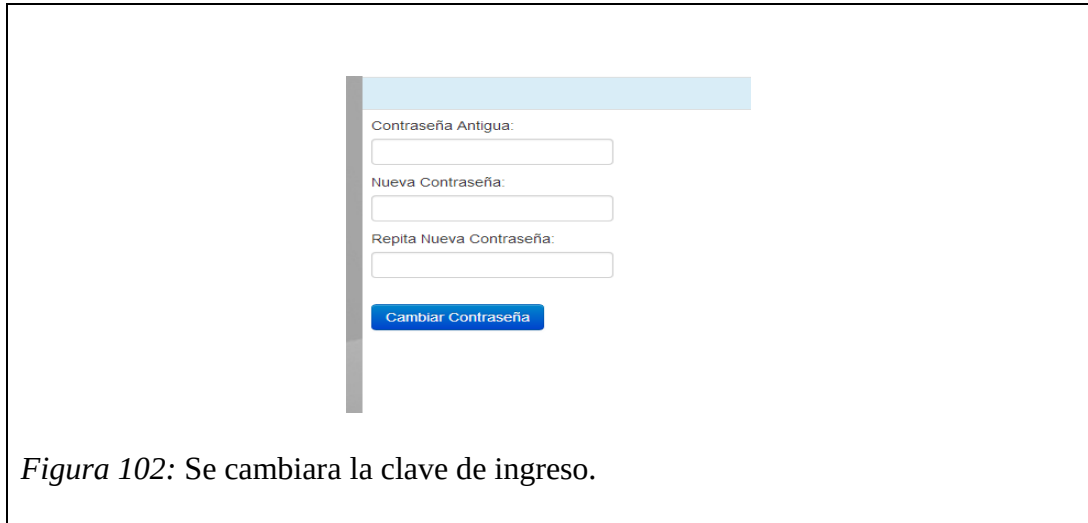
A screenshot of a web application interface for changing a password. The form is centered on a light gray background. It consists of three text input fields stacked vertically, each with a label to its left: 'Contraseña Antigua:', 'Nueva Contraseña:', and 'Repita Nueva Contraseña:'. Below the third input field is a blue button with the text 'Cambiar Contraseña' in white. The entire form is enclosed in a thin black rectangular border.

Figura 102: Se cambiara la clave de ingreso.

A.05 Manual Técnico

A.05.a Instalación de Id de desarrollo

NetBeans es gratuito, de código abierto y hay una comunidad de programadores detrás de él por todo el mundo que lo han convertido en lo que es. Si tuviera que decir una sola cosa de NetBeans diría que su gran virtud es su eficiencia en la generación de código. Hay quien dice que es su mayor defecto y por eso prefiere Eclipse. Para gustos... colores.

Descarga.

En estos momentos, la última versión es NetBeans IDE 7.3. Puede hacerse con ella hay que ir a su web oficial. En la página de bienvenida se pulsa sobre el botón "Download".



Figura 103: Pagina de descarga.

La siguiente página corresponde a la selección de paquetes funcionales.

Lo primero que hay que hacer es seleccionar el idioma. Para ello se despliega el combo de idioma y se realiza la selección que en este caso es Español.

El segundo paso es seleccionar la plataforma. Para ello se despliega el combo de plataforma y se realiza la selección. En este caso Windows.

En tercer y último término hay que seleccionar el paquete funcional. Las opciones disponibles son:

- Java SE. La versión estándar que contiene lo básico.
- Java EE. La versión empresarial para desarrollar aplicaciones web.
- C/C++. Para programar en C/C++. El compilador va aparte.
- PHP. Para programar en PHP. El intérprete PHP también va aparte.
- All. Todo en uno y más.

Si no se tiene problemas de espacio en el disco duro es mejor descargar la opción completa en caso contrario hay que discriminar la versión, y su tamaño, en función de las necesidades. En este caso el objetivo es el pez gordo “Todo en uno” que tiene un peso de poco más de 200 Mbytes de nada y que se obtiene pulsando sobre el correspondiente botón “Download”.

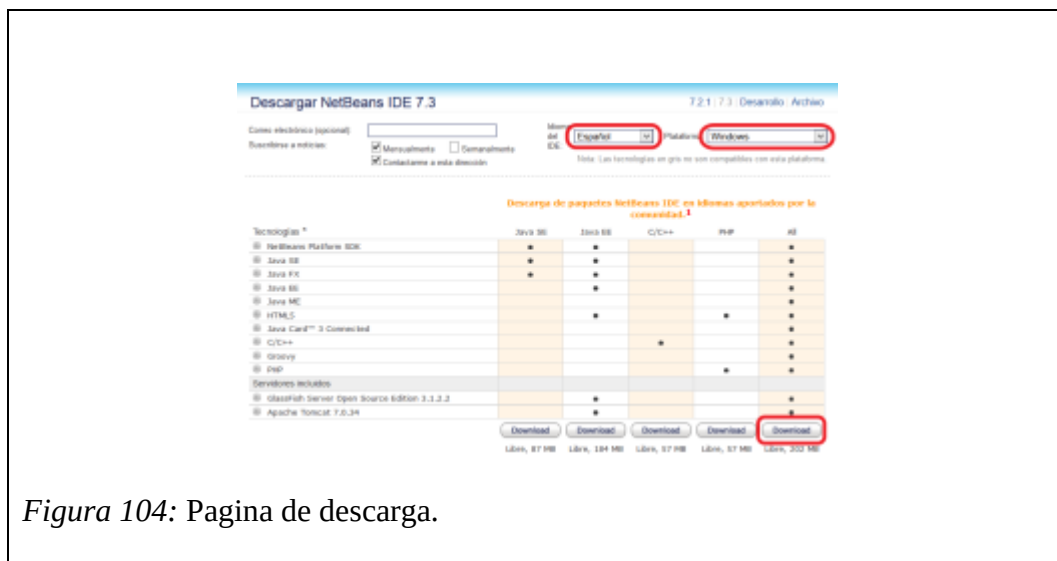


Figura 104: Página de descarga.

Acto seguido el navegador mostrará una página en donde se informa de que la descarga va a empezar y en la que se puede encontrar información y recursos adicionales para NetBeans.



Figura 105: Pagina de descarga.

Un poquitín después debe aparecer un diálogo en el que se pide la confirmación de la descarga. Para ello hay que pulsar sobre el botón “Guardar archivo”.

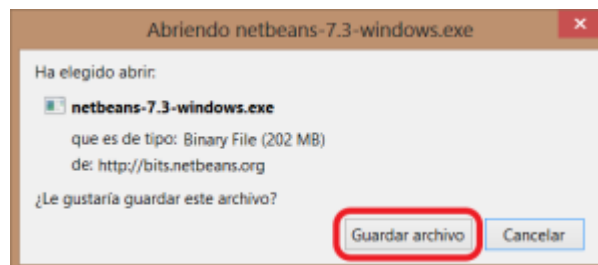


Figura 106: Pagina de descarga.

La descarga del archivo netbeans-7.3-windows.exe de 202 MBytes evoluciona a buen ritmo, los servidores de Oracle y con una conexión a Internet normalita no se demora más allá de los 5 minutos.

Instalación.

Antes de nada hay que tener en cuenta que Windows debe tener instalado un entorno JDK conveniente para que NetBeans funcione correctamente. Lo recomendable es tener instalado un único JDK, de la versión más alta y con la última actualización disponible. En el momento de escribir esta entrada se corresponde con Java Platform (JDK) 7u21.

Pues bien, pasada la comprobación de requisitos el primer paso es ir a la carpeta de descargas y hacer clic derecho con el ratón en el archivo netbeans-7.3-windows.exe. En el menú contextual que se abre hay que hacer clic con el ratón en la opción "Ejecutar como administrador".

En este punto hay que decir que versiones diferentes de NetBeans se instalan sobre carpetas diferentes de disco duro. Esto quiere decir que es posible tener instaladas simultáneamente varias versiones de NetBeans, por lo que la instalación de una nueva versión no debería colisionar con otras versiones ya instaladas.

Tras la confirmación de ejecución del programa en el diálogo de Control de cuentas de usuario se inicia la instalación haciendo una evaluación de la situación del sistema del sistema.

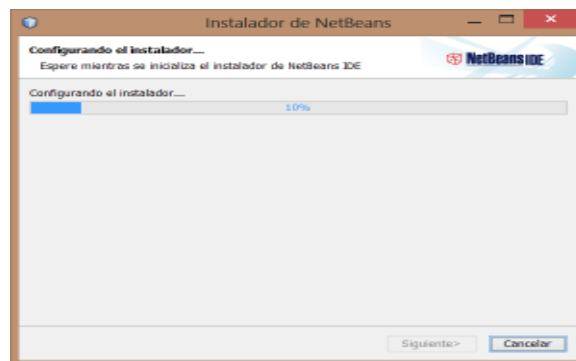


Figura 107: Progreso de instalación.

Cuando termina esta evaluación previa aparece un diálogo que informa de los productos que se van a instalar. Hay que pulsar sobre el botón “Personalizar” para añadir el servidor Apache Tomcat.

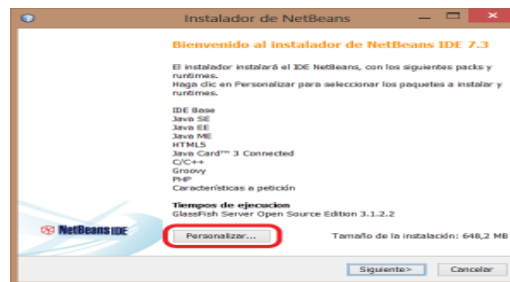


Figura 108: Opciones de instalación.

En el diálogo Personalizar de la instalación se marca la casilla de verificación correspondiente al servidor Apache Tomcat 7.0.34 y se pulsa el botón “Aceptar”

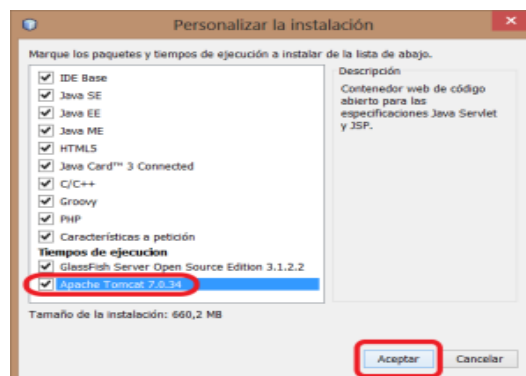


Figura 109: Opciones de instalación.

Ahora la ventana de instalación de NetBeans muestra que se van a instalar todos los productos disponibles. Hay que pulsar sobre el botón “Siguiente”.

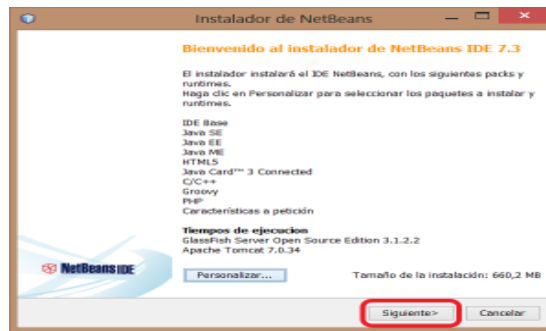


Figura 110: Opciones de instalación.

La siguiente ventana muestra el Contrato de Licencia de NetBeans que deberá leerse y a continuación se deberá activar la casilla de verificación correspondiente a “Aceptar los términos del acuerdo de licencia” para continuar con la instalación. Pulsar el botón “Siguiente”.

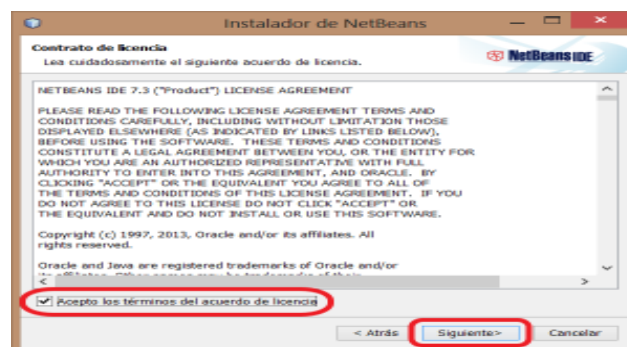


Figura 111: Opciones de instalación.

La siguiente ventana muestra el Contrato de Licencia de JUnit. Este software se utiliza para hacer pruebas de corrección de código y sirve para detectar y eliminar errores de software. Se utiliza en el desarrollo de aplicaciones Java y por ello es muy conveniente instalar JUnit. Así pues se activa el botón “Acepto los términos del contrato de licencia. Instalar JUnit” y se pulsa el botón “Siguiente”.

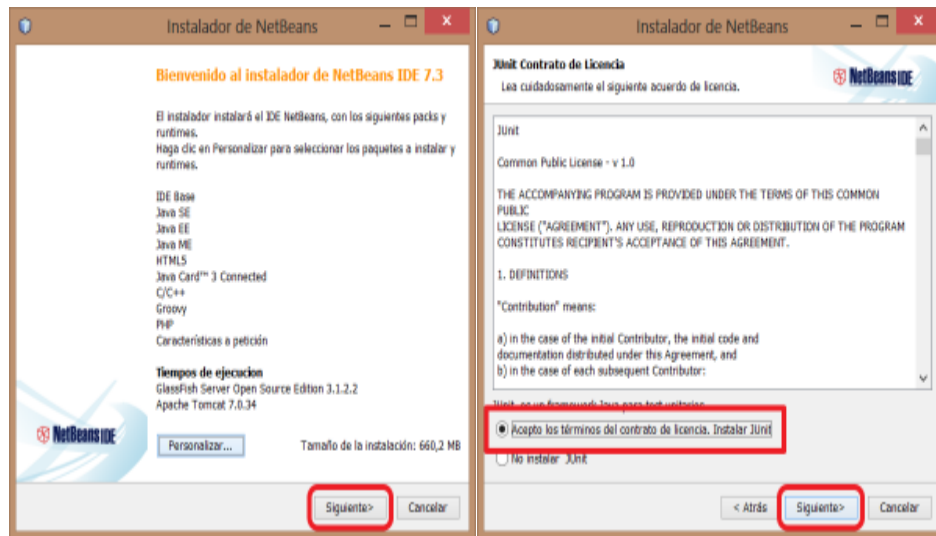


Figura 112: Opciones de instalación.

La siguiente ventana muestra la localización de la carpeta donde se va a instalar NetBeansy la carpeta de la distribución JDK predeterminada. Salvo que haya una buena razón para cambiarlas se mantendrán las localizaciones predeterminadas. Pulsar el botón “Siguiente”.

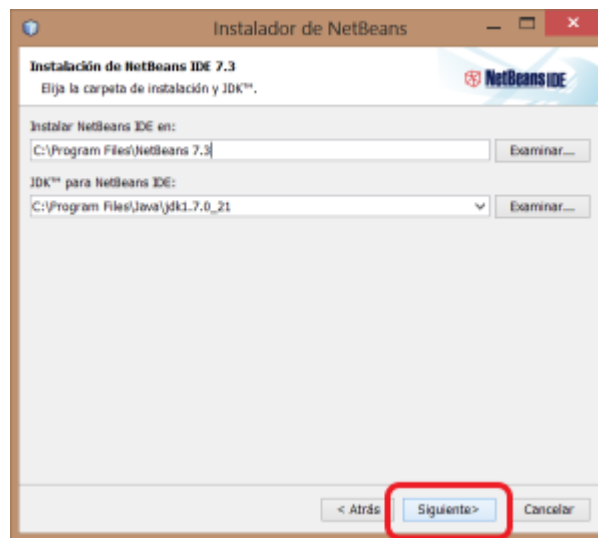


Figura 113: Opciones de instalación.

La siguiente ventana muestra la localización de la carpeta donde se va a instalar el servidor GlassFish. Al igual que antes, salvo que haya una buena razón para cambiarla se mantendrá la localización predeterminada. Pulsar el botón “Siguiente”.

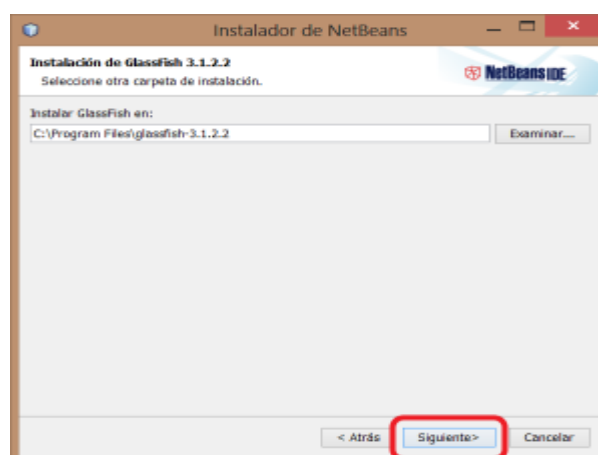


Figura 114: Opciones de instalación.

La siguiente ventana muestra la localización de la carpeta donde se va a instalar el servidor Apache Tomcat. Al igual que antes, salvo que haya una buena razón para cambiarla se mantendrá la localización predeterminada. Pulsar el botón “Siguiente”.

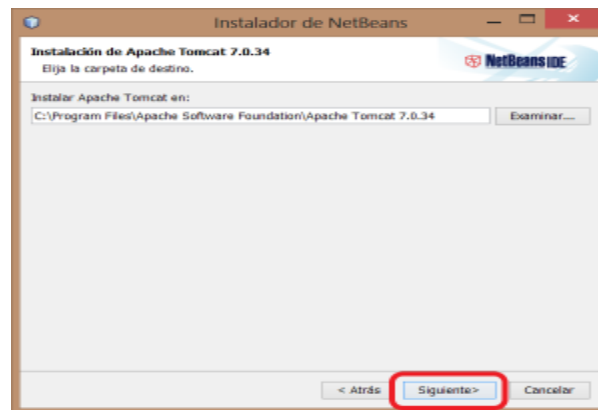


Figura 115: Opciones de instalación.

Antes de que empiece el proceso de copia e instalación de archivos, la última ventana que aparece enumera los productos que se van a instalar. Hay que activar la casilla de comprobación de actualizaciones (Check for updates) que se realizará de forma automática en la parte final de la instalación de NetBeans. Pulsar el botón “Instalar”.

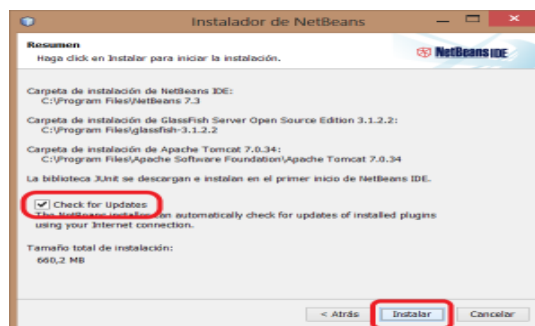


Figura 116: Opciones de instalación.

Este es el proceso más costoso de la instalación. En él una ventana da cuenta del progreso de la instalación cuya duración puede estimarse en 5 minutos al menos, y puede que más.

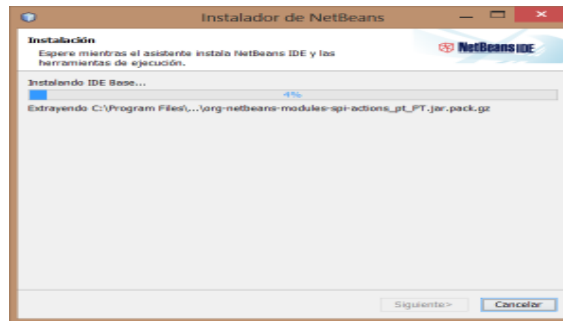


Figura 117: Opciones de instalación.

Una vez terminada la instalación de NetBeans, el instalador se conectará automáticamente a Internet para bajar e instalar JUnit y las actualizaciones disponibles. Cuando termine mostrará una ventana informando de que la instalación se ha completado. Puede dejarse activado o no el envío de estadísticas de utilización. Al pulsar sobre el botón “Terminar” el asistente terminará.

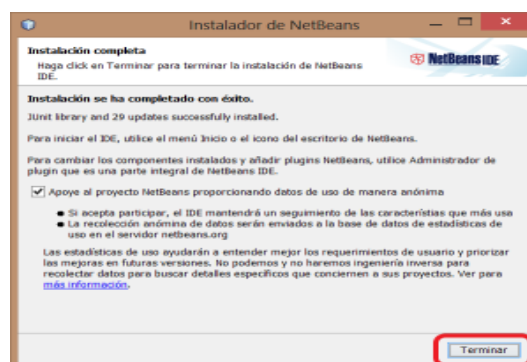


Figura 118: Opciones de instalación.

A.06.b Instalación del servidor

Lo primero que haremos será acceder a la página para descargar "WAMP Server", nosotros lo ejecutaremos directamente para que se instale sin guardar el instalador en nuestro disco duro.

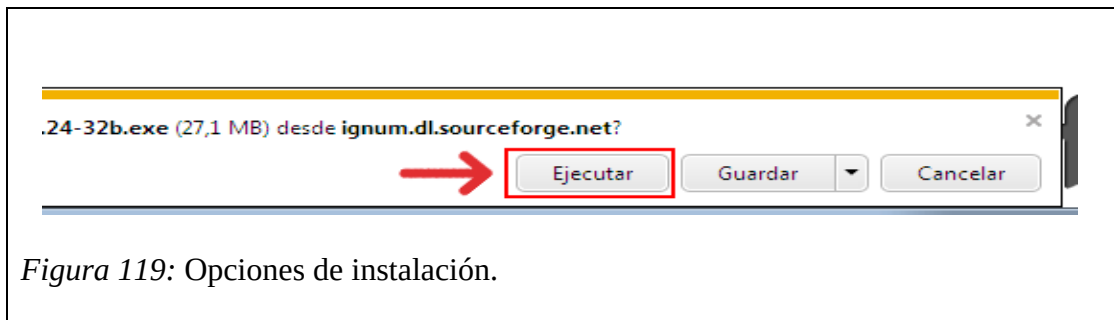


Figura 119: Opciones de instalación.

Si no saliera automáticamente la descarga deberemos pinchar sobre el link "direct link" de la página de SourceForge.

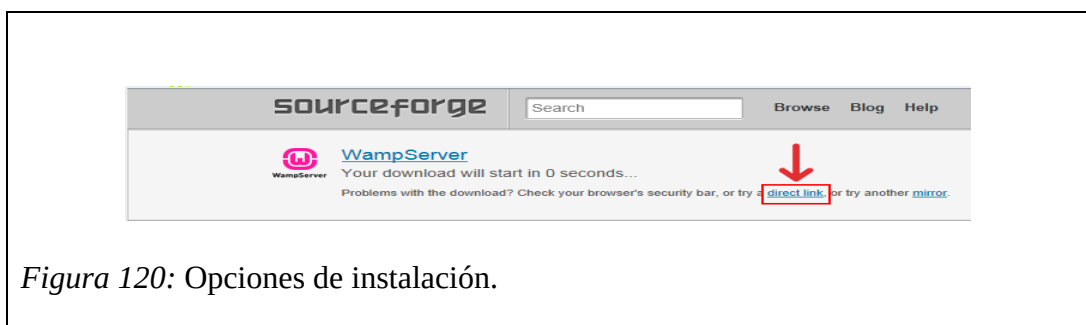


Figura 120: Opciones de instalación.

Cuando pinchemos sobre "Ejecutar" esperamos a que se descargue.

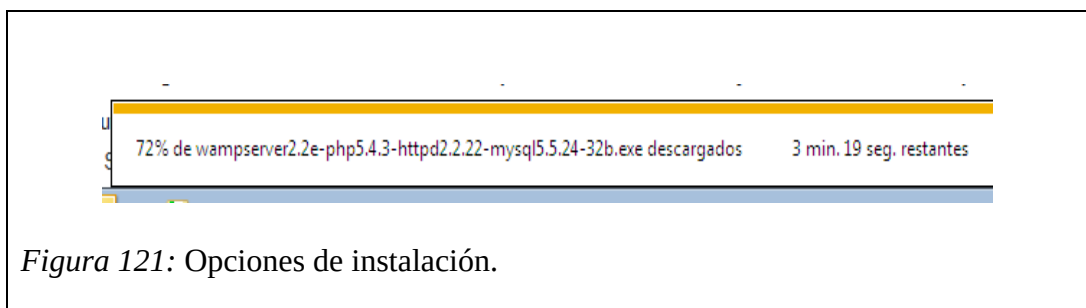


Figura 121: Opciones de instalación.

Una vez descargado se iniciará el instalador del servidor WAMP descargado.

Pincharemos sobre el botón "Next" de la ventana de instalación para continuar.

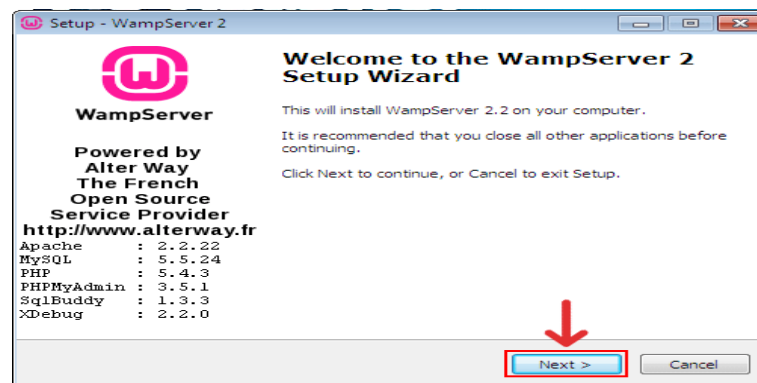


Figura 122: Opciones de instalación.

Seguidamente deberemos aceptar los términos de la licencia, para ello seleccionaremos la opción "I accept the agreement" y pinchamos sobre el botón "Next".

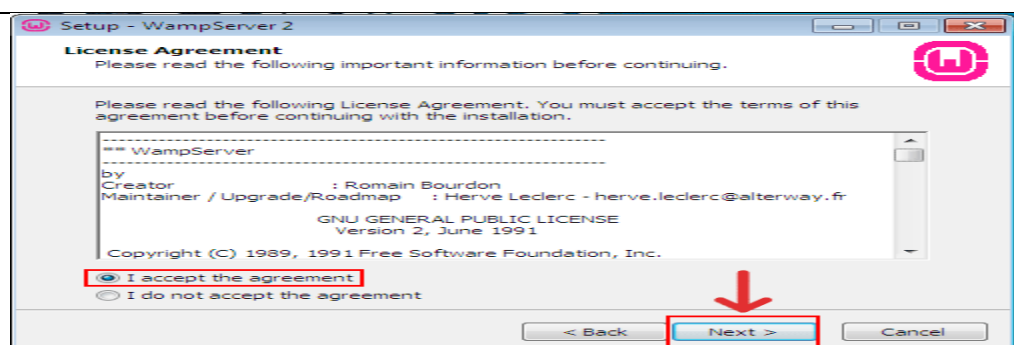


Figura 123: Opciones de instalación.

A continuación seleccionaremos el directorio en nuestro disco duro donde deseemos instalar nuestro servidor WAMP. En este caso lo dejaremos en c:\wamp, ya que será más fácil localizarlo y siempre nos dará un mejor rendimiento. Así que dejamos el directorio de instalación por defecto y pinchamos sobre el botón "Next".

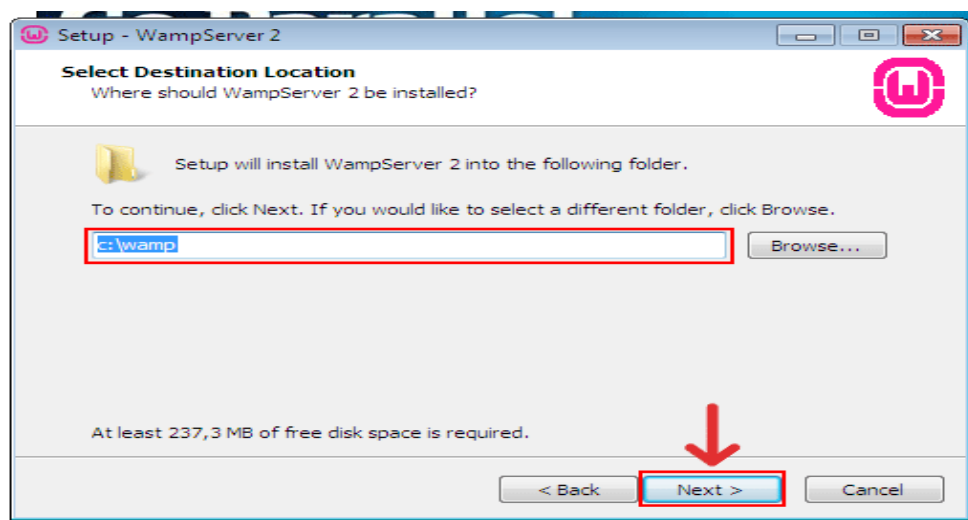


Figura 124: Opciones de instalación.

A continuación nos pregunta qué accesos queremos crear y nos ofrecen 2 posibilidades, "Create a Quick Launch icon" crear un icono en la barra de acceso rápido y "Create a Desktop icon" crear un icono en el escritorio. En esta ocasión sólo elegiremos crear un icono en el escritorio, después pinchamos sobre el botón "Next".

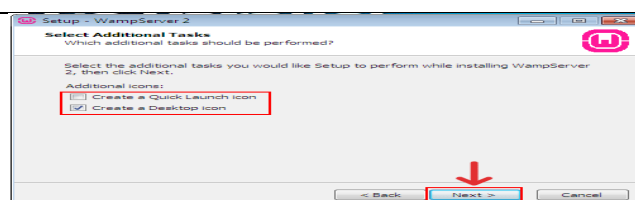


Figura 125: Opciones de instalación.

El siguiente paso es un resumen de todas las opciones que hemos elegido en la instalación, si estamos de acuerdo, que lo estamos, pinchamos en el botón "Install" para iniciar la instalación de nuestro servidor WAMP.

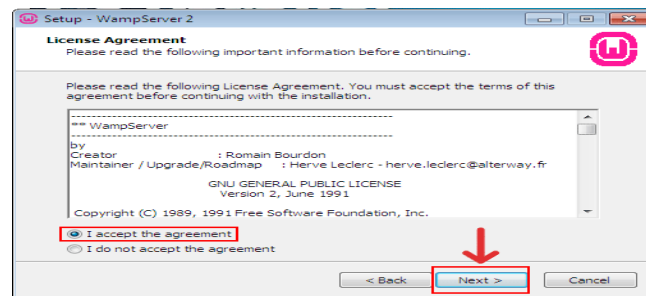


Figura 126: Opciones de instalación.

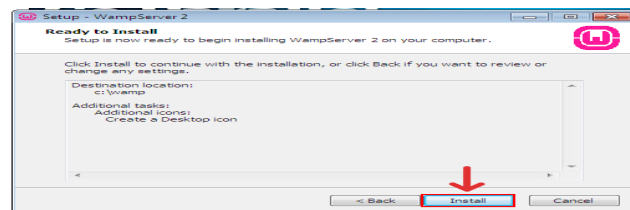


Figura 127: Opciones de instalación.

La instalación se iniciará y esperaremos a que termine.

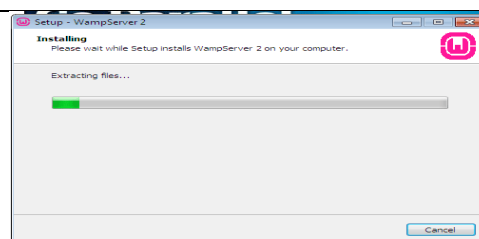


Figura 128: Opciones de instalación.

Antes de terminar la instalación, en algunas ocasiones si tenemos instalado el navegador de internet Firefox nos preguntará si deseamos que Firefox sea nuestro

navegador predeterminado cuando utilicemos WampServer2, lo pregunta de la siguiente manera:

Firefox has been detected en your computer. Would you like to use it as the default browser with WampServer2? "Sí" o "No".

Elegiremos la opción que más nos convenga, si deseamos utilizar Firefox le diremos que Sí, en el caso de que deseemos utilizar el propio Internet Explorer de Windows o el navegador que tengamos predeterminado en Windows le diremos que No.

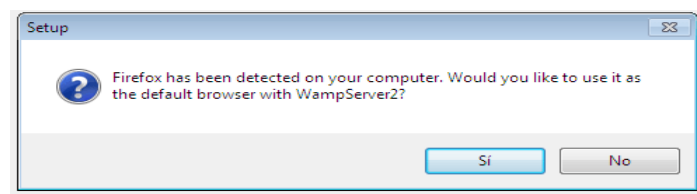


Figura 129: Opciones de instalación.

Si presionamos no mostrará el explorador de archivos de Windows para que busquemos el ejecutable del navegador que queremos utilizar. Por defecto el servidor WAMP ingresa explorer en el directorio de Windows, eso significa que usará Internet Explorer de Windows que ese lo tiene casi todo el mundo. Así que lo dejamos como está y pinchamos sobre el botón "Abrir".

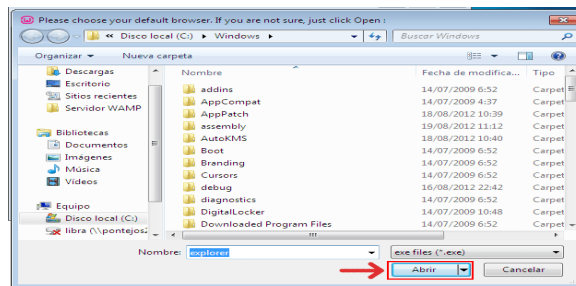


Figura 130: Opciones de instalación.

A continuación nos aconsejará que especifiquemos un servidor SMTP y un Email para utilizar la función mail() que ofrece el lenguaje de programación PHP. Lo dejaremos como está y pinchamos sobre el botón "Next" ya habrá tiempo de cambiar esta característica cuando queramos.

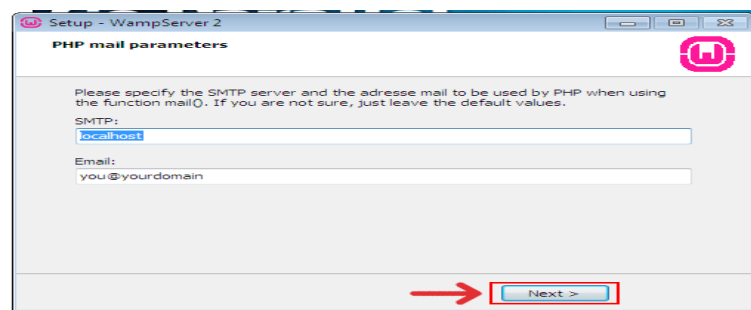


Figura 131: Opciones de instalación.

Por último y para finalizar del todo la instalación nos pregunta si deseamos abrir WampServer 2 Ahora "Launch WampServer2 now". Lo seleccionamos y pinchamos sobre el botón "Finish".

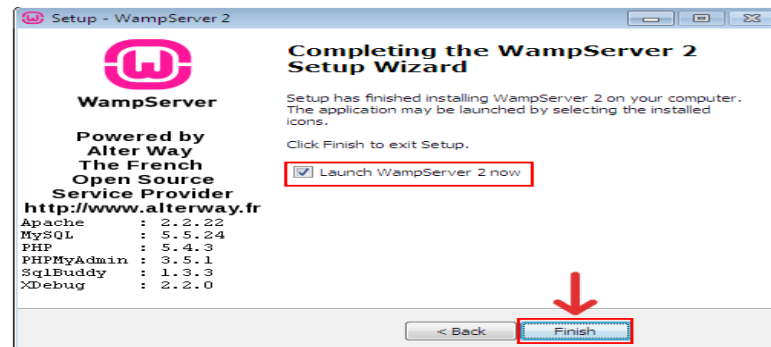


Figura 132: Opciones de instalación.

Como podemos observar, se nos ha abierto un icono en la barra de tareas a la derecha con el siguiente aspecto:

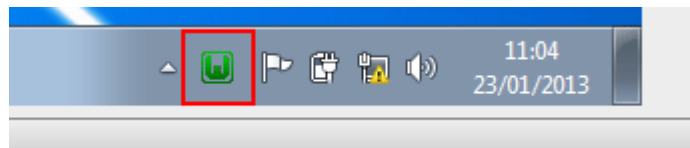


Figura 133: Ver corriendo el servidor.

Ahora vamos a configurarlo en lenguaje Español, para ello pinchamos con el botón derecho del ratón sobre el icono verde que acabamos de mostrar en la barra de tareas a la derecha. Se desplegará un menú en el que deberemos seccionar la opción "Language" y dentro de ésta seleccionaremos "spanish". Ya tendremos nuestro WampServer 2 en español.

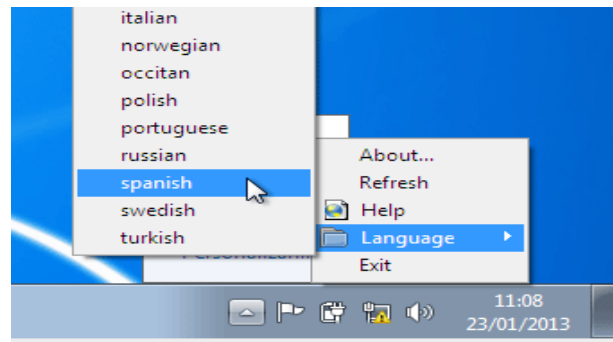


Figura 134: Ver corriendo el servidor.

Lo siguiente que deberemos hacer para que el servidor funcione, es dar en este mismo icono verde con el botón izquierdo del ratón y pinchar sobre la opción "Encender". De esta manera activaremos todos los servicios de WampServer 2.

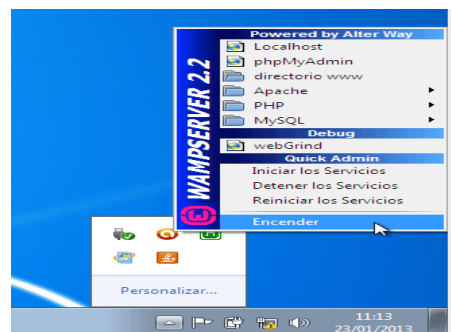


Figura 135: Ver corriendo el servidor.

Ya tenemos instalado y encendido nuestro servidor WAMP para poder hacer nuestros proyectos web. Para comprobar si funciona correctamente abriremos nuestro navegador de internet e ingresaremos en la barra de direcciones "localhost".

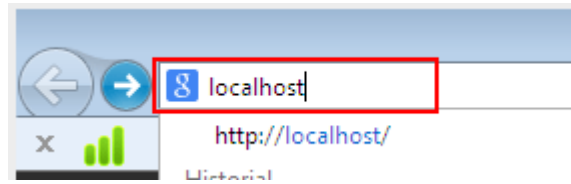


Figura 135: Ver corriendo el servidor.

Si aparece una página web referente a la configuración de nuestro WampServer 2 hemos hecho bien el trabajo y podemos estar seguros de que funciona.

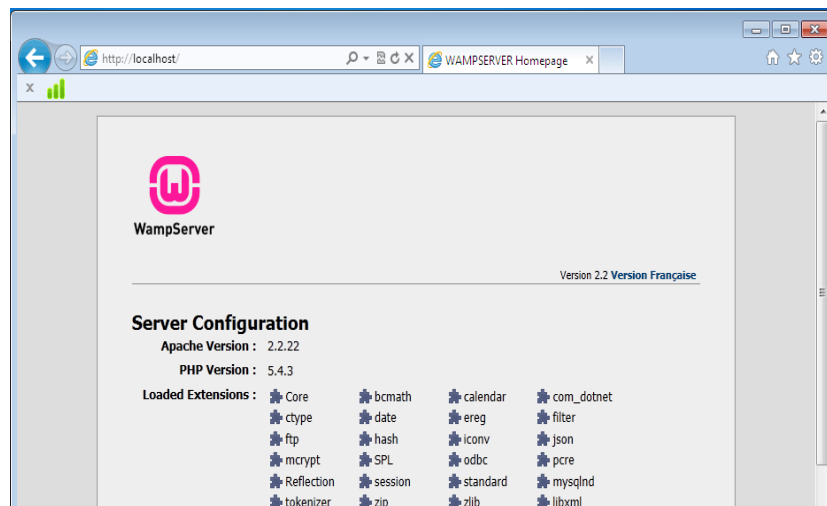


Figura 136: Ver corriendo el servidor.

A.05.c Base de Datos

```
SET FOREIGN_KEY_CHECKS=0;
```

```
-- -----  
-- Table structure for asientos  
-- -----
```

```
DROP TABLE IF EXISTS `asientos`;  
CREATE TABLE `asientos` (  
  `ID_ASIENTOS` int(11) NOT NULL AUTO_INCREMENT,
```



```
`ID_EMPRESA` int(11) NOT NULL,  
`CUEN_ASIENTOS` varchar(50) DEFAULT NULL,  
`DEBE_ASIENTOS` decimal(11,2) DEFAULT NULL,  
`HABE_ASIENTOS` decimal(11,2) DEFAULT NULL,  
`SALD_ASIENTOS` decimal(11,2) DEFAULT NULL,  
`FACT_NUME_ASIENTOS` int(15) DEFAULT NULL,  
`FECH_ASIENTOS` timestamp NULL DEFAULT  
CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,  
PRIMARY KEY (`ID_ASIENTOS`,`ID_EMPRESA`),  
KEY `FK_ASIENTOS_EMPRESA` (`ID_EMPRESA`),  
CONSTRAINT `FK_ASIENTOS_EMPRESA` FOREIGN KEY  
(`ID_EMPRESA`) REFERENCES `empresa` (`ID_EMPRESA`)  
) ENGINE=InnoDB AUTO_INCREMENT=119 DEFAULT  
CHARSET=latin1;
```

```
-- -----  
-- Table structure for auditoria  
-- -----
```

```
DROP TABLE IF EXISTS `auditoria`;  
CREATE TABLE `auditoria` (  
  `FCHA_SALIDA_AUDITORIA` timestamp NULL DEFAULT NULL ON  
UPDATE CURRENT_TIMESTAMP,  
  `ID_EMPRESA` int(11) NOT NULL,  
  `ID_AUDITORIA` int(11) NOT NULL AUTO_INCREMENT,  
  `FCHA_AUDITORIA` timestamp NULL DEFAULT NULL,  
  `IP_AUDITORIA` varchar(15) DEFAULT NULL,  
  `HOST_AUDITORIA` varchar(30) DEFAULT NULL,  
  `USUARIO_AUDITORIA` varchar(30) DEFAULT NULL,  
  `USUARIO_PUERTO` varchar(30) DEFAULT NULL,  
  `USUARIO_NAVEGADOR` varchar(500) DEFAULT NULL,  
  PRIMARY KEY (`ID_AUDITORIA`,`ID_EMPRESA`),  
  KEY `ID_AUDITORIA` (`ID_AUDITORIA`)  
) ENGINE=InnoDB AUTO_INCREMENT=391 DEFAULT  
CHARSET=ascii;
```

```
-- -----  
-- Table structure for cabezera_kardex  
-- -----
```

```
DROP TABLE IF EXISTS `cabezera_kardex`;  
CREATE TABLE `cabezera_kardex` (  
  `ID_EMPRESA` int(11) NOT NULL DEFAULT '0',  
  `ID_CABEZERA_KARDEX` int(11) NOT NULL AUTO_INCREMENT,  
  `ID_PRODUCTO` int(11) DEFAULT NULL,
```




```
`ACTU_PREC_PROD_CABEZERA_kARDEX` decimal(11,2) DEFAULT
NULL,
`ULTI_PREC_PROD_CABEZERA_kARDEX` decimal(11,2) DEFAULT
NULL,
`EXIS_PROD_CABEZERA_kARDEX` decimal(11,0) DEFAULT NULL,
`COSTO_PROD_CABEZERA_kARDEX` decimal(11,2) DEFAULT
NULL,
`PVMAYOR_PROD_CABEZERA_kARDEX` decimal(11,2) DEFAULT
NULL,
`EXISMINIM_PROD_CABEZERA_kARDEX` decimal(11,0) DEFAULT
NULL,
`EST_PROD_CABEZERA_kARDEX` char(1) DEFAULT NULL,
PRIMARY KEY (`ID_CABEZERA_kARDEX`,`ID_EMPRESA`),
KEY `FK_PRODUCTO_KARDEX` (`ID_PRODUCTO`),
KEY `ID_CABEZERA_kARDEX` (`ID_CABEZERA_kARDEX`),
KEY `FK_CABEZERA_KARDEX` (`ID_EMPRESA`),
CONSTRAINT `FK_CABEZERA_KARDEX` FOREIGN KEY
(`ID_EMPRESA`) REFERENCES `producto` (`ID_EMPRESA`),
CONSTRAINT `FK_PRODUCTO_KARDEX` FOREIGN KEY
(`ID_PRODUCTO`) REFERENCES `producto` (`ID_PRODUCTO`)
) ENGINE=InnoDB AUTO_INCREMENT=30 DEFAULT
CHARSET=latin1;
```

-- -----

-- Table structure for cliente_proveedor

-- -----

```
DROP TABLE IF EXISTS `cliente_proveedor`;
CREATE TABLE `cliente_proveedor` (
  `ID_CLIENTE_PROVEDOR` int(11) NOT NULL AUTO_INCREMENT,
  `CED_CLIENTE_CLIENTE_PROVEDOR` varchar(30) DEFAULT
NULL,
  `NOMBRE1_CLIENTE_CLIENTE_PROVEDOR` varchar(30) DEFAULT
NULL,
  `NOMBRE2_CLIENTE_CLIENTE_PROVEDOR` varchar(30) DEFAULT
NULL,
  `APELLIDO1_CLIENTE_CLIENTE_PROVEDOR` varchar(30)
DEFAULT NULL,
  `APELLIDO2_CLIENTE_CLIENTE_PROVEDOR` varchar(30)
DEFAULT NULL,
  `DIRECCION_CLIENTE_CLIENTE_PROVEDOR` varchar(50)
DEFAULT NULL,
  `TELF_CLIENTE_CLIENTE_PROVEDOR` varchar(20) DEFAULT
NULL,
  `EST_CLIENTE_CLIENTE_PROVEDOR` char(1) DEFAULT NULL,
```



```
`TIP_CLIENTE_CLIENTE_PROVEDOR` char(1) DEFAULT NULL,  
`ID_DESCUENTO` int(1) DEFAULT NULL,  
`RUC_CLIENTE_CLIENTE_PROVEDOR` varchar(15) DEFAULT  
NULL,  
`OBLI_CONT_CLIENTE_CLIENTE_PROVEDOR` char(1) DEFAULT  
NULL,  
`RAZO_CLIENTE_CLIENTE_PROVEDOR` varchar(50) DEFAULT  
NULL,  
PRIMARY KEY (`ID_CLIENTE_PROVEDOR`),  
KEY `FK_DESCUENTO` (`ID_DESCUENTO`),  
CONSTRAINT `FK_DESCUENTO` FOREIGN KEY  
(`ID_DESCUENTO`) REFERENCES `descuento` (`ID_DESCUENTO`)  
) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT  
CHARSET=ascii;
```

```
-- -----  
-- Table structure for descuento  
-- -----
```

```
DROP TABLE IF EXISTS `descuento`;  
CREATE TABLE `descuento` (  
  `ID_DESCUENTO` int(11) NOT NULL AUTO_INCREMENT,  
  `DES_DESCUENTO` decimal(8,2) DEFAULT NULL,  
  `EST_DESCUENTO` char(1) DEFAULT NULL,  
  `DSCR_DESCUENTO` varchar(20) DEFAULT NULL,  
  PRIMARY KEY (`ID_DESCUENTO`)  
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT  
CHARSET=latin1;
```

```
-- -----  
-- Table structure for detalle_kardex  
-- -----
```

```
DROP TABLE IF EXISTS `detalle_kardex`;  
CREATE TABLE `detalle_kardex` (  
  `ID_EMPRESA` int(11) NOT NULL DEFAULT '0',  
  `ID_DETALLE_KARDEX` int(11) NOT NULL AUTO_INCREMENT,  
  `ID_CABEZERA_KARDEX` int(11) DEFAULT NULL,  
  `FECH_DETALLE_KARDEX` datetime DEFAULT NULL,  
  `CONP_DETALLE_KARDEX` varchar(50) DEFAULT NULL,  
  `CANENT_DETALLE_KARDEX` decimal(11,0) DEFAULT NULL,  
  `VLUENT_DETALLE_KARDEX` decimal(11,2) DEFAULT NULL,  
  `VLSENT_DETALLE_KARDEX` decimal(11,2) DEFAULT NULL,  
  `CANSAL_DETALLE_KARDEX` decimal(11,0) DEFAULT NULL,  
  `VLUSAL_DETALLE_KARDEX` decimal(11,2) DEFAULT NULL,  
  `VLTSAL_DETALLE_KARDEX` decimal(11,2) DEFAULT NULL,
```



```
`CANSLD_DETALLE_KARDEX` decimal(11,0) DEFAULT NULL,  
`VLUSLD_DETALLE_KARDEX` decimal(11,2) DEFAULT NULL,  
`VLTSLD_DETALLE_KARDEX` decimal(11,2) DEFAULT NULL,  
PRIMARY KEY (`ID_DETALLE_KARDEX`,`ID_EMPRESA`),  
KEY `FK_KARDEX_DETALLE` (`ID_CABEZERA_KARDEX`),  
KEY `ID_EMPRESA` (`ID_EMPRESA`),  
CONSTRAINT `FK_CABESERA_DETALLE` FOREIGN KEY  
(`ID_EMPRESA`) REFERENCES `cabezera_kardex` (`ID_EMPRESA`),  
CONSTRAINT `FK_KARDEX_DETALLE` FOREIGN KEY  
(`ID_CABEZERA_KARDEX`) REFERENCES `cabezera_kardex`  
(`ID_CABEZERA_KARDEX`)  
) ENGINE=InnoDB AUTO_INCREMENT=592 DEFAULT  
CHARSET=ascii;
```

```
-- -----  
-- Table structure for det_factura  
-- -----
```

```
DROP TABLE IF EXISTS `det_factura`;  
CREATE TABLE `det_factura` (  
  `ID_EMPRESA` int(11) NOT NULL,  
  `ID_DETFACT` int(11) NOT NULL AUTO_INCREMENT,  
  `ID_PRODUCTO` int(11) DEFAULT NULL,  
  `ID_FACTURA` int(11) DEFAULT NULL,  
  `REGISTRO_DETFACT` varchar(50) DEFAULT NULL,  
  `CANT_DETFACT` decimal(12,0) DEFAULT NULL,  
  `V_U_DETFACT` decimal(12,2) DEFAULT NULL,  
  `DESCT_DETFACT` decimal(12,2) DEFAULT NULL,  
  `IVA_DETFACT` decimal(12,2) DEFAULT NULL,  
  `VALORNET_DETFACT` decimal(12,2) DEFAULT NULL,  
  `TIPIAGO_DETFACT` varchar(255) DEFAULT NULL,  
  `NUMFACT_DETFACT` varchar(20) DEFAULT NULL,  
  PRIMARY KEY (`ID_DETFACT`,`ID_EMPRESA`),  
  KEY `FK_FACT_DETFA` (`ID_FACTURA`),  
  KEY `FK_PROD_DETFA` (`ID_PRODUCTO`),  
  KEY `FK_ID_EMPRESA` (`ID_EMPRESA`) USING BTREE,  
  CONSTRAINT `FK_FACTURA_PRODUCTO` FOREIGN KEY  
  (`ID_EMPRESA`) REFERENCES `producto` (`ID_EMPRESA`),  
  CONSTRAINT `FK_FACT_DETFA` FOREIGN KEY (`ID_FACTURA`)  
  REFERENCES `factura` (`ID_FACTURA`),  
  CONSTRAINT `FK_PROD_DETFA` FOREIGN KEY  
  (`ID_PRODUCTO`) REFERENCES `producto` (`ID_PRODUCTO`)  
) ENGINE=InnoDB AUTO_INCREMENT=433 DEFAULT  
CHARSET=ascii;
```



-- Table structure for empresa

```
-----  
-- Table structure for empresa  
-----  
DROP TABLE IF EXISTS `empresa`;  
CREATE TABLE `empresa` (  
  `ID_EMPRESA` int(11) NOT NULL AUTO_INCREMENT,  
  `RAZSOC_EMPRESA` varchar(70) DEFAULT NULL,  
  `LOGO_EMPRESA` varchar(200) DEFAULT NULL,  
  `FCHA_EMPRESA` datetime DEFAULT NULL,  
  `NOMCOM_EMPRESA` varchar(50) DEFAULT NULL,  
  `RUC_EMPRESA` char(13) DEFAULT NULL,  
  `DIRECMATRIZ_EMPRESA` varchar(50) DEFAULT NULL,  
  `DIRECESTABL_EMPRESA` varchar(50) DEFAULT NULL,  
  `TEL1_EMPRESA` char(9) DEFAULT NULL,  
  `TEL2_EMPRESA` char(9) DEFAULT NULL,  
  `CEL_EMPRESA` char(10) DEFAULT NULL,  
  `EMAIL_EMPRESA` varchar(20) DEFAULT NULL,  
  `EST_EMPRESA` char(1) DEFAULT NULL,  
  `OBLIGLLEVCONT_EMPRESA` char(1) DEFAULT NULL,  
  PRIMARY KEY (`ID_EMPRESA`)  
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=ascii;
```

-- Table structure for factura

```
-----  
-- Table structure for factura  
-----  
DROP TABLE IF EXISTS `factura`;  
CREATE TABLE `factura` (  
  `ID_EMPRESA` int(11) NOT NULL,  
  `ID_FACTURA` int(11) NOT NULL AUTO_INCREMENT,  
  `ID_CLIENTE_PROVEDOR` int(11) DEFAULT NULL,  
  `NUM_FACTURA` char(15) DEFAULT NULL,  
  `FECHA_FACTURA` timestamp NOT NULL DEFAULT  
CURRENT_TIMESTAMP,  
  `EST_FACTURA` char(1) DEFAULT NULL,  
  `VT_FACTURA` decimal(16,2) DEFAULT NULL,  
  `USU_FACTURA` varchar(50) DEFAULT NULL,  
  `CAJERA_FACTURA` varchar(255) DEFAULT NULL,  
  `TIPO_FAC_FACTURA` char(1) DEFAULT NULL,  
  `TIPO_NOTA_CRED_FACTURA` char(1) DEFAULT NULL,  
  `CANT_NOTA_CRED_FACTURA` decimal(11,2) DEFAULT NULL,  
  `TOTAL_RET_FACTURA` decimal(11,2) DEFAULT '0.00',  
  `NETO_FACTURA` decimal(11,2) DEFAULT NULL,  
  `DESC_FACTURA` decimal(11,2) DEFAULT NULL,  
  `CONT_CRED_FACTURA` varchar(50) DEFAULT NULL,
```



```
`FECH_VENC_FACTURA` timestamp NULL DEFAULT NULL ON
UPDATE CURRENT_TIMESTAMP,
`HIST_NOTA_CRED_FACTURA` decimal(11,2) DEFAULT NULL,
`FACT_NOTA_CRED_FACTURA` int(14) DEFAULT NULL,
PRIMARY KEY (`ID_FACTURA`,`ID_EMPRESA`),
KEY `FK_CLI_FACT` (`ID_CLIENTE_PROVEDOR`),
KEY `FK_REFERENCE_7` (`ID_EMPRESA`),
KEY `ID_FACTURA` (`ID_FACTURA`),
CONSTRAINT `FK_CLI_FACT` FOREIGN KEY
(`ID_CLIENTE_PROVEDOR`) REFERENCES `cliente_provedor`
(`ID_CLIENTE_PROVEDOR`),
CONSTRAINT `FK_FACTURA_DETALLE` FOREIGN KEY
(`ID_EMPRESA`) REFERENCES `det_factura` (`ID_EMPRESA`)
) ENGINE=InnoDB AUTO_INCREMENT=345 DEFAULT
CHARSET=ascii;
```

```
-- -----
-- Table structure for marca_producto
-- -----
```

```
DROP TABLE IF EXISTS `marca_producto`;
CREATE TABLE `marca_producto` (
  `ID_EMPRESA` int(11) NOT NULL DEFAULT '0',
  `ID_MARCA_PRODUCTO` int(11) NOT NULL AUTO_INCREMENT,
  `DESC_MARCA_PRODUCTO` varchar(50) DEFAULT NULL,
  `ESTA_MARCA_PRODUCTO` char(1) DEFAULT NULL,
  PRIMARY KEY (`ID_MARCA_PRODUCTO`,`ID_EMPRESA`),
  KEY `ID_MARCA_PRODUCTO` (`ID_MARCA_PRODUCTO`),
  KEY `ID_EMPRESA` (`ID_EMPRESA`),
  CONSTRAINT `FK_MARCA_EMPRESA` FOREIGN KEY
(`ID_EMPRESA`) REFERENCES `producto` (`ID_EMPRESA`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT
CHARSET=latin1;
```

```
-- -----
-- Table structure for pedido
-- -----
```

```
DROP TABLE IF EXISTS `pedido`;
CREATE TABLE `pedido` (
  `ID_EMPRESA` int(11) NOT NULL DEFAULT '0',
  `ID_PEDIDO` int(11) NOT NULL AUTO_INCREMENT,
  `ID_PRODUCTO` int(11) DEFAULT NULL,
  `CANT_PEDIDO` decimal(11,4) DEFAULT NULL,
  `EST_PEDIDO` char(1) DEFAULT NULL,
  `ID_CLIENTE_PROVEDOR` int(11) DEFAULT NULL,
```



```
PRIMARY KEY (`ID_PEDIDO`,`ID_EMPRESA`),
KEY `FK_PRODUCTO_PEDIDO` (`ID_PRODUCTO`),
KEY `FK_PROVEDOR_PEDIDO` (`ID_CLIENTE_PROVEDOR`),
KEY `FK_PEDIDO_EMPRESA` (`ID_EMPRESA`),
CONSTRAINT `FK_PEDIDO_EMPRESA` FOREIGN KEY
(`ID_EMPRESA`) REFERENCES `producto` (`ID_EMPRESA`),
CONSTRAINT `FK_PRODUCTO_PEDIDO` FOREIGN KEY
(`ID_PRODUCTO`) REFERENCES `producto` (`ID_PRODUCTO`),
CONSTRAINT `FK_PROVEDOR_PEDIDO` FOREIGN KEY
(`ID_CLIENTE_PROVEDOR`) REFERENCES `cliente_proveedor`
(`ID_CLIENTE_PROVEDOR`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

-- -----

-- Table structure for producto

-- -----

```
DROP TABLE IF EXISTS `producto`;
CREATE TABLE `producto` (
  `ID_EMPRESA` int(11) NOT NULL,
  `ID_PRODUCTO` int(11) NOT NULL AUTO_INCREMENT,
  `ID_MARCA_PRODUCTO` int(11) DEFAULT NULL,
  `DESC_PRODUCTO` varchar(100) DEFAULT NULL,
  `FECH_PRODUCTO` timestamp NULL DEFAULT NULL,
  `EST_PRODUCTO` char(1) DEFAULT NULL,
  `IMG_PRODUCTO` varchar(100) DEFAULT NULL,
  `REFE_PRODUCTO` varchar(50) DEFAULT NULL,
  `EXFIS_PRODUCTO` char(1) DEFAULT NULL,
  `CANT_MAYO_PRODUCTO` decimal(11,0) DEFAULT NULL,
  `PVPU_MAYO_PRODUCTO` decimal(11,2) DEFAULT NULL,
  PRIMARY KEY (`ID_PRODUCTO`,`ID_EMPRESA`),
  UNIQUE KEY `DESC_PRODUCTO` (`DESC_PRODUCTO`),
  KEY `FK_ID_EMPRESA_CATEGORIA` (`ID_EMPRESA`) USING
BTREE,
  KEY `ID_PRODUCTO` (`ID_PRODUCTO`),
  KEY `FK_MARCAPRO` (`ID_MARCA_PRODUCTO`),
  CONSTRAINT `FK_EMPRESA_PRODUCTO` FOREIGN KEY
(`ID_EMPRESA`) REFERENCES `empresa` (`ID_EMPRESA`),
  CONSTRAINT `FK_MARCAPRO` FOREIGN KEY
(`ID_MARCA_PRODUCTO`) REFERENCES `marca_producto`
(`ID_MARCA_PRODUCTO`)
) ENGINE=InnoDB AUTO_INCREMENT=36 DEFAULT
CHARSET=ascii;
```

-- -----



-- Table structure for retencion

```
-----  
DROP TABLE IF EXISTS `retencion`;  
CREATE TABLE `retencion` (  
  `ID_EMPRESA` int(11) NOT NULL DEFAULT '0',  
  `ID_RETENCION` int(11) NOT NULL AUTO_INCREMENT,  
  `ID_FACTURA` int(11) DEFAULT NULL,  
  `NUME_RETENCION` char(15) DEFAULT NULL,  
  `FECH_RETENCION` timestamp NULL DEFAULT NULL ON UPDATE  
CURRENT_TIMESTAMP,  
  `EFIS_RETENCION` date DEFAULT NULL,  
  `BINP_RETENCION` decimal(2,0) DEFAULT NULL,  
  `INPU_RETENCION` varchar(50) DEFAULT NULL,  
  `CODI_RETENCION` varchar(10) DEFAULT NULL,  
  `PORC_RETENCION` decimal(2,0) DEFAULT NULL,  
  `VALR_RETENCION` decimal(2,0) DEFAULT NULL,  
  `TIPO_RETEMCIION` char(1) DEFAULT NULL,  
  `ESTA_RETENCION` char(1) DEFAULT NULL,  
  PRIMARY KEY (`ID_RETENCION`,`ID_EMPRESA`),  
  KEY `FK_RETENCION_FACTURA` (`ID_FACTURA`),  
  KEY `FK_EMPRESA_RETENCIO` (`ID_EMPRESA`),  
  CONSTRAINT `FK_EMPRESA_RETENCIO` FOREIGN KEY  
(`ID_EMPRESA`) REFERENCES `factura` (`ID_EMPRESA`),  
  CONSTRAINT `FK_RETENCION_FACTURA` FOREIGN KEY  
(`ID_FACTURA`) REFERENCES `factura` (`ID_FACTURA`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

-- Table structure for rol_usuario

```
-----  
DROP TABLE IF EXISTS `rol_usuario`;  
CREATE TABLE `rol_usuario` (  
  `ID_EMPRESA` int(11) NOT NULL,  
  `ID_ROL` int(11) NOT NULL AUTO_INCREMENT,  
  `DESC_ROL` varchar(30) DEFAULT NULL,  
  `EST_ROL` char(1) DEFAULT NULL,  
  PRIMARY KEY (`ID_ROL`,`ID_EMPRESA`),  
  KEY `ID_ROL` (`ID_ROL`),  
  KEY `ID_EMPRESA` (`ID_EMPRESA`)  
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=ascii;
```

-- Table structure for temp_factura

```
DROP TABLE IF EXISTS `temp_factura`;
CREATE TABLE `temp_factura` (
  `ID_EMPRESA` int(11) NOT NULL,
  `ID_TEMPFACT` int(255) NOT NULL AUTO_INCREMENT,
  `COD_TEMPFACT` varchar(255) CHARACTER SET latin1 NOT NULL,
  `NOMBRE_TEMPFACT` varchar(255) CHARACTER SET latin1 NOT
NULL,
  `VENTA_TEMPFACT` varchar(255) CHARACTER SET latin1 NOT
NULL,
  `CANTIDAD_TEMPFACT` varchar(255) CHARACTER SET latin1 NOT
NULL,
  `IMPORTE_TEMPFACT` varchar(255) CHARACTER SET latin1 NOT
NULL,
  `EXISTENCIA_TEMPFACT` varchar(255) CHARACTER SET latin1
NOT NULL,
  `USUARIO_TEMPFACT` varchar(255) CHARACTER SET latin1 NOT
NULL,
  `ID_CLIENTE` int(11) DEFAULT NULL,
  `REFE_TEMPFACT` varchar(50) DEFAULT NULL,
  `TIPO_TEMPFACT` char(1) DEFAULT NULL,
  PRIMARY KEY (`ID_TEMPFACT`,`ID_EMPRESA`),
  KEY `ID_EMPRESA` (`ID_EMPRESA`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=ascii;
```

```
-- -----
-- Table structure for usuario
-- -----
```

```
DROP TABLE IF EXISTS `usuario`;
CREATE TABLE `usuario` (
  `ID_EMPRESA` int(11) NOT NULL,
  `ID_USUARIO` int(11) NOT NULL AUTO_INCREMENT,
  `ID_ROL` int(11) DEFAULT NULL,
  `NOMBRE_USUARIO` varchar(30) DEFAULT NULL,
  `EST_USUARIO` char(1) DEFAULT NULL,
  `PASS_USUARIO` varchar(20) DEFAULT NULL,
  `NOM_USUARIO` varchar(50) DEFAULT NULL,
  `APE_USUARIO` varchar(50) DEFAULT NULL,
  `APE2_USUARIO` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`ID_USUARIO`,`ID_EMPRESA`),
  KEY `FK_USU_ROL` (`ID_ROL`),
  KEY `FK_ID_USURIOROL` (`ID_EMPRESA`) USING BTREE,
  CONSTRAINT `FK_USU_EMPRESA` FOREIGN KEY
(`ID_EMPRESA`) REFERENCES `empresa` (`ID_EMPRESA`),
```




```

CONSTRAINT `FK_USU_ROL` FOREIGN KEY (`ID_ROL`)
REFERENCES `rol_usuario` (`ID_ROL`)
) ENGINE=InnoDB AUTO_INCREMENT=15 DEFAULT
CHARSET=ascii;

```

```

-----

```

```

-- Table structure for usuario_temporal

```

```

-----

```

```

DROP TABLE IF EXISTS `usuario_temporal`;
CREATE TABLE `usuario_temporal` (
  `ID_AUDITORIA` int(11) DEFAULT NULL,
  `ID_EMPRESA` int(11) NOT NULL,
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `des` varchar(20) DEFAULT NULL,
  `fech` timestamp NULL DEFAULT NULL ON UPDATE
CURRENT_TIMESTAMP,
  `accion` varchar(100) DEFAULT NULL,
  `usu` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`id`,`ID_EMPRESA`),
  KEY `FK_AUDITORIA` (`ID_AUDITORIA`),
  CONSTRAINT `FK_AUDITORIA` FOREIGN KEY (`ID_AUDITORIA`)
REFERENCES `auditoria` (`ID_AUDITORIA`)
) ENGINE=InnoDB AUTO_INCREMENT=799 DEFAULT CHARSET=latin1;

```

A.05.d Código Fuente

```

<?php
    session_start();
    include('Connections/php_conexion.php');
    $usuario=$_SESSION['username'];
    $compania=$_SESSION['compania'];
    if(!$_SESSION['sesion']=='si'){
        header('location:error.php');
    }

    mysql_select_db($database_php_conexion, $php_conexion);

    $scanidfact=mysql_query("SELECT MAX(ID_FACTURA) as maximo
    FROM factura WHERE TIPO_FAC_FACTURA='F' ",
    $php_conexion);//codigo de la factura
    if($datoidfact=mysql_fetch_array($scanidfact))
    {
        $id_factura_cliente=$datoidfact['maximo'];
    }

```



```
$scanidfact=mysql_query("SELECT NUM_FACTURA as maximo FROM
factura WHERE TIPO_NOTA_CRED_FACTURA='F' AND
TIPO_FAC_FACTURA='F' AND ID_FACTURA='$id_factura_cliente'",
$php_conexion);//codigo de la factura
if($datonumfact=mysql_fetch_array($scanidfact))
{
$StrNumeroFactura=$datonumfact['maximo'];
$StrCadenaFactura= substr($StrNumeroFactura, -13);
if($StrCadenaFactura<=1001000000000)
{
$numero_factura_cliente='001001000000001';
}
else
{

$numero_factura_cliente='00'.($StrCadenaFactura+1);
}
$cfactura1=$datoidfact['maximo'];
}

mysql_select_db($database_php_conexion, $php_conexion);
$scanfactid=mysql_query("SELECT MAX(ID_FACTURA) as maxidfact
FROM factura ", $php_conexion);//codigo de la factura
if($datofactid=mysql_fetch_array($scanfactid))
{
$cfacturamaxid=$datofactid['maxidfact']+1;
}

if($numero_factura_cliente==1)
{
$numero_factura_cliente=1000;
} //si es primera factura colocar que
empieze en 1000

$hoy=$fechay=date("Y-m-d");

if($_GET['button']=='Cobrar Dinero Recibido')
{ //contado

if(isset( $_GET['notacredito']))
{

$clientes=$_GET['cliente'];
```

```
$t_importe=0;

$descuento_producto=$_GET['descuento_pro'];

mysql_select_db($database_php_conexion,
$php_conexion);

$scan=mysql_query("SELECT
sum(factura.CANT_NOTA_CRED_FACTURA)

FROM factura

WHERE
factura.ID_EMPRESA='$compania'
factura.TIPO_NOTA_CRED_FACTURA='N' AND

factura.ID_CLIENTE_PROVEDOR='$clientes' ",
$php_conexion);

if($dato=mysql_fetch_array($scan))
{

$notadecredito=$dato['sum(factura.CANT_NOTA_CRED_FACTURA)'];

}

$totccpagar=$_GET['ccpago'];

$subtotal=$_GET['subtota'];
```

```
$totpagar=$subtotal+($subtotal*0.12);
```

```
if($totpagar>$notadecredito)
```

```
{
```

```
    $descuento_factura_neto=0;
```

```
    $subtotal=$_GET['subtota'];
```

```
    $ccpago=$_GET['ccpago'];
```

```
    $tpagar= $totpagar- $notadecredito;
```

```
    $cliente=$_GET['cliente'];
```

```
    if($_GET['descuento']=='descuento')
```

```
{

    $descuento_factura_neto=($_GET['subtota']*$descuen
to_producto)/100;

    $tpagar=$_GET['descuento_pagar']-
$notadecredito;

    $subtotal=$_GET['subtota']-
(($_GET['subtota']*$descuento_producto)/100);

}

$sql1="Update          factura          Set
TIPO_NOTA_CRED_FACTURA='C' Where ID_EMPRESA='$compania'
AND          ID_CLIENTE_PROVEDOR='$cliente'          AND
TIPO_NOTA_CRED_FACTURA='N' ";

mysql_query($sql1);

}
```

else

{

///si deseo el pago con nota de credito en sero

cambiar aqi

if(\$_GET['descuento']=='descuento')

{

\$descuento_factura_neto=(\$_GET['subtota']*\$descuen
to_producto)/100;

\$subtotal=\$_GET['subtota']-
((\$_GET['subtota']*\$descuento_producto)/100);

\$cc1pago=\$_GET['ccpago'];

\$totalpagar=\$subtotal+(\$subtotal*0.12);

\$cliente=\$_GET['cliente'];

\$total=\$notadecredito- \$totalpagar;

\$ccpago=0;

\$tpagar=0;

```
}

else

{

    $descuento_factura_neto=0;

    $subtotal=$_GET['subtotal'];

    $cc1pago=$_GET['ccpago'];

    $totalpagar=$_GET['tpagar'];

    $cliente=$_GET['cliente'];

    $total=$notadecredito-$totalpagar;

    $ccpago=0;

    $tpagar=0;

}

$sql1="Update          factura          Set
TIPO_NOTA_CRED_FACTURA='C' Where ID_EMPRESA='$compania'
AND          ID_CLIENTE_PROVEDOR='$cliente'          AND
TIPO_NOTA_CRED_FACTURA='N' ";

mysql_query($sql1);
```



```
$ca=mysql_query("SELECT max(ID_FACTURA) AS  
ID FROM factura where ID_EMPRESA='$compania' AND  
ID_CLIENTE_PROVEDOR='$cliente' AND  
TIPO_NOTA_CRED_FACTURA='C' ");
```

```
if($date=mysql_fetch_array($ca))  
{  
    $id=$date['ID'];  
}
```

```
$sql="Update factura Set  
TIPO_NOTA_CRED_FACTURA='N',CANT_NOTA_CRED_FACTURA='  
$total' Where ID_EMPRESA='$compania' AND  
ID_CLIENTE_PROVEDOR='$cliente' AND ID_FACTURA='$id' ";  
mysql_query($sql);  
}
```

```
}  
  
else  
{
```

```
if($_GET['descuento']=='descuento')
```

```
{
```



```
$descuento_producto=$_GET['descuento_pro'];

$descuento_factura_neto=(($_GET['subtotal']*$descuento_producto)/100);
echo $subtotal=$_GET['subtotal']-
((($_GET['subtotal']*$descuento_producto)/100);
$ccpago=$_GET['ccpago'];
echo
$tpagar=$subtotal+($subtotal*0.12);
$cliente=$_GET['cliente'];
$t_importe=0;

}

else

$descuento_factura_neto=0;

$subtotal=$_GET['subtotal'];

{

$descuento_producto=$_GET['descuento_pro'];
$ccpago=$_GET['ccpago'];
$tpagar=$subtotal+($subtotal*0.12);

$cliente=$_GET['cliente'];
$t_importe=0;
```

```

    }

    }

    //guarda tabla factura
    if($_GET['checkbox_credito']=='si')
    {

        $contado_credito='CREDITO';

        $fecha_vencimiento=$_GET['txt_fecha_ven'];
        $cxc_sql="INSERT
        INTO asientos (ID_EMPRESA, CUEN_ASIENTOS, DEBE_ASIENTOS,
        HABE_ASIENTOS, SALD_ASIENTOS,
        FACT_NUME_ASIENTOS,FECH_ASIENTOS) VALUES ('$compania',
        'CUENTAS POR COBRAR', '$subtotal', '0', '$subtotal',
        '$numero_factura_cliente',NOW())

        ";
        mysql_query($cxc_sql);

        mysql_select_db($database_php_conexion,
        $php_conexion);
        $query_auditoria = "SELECT max(auditoria.ID_AUDITORIA) FROM
        auditoria WHERE auditoria.ID_EMPRESA='$compania' AND
        auditoria.USUARIO_AUDITORIA='$usuario'";
        $auditoria = mysql_query($query_auditoria, $php_conexion) or
        die(mysql_error());
        $row_auditoria = mysql_fetch_assoc($auditoria);
        $totalRows_auditoria = mysql_num_rows($auditoria);
        $id= $row_auditoria['max(auditoria.ID_AUDITORIA)'];

        $insertSQL1 = sprintf("INSERT INTO usuario_temporal
        (ID_EMPRESA,fecha,accion,usu,ID_AUDITORIA) VALUES
        ('.$compania.',NOW(),'INSERTAR CUENTAS POR
        COBRAR','$usuario','$id')");

        mysql_select_db($database_php_conexion,
        $php_conexion);
    
```



```
$Result2 = mysql_query($insertSQL1, $php_conexion) or
die(mysql_error());
```

```
$ventas=$subtotal+($subtotal*0.12);
```

```
$vent_sql="INSERT
INTO asientos (ID_EMPRESA, CUEN_ASIENTOS, DEBE_ASIENTOS,
HABE_ASIENTOS, SALD_ASIENTOS,
FACT_NUME_ASIENTOS,FECH_ASIENTOS) VALUES ('$compania',
'VENTAS', '$ventas', '0', '$ventas', '$numero_factura_cliente',NOW())
";
```

```
mysql_query($vent_sql);
```

```
$query_auditoria = "SELECT
max(auditoria.ID_AUDITORIA) FROM auditoria WHERE
auditoria.ID_EMPRESA='$compania' AND
auditoria.USUARIO_AUDITORIA='$usuario'";
$auditoria = mysql_query($query_auditoria, $php_conexion) or
die(mysql_error());
$row_auditoria = mysql_fetch_assoc($auditoria);
$totalRows_auditoria = mysql_num_rows($auditoria);
$id= $row_auditoria['max(auditoria.ID_AUDITORIA)'];
```

```
$insertSQL1 = sprintf("INSERT INTO usuario_temporal
(ID_EMPRESA, fech, accion, usu, ID_AUDITORIA) VALUES
('".$compania."',NOW(), 'INSERTAR CUENTA
VENTAS', '$usuario', '$id')");
```

```
mysql_select_db($database_php_conexion,
$php_conexion);
$Result2 = mysql_query($insertSQL1, $php_conexion) or
die(mysql_error());
```

```
$iva_en_ventas=$subtotal*0.12;
```

```
$iva_sql="INSERT INTO
asientos (ID_EMPRESA, CUEN_ASIENTOS, DEBE_ASIENTOS,
HABE_ASIENTOS, SALD_ASIENTOS,
FACT_NUME_ASIENTOS,FECH_ASIENTOS) VALUES ('$compania',
'TVA EN VENTAS', '$iva_en_ventas', '0', '$iva_en_ventas',
'$numero_factura_cliente',NOW())
";
```

```

mysql_query($iva_sql);

$query_auditoria = "SELECT
max(auditoria.ID_AUDITORIA) FROM auditoria WHERE
auditoria.ID_EMPRESA='$compania' AND
auditoria.USUARIO_AUDITORIA='$usuario'";
$auditoria = mysql_query($query_auditoria, $php_conexion) or
die(mysql_error());
$row_auditoria = mysql_fetch_assoc($auditoria);
$totalRows_auditoria = mysql_num_rows($auditoria);
$id= $row_auditoria['max(auditoria.ID_AUDITORIA)'];

$insertSQL1 = sprintf("INSERT INTO usuario_temporal
(ID_EMPRESA,fecha,accion,usu,ID_AUDITORIA) VALUES
('.$compania.',NOW(),'INSERTAR IVA POR COBRAR','$usuario','$id')");
mysql_select_db($database_php_conexion,
$php_conexion);
$result2 = mysql_query($insertSQL1, $php_conexion) or
die(mysql_error());

}
else
{
$contado_credito='CONTADO';
$fecha_vencimiento=date("Y-m-
d");

$cxc_sql="INSERT INTO
asientos (ID_EMPRESA, CUEN_ASIENTOS, DEBE_ASIENTOS,
HABE_ASIENTOS, SALD_ASIENTOS, FACT_NUME_ASIENTOS)
VALUES ('$compania', 'CUENTAS POR COBRAR', '$subtotal', '$subtotal',
'0', '$numero_factura_cliente')
";
mysql_query($cxc_sql);

$query_auditoria =
"SELECT max(auditoria.ID_AUDITORIA) FROM auditoria WHERE

```



```

auditoria.ID_EMPRESA='$compania'
auditoria.USUARIO_AUDITORIA='$usuario';
$auditoria = mysql_query($query_auditoria, $php_conexion) or
die(mysql_error());
$row_auditoria = mysql_fetch_assoc($auditoria);
$totalRows_auditoria = mysql_num_rows($auditoria);
$id= $row_auditoria['max(auditoria.ID_AUDITORIA)'];

$insertSQL1 = sprintf("INSERT INTO usuario_temporal
(ID_EMPRESA,fecha,accion,usu,ID_AUDITORIA)
('$compania.',NOW(),',INSERTAR CUENTA
VENTAS','$usuario','$id')");

mysql_select_db($database_php_conexion,
$php_conexion);
$result2 = mysql_query($insertSQL1, $php_conexion) or
die(mysql_error());
$ventas=$subtotal+($subtotal*0.12);

$vent_sql="INSERT
INTO asientos (ID_EMPRESA, CUEN_ASIENTOS, DEBE_ASIENTOS,
HABE_ASIENTOS, SALD_ASIENTOS, FACT_NUME_ASIENTOS)
VALUES ('$compania', 'VENTAS', '$ventas', '$ventas', '0',
'$numero_factura_cliente')
";

mysql_query($vent_sql);

```

```

$query_auditoria = "SELECT
max(auditoria.ID_AUDITORIA) FROM auditoria WHERE
auditoria.ID_EMPRESA='$compania' AND
auditoria.USUARIO_AUDITORIA='$usuario';
$auditoria = mysql_query($query_auditoria, $php_conexion) or
die(mysql_error());
$row_auditoria = mysql_fetch_assoc($auditoria);
$totalRows_auditoria = mysql_num_rows($auditoria);
$id= $row_auditoria['max(auditoria.ID_AUDITORIA)'];

```



```

$insertSQL1 = sprintf("INSERT INTO usuario_temporal
(ID_EMPRESA, fech, accion, usu, ID_AUDITORIA)
VALUES
('".$compania."', NOW(), '
INSERTAR
CUENTA
VENTAS', '$usuario', '$id')");

```

```

mysql_select_db($database_php_conexion,
$php_conexion);
$result2 = mysql_query($insertSQL1, $php_conexion) or
die(mysql_error());

```

```

$iva_en_ventas = $subtotal * 0.12;

```

```

$iva_sql = "INSERT INTO
asientos (ID_EMPRESA, CUEN_ASIENTOS, DEBE_ASIENTOS,
HABER_ASIENTOS, SALD_ASIENTOS, FACT_NUME_ASIENTOS)
VALUES ('$compania', 'IVA EN VENTAS', '$iva_en_ventas',
'$iva_en_ventas', '0', '$numero_factura_cliente')
";

```

```

mysql_query($iva_sql);

```

```

$query_auditoria = "SELECT
max(auditoria.ID_AUDITORIA) FROM auditoria WHERE
auditoria.ID_EMPRESA = '$compania' AND
auditoria.USUARIO_AUDITORIA = '$usuario'";
$auditoria = mysql_query($query_auditoria, $php_conexion) or
die(mysql_error());
$row_auditoria = mysql_fetch_assoc($auditoria);
$totalRows_auditoria = mysql_num_rows($auditoria);
$id = $row_auditoria['max(auditoria.ID_AUDITORIA)'];

```

```

$insertSQL1 = sprintf("INSERT INTO usuario_temporal
(ID_EMPRESA, fech, accion, usu, ID_AUDITORIA)
VALUES
('".$compania."', NOW(), 'INSERTAR IVA POR COBRAR', '$usuario', '$id')");
mysql_select_db($database_php_conexion,
$php_conexion);
$result2 = mysql_query($insertSQL1, $php_conexion) or
die(mysql_error());

```

```

}

```

```

        if($_GET['fecha']==1)
        {
            $fecha1=$_GET['FCH'].'
'. '00.00.00';

            $factura_sql="INSERT INTO
factura (ID_EMPRESA, ID_CLIENTE_PROVEDOR, NUM_FACTURA,
FECHA_FACTURA, EST_FACTURA, VT_FACTURA,
USU_FACTURA,CAJERA_FACTURA,TIPO_NOTA_CRED_FACTURA,
NETO_FACTURA,TIPO_FAC_FACTURA,DESC_FACTURA,

CONT_CRED_FACTURA,FECH_VENC_FACTURA)

VALUES

('$compania','$cliente','$numero_factura_cliente','$fecha1','A','$tpagar','$susua
rio','$usuario','F','$subtotal','F','$descuento_factura_neto',

'$contado_credito','$fecha_vencimiento')";

mysql_query($factura_sql);

$query_auditoria = "SELECT
max(auditoria.ID_AUDITORIA) FROM auditoria WHERE
auditoria.ID_EMPRESA='$compania' AND
auditoria.USUARIO_AUDITORIA='$usuario'";
$auditoria = mysql_query($query_auditoria, $php_conexion) or
die(mysql_error());
$row_auditoria = mysql_fetch_assoc($auditoria);
$totalRows_auditoria = mysql_num_rows($auditoria);
$id= $row_auditoria['max(auditoria.ID_AUDITORIA)'];

```



```
$insertSQL1 = sprintf("INSERT INTO usuario_temporal
(ID_EMPRESA,fech,accion,usu,ID_AUDITORIA) VALUES
('".$compania."',NOW(),'INSERTAR FACTURA CON
FECHA','$usuario','$id')");
```

```
mysql_select_db($database_php_conexion,
$php_conexion);
$Result2 = mysql_query($insertSQL1, $php_conexion) or
die(mysql_error());
```

```
}
else
{
```

```
$factura_sql="INSERT INTO
factura (ID_EMPRESA, ID_CLIENTE_PROVEDOR, NUM_FACTURA,
FECHA_FACTURA, EST_FACTURA, VT_FACTURA,
USU_FACTURA,CAJERA_FACTURA,TIPO_NOTA_CRED_FACTURA,
NETO_FACTURA,TIPO_FAC_FACTURA,DESC_FACTURA,CONT_CR
ED_FACTURA,
FECH_VENC_FACTURA
)
```

```
VALUES
('$compania','$cliente','$numero_factura_cliente',NOW(),'A','$tpagar','$usuari
o','$usuario','F','$subtotal','F','$descuento_factura_neto','$contado_credito','$f
echa_vencimiento')";
```

```
mysql_query($factura_sql);
```

```
$query_auditoria = "SELECT
max(auditoria.ID_AUDITORIA) FROM auditoria WHERE
auditoria.ID_EMPRESA='$compania' AND
auditoria.USUARIO_AUDITORIA='$usuario'";
$auditoria = mysql_query($query_auditoria, $php_conexion) or
die(mysql_error());
$row_auditoria = mysql_fetch_assoc($auditoria);
$totalRows_auditoria = mysql_num_rows($auditoria);
$id= $row_auditoria['max(auditoria.ID_AUDITORIA)'];
```

```
$insertSQL1 = sprintf("INSERT INTO usuario_temporal
(ID_EMPRESA,fech,accion,usu,ID_AUDITORIA) VALUES
('".$compania."',NOW(),'INSERTAR FACTURA','$usuario','$id')");
```



```

mysql_select_db($database_php_conexion,
$php_conexion);
$Result2 = mysql_query($insertSQL1, $php_conexion) or
die(mysql_error());
    }

```

detalles //codigo de la factura / guarda en

```

mysql_select_db($database_php_conexion,
$php_conexion);

$scan=mysql_query("SELECT * FROM temp_factura
where ID_EMPRESA='$compania' AND
USUARIO_TEMPFACT='$usuario'", $php_conexion);

```

```
$cantotal=0;
```

```
while($dato=mysql_fetch_array($scan))
```

```
{
```

```

$scanDfactid=mysql_query("SELECT
MAX(ID_DETFACT) as maxidDfact FROM det_factura", $php_conexion);

```

//codigo de la factura

```
if($datoDfactid=mysql_fetch_array($scanDfactid))
```

```
{
```

```
$cfacDturamaxid=$datoDfactid['maxidDfact']+1;
```

```
    }

    //obtiene secuencial de
    kardex

    mysql_select_db($database_php_conexion, $php_conexion);

    $kardexseq =
    mysql_query("SELECT MAX(ID_CABEZERA_KARDEX) as maxikardex
    FROM cabeza_kardex", $php_conexion);

    if($datokardex =
    mysql_fetch_array($kardexseq))
    {
        $secidkardex =
        $datokardex['maxikardex']+1;
    }

    $cod=$dato['COD_TEMPFACT'];//codigo de producto
    en temporal de la factura

    $nom=$dato['NOMBRE_TEMPFACT'];

    $cant=$dato['CANTIDAD_TEMPFACT'];

    $cantotal=$dato['CANTIDAD_TEMPFACT']+$cantot
    al;

    $venta=$dato['VENTA_TEMPFACT'];

    $importe=$dato['IMPORTE_TEMPFACT'];

    $t_importe=$t_importe+$importe;
```

```

mysql_select_db($database_php_conexion,
$php_conexion);
$PRODUCTO=mysql_query("SELECT producto.ID_PRODUCTO
FROM producto
WHERE producto.ID_EMPRESA='$compania' AND
producto.DESCR_PRODUCTO='$nom' ", $php_conexion);
if($datosPRODUCTO=mysql_fetch_array($PRODUCTO))
{

echo $id_pro =
$datosPRODUCTO['ID_PRODUCTO'];
}

mysql_select_db($database_php_conexion, $php_conexion);
$PRODUCTO1=mysql_query("SELECT
cabezera_kardex.ID_CABEZERA_KARDEX
FROM
cabezera_kardex
WHERE
cabezera_kardex.ID_PRODUCTO = '$id_pro' AND
cabezera_kardex.ID_EMPRESA = '$compania'", $php_conexion);
if($datosPRODUCTO1=mysql_fetch_array($PRODUCTO1))
{

echo $id_kar =
$datosPRODUCTO1['ID_CABEZERA_KARDEX'];
}

$dettalle_sql="INSERT INTO det_factura (ID_EMPRESA,ID_DETFACT,
ID_PRODUCTO,ID_FACTURA, REGISTRO_DETFACT,
CANT_DETFACT, V_U_DETFACT, DESCT_DETFACT,
IVA_DETFACT, VALORNET_DETFACT,
TIPIPAGO_DETFACT,NUMFACT_DETFACT)
VALUES
('$compania','$cfacDturamaxid','$cod','$cfacturamaxid','$nom','$cant','$venta'
,'0','12','$importe','CONTADO','$cfactura')";

mysql_query($dettalle_sql);

$query_auditoria =
"SELECT max(auditoria.ID_AUDITORIA) FROM auditoria WHERE
auditoria.ID_EMPRESA='$compania' AND
auditoria.USUARIO_AUDITORIA='$usuario'";

```



```

$auditoria = mysql_query($query_auditoria, $php_conexion) or
die(mysql_error());
$row_auditoria = mysql_fetch_assoc($auditoria);
$totalRows_auditoria = mysql_num_rows($auditoria);
$id= $row_auditoria['max(auditoria.ID_AUDITORIA)'];

$insertSQL1 = sprintf("INSERT INTO usuario_temporal
(ID_EMPRESA,fecha,accion,usu,ID_AUDITORIA) VALUES
('.$compania.',NOW(),'INSERTAR DETALLE DE LA
FACTURA','$usuario','$id')");
mysql_select_db($database_php_conexion,
$php_conexion);
$result2 = mysql_query($insertSQL1, $php_conexion) or
die(mysql_error());
//INGRESAR DETALLE EN KARDEX

//saco el saldo anterior para calcular el promedio ponderado

$id_detalle_kardex=mysql_query("SELECT
max(ID_DETALLE_KARDEX) FROM detalle_kardex where
ID_EMPRESA='$compania' AND ID_CABEZERA_KARDEX='$id_kar' ");

if($id_detalle_kardex_1=mysql_fetch_array($id_detalle_kardex))
{
echo
$id_detalle_kardex_id=$id_detalle_kardex_1['max(ID_DETALLE_KARDEX)'];
}
//busco el saldo existente
por el id

$saldo_detalle_kardex=mysql_query("SELECT
CANSAL_DETALLE_KARDEX,VLTSAL_DETALLE_KARDEX,VLUSAL_DETALLE_KARDEX
FROM detalle_kardex where
ID_EMPRESA='$compania' AND
ID_DETALLE_KARDEX='$id_detalle_kardex_id' ");

if($id_detalle_kardex_saldo=mysql_fetch_array($saldo_detalle_kardex))

```

```
{

$cantidad_detalle_kardex_saldo=$id_detalle_kardex_saldo['CANSAL_DET
ALLE_KARDEX'];

$valtotal_detalle_kardex_saldo=$id_detalle_kardex_saldo['VLTSAL_DET
ALLE_KARDEX'];

$costo_detalle_kardex_saldo=$id_detalle_kardex_saldo['VLUSAL_DETAL
LE_KARDEX'];

}

// calculo el promedio
ponderado

$saldo_valortotal=$valtotal_detalle_kardex_saldo-
$importe;

$saldo_cantitotal=$cantidad_detalle_kardex_saldo-
$cant;

$costo_venta=$costo_detalle_kardex_saldo;

//INGRESAR DETALLE
EN KARDEX

$Kardex_sql1="INSERT INTO detalle_kardex
(ID_EMPRESA,
ID_CABEZERA_KARDEX,
FECH_DETALLE_KARDEX,
CONP_DETALLE_KARDEX,
CANENT_DETALLE_KARDEX,
VLUENT_DETALLE_KARDEX,
VLTENT_DETALLE_KARDEX,
CANSAL_DETALLE_KARDEX,
VLUSAL_DETALLE_KARDEX,
VLTSAL_DETALLE_KARDEX,
CANSLD_DETALLE_KARDEX,
VLUSLD_DETALLE_KARDEX,
VLTSLD_DETALLE_KARDEX)

VALUES

('$compania',
```

```
'$id_kar',  
  
CURRENT_TIMESTAMP,  
  
'VENTAS',  
  
'0',  
  
'0',  
  
'0',  
  
'$saldo_cantitotal',  
  
'$costo_venta',  
  
'$saldo_valortotal',  
  
'$cant',  
  
'$costo_venta',  
  
'$importe')";  
  
mysql_query($Kardex_sql1);
```

```
                                $query_auditoria      =  
"SELECT  max(auditoria.ID_AUDITORIA) FROM auditoria WHERE  
auditoria.ID_EMPRESA='$compania'                                AND  
auditoria.USUARIO_AUDITORIA='$usuario';  
$auditoria      =      mysql_query($query_auditoria,      $php_conexion)      or  
die(mysql_error());  
$row_auditoria = mysql_fetch_assoc($auditoria);  
$totalRows_auditoria = mysql_num_rows($auditoria);  
$id= $row_auditoria['max(auditoria.ID_AUDITORIA)'];
```



```
$insertSQL1 = sprintf("INSERT INTO usuario_temporal
(ID_EMPRESA, fech, accion, usu, ID_AUDITORIA)
VALUES
('".$compania."', NOW(), 'INSERTAR KARDEX', '$usuario', '$id')");
mysql_select_db($database_php_conexion,
$php_conexion);
$Result2 = mysql_query($insertSQL1, $php_conexion) or
die(mysql_error());
```

```
////ACTUALIZAR LA
EXISTENCIA//////////
```

```
$ca=mysql_query("SELECT * FROM
cabezera_kardex where ID_EMPRESA='$compania' AND
ID_PRODUCTO='$cod' ");
```

```
if($date=mysql_fetch_array($ca))
{
$e_actual=$date['EXIS_PROD_CABEZERA_KARDE
X'];
}
```

```
$n_cantidad=$e_actual-
$cant;
```

```
if($saldo_cantitotal<0)
{
$saldo_cantitotal=0;
} // si la cantidad da
negativo ponerlo en 0
```

```
$sql="Update
cabezera_kardex Set
EXIS_PROD_CABEZERA_KARDEX='$saldo_cantitotal' Where
ID_EMPRESA='$compania' AND ID_PRODUCTO='$cod' ";
mysql_query($sql);
```

```
//////////
```

```

                                $query_auditoria      =
"SELECT  max(auditoria.ID_AUDITORIA) FROM auditoria WHERE
auditoria.ID_EMPRESA='$compania'                                AND
auditoria.USUARIO_AUDITORIA='$usuario'";
$auditoria      =      mysql_query($query_auditoria,      $php_conexion)      or
die(mysql_error());
$row_auditoria = mysql_fetch_assoc($auditoria);
$totalRows_auditoria = mysql_num_rows($auditoria);
$id= $row_auditoria['max(auditoria.ID_AUDITORIA)'];

$insertSQL1      =      sprintf("INSERT      INTO      usuario_temporal
(ID_EMPRESA,fecha,accion,usu,ID_AUDITORIA)                                VALUES
('".$compania."',NOW(),'ACTUALIZAR      ESTADOS      DE
PRODUCTOS','$usuario','$id')");
                                mysql_select_db($database_php_conexion,
$php_conexion);
$result2      =      mysql_query($insertSQL1,      $php_conexion)      or
die(mysql_error());
                                }
                                $borrar_sql="DELETE FROM temp_factura
WHERE                                ID_EMPRESA='$compania'                                AND
USUARIO_TEMPFACT='$usuario';mysql_query($borrar_sql);
                                echo $numero_factura_cliente;
                                $query_auditoria      =      "SELECT
max(auditoria.ID_AUDITORIA) FROM auditoria WHERE
auditoria.ID_EMPRESA='$compania'                                AND
auditoria.USUARIO_AUDITORIA='$usuario'";
$auditoria      =      mysql_query($query_auditoria,      $php_conexion)      or
die(mysql_error());
$row_auditoria = mysql_fetch_assoc($auditoria);
$totalRows_auditoria = mysql_num_rows($auditoria);
$id= $row_auditoria['max(auditoria.ID_AUDITORIA)'];

$insertSQL1      =      sprintf("INSERT      INTO      usuario_temporal
(ID_EMPRESA,fecha,accion,usu,ID_AUDITORIA)                                VALUES
('".$compania."',NOW(),'ELIMINAR LOS DATOS DE LA FACTURA
TEMPORAL','$usuario','$id')");
                                mysql_select_db($database_php_conexion,
$php_conexion);
$result2      =      mysql_query($insertSQL1,      $php_conexion)      or
die(mysql_error());

```



```
header('location:factura_imprimir.php?tpagar='.$tpagar.'  
&ccpago='.$ccpago.'&factura='.$numero_factura_cliente.'&cliente='.$cliente);
```

```
}  
$_SESSION['ddes']=0; ?>
```

```
<?php session_start();  
include('Connections/php_conexion.php');  
$usuario=$_SESSION['username'];  
$compania=$_SESSION['compania'];  
if(!$_SESSION['sesion']=='si' ){  
header('location:error.php');  
}  
mysql_select_db($database_php_conexion, $php_conexion);  
$scanidfact=mysql_query("SELECT MAX(ID_FACTURA) as maximo  
FROM factura WHERE TIPO_FAC_FACTURA='N' ",  
$php_conexion);//codigo de la factura  
if($datoidfact=mysql_fetch_array($scanidfact))  
{  
$id_factura_cliente=$datoidfact['maximo'];  
}  
$scanidfact=mysql_query("SELECT NUM_FACTURA as maximo FROM  
factura WHERE TIPO_NOTA_CRED_FACTURA='N' AND  
TIPO_FAC_FACTURA='N' AND ID_FACTURA='$id_factura_cliente' ",  
$php_conexion);//codigo de la factura  
if($datonumfact=mysql_fetch_array($scanidfact))  
{  
$StrNumeroFactura=$datonumfact['maximo'];  
$StrCadenaFactura= substr($StrNumeroFactura, -13);  
if($StrCadenaFactura<=10010000000000)  
{  
$numero_nota_cliente='0010010000000001';  
}  
else  
{  
$numero_nota_cliente='00'.($StrCadenaFactura+1);  
}  
$cfactura1=$datoidfact['maximo'];  
}  
mysql_select_db($database_php_conexion, $php_conexion);
```



```

$canfactid=mysql_query("SELECT MAX(ID_FACTURA) as maxidf
FROM factura ", $php_conexion);//codigo de la factura
if($datofactid=mysql_fetch_array($canfactid))
{
$cfacturamaxid=$datofactid['maxidf']+1;
}

if($_GET['button']=='Guardar')
{ //contado
$descuento_producto=$_GET['descuento_pro'];
$factura=$_GET['factura'];
$tpagar=$_GET['tpagar'];
$cliente=$_GET['cliente'];
$t_importe=0;
$subtotal=$_GET['subtotal'];
$numero_factura=$_GET['txt_numero_factura'];
$total=$_GET['total'];
//guarda tabla factura

$factura_sql="INSERT INTO
factura (ID_EMPRESA,

ID_CLIENTE_PROVEDOR,

NUM_FACTURA,

FECHA_FACTURA,

EST_FACTURA,

VT_FACTURA,

USU_FACTURA

,CAJERA_FACTURA,

TIPO_NOTA_CRED_FACTURA,

NETO_FACTURA,

TIPO_FAC_FACTURA

CANT_NOTA_CRED_FACTURA,
```

HIST_NOTA_CRED_FACTURA,

FACT_NOTA_CRED_FACTURA)

VALUES

('Scompania','\$cliente','\$numero_nota_cliente',NOW(),'A','\$total','\$usuario','\$usuario','N','\$subtotal','N','\$total','\$total','\$numero_factura')";

mysql_query(\$factura_sql);

echo mysql_error();

max(auditoria.ID_AUDITORIA) FROM auditoria WHERE
auditoria.ID_EMPRESA='\$compania' AND
auditoria.USUARIO_AUDITORIA='\$usuario';

\$auditoria = mysql_query(\$query_auditoria, \$php_conexion) or
die(mysql_error());

\$row_auditoria = mysql_fetch_assoc(\$auditoria);

\$totalRows_auditoria = mysql_num_rows(\$auditoria);

\$id= \$row_auditoria['max(auditoria.ID_AUDITORIA)'];

\$insertSQL1 = sprintf("INSERT INTO usuario_temporal
(ID_EMPRESA,fecha,accion,usu,ID_AUDITORIA) VALUES
('.\$compania.',NOW(),'INSERTAR NOTA DE
CREDITO','\$usuario','\$id')");

mysql_select_db(\$database_php_conexion,
\$php_conexion);

\$Result2 = mysql_query(\$insertSQL1, \$php_conexion) or
die(mysql_error());

//codigo de la factura / guarda en

detalles

mysql_select_db(\$database_php_conexion,
\$php_conexion);

\$scan=mysql_query("SELECT * FROM temp_factura
where ID_EMPRESA='\$compania' AND



```
USUARIO_TEMPFACT='$usuario' AND TIPO_TEMPFACT='N' ",  
$php_conexion);
```

```
$cantotal=0;
```

```
while($dato=mysql_fetch_array($scan))
```

```
{
```

```
scanDfactid=mysql_query("SELECT  
MAX(ID_DETFACT) as maxidDfact FROM det_factura", $php_conexion);
```

```
//codigo de la factura
```

```
if($datoDfactid=mysql_fetch_array(scanDfactid))
```

```
{
```

```
$cfacDturamaxid=$datoDfactid['maxidDfact']+1;
```

```
}
```

```
//obtiene secuencial de
```

```
kardex
```

```
mysql_select_db($database_php_conexion, $php_conexion);
```

```
$kardexseq =  
mysql_query("SELECT MAX(ID_CABEZERA_kARDEX) as maxikardex  
FROM cabeza_kardex", $php_conexion);
```

```
if($datokardex =  
mysql_fetch_array($kardexseq))
```

```
{
```

```
$secidkardex =  
$datokardex['maxikardex']+1;  
  
}  
  
$cod=$dato['COD_TEMPFACT'];//codigo de producto  
en temporal de la factura  
  
$nom=$dato['NOMBRE_TEMPFACT'];  
  
$cant=$dato['CANTIDAD_TEMPFACT'];  
  
$cantotal=$dato['CANTIDAD_TEMPFACT']+$cantot  
al;  
  
$venta=$dato['VENTA_TEMPFACT'];  
  
$importe=$dato['IMPORTE_TEMPFACT'];  
  
$costo=$dato['VENTA_TEMPFACT'];  
  
$t_importe=$t_importe+$importe;  
  
mysql_select_db($database_php_conexion,  
$php_conexion);  
$PRODUCTO=mysql_query("SELECT producto.ID_PRODUCTO  
FROM producto  
WHERE producto.ID_EMPRESA='$compania' AND  
producto.DESCR_PRODUCTO='$nom' ", $php_conexion);  
if($datosPRODUCTO=mysql_fetch_array($PRODUCTO))  
{  
  
echo $sid_pro =  
$datosPRODUCTO['ID_PRODUCTO'];  
}  
  
mysql_select_db($database_php_conexion, $php_conexion);  
$PRODUCTO1=mysql_query("SELECT  
cabecera_kardex.ID_CABEZERA_KARDEX  
FROM  
cabecera_kardex  
WHERE
```



```

cabezera_kardex.ID_PRODUCTO = '$id_pro' AND
cabezera_kardex.ID_EMPRESA = '$compania"', $php_conexion);
if($datosPRODUCTO1=mysql_fetch_array($PRODUCTO1))
{

                                echo                                $id_kar                                =
$datosPRODUCTO1['ID_CABEZERA_KARDEX'];
}

$detalle_sql="INSERT INTO det_factura (ID_EMPRESA,ID_DETFACT,
ID_PRODUCTO,ID_FACTURA,                                REGISTRO_DETFACT,
CANT_DETFACT,                                V_U_DETFACT,                                DESCOT_DETFACT,
IVA_DETFACT,                                VALORNET_DETFACT,
TIPIPAGO_DETFACT,NUMFACT_DETFACT)

                                VALUES
('$compania','$cfacDturamaxid','$cod','$cfacturamaxid','$nom','$cant','$venta'
,'0','12','$importe','NOTA CREDITO','$cfactura')";

                                mysql_query($detalle_sql);

$query_auditoria = "SELECT max(auditoria.ID_AUDITORIA) FROM
auditoria WHERE auditoria.ID_EMPRESA='$compania' AND
auditoria.USUARIO_AUDITORIA='$usuario'";
$auditoria = mysql_query($query_auditoria, $php_conexion) or
die(mysql_error());
$row_auditoria = mysql_fetch_assoc($auditoria);
$totalRows_auditoria = mysql_num_rows($auditoria);
$id= $row_auditoria['max(auditoria.ID_AUDITORIA)'];

$insertSQL1 = sprintf("INSERT INTO usuario_temporal
(ID_EMPRESA,fecha,accion,usu,ID_AUDITORIA) VALUES
('".$compania."',NOW(),'INSERTAR DETALLE DE LA NOTA DE
CREDITO','$usuario','$id')");

                                mysql_select_db($database_php_conexion,
$php_conexion);
$result2 = mysql_query($insertSQL1, $php_conexion) or
die(mysql_error());
//saco el saldo anterior para calcular el promedio ponderado

$id_detalle_kardex=mysql_query("SELECT
max(ID_DETALLE_KARDEX) FROM detalle_kardex where
ID_EMPRESA='$compania' AND ID_CABEZERA_KARDEX='$id_kar' ");

```

```
if($id_detalle_kardex_1=mysql_fetch_array($id_detalle_kardex))
{
    $id_detalle_kardex_id=$id_detalle_kardex_1['max(ID_DETALLE_KARDEX)'];

}

//busco el saldo existente
por el id

    $saldo_detalle_kardex=mysql_query("SELECT
CANSAL_DETALLE_KARDEX,VLTSAL_DETALLE_KARDEX FROM
detalle_kardex where ID_EMPRESA='$compania' AND
ID_DETALLE_KARDEX='$id_detalle_kardex_id' ");

if($id_detalle_kardex_saldo=mysql_fetch_array($saldo_detalle_kardex))
{
    echo
    $cantidad_detalle_kardex_saldo=$id_detalle_kardex_saldo['CANSAL_DETALLE_KARDEX'];

    echo
    $valtotal_detalle_kardex_saldo=$id_detalle_kardex_saldo['VLTSAL_DETALLE_KARDEX'];

}

// calculo el promedio
ponderado

    echo
    $saldo_valortotal=$valtotal_detalle_kardex_saldo+$importe;
    echo
    $saldo_cantitotal=$cantidad_detalle_kardex_saldo+$cant;
    echo
    $costo_venta=$saldo_valortotal/$saldo_cantitotal;

//INGRESAR DETALLE
EN KARDEX
```

```
$Kardex_sql1="INSERT INTO detalle_kardex  
(ID_EMPRESA,  
ID_CABEZERA_KARDEX,  
FECH_DETALLE_KARDEX,  
CONP_DETALLE_KARDEX,  
CANENT_DETALLE_KARDEX,  
VLUENT_DETALLE_KARDEX,  
VLTENT_DETALLE_KARDEX,  
CANSAL_DETALLE_KARDEX,  
VLUSAL_DETALLE_KARDEX,  
VLTSAL_DETALLE_KARDEX,  
CANSLD_DETALLE_KARDEX,  
VLUSLD_DETALLE_KARDEX,  
VLTSLD_DETALLE_KARDEX)
```

VALUES

('\$compania',

'\$id_kar',

CURRENT_TIMESTAMP,

'DEBOLUCION',

'\$cant',

'\$costo',

'\$importe',

'\$saldo_cantitotal',

'\$costo_venta',

'\$saldo_valortotal',

'0',

'0',

'0')";


```

mysql_query($Kardex_sql1);

echo mysql_error();

$query_auditoria = "SELECT
max(auditoria.ID_AUDITORIA) FROM auditoria WHERE
auditoria.ID_EMPRESA='$compania' AND
auditoria.USUARIO_AUDITORIA='$usuario'";
$auditoria = mysql_query($query_auditoria, $php_conexion) or
die(mysql_error());
$row_auditoria = mysql_fetch_assoc($auditoria);
$totalRows_auditoria = mysql_num_rows($auditoria);
$id= $row_auditoria['max(auditoria.ID_AUDITORIA)'];

$insertSQL1 = sprintf("INSERT INTO usuario_temporal
(ID_EMPRESA,fecha,accion,usu,ID_AUDITORIA) VALUES
('".$compania."',NOW(),'INSERTAR DETALLE DEL
KARDEX','$usuario','$id')");
mysql_select_db($database_php_conexion,
$php_conexion);
$result2 = mysql_query($insertSQL1, $php_conexion) or
die(mysql_error());

$sql="Update
cabezera_kardex Set
EXIS_PROD_CABEZERA_kARDEX='$saldo_cantitotal',COSTO_PROD_
CABEZERA_kARDEX='$costo_venta' Where ID_EMPRESA='$compania'
AND ID_PRODUCTO='$cod' ";

mysql_query($sql);
$query_auditoria =
"SELECT max(auditoria.ID_AUDITORIA) FROM auditoria WHERE
auditoria.ID_EMPRESA='$compania' AND
auditoria.USUARIO_AUDITORIA='$usuario'";
$auditoria = mysql_query($query_auditoria, $php_conexion) or
die(mysql_error());
$row_auditoria = mysql_fetch_assoc($auditoria);
$totalRows_auditoria = mysql_num_rows($auditoria);

```



```
$id= $row_auditoria['max(auditoria.ID_AUDITORIA)'];
```

```
$insertSQL1 = sprintf("INSERT INTO usuario_temporal
(ID_EMPRESA,fecha,accion,usu,ID_AUDITORIA) VALUES
('.$compania.',NOW(),'ACTUALIZAR ESTADO DE LOS
PRODUCTOS','$usuario','$id')");
mysql_select_db($database_php_conexion,
$php_conexion);
$Result2 = mysql_query($insertSQL1, $php_conexion) or
die(mysql_error());
```

```
////////////////////////////////////
```

```
$borrar_sql="DELETE FROM temp_factura
WHERE ID_EMPRESA='$compania' AND
USUARIO_TEMPFACT='$usuario' AND
TIPO_TEMPFACT='N'";mysql_query($borrar_sql);
```

```
$query_auditoria =
"SELECT max(auditoria.ID_AUDITORIA) FROM auditoria WHERE
auditoria.ID_EMPRESA='$compania' AND
auditoria.USUARIO_AUDITORIA='$usuario'";
$auditoria = mysql_query($query_auditoria, $php_conexion) or
die(mysql_error());
$row_auditoria = mysql_fetch_assoc($auditoria);
$totalRows_auditoria = mysql_num_rows($auditoria);
$id= $row_auditoria['max(auditoria.ID_AUDITORIA)'];
```

```
$insertSQL1 = sprintf("INSERT INTO usuario_temporal
(ID_EMPRESA,fecha,accion,usu,ID_AUDITORIA) VALUES
('.$compania.',NOW(),'ELIMINAR LA FACTURA
TEMPORAL','$usuario','$id')");
mysql_select_db($database_php_conexion,
$php_conexion);
$Result2 = mysql_query($insertSQL1, $php_conexion) or
die(mysql_error());
```



}

\$numero_nota_cliente;

```
mysql_select_db($database_php_conexion, $php_conexion);  
$canfactid=mysql_query("SELECT ID_FACTURA FROM factura  
WHERE factura.NUM_FACTURA='$numero_nota_cliente' AND  
factura.TIPO_FAC_FACTURA = 'N' AND
```

A.06 Cronograma de Actividades

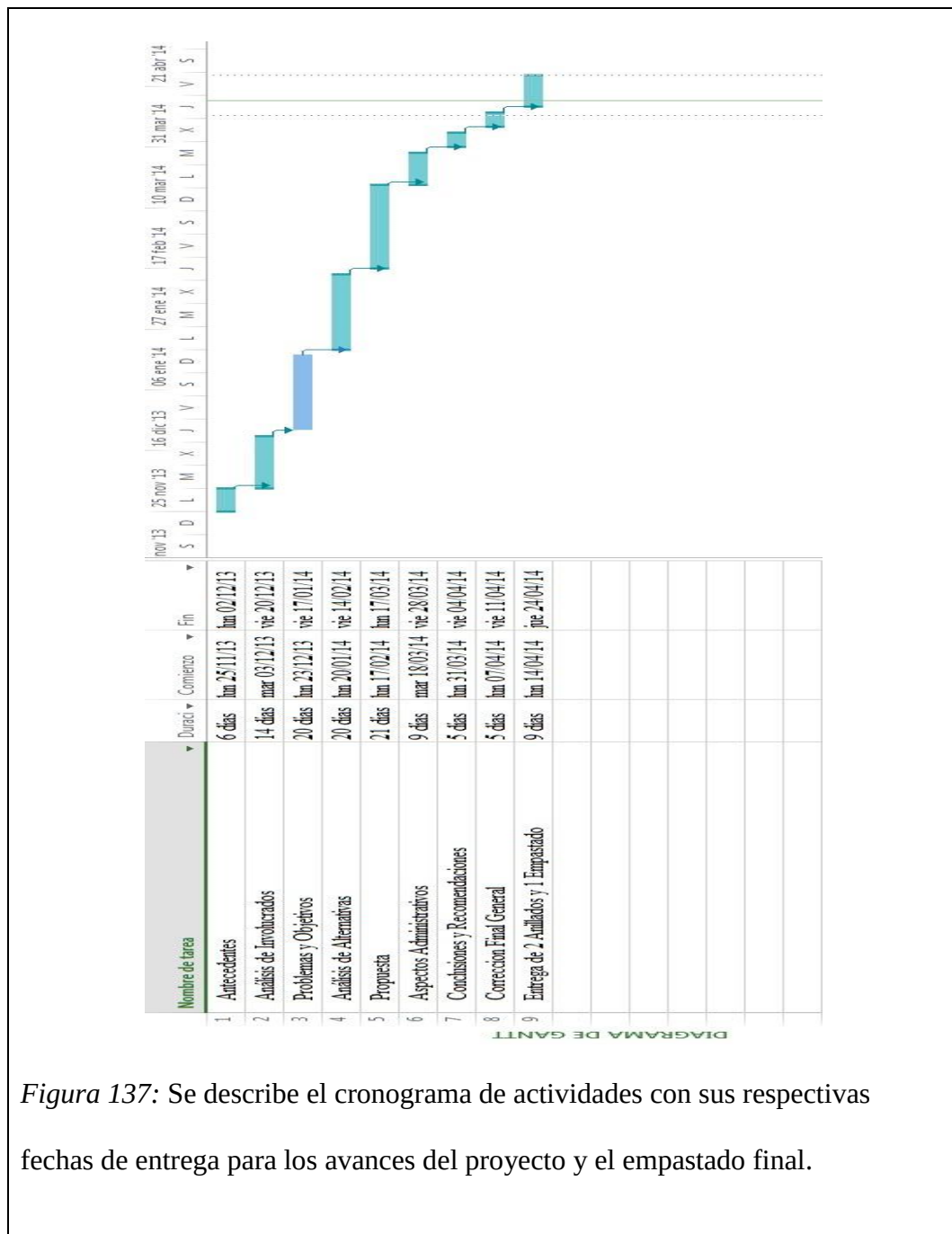


Figura 137: Se describe el cronograma de actividades con sus respectivas fechas de entrega para los avances del proyecto y el empastado final.

A.07 Glosario de Términos

Buscador: Sitios en la web que catalogan y organizan el contenido en la web para ayudar a los usuarios a encontrar la información que están buscando.

Browser (Navegador): Un programa que permite al usuario navegar por la web. Los más populares son Mozilla Firefox e Internet Explorer. También existen otros browsers. Lynx es un programa gratuito que permita navegar la WEB en terminales que carecen de gráficas. Otro browser importante es Mosaic, que es el que comenzó la gran expansión en la web en 1994. Opera es un browser reciente cuyo atractivo principal es que despliega las páginas mucho más rápido que Explorer, Firefox o Netscape.

Cookies:: Los cookies o galletas son una tecnología implementada en los browsers de Internet que permiten que el servidor archive información relacionada a los patrones de comportamiento del usuario en su propia máquina. Son pequeños archivos que un programa deja en su PC, por ejemplo para saber que ese programa en ese equipo ya se ejecutó.

Dominio: Es la parte de una dirección de Internet que indica el domicilio del servidor. Su utilidad principal es la búsqueda de direcciones IP de sistemas centrales ("hosts") basándose en los nombres de estos. Por ejemplo en la dirección: <http://www.mecon.gov.ar/consumidor> la parte que dice www.mecon.gov.ar es el dominio.

El resto de la dirección apunta a archivos dentro del servidor del Ministerio de Economía.

Las partes que lo componen se denominan "sub-dominios". Por ejemplo: Mecon es

sub-dominio de nombre de la empresa/institución (en este caso el Ministerio de Economía”).

gov/com/edu/mil/net son sub-dominios de clase o de rubro ar/br/eu/au/es/etc. Son sub-dominios del país.

EDI: Intercambio electrónico de datos.

Encriptado: Es el tratamiento de los datos a fin de impedir que nadie excepto el destinatario de los mismos pueda leerlos. Hay muchos tipos de cifrado de datos, que constituyen la base de la seguridad de la red.

FTP: File Transfer Protocol (FTP) (Protocolo de Transferencia de Archivos): Protocolo que permite a un usuario de un sistema acceder y transferir, a y desde, otro sistema de red.

FTP es también habitualmente el nombre del programa que el usuario invoca para ejecutar el protocolo.

Html5: es el nombre que se usa para referirse a la quinta revisión del lenguaje HTML. Una característica especial de HTML 5 es que es el resultado de agrupar las especificaciones relacionadas al desarrollo de páginas web: HTML 4 (es el estándar en uso desde 1997), XHTML 1 ([enlace en inglés](#)), DOM nivel 2 (Document Object Model -Modelo de objetos del documento, en español-), e integrar algunos elementos de CSS nivel 2.

Host: (Sistema Central): Ordenador que permite a los usuarios conectarse con otros sistemas centrales de una red. Los usuarios se comunican utilizando programas de aplicación, tales como el correo electrónico, Telnet, WWW y FTP.

HTML (Hyper Text Markup Language): Es un lenguaje de programación de computadoras que permite al usuario crear documentos de hipertexto para su publicación en la web.

HTTP (Hyper Text Transfer Protocol): Es un protocolo de comunicación de datos que permite la transmisión de documentos de hipertexto a través de redes. Es el protocolo en el que está basado el website.

Hypertext (Hipertexto): Documento escrito en HTML que contiene enlaces a otros documentos que pueden ser a su vez documentos hipertexto. Los documentos hipertexto son accedidos normalmente a través de navegadores WWW.

Hosting: El alojamiento web (en inglés web hosting) es el servicio que provee a los usuarios de Internet un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía Web. Los Web Host son compañías que proporcionan espacio de un servidor a sus clientes.

Internet: Internet es la red de computadoras más grande del mundo. Tuvo sus inicios en los EE.UU. durante la guerra fría bajo la amenaza de un holocausto nuclear. El Dpto. de Defensa de los EE.UU. desarrolló un sistema de telecomunicaciones digitales descentralizado que es extremadamente fluido y flexible.

En un principio la aplicación principal era el correo electrónico, que permite a personas geográficamente distantes comunicarse de maneras muy eficientes. Con la aparición del "World-Wide Web" en 1993, Internet gozó de un resurgimiento de interés por parte del público en general.

Intranet: Red propia de una organización, diseñada y desarrollada siguiendo los protocolos propios de Internet; en particular el protocolo TCP/IP. Puede tratarse de una red aislada, es decir, no conectada a Internet.

Ip Address (Dirección IP): Dirección de 32 bits definida por el Protocolo de Internet. Se representa usualmente mediante notación decimal separada por puntos. Un ejemplo de Ip Address es: 193.127.88.345.

JavaScript: abreviado comúnmente "JS" es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos,³ basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas⁴ aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

NetBeans: NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE2 es un producto libre y gratuito sin restricciones de uso.

Página Web: Una página es un documento dentro de un sitio de Internet que contiene información. Por lo general (aunque no necesariamente) están en un lenguaje que se llama HTML.

Password: Contraseña.

PHP: Lenguaje de programación que se ejecuta del lado del servidor.

PDF (Portable Document Format): El formato utilizado por el Acrobat de Adobe para representar documentos de manera independiente al sistema operativo en que fueron creados.

Los archivos *.PDF pueden incluir texto y gráficas; además de poder ser representados en diferentes computadoras y sistemas operativos. Para poder ver un documento *.PDF, es necesario utilizar el Acrobat de Adobe Systems.

Plug and Play (Enchufar y desenchufar): Se aplica a los accesorios para computadoras, internos y/o externos, que no necesitan de una configuración técnica, sino que con sólo enchufarlos estarán listos para ser utilizados.

Protocolo: Es una lengua común que se establece entre dos computadoras para que estas puedan comunicarse. Existen muchos protocolos. En Internet, el protocolo más importante es el TCP/IP.

Red: Dos o más computadoras conectadas entre sí, permitiendo a sus usuarios compartir archivos y recursos.

SEC: Securities and Exchange Commission (Bolsa de Valores Norteamericana).

Servidor: Una computadora en una red cuyo propósito es “dar acceso a” (servir) archivos. El otro polo en la relación es el cliente, que es la computadora que interpreta y despliega la información siendo servida.

Sistema Operativo: El software de nivel bajo que se encarga de organizar la operación de la computadora, asignar recursos, manejar la interfaz de los diversos periféricos, y comunicarse con el usuario.

El sistema operativo es el software más importante de un computador, ya que sin él

la operación del computador sería imposible. Algunos de los sistemas operativos más comunes son: Windows, MS-DOS, MacOS, Linux y Solaris.

Spam: Es cualquier mensaje destinado a una audiencia general (o sea, un comunicado de masas) que se envía a la gente por e-mail sin que lo soliciten. Generalmente las direcciones a las que llegan esos mails son robadas y la mayoría de las veces esos mails son publicidad o contienen algún virus. Representa una violación a la privacidad del usuario.

Start ups: Puesta en marcha.

TI: Tecnología Informática.

Usenet: Una aplicación de Internet que permite a personas con intereses similares compartir información a través de listas de correo electrónico organizadas y archivables.

Los grupos de noticias son una herramienta para buscar información sobre diversos temas, ya que reúne el conocimiento colectivo de grandes masas de personas.

WWW (World Wide Web): La aplicación más importante de Internet es este sistema de hipermedia desarrollado por Tim Berners Lee en 1993. Está basado en un lenguaje de programación que se llama HTML, y permite desplegar archivos e información gráfica y auditiva de manera que uno puede interconectar documentos fácilmente. La web es tan popular que mucha gente se refiere al mismo simplemente como Internet, sin tomar en cuenta que la web es sólo una de las muchas aplicaciones de Internet.

Webmaster: Es la persona encargada de la programación, operación y mantenimiento de un sitio Web y en algunos casos se asigna la tarea de análisis

comercial del sitio para su promoción y posicionamiento en buscadores con la finalidad de vender o darse a conocer por este canal.

XML: Las siglas de "eXtensible Markup Language" o Lenguaje de Marcas Extensible, es un estándar para el intercambio de información estructurada entre diferentes plataformas.

Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

XML es una tecnología que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.