

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
FACULTAD DE INGENIERÍA
ESCUELA DE SISTEMAS**



**DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
SISTEMAS Y COMPUTACIÓN**

**“DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE
PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE
FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER
SOLUTION”**

AUTOR:

**FERNANDA CAROLINA NICOLALDE TIPÁN
DIEGO PATRICIO ROSERO VALENCIA**

DIRECTOR:

ING. ALFREDO CALDERÓN

QUITO, MARZO 2016

RESUMEN

En esta disertación de grado se presenta principalmente el desarrollo del sistema para control de adquisición y venta de productos tecnológicos incluyendo un módulo de facturación electrónica aplicado a la empresa Computer Solution; en los primeros capítulos se explica la problemática del tema en general y a su vez el estudio de las técnicas para la construcción y desarrollo de la aplicación.

Dentro del primer capítulo se explica, la formulación del problema, tomando en cuenta la situación de la empresa y el por qué se ha tomado una solución informática dentro de la misma, además cumpliendo con los estándares de envío de facturas en formato PDF la empresa ha buscado una solución adicional a la antes ya propuesta lo cual implica en el desarrollo de un módulo de facturación electrónica.

En el segundo capítulo se puede encontrar el marco teórico en base a los principales lenguajes de programación y gestores de bases de datos que se encuentran en el mundo de la informática siendo los más usados actualmente en diferentes sistemas aplicativos.

Luego se desarrolla el tercer capítulo donde se ven los objetivos, misión, visión, cuadro jerárquico y principales actividades del proceso de adquisición, venta de productos y generación de facturas para la empresa Computer Solution.

A través de levantamiento de procesos se puede hacer un análisis del proceso principal y sus subprocesos los cuales deben ser estudiados para generar un producto de calidad, con la ayuda de algunos diagramas entre los cuales se encuentran:

- Cadenas de Valor.
- Mapa de Procesos.
- Sipoc.

En el cuarto capítulo se habla de las herramientas las cuales se va a utilizar para realizar el desarrollo, contando con las diferentes técnicas de desarrollo, junto con la arquitectura de la aplicación; en el mismo capítulo se hace un análisis de las herramientas estudiadas anteriormente y se realiza cuadros comparativos entre los lenguajes de programación y gestores de bases para poder llegar a una selección de los mismos.

El capítulo cinco se toma el análisis y se procede a realizar la construcción del sistema en sí, previamente realizados los procesos de desarrollo de software los cuales son:

- Análisis de Requerimientos.

- Identificación de Actores.
- Especificación de Requerimientos.
- Arquitectura.
- Diseño.
- Diagramas UML (Incluido diagramas de despliegue y de secuencia).

Una vez realizado estos pasos se procedió a elaborar un prototipo ejecutable demostrado mediante capturas de pantalla que muestran la funcionalidad del sistema y posteriormente proseguir con la construcción del sistema completo; no antes de proceder a realizar una lista detallada de riesgos que representa el proyecto.

En la construcción final del sistema en el cual se muestra la implementación y pruebas del mismo para saber cómo funciona en general, esto se realiza con cuadros de detalle para tener conocimiento de cómo interactúa el usuario con el sistema y el mismo con el manejo de datos. Dentro del mismo capítulo se encuentra la información a detalle de que errores que se pueden cometer el usuario y a su vez cómo el sistema reaccionará a tales errores, junto a la interacción del sistema con el gestor de base de datos que permite hacer análisis a detalle de qué ingresa o qué no ingresa al mismo.

Finalmente se hace como se conoce en proyectos informáticos, “la transición”, la cual consiste en hacer un análisis de seguridades, costos del proyecto, despliegue y mantenimiento del sistema terminado.

Para la terminación de esta disertación de grado se podrá encontrar las conclusiones y las recomendaciones necesarias que implicaron el desarrollo de esta disertación de grado.

DEDICATORIA

Este trabajo de Tesis va dedicado a mi Familia ya que gracias a ellos estoy cumpliendo una de mis metas, a mi esposa por brindarme toda su comprensión y ayuda en momentos malos y buenos, me han enseñado a enfrentar las adversidades de la vida sin perder nunca la calma, ni desfallecer en el intento. Gracias por formar una persona con valores, principios y empeño, todo ello con una gran perseverancia y sin recibir nunca nada a cambio y así terminar esta disertación.

A todos ellos, muchas gracias.

DIEGO ROSERO

Esta tesis va dedicada a Dios que guiándome cada día me permitió cumplir esta meta, a mi madre y a mi tía, pilares fundamentales en mi vida que con su amor, cariño, con cada una de las enseñanzas siempre velaron por mi bienestar y educación siendo mi apoyo en cada instante. Es por ellas que soy lo que soy ahora.

FERNANDA NICOLALDE

AGRADECIMIENTOS

A mi Familia.

A mis padres Ing. Jaime Rosero y Lic. Martha Valencia los cuales fueron mi sustento económico y anímico en toda mi carrera universitaria, a mi hermano el Lic. Jaime Rosero ya que siempre creyó en mí al momento de plasmar mi idea del plan de tesis en un desarrollo digno de transformarse en Tesis y contar siempre con su apoyo.

A mi Esposa e Hija

A mi esposa Sigrid Beltrán por siempre creer en mí y brindarme su apoyo en todo momento de mi vida estudiantil y de la misma forma a mi hija Kamila porque ella siempre me ha dados momentos de felicidad como solo un hijo los puede ya que ella con solo una sonrisa me alegra la vida.

A mis Tutores

Quienes fueron el motor importante en el desarrollo Ing. Alfredo Calderón, Ing. Ana Urgilés e Ing. Xavier Córdor, gracias a ustedes a su apoyo y a quienes me dictaron clases, gracias por llenarme de sus conocimientos, y a quien no fue así le agradezco por brindarme su amistad y su apoyo desde el primer día que le expuse el tema de mi tesis, gracias Anita.

DIEGO ROSERO

Agradezco a Dios por haberme guiado y acompañado durante este caminar, por darme una gran fortaleza en los momentos difíciles.

Agradezco a mi madre y mi tía por siempre ser un gran apoyo, por los valores inculcados, por permitirme tener una excelente educación, por el gran ejemplo de vida a seguir.

Agradezco a mi familia por ser parte importante para poder llegar a culminar esta etapa de mi vida, por brindarme cada consejo y ser una fuente de apoyo constante.

Agradezco a cada uno de los ingenieros que con sus conocimientos impartidos hicieron de mí una gran profesional.

FERNANDA NICOLALDE

TABLA DE CONTENIDOS

1. CAPÍTULO: FORMULACIÓN DEL PROBLEMA	1
1.1. Justificación	1
1.2. Antecedentes	1
1.3. Objetivos	1
1.3.1. General	1
1.3.2. Específicos	1
1.4. Alcance	2
2. CAPÍTULO: MARCO TEORICO	3
2.1. Técnicas de Desarrollo de Aplicaciones	3
2.1.1. Diseño Ágil de Aplicaciones (Extreme Programming).....	3
2.1.2. Rad (Diseño Rápido de Aplicaciones).....	4
2.1.2.1. Planificación de los Requisitos.	5
2.1.2.2. Diseño.....	5
2.1.2.3. Construcción:.....	5
2.1.2.4. Implementación:.....	5
2.1.2.5. Modelado de Gestión:	5
2.1.2.6. Modelado de Datos:	5
2.1.2.7. Modelado del Proceso:	6
2.1.2.8. Generación de Aplicaciones.	6
2.1.2.9. Entorno RAD Sobre Navegadores Web.....	6
2.1.3. JAD (Diseño de Aplicaciones por Juntas).....	7
2.1.3.1. ¿Qué es JAD?	8
2.1.3.2. Instalaciones Para el Desarrollo de JAD	8
2.1.3.3. Realizar la Sesión JAD.....	8
2.1.3.4. Preparación	8
2.1.3.5. El Período de Sesiones	9
2.1.3.6. Beneficios Principales	9
2.2. Análisis de Herramientas de Desarrollo	10
2.2.1. Sistema de Gestión de Bases de Datos	10
2.2.1.1. Servicios.....	10
2.2.1.2. Arquitectura	11
2.2.1.3. SQL Server	12
2.2.2. Lenguajes De Programación	13
2.2.2.1. .NET	13
2.2.2.2. Java.....	14
2.2.2.1. XML	14
2.3. Arquitectura De Aplicaciones.....	17
2.3.1. Por Capas	17
2.3.1.1. Capa de presentación.....	18
2.3.1.2. Capa de negocio.....	18
2.3.1.3. Capa de datos	18
3. CAPÍTULO: SITUACION DE LA EMPRESA.	19
3.1. Proceso de Solicitud y Entrega de Productos Tecnológicos en la Empresa Computer Solution. ...	19
3.1.1. Objetivos.....	19
3.1.1.1. General.....	19
3.1.1.2. Específicos	19

3.1.2.	Misión	19
3.1.3.	Visión	19
3.1.4.	Cuadro Jerárquico.	20
3.2.	Mapa de Procesos	21
3.2.1.	Descripción	21
3.2.2.	Actores.....	21
3.2.3.	Mapa de Procesos de la Empresa Computer Solution.....	21
3.2.4.	Actividades Principales del proceso de adquisición, venta de productos tecnológicos y generación de factura.	22
3.2.4.1.	Proceso de Adquisición de productos tecnológicos.....	22
3.2.4.2.	Proceso de Venta de productos tecnológicos.....	22
3.2.4.3.	Proceso de Generación de Facturas.	22
3.2.4.4.	Proceso Selección de Proveedor	22
3.2.5.	Problemas del Proceso Anterior	23
3.2.6.	Cadenas de Valor	23
3.2.7.	Mapa de procesos.....	24
3.2.1.	Gráficos de Soporte.....	25
3.2.1.1.	Tabla Proceso Administrar Productos.	25
3.2.1.2.	Gráfico Proceso Administrar Venta.	25
3.2.1.3.	Administrar Generación de XML	26
3.1.	Diagrama de flujo con metodología SIPOC.....	27
4.	CAPÍTULO: HERRAMIENTAS PARA EL DESARROLLO DEL SISTEMA.....	28
4.1.	Técnica de Desarrollo de Aplicaciones.	28
4.2.	Arquitectura De Aplicación.	28
4.3.	Lenguaje De Programación.	29
4.3.1.	C# vs JAVA.....	29
4.4.	Sistema Gestor de Base de Datos.....	30
4.4.1.	SQL Server vs MySQL.....	30
4.5.	IDE Para Desarrollo.	31
4.6.	Archivo XML.	32
5.	CAPÍTULO: CONSTRUCCIÓN DEL PRODUCTO.....	33
5.1.	Inicio.....	33
5.2.	Objetivos de la Aplicación	33
5.2.1.	Análisis De Requerimientos.....	33
5.2.2.	Definición De Actores.....	33
5.2.3.	Especificaciones De Requerimientos	35
5.2.3.1.	Funcionales.....	35
5.2.3.2.	No Funcionales	35
5.2.4.	Arquitectura.....	36
5.2.5.	Diseño.....	37
5.2.5.1.	Diseño Entidad/Relación (E/R).	37
5.2.5.2.	Diagramas UML	38
5.3.	Elaboración.....	48
5.3.1.	Prototipo Ejecutable	48
5.3.1.1.	Ventana Inicial (Página de Inicio).....	48
5.3.1.2.	Ventana Principal.....	48
5.3.2.	Lista De Riesgos.....	49
5.3.2.1.	Descripción	49
5.3.2.2.	Riesgos	50
5.4.	Construcción	51

5.4.1.	Implementación y Pruebas.....	51
5.4.1.1.	Ingreso al Sistema.....	51
5.4.1.2.	Ingreso de Usuarios.....	52
5.4.1.3.	Ingreso de Clientes.....	54
5.4.1.4.	Ingreso de proveedores.....	56
5.4.1.5.	Ingreso de Categorías.....	57
5.4.1.6.	Ingreso de Productos.....	57
5.4.1.7.	Ingreso inventario.....	58
5.4.1.7.1.	Pruebas validaciones de usuario.....	59
5.4.1.8.	Ingreso Venta.....	60
5.4.1.8.1.	Pruebas validaciones de usuario.....	61
5.4.1.9.	Generación de Factura PDF.....	61
5.4.1.10.	Creación Archivo XML.....	61
5.5.	Transición.....	62
5.5.1.	Seguridades.....	62
5.5.2.	Costos.....	62
5.5.3.	Despliegue.....	63
5.5.4.	Mantenimiento.....	63
6.	CAPÍTULO: CONCLUSIONES Y RECOMENDACIONES.....	65
6.1.	Conclusiones.....	65
6.2.	Recomendaciones.....	66
	BIBLIOGRAFÍA.....	68
	ANEXO A.....	70
1.	Manual de Usuario.....	70
1.1.	Módulo Bodega.....	70
1.1.1.	Ingreso Nuevo Producto.....	70
1.1.2.	Reporte de Productos en el Sistema.....	71
1.1.3.	Edición de Productos.....	72
1.1.4.	Ingreso de Nueva Categoría.....	73
1.1.5.	Reporte de Categorías en el Sistema.....	74
1.1.6.	Edición de Categoría.....	75
1.2.	Módulo Inventario.....	76
1.2.1.	Ingreso Nueva Compra.....	76
1.2.2.	Reporte de Compras en el Sistema.....	77
1.2.3.	Edición de Compras.....	78
1.2.4.	Ingreso Nuevo Proveedor.....	79
1.2.5.	Reporte de Proveedores.....	80
1.2.6.	Edición Proveedor.....	81
1.3.	Modulo Ventas.....	82
1.3.1.	Ingreso Nueva Venta.....	82
1.3.2.	Reporte de Venta.....	83
1.3.3.	Edición de Venta.....	85
1.3.4.	Ingreso Nuevo Cliente.....	86
1.3.5.	Reporte de Clientes en el Sistema.....	87
1.3.6.	Edición de Cliente.....	88
1.4.	Módulo Administración.....	89
1.4.1.	Ingreso Nuevo Usuario.....	89

1.4.2.	Reporte de Usuarios en el Sistema	90
1.4.3.	Edición de Usuarios	92
1.5.	Módulo Reportes.....	92
1.5.1.	Ventas.....	92
1.5.2.	Compras.....	93
1.5.3.	Stock Productos.....	94
ANEXO B.....		95
2.	Manual Técnico	95
2.1.	Capa de Acceso a Datos	95
2.1.1.	Conexión	95
2.1.2.	DArticulo.....	95
2.1.3.	DCliente	102
2.1.4.	DCategoria	111
2.1.5.	DDetalle_Ingreso.....	116
2.1.6.	DDetalle_Venta	119
2.1.7.	DIngreso.....	122
2.1.8.	DPresentacion.....	128
2.1.9.	DProveedores	134
2.1.10.	DTrabajador	143
2.1.11.	DVenta.....	153
2.2.	Capa de Negocios	162
2.2.1.	NArticulo.....	162
2.2.2.	NCategoria	163
2.2.3.	NCliente.....	164
2.2.4.	NIngreso.....	166
2.2.5.	NPresentacion.....	168
2.2.6.	NProveedor.....	169
2.2.7.	NTrabajador	171
2.2.8.	NVenta.....	173
2.3.	Script SQL de Base de Datos.....	175

TABLA DE ILUSTRACIONES

ILUSTRACIÓN 1: NIVELES DE ABSTRACCIÓN DE LA ARQUITECTURA ANSI. (RAMOS, RAMOS, & MONTERO, 2006).....	11
ILUSTRACIÓN 2: HTTP://WWW.MUNDOLINUX.INFO/INDEX.HTM	15
ILUSTRACIÓN 3: HTTP://WWW.MUNDOLINUX.INFO/INDEX.HTM	16
ILUSTRACIÓN 4: EJEMPLO XML HTTP://WWW.MUNDOLINUX.INFO/INDEX.HTM	17
ILUSTRACIÓN 5: MODELO 3 CAPAS REFERENCIA: HTTP://WWW.DESARROLLOWEB.COM/ARTICULOS/ARQUITECTURA-CLIENTE-SERVIDOR.HTML	18
ILUSTRACIÓN 6: CUADRO JERÁRQUICO. AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO.....	20
ILUSTRACIÓN 7: MAPA DE PROCESOS. AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO.	21
ILUSTRACIÓN 8: CADENA DE VALOR. AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO.....	23
ILUSTRACIÓN 9: MACROPROCESOS Y PROCESOS. AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO.	23
ILUSTRACIÓN 10: MAPA DE PROCESOS. AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO.....	24
ILUSTRACIÓN 11: MAPA DE PROCESOS. AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO.....	27
ILUSTRACIÓN 12: COMPARACIÓN GESTORES BASES DE DATOS. AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO.	31
ILUSTRACIÓN 13: ACTORES. AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO.	34
ILUSTRACIÓN 14: ARQUITECTURA 3 CAPAS. AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO.....	36
ILUSTRACIÓN 15: DISEÑO E/R. AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO.....	37
ILUSTRACIÓN 16: DIAGRAMA DP. AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO.	38
ILUSTRACIÓN 17: DIAGRAMA MD. AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO	39
ILUSTRACIÓN 18: DIAGRAMA GUI. AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO.....	40
ILUSTRACIÓN 19: DIAGRAMA DE DESPLIEGUE. AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO.	41
ILUSTRACIÓN 20: INGRESO DE PRODUCTOS. AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO.	42
ILUSTRACIÓN 21: MODIFICAR PRODUCTO. AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO.	43
ILUSTRACIÓN 22: ELIMINAR PRODUCTO. AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO.....	44
ILUSTRACIÓN 23: CONSULTA GENERAL PRODUCTOS. AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO.	45
ILUSTRACIÓN 24: CONSULTA PARÁMETROS PRODUCTOS. AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO.	46
ILUSTRACIÓN 25: INGRESAR PEDIDO. AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO.	47

INDICE DE TABLAS

TABLA 1: PROCESO ADMINISTRAR PRODUCTOS. AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO.	25
TABLA 2: PROCESO ADMINISTRAR VENTAS. AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO.	26
TABLA 3: PROCESO ADMINISTRAR GENERACIÓN DE XML. AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO.....	26
TABLA 4: CUADRO COMPARATIVO C# Y JAVA. AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO.	29
TABLA 5: CUADRO COMPARATIVO SQL SERVER VS MYSQL. AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO.	30
TABLA 6: DETALLE DE LA TABLA DE RIESGOS. AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO.....	49
TABLA 7: RIESGOS AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO	50
TABLA 8: INGRESO AL SISTEMA/IMPLEMENTACIÓN AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO	51
TABLA 9: INGRESO AL SISTEMA/PRUEBA USUARIO AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO.....	51
TABLA 10: INGRESO DE USUARIOS/PRUEBA USUARIO AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO.....	52
TABLA 11: INGRESO DE USUARIOS/PRUEBA USUARIO AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO	53
TABLA 12: INGRESO DE CLIENTES/IMPLEMENTACIÓN AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO	54
TABLA 13: INGRESO DE CLIENTES/PRUEBA USUARIO AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO	55
TABLA 14: INGRESO DE PROVEEDORES/IMPLEMENTACIÓN AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO	56
TABLA 15: INGRESO DE PROVEEDORES/PRUEBA USUARIO AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO	56
TABLA 16: INGRESO DE CATEGORÍAS/IMPLEMENTACIÓN AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO	57
TABLA 17: INGRESO DE CATEGORÍAS/PRUEBA USUARIO AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO	57
TABLA 18: INGRESO DE PRODUCTOS/IMPLEMENTACIÓN AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO	57
TABLA 19: INGRESO DE PRODUCTOS/PRUEBA USUARIO AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO	58
TABLA 20: INGRESO INVENTARIO /IMPLEMENTACIÓN AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO.....	58
TABLA 21: INGRESO VENTAS/PRUEBA USUARIO AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO	59
TABLA 22: INGRESO VENTAS/IMPLEMENTACIÓN AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO	60
TABLA 23: INGRESO VENTAS/PRUEBA USUARIO AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO	61
TABLA 24: GENERACIÓN FACTURA PDF AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO.....	61
TABLA 25: CREACIÓN ARCHIVO XML AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO	61
TABLA 26: COSTOS INICIALES AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO	62
TABLA 27: COSTOS ADMINISTRATIVOS AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO	62
TABLA 28: COSTOS OPERATIVOS AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO	62
TABLA 29: COSTOS SOPORTE TÉCNICO AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO	63
TABLA 30: RESUMEN COSTO DEL PROYECTO AUTOR: FERNANDA NICOLALDE Y DIEGO ROSERO	63

1. CAPÍTULO: FORMULACIÓN DEL PROBLEMA

1.1. Justificación

La empresa se dedica a la venta de productos tecnológicos por lo que necesita conocer los productos que tiene en stock es decir contar con la información real y confiable y así poder brindar un mejor servicio sin tener la necesidad de pedir una extensión para la entrega del producto, además es muy importante poder entregar facturación electrónica a los clientes.

1.2. Antecedentes

Actualmente la empresa Computer Solution Cía. Trabaja de manera que lleva el control administrativo en cuanto a adquisición de nueva mercadería mediante cuadros de Excel lo cual no es aconsejable, ya que en algunas ocasiones no se puede llevar un correcto control de inventario en cuanto a los objetos tecnológicos que la empresa tiene a su disposición, con lo cual es necesario un sistema para la automatización de pedidos y ventas de equipos tecnológicos, además la empresa necesita un sistema de facturación electrónica para el cumplimiento de entregas de facturas a proveedores y en sí al mismo estado Ecuatoriano.

1.3. Objetivos

1.3.1. General

Desarrollar un sistema de control de adquisición y venta de productos tecnológicos incluyendo un módulo de facturación electrónica.

1.3.2. Específicos

- Analizar los procesos de control de adquisición y venta de productos tecnológicos.
- Analizar el proceso de facturación electrónica.
- Diseñar y desarrollar un sistema para los procesos.
- Desarrollar el sistema de adquisición y venta, incluyendo módulo de facturación electrónica.
- Generar reportes finales de lo implementado.

1.4. Alcance

El trabajo culmina con el desarrollo del sistema de control de adquisición y venta de productos tecnológicos incluyendo la generación del código XML para el módulo de facturación electrónica.

2. CAPÍTULO: MARCO TEORICO.

2.1. Técnicas de Desarrollo de Aplicaciones

En las siguientes definiciones se presentará algunos tipos o técnicas de desarrollo de aplicaciones, mediante las cuales siguiendo los lineamientos del software a desarrollar se seleccionará una.

2.1.1. *Diseño Ágil de Aplicaciones (Extreme Programming)*

“La Programación Extrema o Extreme Programming es un enfoque de la Ingeniería de Software formulada por Kent Beck, Ron Jeffries y Ward Cunningham.

Es el más destacado de los procesos ágiles de desarrollo de software. La programación extrema se diferencia de las metodologías tradicionales principalmente en que brinda más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de Extreme Programming consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. La capacidad de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.” (Wikipedia, 2015)

“La simplicidad es la base de la programación extrema. Se simplifica el diseño para agilizar el desarrollo y facilitar el mantenimiento. Un diseño complejo del código junto a sucesivas modificaciones por parte de diferentes desarrolladores hace que la complejidad aumente exponencialmente.”

Para mantener la simplicidad es necesaria la reorganización del código, ésta es la manera de mantener el código simple a medida que crece.

También se aplica la simplicidad en la documentación. De esta manera, el código debe comentarse en su justa medida, intentando que el código esté autocomentado. Para ello se deben elegir adecuadamente los nombres de las variables, métodos y clases. Los nombres extensos no disminuyen la eficiencia del código ni el tiempo de desarrollo, gracias a las herramientas de autocompletado y reorganización que existen actualmente.

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

Aplicando la simplicidad junto con la autoría colectiva del código y la programación por parejas se asegura que cuanto más grande se haga el proyecto, todo el equipo conocerá más y mejor el sistema completo.” (Reyes, 2014)

“Al estar el cliente integrado en el proyecto, su opinión sobre el estado del proyecto se conoce en tiempo real.

Al realizarse ciclos muy cortos tras los cuales se muestran resultados, se minimiza el tener que rehacer partes que no cumplen con los requisitos y ayuda a los programadores a centrarse en lo que es más importante.

Considérense los problemas que derivan de tener ciclos muy largos. Meses de trabajo pueden tirarse por la borda debido a cambios en los criterios del cliente o malentendidos por parte del equipo de desarrollo. El código también es una fuente de retroalimentación gracias a las herramientas de desarrollo. Por ejemplo, las pruebas unitarias informan sobre el estado de salud del código. Ejecutar las pruebas unitarias frecuentemente permite descubrir fallos debido a cambios recientes en el código.” (Wikipedia, 2015)

Ventajas:

- Programación establecida.
- Menor tasa de errores.
- Bienestar del programador.

Desventajas:

- Es recomendable emplearlo solo en proyectos a corto plazo.
- Altas comisiones en caso de fallar.

2.1.2. Rad (Diseño Rápido de Aplicaciones)

“El método comprende el desarrollo interactivo, la construcción de prototipos y el uso de utilidades CASE (Computer Aided Software Engineering). Tradicionalmente, el desarrollo rápido de aplicaciones tiende a englobar también la usabilidad, utilidad y la rapidez de ejecución.

Hoy en día se suele utilizar para referirnos al desarrollo rápido de interfaces gráficas de usuario tales como Glade, o entornos de desarrollo integrado completos. Algunas de las plataformas más conocidas son Visual Studio, Lazarus, Gambas, Delphi, FoxPro, Anjuta, Game Maker, Velneo o Clarion. En el área de la autoría multimedia, software como

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

Neosoft Neoboo y MediaChance Multimedia Builder proveen plataformas de desarrollo rápido de aplicaciones, dentro de ciertos límites.

2.1.2.1. *Planificación de los Requisitos.*

Requiere que usuarios con un vasto conocimiento de los procesos de la compañía determinen cuáles serán las funciones del sistema. Debe darse una discusión estructurada sobre los problemas de la compañía que necesitan solución.

2.1.2.2. *Diseño.*

Se hace un análisis de las actividades de la compañía en relación al sistema propuesto. Los usuarios participan en talleres que descomponen funciones y definen entidades asociadas con el sistema.

2.1.2.3. *Construcción:*

Se afirman los requisitos, repasan los resultados y se hacen pruebas al sistema. También se crean las instrucciones, rutinas y procedimientos para operar la nueva aplicación.

2.1.2.4. *Implementación:*

La implementación y el manejo del cambio del viejo al nuevo sistema. Se hacen pruebas y se adiestran los usuarios. Los cambios organizacionales y la operación del nuevo sistema se hacen en paralelo con el viejo sistema hasta que el nuevo se establezca completamente.

2.1.2.5. *Modelado de Gestión:*

El flujo de información entre las funciones de gestión se modela de forma que responda a las siguientes preguntas: ¿Qué información conduce el proceso de gestión? ¿Qué información se genera?, ¿Quién la genera?, ¿A dónde va la información?, ¿Quién la procesa?

2.1.2.6. *Modelado de Datos:*

El flujo de información, definido como parte de la fase de modelado de gestión, se redefine como un conjunto de objetos de datos necesarios para apoyar la empresa.

2.1.2.7. *Modelado del Proceso:*

Los objetos de datos definidos en la fase de modelado de datos quedan transformados para lograr el flujo de información necesario para implementar una función de gestión.

2.1.2.8. *Generación de Aplicaciones.*

El proceso DRA trabaja para volver a utilizar componentes de programas ya existentes (cuando es posible) o a crear componentes reutilizables (cuando sea necesario). En todos los casos se utilizan herramientas para facilitar la construcción del software.” (Gervacio, 2013)

2.1.2.9. *Entorno RAD Sobre Navegadores Web*

Muchos entornos RAD permiten desarrollar interfaces de bases de datos basadas en navegadores web; que permiten acceder a la base de datos desde cualquier lugar a través de Internet, esto reduce los costos y permite a los usuarios externos tener acceso a una base de datos; sin embargo, las limitaciones propias de los navegadores web y los protocolos de Internet hacen a este enfoque no adecuado para sistemas donde se requieran respuestas interactivas muy rápidas.

Ventajas:

- “Comprar puede ahorrar dinero en comparación con construir.
- Los entregables pueden ser fácilmente trasladados a otra plataforma.
- El desarrollo se realiza a un nivel de abstracción mayor.
- Visibilidad temprana (Ingeniería de Software).
- Mayor flexibilidad.
- Menor codificación manual.
- Mayor involucramiento de los usuarios.
- Posiblemente menos fallas.
- Posiblemente menor costo.
- Ciclos de desarrollo más pequeños.
- Interfaz gráfica estándar.” (Galeano, 2013)

Desventajas:

- “Para proyectos en gran escala se requiere recursos humanos suficientes como para crear el número suficiente de equipos.
- Debe haber un compromiso fuerte entre todas las partes para completar el sistema en el tiempo necesario.
- No es adecuado cuando los riesgos técnicos son muy altos.” (Rodríguez, 2011)

2.1.3. JAD (Diseño de Aplicaciones por Juntas)

“Al elaborar un sistema de información, se sabe que se deben de seguir pasos y etapas para realizar software de calidad; dentro de la etapa de Análisis se ve la determinación de requerimientos, que se pueden obtener mediante diversas técnicas, entre ellas la entrevista, que es un método en el que se interactúa personalmente con los usuarios del sistema.

A través de la entrevista se puede estar consciente de las necesidades de los usuarios; proporciona además detalles que llegan a ser muy importantes para la creación del sistema, y hace que el entrevistado se dé cuenta que la empresa está interesada en su opinión y lo hace sentir parte de ella.

La entrevista no puede ser realizada por cualquier persona y de cualquier manera, hay pasos o procedimientos a seguir para lograr el éxito en la obtención de información clave para la realización del sistema. Así se observa que la persona encargada de realizar las entrevistas dentro de la empresa debe ser alguien afable, alguien con quien sea grato hablar pero que también sea experto en su campo, un analista que sepa obtener la información necesaria.

“Ésta es una técnica muy usada debido a sus grandes resultados porque se logra tener conexión con las partes, que son las que en realidad interactúan el sistema de la organización y que no se deben pasar por alto al analizar las opciones de obtención de datos. A pesar de los grandes resultados y el beneficio que trae la entrevista dentro de las empresas, también conlleva una gran desventaja, tiende a ser tardía, por el gran detalle que se le aplica a esta técnica; lleva mucho tiempo entrevistar a las personas que se cree nos darán la información necesaria, pues las entrevistas son de persona a persona.” (Camacho Suárez, Lagunes Collazo, & Reyes Pérez, 2002)

2.1.3.1. ¿Qué es JAD?

“Diseño Conjunto de Aplicaciones (JAD), “también conocido como el Desarrollo de Aplicaciones Comunes, es una técnica para determinar rápidamente los requerimientos del sistema mediante la obtención de aportes de una muestra representativa de las partes interesadas. Un equipo adecuado compuesto de los principales usuarios, administradores y analistas de sistemas se ensambla. Posteriormente, el equipo se reúne en una sesión intensiva para recopilar datos, una lluvia de ideas, debatir ideas, conciliar las diferencias, identificar y priorizar las necesidades y generar soluciones alternativas.” (Martinez, 2015)

“JAD también puede ser utilizado para realizar análisis de viabilidad, análisis de costo / beneficio y análisis de riesgo. A menudo, el diseño de especificaciones tales como diagramas de flujo de datos, diagramas entidad relación, y diagramas de flujo del sistema se generan durante el período de sesiones JAD.” (Vargas, 2011)

2.1.3.2. Instalaciones Para el Desarrollo de JAD

“Como mínimo, una sala de conferencias lo suficientemente grande como para acomodar a todos los miembros del equipo y están equipadas con pizarras o tableros, un proyector debe estar disponible. Con la aparición de los sistemas de reunión electrónica, el grupo de sistemas de soporte de decisiones, y la ingeniería de software asistida por ordenador, los requisitos adicionales pueden incluir equipos para llevar a cabo reuniones por medios electrónicos, servicios de tele-conferencia y una estación de maestros equipados con software.” (Martinez, 2015)

2.1.3.3. Realizar la Sesión JAD

“Una sesión JAD es un curso intensivo (por lo general) de dos o tres días de reunión de todo el equipo de JAD. Se espera que los miembros del equipo que están en período de sesiones JAD, ejerzan una atención completa para la programación de actividades, las actividades conflictivas no se encuentran dentro de la programación.” (Martinez, 2015)

2.1.3.4. Preparación

Antes del inicio de la sesión JAD, los analistas responsables de los sistemas de información dicen que los consultores deben:

- “Definir el alcance del sistema.

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

- Identificar los problemas, limitaciones y restricciones.
- Estimar las necesidades de recursos (tiempo, presupuesto, personal) para desarrollar el sistema.
- Identificar los gastos preliminares, beneficios, riesgos e impactos del proyecto.
- Identificar la naturaleza y los atributos principales del proyecto, las dependencias del proyecto y la interrelación del proyecto.
- Identificar los sub-proyectos apropiados. (El proyecto es a veces, descompuesto en varias sub-proyectos, debido a la oportunidad y / o las limitaciones presupuestarias.)
- Realizar el análisis de los antecedentes necesarios para definir parámetros fundamentales como el número de usuarios, el tamaño de la base de datos, el rendimiento requerido y la respuesta mínimo aceptable.
- Planificar la sesión JAD.

2.1.3.5. *El Período de Sesiones*

Una sesión JAD comienza con una descripción del material recogido durante la etapa de preparación. Una vez que los participantes entienden el problema, comienza el proceso de identificación de las dimensiones del problema, las posibles causas, necesidades y soluciones alternativas.

Durante una sesión JAD, es responsabilidad del moderador gestionar con eficacia el período de sesiones, para asegurar que el equipo se mantenga enfocado en los temas del programa, para alentar a todos los miembros del equipo a participar, y para resolver los conflictos generados durante el período de sesiones. Debido a que el equipo está compuesto en gran parte del personal no técnico, es importante que los analistas de los sistemas de información traten de minimizar el uso de términos técnicos.” (Martinez, 2015)

2.1.3.6. *Beneficios Principales*

- “Ahorro del tiempo en comparación con las entrevistas tradicionales persona a persona.
- Desarrollo rápido por medio del JAD (debido a que las entrevistas de usuarios no son realizadas seriamente a lo largo de semanas o meses, el desarrollo puede ser más rápido).
- Permite mejoras en el sistema de información, ya que su naturaleza interactiva y la alta visibilidad permiten que el JAD ayude a los usuarios a involucrarse desde el

principio en el proyecto del sistema y traten su realimentación seriamente.” (Vargas, 2011)

2.2. Análisis de Herramientas de Desarrollo

Se presenta el análisis y descripción de las herramientas que serán utilizadas en el proyecto:

2.2.1. Sistema de Gestión de Bases de Datos

“Colección de datos relacionados entre sí, estructurados y organizados, y un conjunto de programas que acceden y gestionan esos datos. La colección de estos datos se denomina Base de datos.

Antes de aparecer los SGBD, la información se trataba y se gestionaba utilizando los típicos sistemas de gestión de archivos que iban soportados sobre un sistema operativo. Estos consistían en un conjunto de programas que definen y trabajaban sus propios datos. Los datos se almacenan en archivos y los programas manejan esos archivos para obtener la información.

EL SGBD es una aplicación que permite a los usuarios definir, crear y mantener la BD y proporciona un acceso controlado a la misma.” (Ramos, Ramos, & Montero, 2006)

2.2.1.1. Servicios

- *“Creación y definición de la BD:* Especificación de la estructura, el tipo de los datos, las restricciones y relaciones entre ellos mediante lenguajes de definición de datos.
- *Manipulación de datos:* Realizando consultas, inserciones y actualizaciones de los mismos utilizando lenguajes de manipulación de datos.
- *Acceso controlado a los datos de la BD:* Mediante mecanismos de seguridad de acceso a los usuarios.
- *Mantener la integridad y consistencia:* Integridad y consistencia de los datos utilizando mecanismos para evitar que los datos sean perjudicados por cambios no autorizados.
- *Acceso compartido a la BD:* Controlando la interacción entre usuarios concurrentes.

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

- *Mecanismo de respaldo y recuperación:* Respaldo y recuperación para establecer la información en caso de fallos en el sistema.” (Ramos, Ramos, & Montero, 2006)

2.2.1.2. Arquitectura

- “*Nivel interno o físico:* el más cercano al almacenamiento físico, es decir tal y como están almacenados en el ordenador. Describe la estructura física de la BD mediante un esquema interno. Este esquema se especifica con un modelo físico y describe los detalles de cómo se almacena físicamente los datos.
- *Nivel externo o de visión:* es el más cercano a los usuarios, es decir es donde se describen varios esquemas externos o de vista de usuarios. Cada esquema describe la parte de la BD que interesa a un grupo de usuarios en este nivel se representa la visión individual de un usuario o de un grupo de usuarios.
- *Nivel Conceptual:* Describe la estructura de toda la DB para un grupo de usuarios mediante un esquema conceptual. Este esquema describe las entidades, atributos, relaciones, operaciones de los usuarios y restricciones, ocultando los detalles de las estructuras físicas de almacenamiento. Representa la información contenida en la BD.” (Israfevis, 2012)

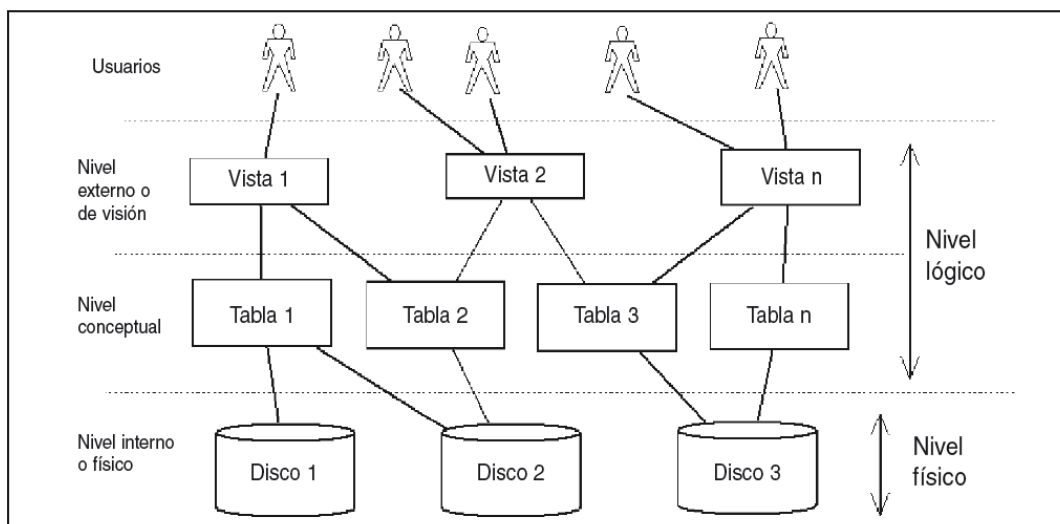


Ilustración 1: Niveles de abstracción de la arquitectura ANSI. (Ramos, Ramos, & Montero, 2006)

2.2.1.3. SQL Server

“SQL server 2014, al igual que sus predecesores, es más que un motor de base de datos. Es una colección de los componentes que se pueden implementar por separado o como un grupo para formar una solución escalable, plataforma de información lista en la nube. En términos generales, esta plataforma está diseñada para dos propósitos: para ayudar a administrar los datos y para ayudar a ofrecer inteligencia de negocio.” (Mistry & Misner, 2014)

2.2.1.3.1. Ventajas

- *“Rendimiento confiable:* Sql Server 2014 agiliza las aplicaciones críticas con un nuevo motor OLTP in-memory que proporcionan un rendimiento transaccional hasta de 30 veces superior. En cuanto al almacenamiento de datos, el nuevo almacén de columnas in-memory actualizable ofrece una velocidad de procesamiento de consultas 100 veces superior a la de las soluciones antiguas. SQL Server transmite una gran confianza, ya que ofrece una seguridad sin parangón y se ha considerado la base de datos empresarial con menos vulnerabilidades durante seis años seguidos.
- *Mayor rapidez en la obtención de la información privilegiada que subyace en datos de cualquier tipo:* Obtenga información privilegiada más rápido con una plataforma BI completa que agiliza las operaciones de acceso, análisis, limpieza y formato de datos internos y externos. Con SQL Server 2014 y Microsoft Power BI es muy sencillo conectar a cada usuario de la organización con los datos correctos que necesita para tomar decisiones más inteligentes y rápidas.
- *Plataforma para la nube híbrida:* Se ha diseñado para trabajar en entornos híbridos, ya sea de manera local o en la nube. Las nuevas herramientas de SQL Server y de Microsoft Azure facilitan aún más la creación de soluciones de aplicación de revisiones, de recuperación ante desastres y las copias de seguridad con Microsoft Azure.” (Microsoft, 2015)

2.2.2. Lenguajes De Programación

2.2.2.1. .NET

“Microsoft .NET es un conjunto de tecnologías de software para conectar personas, sistemas, información y dispositivos. Se basa en los servicios Web XML, de modo que es posible conectar aplicaciones nuevas y existentes con software y servicios en diversos sistemas operativos, programas y lenguajes de programación.” (Information technology services, 2014)

“Microsoft .NET es un conjunto de tecnologías de software, compuesto de varios lenguajes de programación que se ejecutan bajo el .NET Framework. Es además un entorno completamente orientado a objetos y que es capaz de ejecutarse bajo cualquier plataforma.” (Mesa, 2008)

“Podríamos dividir el entorno .NET en las siguientes partes:

- .NET Framework, que es el entorno de trabajo de la plataforma .NET y que la engloba completamente. Toda la plataforma .NET forma parte de .NET framework.
- Lenguajes .NET. Destacan C# y VB.NET, y recientemente J#, aunque existen más lenguajes y con toda seguridad surgirán nuevos.
- El Common Runtime Language CRL, que es el motor de ejecución común a todos los lenguajes .NET.
- MSIL, Microsoft Intermedial language, es el lenguaje intermedio al que compilan las aplicaciones (Asemmblies) .NET. Este lenguaje intermedio es interpretado por el CRL en tiempo de ejecución.
- CLS, Common Language Specification, que engloban las pautas que deben cumplir los lenguajes .NET. Es está característica la que va a permitir a otras compañías producir lenguajes compatibles con .NET.
- ADO.NET, es la nueva interfaz de bases de datos. No se trata de una evolución de ADO, sino que se trata de una interfaz completamente nueva.
- ASP.NET, es la nueva tecnología para páginas web dinámicas completamente integrada dentro del entorno .NET. Representa una auténtica revolución en el desarrollo Web (Internet e Intranet).

- Biblioteca de clases .NET, es el conjunto de clases que componen el .NET framework y que permiten realizar casi cualquier tarea de una manera fácil y rápida.” (Information technology services, 2014)

2.2.2.2. Java

“El significado de java, tal y como se le conoce en la actualidad, es el lenguaje de programación y un entorno de ejecución de programas escritos en java. Al contrario de los compiladores tradicionales, que convierten el código fuente en instrucciones a nivel de máquina, el compilador java traduce el código fuente java en instrucciones que son interpretadas por la máquina virtual de java (JVM, Java Virtual Machine). A diferencia de C y C++ en los que está inspirado. Java es un lenguaje

Características

Java es un lenguaje interpretado. Cuando se escriben programas en java, bien en un entorno de desarrollo o un editor de texto necesita ser compilado en un conjunto de instrucciones optimizadas denominadas programas “bytecode”. Este programa es independiente de la plataforma y no se puede ejecutar directamente por procesador. En su lugar, una máquina virtual java ejecuta (interpreta) los bytecode. Existen numerosas JVM disponibles para una gran variedad de plataformas que permiten a los programas Java ser independientes de la plataforma.” (Cruz, 2015)

2.2.2.1. XML

“XML es un subconjunto de SGML (Estándar Generalised Mark-up Language), simplificado y adaptado a Internet. Es un estándar internacionalmente conocido, no pertenece a ninguna compañía y permite una utilización efectiva en Internet para sus diferentes terminales.

Especificación para diseñar lenguajes de marcado, que permite definir etiquetas personalizadas para descripción y organización de datos. Sirve para representar información estructurada en la web (todos documentos), de modo que esta información pueda ser almacenada, transmitida, procesada, visualizada e impresa, por muy diversos tipos de aplicaciones y dispositivos.

Ventajas

- Fácilmente procesable.

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

- Separa radicalmente el contenido y el formato de presentación.
- Diseñado para cualquier lenguaje y alfabeto. (encoding)

Características

- XML es un subconjunto de SGML que incorpora las tres características más importantes de este:
 - Extensibilidad
 - Estructura
 - Validación
- Basado en texto.
- Orientado a los contenidos no presentación.
- Las etiquetas se definen para crear los documentos, no tienen un significado preestablecido.
- No es sustituto de HTML.
- No existe un visor genérico de XML.

2.2.2.1.1. Documento XML

Es un conjunto de datos con sus respectivas etiquetas de marcado XML, el cual se almacena como texto en archivo con extensión .xml. Un documento XML puede incluir cualquier flujo de datos basado en texto: un artículo de una revista, un resumen de cotizaciones de bolsa, un conjunto de registros de una base de datos, etc.

Estructura de un documento XML

Un documento XML está formado por **datos de caracteres** y **marcado**, el marcado lo forman las etiquetas.” (Exces Consultoria, 2013)

```
Prologo { <?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
          <! DOCTYPE persona SYSTEM "persona.dtd"
Cuerpo { <persona>
          <nombre>Luis</nombre>
          <apellidos>Pérez</apellidos>
          </persona>
```

Ilustración 2: <http://www.mundolinux.info/index.htm>

Estructura



2.2.2.1.2. Componentes de un documento XML

“En un documento XML existen los siguientes componentes:

- Elementos: Pieza lógica del marcado, se representa con una cadena de texto(dato) encerrada entre etiquetas.
Pueden existir elementos vacíos (
). Los elementos pueden contener atributos.
- Instrucciones: Ordenes especiales para ser utilizadas por la aplicación que procesa
- `<?xml-stylesheet type="text/css" href="estilo.css">`
- Las instrucciones XML. Comienzan por `<?` Y terminan por `?>`.
- Comentarios: Información que no forma parte del documento. Comienzan por `<!--` y terminan por `-->`.
- Declaraciones de tipo: Especifican información acerca del documento:
- `<!DOCTYPE persona SYSTEM "persona.dtd">`
- Secciones CDATA: Se trata de un conjunto de caracteres que no deben ser interpretados por el procesador:
- `<![CDATA[Aquí se puede meter cualquier carácter, como <, &, >, ... Sin que sean interpretados como marcación]]>` (Exces Consultoria, 2013)

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

EJEMPLO

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE peliculas SYSTEM "Peliculas.dtd">

<peliculas>
  <pelicula tipo="comedia" sistema="PG-13" ejemplares="5" año="1987">
    <titulo>Raising Arizona</titulo>
    <guionista>Ethan Coen</guionista>
    <guionista>Joel Coen</guionista>
    <productor>Ethan Coen</productor>
    <director>Joel Coen</director>
    <actor>Nicolas Cage</actor>
    <actor>Holly Hunter</actor>
    <actor>John Goodman</actor>
    <comentarios>Una pelicula clasica de Comedia.</comentarios>
  </pelicula>

  <pelicula tipo="ciencia-ficcion" sistema="PG-13" ejemplares="4" año="1989">
    <titulo>The Abyss</titulo>
    <guionista>James Cameron</guionista>
    <productor>Gale Anne Hurd</productor>
    <director>James Cameron</director>
    <actor>Ed Harris</actor>
    <actor>Mary Elizabeth Mastrantonio</actor>
    <comentarios>Una buena pelicula.</comentarios>
  </pelicula>
</peliculas>
```

Ilustración 4: Ejemplo XML <http://www.mundolinux.info/index.htm>

2.3. Arquitectura De Aplicaciones

2.3.1. Por Capas

Al realizar sistemas en un solo lineamiento en muchas de las ocasiones presentan problemas para las empresas cuando estos sistemas necesitan o deben integrarse con otros, los problemas más comunes son escalabilidad, seguridad, integración.

Para ayudar a resolver dichos problemas se presenta la elaboración de sistemas en capas, las cuales en la mayoría de veces son 3: capa de presentación, capa negocio y capa de datos.

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

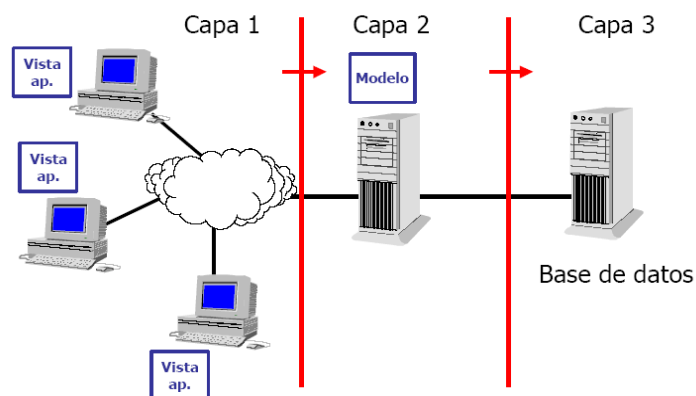


Ilustración 5: Modelo 3 Capas Referencia: <http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html>

2.3.1.1. Capa de presentación

“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.” (Za, 2015)

“También es conocida como interfaz gráfica y debe tener la característica de ser entendible y fácil de usar para el usuario. Esta capa se comunica únicamente con la capa de negocio.

2.3.1.2. Capa de negocio

Es donde residen los programas que se ejecutan, 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 almacenar o recuperar datos de él. También, se consideran aquí los programas de aplicación.

2.3.1.3. Capa de datos

Es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.” (Wikipedia, 2015)

3. CAPÍTULO: SITUACION DE LA EMPRESA.

3.1. Proceso de Solicitud y Entrega de Productos Tecnológicos en la Empresa Computer Solution.

Este capítulo se basará en algunos puntos tomados de la problemática del proyecto y el estudio de herramientas para la solución del mismo.

3.1.1. Objetivos

3.1.1.1. General

Mantener un stock para cumplir los pedidos del cliente y generando el archivo XML para la elaboración de la factura electrónica.

3.1.1.2. Específicos

- Llevar un control de los productos que se tienen en bodega.
- Entregar a tiempo los pedidos al cliente.
- Elaborar de manera correcta el archivo XML.

3.1.2. Misión

Ser una empresa que haga de las necesidades sociales y empresariales soluciones tecnológicas que contribuyan con la evolución del ser humano a través de la actualización de equipos tecnológicos, ofreciendo soluciones integrales con la finalidad de satisfacer a los clientes en la venta de equipos de última generación, calidad y presencia en el mercado laboral.

3.1.3. Visión

Ser una empresa de reconocido prestigio, con autonomía administrativa, excelencia en ventas de computadoras, donde se brinde un producto de calidad, de eficiente gestión, competitiva, comprometida con el servicio al consumidor y la formación integral de su recurso humano y tecnológico.

3.1.4. Cuadro Jerárquico.

El cuadro indica perfiles y tareas dentro de la Empresa Computer Solution.

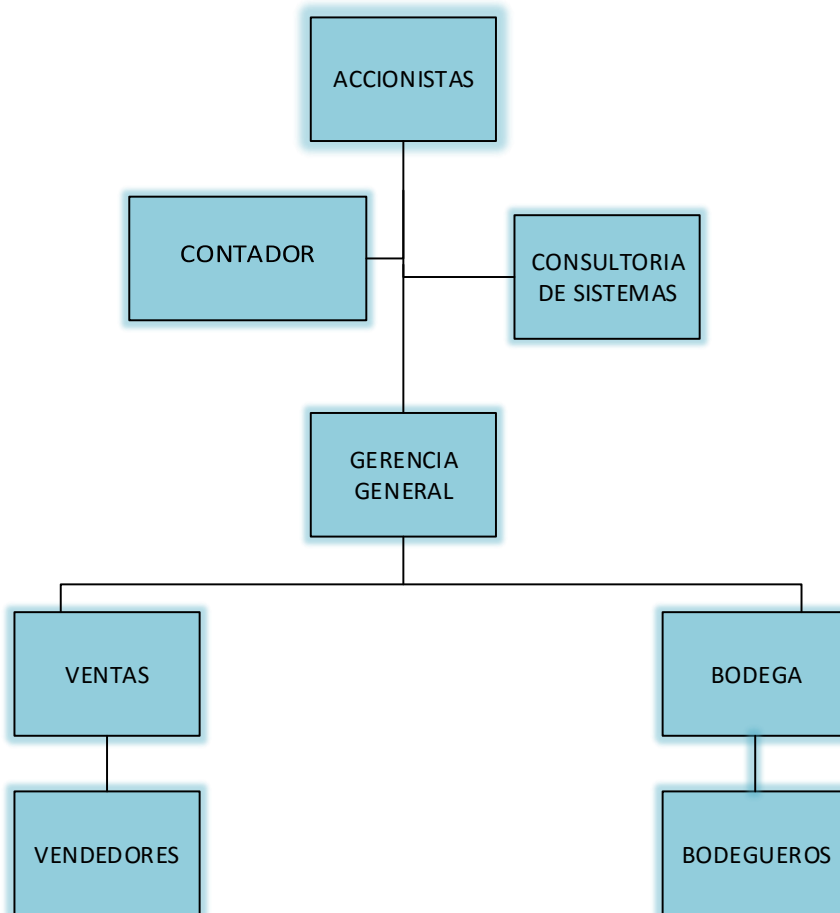


Ilustración 6: Cuadro Jerárquico. Autor: Fernanda Nicolalde y Diego Rosero.

3.2. Mapa de Procesos

3.2.1. Descripción

El sistema tiene varias personas (actores) que administrarán el sistema dependiendo su cargo, por ende, se describen los diferentes macroprocesos a desarrollar.

3.2.2. Actores

- Administrador.
- Bodeguero.
- Vendedor.

3.2.3. Mapa de Procesos de la Empresa Computer Solution.

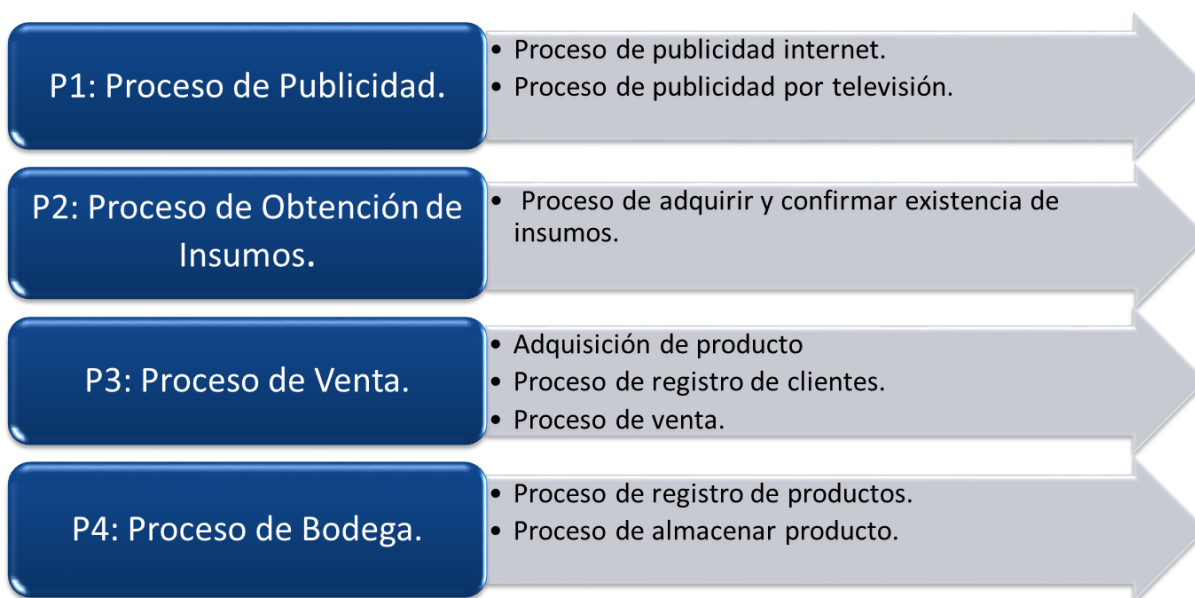


Ilustración 7: Mapa de Procesos. Autor: Fernanda Nicolalde y Diego Rosero.

En virtud de esta información general de los procesos que la empresa tiene, se va implementar los procesos de adquisición, venta y bodega.

3.2.4. Actividades Principales del proceso de adquisición, venta de productos tecnológicos y generación de factura.

Actividades que se realizan en el proceso de administración de control de adquisición y venta de productos tecnológicos:

3.2.4.1. Proceso de Adquisición de productos tecnológicos.

1. Buscar proveedor para adquisición de productos tecnológicos
2. Realizar inventario del stock de productos tecnológicos.
3. Realizar pedido de productos tecnológicos.
4. Verificar adquisición de productos.
5. Registrar productos tecnológicos adquiridos.

3.2.4.2. Proceso de Venta de productos tecnológicos.

1. Costear productos
2. Verificar stock de productos tecnológicos.
3. Realizar venta de productos tecnológicos.
4. Crear orden de compra para entrega de producto.
5. Entregar la venta de productos tecnológicos.

3.2.4.3. Proceso de Generación de Facturas.

1. Realizar ingreso de datos del cliente.
2. Verificar detalle de la venta de productos tecnológicos.
3. Generar archivo de Factura sobre la venta.
4. Quitar de inventario

3.2.4.4. Proceso Selección de Proveedor

1. Buscar proveedores con los mejores precios.
2. Cotizar precios de productos tecnológicos.
3. Seleccionar proveedor con precios más accesibles.
4. Determinar métodos de pago al proveedor.
5. Adquirir productos tecnológicos.

3.2.5. Problemas del Proceso Anterior

Los problemas que se tienen en los procesos es que, al momento de generar una venta, no se tiene actualizado el stock de productos, por lo cual muchos clientes han quedado insatisfechos con el servicio.

En cuanto al stock de productos la información generada, no se actualiza al momento de solicitar el pedido al proveedor por lo que la información de productos se duplicaba y generaba gastos innecesarios, siendo una información no confiable.

Al momento de generar facturas el proceso llevaba mucho tiempo porque se lo realiza manualmente.

3.2.6. Cadenas de Valor

Se muestra los diferentes procesos que se van a llevar a cabo en el desarrollo de este sistema.



Ilustración 8: Cadena De Valor. Autor: Fernanda Nicolalde y Diego Rosero.

Macroprocesos y procesos



Ilustración 9: Macroprocesos y procesos. Autor: Fernanda Nicolalde y Diego Rosero.

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

3.2.7. Mapa de procesos.

A continuación, se presenta el mapa de procesos:

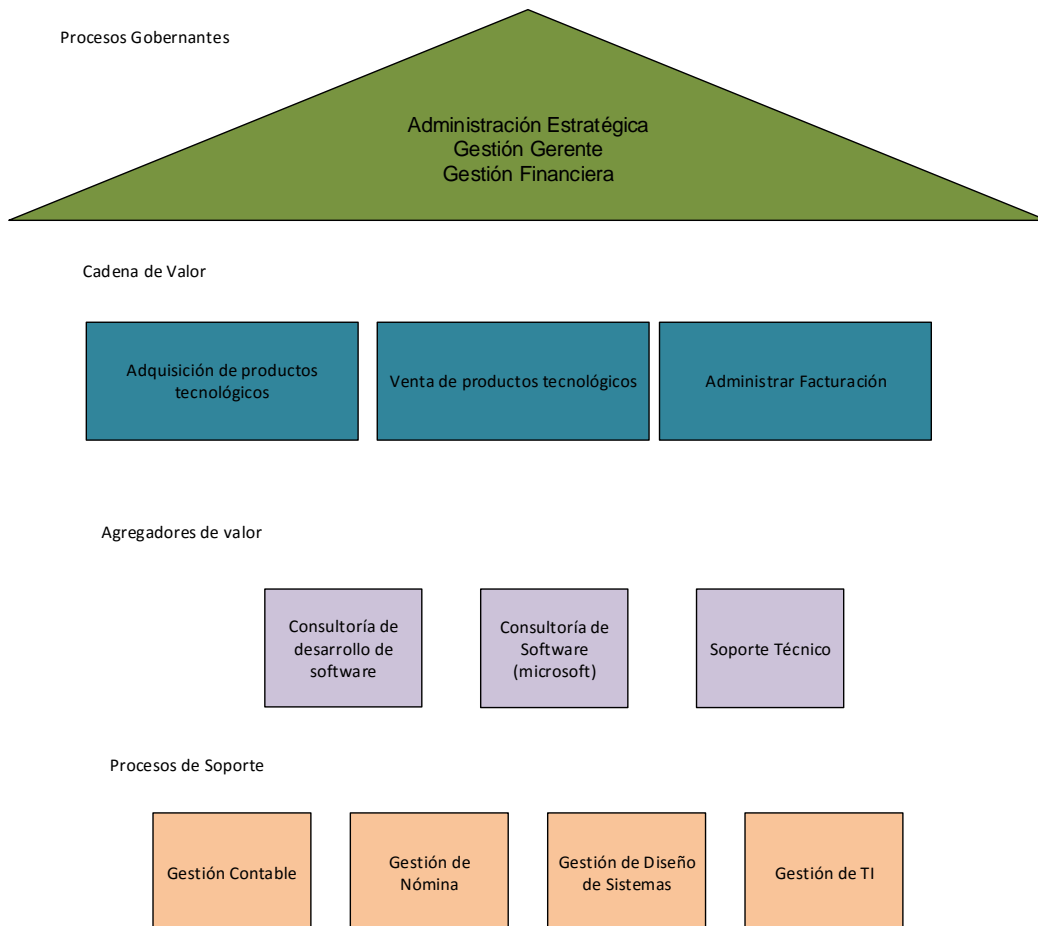


Ilustración 10: Mapa De Procesos. Autor: Fernanda Nicolalde y Diego Rosero

3.2.1. Gráficos de Soporte.

3.2.1.1. Tabla Proceso Administrar Productos.

Nombre:	Productos
Administrar	
Propósito	<ul style="list-style-type: none"> Automatizar el ingreso y salida de productos para llevar un inventario actualizado.
Entradas	<ul style="list-style-type: none"> Código del producto Datos del usuario Datos del producto
Salidas	<ul style="list-style-type: none"> Informe de productos
Controles	<ul style="list-style-type: none"> Productos en buen estado
Mecanismos	<ul style="list-style-type: none"> Recursos humanos, tecnología, bodega
Actividades	<ul style="list-style-type: none"> Levantar el listado de productos existentes en bodega
Unidad Organizacional	<ul style="list-style-type: none"> Departamento de adquisición

Tabla 1: Proceso Administrar Productos. Autor: Fernanda Nicolalde y Diego Rosero.

3.2.1.2. Gráfico Proceso Administrar Venta.

Nombre:	Administrar	Ventas
Propósito	<ul style="list-style-type: none"> Automatizar la venta de productos para brindar un mejor servicio. 	
Entradas	<ul style="list-style-type: none"> Código del producto Datos del cliente Datos del producto Precio del producto 	
Salidas	<ul style="list-style-type: none"> Venta de productos Utilidad 	
Controles	<ul style="list-style-type: none"> Stock de producto 	
Mecanismos	<ul style="list-style-type: none"> Recursos humanos, tecnología, administración 	
Actividades	<ul style="list-style-type: none"> Venta de los diferentes productos existentes en el stock 	
Unidad Organizacional	<ul style="list-style-type: none"> Departamento de ventas 	

Tabla 2: Proceso Administrar Ventas. Autor: Fernanda Nicolalde y Diego Rosero.

3.2.1.3. Administrar Generación de XML

Nombre:	Administrar	Generación de XML
Propósito		<ul style="list-style-type: none"> • Generar archivo XML para la factura electrónica.
Entradas		<ul style="list-style-type: none"> • Código del producto • Datos del cliente • Datos del producto
Salidas		<ul style="list-style-type: none"> • Generación de archivo XML.
Controles		<ul style="list-style-type: none"> • Stock de Producto • Verificar datos.
Mecanismos		<ul style="list-style-type: none"> • Tecnología
Actividades		<ul style="list-style-type: none"> • Creación de archivo XML para facturación electrónica
Unidad Organizacional		<ul style="list-style-type: none"> • Departamento de Ventas

Tabla 3: Proceso Administrar Generación de XML. Autor: Fernanda Nicolalde y Diego Rosero.

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

3.1. Diagrama de flujo con metodología SIPOC

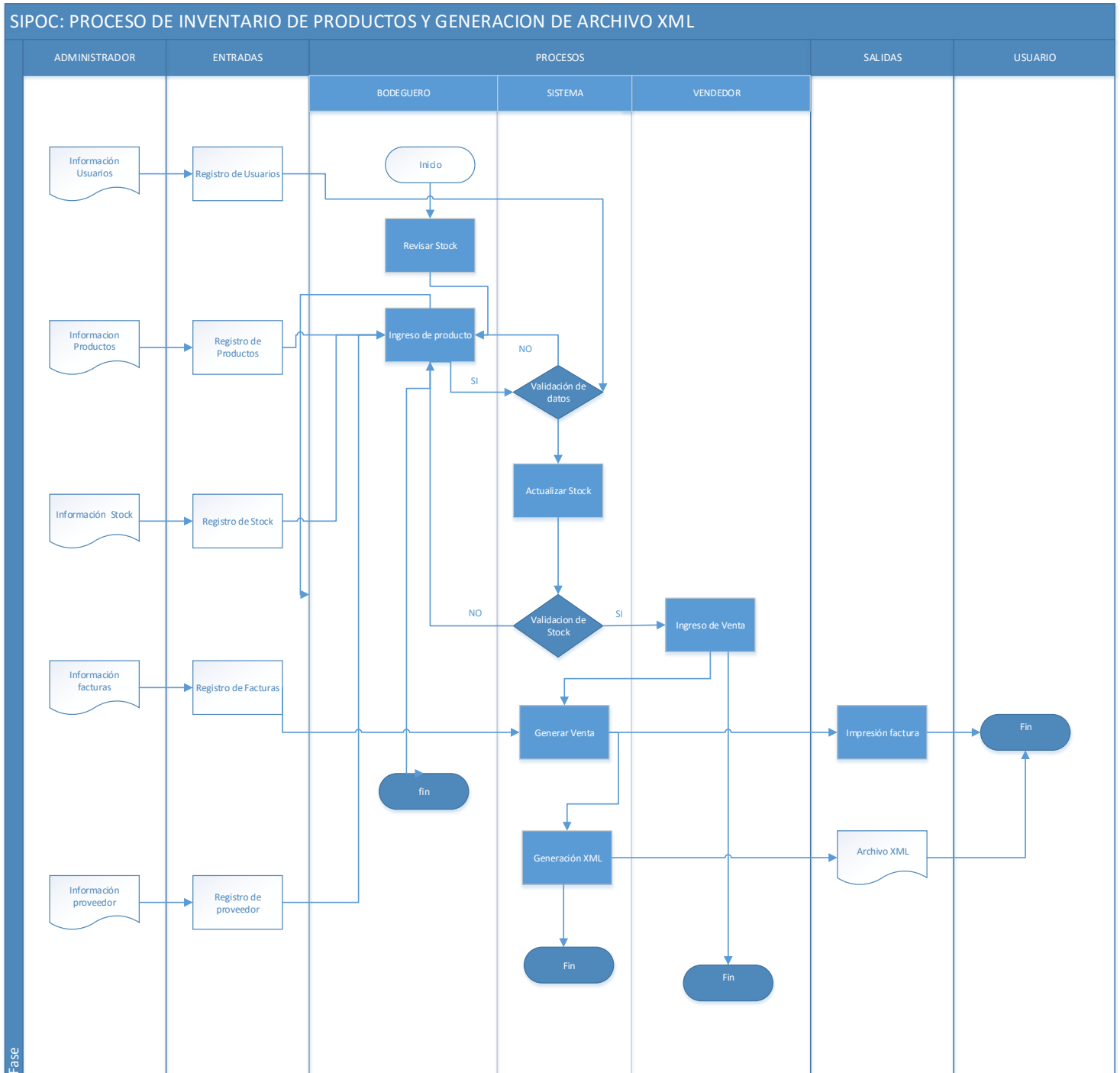


Ilustración 11: Mapa De Procesos. Autor: Fernanda Nicolalde y Diego Rosero

4. CAPÍTULO: HERRAMIENTAS PARA EL DESARROLLO DEL SISTEMA

Se debe considerar algunas herramientas para desarrollar este proyecto, se ha medido y comprobado su eficiencia con la selección individual de las herramientas que aportarán al desarrollo del proyecto permitiendo la construcción y elaboración del mismo.

4.1. Técnica de Desarrollo de Aplicaciones.

EL método para desarrollar el sistema se lo realizará mediante RAD (Diseño Rápido de Aplicaciones), descrito en el anterior capítulo, la eficiencia del método RAD radica en el acercamiento y el aporte que tiene el cliente para la construcción del aplicativo.

La metodología RAD está enfocado con la participación activa del cliente mientras que Diseño Ágil (Extreme Programming) contiene menos contacto, y la misma tiene menos contacto que JAD (Diseño de Aplicaciones por Juntas), ésta sirve para trabajar en el desarrollo de grandes proyectos con un gran número de procesos.

Esta técnica permite hacer una planificación a corto plazo para la elaboración del prototipo y la entrega a tiempo del producto.

4.2. Arquitectura De Aplicación.

A pesar de no terminar nuestro sistema en un aplicativo web trabajaremos en el esquema de 3 capas ya que esto nos permitirá desarrollar el sistema de una manera más fácil, permitiendo hacer cambios en el nivel requerido, las capas a trabajar son las siguientes: la capa de datos (sistema gestor de base de datos), la capa del negocio (la interacción entre la capa de presentación y capa de datos) y la capa de presentación.

Para facilitar el desarrollo del aplicativo se utilizará el framework .Net, facilitando el trabajo con el lenguaje de programación C#.

4.3. Lenguaje De Programación.

Una vez analizado los requerimientos de la empresa y la infraestructura con la que trabaja se pudo llegar a un acuerdo de selección del lenguaje con el que se realizará el desarrollo del sistema, a pesar de que existen otras alternativas para desarrollar.

A continuación, se presentará la comparación de uno de lenguajes existentes en el mercado como opción de desarrollo.

4.3.1. C# vs JAVA

En el siguiente cuadro se muestran las ventajas más básicas de cada lenguaje de programación.

	QUE ES	VENTAJA	DESVENTAJA
C#	Lenguaje orientado a objetos, desarrollado por Microsoft	Variedad de tipo de datos	Versiones anteriores de programas no puede ser ejecutados en version actuales
	Diseñado para compilar aplicaciones ejecutadas en .NET	Tiene atributos para cada clase según su prioridad	sistema operativo requiere de requerimientos minimos para ejecutar
	Simple, eficaz y seguro	(publico, protegido, interno, interno protegido y privado)	
JAVA	Lenguaje orientado a objetos, propiedad de Oracle	Desarrollo de applet	Menos eficiente que C/C++
	Maneja algunas plataformas de desarrollo	Desarrollo de aplicaciones que se ejecutan dentro de una página HTML	Mala implementacion de un aplicativo en java, demora su ejecución
		Aplicaciones de escrito, se ejecutan independientemente	

Tabla 4: Cuadro Comparativo C# y JAVA. Autor: Fernanda Nicolalde y Diego Rosero.

En conclusión, debido a que C# es un lenguaje simple, eficaz, seguro y nos permite desarrollar el aplicativo para la empresa, se decidió trabajar en el mismo.

4.4. Sistema Gestor de Base de Datos.

Para un óptimo acople de la base de datos con el lenguaje de programación a utilizar, nos hemos orientado en la compatibilidad que estos programas nos brindan, en consecuencia, se ha optado por la herramienta Visual Studio la cual interactúa de manera adecuada con SQL Server ya que los dos provienen de la misma casa de desarrollo (Microsoft).

A continuación, se realizará una comparativa entre dos motores de bases de datos.

4.4.1. SQL Server vs MySQL

SQL Server y MySQL se determinan por ser motores de bases de datos muy potentes dentro del mercado y de la misma forma en manejo de grandes volúmenes de datos, por esta razón realizaremos una comparativa simple entre los dos; por otra parte, SQL Server trabaja de manera estable con el lenguaje Visual Studio por la razón de ser de la misma casa de desarrollo como ya se lo mencionó anteriormente.

En el siguiente cuadro se muestran las ventajas de los dos motores de bases de datos, mediante las cuales se llegará a una conclusión de que motor es el más aceptable para el gestionamiento de los datos del sistema.

	QUE ES	VENTAJA	DESVENTAJA
SQL SERVER	Sistema de manejo de base de datos desarrollado por Microsoft	Utiliza una extensión al SQL estandar, que se denomina Transact SQL	Gran cantidad de memoria RAM utilizada para la instalación y utilización
	Permite crear tablas, definir relaciones entre las mismas	Sistema de gestor de base de Datos relacionales	Restricciones de versionamientos
	Para el desarrollo de aplicaciones más complejas (tres o más capas) incluye interfaces de acceso para varias plataformas de desarrollo como es .NET.	Permite administrar permisos a todo como es nivel de servidor, seguridad de tablas, tiene varias configuraciones	
MYSQL	Gestor de base de datos relacional, multihilo y multiusuario	Menor necesidad de limpieza de las memorias internas durante el procesamiento de las transacciones	Limite de tamaño de la base de datos según el sistema operativo
	Utilizado en aplicaciones web.	Es muy rápida, fiable y facil de usar	Restricciones de versionamientos
	Tiene una lectura rápida cuando utiliza el motor no transaccional.	Gran portabilidad entre distintos sistemas o plataformas Uso de MyODBC, permite realizar conexiones ODBC.	Herremientas no documenadas Lento con grandes Dases de Datos

Tabla 5: Cuadro Comparativo SQL Server vs MYSQL. Autor: Fernanda Nicolalde y Diego Rosero.

Observamos la comparación de los tres gestores de bases de datos más utilizados

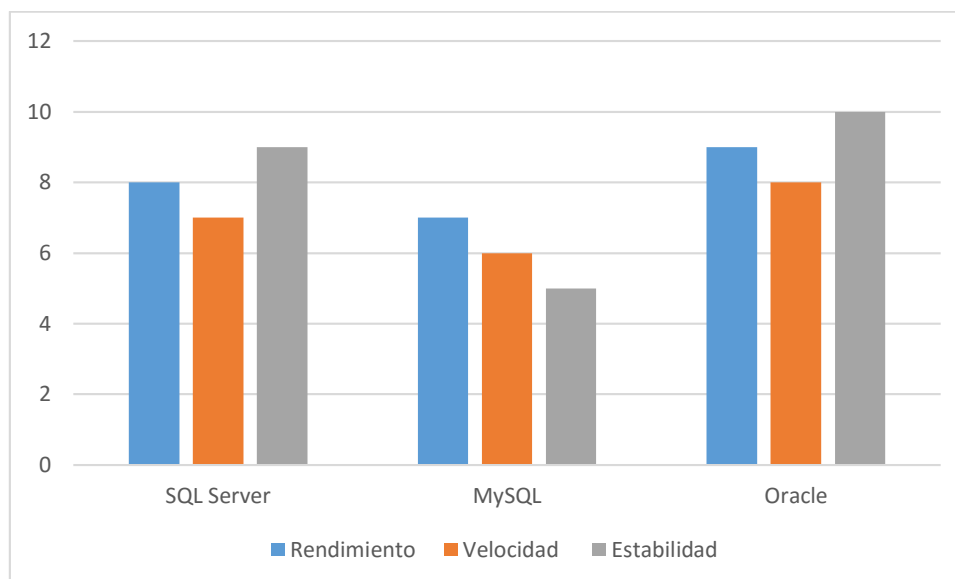


Ilustración 12: Comparación Gestores Bases de Datos. Autor: Fernanda Nicolalde y Diego Rosero.

Con la gráfica anterior concluimos, de acuerdo al desarrollo que se va a realizar del aplicativo, el cual es sobre la plataforma Microsoft se necesita obtener estabilidad, calidad en tiempo de respuesta y volatilidad en los datos que el motor de base de datos gestiona, con esta información el motor de base de datos que se elegiría es SQL Server, sin embargo hacemos notar que la plataforma de Oracle supera las características de Sql Server pero por su alto costo tanto en software como en hardware se toma la decisión seleccionar a SQL Server para el manejo de datos .

4.5. IDE Para Desarrollo.

En las herramientas actuales “un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, o sea, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.” (EcuREd, 2012)

En el mercado de desarrollo de software existen muchos IDEs los cuales ayudan para el desarrollo de aplicaciones y ambientes web, estos son seleccionados en algunos casos por su uso cotidiano y también por el ranking de popularidad en nuestro caso se ha seleccionado a Visual Studio, con esto lo que se busca es tener un mejor acoplamiento entre el trabajo a desarrollarse en el IDE y el Framework .Net de Microsoft.

“Microsoft Visual Studio 2015 es una suite de herramientas para la creación de software, desde la fase de planificación a través del diseño de interfaz de usuario, codificación,

pruebas, depuración, análisis de la calidad del código y el rendimiento, la implementación de los clientes, y la recolección de la telemetría del uso. Estas herramientas están diseñadas para trabajar en conjunto la mayor fluidez posible, y están expuestos a través del entorno de Visual Studio de desarrollo integrado (IDE).” (Microsoft, 2015)

4.6. Archivo XML.

Para la creación del documento XML se lo realiza en el IDE Visual Studio, con el fin de no anexar algo externo (componentes que no pertenezcan al IDE), se realizó una consulta extensa para el desarrollo del archivo lo cual fue favorable ya que solo se tuvo que anexar una biblioteca XML para la creación de este.

La biblioteca que utilizamos para la creación fue: using System.XML; esta biblioteca nos ayudó en la creación y estructuración, basándose en la parametrización y datos que el documento XML o plantilla que la empresa Computer Solution envía a una tercera que es la encargada de realizar la validación y encriptación de la Factura XML con la llave otorgada por el SRI.

5. CAPÍTULO: CONSTRUCCIÓN DEL PRODUCTO

5.1. Inicio

En este capítulo se presentarán los diagramas que serán necesarios para el desarrollo del sistema, esto nos permitirán tener un conocimiento general de la funcionalidad del mismo.

5.2. Objetivos de la Aplicación

- Objetivo General: Desarrollar un sistema de control de adquisición y venta de productos tecnológicos incluyendo un módulo de facturación electrónica.

- Objetivos Específicos:
 - Analizar los procesos de control de adquisición y venta de productos tecnológicos.
 - Analizar el proceso de facturación electrónica.
 - Diseñar y desarrollar un sistema para atender los procesos requeridos.
 - Desarrollar el sistema de adquisición y venta, incluyendo módulo de facturación electrónica.
 - Generar reportes finales de lo implementado.

5.2.1. Análisis De Requerimientos

En el levantamiento de requerimientos podremos conocer las especificaciones necesarias que tiene el cliente para desarrollar el sistema, a continuación, mostramos los principales requerimientos:

- Definición de actores.
- Especificación de funcionalidades.
- Arquitectura.
- Diseño.

5.2.2. Definición De Actores.

El sistema contará con 3 actores específicos que manejaran el sistema con sus respectivas restricciones.

Administrador: persona encargada de administrar la totalidad del sistema.

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

Bodeguero: responsable del despacho de productos y el ingreso de los mismos

Vendedor: quien realizará las ventas y emitirá la factura.

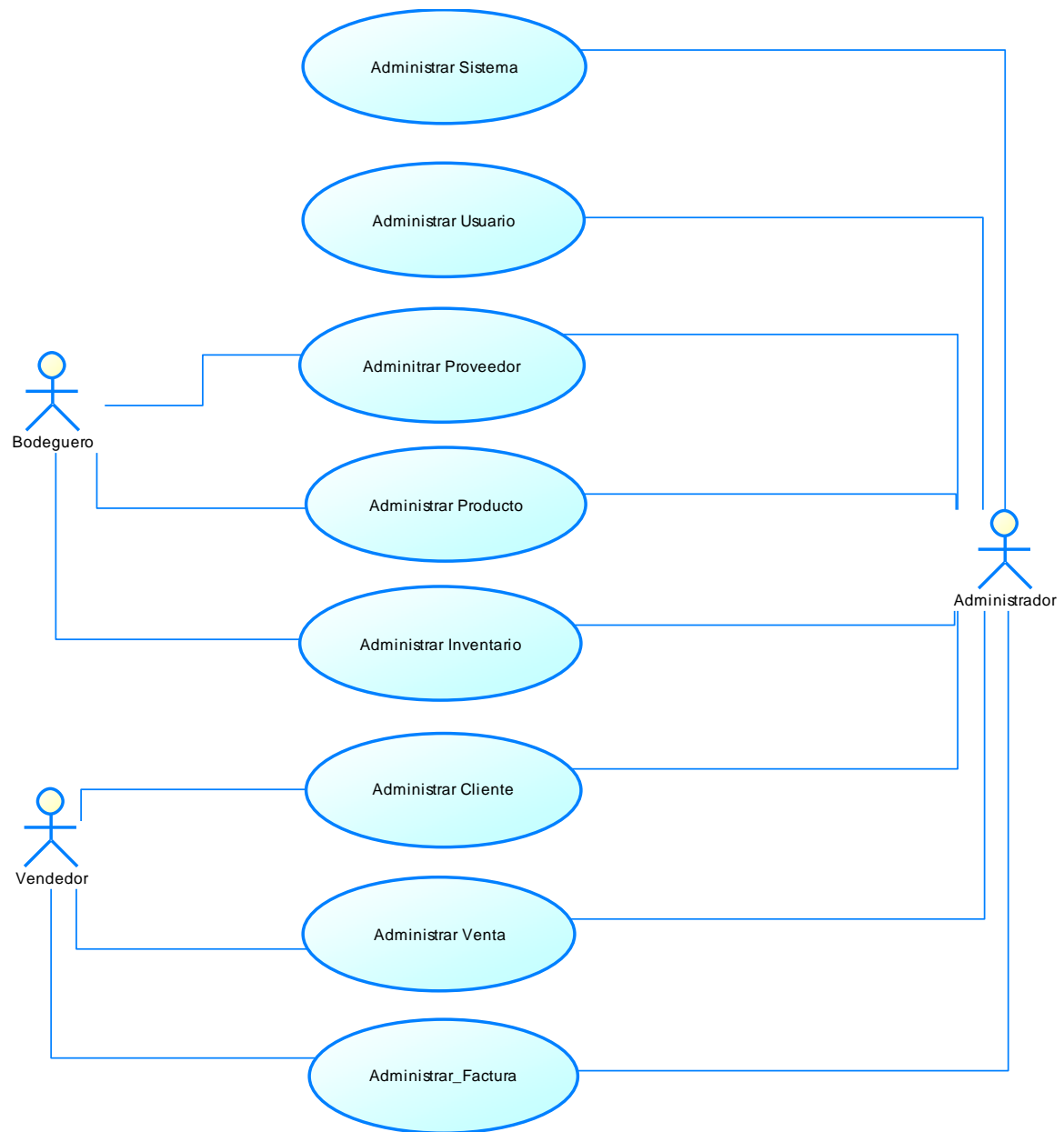


Ilustración 13: Actores. Autor: Fernanda Nicolalde y Diego Rosero.

5.2.3. Especificaciones De Requerimientos

5.2.3.1. Funcionales

- El sistema administra el control de productos tecnológicos de la empresa.
- El sistema presenta reportes mensuales
- El sistema administrara el control de insumos y víveres dentro de un barco turístico.
- El sistema generara ordenes de pedidos para la requisición de productos próximos a agotarse o agotados en bodega (inventario).
- El sistema registrara un stock de productos.
- El sistema disminuye productos del stock dependiendo las ventas generadas.
- El sistema dará a conocer la cantidad real de productos existentes en stock.

5.2.3.2. No Funcionales

Formato de archivos

Formato de C#: (.sln)

Lenguajes

C#

Base de datos

SQL Server 2014

Compatibilidad

Sistema Operativo: Windows 8, Windows 10.

Hardware: Memoria RAM: 256 MB, Procesador: Core Duo

Servidor

Hardware: Memoria RAM: 256 MB, Espacio en Disco: 1 GB, Procesador: Núcleo XEON

Sistema Operativo: Windows Server 2012

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

Atributos

- **Seguridad**
- ESETSMART SECURITY 8
- **Mantenimiento**
- Se dará mantenimiento cada 6 meses al sistema.
- **Conversión**
- Conforme el sistema se expanda se realizarán modificaciones es decir en el aumento de la base de datos y en las diferentes administraciones.

5.2.4. Arquitectura

El sistema será construido bajo la arquitectura de tres capas con un ambiente de aplicación de escritorio.



Ilustración 14: Arquitectura 3 Capas. Autor: Fernanda Nicolalde y Diego Rosero

5.2.5. Diseño

5.2.5.1. Diseño Entidad/Relación (E/R).

Se mostrada el diseño de Entidad/Relación para entender el funcionamiento del sistema

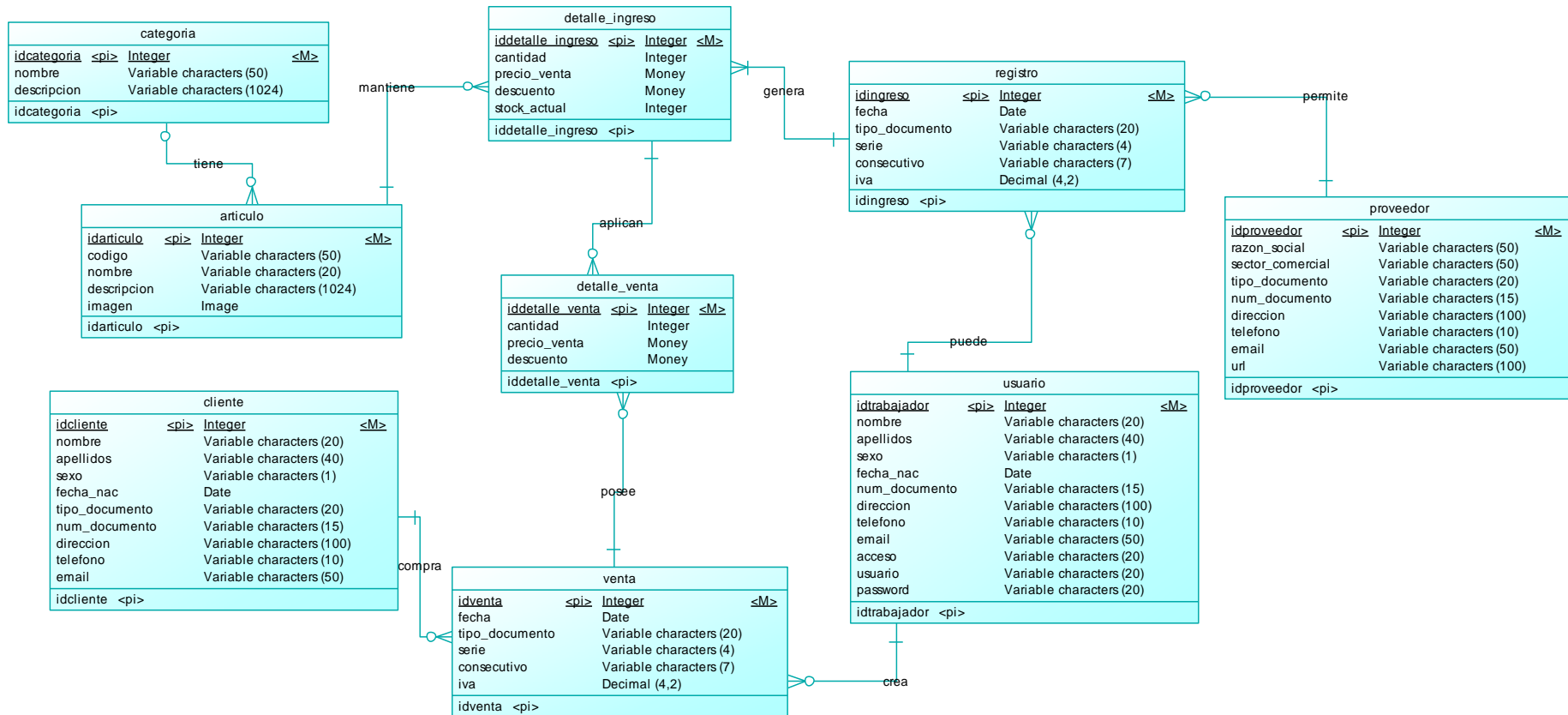


Ilustración 15: Diseño E/R. Autor: Fernanda Nicolalde y Diego Rosero.

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

5.2.5.2. Diagramas UML

Clase DP.

En esta imagen podemos ver las acciones que tendrá esta clase con los diferentes métodos definidos.

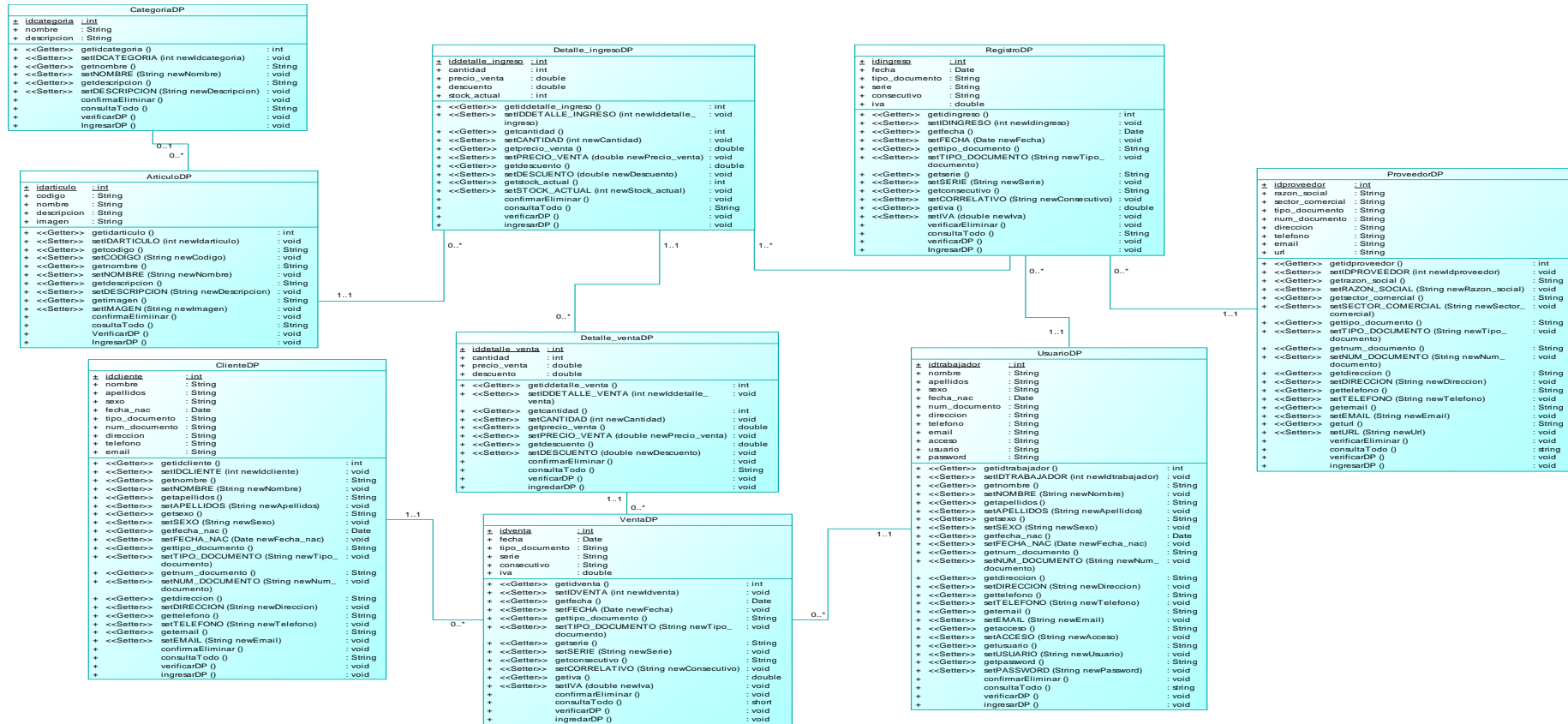


Ilustración 16: Diagrama DP. Autor: Fernanda Nicolalde y Diego Rosero.

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

Clase MD.

Podemos entender la relación que tiene el manejo de datos y la base de datos directamente

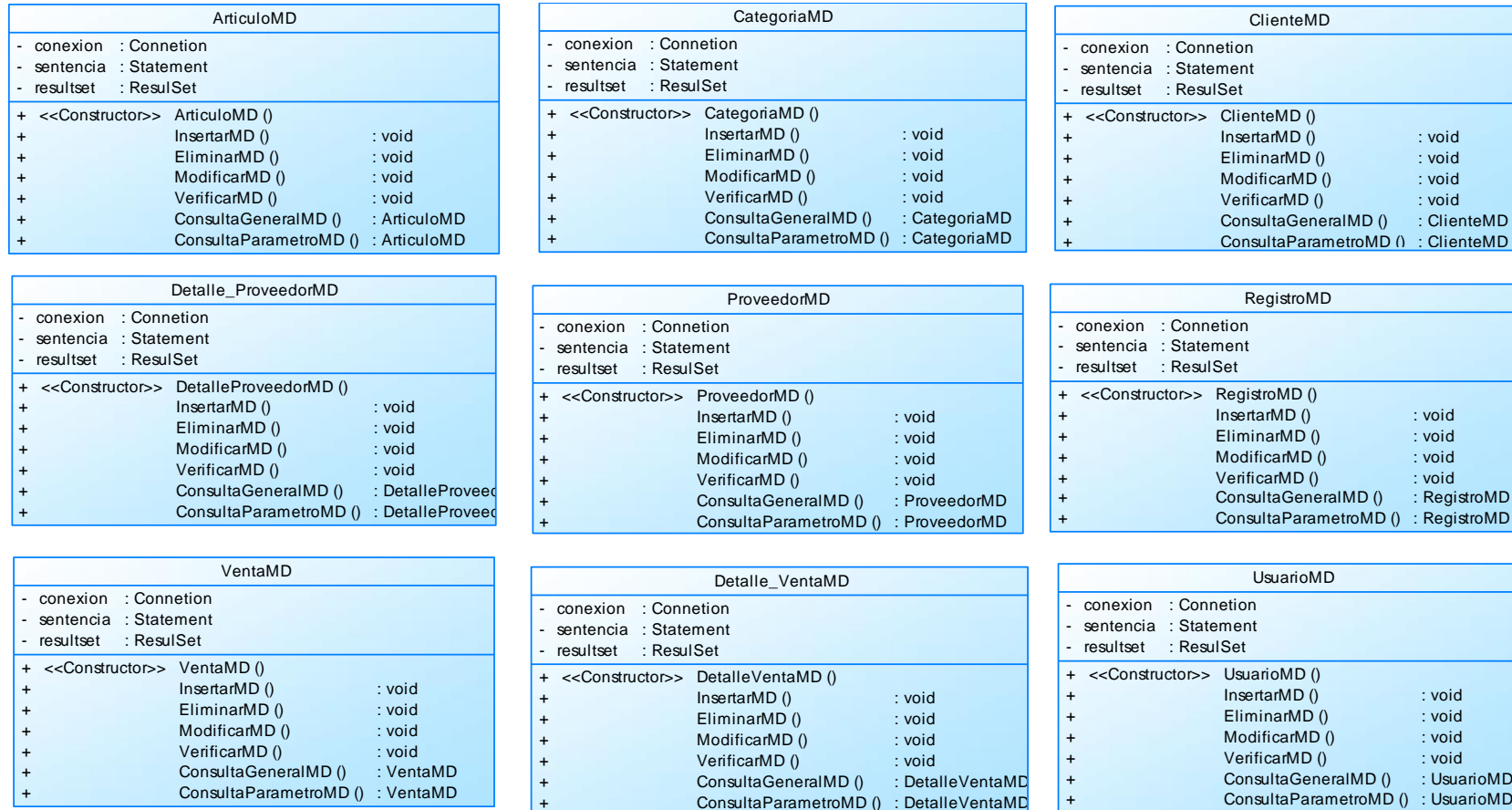


Ilustración 17: Diagrama MD. Autor: Fernanda Nicolalde y Diego Rosero

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

Clase GUI.

En el diagrama se puede comprender la manera en que esta empaquetado el GUI (Graphical User Interface) así como el despliegue de estas interfaces dependientes de una interfaz superior.

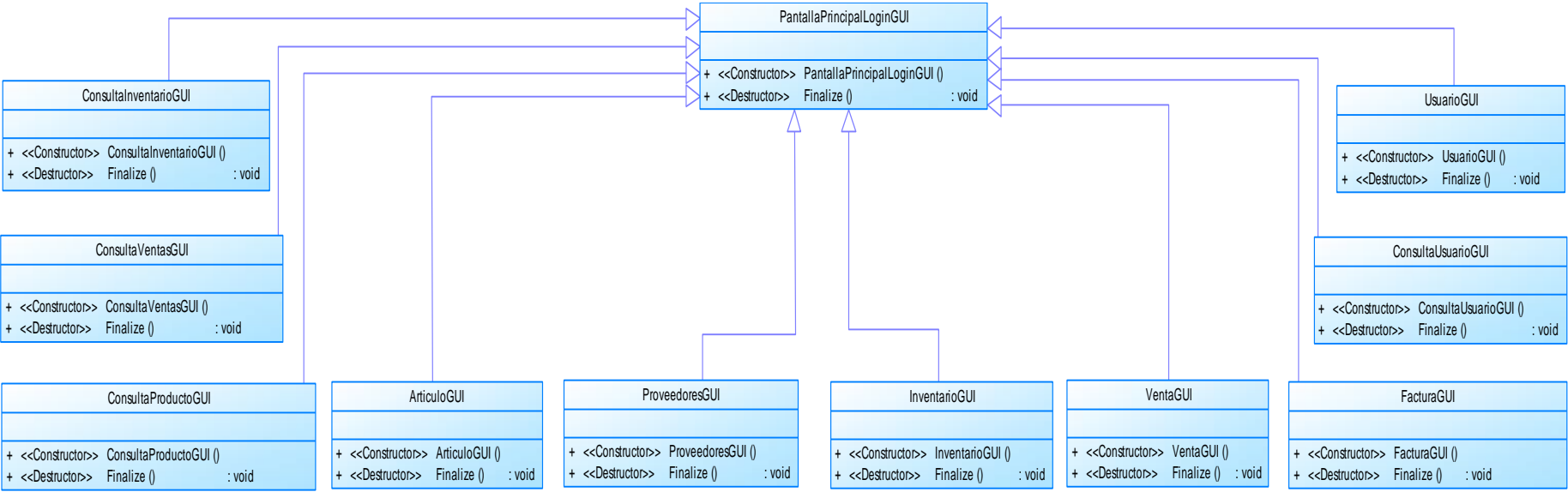


Ilustración 18: Diagrama GUI. Autor: Fernanda Nicolalde y Diego Rosero

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

Diagramas de Despliegue:

En el diagrama se puede comprender la cantidad de servidores que se presentan en el sistema y su interacción entre sí para el manejo de los procesos.

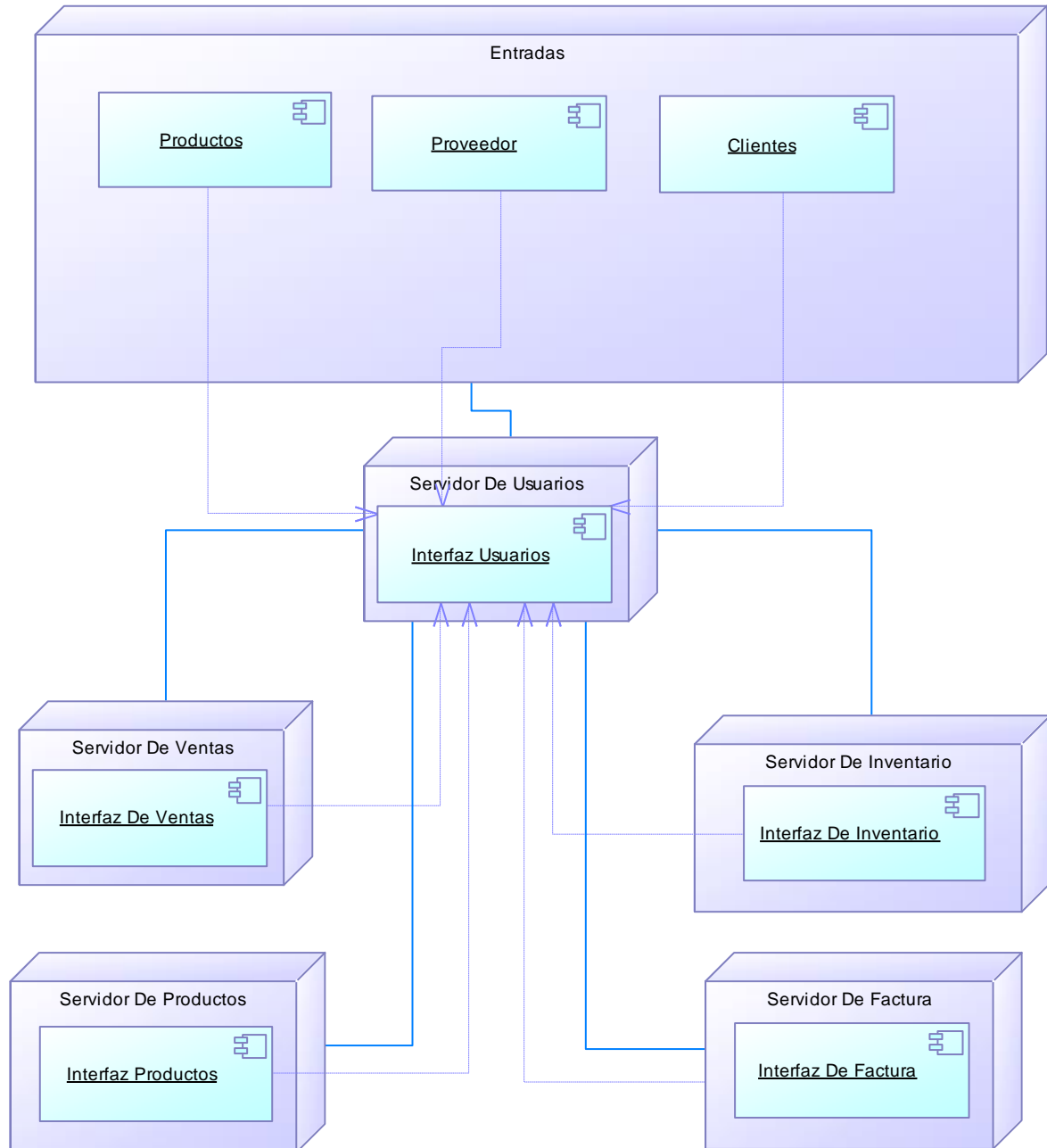


Ilustración 19: Diagrama de Despliegue. Autor: Autor: Fernanda Nicolalde y Diego Rosero.

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

Diagramas de Secuencia:

Proceso: Administración de Productos (Ingreso Productos)

En el diagrama se observa el proceso para el ingreso de los productos al sistema.

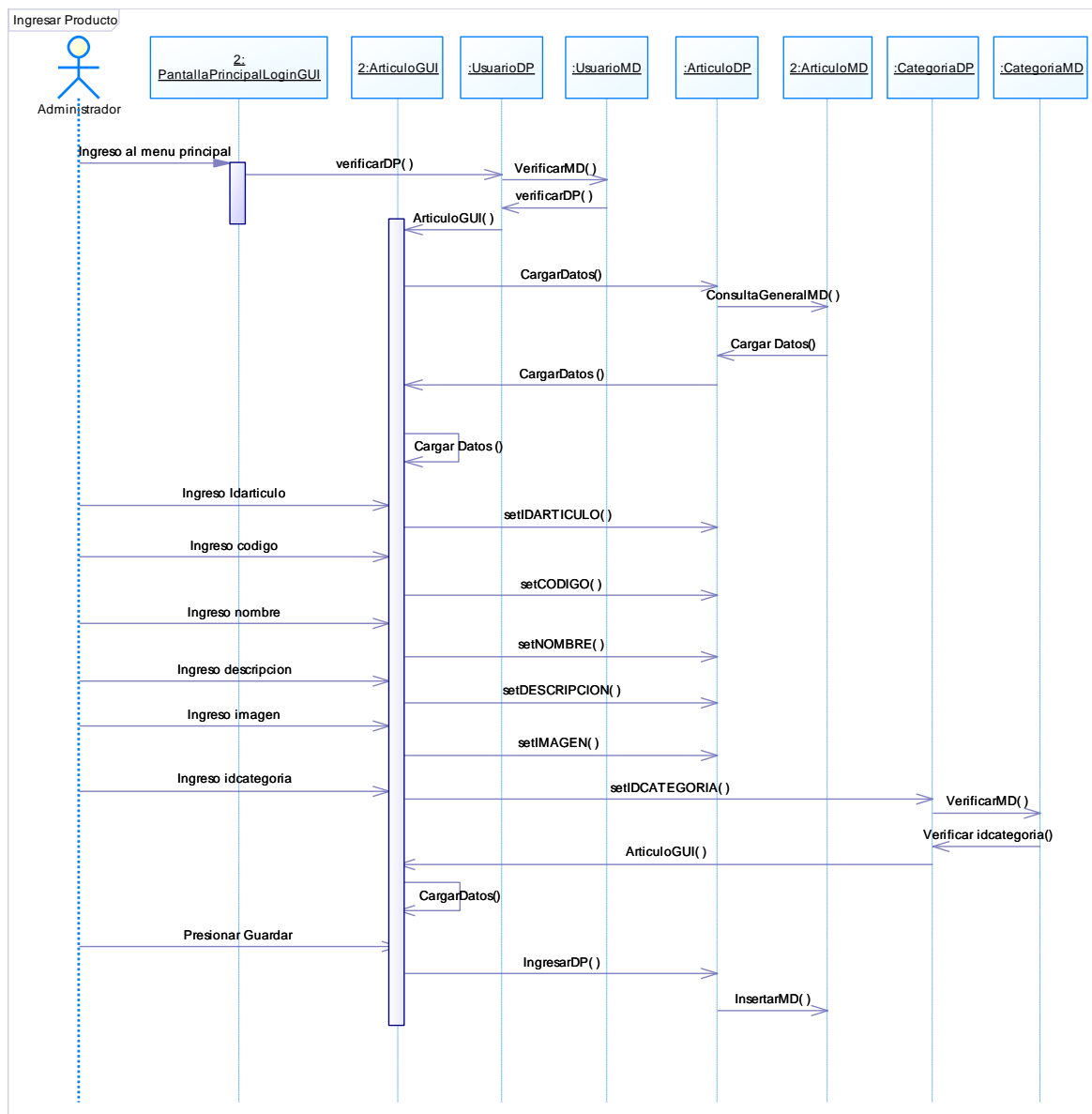


Ilustración 20: Ingreso de Productos. Autor: Fernanda Nicolalde y Diego Rosero.

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

Proceso: Administración de Productos (Modificar de Productos)

En el diagrama se observa el proceso para modificar productos en el sistema.

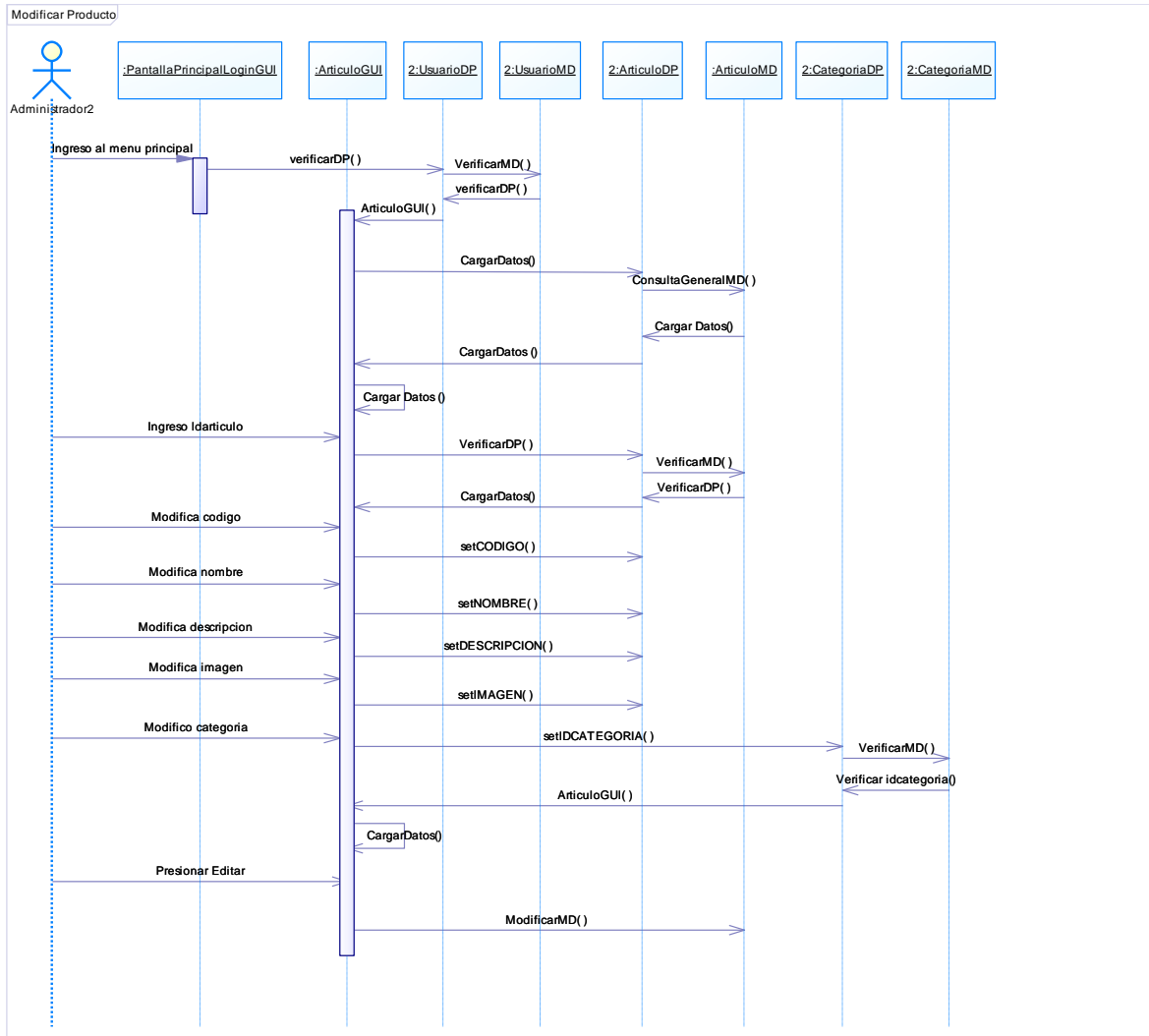


Ilustración 21: Modificar Producto. Autor: Fernanda Nicolalde y Diego Rosero.

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

Proceso: Administración de Productos (Eliminar Productos)

En el diagrama se observa el proceso para eliminar productos del sistema.

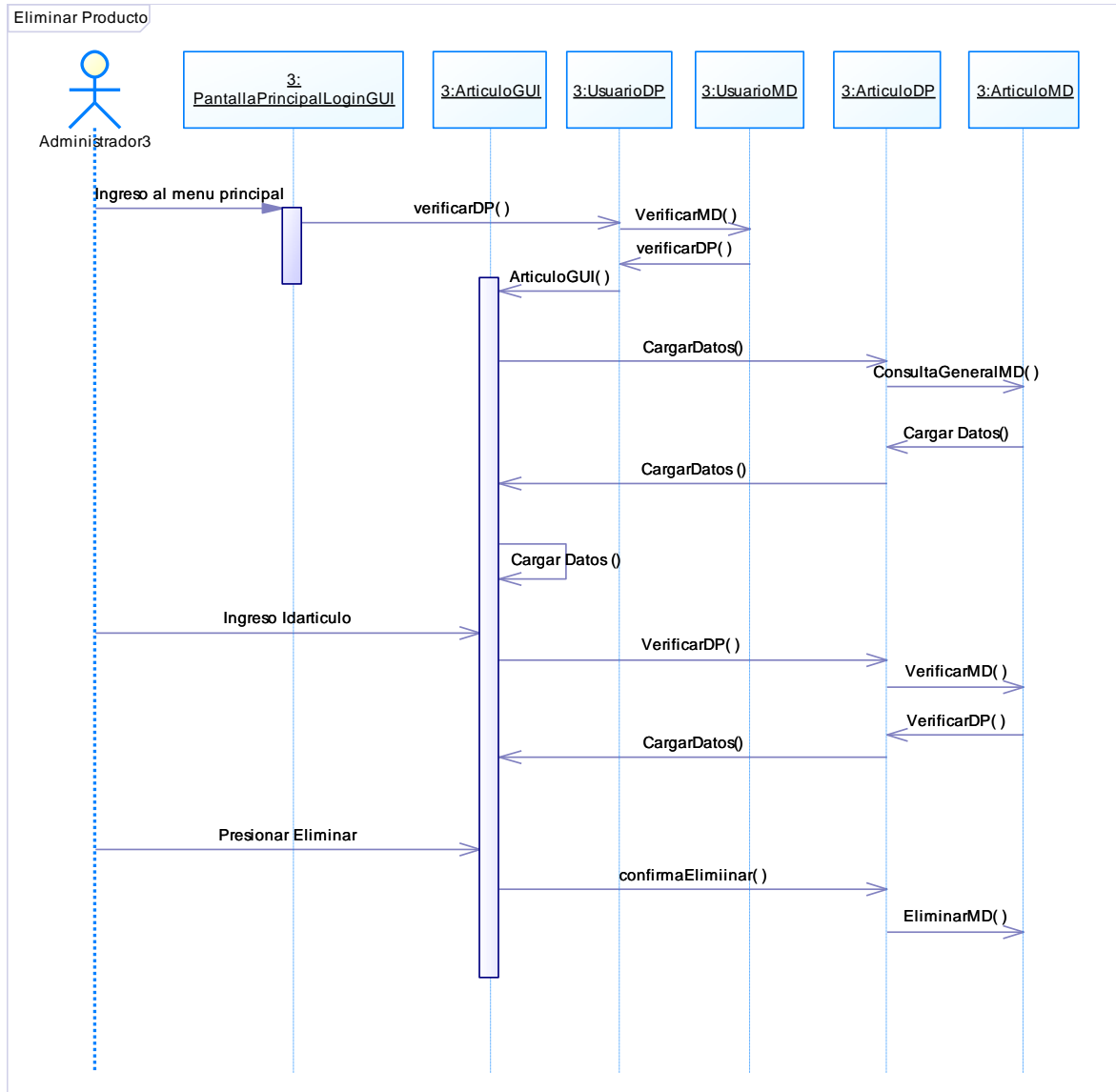


Ilustración 22: Eliminar Producto. Autor: Fernanda Nicolalde y Diego Rosero

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

Proceso: Administración de Productos (Consulta General)

En el diagrama se observa el proceso para eliminar productos del sistema.

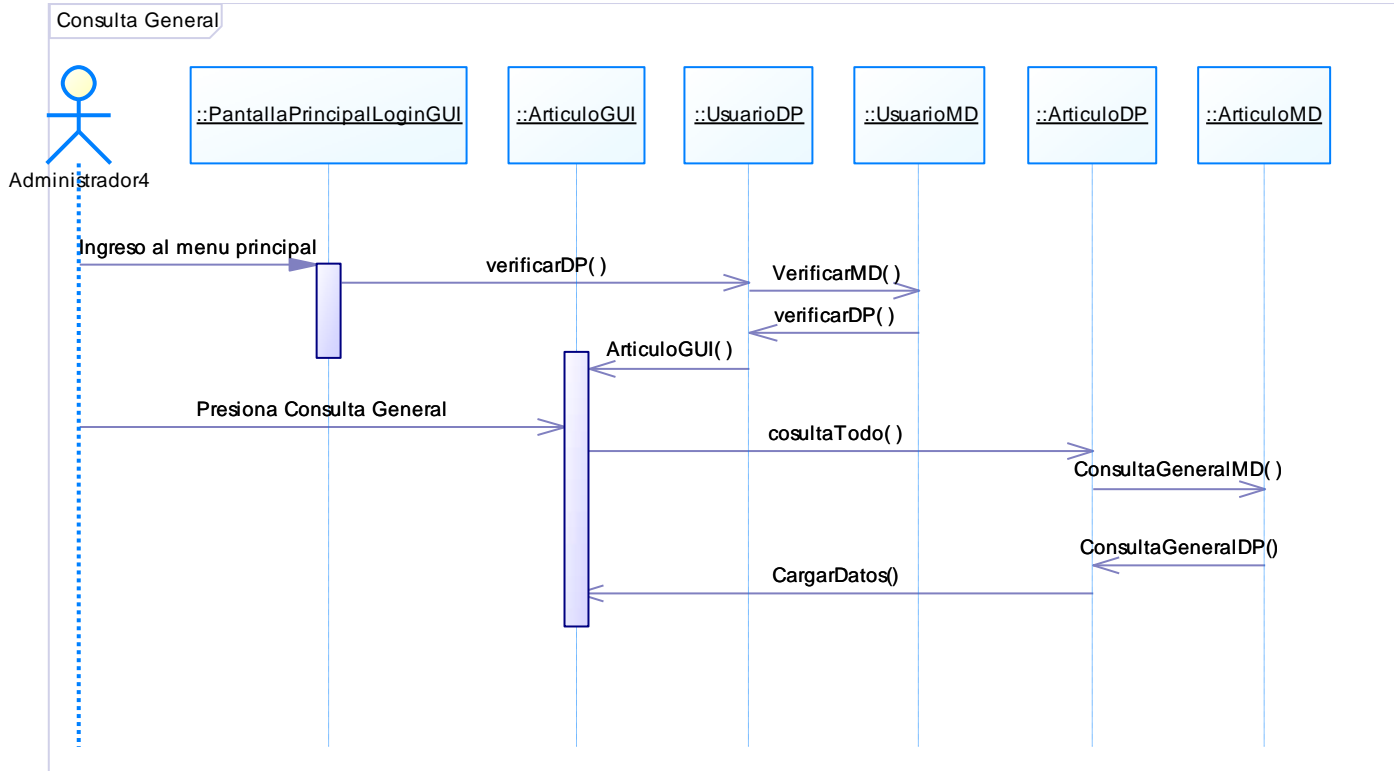


Ilustración 23: Consulta General Productos. Autor: Fernanda Nicolalde y Diego Rosero.

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

Proceso: Administración de Productos (Consulta por Parámetros)

En el diagrama se observa el proceso para consultar por un parámetro en el sistema

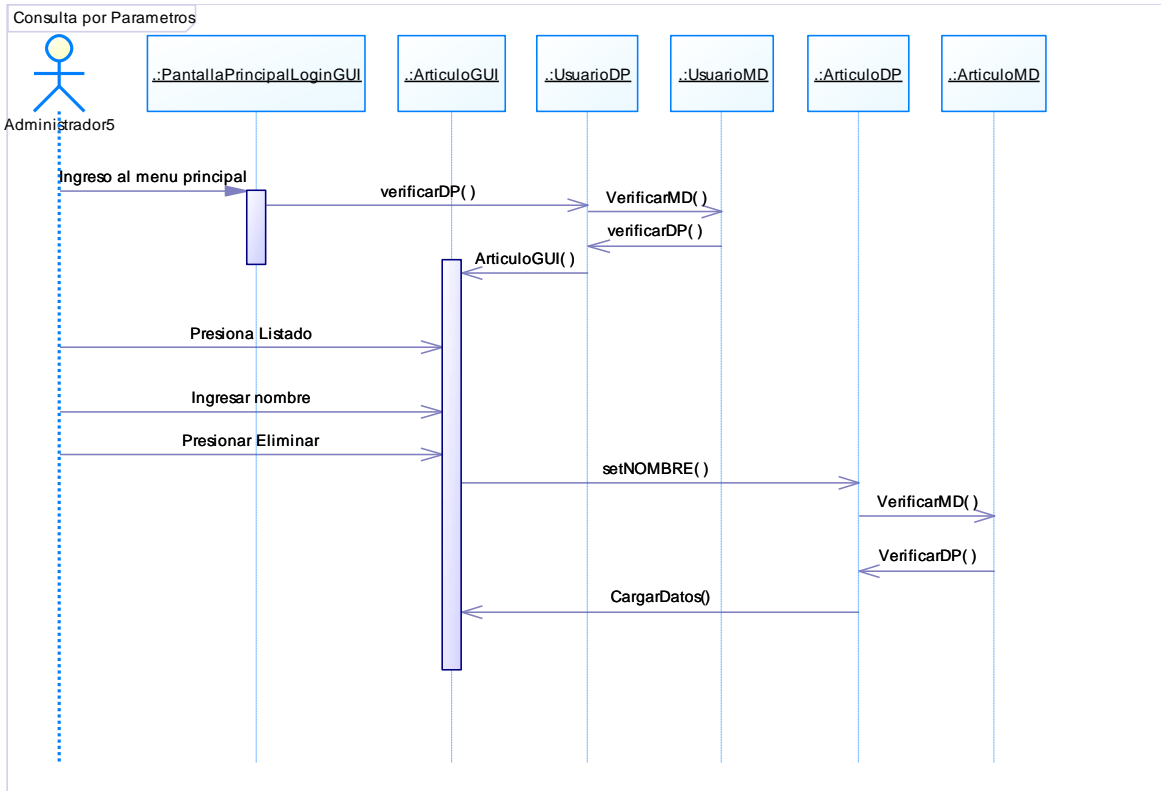


Ilustración 24: Consulta Parámetros Productos. Autor: Fernanda Nicolalde y Diego Rosero.

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

Proceso: Administración de Pedidos (Ingresar Pedido).

En el diagrama se presenta al solicitante como interactúa con el programa para generar pedidos.

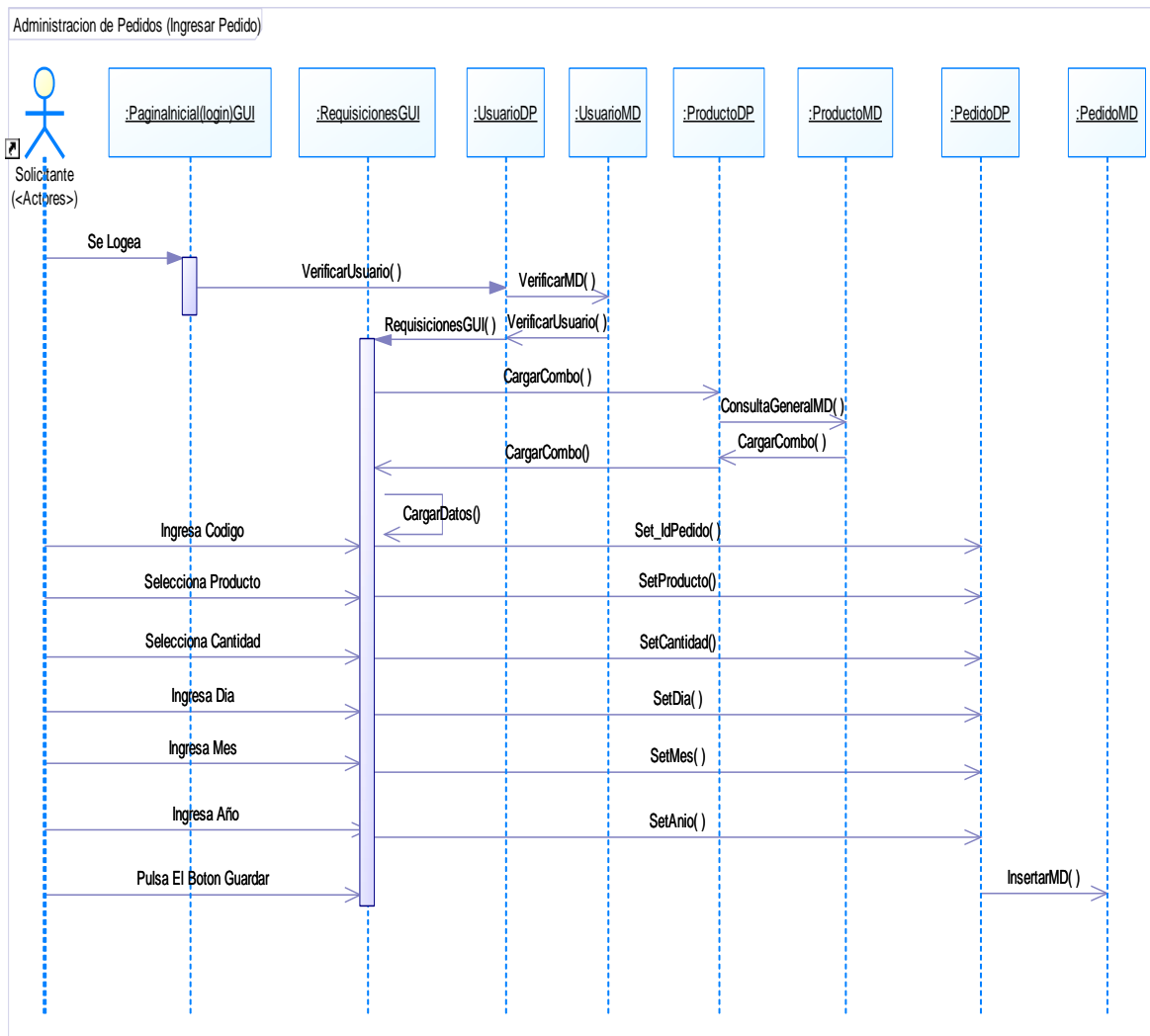


Ilustración 25: Ingresar Pedido. Autor: Fernanda Nicolalde y Diego Rosero.

5.3. Elaboración

Se detalla la elaboración del sistema y su funcionalidad.

5.3.1. Prototipo Ejecutable

Observamos las diferentes pantallas con su respectiva funcionalidad.

5.3.1.1. Ventana Inicial (Página de Inicio).

En esta pantalla inicial se realiza el ingreso de cada usuario con su respectivo rol para permitir ingresar a los módulos autorizados, para esto se entrega un usuario y clave, el administrador es la persona autorizada para acceder a todos los módulos del sistema.

5.3.1.2. Ventana Principal

Pantalla principal del sistema a la cual se tiene acceso dependiendo el permiso existente.

Administrador: SisInventario, bodega, inventario administración reportes, ver, ventanas y ayuda.

Bodeguero: bodega, inventario, reportes.

Vendedor: ventas, clientes, reportes.

SisInventario: tiene la opción para salir del sistema.

Bodega: permite el ingreso de nuevos productos, categorías de productos, dentro de las opciones de este módulo presenta un informe por fecha de los productos registrados.

Inventario: permite el registro de los productos adquiridos con sus respectivas cantidades para llevar el control del stock y tiene la opción de presentar un reporte de todos los productos; se ingresa también los proveedores.

Ventas: realiza el registro de las ventas de los productos permitiendo generar la factura, también tiene la opción para el registro de nuevos clientes, por otro lado, dentro de las opciones de este módulo genera un informe por fecha de las ventas efectuadas y de los clientes existentes.

Reportes: Presenta el reporte del stock actual de los productos que tiene la empresa.

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

Administración: permite registrar usuarios con sus respectivos roles para ingreso al sistema.

Ver: visualiza la barra de herramientas y barra de estado que por defecto se crea al momento de generar el formulario de la pantalla principal.

Ventanas: permite organizar las ventanas del sistema en varias opciones como nueva ventana, cascada, mosaico vertical, organizar iconos, y a su vez tiene la opción de cerrar todas las ventanas.

Ayuda: esta pestaña muestra la información de versionamiento del sistema y sus desarrolladores.

5.3.2. Lista De Riesgos

5.3.2.1. Descripción

Propósito:

El realizar esta lista nos ayuda a definir los posibles riesgos que se puedan presentar en la elaboración del proyecto, el hacer un detalle de cada uno con su respectiva acción a tomar reduce la posibilidad de que aparezca nuevamente.

Perspectiva general:

Se listan todos los riesgos identificados y analizados cada uno con la siguiente descripción:

TIPO DE RIESGO	RIESGO	PROBABILIDAD	EFECTO	ESTRATEGIA
Se detalla a que grupo de riesgo pertenece, puede ser de tipo proyecto (personal), producto (requisitos, estimación), negocio (comunicación)	Detalle del riesgo	Alta, Baja, Moderada	Catastrófico, Serio, Tolerable	Identificar la solución al riesgo presentado.

Tabla 6: Detalle de la tabla de Riesgos. Autor: Fernanda Nicolalde y Diego Rosero

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

5.3.2.2. Riesgos

TIPO DE RIESGO	RIESGO	PROBABILIDAD	EFECTO	ESTRATEGIA
Producto -requisitos	Cambios solicitados por el cliente por falta de conocimiento de sistema final.	Alta	Catastrófico	Solicitar todos los requerimientos que necesita la empresa y presentar un diseño previo a lo solicitado.
Proyecto- Personal	Posibilidad de que uno de los desarrolladores desista del proyecto.	Baja	Serio	Indicar cuál es el alcance del desarrollo y conocer las capacidades y cualidades de cada uno de los integrantes
Negocio-Comunicación	Empresa tiene que disolverse por falta de ingresos.	Moderada	Catastrófico,	Firmar un contrato con sus respectivas clausulas

Tabla 7: Riesgos Autor: Fernanda Nicolalde y Diego Rosero

5.4. Construcción

En el siguiente capítulo se describirá la implementación para cada perfil de usuario existente y pruebas del sistema en detalle.

5.4.1. Implementación y Pruebas

5.4.1.1. Ingreso al Sistema

5.4.1.1.1. Implementación

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Usuario	Usuario	Ingreso usuario mayor al permitido	Registro almacenado en la base de datos
Password	Password del usuario	Ingreso password mayor al permitido	Registro almacenado en la base de datos

Tabla 8: Ingreso al Sistema/Implementación Autor: Fernanda Nicolalde y Diego Rosero

5.4.1.1.2. Pruebas validaciones de usuario

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Usuario	Usuario	Ingreso usuario mayor al permitido	Mensaje ingrese un dato
Password	Password del usuario	Ingreso password mayor al permitido	Mensaje ingrese un dato

Tabla 9: Ingreso al Sistema/Prueba usuario Autor: Fernanda Nicolalde y Diego Rosero

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

5.4.1.2. *Ingreso de Usuarios*

5.4.1.2.1. *Implementación*

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Código	Código del usuario	Ningún error código autogenerado	Registro almacenado en la base de datos
Nombre	Nombre del usuario	Ingreso nombre mayor al permitido.	Registro almacenado en la base de datos
Apellido	Apellido del usuario	Ingreso apellido mayor al permitido.	Registro almacenado en la base de datos
Fecha de Nacimiento	Ingreso fecha de nacimiento	Ningún error fecha preestablecida	Registro almacenado en la base de datos
Sexo	Selección sexo del usuario	Ingreso sexo establecido por defecto.	Registro almacenado en la base de datos
Num. Documento	Selección tipo de documento/ ingreso tipo de documento del usuario	Ingreso tipo de documento establecido por defecto/ tipo de documento mayor al permitido	Registro almacenado en la base de datos
Dirección	Dirección del usuario	Ingreso de dirección mayor al permitido.	Registro almacenado en la base de datos
Teléfono	Teléfono del usuario	Ingreso de teléfono mayor al permitido	Registro almacenado en la base de datos
Email	Email del usuario	Ingreso email mayor al permitido	Registro almacenado en la base de datos
Acceso	Selección de acceso	Ingreso tipo de documento establecido por defecto	Registro almacenado en la base de datos
Usuario	Usuario	Ingreso usuario mayor al permitido	Registro almacenado en la base de datos
Password	Password del usuario	Ingreso password mayor al permitido	Registro almacenado en la base de datos

Tabla 10: Ingreso de usuarios/Prueba usuario Autor: Fernanda Nicolalde y Diego Rosero

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

5.4.1.2.2. Pruebas validaciones de usuario

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Código	Código del usuario	Ningún error código autogenerado	Ninguno
Nombre	Nombre del usuario	Ingreso nombre mayor al permitido.	Mensaje ingrese un dato
Apellido	Apellido del usuario	Ingreso apellido mayor al permitido.	Mensaje ingrese un dato
Fecha de Nacimiento	Ingreso fecha de nacimiento	Ningún error fecha preestablecida	Ninguno
Sexo	Selección sexo del usuario	Ingreso sexo establecido por defecto.	Mensaje ingrese un dato
Num. Documento	Selección tipo de documento/ ingreso tipo de documento del usuario	Ingreso tipo de documento establecido por defecto/ tipo de documento mayor al permitido	Mensaje ingrese un dato
Dirección	Dirección del usuario	Ingreso de dirección mayor al permitido.	Mensaje ingrese un dato
Teléfono	Teléfono del usuario	Ingreso de teléfono mayor al permitido	Mensaje ingrese un dato
Email	Email del usuario	Ingreso email mayor al permitido	Mensaje ingrese un dato
Acceso	Selección de acceso	Ingreso tipo de documento establecido por defecto	Mensaje ingrese un dato
Usuario	Usuario	Ingreso usuario mayor al permitido	Mensaje ingrese un dato
Password	Password del usuario	Ingreso password mayor al permitido	Mensaje ingrese un dato

Tabla 11: Ingreso de usuarios/Prueba usuario Autor: Fernanda Nicolalde y Diego Rosero

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

5.4.1.3. *Ingreso de Clientes*

5.4.1.3.1. *Implementación*

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Código	Código del cliente	Ningún error código autogenerado	Registro almacenado en la base de datos
Nombre	Nombre del cliente	Ingreso nombre mayor al permitido.	Registro almacenado en la base de datos
Apellido	Apellido del cliente	Ingreso apellido mayor al permitido.	Registro almacenado en la base de datos
Fecha de Nacimiento	Ingreso fecha de nacimiento	Ningún error fecha preestablecida	Registro almacenado en la base de datos
Sexo	Selección sexo del cliente	Ingreso sexo establecido por defecto.	Registro almacenado en la base de datos
Tipo de Documento	Selección tipo de documento/ ingreso tipo de documento del cliente	Ingreso tipo de documento establecido por defecto/ tipo de documento mayor al permitido	Registro almacenado en la base de datos
Dirección	Dirección del cliente	Ingreso de dirección mayor al permitido.	Registro almacenado en la base de datos
Teléfono	Teléfono del cliente	Ingreso de teléfono mayor al permitido	Registro almacenado en la base de datos
URL	URL del cliente	Ingreso de URL mayor al permitido	Registro almacenado en la base de datos
Email	Email del cliente	Ingreso email mayor al permitido	Registro almacenado en la base de datos

Tabla 12: Ingreso de clientes/Implementación Autor: Fernanda Nicolalde y Diego Rosero

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

5.4.1.3.2. Pruebas validaciones de usuario

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Código	Código del cliente	Ningún error código autogenerado	Ninguno
Nombre	Nombre del cliente	Ingreso nombre mayor al permitido.	Mensaje ingrese un dato
Apellido	Apellido del cliente	Ingreso apellido mayor al permitido.	Mensaje ingrese un dato
Fecha de Nacimiento	Ingreso fecha de nacimiento	Ningún error fecha preestablecida	Ninguno
Sexo	Selección sexo del cliente	Ingreso sexo establecido por defecto.	Mensaje ingrese un dato
Tipo de Documento	Selección tipo de documento/ ingreso tipo de documento del cliente	Ingreso tipo de documento establecido por defecto/ tipo de documento mayor al permitido	Mensaje ingrese un dato
Dirección	Dirección del cliente	Ingreso de dirección mayor al permitido.	Mensaje ingrese un dato
Teléfono	Teléfono del cliente	Ingreso de teléfono mayor al permitido	Mensaje ingrese un dato
URL	URL del cliente	Ingreso de URL mayor al permitido	Mensaje ingrese un dato
Email	Email del cliente	Ingreso email mayor al permitido	Mensaje ingrese un dato

Tabla 13: Ingreso de clientes/Prueba usuario Autor: Fernanda Nicolalde y Diego Rosero

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

5.4.1.4. Ingreso de proveedores

5.4.1.4.1. Implementación

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Código	Código del proveedor.	Ningún error código autogenerado	Registro almacenado en la base de datos
Razón social	Razón social del proveedor	Ingreso de razón social mayor al permitido.	Registro almacenado en la base de datos
Sector Comercial	Selección sector comercial del proveedor	Ingreso sector comercial establecido por defecto.	Registro almacenado en la base de datos
Tipo de Documento	Selección tipo de documento/ ingreso tipo de documento del proveedor	Ingreso tipo de documento establecido por defecto/ tipo de documento mayor al permitido	Registro almacenado en la base de datos
Dirección	Dirección del proveedor	Ingreso de dirección mayor al permitido.	Registro almacenado en la base de datos
Teléfono	Teléfono del proveedor	Ingreso de teléfono mayor al permitido	Registro almacenado en la base de datos
URL	URL del proveedor	Ingreso de UR mayor al permitido	Registro almacenado en la base de datos
Email	Email del proveedor	Ingreso email mayor al permitido	Registro almacenado en la base de datos

Tabla 14: Ingreso de proveedores/Implementación Autor: Fernanda Nicolalde y Diego Rosero

5.4.1.4.2. Pruebas validación de usuario

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Código	Código del proveedor.	Ningún error código autogenerado	Ninguno
Razón social	Razón social del proveedor	Ingreso de razón social mayor al permitido.	Mensaje ingrese un dato
Sector Comercial	Selección sector comercial del proveedor	Ingreso sector comercial establecido por defecto.	Ninguno
Tipo de Documento	Selección tipo de documento/ ingreso tipo de documento del proveedor	Ingreso tipo de documento establecido por defecto/ tipo de documento mayor al permitido	Ninguno/Mensaje ingrese un dato
Dirección	Dirección del proveedor	Ingreso de dirección mayor al permitido.	Mensaje ingrese un dato
Teléfono	Teléfono del proveedor	Ingreso de teléfono mayor al permitido	Mensaje ingrese un dato
URL	URL del proveedor	Ingreso de UR mayor al permitido	Mensaje ingrese un dato
Email	Email del proveedor	Ingreso email mayor al permitido	Mensaje ingrese un dato

Tabla 15: Ingreso de proveedores/Prueba usuario Autor: Fernanda Nicolalde y Diego Rosero

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

5.4.1.5. *Ingreso de Categorías*

5.4.1.5.1. *Implementación*

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Código	Código de la categoría.	Ingreso de código mayor al permitido.	Registro almacenado en la base de datos
Nombre	Nombre de la categoría	Ingreso de nombre mayor al permitido.	Registro almacenado en la base de datos
Descripción	Descripción de la categoría	Ingreso de detalle mayor al permitido.	Registro almacenado en la base de datos

Tabla 16: Ingreso de categorías/Implementación Autor: Fernanda Nicolalde y Diego Rosero

5.4.1.5.2. *Pruebas validaciones de usuario*

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Código	Código de la categoría.	Ingreso de código mayor al permitido.	Mensaje ingrese un dato
Nombre	Nombre de la categoría	Ingreso de nombre mayor al permitido.	Mensaje ingrese un dato
Descripción	Descripción de la categoría	Ingreso de detalle mayor al permitido.	Mensaje ingrese un dato

Tabla 17: Ingreso de categorías/Prueba usuario Autor: Fernanda Nicolalde y Diego Rosero

5.4.1.6. *Ingreso de Productos*

5.4.1.6.1. *Implementación*

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Código	Código del producto.	Ningún error código autogenerado	Registro almacenado en la base de datos
Código Ventas	Código del producto interno.	Ingreso de código ventas mayor al permitido.	Registro almacenado en la base de datos
Nombre	Nombre del producto	Ingreso de nombre mayor al permitido.	Registro almacenado en la base de datos
Descripción	Descripción del producto	Ingreso de la descripción mayor al permitido.	Registro almacenado en la base de datos
Categoría	Selección de Categoría	Categoría no ingresada	Registro almacenado en la base de datos
Imagen	Selección de una imagen	Ninguno	Registro almacenado en la base de datos

Tabla 18: Ingreso de productos/Implementación Autor: Fernanda Nicolalde y Diego Rosero

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

5.4.1.6.2. Pruebas validaciones de usuario

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Código	Código del producto.	Ingreso de código mayor al permitido.	Mensaje ingrese un dato
Código Ventas	Código del producto interno.	Ingreso de código mayor al permitido.	Mensaje ingrese un dato
Nombre	Nombre del producto	Ingreso de nombre mayor al permitido.	Mensaje ingrese un dato
Descripción	Descripción del producto	Ingreso de la descripción mayor al permitido.	Mensaje ingrese un dato
Categoría	Selección de Categoría	Categoría no ingresada	Mensaje ingrese un dato
Imagen	Selección de un imagen	Ninguno	Sin imagen

Tabla 19: Ingreso de productos/Prueba usuario Autor: Fernanda Nicolalde y Diego Rosero

5.4.1.7. Ingreso inventario

5.4.1.7.1. Implementación

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Código	Código de veta	Ningún error código autogenerado	Registro almacenado en la base de datos
Comprobante	Selección tipo de comprobante.	Comprobante seleccionado por defecto	Registro almacenado en la base de datos
Proveedor	Seleccionar proveedor	Proveedor no seleccionado	Registro almacenado en la base de datos
Numero	Número del secuencia del producto	Ingreso de números mayor al permitido.	Registro almacenado en la base de datos
Fecha	Fecha de registro	Ningún error fecha preestablecida	Registro almacenado en la base de datos
Producto	Selección de producto	Producto no ingresado	Registro almacenado en la base de datos
Cantidad	Cantidad del producto	Ingreso de cantidad mayor al permitido	Registro almacenado en la base de datos
Precio Compra	Precio de compra del producto	Ingreso de precio mayor al permitido	Registro almacenado en la base de datos
Precio Venta	Precio de venta del producto	Ingreso de precio mayor al permitido	Registro almacenado en la base de datos

Tabla 20: Ingreso inventario /Implementación Autor: Fernanda Nicolalde y Diego Rosero

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

5.4.1.7.1. Pruebas validaciones de usuario

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Código	Código de veta	Ningún error código autogenerado	Ninguno
Comprobante	Selección tipo de comprobante.	Comprobante seleccionado por defecto	Ninguno
Proveedor	Seleccionar proveedor	Proveedor no seleccionado	Mensaje ingrese un dato
Numero	Número del secuencia del producto	Ingreso de números mayor al permitido.	Mensaje ingrese un dato
Fecha	Fecha de registro	Ningún error fecha preestablecida	Ninguno
Producto	Selección de producto	Producto no ingresado	Mensaje ingrese un dato
Cantidad	Cantidad del producto	Ingreso de cantidad mayor al permitido	Mensaje ingrese un dato
Precio Compra	Precio de compra del producto	Ingreso de precio mayor al permitido	Mensaje ingrese un dato
Precio Venta	Precio de venta del producto	Ingreso de precio mayor al permitido	Mensaje ingrese un dato

Tabla 21: Ingreso ventas/Prueba usuario Autor: Fernanda Nicolalde y Diego Rosero

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

5.4.1.8. *Ingreso Venta*

5.4.1.8.1. Implementación

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Código	Código de veta	Ningún error código autogenerado	Registro almacenado en la base de datos
Comprobante	Selección tipo de comprobante.	Comprobante seleccionado por defecto	Registro almacenado en la base de datos
Cliente	Nombre del Cliente	Ingreso de comprobante mayor al permitido	Registro almacenado en la base de datos
Numero	Número del secuencia del producto	Ingreso de números mayor al permitido.	Registro almacenado en la base de datos
Fecha	Fecha de registro	Ningún error fecha preestablecida	Registro almacenado en la base de datos
IVA	IVA del producto	Ningún error IVA autodimensionado	Registro almacenado en la base de datos
Producto	Selección de producto	Producto no ingresado	Registro almacenado en la base de datos
Stock Inicial	Stock inicial del producto	Ingreso de cantidad mayor al permitido	Registro almacenado en la base de datos
Precio Compra	Precio de compra del producto	Ingreso de precio mayor al permitido	Registro almacenado en la base de datos
Precio Venta	Precio de venta del producto	Ingreso de precio mayor al permitido	Registro almacenado en la base de datos
Descuento	Descuento del producto	Descuento mayor al valor permitido	Registro almacenado en la base de datos

Tabla 22: Ingreso ventas/Implementación Autor: Fernanda Nicolalde y Diego Rosero

5.4.1.8.1. Pruebas validaciones de usuario

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Código	Código de veta	Ningún error código autogenerated	Ninguno
Comprobante	Selección tipo de comprobante.	Comprobante Seleccionado por defecto	Ninguno
Cliente	Nombre del Cliente	Ingreso de comprobante mayor al permitido	Mensaje ingrese un dato
Numero	Número del secuencia del producto	Ingreso de números mayor al permitido.	Mensaje ingrese un dato
Fecha	Fecha de registro	Ningún error fecha preestablecida	Ninguno
IVA	IVA del producto	Ningún error IVA autodimensionado	Mensaje ingrese un dato
Producto	Selección de producto	Producto no ingresado	Mensaje ingrese un dato
Stock Inicial	Cantidad del producto	Ingreso de cantidad mayor al permitido	Mensaje ingrese un dato
Precio Compra	Precio de compra del producto	Ingreso de precio mayor al permitido	Mensaje ingrese un dato
Precio Venta	Precio de venta del producto	Ingreso de precio mayor al permitido	Mensaje ingrese un dato
Descuento	Descuento del producto	Ninguno	Opcional el ingreso del valor

Tabla 23: Ingreso ventas/Prueba usuario Autor: Fernanda Nicolalde y Diego Rosero

5.4.1.9. Generación de Factura PDF

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Comprobante	Datos	Ninguno	Archivo factura PDF

Tabla 24: Generación Factura PDF Autor: Fernanda Nicolalde y Diego Rosero

5.4.1.10. Creación Archivo XML

Campo	Entradas Esperadas	Rango de Errores	Salidas Esperadas
Crear XML	Datos	Ninguno	Archivo factura XML

Tabla 25: Creación archivo XML Autor: Fernanda Nicolalde y Diego Rosero

5.5. Transición

5.5.1. Seguridades

El sistema presenta seguridades a nivel de usuarios, su funcionalidad permite que el usuario en la definición de su rol como por ejemplo administrador, vendedor, bodeguero, tengan diferentes accesos al sistema ayudando a tener nuestras categorías de ventanas separadas un usuario de otro. A nivel de base de datos tenemos la seguridad que brinda SQL Server permitiendo tener un administrador general de la base y roles en el caso de ser necesario.

5.5.2. Costos

Los costos de acuerdo al proyecto desarrollado:

COSTOS INICIALES DEL PROYECTO	
HADWARE	2200
SOFTWARE	1360
INSTALACION	400
CONFIGURACIÓN	400
TOTAL	4360

Tabla 26: Costos iniciales Autor: Fernanda Nicolalde y Diego Rosero

COSTOS ADMINISTRATIVOS MENSUALES	
COSTO POR HORA	8
HORAS DE TRABAJO	160
TOTAL	1280

Tabla 27: Costos Administrativos Autor: Fernanda Nicolalde y Diego Rosero

COSTOS OPERATIVOS	
INGENIEROS	2
HORAS LABORABLES MENSUALES	64
COSTO POR HORA	8
TIEMPO ESTIMADO MENSUAL	2
TOTAL	2048

Tabla 28: Costos Operativos Autor: Fernanda Nicolalde y Diego Rosero

COSTOS SOPORTE TÉCNICO	
COSTO PROMEDIO POR HORA DE UN INGENIERO DE SOPORTE	120
NUMERO DE INCIDENTES PROMEDIO AL AÑO	10
TIEMPO DE SOPORTE ESTIMANOS EN HORAS	2
TOTAL	2400

Tabla 29: Costos soporte técnico Autor: Fernanda Nicolalde y Diego Rosero

RESUMEN COSTOS DEL PROYECTO	
COSTOS INICIALES DEL PROYECTO	4360
COSTOS OPERATIVOS	2048
COSTOS SOPORTE TÉCNICO	2400
TOTAL	8808

Tabla 30: Resumen Costo del proyecto Autor: Fernanda Nicolalde y Diego Rosero

5.5.3. Despliegue

Una vez concluido el sistema se le recomendará a la empresa, que va adquirir el producto, brindarles una capacitación completa funcional del sistema, con esto la parte centrada da finalización al proyecto de desarrollo.

El tiempo estimado para la capacitación depende del número de módulos que el sistema contiene y las diferentes personas a las cuales se les brinde la capacitación.

5.5.4. Mantenimiento

EL mantenimiento se lo realizara de acuerdo a un periodo especificado en el contrato, normalmente la empresa brinda este servicio por los primeros 2 años sin costo alguno en los periodos de 6 meses, si la empresa necesitara un cambio adicional el cual no esté dentro de los módulos establecidos la misma tendrá que adquirir el costo por el nuevo desarrollo, caso contrario nosotros realizaremos los cambios dentro de los módulos desarrollados.

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

Por medio de las normativas, el costo por mantenimiento se mantendría en el 10% del costo del producto terminado.

6. CAPÍTULO: CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

- Por medio de la metodología RAD, se realizó el análisis de los requerimientos y la construcción del sistema.
- Mediante el IDE Visual Studio, se realizó la correcta elaboración del archivo XML, ya que el mismo constaba con una librería que nos facilitó la construcción del mismo.
- La base de datos que se utiliza para el desarrollo es Sql Server por la compatibilidad que tiene con .Net, sin embargo, Oracle también es compatible, pero por su alto costo en licenciamiento y por el tipo de servidor que ocupa no se usa este motor.
- Los diagramas UML, sirven para conocer los actores que van a interactuar en el sistema y conocer los permisos que van a tener el sistema.
- El trabajar en 3 capas el sistema permite un fácil manejo de los datos y de las diferentes conexiones, así como también la seguridad del manejo de los datos.
- Mediante la programación en 3 capas podemos concluir que el desarrollo de un sistema sea este cualquiera, llega a ser más cómodo en cuanto a requerimientos de cambios que el usuario necesite.
- En cuanto al manejo de procesos ya establecidos y procesos a construir, se realizó un estudio de los procesos ya establecidos con el fin de llegar a un correcto desempeño con los procesos a desarrollar para que lleven una buena interacción, ya que los procesos nuevos dependen de los anteriores.
- Mediante el uso de la factura electrónica la empresa tendrá un consumo menor en papel ya que genera documentos PDF y archivos en XML para el proceso de facturación, permitiendo cuidar el ambiente con la disminución de la tala de árboles.

6.2. Recomendaciones

- La compatibilidad de sistemas o casas comerciales como en este caso, Microsoft, es muy importante, con el fin de que todos los sistemas funcionen de manera unificada, es recomendable que si se ocupa un IDE de la casa comercial Microsoft, como lo es Visual Studio, este debe ir de la mano con un motor de Base de Datos de la misma compañía como lo es SQL server, de esta manera se evita la incompatibilidad entre ellos.
- Una de las mejores maneras de llevar una aplicación altamente estable y dedicada a las necesidades del usuario es saber escuchar lo que usuario necesita, de la misma forma se debe tomar en cuenta los procesos que la empresa lleva y saber unirlos a los requerimientos o procesos nuevos que se van a implementar.
- La mejor táctica en el desarrollo de un sistema es llevarlo al límite en donde el usuario se identifique con el sistema, esto se logra con la metodología RAD, la cual arranca desde un diseño aceptable por el cliente y posible para el equipo de desarrolladores, limitando la exposición del proyecto a fuertes cambios.
- Entregar el software al tiempo contratado, con el fin de no llegar a tener malos entendidos con la empresa la cual contrata los servicios de desarrollo de software y poder concluir con las pruebas y cargas de información antes de que el aplicativo entre a producción.
- Implementar en un futuro los diferentes archivos de facturación electrónica que deben ser entregados al SRI.
- Realizar periódicamente copias de seguridad de la base de datos para no perder información y tener un respaldo de los movimientos que se han realizado en la base.
- Actualizar los contenidos de las materias de carrera dependiendo las nuevas tecnologías implementadas en el desarrollo de software como, por ejemplo: desarrollo de aplicaciones para Smartphone y la tecnología de facturación electrónica la cual está en auge en el país.
- Para la interacción con el sistema se recomienda llevar una infraestructura adecuada en el rendimiento de hardware, con el fin de llevar una

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

transaccionalidad aceptable del sistema hacia el motor de bases de datos, con esto se evitará la pérdida de información.

BIBLIOGRAFÍA

DE LIBROS:

- Aguilar, L. J. (1997). Ingeniería del software: un enfoque práctico. En L. J. Aguilar, *Ingeniería del software* (pág. 581). Madrid: Mikel Angoar.
- Murrilo Alfaro, F. e. (1999). *Herramienta Case*.
- Ramos, M. J., Ramos, A., & Montero, F. (2006). *Sistemas gestores de base de datos*. España: McGraw-Hill/Interamericana de España.
- Mistry, R., & Misner, S. (2014). *Introducing Microsoft SQL Server 2014 Technical Overview*. E.E.U.U: Microsoft Press.

DE INTERNET:

- Bautista Q, J. M. (24 de 06 de 2009). *Ingeniería de Software*. Obtenido de http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html
- Camacho Suárez, A., Lagunes Collazo, P., & Reyes Pérez, L. (19 de 10 de 2002). *El periódico de los Universitarios*. Obtenido de <https://www.uv.mx/universo/70/Date%20Vuelo/vuelo02.htm>
- Cruz Vilchez, F. (2012). *Programación en java*. Malaga: Fundacion Universitaria Andaluza Inca Garcilaso para eumed.net.
- Cruz, F. (05 de diciembre de 2015). *Eumed.net*. Obtenido de <http://www.eumed.net/libros-gratis/ciencia/2012/12/que-es-java.html>
- EcuREd. (24 de Septiembre de 2012). Obtenido de EcuREd: http://www.ecured.cu/IDE_de_Programaci%C3%B3n
- Exces Consultoria. (13 de 12 de 2013). *Manual de XML*. Obtenido de <http://www.mundolinux.info/index.htm>
- Galeano, J. (17 de octubre de 2013). *Prezi*. Obtenido de <https://prezi.com/sqoaduiqahuh/rad/>
- ervacio, J. (05 de junio de 2013). *Prezi*. Obtenido de <https://prezi.com/caho6vnry56-/untitled-prezi/>
- Information technology services. (2014). *Microsoft .NET Developemt*. Obtenido de <http://www.mooovit.com/tecnologia/microsoft-net-development>

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

- Israfevis. (06 de diciembre de 2012). *Arquitectura de la base de datos*. Obtenido de <https://israfives.wordpress.com/2012/12/06/1-4-arquitectura-de-la-base-de-datos/>
- Martinez, M. (02 de marzo de 2015). *Prezi*. Obtenido de <https://prezi.com/twuzzym3th5p/joint-application-design-jad/>
- Mesa, S. (16 de octubre de 2008). *Wordpress*. Obtenido de <https://silmc1515.wordpress.com/>
- Microsoft. (Octubre de 2015). *Microsoft Developer Network*. Obtenido de <https://msdn.microsoft.com/es-ec/library/dn762121.aspx>
- Microsoft. (Octubre de 2015). *SQL Server 2014*. Obtenido de <https://www.microsoft.com/es-es/server-cloud/products/sql-server/>
- Reyes, R. (14 de julio de 2014). *Clubensayos*. Obtenido de <https://www.clubensayos.com/Tecnolog%C3%ADa/Metodologia-XP/1874040.html>
- Rodriguez, E. (20 de enero de 2011). *Modelos del ciclo de vida vs. Metodologías*. Obtenido de <http://modelosymetodologias.blogspot.com/2011/01/modelo-dra.html>
- Vargas, S. (16 de 09 de 2011). *Mundo de Selenita*. Obtenido de <http://selene-vargas.blogspot.com/2011/09/joint-application-developmentjad.html>
- Wikipedia. (09 de noviembre de 2015). *Programación extrema*. Obtenido de https://es.wikipedia.org/wiki/Programaci%C3%B3n_extrema
- Wikipedia. (2015 de noviembre de 2015). *Programacion por capas*. Obtenido de https://es.wikipedia.org/wiki/Programaci%C3%B3n_por_capas
- Za, Y. (08 de febrero de 2015). Obtenido de https://prezi.com/oec_onqg5la6/copy-of-112-relacion-del-dba-con-otras-areas-de-los-sistemas/
- House, S. S. (14 de Julio de 2013). *SlideShare*. Obtenido de <http://es.slideshare.net/SoftexSoftware/sql-server-vs-oracle-dbms-comparison-24215631>

INVESTIGADOR

- Fernanda Nicolalde, Diego Rosero, Estudiantes de Ingeniería en Sistemas, PUCE.

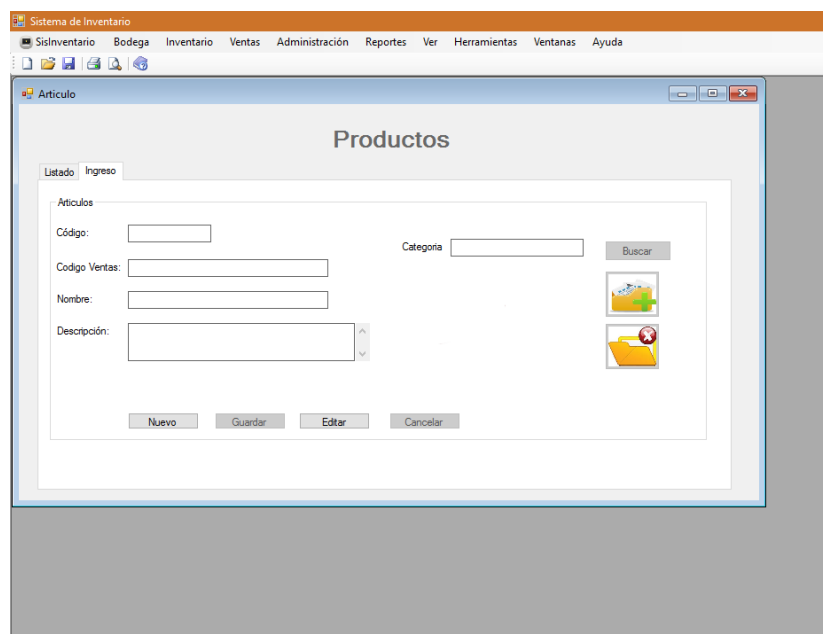
Anexo A

1. Manual de Usuario

1.1. Módulo Bodega.

En este módulo se hacen los ingresos y modificaciones de las categorías, de la misma forma se realizan los ingresos y modificaciones de productos que son registrados en el sistema.

1.1.1. Ingreso Nuevo Producto.

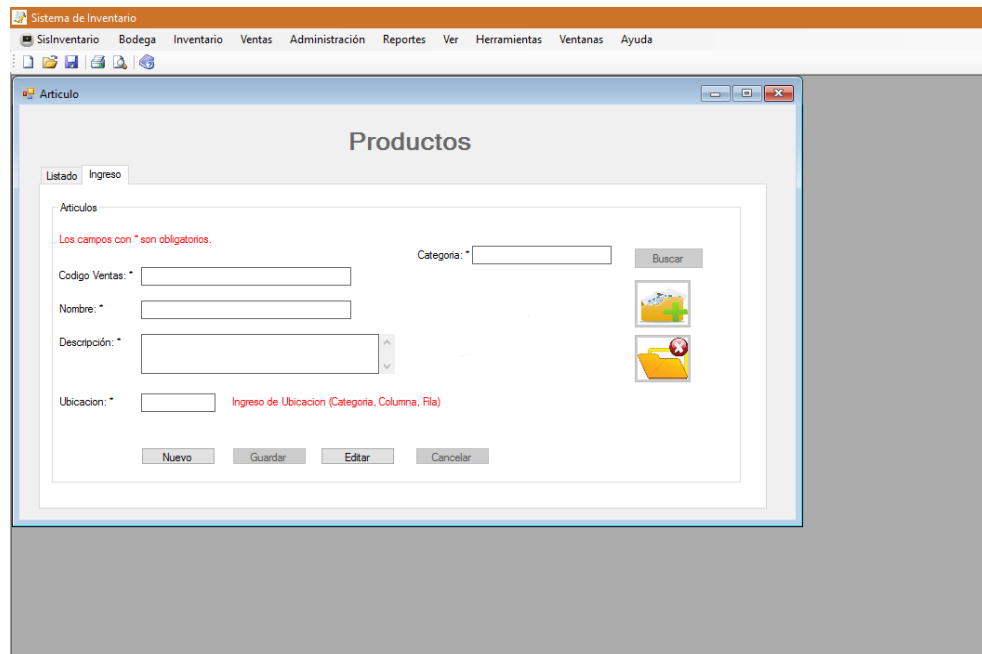


The screenshot shows a web application window titled 'Sistema de Inventario'. The main menu includes 'SisInventario', 'Bodega', 'Inventario', 'Ventas', 'Administración', 'Reportes', 'Ver', 'Herramientas', 'Ventanas', and 'Ayuda'. The active window is 'Artículo', which displays a 'Productos' form. The form has two tabs: 'Listado' and 'Ingreso'. The 'Ingreso' tab is selected. The form contains the following fields and controls:

- 'Código:' text input field
- 'Codigo Ventas:' text input field
- 'Nombre:' text input field
- 'Descripción:' text area with a vertical scrollbar
- 'Categoría:' dropdown menu
- 'Buscar' button
- 'Nuevo', 'Guardar', 'Editar', and 'Cancelar' buttons at the bottom

En este formulario todos los campos se los debe de llenar de forma obligatoria; en el caso de que algún campo falte se desplegará una alerta con un mensaje correspondiente al campo donde falte información.

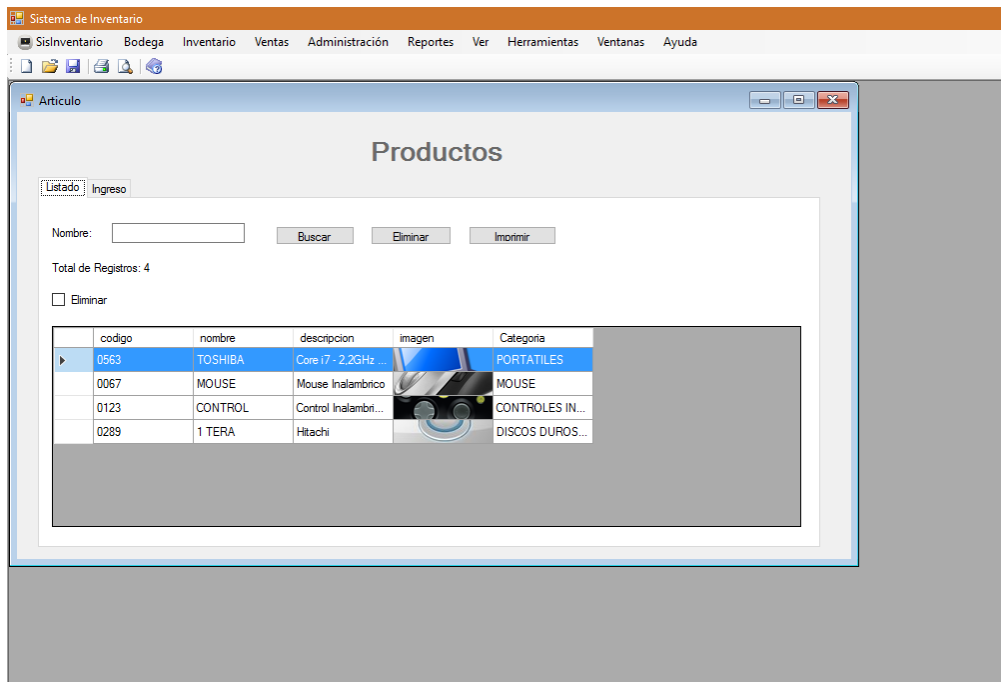
DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION



1.1.2. Reporte de Productos en el Sistema.

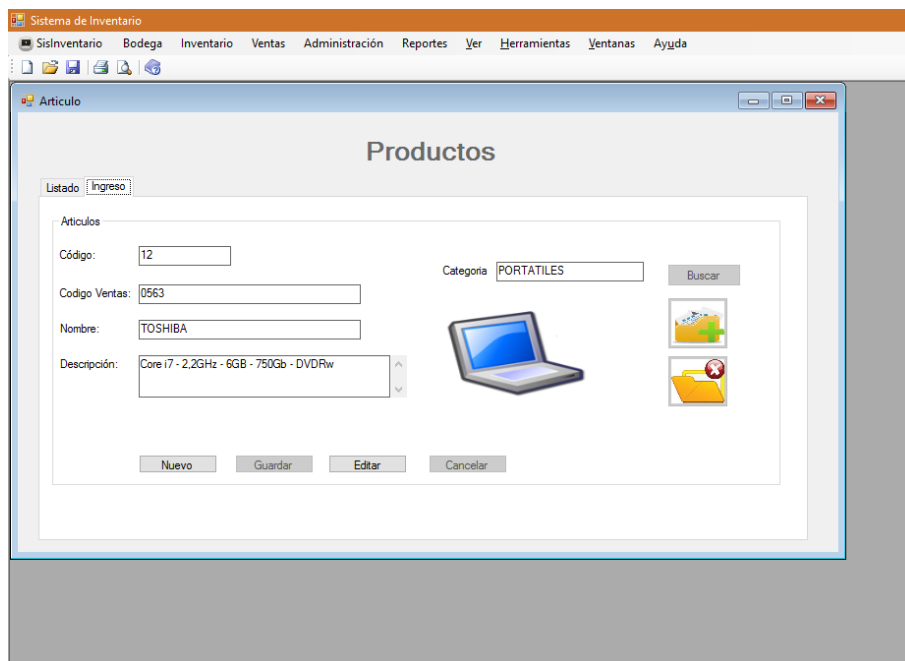
En esta opción se observa todos los productos que se encuentran ingresados en el sistema. La tabla es dinámica, lo cual permite búsqueda por parámetro y ordenamiento ascendente o descendente por medio de la columna que se elija como se muestra en la imagen.

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION



Nota: Si se da doble click encima del producto el sistema redirige a la pantalla de edición.

1.1.3. Edición de Productos.

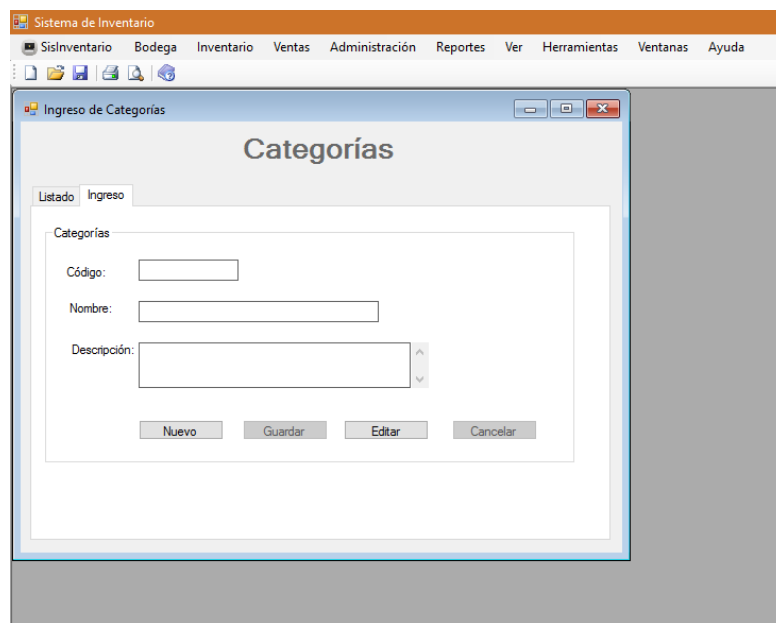


DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

Esta pestaña, que es parecida a la de ingreso de producto, se puede editar los campos que se desee, menos el campo código ya que éste es autogenerado por la base de datos.

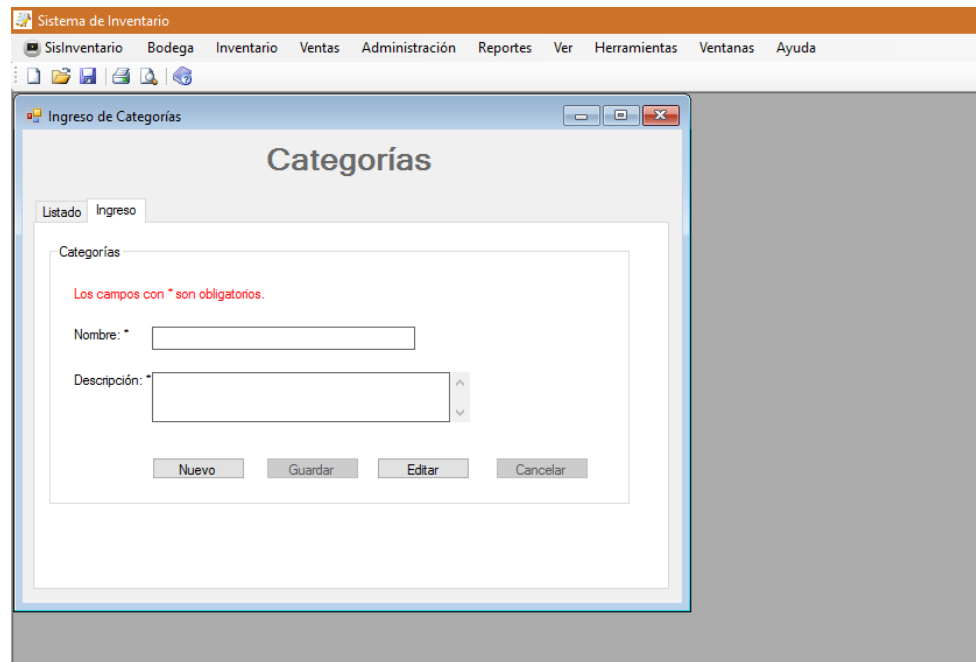
Nota: Esta ventana tiene los mismos controles que la pestaña de ingreso de productos.

1.1.4. Ingreso de Nueva Categoría.



En este formulario todos los campos se los debe de llenar de forma obligatoria; en el caso de que algún campo falte se desplegara una alerta con un mensaje correspondiente al campo donde falte información.

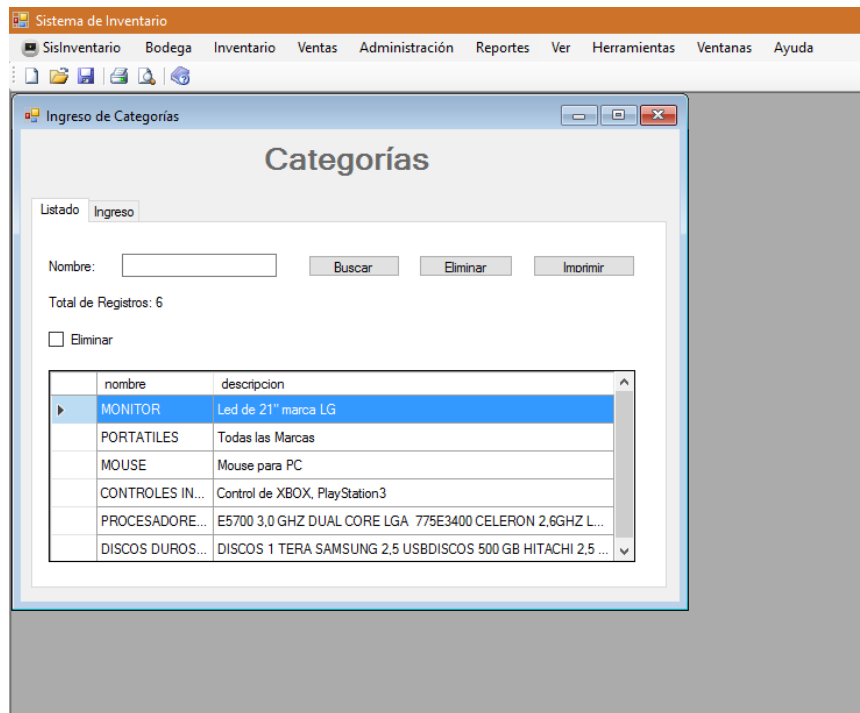
DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION



1.1.5. Reporte de Categorías en el Sistema.

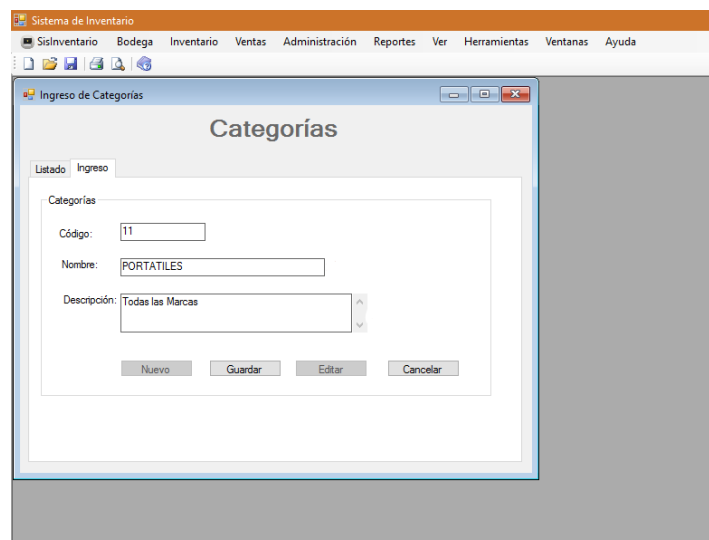
En esta opción se observa todas las categorías que se encuentran ingresados en el sistema. La tabla es dinámica lo cual permite búsqueda por parámetro y ordenamiento ascendente o descendente por medio de la columna que se elija como se muestra en la imagen.

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION



Nota: Si se da doble click encima de la categoría el sistema redirige a la pantalla de edición.

1.1.6. Edición de Categoría.



Esta pantalla, que es parecida a la de ingreso de categoría, se puede editar los campos que se desee, menos el campo código ya que este es autogenerado por la base de datos.

Nota: Esta ventana tiene los mismos controles que la pestaña de ingreso de categorías

1.2. Módulo Inventario.

En este módulo se hacen los ingresos y modificaciones de las compras que la empresa obtiene de proveedores externos, una vez creado el ingreso en el inventario no se lo puede eliminar, solo se lo podrá anular.

1.2.1. Ingreso Nueva Compra

The screenshot shows a web application window titled 'Sistema de Inventario' with a menu bar containing 'SisInventario', 'Bodega', 'Inventario', 'Ventas', 'Administración', 'Reportes', 'Ver', 'Herramientas', 'Ventanas', and 'Ayuda'. The main window is titled 'Ingreso Bodega' and has two tabs: 'Listado' and 'Ingreso'. The 'Ingreso' tab is active, showing a form titled 'Ingresos Almacen'. The form contains the following fields and controls:

- Proveedor:** Text input field with a 'Buscar' button.
- Fecha:** Date dropdown menu showing '17/03/2016'.
- Comprobante:** Dropdown menu with 'FACTURA' selected.
- Numero:** Text input field.
- Producto:** Text input field with a 'Buscar' button.
- Cantidad:** Text input field.
- Precio Compra:** Text input field with a '+' icon.
- Precio Venta:** Text input field with a '-' icon.

Below the form is a table with the following columns: 'idarticulo', 'articulo', 'precio_compra', 'precio_venta', 'stock_inicial', and 'subtotal'. The table body is currently empty. At the bottom left, it says 'Total Pagado: \$ 0.0'. At the bottom right, there are three buttons: 'Nuevo', 'Guardar', and 'Cancelar'.

Se llenan los campos correspondientes; todos los campos en este formulario son obligatorios; en el caso de que algún campo obligatorio falte se desplegará una alerta con un mensaje correspondiente al error.

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

1.2.2. Reporte de Compras en el Sistema.

En esta pestaña se observa todos los ingresos de compras que se encuentran en el sistema. La tabla es dinámica lo cual permite búsqueda por fecha y ordenamiento ascendente o descendente por medio de la columna que se elija como se muestra en la imagen.

Sistema de Inventario

SisInventario Bodega Inventario Ventas Administración Reportes Ver Herramientas Ventanas Ayuda

Ingreso Bodega

Listado Ingreso

Fecha Inicio: 12/02/2016 Fecha Fin: 12/02/2016 Buscar Anular Imprimir

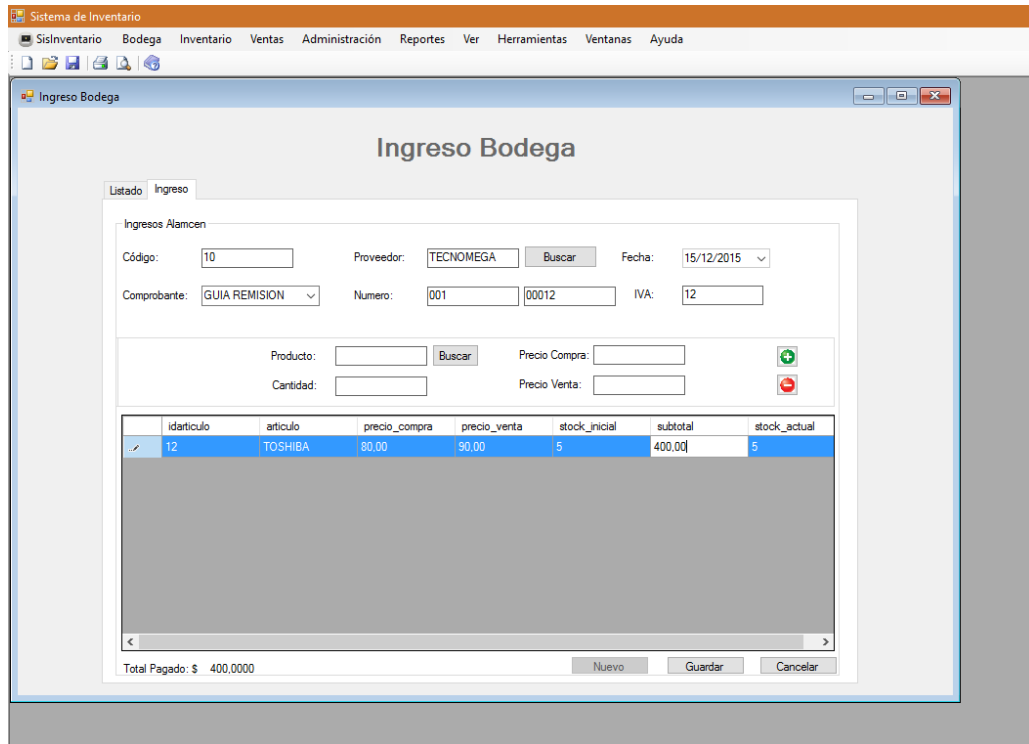
Anular Total de Registros: 5

Trabajador	proveedor	fecha	tipo_comprobante	serie	comrelativo	estado
ROSERO DIEGO	L.M.S CIA. LTDA.	14/12/2015	GUIA REMISION	001	00015	EMITIDO
ROSERO DIEGO	TECNOMEGA	15/12/2015	GUIA REMISION	001	00012	EMITIDO
ROSERO DIEGO	TECNOMEGA	12/12/2015	GUIA REMISION	001	0008	EMITIDO
ROSERO DIEGO	TECNOMEGA	12/09/2015	FACTURA	001	0006	EMITIDO
ROSERO DIEGO	TECNOMEGA	12/08/2015	FACTURA	0001	0005	EMITIDO

Nota: Si se da doble click encima de la compra el sistema redirige a la pantalla de edición.

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

1.2.3. Edición de Compras.



Nota: Esta ventana tiene los mismos controles que la pestaña de ingreso de compras, para editar las mismas se lo debe hacer en el contenedor que se carga la compra como se muestra en la figura.

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

1.2.4. Ingreso Nuevo Proveedor.

Sistema de Inventario

SisInventario Bodega Inventario Ventas Administración Reportes Ver Herramientas Ventanas Ayuda

Proveedores

Proveedores

Listado Ingreso

Código:

Razón Social: Sector Comercial: Tecnología al XMayor

Tipo de Documento: RUC

Dirección:

Teléfono:

Url:

E-mail:

Nuevo Guardar Editar Cancelar

Se llenan los campos correspondientes; todos los campos en este formulario son obligatorios; en el caso de que algún campo obligatorio falte se desplegará una alerta con un mensaje correspondiente al error.

Sistema de Inventario

SisInventario Bodega Inventario Ventas Administración Reportes Ver Herramientas Ventanas Ayuda

Proveedores

Proveedores

Listado Ingreso

Los campos con * son obligatorios.

Razón Social: * Sector Comercial: Tecnología al XMayor

Tipo de Documento: * RUC

Dirección: *

Teléfono: *

Url: *

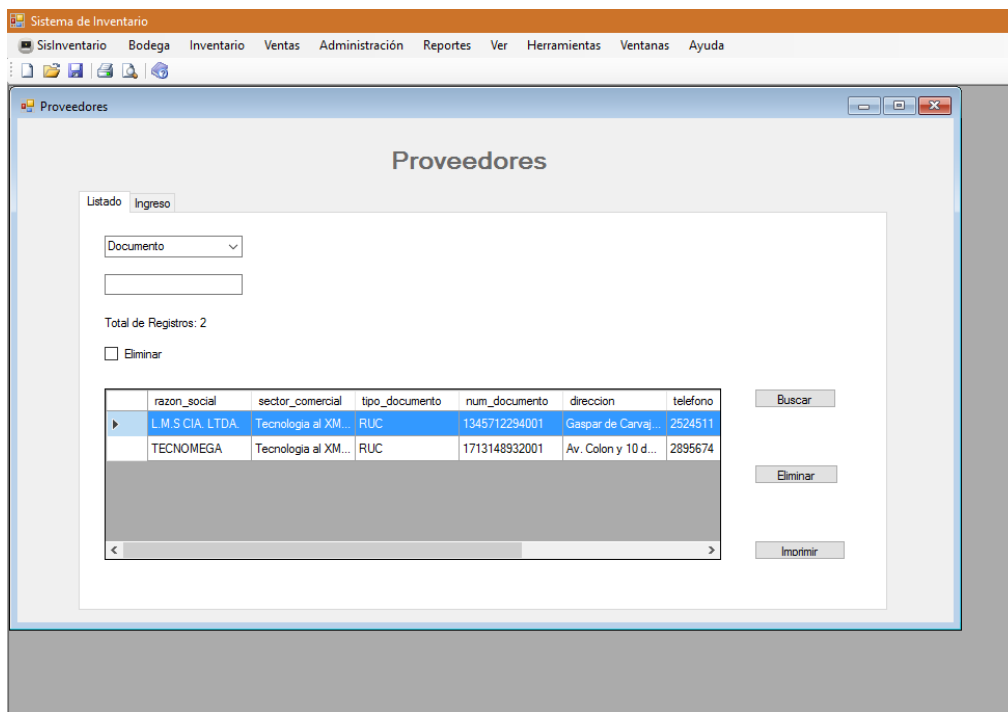
E-mail: *

Nuevo Guardar Editar Cancelar

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

1.2.5. Reporte de Proveedores.

En esta opción se observa todos los proveedores que se encuentran ingresados en el sistema. La tabla es dinámica lo cual permite búsqueda por documento (cédula), por razón social y ordenamiento ascendente o descendente por medio de la columna que se elija como se muestra en la imagen.



Nota: Si se da doble click encima de la compra el sistema redirige a la pantalla de edición.

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

1.2.6. Edición Proveedor.

Sistema de Inventario

SisInventario Bodega Inventario Ventas Administración Reportes Ver Herramientas Ventanas Ayuda

Proveedores

Listado Ingreso

Proveedores

Código: 4

Razón Social: L.M.S CIA. LTDA. Sector Comercial: Tecnología al XMayor

Tipo de Documento: RUC 1345712294001

Dirección: Gaspar de Carvajal N27-120 y Selva Alegre

Teléfono: 2524511

Url: www.lmsistemas.com

Email: vend4_lmsistemas@hotmail.com

Nuevo Guardar Editar Cancelar

Esta pestaña, que es parecida a la de ingreso de proveedor, se puede, editar los campos que se desee, menos el campo código ya que este es autogenerado por la base de datos.

1.3. Modulo Ventas.

En este módulo se hacen los ingresos y modificaciones de las ventas que la empresa realiza dentro de un determinado tiempo.

1.3.1. Ingreso Nueva Venta.

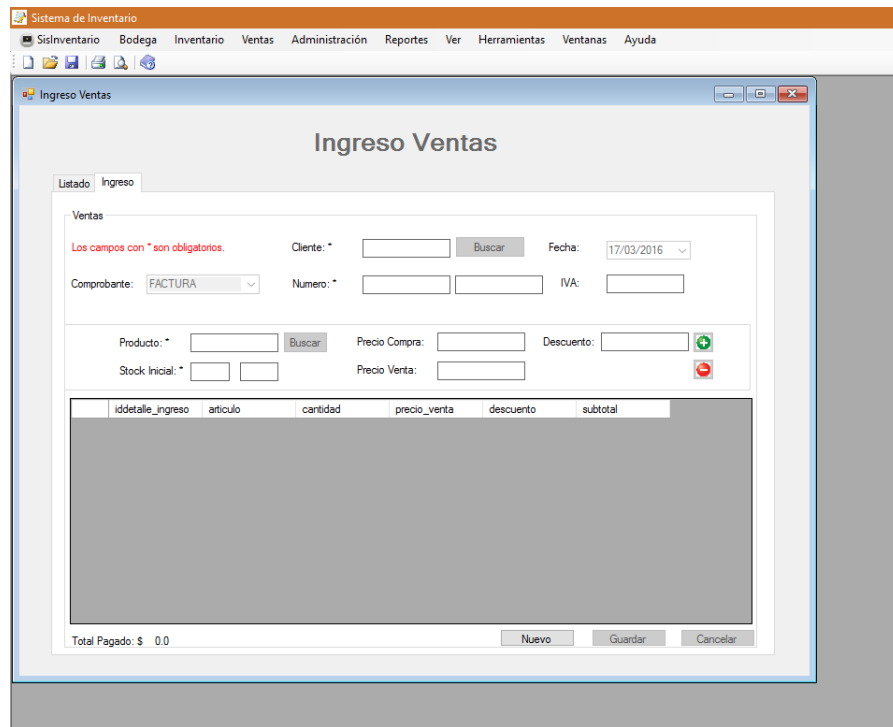
The screenshot shows a web-based application window titled 'Sistema de Inventario'. The main window is 'Ingreso Ventas'. It has a menu bar with 'SisInventario', 'Bodega', 'Inventario', 'Ventas', 'Administración', 'Reportes', 'Ver', 'Herramientas', 'Ventanas', and 'Ayuda'. The 'Ingreso Ventas' window has two tabs: 'Listado' and 'Ingreso'. The 'Ingreso' tab is active. The form contains the following fields and controls:

- Ventas:** Código (text), Cliente (text), Buscar (button), Fecha (dropdown, 13/02/2016).
- Comprobante:** FACTURA (dropdown), Numero (text), IVA (text).
- Producto:** Producto (text), Buscar (button), Precio Compra (text), Descuento (text), + (button).
- Stock Inicial:** Stock Inicial (text), Precio Venta (text), - (button).

Below the form is a table with the following columns: iddetalle_ingreso, articulo, cantidad, precio_venta, descuento, subtotal. The table is currently empty. At the bottom left, it says 'Total Pagado: \$ 0.0'. At the bottom right, there are buttons for 'Nuevo', 'Guardar', and 'Cancelar'.

Se llenan los campos correspondientes; todos los campos en este formulario son obligatorios; en el caso de que algún campo obligatorio falte se desplegará una alerta con un mensaje correspondiente al error.

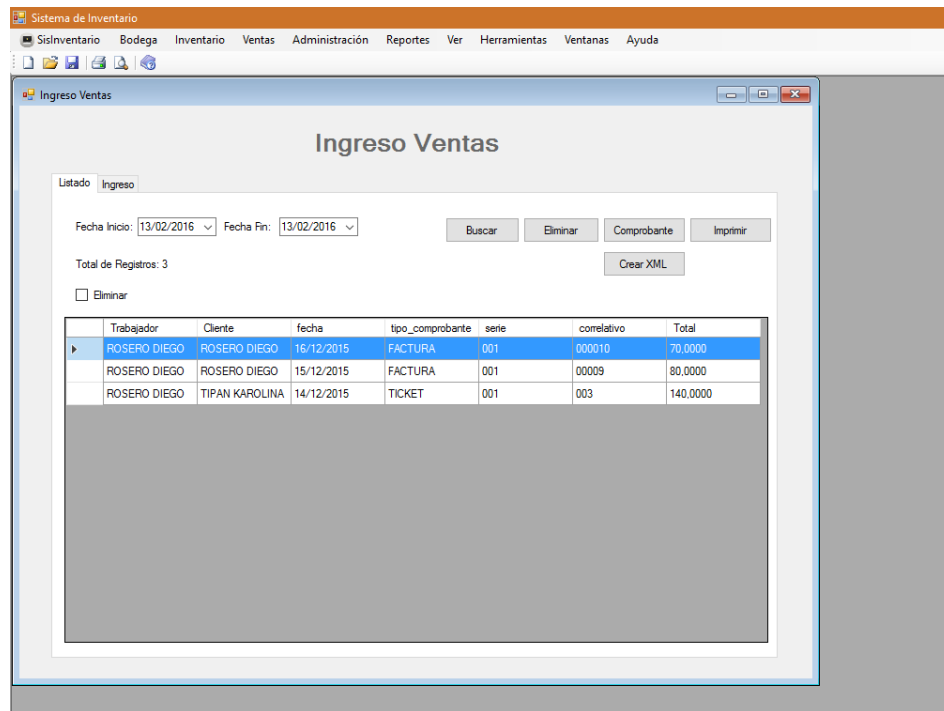
DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION



1.3.2. Reporte de Venta.

En esta pestaña se observa todas las ventas realizadas en el sistema. La tabla es dinámica lo cual permite búsqueda por fecha y ordenamiento ascendente o descendente por medio de la columna que se elija como se muestra en la imagen.

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

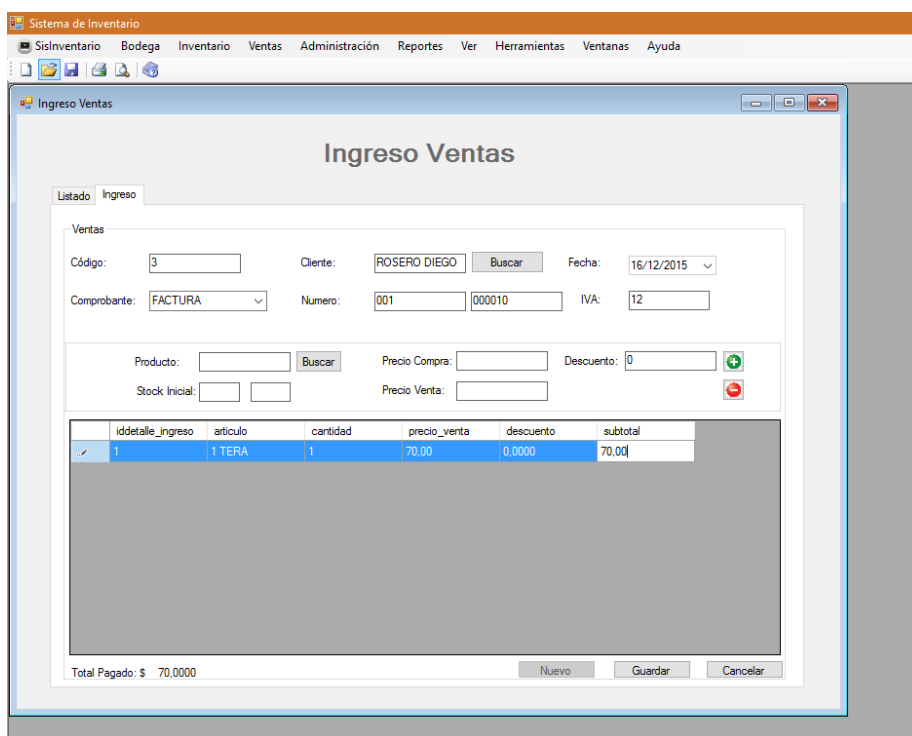


En esta parte del sistema se realizan los documentos XML de cada una de las facturas.

Nota: Si se da doble click encima de la compra el sistema redirige a la pantalla de edición.

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

1.3.3. Edición de Venta.

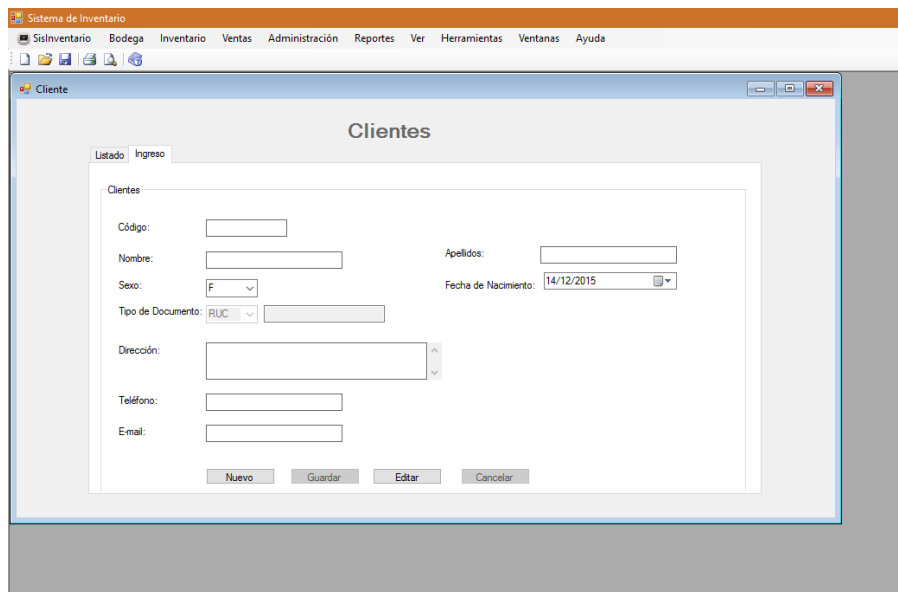


Nota: Esta ventana tiene los mismos controles que la pestaña de ingreso de ventas, para editar las mismas se lo debe hacer en el contenedor que se carga la venta como se muestra en la figura.

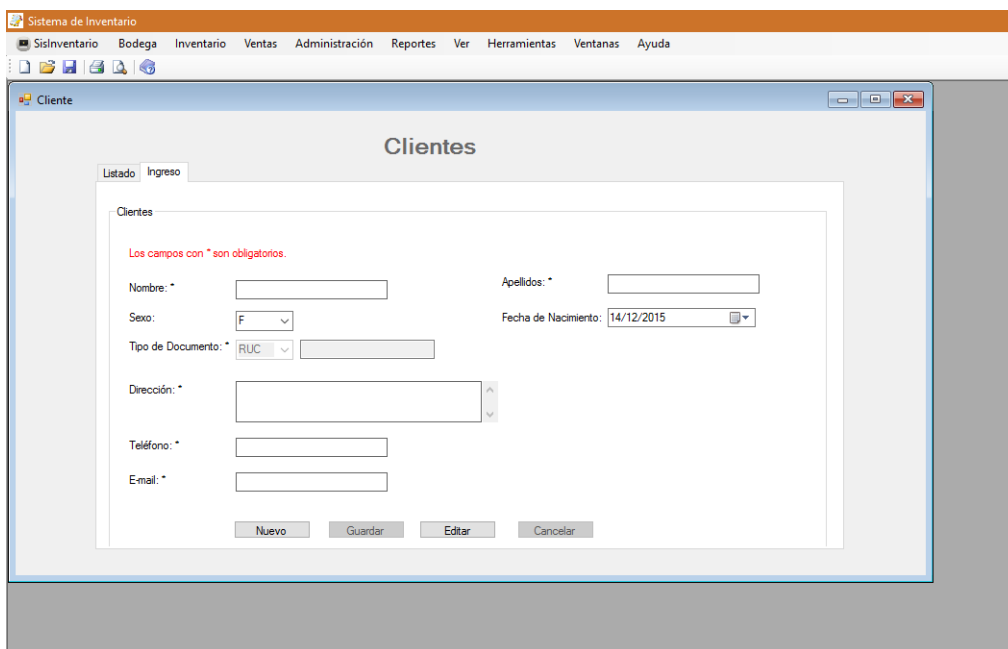
En este módulo se hacen los ingresos y modificaciones de los clientes, esto se realiza cada vez que la empresa obtiene nuevos clientes.

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

1.3.4. Ingreso Nuevo Cliente.

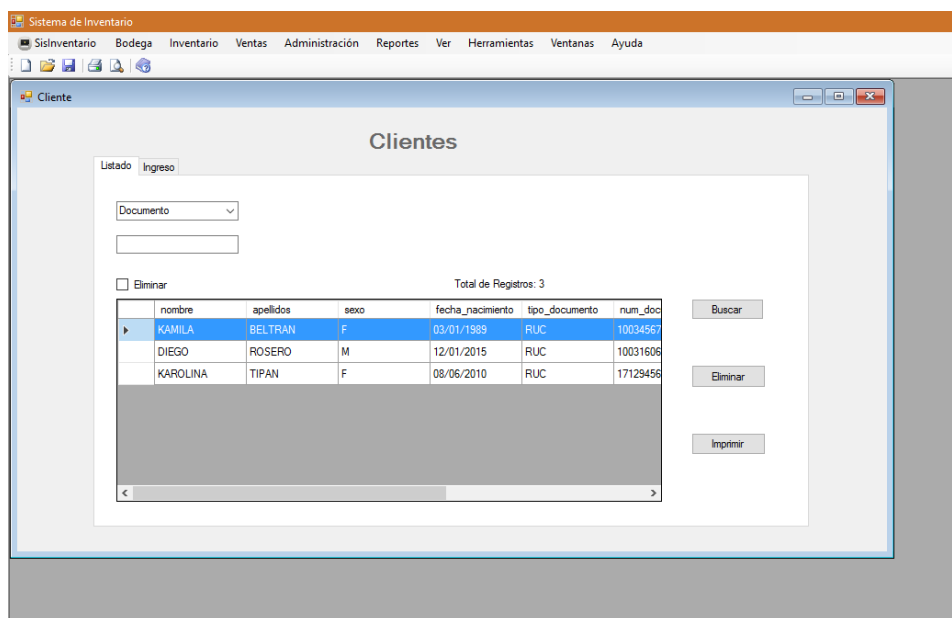


Se llenan los campos correspondientes; todos los campos en este formulario son obligatorios; en el caso de que algún campo obligatorio falte se desplegará una alerta con un mensaje correspondiente al error.



1.3.5. Reporte de Clientes en el Sistema

En esta pestaña se observa todos los usuarios que se encuentran ingresados en el sistema ya estén activos o inactivos. La tabla es dinámica lo cual permite búsqueda por parámetro y ordenamiento ascendente o descendente por medio de la columna que se elija como se muestra en la imagen.



Nota: Si se da doble click encima de la compra el sistema redirige a la pantalla de edición.

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

1.3.6. Edición de Cliente.

The screenshot shows a web application window titled 'Sistema de Inventario'. The main menu includes 'SisInventario', 'Bodega', 'Inventario', 'Ventas', 'Administración', 'Reportes', 'Ver', 'Herramientas', 'Ventanas', and 'Ayuda'. The active window is 'Cliente', which has two tabs: 'Listado' and 'Ingreso'. The 'Ingreso' tab is selected, and the form is titled 'Clientes'. The form contains the following fields and values:

Código:	2	Apellidos:	TIPAN
Nombre:	KAROLINA	Fecha de Nacimiento:	08/06/2010
Sexo:	F	Tipo de Documento:	RIU 1712945698001
Dirección:	Carapungo		
Teléfono:	2456789		
E-mail:	karo@gmail.com		

At the bottom of the form, there are four buttons: 'Nuevo', 'Guardar', 'Editar', and 'Cancelar'.

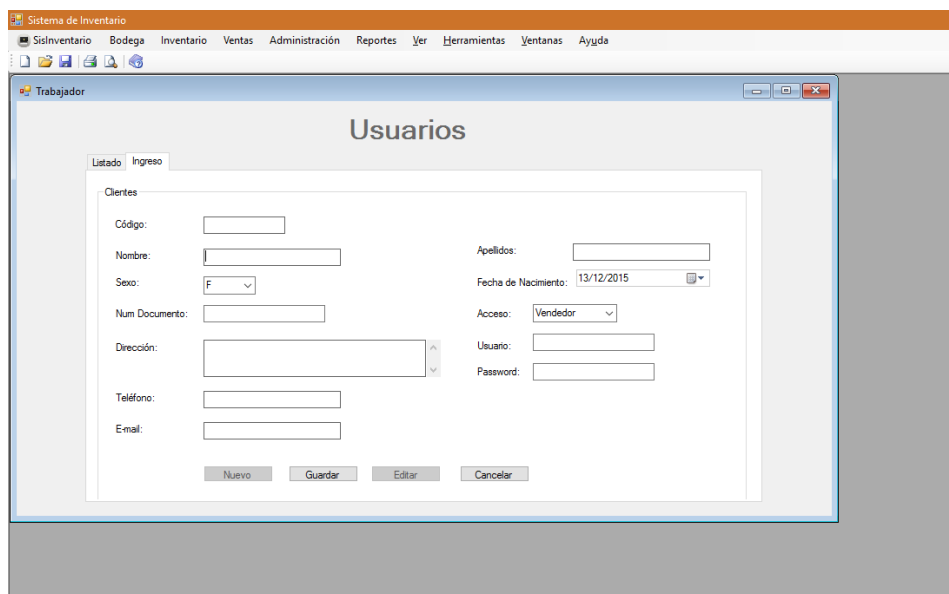
Esta pantalla, que es parecida a la de ingreso de cliente, se pueden editar los campos que se desee; menos el campo código ya que este es autogenerado por la base de datos.

Nota: Esta ventana tiene los mismos controles que la pestaña de ingreso de cliente.

1.4. Módulo Administración.

En este módulo se hace el ingreso, modificación y cambio de usuarios que hacen uso del sistema, manejando una lista de perfiles previamente instalado, con lo cual restringe a cada usuario cumplir con su rol. (Los nombres de usuario no se pueden repetir).

1.4.1. Ingreso Nuevo Usuario.



The screenshot shows a web-based application window titled 'Sistema de Inventario'. The main menu includes 'SisInventario', 'Bodega', 'Inventario', 'Ventas', 'Administración', 'Reportes', 'Ver', 'Herramientas', 'Ventanas', and 'Ayuda'. The active window is titled 'Trabajador' and contains a form titled 'Usuarios'. The form has two tabs: 'Listado' and 'Ingreso'. The 'Ingreso' tab is active, showing a form with the following fields: 'Código', 'Nombre', 'Apellidos', 'Sexo' (with a dropdown menu showing 'F'), 'Fecha de Nacimiento' (with a date picker showing '13/12/2015'), 'Num Documento', 'Acceso' (with a dropdown menu showing 'Vendedor'), 'Dirección', 'Usuario', 'Teléfono', 'Email', and 'Password'. At the bottom of the form are four buttons: 'Nuevo', 'Guardar', 'Editar', and 'Cancelar'.

Se llenan los campos correspondientes; todos los campos en este formulario son obligatorios; en el caso de que algún campo obligatorio falte se desplegara una alerta con un mensaje correspondiente al error.

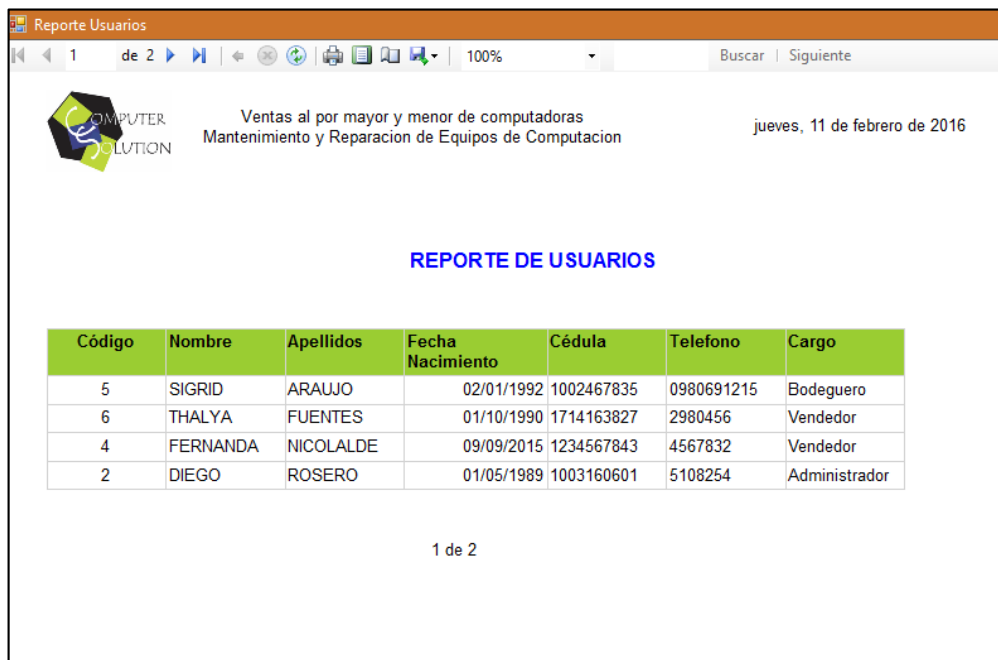
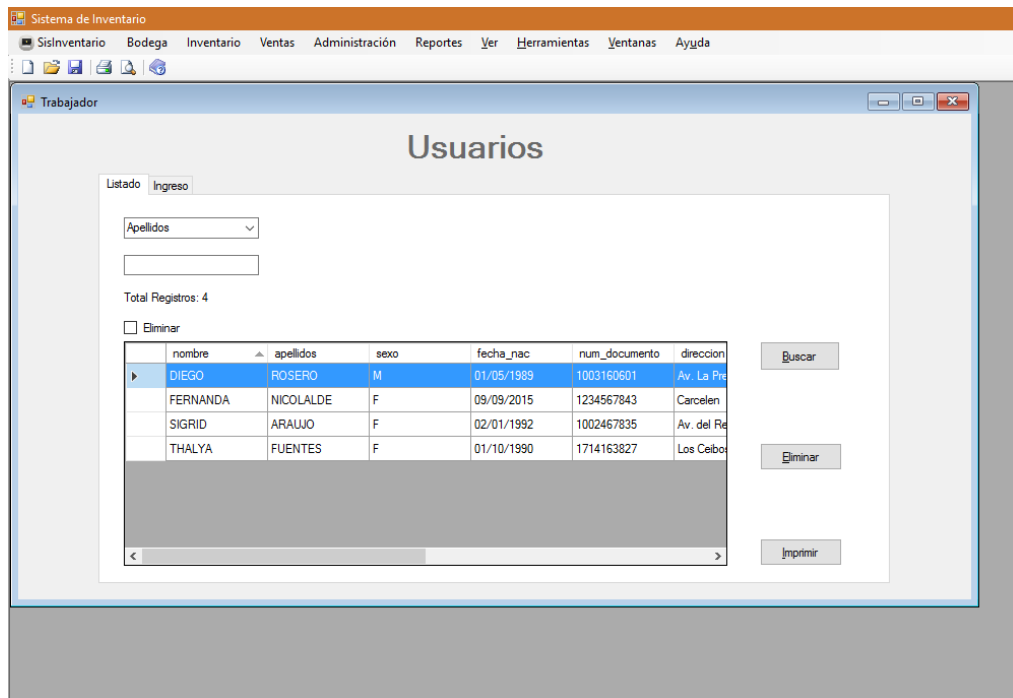
DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

The screenshot shows a web application window titled 'Sistema de Inventario' with a menu bar containing 'SisInventario', 'Bodega', 'Inventario', 'Ventas', 'Administración', 'Reportes', 'Ver', 'Herramientas', 'Ventanas', and 'Ayuda'. The main content area is a window titled 'Trabajador' with a sub-window titled 'Usuarios'. The 'Usuarios' window has two tabs: 'Listado' and 'Ingreso'. The 'Ingreso' tab is active, showing a form for adding a new user. The form is titled 'Cientes' and contains the following fields: 'Nombre: *', 'Apellidos: *', 'Sexo: F', 'Fecha de Nacimiento: 29/02/2016', 'Num Cedula: *', 'Acceso: Vendedor', 'Dirección: *', 'Usuario: *', 'Teléfono: *', 'Email: *', and 'Password: *'. A red message above the form states 'Los campos con * son obligatorios.' Below the form are four buttons: 'Nuevo', 'Guardar', 'Editar', and 'Cancelar'.

1.4.2. Reporte de Usuarios en el Sistema.

En esta pestaña se observa todos los usuarios que se encuentran ingresados en el sistema, activos o inactivos. La tabla es dinámica lo cual permite búsqueda por parámetro y ordenamiento ascendente o descendente por medio de la columna que se elija como se muestra en la imagen, para la obtención de un reporte de todos los usuarios que contiene el sistema solo se debe dar click en imprimir y el sistema automáticamente genera un reporte de todos los usuarios existentes.

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION



1.4.3. Edición de Usuarios.

The screenshot shows a web application window titled 'Sistema de Inventario' with a menu bar containing 'SisInventario', 'Bodega', 'Inventario', 'Ventas', 'Administración', 'Reportes', 'Ver', 'Herramientas', 'Ventanas', and 'Ayuda'. The main content area is titled 'Usuarios' and has two tabs: 'Listado' and 'Ingreso'. The 'Ingreso' tab is active, displaying a form for editing a user. The form is titled 'Clientes' and contains the following fields and values:

Código:	5	Apellidos:	ARAUJO
Nombre:	SIGRID	Fecha de Nacimiento:	02/01/1992
Sexo:	F	Acceso:	Bodeguero
Num Documento:	1002467835	Usuario:	bod1
Dirección:	Av. del Retrono 3-45 y Aguarico	Password:	bod1
Teléfono:	0980691215		
Email:	sigrid01@hotmail.com		

At the bottom of the form are four buttons: 'Nuevo', 'Guardar', 'Editar', and 'Cancelar'.

Esta pantalla, que es parecida a la de ingreso de usuario, se pueden editar los campos que se desee; menos el campo código ya que este es autogenerado por la base de datos, también sirve para cambiar o no la contraseña del usuario.

Nota: Esta ventana tiene los mismos controles que la pestaña de ingreso de usuario.

1.5. Módulo Reportes.

En este módulo se obtienen los reportes más esenciales del sistema, con uno solo click el vendedor o el bodeguero pueden tener acceso a ellos.

1.5.1. Ventas.

Al momento de hacer click en el botón Ventas automáticamente se genera un reporte de las mismas.

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

Sistema de Inventario

SisInventario Bodega Inventario Ventas Administración Reportes Ver Herramientas Ventanas Ayuda

Reporte Ventas

Ventas al por mayor y menor de computadoras
Mantenimiento y Reparación de Equipos de Computacion

sábado, 13 de febrero de 2016

REPORTE DE VENTAS

ID Ventas	Trabajador	Cliente	Fecha	Tipo comprobante	Total
3	ROSERO DIEGO	ROSERO DIEGO	16/12/2015 0:00:00	FACTURA	70,00
2	ROSERO DIEGO	ROSERO DIEGO	15/12/2015 0:00:00	FACTURA	80,00
1	ROSERO DIEGO	TIPAN KAROLINA	14/12/2015 0:00:00	TICKET	140,00

1 de 2

1.5.2. Compras.

Al momento de hacer click en el botón Compras automáticamente se genera un reporte de las mismas.

Sistema de Inventario

SisInventario Bodega Inventario Ventas Administración Reportes Ver Herramientas Ventanas Ayuda

PantallaReporteIngreso

Ventas al por mayor y menor de computadoras
Mantenimiento y Reparación de Equipos de Computacion

sábado, 13 de febrero de 2016

REPORTE DE INGRESOS

ID Ingreso	Trabajador	Proveedor	Fecha	Tipo comprobante	Estado	Total
16	ROSERO DIEGO	L.M.S CIA. LTDA.	14/12/2015	GUIA REMISION	EMITIDO	175,00
10	ROSERO DIEGO	TECNOMEGA	15/12/2015	GUIA REMISION	EMITIDO	400,00
5	ROSERO DIEGO	TECNOMEGA	12/12/2015	GUIA REMISION	EMITIDO	140,00
2	ROSERO DIEGO	TECNOMEGA	12/09/2015	FACTURA	EMITIDO	170,00
1	ROSERO DIEGO	TECNOMEGA	12/08/2015	FACTURA	EMITIDO	280,00

1 de 2

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

1.5.3. Stock Productos.

Al momento de hacer click en el botón Stock automáticamente se genera un reporte del mismo.

Sistema de Inventario

SisInventario Bodega Inventario Ventas Administración Reportes Ver Herramientas Ventanas Ayuda

Reporte de Stock

1 de 2

Buscar | Siguiente

COMPUTER SOLUTION

Ventas al por mayor y menor de computadoras
Mantenimiento y Reparacion de Equipos de Computacion

sábado, 13 de febrero de 2016

REPORTE DE STOCK

Código	Nombre	Descripción	Categoría
8965	LG	Monitor de 19" Pulgadas	MONITOR
0563	TOSHIBA	Core i7 - 2.2GHz - 6GB - 750Gb - DVDRw	PORTATILES
0067	MOUSE	Mouse Inalambrico	MOUSE
0123	CONTROL	Control Inalambrico Xbox	CONTROLES INALAMBRICOS
0289	1 TERA	Hitachi	DISCOS DUROS EXTERNOS

1 de 2

Anexo B

2. Manual Técnico

Debido a que se utiliza un framework con autogenerador de código estandarizado se incluye el código de desarrollo por capas, las más principales son: capa de Datos y capa del Negocio, en donde se realiza un énfasis de construcción de métodos y ocupación de los mismos, el código se lo inserta de manera esquemática para su entendimiento.

2.1. Capa de Acceso a Datos

2.1.1. Conexión

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CapaDatos
{
    class Conexion
    {
        public static string Cn = "Data Source=DIEGOR; Initial
Catalog=dbventas; Integrated Security=true";
    }
}
```

2.1.2. DArticulo

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data;
using System.Data.SqlClient;

namespace CapaDatos
{
    public class Articulo
    {
        private int _Idarticulo;
        private string _Codigo;
        private string _Nombre;
        private string _Descripcion;
        private byte[] _Imagen;
    }
}
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
private int _IdCategoria;
private string _TextoBuscar;

public int Idarticulo
{
    get
    {
        return _Idarticulo;
    }

    set
    {
        _Idarticulo = value;
    }
}

public string Codigo
{
    get
    {
        return _Codigo;
    }

    set
    {
        _Codigo = value;
    }
}

public string Nombre
{
    get
    {
        return _Nombre;
    }

    set
    {
        _Nombre = value;
    }
}

public string Descripcion
{
    get
    {
        return _Descripcion;
    }

    set
    {
        _Descripcion = value;
    }
}

public byte[] Imagen
{
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
        get
        {
            return _Imagen;
        }

        set
        {
            _Imagen = value;
        }
    }

    public int IdCategoria
    {
        get
        {
            return _IdCategoria;
        }

        set
        {
            _IdCategoria = value;
        }
    }

    public string TextoBuscar
    {
        get
        {
            return _TextoBuscar;
        }

        set
        {
            _TextoBuscar = value;
        }
    }

    public Articulo()
    { }

    public Articulo(int idarticulo, string codigo, string nombre, string
descripcion, byte[] imagen, int idCategoria, string textoBuscar)
    {
        this.Idarticulo = idarticulo;
        this.Codigo = codigo;
        this.Nombre = nombre;
        this.Descripcion = descripcion;
        this.Imagen = imagen;
        this.IdCategoria = IdCategoria;
        this.TextoBuscar = textoBuscar;
    }

    //Método Insertar

    public string Insertar(Articulo Articulo)
    {
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
string rpta = "";
SqlConnection SqlCon = new SqlConnection();
try
{
    //Código
    SqlCon.ConnectionString = Conexion.Cn;
    SqlCon.Open();
    //Establecer el Comando
    SqlCommand SqlCmd = new SqlCommand();
    SqlCmd.Connection = SqlCon;
    SqlCmd.CommandText = "spi_articulo";
    SqlCmd.CommandType = CommandType.StoredProcedure;

    SqlParameter IdArticulo = new SqlParameter();
    IdArticulo.ParameterName = "@Idarticulo";
    IdArticulo.SqlDbType = SqlDbType.Int;
    IdArticulo.Value = Articulo.Idarticulo;
    SqlCmd.Parameters.Add(IdArticulo);

    SqlParameter Codigo = new SqlParameter();
    Codigo.ParameterName = "@codigo";
    Codigo.SqlDbType = SqlDbType.VarChar;
    Codigo.Size = 250;
    Codigo.Value = Articulo.Codigo;
    SqlCmd.Parameters.Add(Codigo);

    SqlParameter Nombre = new SqlParameter();
    Nombre.ParameterName = "@nombre";
    Nombre.SqlDbType = SqlDbType.VarChar;
    Nombre.Size = 50;
    Nombre.Value = Articulo.Nombre;
    SqlCmd.Parameters.Add(Nombre);

    SqlParameter Descripcion = new SqlParameter();
    Descripcion.ParameterName = "@descripcion";
    Descripcion.SqlDbType = SqlDbType.VarChar;
    Descripcion.Size = 256;
    Descripcion.Value = Articulo.Descripcion;
    SqlCmd.Parameters.Add(Descripcion);

    SqlParameter Imagen = new SqlParameter();
    Imagen.ParameterName = "@imagen";
    Imagen.SqlDbType = SqlDbType.Image;
    Imagen.Value = Articulo.Imagen;
    SqlCmd.Parameters.Add(Imagen);

    SqlParameter IdCategoria = new SqlParameter();
    IdCategoria.ParameterName = "@idcategoria";
    IdCategoria.SqlDbType = SqlDbType.VarChar;
    IdCategoria.Value = Articulo.IdCategoria;
    SqlCmd.Parameters.Add(IdCategoria);

    //Ejecutamos nuestro comando

    rpta = SqlCmd.ExecuteNonQuery() == 1 ? "OK" : "Registro NO
    ingresado";
}
```

```
    }
    catch (Exception ex)
    {
        rpta = ex.Message;
    }
    finally
    {
        if (SqlCon.State == ConnectionState.Open) SqlCon.Close();
    }
    return rpta;
}

//Método Editar
public string Editar(Articulo Articulo)
{
    string rpta = "";
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        //Código
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCon.Open();
        //Establecer el Comando
        SqlCommand SqlCmd = new SqlCommand();
        SqlCmd.Connection = SqlCon;
        SqlCmd.CommandText = "spe_articulo";
        SqlCmd.CommandType = CommandType.StoredProcedure;

        SqlParameter IdArticulo = new SqlParameter();
        IdArticulo.ParameterName = "@Idarticulo";
        IdArticulo.SqlDbType = SqlDbType.Int;
        IdArticulo.Value = Articulo.Idarticulo;
        SqlCmd.Parameters.Add(IdArticulo);

        SqlParameter Codigo = new SqlParameter();
        Codigo.ParameterName = "@codigo";
        Codigo.SqlDbType = SqlDbType.VarChar;
        Codigo.Size = 250;
        Codigo.Value = Articulo.Codigo;
        SqlCmd.Parameters.Add(Codigo);

        SqlParameter Nombre = new SqlParameter();
        Nombre.ParameterName = "@nombre";
        Nombre.SqlDbType = SqlDbType.VarChar;
        Nombre.Size = 50;
        Nombre.Value = Articulo.Nombre;
        SqlCmd.Parameters.Add(Nombre);

        SqlParameter Descripcion = new SqlParameter();
        Descripcion.ParameterName = "@descripcion";
        Descripcion.SqlDbType = SqlDbType.VarChar;
        Descripcion.Size = 256;
        Descripcion.Value = Articulo.Descripcion;
        SqlCmd.Parameters.Add(Descripcion);
    }
    catch (Exception ex)
    {
        rpta = ex.Message;
    }
    finally
    {
        if (SqlCon.State == ConnectionState.Open) SqlCon.Close();
    }
    return rpta;
}
}
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
        SqlParameter Imagen = new SqlParameter();
        Imagen.ParameterName = "@imagen";
        Imagen.SqlDbType = SqlDbType.Image;
        Imagen.Value = Articulo.Imagen;
        SqlCommand.Parameters.Add(Imagen);

        SqlParameter IdCategoria = new SqlParameter();
        IdCategoria.ParameterName = "@idcategoria";
        IdCategoria.SqlDbType = SqlDbType.VarChar;
        IdCategoria.Value = Articulo.IdCategoria;
        SqlCommand.Parameters.Add(IdCategoria);

        //Ejecutamos nuestro comando

        rpta = SqlCommand.ExecuteNonQuery() == 1 ? "OK" : "Registro NO
actualizado";

    }
    catch (Exception ex)
    {
        rpta = ex.Message;
    }
    finally
    {
        if (SqlCon.State == ConnectionState.Open) SqlCon.Close();
    }
    return rpta;
}

//Método Eliminar
public string Eliminar(Articulo Articulo)
{
    string rpta = "";
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        //Código
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCon.Open();
        //Establecer el Comando
        SqlCommand SqlCommand = new SqlCommand();
        SqlCommand.Connection = SqlCon;
        SqlCommand.CommandText = "spel_articulo";
        SqlCommand.CommandType = CommandType.StoredProcedure;

        SqlParameter IdArticulo = new SqlParameter();
        IdArticulo.ParameterName = "@idarticulo";
        IdArticulo.SqlDbType = SqlDbType.Int;
        IdArticulo.Value = Articulo.Idarticulo;
        SqlCommand.Parameters.Add(IdArticulo);

        //Ejecutamos nuestro comando

        rpta = SqlCommand.ExecuteNonQuery() == 1 ? "OK" : "Registro NO
eliminado";
    }
}
```

```
    }
    catch (Exception ex)
    {
        rpta = ex.Message;
    }
    finally
    {
        if (SqlCon.State == ConnectionState.Open) SqlCon.Close();
    }
    return rpta;
}

//Método Mostrar
public DataTable Mostrar()
{
    DataTable DtResultado = new DataTable("articulo");
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCommand SqlCmd = new SqlCommand();
        SqlCmd.Connection = SqlCon;
        SqlCmd.CommandText = "spmostrar_articulo";
        SqlCmd.CommandType = CommandType.StoredProcedure;

        SqlDataAdapter SqlDat = new SqlDataAdapter(SqlCmd);
        SqlDat.Fill(DtResultado);
    }
    catch (Exception ex)
    {
        DtResultado = null;
    }
    return DtResultado;
}

//Método BuscarNombre
public DataTable BuscarNombre(Articulo Articulo)
{
    DataTable DtResultado = new DataTable("articulo");
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCommand SqlCmd = new SqlCommand();
        SqlCmd.Connection = SqlCon;
        SqlCmd.CommandText = "spb_articulo_nombre";
        SqlCmd.CommandType = CommandType.StoredProcedure;

        SqlParameter TextoBuscar = new SqlParameter();
        TextoBuscar.ParameterName = "@textobuscar";
        TextoBuscar.SqlDbType = SqlDbType.VarChar;
        TextoBuscar.Size = 50;
    }
}
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
        TextoBuscar.Value = Articulo.TextoBuscar;
        SqlCommand.Parameters.Add(TextoBuscar);

        SqlDataAdapter SqlDat = new SqlDataAdapter(SqlCmd);
        SqlDat.Fill(DtResultado);
    }
    catch (Exception ex)
    {
        DtResultado = null;
    }
    return DtResultado;
}
}
}
```

2.1.3. DCliente

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data;
using System.Data.SqlClient;

namespace CapaDatos
{
    public class Cliente
    {
        //Variables
        private int _Idcliente;
        private string _Nombre;
        private string _Apellidos;
        private string _Sexo;
        private DateTime _Fecha_Nacimiento;
        private string _Tipo_Documento;
        private string _Num_Documento;
        private string _Direccion;
        private string _Telefono;
        private string _Email;
        private string _TextoBuscar;

        public int Idcliente
        {
            get
            {
                return _Idcliente;
            }
            set
            {
                _Idcliente = value;
            }
        }
    }
}
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
    }

    public string Nombre
    {
        get
        {
            return _Nombre;
        }

        set
        {
            _Nombre = value;
        }
    }

    public string Apellidos
    {
        get
        {
            return _Apellidos;
        }

        set
        {
            _Apellidos = value;
        }
    }

    public string Sexo
    {
        get
        {
            return _Sexo;
        }

        set
        {
            _Sexo = value;
        }
    }

    public DateTime Fecha_Nacimiento
    {
        get
        {
            return _Fecha_Nacimiento;
        }

        set
        {
            _Fecha_Nacimiento = value;
        }
    }

    public string Tipo_Documento
    {
        get
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
    {
        return _Tipo_Documento;
    }

    set
    {
        _Tipo_Documento = value;
    }
}

public string Num_Documento
{
    get
    {
        return _Num_Documento;
    }

    set
    {
        _Num_Documento = value;
    }
}

public string Direccion
{
    get
    {
        return _Direccion;
    }

    set
    {
        _Direccion = value;
    }
}

public string Telefono
{
    get
    {
        return _Telefono;
    }

    set
    {
        _Telefono = value;
    }
}

public string Email
{
    get
    {
        return _Email;
    }

    set
```

```
        {
            _Email = value;
        }
    }

    public string TextoBuscar
    {
        get
        {
            return _TextoBuscar;
        }

        set
        {
            _TextoBuscar = value;
        }
    }

    //Constructores
    public Cliente()
    {
    }

    public Cliente(int idcliente, string nombre, string apellidos,
string sexo, DateTime fecha_nacimiento, string tipo_documento, string
num_documento, string direccion, string telefono, string email, string
textobuscar)
    {
        this.Idcliente = idcliente;
        this.Nombre = nombre;
        this.Apellidos = apellidos;
        this.Sexo = sexo;
        this.Fecha_Nacimiento = fecha_nacimiento;
        this.Tipo_Documento = tipo_documento;
        this.Num_Documento = num_documento;
        this.Direccion = direccion;
        this.Telefono = telefono;
        this.Email = email;
        this.TextoBuscar = textobuscar;
    }

    //Método Insertar

    public string Insertar(Cliente Cliente)
    {
        string rpt = "";
        SqlConnection SqlCon = new SqlConnection();
        try
        {
            //Código
            SqlCon.ConnectionString = Conexion.Cn;
            SqlCon.Open();
            //Establecer el Comando
            SqlCommand SqlCmd = new SqlCommand();
            SqlCmd.Connection = SqlCon;
            SqlCmd.CommandText = "spi_cliente";
        }
    }
}
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
SqlCommand.CommandType = CommandType.StoredProcedure;

SqlParameter ParIdcliente = new SqlParameter();
ParIdcliente.ParameterName = "@idcliente";
ParIdcliente.SqlDbType = SqlDbType.Int;
ParIdcliente.Direction = ParameterDirection.Output;
SqlCommand.Parameters.Add(ParIdcliente);

SqlParameter ParNombre = new SqlParameter();
ParNombre.ParameterName = "@nombre";
ParNombre.SqlDbType = SqlDbType.VarChar;
ParNombre.Size = 20;
ParNombre.Value = Cliente.Nombre;
SqlCommand.Parameters.Add(ParNombre);

SqlParameter ParApellidos = new SqlParameter();
ParApellidos.ParameterName = "@apellidos";
ParApellidos.SqlDbType = SqlDbType.VarChar;
ParApellidos.Size = 40;
ParApellidos.Value = Cliente.Apellidos;
SqlCommand.Parameters.Add(ParApellidos);

SqlParameter ParSexo = new SqlParameter();
ParSexo.ParameterName = "@sexo";
ParSexo.SqlDbType = SqlDbType.VarChar;
ParSexo.Size = 1;
ParSexo.Value = Cliente.Sexo;
SqlCommand.Parameters.Add(ParSexo);

SqlParameter ParFecha_Nacimiento = new SqlParameter();
ParFecha_Nacimiento.ParameterName = "@fecha_nacimiento";
ParFecha_Nacimiento.SqlDbType = SqlDbType.VarChar;
ParFecha_Nacimiento.Size = 40;
ParFecha_Nacimiento.Value = Cliente.Fecha_Nacimiento;
SqlCommand.Parameters.Add(ParFecha_Nacimiento);

SqlParameter ParTipo_Documento = new SqlParameter();
ParTipo_Documento.ParameterName = "@tipo_documento";
ParTipo_Documento.SqlDbType = SqlDbType.VarChar;
ParTipo_Documento.Size = 20;
ParTipo_Documento.Value = Cliente.Tipo_Documento;
SqlCommand.Parameters.Add(ParTipo_Documento);

SqlParameter ParNum_Documento = new SqlParameter();
ParNum_Documento.ParameterName = "@num_documento";
ParNum_Documento.SqlDbType = SqlDbType.VarChar;
ParNum_Documento.Size = 15;
ParNum_Documento.Value = Cliente.Num_Documento;
SqlCommand.Parameters.Add(ParNum_Documento);

SqlParameter ParDireccion = new SqlParameter();
ParDireccion.ParameterName = "@direccion";
ParDireccion.SqlDbType = SqlDbType.VarChar;
ParDireccion.Size = 100;
ParDireccion.Value = Cliente.Direccion;
SqlCommand.Parameters.Add(ParDireccion);
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
        SqlParameter ParTelefono = new SqlParameter();
        ParTelefono.ParameterName = "@telefono";
        ParTelefono.SqlDbType = SqlDbType.VarChar;
        ParTelefono.Size = 10;
        ParTelefono.Value = Cliente.Telefono;
        SqlCommand.Parameters.Add(ParTelefono);

        SqlParameter ParEmail = new SqlParameter();
        ParEmail.ParameterName = "@email";
        ParEmail.SqlDbType = SqlDbType.VarChar;
        ParEmail.Size = 50;
        ParEmail.Value = Cliente.Email;
        SqlCommand.Parameters.Add(ParEmail);

        //Ejecutamos nuestro comando
        rpta = SqlCommand.ExecuteNonQuery() == 1 ? "OK" : "NO se Ingreso
el Registro";
    }
    catch (Exception ex)
    {
        rpta = ex.Message;
    }
    finally
    {
        if (SqlCon.State == ConnectionState.Open) SqlCon.Close();
    }
    return rpta;
}

//Método Editar
public string Editar(Cliente Cliente)
{
    string rpta = "";
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        //Código
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCon.Open();
        //Establecer el Comando
        SqlCommand SqlCmd = new SqlCommand();
        SqlCmd.Connection = SqlCon;
        SqlCmd.CommandText = "sped_cliente";
        SqlCmd.CommandType = CommandType.StoredProcedure;

        SqlParameter ParIdcliente = new SqlParameter();
        ParIdcliente.ParameterName = "@idcliente";
        ParIdcliente.SqlDbType = SqlDbType.Int;
        ParIdcliente.Value = Cliente.Idcliente;
        SqlCommand.Parameters.Add(ParIdcliente);

        SqlParameter ParNombre = new SqlParameter();
        ParNombre.ParameterName = "@nombre";
        ParNombre.SqlDbType = SqlDbType.VarChar;
        ParNombre.Size = 20;
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
ParNombre.Value = Cliente.Nombre;
SqlCommand.Parameters.Add(ParNombre);

SqlParameter ParApellidos = new SqlParameter();
ParApellidos.ParameterName = "@apellidos";
ParApellidos.SqlDbType = SqlDbType.VarChar;
ParApellidos.Size = 40;
ParApellidos.Value = Cliente.Apellidos;
SqlCommand.Parameters.Add(ParApellidos);

SqlParameter ParSexo = new SqlParameter();
ParSexo.ParameterName = "@sexo";
ParSexo.SqlDbType = SqlDbType.VarChar;
ParSexo.Size = 1;
ParSexo.Value = Cliente.Sexo;
SqlCommand.Parameters.Add(ParSexo);

SqlParameter ParFecha_Nacimiento = new SqlParameter();
ParFecha_Nacimiento.ParameterName = "@fecha_nacimiento";
ParFecha_Nacimiento.SqlDbType = SqlDbType.VarChar;
ParFecha_Nacimiento.Size = 40;
ParFecha_Nacimiento.Value = Cliente.Fecha_Nacimiento;
SqlCommand.Parameters.Add(ParFecha_Nacimiento);

SqlParameter ParTipo_Documento = new SqlParameter();
ParTipo_Documento.ParameterName = "@tipo_documento";
ParTipo_Documento.SqlDbType = SqlDbType.VarChar;
ParTipo_Documento.Size = 20;
ParTipo_Documento.Value = Cliente.Tipo_Documento;
SqlCommand.Parameters.Add(ParTipo_Documento);

SqlParameter ParNum_Documento = new SqlParameter();
ParNum_Documento.ParameterName = "@num_documento";
ParNum_Documento.SqlDbType = SqlDbType.VarChar;
ParNum_Documento.Size = 15;
ParNum_Documento.Value = Cliente.Num_Documento;
SqlCommand.Parameters.Add(ParNum_Documento);

SqlParameter ParDireccion = new SqlParameter();
ParDireccion.ParameterName = "@direccion";
ParDireccion.SqlDbType = SqlDbType.VarChar;
ParDireccion.Size = 100;
ParDireccion.Value = Cliente.Direccion;
SqlCommand.Parameters.Add(ParDireccion);

SqlParameter ParTelefono = new SqlParameter();
ParTelefono.ParameterName = "@telefono";
ParTelefono.SqlDbType = SqlDbType.VarChar;
ParTelefono.Size = 10;
ParTelefono.Value = Cliente.Telefono;
SqlCommand.Parameters.Add(ParTelefono);

SqlParameter ParEmail = new SqlParameter();
ParEmail.ParameterName = "@email";
ParEmail.SqlDbType = SqlDbType.VarChar;
ParEmail.Size = 50;
ParEmail.Value = Cliente.Email;
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
        SqlCommand.Parameters.Add(ParEmail);

        //Ejecutamos nuestro comando

        rpta = SqlCommand.ExecuteNonQuery() == 1 ? "OK" : "NO se
Actualizo el Registro";
    }
    catch (Exception ex)
    {
        rpta = ex.Message;
    }
    finally
    {
        if (SqlConnection.State == ConnectionState.Open) SqlConnection.Close();
    }
    return rpta;
}

//Método Eliminar
public string Eliminar(Cliente Cliente)
{
    string rpta = "";
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        //Código
        SqlConnection.ConnectionString = Conexion.Cn;
        SqlConnection.Open();
        //Establecer el Comando
        SqlCommand SqlCommand = new SqlCommand();
        SqlCommand.Connection = SqlCon;
        SqlCommand.CommandText = "spel_cliente";
        SqlCommand.CommandType = CommandType.StoredProcedure;

        SqlParameter ParIdcliente = new SqlParameter();
        ParIdcliente.ParameterName = "@idcliente";
        ParIdcliente.SqlDbType = SqlDbType.Int;
        ParIdcliente.Value = Cliente.Idcliente;
        SqlCommand.Parameters.Add(ParIdcliente);
        //Ejecutamos nuestro comando

        rpta = SqlCommand.ExecuteNonQuery() == 1 ? "OK" : "NO se Elimino
el Registro";
    }
    catch (Exception ex)
    {
        rpta = ex.Message;
    }
    finally
    {
        if (SqlConnection.State == ConnectionState.Open) SqlConnection.Close();
    }
    return rpta;
}

//Método Mostrar
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
public DataTable Mostrar()
{
    DataTable DtResultado = new DataTable("cliente");
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCommand SqlCmd = new SqlCommand();
        SqlCmd.Connection = SqlCon;
        SqlCmd.CommandText = "spm_cliente";
        SqlCmd.CommandType = CommandType.StoredProcedure;

        SqlDataAdapter SqlDat = new SqlDataAdapter(SqlCmd);
        SqlDat.Fill(DtResultado);

    }
    catch (Exception ex)
    {
        DtResultado = null;
    }
    return DtResultado;
}

//Método BuscarApellidos
public DataTable BuscarApellidos(Cliente Cliente)
{
    DataTable DtResultado = new DataTable("cliente");
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCommand SqlCmd = new SqlCommand();
        SqlCmd.Connection = SqlCon;
        SqlCmd.CommandText = "spb_cliente_apellidos";
        SqlCmd.CommandType = CommandType.StoredProcedure;

        SqlParameter ParTextoBuscar = new SqlParameter();
        ParTextoBuscar.ParameterName = "@textobuscar";
        ParTextoBuscar.SqlDbType = SqlDbType.VarChar;
        ParTextoBuscar.Size = 50;
        ParTextoBuscar.Value = Cliente.TextoBuscar;
        SqlCmd.Parameters.Add(ParTextoBuscar);

        SqlDataAdapter SqlDat = new SqlDataAdapter(SqlCmd);
        SqlDat.Fill(DtResultado);

    }
    catch (Exception ex)
    {
        DtResultado = null;
    }
    return DtResultado;
}
```

```
//Método BuscarNum_Documento
public DataTable BuscarNum_Documento(Cliente Cliente)
{
    DataTable DtResultado = new DataTable("cliente");
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCommand SqlCmd = new SqlCommand();
        SqlCmd.Connection = SqlCon;
        SqlCmd.CommandText = "spb_cliente_num_documento";
        SqlCmd.CommandType = CommandType.StoredProcedure;

        SqlParameter ParTextoBuscar = new SqlParameter();
        ParTextoBuscar.ParameterName = "@textobuscar";
        ParTextoBuscar.SqlDbType = SqlDbType.VarChar;
        ParTextoBuscar.Size = 50;
        ParTextoBuscar.Value = Cliente.TextoBuscar;
        SqlCmd.Parameters.Add(ParTextoBuscar);

        SqlDataAdapter SqlDat = new SqlDataAdapter(SqlCmd);
        SqlDat.Fill(DtResultado);

    }
    catch (Exception ex)
    {
        DtResultado = null;
    }
    return DtResultado;
}
}
```

2.1.4. DCategoria

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

using System.Data;
using System.Data.SqlClient;

namespace CapaDatos
{
    public class DCategoria
    {
        private int _Idcategoria;
        private string _Nombre;
        private string _Descripcion;
    }
}
```

```
private string _TextoBuscar;

public int Idcategoria
{
    get { return _Idcategoria; }
    set { _Idcategoria = value; }
}

public string Nombre
{
    get { return _Nombre; }
    set { _Nombre = value; }
}

public string Descripcion
{
    get { return _Descripcion; }
    set { _Descripcion = value; }
}

public string TextoBuscar
{
    get { return _TextoBuscar; }
    set { _TextoBuscar = value; }
}

//Constructor Vacío
public DCategoria()
{
}

//Constructor con parámetros
public DCategoria(int idcategoria,string nombre,string
descripcion,string textobuscar)
{
    this.Idcategoria = idcategoria;
    this.Nombre = nombre;
    this.Descripcion = descripcion;
    this.TextoBuscar = textobuscar;
}

//Método Insertar
public string Insertar(DCategoria Categoria)
{
    string rpt = "";
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        //Código
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCon.Open();
        //Establecer el Comando
        SqlCommand SqlCmd = new SqlCommand();
    }
}
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
        SqlCommand.Connection = SqlCon;
        SqlCommand.CommandText = "spinsertar_categoria";
        SqlCommand.CommandType = CommandType.StoredProcedure;

        SqlParameter ParIdcategoria = new SqlParameter();
        ParIdcategoria.ParameterName = "@idcategoria";
        ParIdcategoria.SqlDbType = SqlDbType.Int;
        ParIdcategoria.Direction = ParameterDirection.Output;
        SqlCommand.Parameters.Add(ParIdcategoria);

        SqlParameter ParNombre = new SqlParameter();
        ParNombre.ParameterName = "@nombre";
        ParNombre.SqlDbType = SqlDbType.VarChar;
        ParNombre.Size = 50;
        ParNombre.Value = Categoria.Nombre;
        SqlCommand.Parameters.Add(ParNombre);

        SqlParameter ParDescripcion = new SqlParameter();
        ParDescripcion.ParameterName = "@descripcion";
        ParDescripcion.SqlDbType = SqlDbType.VarChar;
        ParDescripcion.Size = 256;
        ParDescripcion.Value = Categoria.Descripcion;
        SqlCommand.Parameters.Add(ParDescripcion);

        //Ejecutamos nuestro comando

        rpta = SqlCommand.ExecuteNonQuery() == 1 ? "OK" : "NO se Ingreso
el Registro";

    }
    catch (Exception ex)
    {
        rpta = ex.Message;
    }
    finally
    {
        if (SqlCon.State == ConnectionState.Open) SqlCon.Close();
    }
    return rpta;
}

//Método Editar
public string Editar(DCategoria Categoria)
{
    string rpta = "";
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        //Código
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCon.Open();
        //Establecer el Comando
        SqlCommand SqlCommand = new SqlCommand();
        SqlCommand.Connection = SqlCon;
        SqlCommand.CommandText = "speditar_categoria";
    }
    catch (Exception ex)
    {
        rpta = ex.Message;
    }
    finally
    {
        if (SqlCon.State == ConnectionState.Open) SqlCon.Close();
    }
    return rpta;
}
}
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
        SqlCommand.CommandType = CommandType.StoredProcedure;

        SqlParameter ParIdcategoria = new SqlParameter();
        ParIdcategoria.ParameterName = "@idcategoria";
        ParIdcategoria.SqlDbType = SqlDbType.Int;
        ParIdcategoria.Value = Categoria.Idcategoria;
        SqlCommand.Parameters.Add(ParIdcategoria);

        SqlParameter ParNombre = new SqlParameter();
        ParNombre.ParameterName = "@nombre";
        ParNombre.SqlDbType = SqlDbType.VarChar;
        ParNombre.Size = 50;
        ParNombre.Value = Categoria.Nombre;
        SqlCommand.Parameters.Add(ParNombre);

        SqlParameter ParDescripcion = new SqlParameter();
        ParDescripcion.ParameterName = "@descripcion";
        ParDescripcion.SqlDbType = SqlDbType.VarChar;
        ParDescripcion.Size = 256;
        ParDescripcion.Value = Categoria.Descripcion;
        SqlCommand.Parameters.Add(ParDescripcion);

        //Ejecutamos nuestro comando

        rpta = SqlCommand.ExecuteNonQuery() == 1 ? "OK" : "NO se
Actualizo el Registro";

    }
    catch (Exception ex)
    {
        rpta = ex.Message;
    }
    finally
    {
        if (SqlCon.State == ConnectionState.Open) SqlCon.Close();
    }
    return rpta;
}

//Método Eliminar
public string Eliminar(DCategoria Categoria)
{
    string rpta = "";
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        //Código
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCon.Open();
        //Establecer el Comando
        SqlCommand SqlCmd = new SqlCommand();
        SqlCmd.Connection = SqlCon;
        SqlCmd.CommandText = "speliminar_categoria";
        SqlCmd.CommandType = CommandType.StoredProcedure;

        SqlParameter ParIdcategoria = new SqlParameter();
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
        ParIdcategoria.ParameterName = "@idcategoria";
        ParIdcategoria.SqlDbType = SqlDbType.Int;
        ParIdcategoria.Value = Categoria.Idcategoria;
        SqlCommand.Parameters.Add(ParIdcategoria);

        //Ejecutamos nuestro comando

        rpta = SqlCommand.ExecuteNonQuery() == 1 ? "OK" : "NO se Elimino
el Registro";

    }
    catch (Exception ex)
    {
        rpta = ex.Message;
    }
    finally
    {
        if (SqlCon.State == ConnectionState.Open) SqlCon.Close();
    }
    return rpta;
}

//Método Mostrar
public DataTable Mostrar()
{
    DataTable DtResultado = new DataTable("categoria");
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCommand SqlCmd = new SqlCommand();
        SqlCmd.Connection = SqlCon;
        SqlCmd.CommandText = "spmostrar_categoria";
        SqlCmd.CommandType = CommandType.StoredProcedure;

        SqlDataAdapter SqlDat = new SqlDataAdapter(SqlCmd);
        SqlDat.Fill(DtResultado);

    }
    catch (Exception ex)
    {
        DtResultado = null;
    }
    return DtResultado;
}

//Método BuscarNombre
public DataTable BuscarNombre (DCategoria Categoria)
{
    DataTable DtResultado = new DataTable("categoria");
    SqlConnection SqlCon = new SqlConnection();
    try
    {
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCommand SqlCmd = new SqlCommand();
        SqlCmd.Connection = SqlCon;
        SqlCmd.CommandText = "spbuscar_categoria";
        SqlCmd.CommandType = CommandType.StoredProcedure;

        SqlParameter ParTextoBuscar = new SqlParameter();
        ParTextoBuscar.ParameterName = "@textobuscar";
        ParTextoBuscar.SqlDbType = SqlDbType.VarChar;
        ParTextoBuscar.Size = 50;
        ParTextoBuscar.Value = Categoria.TextoBuscar;
        SqlCmd.Parameters.Add(ParTextoBuscar);

        SqlDataAdapter SqlDat = new SqlDataAdapter(SqlCmd);
        SqlDat.Fill(DtResultado);
    }
    catch (Exception ex)
    {
        DtResultado = null;
    }
    return DtResultado;
}
}
```

2.1.5. DDetalle_Ingreso

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data;
using System.Data.SqlClient;

namespace CapaDatos
{
    public class Detalle_Ingreso
    {
        //Variables
        private int _Iddetalle_ingreso;
        private int _Idingreso;
        private int _Idarticulo;
        private decimal _Precio_Compra;
        private decimal _precio_Venta;
        private int _Stock_Inicial;
        private int _Stock_Actual;

        //Propiedades
        public int Iddetalle_ingreso
        {
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
        get { return _Iddetalle_ingreso; }
        set { _Iddetalle_ingreso = value; }
    }

    public int Idingreso
    {
        get { return _Idingreso; }
        set { _Idingreso = value; }
    }

    public int Idarticulo
    {
        get { return _Idarticulo; }
        set { _Idarticulo = value; }
    }

    public decimal Precio_Compra
    {
        get { return _Precio_Compra; }
        set { _Precio_Compra = value; }
    }

    public decimal Precio_Venta
    {
        get { return _precio_Venta; }
        set { _precio_Venta = value; }
    }

    public int Stock_Inicial
    {
        get { return _Stock_Inicial; }
        set { _Stock_Inicial = value; }
    }

    public int Stock_Actual
    {
        get { return _Stock_Actual; }
        set { _Stock_Actual = value; }
    }

    //Constructores
    public Detalle_Ingreso()
    {
    }

    public Detalle_Ingreso(int iddetalle_ingreso, int idingreso, int
idarticulo, decimal precio_compra, decimal precio_venta, int stock_inicial,
int stock_actual)
    {
        this.Iddetalle_ingreso = iddetalle_ingreso;
        this.Idingreso = idingreso;
        this.Idarticulo = idarticulo;
        this.Precio_Compra = precio_compra;
        this.Precio_Venta = precio_venta;
        this.Stock_Inicial = stock_inicial;
        this.Stock_Actual = stock_actual;
    }
}
```

```
}  
  
//Método Insertar  
public string Insertar(Detalle_Ingreso Detalle_Ingreso, ref  
SqlConnection SqlCon, ref SqlTransaction SqlTra)  
{  
    string rpta = "";  
    try  
    {  
  
        SqlCommand SqlCmd = new SqlCommand();  
        SqlCmd.Connection = SqlCon;  
        SqlCmd.Transaction = SqlTra;  
        SqlCmd.CommandText = "spinsertar_detalle_ingreso";  
        SqlCmd.CommandType = CommandType.StoredProcedure;  
  
        SqlParameter ParIddetalle_ingreso = new SqlParameter();  
        ParIddetalle_ingreso.ParameterName = "@iddetalle_ingreso";  
        ParIddetalle_ingreso.SqlDbType = SqlDbType.Int;  
        ParIddetalle_ingreso.Direction = ParameterDirection.Output;  
        SqlCmd.Parameters.Add(ParIddetalle_ingreso);  
  
        SqlParameter ParIdingreso = new SqlParameter();  
        ParIdingreso.ParameterName = "@idingreso";  
        ParIdingreso.SqlDbType = SqlDbType.Int;  
        ParIdingreso.Value = Detalle_Ingreso.Idingreso;  
        SqlCmd.Parameters.Add(ParIdingreso);  
  
        SqlParameter ParIdarticulo = new SqlParameter();  
        ParIdarticulo.ParameterName = "@idarticulo";  
        ParIdarticulo.SqlDbType = SqlDbType.Int;  
        ParIdarticulo.Value = Detalle_Ingreso.Idarticulo;  
        SqlCmd.Parameters.Add(ParIdarticulo);  
  
        SqlParameter ParPrecio_Compra = new SqlParameter();  
        ParPrecio_Compra.ParameterName = "@precio_compra";  
        ParPrecio_Compra.SqlDbType = SqlDbType.Decimal;  
        ParPrecio_Compra.Precision = 4;  
        ParPrecio_Compra.Scale = 2;  
        ParPrecio_Compra.Value = Detalle_Ingreso.Precio_Compra;  
        SqlCmd.Parameters.Add(ParPrecio_Compra);  
  
        SqlParameter ParPrecio_Venta = new SqlParameter();  
        ParPrecio_Venta.ParameterName = "@precio_venta";  
        ParPrecio_Venta.SqlDbType = SqlDbType.Decimal;  
        ParPrecio_Venta.Precision = 4;  
        ParPrecio_Venta.Scale = 2;  
        ParPrecio_Venta.Value = Detalle_Ingreso.Precio_Venta;  
        SqlCmd.Parameters.Add(ParPrecio_Venta);  
  
        SqlParameter ParStock_Inicial = new SqlParameter();  
        ParStock_Inicial.ParameterName = "@stock_inicial";  
        ParStock_Inicial.SqlDbType = SqlDbType.Int;  
        ParStock_Inicial.Value = Detalle_Ingreso.Stock_Inicial;  
        SqlCmd.Parameters.Add(ParStock_Inicial);  
    }  
    }  
}
```

```
        SqlParameter ParStock_Actual = new SqlParameter();
        ParStock_Actual.ParameterName = "@stock_actual";
        ParStock_Actual.SqlDbType = SqlDbType.Int;
        ParStock_Actual.Value = Detalle_Ingreso.Stock_Actual;
        SqlCommand.Parameters.Add(ParStock_Actual);

        //Ejecutamos nuestro comando
        rpta = SqlCommand.ExecuteNonQuery() == 1 ? "OK" : "NO se Ingreso
el Registro";
    }
    catch (Exception ex)
    {
        rpta = ex.Message;
    }

    return rpta;
}
}
```

2.1.6. DDetalle_Venta

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data;
using System.Data.SqlClient;

namespace CapaDatos
{
    public class Detalle_Venta
    {
        private int _Iddetalle_venta;
        private int _Idventa;
        private int _Iddetalle_Ingreso;
        private int _Cantidad;
        private decimal _Precio_Venta;
        private decimal _Descuento;
        public int Iddetalle_venta
        {
            get { return _Iddetalle_venta; }
        }
    }
}
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
        set { _Iddetalle_venta = value; }
    }

    public int Idventa
    {
        get { return _Idventa; }
        set { _Idventa = value; }
    }

    public int Iddetalle_Ingreso
    {
        get { return _Iddetalle_Ingreso; }
        set { _Iddetalle_Ingreso = value; }
    }

    public int Cantidad
    {
        get { return _Cantidad; }
        set { _Cantidad = value; }
    }

    public decimal Precio_Venta
    {
        get { return _Precio_Venta; }
        set { _Precio_Venta = value; }
    }

    public decimal Descuento
    {
        get { return _Descuento; }
        set { _Descuento = value; }
    }
    //Constructores
    public Detalle_Venta()
    {
    }

    public Detalle_Venta(int iddetalle_venta, int idventa, int
iddetalle_ingreso, int cantidad, decimal precio_venta, decimal descuento)
    {
        this.Iddetalle_venta = iddetalle_venta;
        this.Idventa = idventa;
        this.Iddetalle_Ingreso = iddetalle_ingreso;
        this.Cantidad = cantidad;
        this.Precio_Venta = precio_venta;
        this.Descuento = descuento;
    }

    //Método Insertar
    public string Insertar(Detalle_Venta Detalle_Venta, ref
SqlConnection SqlCon, ref SqlTransaction SqlTra)
    {
        string rpta = "";
        try
```

```
{  
  
    SqlCommand SqlCmd = new SqlCommand();  
    SqlCmd.Connection = SqlCon;  
    SqlCmd.Transaction = SqlTra;  
    SqlCmd.CommandText = "spinsertar_detalle_venta";  
    SqlCmd.CommandType = CommandType.StoredProcedure;  
  
    SqlParameter ParIddetalle_venta = new SqlParameter();  
    ParIddetalle_venta.ParameterName = "@iddetalle_venta";  
    ParIddetalle_venta.SqlDbType = SqlDbType.Int;  
    ParIddetalle_venta.Direction = ParameterDirection.Output;  
    SqlCmd.Parameters.Add(ParIddetalle_venta);  
  
    SqlParameter ParIdventa = new SqlParameter();  
    ParIdventa.ParameterName = "@idventa";  
    ParIdventa.SqlDbType = SqlDbType.Int;  
    ParIdventa.Value = Detalle_Venta.Idventa;  
    SqlCmd.Parameters.Add(ParIdventa);  
  
    SqlParameter ParIddetalle_ingreso = new SqlParameter();  
    ParIddetalle_ingreso.ParameterName = "@iddetalle_ingreso";  
    ParIddetalle_ingreso.SqlDbType = SqlDbType.Int;  
    ParIddetalle_ingreso.Value =  
Detalle_Venta.Iddetalle_Ingreso;  
    SqlCmd.Parameters.Add(ParIddetalle_ingreso);  
  
    SqlParameter ParCantidad = new SqlParameter();  
    ParCantidad.ParameterName = "@cantidad";  
    ParCantidad.SqlDbType = SqlDbType.Int;  
    ParCantidad.Value = Detalle_Venta.Cantidad;  
    SqlCmd.Parameters.Add(ParCantidad);  
  
    SqlParameter ParPrecio_Venta = new SqlParameter();  
    ParPrecio_Venta.ParameterName = "@precio_venta";  
    ParPrecio_Venta.SqlDbType = SqlDbType.Money;  
    ParPrecio_Venta.Value = Detalle_Venta.Precio_Venta;  
    SqlCmd.Parameters.Add(ParPrecio_Venta);  
  
    SqlParameter ParDescuento = new SqlParameter();  
    ParDescuento.ParameterName = "@descuento";  
    ParDescuento.SqlDbType = SqlDbType.Money;  
    ParDescuento.Value = Detalle_Venta.Descuento;  
    SqlCmd.Parameters.Add(ParDescuento);  
  
    //Ejecutamos nuestro comando  
    rpta = SqlCmd.ExecuteNonQuery() == 1 ? "OK" :  
Convert.ToString(Idventa) + Convert.ToString(Iddetalle_Ingreso) +  
Convert.ToString(Cantidad) + Convert.ToString(Precio_Venta) +  
Convert.ToString(Descuento);  
  
    }  
    catch (Exception ex)  
    {  
        rpta = ex.Message;  
    }  
}
```

```
        return rptA;  
    }  
}
```

2.1.7. DIngreso

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Data;  
using System.Data.SqlClient;  
  
namespace CapaDatos  
{  
    public class Ingreso  
    {  
        //Variables  
        private int _Idingreso;  
        private int _Idproveedor;  
        private int _Idtrabajador;  
        private DateTime _Fecha;  
        private string _Tipo_Comprobante;  
        private string _Serie;  
        private string _Correlativo;  
        private decimal _Iva;  
        private string _Estado;  
  
        //Propiedades  
        public int Idingreso  
        {  
            get { return _Idingreso; }  
            set { _Idingreso = value; }  
        }  
  
        public int Idproveedor  
        {  
            get { return _Idproveedor; }  
            set { _Idproveedor = value; }  
        }  
  
        public int Idtrabajador  
        {  
            get { return _Idtrabajador; }  
            set { _Idtrabajador = value; }  
        }  
  
        public DateTime Fecha  
        {  
            get { return _Fecha; }  
        }  
    }  
}
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
        set { _Fecha = value; }
    }

    public string Tipo_Comprobante
    {
        get { return _Tipo_Comprobante; }
        set { _Tipo_Comprobante = value; }
    }

    public string Serie
    {
        get { return _Serie; }
        set { _Serie = value; }
    }

    public string Correlativo
    {
        get { return _Correlativo; }
        set { _Correlativo = value; }
    }

    public decimal Iva
    {
        get { return _Iva; }
        set { _Iva = value; }
    }

    public string Estado
    {
        get { return _Estado; }
        set { _Estado = value; }
    }

    //Constructores
    public Ingreso()
    {
    }

    public Ingreso(int idingreso, int idtrabajador, int idproveedor,
DateTime fecha, string tipo_comprobante, string serie, string correlativo,
decimal iva, string estado)
    {
        this.Idingreso = idingreso;
        this.Idtrabajador = idtrabajador;
        this.Idproveedor = idproveedor;
        this.Fecha = fecha;
        this.Tipo_Comprobante = tipo_comprobante;
        this.Serie = serie;
        this.Correlativo = correlativo;
        this.Iva = iva;
        this.Estado = estado;
    }

    //Métodos
    public string Insertar(Ingreso Ingreso, List<Detalle_Ingreso>
Detalles)
    {
        string rpt = "";
```

```
SqlConnection SqlCon = new SqlConnection();
try
{
    //Código
    SqlCon.ConnectionString = Conexion.Cn;
    SqlCon.Open();
    //Establecer la transacción
    SqlTransaction SqlTra = SqlCon.BeginTransaction();
    //Establecer el Comando
    SqlCommand SqlCmd = new SqlCommand();
    SqlCmd.Connection = SqlCon;
    SqlCmd.Transaction = SqlTra;
    SqlCmd.CommandText = "spinsertar_ingreso";
    SqlCmd.CommandType = CommandType.StoredProcedure;

    //Parámetros
    SqlParameter ParIdingreso = new SqlParameter();
    ParIdingreso.ParameterName = "@idingreso";
    ParIdingreso.SqlDbType = SqlDbType.Int;
    ParIdingreso.Direction = ParameterDirection.Output;
    SqlCmd.Parameters.Add(ParIdingreso);

    SqlParameter ParIdTrabajador = new SqlParameter();
    ParIdTrabajador.ParameterName = "@idtrabajador";
    ParIdTrabajador.SqlDbType = SqlDbType.Int;
    ParIdTrabajador.Value = Ingreso.Idtrabajador;
    SqlCmd.Parameters.Add(ParIdTrabajador);

    SqlParameter ParIdproveedor = new SqlParameter();
    ParIdproveedor.ParameterName = "@idproveedor";
    ParIdproveedor.SqlDbType = SqlDbType.Int;
    ParIdproveedor.Value = Ingreso.Idproveedor;
    SqlCmd.Parameters.Add(ParIdproveedor);

    SqlParameter ParFecha = new SqlParameter();
    ParFecha.ParameterName = "@fecha";
    ParFecha.SqlDbType = SqlDbType.Date;
    ParFecha.Value = Ingreso.Fecha;
    SqlCmd.Parameters.Add(ParFecha);

    SqlParameter ParTipo_Comprobante = new SqlParameter();
    ParTipo_Comprobante.ParameterName = "@tipo_comprobante";
    ParTipo_Comprobante.SqlDbType = SqlDbType.VarChar;
    ParTipo_Comprobante.Size = 20;
    ParTipo_Comprobante.Value = Ingreso.Tipo_Comprobante;
    SqlCmd.Parameters.Add(ParTipo_Comprobante);

    SqlParameter ParSerie = new SqlParameter();
    ParSerie.ParameterName = "@serie";
    ParSerie.SqlDbType = SqlDbType.VarChar;
    ParSerie.Size = 4;
    ParSerie.Value = Ingreso.Serie;
    SqlCmd.Parameters.Add(ParSerie);

    SqlParameter ParCorrelativo = new SqlParameter();
    ParCorrelativo.ParameterName = "@correlativo";
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
ParCorrelativo.SqlDbType = SqlDbType.VarChar;
ParCorrelativo.Size = 7;
ParCorrelativo.Value = Ingreso.Correlativo;
SqlCommand.Parameters.Add(ParCorrelativo);

SqlParameter ParIva = new SqlParameter();
ParIva.ParameterName = "@iva";
ParIva.SqlDbType = SqlDbType.Decimal;
ParIva.Precision = 4;
ParIva.Scale = 2;
ParIva.Value = Ingreso.Iva;
SqlCommand.Parameters.Add(ParIva);

SqlParameter ParEstado = new SqlParameter();
ParEstado.ParameterName = "@estado";
ParEstado.SqlDbType = SqlDbType.VarChar;
ParEstado.Size = 7;
ParEstado.Value = Ingreso.Estado;
SqlCommand.Parameters.Add(ParEstado);

//Ejecutamos nuestro comando
el Registro";
rpta = SqlCommand.ExecuteNonQuery() == 1 ? "OK" : "NO se Ingreso
base de datos
if (rpta.Equals("OK"))
{
    //Obtenemos el codigo del ingreso que se genero por la
    this.Idingreso =
Convert.ToInt32(SqlCmd.Parameters["@idingreso"].Value);
    foreach (Detalle_Ingreso det in Detalles)
    {
        //Establecemos el codigo del ingreso que se
        autogenero
        det.Idingreso = this.Idingreso;
        //Llamamos al metodo insertar de la clase
        DetalleIngreso
        //y le pasamos la conexion y la transaccion que debe
        de usar
        rpta = det.Insertar(det, ref SqlCon, ref SqlTra);
        if (!rpta.Equals("OK"))
        {
            //Si ocurre un error al insertar un detalle de
            ingreso salimos del for
            break;
        }
    }
}
if (rpta.Equals("OK"))
{
    //Se inserto todo los detalles y confirmamos la
    transaccion
    SqlTra.Commit();
}
else
{
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
        //Algun detalle no se inserto y negamos la transaccion
        SqlTra.Rollback();
    }

}

catch (Exception ex)
{
    rpta = ex.Message;
}
finally
{
    if (SqlCon.State == ConnectionState.Open) SqlCon.Close();
}
return rpta;
}

public string Anular(Ingreso Ingreso)
{
    string rpta = "";
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        //Código
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCon.Open();
        //Establecer el Comando
        SqlCommand SqlCmd = new SqlCommand();
        SqlCmd.Connection = SqlCon;
        SqlCmd.CommandText = "spanular_ingreso";
        SqlCmd.CommandType = CommandType.StoredProcedure;

        SqlParameter ParIdingreso = new SqlParameter();
        ParIdingreso.ParameterName = "@idingreso";
        ParIdingreso.SqlDbType = SqlDbType.Int;
        ParIdingreso.Value = Ingreso.Idingreso;
        SqlCmd.Parameters.Add(ParIdingreso);
        //Ejecutamos nuestro comando

        rpta = SqlCmd.ExecuteNonQuery() == 1 ? "OK" : "NO se anulo
el Ingreso";
    }
    catch (Exception ex)
    {
        rpta = ex.Message;
    }
    finally
    {
        if (SqlCon.State == ConnectionState.Open) SqlCon.Close();
    }
    return rpta;
}

//Método Mostrar
public DataTable Mostrar()
```

```
{
    DataTable DtResultado = new DataTable("ingreso");
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCommand SqlCmd = new SqlCommand();
        SqlCmd.Connection = SqlCon;
        SqlCmd.CommandText = "spmostrar_ingreso";
        SqlCmd.CommandType = CommandType.StoredProcedure;

        SqlDataAdapter SqlDat = new SqlDataAdapter(SqlCmd);
        SqlDat.Fill(DtResultado);

    }
    catch (Exception ex)
    {
        DtResultado = null;
    }
    return DtResultado;
}

//Método BuscarFechas
public DataTable BuscarFechas(String TextoBuscar, String
TextoBuscar2)
{
    DataTable DtResultado = new DataTable("ingreso");
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCommand SqlCmd = new SqlCommand();
        SqlCmd.Connection = SqlCon;
        SqlCmd.CommandText = "spbuscar_ingreso_fecha";
        SqlCmd.CommandType = CommandType.StoredProcedure;

        SqlParameter ParTextoBuscar = new SqlParameter();
        ParTextoBuscar.ParameterName = "@textobuscar";
        ParTextoBuscar.SqlDbType = SqlDbType.VarChar;
        ParTextoBuscar.Size = 50;
        ParTextoBuscar.Value = TextoBuscar;
        SqlCmd.Parameters.Add(ParTextoBuscar);

        SqlParameter ParTextoBuscar2 = new SqlParameter();
        ParTextoBuscar2.ParameterName = "@textobuscar2";
        ParTextoBuscar2.SqlDbType = SqlDbType.VarChar;
        ParTextoBuscar2.Size = 50;
        ParTextoBuscar2.Value = TextoBuscar2;
        SqlCmd.Parameters.Add(ParTextoBuscar2);

        SqlDataAdapter SqlDat = new SqlDataAdapter(SqlCmd);
        SqlDat.Fill(DtResultado);

    }
    catch (Exception ex)
```

```
        {
            DtResultado = null;
        }
        return DtResultado;
    }

    //Método BuscarFechas
    public DataTable MostrarDetalle(String TextoBuscar)
    {
        DataTable DtResultado = new DataTable("detalle_ingreso");
        SqlConnection SqlCon = new SqlConnection();
        try
        {
            SqlCon.ConnectionString = Conexion.Cn;
            SqlCommand SqlCmd = new SqlCommand();
            SqlCmd.Connection = SqlCon;
            SqlCmd.CommandText = "spmostrar_detalle_ingreso";
            SqlCmd.CommandType = CommandType.StoredProcedure;

            SqlParameter ParTextoBuscar = new SqlParameter();
            ParTextoBuscar.ParameterName = "@textobuscar";
            ParTextoBuscar.SqlDbType = SqlDbType.VarChar;
            ParTextoBuscar.Size = 50;
            ParTextoBuscar.Value = TextoBuscar;
            SqlCmd.Parameters.Add(ParTextoBuscar);

            SqlDataAdapter SqlDat = new SqlDataAdapter(SqlCmd);
            SqlDat.Fill(DtResultado);

        }
        catch (Exception ex)
        {
            DtResultado = null;
        }
        return DtResultado;
    }
}
```

2.1.8. DPresentacion

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data;
using System.Data.SqlClient;

namespace CapaDatos
{
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
public class Presentacion
{
    private int _IdPresentacion;
    private string _Nombre;
    private string _Descripcion;
    private string _TextoBuscar;

    public int IdPresentacion
    {
        get
        {
            return _IdPresentacion;
        }

        set
        {
            _IdPresentacion = value;
        }
    }

    public string Nombre
    {
        get
        {
            return _Nombre;
        }

        set
        {
            _Nombre = value;
        }
    }

    public string Descripcion
    {
        get
        {
            return _Descripcion;
        }

        set
        {
            _Descripcion = value;
        }
    }

    public string TextoBuscar
    {
        get
        {
            return _TextoBuscar;
        }

        set
        {
            _TextoBuscar = value;
        }
    }
}
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
    }

    public Presentacion()
    {
    }

    public Presentacion(int idpresentacion, string nombre, string
descripcion, string textobuscar)
    {
        this.IdPresentacion = idpresentacion;
        this.Nombre = nombre;
        this.Descripcion = descripcion;
        this.TextoBuscar = textobuscar;
    }

    //Método Insertar

    public string Insertar(Presentacion Presentacion)
    {
        string rpta = "";
        SqlConnection SqlCon = new SqlConnection();
        try
        {
            //Código
            SqlCon.ConnectionString = Conexion.Cn;
            SqlCon.Open();
            //Establecer el Comando
            SqlCommand SqlCmd = new SqlCommand();
            SqlCmd.Connection = SqlCon;
            SqlCmd.CommandText = "spinsertar_presentacion";
            SqlCmd.CommandType = CommandType.StoredProcedure;

            SqlParameter Idpresentacion = new SqlParameter();
            Idpresentacion.ParameterName = "@idpresentacion";
            Idpresentacion.SqlDbType = SqlDbType.Int;
            Idpresentacion.Direction = ParameterDirection.Output;
            SqlCmd.Parameters.Add(Idpresentacion);

            SqlParameter Nombre = new SqlParameter();
            Nombre.ParameterName = "@nombre";
            Nombre.SqlDbType = SqlDbType.VarChar;
            Nombre.Size = 50;
            Nombre.Value = Presentacion.Nombre;
            SqlCmd.Parameters.Add(Nombre);

            SqlParameter Descripcion = new SqlParameter();
            Descripcion.ParameterName = "@descripcion";
            Descripcion.SqlDbType = SqlDbType.VarChar;
            Descripcion.Size = 256;
            Descripcion.Value = Presentacion.Descripcion;
            SqlCmd.Parameters.Add(Descripcion);

            //Ejecutamos nuestro comando

            rpta = SqlCmd.ExecuteNonQuery() == 1 ? "OK" : "Registro NO
ingresado";
        }
        catch { }
    }
}
```

```
    }  
    catch (Exception ex)  
    {  
        rpta = ex.Message;  
    }  
    finally  
    {  
        if (SqlCon.State == ConnectionState.Open) SqlCon.Close();  
    }  
    return rpta;  
}  
  
//Método Editar  
public string Editar(Presentacion Presentacion)  
{  
    string rpta = "";  
    SqlConnection SqlCon = new SqlConnection();  
    try  
    {  
        //Código  
        SqlCon.ConnectionString = Conexion.Cn;  
        SqlCon.Open();  
        //Establecer el Comando  
        SqlCommand SqlCmd = new SqlCommand();  
        SqlCmd.Connection = SqlCon;  
        SqlCmd.CommandText = "speditar_presentacion";  
        SqlCmd.CommandType = CommandType.StoredProcedure;  
  
        SqlParameter Idpresentacion = new SqlParameter();  
        Idpresentacion.ParameterName = "@idpresentacion";  
        Idpresentacion.SqlDbType = SqlDbType.Int;  
        Idpresentacion.Value = Presentacion.IdPresentacion;  
        SqlCmd.Parameters.Add(Idpresentacion);  
  
        SqlParameter Nombre = new SqlParameter();  
        Nombre.ParameterName = "@nombre";  
        Nombre.SqlDbType = SqlDbType.VarChar;  
        Nombre.Size = 50;  
        Nombre.Value = Presentacion.Nombre;  
        SqlCmd.Parameters.Add(Nombre);  
  
        SqlParameter Descripcion = new SqlParameter();  
        Descripcion.ParameterName = "@descripcion";  
        Descripcion.SqlDbType = SqlDbType.VarChar;  
        Descripcion.Size = 256;  
        Descripcion.Value = Presentacion.Descripcion;  
        SqlCmd.Parameters.Add(Descripcion);  
  
        //Ejecutamos nuestro comando  
  
        rpta = SqlCmd.ExecuteNonQuery() == 1 ? "OK" : "Registro NO  
actualizado";  
    }  
    catch { }  
    return rpta;  
}
```

```
    }  
    catch (Exception ex)  
    {  
        rpta = ex.Message;  
    }  
    finally  
    {  
        if (SqlCon.State == ConnectionState.Open) SqlCon.Close();  
    }  
    return rpta;  
}  
  
//Método Eliminar  
public string Eliminar(Presentacion Presentacion)  
{  
    string rpta = "";  
    SqlConnection SqlCon = new SqlConnection();  
    try  
    {  
        //Código  
        SqlCon.ConnectionString = Conexion.Cn;  
        SqlCon.Open();  
        //Establecer el Comando  
        SqlCommand SqlCmd = new SqlCommand();  
        SqlCmd.Connection = SqlCon;  
        SqlCmd.CommandText = "speliminar_presentacion";  
        SqlCmd.CommandType = CommandType.StoredProcedure;  
  
        SqlParameter Idpresentacion = new SqlParameter();  
        Idpresentacion.ParameterName = "@idpresentacion";  
        Idpresentacion.SqlDbType = SqlDbType.Int;  
        Idpresentacion.Value = Presentacion.IdPresentacion;  
        SqlCmd.Parameters.Add(Idpresentacion);  
  
        //Ejecutamos nuestro comando  
  
        rpta = SqlCmd.ExecuteNonQuery() == 1 ? "OK" : "Registro NO  
eliminado";  
    }  
    catch (Exception ex)  
    {  
        rpta = ex.Message;  
    }  
    finally  
    {  
        if (SqlCon.State == ConnectionState.Open) SqlCon.Close();  
    }  
    return rpta;  
}  
  
//Método Mostrar  
public DataTable Mostrar()  
{  
    DataTable DtResultado = new DataTable("presentacion");
```

```
SqlConnection SqlCon = new SqlConnection();
try
{
    SqlCon.ConnectionString = Conexion.Cn;
    SqlCommand SqlCmd = new SqlCommand();
    SqlCmd.Connection = SqlCon;
    SqlCmd.CommandText = "spmostrar_presentacion";
    SqlCmd.CommandType = CommandType.StoredProcedure;

    SqlDataAdapter SqlDat = new SqlDataAdapter(SqlCmd);
    SqlDat.Fill(DtResultado);

}
catch (Exception ex)
{
    DtResultado = null;
}
return DtResultado;
}

//Método BuscarNombre
public DataTable BuscarNombre(Presentacion Presentacion)
{
    DataTable DtResultado = new DataTable("categoria");
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCommand SqlCmd = new SqlCommand();
        SqlCmd.Connection = SqlCon;
        SqlCmd.CommandText = "spbuscar_presentacion";
        SqlCmd.CommandType = CommandType.StoredProcedure;

        SqlParameter TextoBuscar = new SqlParameter();
        TextoBuscar.ParameterName = "@textobuscar";
        TextoBuscar.SqlDbType = SqlDbType.VarChar;
        TextoBuscar.Size = 50;
        TextoBuscar.Value = Presentacion.TextoBuscar;
        SqlCmd.Parameters.Add(TextoBuscar);

        SqlDataAdapter SqlDat = new SqlDataAdapter(SqlCmd);
        SqlDat.Fill(DtResultado);

    }
    catch (Exception ex)
    {
        DtResultado = null;
    }
    return DtResultado;
}
}
```

2.1.9. DProveedores

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data;
using System.Data.SqlClient;

namespace CapaDatos
{
    public class Proveedor
    {
        private int _Idproveedor;
        private string _Razon_Social;
        private string _Apellidos;
        private string _Sector_Comercial;
        private string _Tipo_Documento;
        private string _Num_Documento;
        private string _Direccion;
        private string _Telefono;
        private string _Email;
        private string _Url;
        private string _TextoBuscar;

        public int Idproveedor
        {
            get
            {
                return _Idproveedor;
            }
            set
            {
                _Idproveedor = value;
            }
        }

        public string Razon_Social
        {
            get
            {
                return _Razon_Social;
            }
            set
            {
                _Razon_Social = value;
            }
        }

        public string Apellidos
        {
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
        get
        {
            return _Apellidos;
        }

        set
        {
            _Apellidos = value;
        }
    }

    public string Sector_Comercial
    {
        get
        {
            return _Sector_Comercial;
        }

        set
        {
            _Sector_Comercial = value;
        }
    }

    public string Tipo_Documento
    {
        get
        {
            return _Tipo_Documento;
        }

        set
        {
            _Tipo_Documento = value;
        }
    }

    public string Num_Documento
    {
        get
        {
            return _Num_Documento;
        }

        set
        {
            _Num_Documento = value;
        }
    }

    public string Direccion
    {
        get
        {
            return _Direccion;
        }
    }
}
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
        set
        {
            _Direccion = value;
        }
    }

    public string Telefono
    {
        get
        {
            return _Telefono;
        }

        set
        {
            _Telefono = value;
        }
    }

    public string Email
    {
        get
        {
            return _Email;
        }

        set
        {
            _Email = value;
        }
    }

    public string Url
    {
        get
        {
            return _Url;
        }

        set
        {
            _Url = value;
        }
    }

    public string TextoBuscar
    {
        get
        {
            return _TextoBuscar;
        }

        set
        {
            _TextoBuscar = value;
        }
    }
}
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
public Proveedor()
{
}

public Proveedor(int idproveedor, string razon_social, string
sector_comercial, string tipo_documento, string num_documento, string
direccion, string telefono, string email, string url, string textobuscar)
{
    this.Idproveedor = idproveedor;
    this.Razon_Social = razon_social;
    this.Sector_Comercial = sector_comercial;
    this.Tipo_Documento = tipo_documento;
    this.Num_Documento = num_documento;
    this.Direccion = direccion;
    this.Telefono = telefono;
    this.Email = email;
    this.Url = url;
    this.TextoBuscar = textobuscar;
}

//Método Insertar

public string Insertar(Proveedor Proveedor)
{
    string rpt = "";
    SqlConnection sqlCon = new SqlConnection();
    try
    {
        //Código
        sqlCon.ConnectionString = Conexion.Cn;
        sqlCon.Open();
        //Establecer el Comando
        SqlCommand sqlCmd = new SqlCommand();
        sqlCmd.Connection = sqlCon;
        sqlCmd.CommandText = "spi_proveedor";
        sqlCmd.CommandType = CommandType.StoredProcedure;

        SqlParameter ParIdproveedor = new SqlParameter();
        ParIdproveedor.ParameterName = "@idproveedor";
        ParIdproveedor.SqlDbType = SqlDbType.Int;
        ParIdproveedor.Direction = ParameterDirection.Output;
        sqlCmd.Parameters.Add(ParIdproveedor);

        SqlParameter ParRazon_Social = new SqlParameter();
        ParRazon_Social.ParameterName = "@razon_social";
        ParRazon_Social.SqlDbType = SqlDbType.VarChar;
        ParRazon_Social.Size = 20;
        ParRazon_Social.Value = Proveedor.Razon_Social;
        sqlCmd.Parameters.Add(ParRazon_Social);

        SqlParameter ParSector_Comercial = new SqlParameter();
        ParSector_Comercial.ParameterName = "@sector_comercial";
        ParSector_Comercial.SqlDbType = SqlDbType.VarChar;
        ParSector_Comercial.Size = 40;
        ParSector_Comercial.Value = Proveedor.Sector_Comercial;
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
SqlCmd.Parameters.Add(ParSector_Comercial);

SqlParameter ParTipo_Documento = new SqlParameter();
ParTipo_Documento.ParameterName = "@tipo_documento";
ParTipo_Documento.SqlDbType = SqlDbType.VarChar;
ParTipo_Documento.Size = 20;
ParTipo_Documento.Value = Proveedor.Tipo_Documento;
SqlCmd.Parameters.Add(ParTipo_Documento);

SqlParameter ParNum_Documento = new SqlParameter();
ParNum_Documento.ParameterName = "@num_documento";
ParNum_Documento.SqlDbType = SqlDbType.VarChar;
ParNum_Documento.Size = 15;
ParNum_Documento.Value = Proveedor.Num_Documento;
SqlCmd.Parameters.Add(ParNum_Documento);

SqlParameter ParDireccion = new SqlParameter();
ParDireccion.ParameterName = "@direccion";
ParDireccion.SqlDbType = SqlDbType.VarChar;
ParDireccion.Size = 100;
ParDireccion.Value = Proveedor.Direccion;
SqlCmd.Parameters.Add(ParDireccion);

SqlParameter ParTelefono = new SqlParameter();
ParTelefono.ParameterName = "@telefono";
ParTelefono.SqlDbType = SqlDbType.VarChar;
ParTelefono.Size = 15;
ParTelefono.Value = Proveedor.Telefono;
SqlCmd.Parameters.Add(ParTelefono);

SqlParameter ParEmail = new SqlParameter();
ParEmail.ParameterName = "@email";
ParEmail.SqlDbType = SqlDbType.VarChar;
ParEmail.Size = 50;
ParEmail.Value = Proveedor.Email;
SqlCmd.Parameters.Add(ParEmail);

SqlParameter ParUrl = new SqlParameter();
ParUrl.ParameterName = "@url";
ParUrl.SqlDbType = SqlDbType.VarChar;
ParUrl.Size = 50;
ParUrl.Value = Proveedor.Url;
SqlCmd.Parameters.Add(ParUrl);

//Ejecutamos nuestro comando
rpta = SqlCommand.ExecuteNonQuery() == 1 ? "OK" : "NO se Ingreso
el Registro";
}
catch (Exception ex)
{
    rpta = ex.Message;
}
finally
{
    if (SqlCon.State == ConnectionState.Open) SqlCon.Close();
}
}
```

```
        return rptA;
    }

    //Método Editar
    public string Editar(Proveedor Proveedor)
    {
        string rptA = "";
        SqlConnection SqlCon = new SqlConnection();
        try
        {
            //Código
            SqlCon.ConnectionString = Conexion.Cn;
            SqlCon.Open();
            //Establecer el Comando
            SqlCommand SqlCmd = new SqlCommand();
            SqlCmd.Connection = SqlCon;
            SqlCmd.CommandText = "spe_proveedor";
            SqlCmd.CommandType = CommandType.StoredProcedure;

            SqlParameter ParIdcliente = new SqlParameter();
            ParIdcliente.ParameterName = "@idproveedor";
            ParIdcliente.SqlDbType = SqlDbType.Int;
            ParIdcliente.Value = Proveedor.Idproveedor;
            SqlCmd.Parameters.Add(ParIdcliente);

            SqlParameter ParRazon_Social = new SqlParameter();
            ParRazon_Social.ParameterName = "@razon_social";
            ParRazon_Social.SqlDbType = SqlDbType.VarChar;
            ParRazon_Social.Size = 20;
            ParRazon_Social.Value = Proveedor.Razon_Social;
            SqlCmd.Parameters.Add(ParRazon_Social);

            SqlParameter ParSector_Comercial = new SqlParameter();
            ParSector_Comercial.ParameterName = "@sector_comercial";
            ParSector_Comercial.SqlDbType = SqlDbType.VarChar;
            ParSector_Comercial.Size = 40;
            ParSector_Comercial.Value = Proveedor.Sector_Comercial;
            SqlCmd.Parameters.Add(ParSector_Comercial);

            SqlParameter ParTipo_Documento = new SqlParameter();
            ParTipo_Documento.ParameterName = "@tipo_documento";
            ParTipo_Documento.SqlDbType = SqlDbType.VarChar;
            ParTipo_Documento.Size = 20;
            ParTipo_Documento.Value = Proveedor.Tipo_Documento;
            SqlCmd.Parameters.Add(ParTipo_Documento);

            SqlParameter ParNum_Documento = new SqlParameter();
            ParNum_Documento.ParameterName = "@num_documento";
            ParNum_Documento.SqlDbType = SqlDbType.VarChar;
            ParNum_Documento.Size = 15;
            ParNum_Documento.Value = Proveedor.Num_Documento;
            SqlCmd.Parameters.Add(ParNum_Documento);

            SqlParameter ParDireccion = new SqlParameter();
            ParDireccion.ParameterName = "@direccion";
            ParDireccion.SqlDbType = SqlDbType.VarChar;
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
        ParDireccion.Size = 100;
        ParDireccion.Value = Proveedor.Direccion;
        SqlCommand.Parameters.Add(ParDireccion);

        SqlParameter ParTelefono = new SqlParameter();
        ParTelefono.ParameterName = "@telefono";
        ParTelefono.SqlDbType = SqlDbType.VarChar;
        ParTelefono.Size = 15;
        ParTelefono.Value = Proveedor.Telefono;
        SqlCommand.Parameters.Add(ParTelefono);

        SqlParameter ParEmail = new SqlParameter();
        ParEmail.ParameterName = "@email";
        ParEmail.SqlDbType = SqlDbType.VarChar;
        ParEmail.Size = 50;
        ParEmail.Value = Proveedor.Email;
        SqlCommand.Parameters.Add(ParEmail);

        SqlParameter ParUrl = new SqlParameter();
        ParUrl.ParameterName = "@url";
        ParUrl.SqlDbType = SqlDbType.VarChar;
        ParUrl.Size = 50;
        ParUrl.Value = Proveedor.Url;
        SqlCommand.Parameters.Add(ParUrl);

        //Ejecutamos nuestro comando

        rpta = SqlCommand.ExecuteNonQuery() == 1 ? "OK" : "NO se
Actualizo el Registro";
    }
    catch (Exception ex)
    {
        rpta = ex.Message;
    }
    finally
    {
        if (SqlCon.State == ConnectionState.Open) SqlCon.Close();
    }
    return rpta;
}

//Método Eliminar
public string Eliminar(Proveedor Proveedor)
{
    string rpta = "";
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        //Código
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCon.Open();
        //Establecer el Comando
        SqlCommand SqlCommand = new SqlCommand();
        SqlCommand.Connection = SqlCon;
        SqlCommand.CommandText = "spe1_proveedor";
        SqlCommand.CommandType = CommandType.StoredProcedure;
    }
}
```

```
        SqlParameter ParIdcliente = new SqlParameter();
        ParIdcliente.ParameterName = "@idproveedor";
        ParIdcliente.SqlDbType = SqlDbType.Int;
        ParIdcliente.Value = Proveedor.Idproveedor;
        SqlCommand.Parameters.Add(ParIdcliente);
        //Ejecutamos nuestro comando

        rpta = SqlCommand.ExecuteNonQuery() == 1 ? "OK" : "NO se Elimino
el Registro";
    }
    catch (Exception ex)
    {
        rpta = ex.Message;
    }
    finally
    {
        if (SqlConnection.State == ConnectionState.Open) SqlConnection.Close();
    }
    return rpta;
}

//Método Mostrar
public DataTable Mostrar()
{
    DataTable DtResultado = new DataTable("proveedor");
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCommand SqlCmd = new SqlCommand();
        SqlCmd.Connection = SqlCon;
        SqlCmd.CommandText = "spm_proveedor";
        SqlCmd.CommandType = CommandType.StoredProcedure;

        SqlDataAdapter SqlDat = new SqlDataAdapter(SqlCmd);
        SqlDat.Fill(DtResultado);

    }
    catch (Exception ex)
    {
        DtResultado = null;
    }
    return DtResultado;
}

//Método BuscarRazon_Social
public DataTable BuscarRazon_Social(Proveedor Proveedor)
{
    DataTable DtResultado = new DataTable("proveedor");
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        SqlCon.ConnectionString = Conexion.Cn;
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
        SqlCommand SqlCmd = new SqlCommand();
        SqlCmd.Connection = SqlCon;
        SqlCmd.CommandText = "spb_proveedor_razon_social";
        SqlCmd.CommandType = CommandType.StoredProcedure;

        SqlParameter ParTextoBuscar = new SqlParameter();
        ParTextoBuscar.ParameterName = "@textobuscar";
        ParTextoBuscar.SqlDbType = SqlDbType.VarChar;
        ParTextoBuscar.Size = 50;
        ParTextoBuscar.Value = Proveedor.TextoBuscar;
        SqlCmd.Parameters.Add(ParTextoBuscar);

        SqlDataAdapter SqlDat = new SqlDataAdapter(SqlCmd);
        SqlDat.Fill(DtResultado);
    }
    catch (Exception ex)
    {
        DtResultado = null;
    }
    return DtResultado;
}

//Método BuscarNum_Documento
public DataTable BuscarNum_Documento(Proveedor Proveedor)
{
    DataTable DtResultado = new DataTable("proveedor");
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCommand SqlCmd = new SqlCommand();
        SqlCmd.Connection = SqlCon;
        SqlCmd.CommandText = "spb_proveedor_num_documento";
        SqlCmd.CommandType = CommandType.StoredProcedure;

        SqlParameter ParTextoBuscar = new SqlParameter();
        ParTextoBuscar.ParameterName = "@textobuscar";
        ParTextoBuscar.SqlDbType = SqlDbType.VarChar;
        ParTextoBuscar.Size = 50;
        ParTextoBuscar.Value = Proveedor.TextoBuscar;
        SqlCmd.Parameters.Add(ParTextoBuscar);

        SqlDataAdapter SqlDat = new SqlDataAdapter(SqlCmd);
        SqlDat.Fill(DtResultado);
    }
    catch (Exception ex)
    {
        DtResultado = null;
    }
    return DtResultado;
}
}
```

2.1.10. DTrabajador

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data;
using System.Data.SqlClient;

namespace CapaDatos
{
    public class Trabajador
    {
        //Variables
        private int _Idtrabajador;
        private string _Nombre;
        private string _Apellidos;
        private string _Sexo;
        private DateTime _Fecha_Nacimiento;
        private string _Num_Documento;
        private string _Direccion;
        private string _Telefono;
        private string _Email;
        private string _Acceso;
        private string _Usuario;
        private string _Password;
        private string _TextoBuscar;

        public int Idtrabajador
        {
            get
            {
                return _Idtrabajador;
            }
            set
            {
                _Idtrabajador = value;
            }
        }

        public string Nombre
        {
            get
            {
                return _Nombre;
            }
        }
    }
}
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
        set
        {
            _Nombre = value;
        }
    }

    public string Apellidos
    {
        get
        {
            return _Apellidos;
        }

        set
        {
            _Apellidos = value;
        }
    }

    public string Sexo
    {
        get
        {
            return _Sexo;
        }

        set
        {
            _Sexo = value;
        }
    }

    public DateTime Fecha_Nacimiento
    {
        get
        {
            return _Fecha_Nacimiento;
        }

        set
        {
            _Fecha_Nacimiento = value;
        }
    }

    public string Num_Documento
    {
        get
        {
            return _Num_Documento;
        }

        set
        {
            _Num_Documento = value;
        }
    }
}
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
}

public string Direccion
{
    get
    {
        return _Direccion;
    }

    set
    {
        _Direccion = value;
    }
}

public string Telefono
{
    get
    {
        return _Telefono;
    }

    set
    {
        _Telefono = value;
    }
}

public string Email
{
    get
    {
        return _Email;
    }

    set
    {
        _Email = value;
    }
}

public string Acceso
{
    get
    {
        return _Acceso;
    }

    set
    {
        _Acceso = value;
    }
}

public string Usuario
{
    get
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
        {
            return _Usuario;
        }

        set
        {
            _Usuario = value;
        }
    }

    public string Password
    {
        get
        {
            return _Password;
        }

        set
        {
            _Password = value;
        }
    }

    public string TextoBuscar
    {
        get
        {
            return _TextoBuscar;
        }

        set
        {
            _TextoBuscar = value;
        }
    }
}

//Método Insertar

public string Insertar(Trabajador Trabajador)
{
    string rpta = "";
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        //Código
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCon.Open();
        //Establecer el Comando
        SqlCommand SqlCmd = new SqlCommand();
        SqlCmd.Connection = SqlCon;
        SqlCmd.CommandText = "spi_trabajador";
        SqlCmd.CommandType = CommandType.StoredProcedure;

        SqlParameter ParIdtrabajador = new SqlParameter();
        ParIdtrabajador.ParameterName = "@idtrabajador";
        ParIdtrabajador.SqlDbType = SqlDbType.Int;
        ParIdtrabajador.Direction = ParameterDirection.Output;
    }
    catch
    {
        rpta = "Error al insertar el trabajador";
    }
    return rpta;
}
```

```
SqlCommand.Parameters.Add(ParIdtrabajador);

SqlParameter ParNombre = new SqlParameter();
ParNombre.ParameterName = "@nombre";
ParNombre.SqlDbType = SqlDbType.VarChar;
ParNombre.Size = 20;
ParNombre.Value = Trabajador.Nombre;
SqlCommand.Parameters.Add(ParNombre);

SqlParameter ParApellidos = new SqlParameter();
ParApellidos.ParameterName = "@apellidos";
ParApellidos.SqlDbType = SqlDbType.VarChar;
ParApellidos.Size = 40;
ParApellidos.Value = Trabajador.Apellidos;
SqlCommand.Parameters.Add(ParApellidos);

SqlParameter ParSexo = new SqlParameter();
ParSexo.ParameterName = "@sexo";
ParSexo.SqlDbType = SqlDbType.VarChar;
ParSexo.Size = 1;
ParSexo.Value = Trabajador.Sexo;
SqlCommand.Parameters.Add(ParSexo);

SqlParameter ParFecha_Nacimiento = new SqlParameter();
ParFecha_Nacimiento.ParameterName = "@fecha_nacimiento";
ParFecha_Nacimiento.SqlDbType = SqlDbType.VarChar;
ParFecha_Nacimiento.Size = 40;
ParFecha_Nacimiento.Value = Trabajador.Fecha_Nacimiento;
SqlCommand.Parameters.Add(ParFecha_Nacimiento);

SqlParameter ParNum_Documento = new SqlParameter();
ParNum_Documento.ParameterName = "@num_documento";
ParNum_Documento.SqlDbType = SqlDbType.VarChar;
ParNum_Documento.Size = 15;
ParNum_Documento.Value = Trabajador.Num_Documento;
SqlCommand.Parameters.Add(ParNum_Documento);

SqlParameter ParDireccion = new SqlParameter();
ParDireccion.ParameterName = "@direccion";
ParDireccion.SqlDbType = SqlDbType.VarChar;
ParDireccion.Size = 100;
ParDireccion.Value = Trabajador.Direccion;
SqlCommand.Parameters.Add(ParDireccion);

SqlParameter ParTelefono = new SqlParameter();
ParTelefono.ParameterName = "@telefono";
ParTelefono.SqlDbType = SqlDbType.VarChar;
ParTelefono.Size = 10;
ParTelefono.Value = Trabajador.Telefono;
SqlCommand.Parameters.Add(ParTelefono);

SqlParameter ParEmail = new SqlParameter();
ParEmail.ParameterName = "@email";
ParEmail.SqlDbType = SqlDbType.VarChar;
ParEmail.Size = 50;
ParEmail.Value = Trabajador.Email;
SqlCommand.Parameters.Add(ParEmail);
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
        SqlParameter ParAcceso = new SqlParameter();
        ParAcceso.ParameterName = "@acceso";
        ParAcceso.SqlDbType = SqlDbType.VarChar;
        ParAcceso.Size = 50;
        ParAcceso.Value = Trabajador.Acceso;
        SqlCommand.Parameters.Add(ParAcceso);

        SqlParameter ParUsuario = new SqlParameter();
        ParUsuario.ParameterName = "@usuario";
        ParUsuario.SqlDbType = SqlDbType.VarChar;
        ParUsuario.Size = 50;
        ParUsuario.Value = Trabajador.Usuario;
        SqlCommand.Parameters.Add(ParUsuario);

        SqlParameter ParPassword = new SqlParameter();
        ParPassword.ParameterName = "@password";
        ParPassword.SqlDbType = SqlDbType.VarChar;
        ParPassword.Size = 50;
        ParPassword.Value = Trabajador.Password;
        SqlCommand.Parameters.Add(ParPassword);

        //Ejecutamos nuestro comando
        rpta = SqlCommand.ExecuteNonQuery() == 1 ? "OK" : "NO se Ingreso
el Registro";
    }
    catch (Exception ex)
    {
        rpta = ex.Message;
    }
    finally
    {
        if (SqlCon.State == ConnectionState.Open) SqlCon.Close();
    }
    return rpta;
}

//Método Editar
public string Editar(Trabajador Trabajador)
{
    string rpta = "";
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        //Código
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCon.Open();
        //Establecer el Comando
        SqlCommand SqlCommand = new SqlCommand();
        SqlCommand.Connection = SqlCon;
        SqlCommand.CommandText = "sped_trabajador";
        SqlCommand.CommandType = CommandType.StoredProcedure;

        SqlParameter ParIdtrabajador = new SqlParameter();
        ParIdtrabajador.ParameterName = "@idtrabajador";
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
ParIdtrabajador.SqlDbType = SqlDbType.Int;
ParIdtrabajador.Value = Trabajador.Idtrabajador;
SqlCommand.Parameters.Add(ParIdtrabajador);

SqlParameter ParNombre = new SqlParameter();
ParNombre.ParameterName = "@nombre";
ParNombre.SqlDbType = SqlDbType.VarChar;
ParNombre.Size = 20;
ParNombre.Value = Trabajador.Nombre;
SqlCommand.Parameters.Add(ParNombre);

SqlParameter ParApellidos = new SqlParameter();
ParApellidos.ParameterName = "@apellidos";
ParApellidos.SqlDbType = SqlDbType.VarChar;
ParApellidos.Size = 40;
ParApellidos.Value = Trabajador.Apellidos;
SqlCommand.Parameters.Add(ParApellidos);

SqlParameter ParSexo = new SqlParameter();
ParSexo.ParameterName = "@sexo";
ParSexo.SqlDbType = SqlDbType.VarChar;
ParSexo.Size = 1;
ParSexo.Value = Trabajador.Sexo;
SqlCommand.Parameters.Add(ParSexo);

SqlParameter ParFecha_Nacimiento = new SqlParameter();
ParFecha_Nacimiento.ParameterName = "@fecha_nacimiento";
ParFecha_Nacimiento.SqlDbType = SqlDbType.VarChar;
ParFecha_Nacimiento.Size = 40;
ParFecha_Nacimiento.Value = Trabajador.Fecha_Nacimiento;
SqlCommand.Parameters.Add(ParFecha_Nacimiento);

SqlParameter ParNum_Documento = new SqlParameter();
ParNum_Documento.ParameterName = "@num_documento";
ParNum_Documento.SqlDbType = SqlDbType.VarChar;
ParNum_Documento.Size = 15;
ParNum_Documento.Value = Trabajador.Num_Documento;
SqlCommand.Parameters.Add(ParNum_Documento);

SqlParameter ParDireccion = new SqlParameter();
ParDireccion.ParameterName = "@direccion";
ParDireccion.SqlDbType = SqlDbType.VarChar;
ParDireccion.Size = 100;
ParDireccion.Value = Trabajador.Direccion;
SqlCommand.Parameters.Add(ParDireccion);

SqlParameter ParTelefono = new SqlParameter();
ParTelefono.ParameterName = "@telefono";
ParTelefono.SqlDbType = SqlDbType.VarChar;
ParTelefono.Size = 10;
ParTelefono.Value = Trabajador.Telefono;
SqlCommand.Parameters.Add(ParTelefono);

SqlParameter ParEmail = new SqlParameter();
ParEmail.ParameterName = "@email";
ParEmail.SqlDbType = SqlDbType.VarChar;
ParEmail.Size = 50;
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
ParEmail.Value = Trabajador.Email;
SqlCommand.Parameters.Add(ParEmail);

SqlParameter ParAcceso = new SqlParameter();
ParAcceso.ParameterName = "@acceso";
ParAcceso.SqlDbType = SqlDbType.VarChar;
ParAcceso.Size = 50;
ParAcceso.Value = Trabajador.Acceso;
SqlCommand.Parameters.Add(ParAcceso);

SqlParameter ParUsuario = new SqlParameter();
ParUsuario.ParameterName = "@usuario";
ParUsuario.SqlDbType = SqlDbType.VarChar;
ParUsuario.Size = 50;
ParUsuario.Value = Trabajador.Usuario;
SqlCommand.Parameters.Add(ParUsuario);

SqlParameter ParPassword = new SqlParameter();
ParPassword.ParameterName = "@password";
ParPassword.SqlDbType = SqlDbType.VarChar;
ParPassword.Size = 50;
ParPassword.Value = Trabajador.Password;
SqlCommand.Parameters.Add(ParPassword);
//Ejecutamos nuestro comando

rpta = SqlCommand.ExecuteNonQuery() == 1 ? "OK" : "NO se
Actualizo el Registro";
}
catch (Exception ex)
{
    rpta = ex.Message;
}
finally
{
    if (SqlCon.State == ConnectionState.Open) SqlCon.Close();
}
return rpta;
}

//Método Eliminar
public string Eliminar(Trabajador Trabajador)
{
    string rpta = "";
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        //Código
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCon.Open();
        //Establecer el Comando
        SqlCommand SqlCmd = new SqlCommand();
        SqlCmd.Connection = SqlCon;
        SqlCmd.CommandText = "spe1_trabajador";
        SqlCmd.CommandType = CommandType.StoredProcedure;

        SqlParameter ParIdtrabajador = new SqlParameter();
        ParIdtrabajador.ParameterName = "@idtrabajador";
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
        ParIdtrabajador.SqlDbType = SqlDbType.Int;
        ParIdtrabajador.Value = Trabajador.Idtrabajador;
        SqlCommand.Parameters.Add(ParIdtrabajador);
        //Ejecutamos nuestro comando

        rpta = SqlCommand.ExecuteNonQuery() == 1 ? "OK" : "NO se Elimino
el Registro";
    }
    catch (Exception ex)
    {
        rpta = ex.Message;
    }
    finally
    {
        if (SqlConnection.State == ConnectionState.Open) SqlConnection.Close();
    }
    return rpta;
}

//Método Mostrar
public DataTable Mostrar()
{
    DataTable DtResultado = new DataTable("trabajador");
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCommand SqlCmd = new SqlCommand();
        SqlCmd.Connection = SqlCon;
        SqlCmd.CommandText = "spm_trabajador";
        SqlCmd.CommandType = CommandType.StoredProcedure;

        SqlDataAdapter SqlDat = new SqlDataAdapter(SqlCmd);
        SqlDat.Fill(DtResultado);
    }
    catch (Exception ex)
    {
        DtResultado = null;
    }
    return DtResultado;
}

//Método BuscarApellidos
public DataTable BuscarApellidos(Trabajador Trabajador)
{
    DataTable DtResultado = new DataTable("trabajador");
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCommand SqlCmd = new SqlCommand();
        SqlCmd.Connection = SqlCon;
        SqlCmd.CommandText = "spb_trabajador_apellidos";
    }
}
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
        SqlCommand.CommandType = CommandType.StoredProcedure;

        SqlParameter ParTextoBuscar = new SqlParameter();
        ParTextoBuscar.ParameterName = "@textobuscar";
        ParTextoBuscar.SqlDbType = SqlDbType.VarChar;
        ParTextoBuscar.Size = 50;
        ParTextoBuscar.Value = Trabajador.TextoBuscar;
        SqlCommand.Parameters.Add(ParTextoBuscar);

        SqlDataAdapter SqlDat = new SqlDataAdapter(SqlCmd);
        SqlDat.Fill(DtResultado);
    }
    catch (Exception ex)
    {
        DtResultado = null;
    }
    return DtResultado;
}

//Método BuscarNum_Documento
public DataTable BuscarNum_Documento(Trabajador Trabajador)
{
    DataTable DtResultado = new DataTable("trabajador");
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCommand SqlCmd = new SqlCommand();
        SqlCmd.Connection = SqlCon;
        SqlCmd.CommandText = "spb_trabajador_num_documento";
        SqlCommand.CommandType = CommandType.StoredProcedure;

        SqlParameter ParTextoBuscar = new SqlParameter();
        ParTextoBuscar.ParameterName = "@textobuscar";
        ParTextoBuscar.SqlDbType = SqlDbType.VarChar;
        ParTextoBuscar.Size = 50;
        ParTextoBuscar.Value = Trabajador.TextoBuscar;
        SqlCommand.Parameters.Add(ParTextoBuscar);

        SqlDataAdapter SqlDat = new SqlDataAdapter(SqlCmd);
        SqlDat.Fill(DtResultado);
    }
    catch (Exception ex)
    {
        DtResultado = null;
    }
    return DtResultado;
}

//Método Login
public DataTable Login(Trabajador Trabajador)
{
    DataTable DtResultado = new DataTable("trabajador");
```

```
SqlConnection SqlCon = new SqlConnection();
try
{
    SqlCon.ConnectionString = Conexion.Cn;
    SqlCommand SqlCmd = new SqlCommand();
    SqlCmd.Connection = SqlCon;
    SqlCmd.CommandText = "splogin";
    SqlCmd.CommandType = CommandType.StoredProcedure;

    SqlParameter ParUsuario = new SqlParameter();
    ParUsuario.ParameterName = "@usuario";
    ParUsuario.SqlDbType = SqlDbType.VarChar;
    ParUsuario.Size = 20;
    ParUsuario.Value = Trabajador.Usuario;
    SqlCmd.Parameters.Add(ParUsuario);

    SqlParameter ParPassword = new SqlParameter();
    ParPassword.ParameterName = "@password";
    ParPassword.SqlDbType = SqlDbType.VarChar;
    ParPassword.Size = 20;
    ParPassword.Value = Trabajador.Password;
    SqlCmd.Parameters.Add(ParPassword);

    SqlDataAdapter SqlDat = new SqlDataAdapter(SqlCmd);
    SqlDat.Fill(DtResultado);

}
catch (Exception ex)
{
    DtResultado = null;
}
return DtResultado;
}
}
```

2.1.11. DVenta

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data;
using System.Data.SqlClient;
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
namespace CapaDatos
{
    public class Venta
    {
        private int _Idventa;
        private int _Idcliente;
        private int _Idtrabajador;
        private DateTime _Fecha;
        private string _Tipo_Comprobante;
        private string _Serie;
        private string _Correlativo;
        private decimal _Iva;
        private string _Estado;

        //Propiedades
        public int Idventa
        {
            get { return _Idventa; }
            set { _Idventa = value; }
        }

        public int Idcliente
        {
            get { return _Idcliente; }
            set { _Idcliente = value; }
        }

        public int Idtrabajador
        {
            get { return _Idtrabajador; }
            set { _Idtrabajador = value; }
        }

        public DateTime Fecha
        {
            get { return _Fecha; }
            set { _Fecha = value; }
        }

        public string Tipo_Comprobante
        {
            get { return _Tipo_Comprobante; }
            set { _Tipo_Comprobante = value; }
        }

        public string Serie
        {
            get { return _Serie; }
            set { _Serie = value; }
        }

        public string Correlativo
        {
            get { return _Correlativo; }
            set { _Correlativo = value; }
        }
    }
}
```

```
public decimal Iva
{
    get { return _Iva; }
    set { _Iva = value; }
}

public string Estado
{
    get { return _Estado; }
    set { _Estado = value; }
}

//Constructores
public Venta()
{
}

public Venta(int idventa, int idcliente, int idtrabajador, DateTime
fecha, string tipo_comprobante, string serie, string correlativo, decimal
iva)
{
    this.Idventa = idventa;
    this.Idcliente = idcliente;
    this.Idtrabajador = idtrabajador;
    this.Fecha = fecha;
    this.Tipo_Comprobante = tipo_comprobante;
    this.Serie = serie;
    this.Correlativo = correlativo;
    this.Iva = iva;
}

//Métodos
public string Insertar(Venta Venta, List<Detalle_Venta> Detalles)
{
    string rptA = "";
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        //Código
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCon.Open();
        //Establecer la transacción
        SqlTransaction SqlTra = SqlCon.BeginTransaction();
        //Establecer el Comando
        SqlCommand SqlCmd = new SqlCommand();
        SqlCmd.Connection = SqlCon;
        SqlCmd.Transaction = SqlTra;
        SqlCmd.CommandText = "spinsertar_venta";
        SqlCmd.CommandType = CommandType.StoredProcedure;

        //Parámetros
        SqlParameter ParIdventa = new SqlParameter();
        ParIdventa.ParameterName = "@idventa";
        ParIdventa.SqlDbType = SqlDbType.Int;
        ParIdventa.Direction = ParameterDirection.Output;
        SqlCmd.Parameters.Add(ParIdventa);
    }
    catch { }
}
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
SqlParameter ParIdcliente = new SqlParameter();
ParIdcliente.ParameterName = "@idcliente";
ParIdcliente.SqlDbType = SqlDbType.Int;
ParIdcliente.Value = Venta.Idcliente;
SqlCommand.Parameters.Add(ParIdcliente);

SqlParameter ParIdTrabajador = new SqlParameter();
ParIdTrabajador.ParameterName = "@idtrabajador";
ParIdTrabajador.SqlDbType = SqlDbType.Int;
ParIdTrabajador.Value = Venta.Idtrabajador;
SqlCommand.Parameters.Add(ParIdTrabajador);

SqlParameter ParFecha = new SqlParameter();
ParFecha.ParameterName = "@fecha";
ParFecha.SqlDbType = SqlDbType.Date;
ParFecha.Value = Venta.Fecha;
SqlCommand.Parameters.Add(ParFecha);

SqlParameter ParTipo_Comprobante = new SqlParameter();
ParTipo_Comprobante.ParameterName = "@tipo_comprobante";
ParTipo_Comprobante.SqlDbType = SqlDbType.VarChar;
ParTipo_Comprobante.Size = 20;
ParTipo_Comprobante.Value = Venta.Tipo_Comprobante;
SqlCommand.Parameters.Add(ParTipo_Comprobante);

SqlParameter ParSerie = new SqlParameter();
ParSerie.ParameterName = "@serie";
ParSerie.SqlDbType = SqlDbType.VarChar;
ParSerie.Size = 4;
ParSerie.Value = Venta.Serie;
SqlCommand.Parameters.Add(ParSerie);

SqlParameter ParCorrelativo = new SqlParameter();
ParCorrelativo.ParameterName = "@correlativo";
ParCorrelativo.SqlDbType = SqlDbType.VarChar;
ParCorrelativo.Size = 7;
ParCorrelativo.Value = Venta.Correlativo;
SqlCommand.Parameters.Add(ParCorrelativo);

SqlParameter ParIva = new SqlParameter();
ParIva.ParameterName = "@iva";
ParIva.SqlDbType = SqlDbType.Decimal;
ParIva.Precision = 4;
ParIva.Scale = 2;
ParIva.Value = Venta.Iva;
SqlCommand.Parameters.Add(ParIva);

//SqlParameter ParEstado = new SqlParameter();
//ParEstado.ParameterName = "@estado";
//ParEstado.SqlDbType = SqlDbType.VarChar;
//ParEstado.Size = 7;
//ParEstado.Value = Venta.Estado;
//SqlCommand.Parameters.Add(ParEstado);

//Ejecutamos nuestro comando
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```

    rpta = SqlCommand.ExecuteNonQuery() == 1 ? "OK" : "NO se Ingreso
el Registro";
    if (rpta.Equals("OK"))
    {
        //Obtenemos el codigo del ingreso que se genero por la
base de datos
        this.Idventa =
Convert.ToInt32(SqlCmd.Parameters["@idventa"].Value);
        foreach (Detalle_Venta det in Detalles)
        {
            //Establecemos el codigo del ingreso que se
autogenero
            det.Idventa = this.Idventa;
            //Llamamos al metodo insertar de la clase
DetalleIngreso
            //y le pasamos la conexion y la transaccion que debe
de usar
            rpta = det.Insertar(det, ref SqlCon, ref SqlTra);
            if (!rpta.Equals("OK"))
            {
                //Si ocurre un error al insertar un detalle de
ingreso salimos del for
                break;
            }
            else
            {
                //Actualizamos el Stock
                rpta = DisminuirStock(det.Iddetalle_Ingreso,
det.Cantidad);
                if (!rpta.Equals("OK"))
                {
                    break;
                }
            }
        }
    }
    if (rpta.Equals("OK"))
    {
        //Se inserto todo los detalles y confirmamos la
transaccion
        SqlTra.Commit();
    }
    else
    {
        //Algun detalle no se inserto y negamos la transaccion
        SqlTra.Rollback();
    }
}
catch (Exception ex)
{
    rpta = ex.Message;
}
finally
{

```

```
        if (SqlCon.State == ConnectionState.Open) SqlCon.Close();
    }
    return rpt;
}

public string Eliminar(Venta Venta)
{
    string rpt = "";
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        //Código
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCon.Open();
        //Establecer el Comando
        SqlCommand SqlCmd = new SqlCommand();
        SqlCmd.Connection = SqlCon;
        SqlCmd.CommandText = "speliminar_venta";
        SqlCmd.CommandType = CommandType.StoredProcedure;

        SqlParameter ParIdventa = new SqlParameter();
        ParIdventa.ParameterName = "@idventa";
        ParIdventa.SqlDbType = SqlDbType.Int;
        ParIdventa.Value = Venta.Idventa;
        SqlCmd.Parameters.Add(ParIdventa);
        //Ejecutamos nuestro comando

        rpt = SqlCmd.ExecuteNonQuery() == 1 ? "OK" : "OK";
    }
    catch (Exception ex)
    {
        rpt = ex.Message;
    }
    finally
    {
        if (SqlCon.State == ConnectionState.Open) SqlCon.Close();
    }
    return rpt;
}
//Disminuir Stock
public string DisminuirStock(int iddetalle_ingreso, int cantidad)
{
    string rpt = "";
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        //Código
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCon.Open();
        //Establecer el Comando
        SqlCommand SqlCmd = new SqlCommand();
        SqlCmd.Connection = SqlCon;
        SqlCmd.CommandText = "spdisminuir_stock";
        SqlCmd.CommandType = CommandType.StoredProcedure;
    }
}
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
        SqlParameter ParIddetalle_ingreso = new SqlParameter();
        ParIddetalle_ingreso.ParameterName = "@iddetalle_ingreso";
        ParIddetalle_ingreso.SqlDbType = SqlDbType.Int;
        ParIddetalle_ingreso.Value = iddetalle_ingreso;
        SqlCommand.Parameters.Add(ParIddetalle_ingreso);

        SqlParameter ParCantidad = new SqlParameter();
        ParCantidad.ParameterName = "@cantidad";
        ParCantidad.SqlDbType = SqlDbType.Int;
        ParCantidad.Value = cantidad;
        SqlCommand.Parameters.Add(ParCantidad);
        //Ejecutamos nuestro comando

        rpta = SqlCommand.ExecuteNonQuery() == 1 ? "OK" : "Se actualizó
el Stock";
    }
    catch (Exception ex)
    {
        rpta = ex.Message;
    }
    finally
    {
        if (SqlCon.State == ConnectionState.Open) SqlCon.Close();
    }
    return rpta;
}

//Método Mostrar
public DataTable Mostrar()
{
    DataTable DtResultado = new DataTable("venta");
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCommand SqlCmd = new SqlCommand();
        SqlCmd.Connection = SqlCon;
        SqlCmd.CommandText = "spmostrar_venta";
        SqlCmd.CommandType = CommandType.StoredProcedure;

        SqlDataAdapter SqlDat = new SqlDataAdapter(SqlCmd);
        SqlDat.Fill(DtResultado);
    }
    catch (Exception ex)
    {
        DtResultado = null;
    }
    return DtResultado;
}

//Método BuscarFechas
```

```
public DataTable BuscarFechas(String TextoBuscar, String
TextoBuscar2)
{
    DataTable DtResultado = new DataTable("venta");
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCommand SqlCmd = new SqlCommand();
        SqlCmd.Connection = SqlCon;
        SqlCmd.CommandText = "spbuscar_venta_fecha";
        SqlCmd.CommandType = CommandType.StoredProcedure;

        SqlParameter ParTextoBuscar = new SqlParameter();
        ParTextoBuscar.ParameterName = "@textobuscar";
        ParTextoBuscar.SqlDbType = SqlDbType.VarChar;
        ParTextoBuscar.Size = 50;
        ParTextoBuscar.Value = TextoBuscar;
        SqlCmd.Parameters.Add(ParTextoBuscar);

        SqlParameter ParTextoBuscar2 = new SqlParameter();
        ParTextoBuscar2.ParameterName = "@textobuscar2";
        ParTextoBuscar2.SqlDbType = SqlDbType.VarChar;
        ParTextoBuscar2.Size = 50;
        ParTextoBuscar2.Value = TextoBuscar2;
        SqlCmd.Parameters.Add(ParTextoBuscar2);

        SqlDataAdapter SqlDat = new SqlDataAdapter(SqlCmd);
        SqlDat.Fill(DtResultado);
    }
    catch (Exception ex)
    {
        DtResultado = null;
    }
    return DtResultado;
}

//Método BuscarFechas
public DataTable MostrarDetalle(String TextoBuscar)
{
    DataTable DtResultado = new DataTable("detalle_venta");
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCommand SqlCmd = new SqlCommand();
        SqlCmd.Connection = SqlCon;
        SqlCmd.CommandText = "spmostrar_detalle_venta";
        SqlCmd.CommandType = CommandType.StoredProcedure;

        SqlParameter ParTextoBuscar = new SqlParameter();
        ParTextoBuscar.ParameterName = "@textobuscar";
        ParTextoBuscar.SqlDbType = SqlDbType.VarChar;
        ParTextoBuscar.Size = 50;
        ParTextoBuscar.Value = TextoBuscar;
    }
}
```

```
        SqlCommand.Parameters.Add(ParTextoBuscar);

        SqlDataAdapter SqlDat = new SqlDataAdapter(SqlCmd);
        SqlDat.Fill(DtResultado);

    }
    catch (Exception ex)
    {
        DtResultado = null;
    }
    return DtResultado;
}

public DataTable MostrarArticulo_Venta_Nombre(String TextoBuscar)
{
    DataTable DtResultado = new DataTable("articulo");
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCommand SqlCmd = new SqlCommand();
        SqlCmd.Connection = SqlCon;
        SqlCmd.CommandText = "spbuscararticulo_venta_nombre";
        SqlCmd.CommandType = CommandType.StoredProcedure;

        SqlParameter ParTextoBuscar = new SqlParameter();
        ParTextoBuscar.ParameterName = "@textobuscar";
        ParTextoBuscar.SqlDbType = SqlDbType.VarChar;
        ParTextoBuscar.Size = 50;
        ParTextoBuscar.Value = TextoBuscar;
        SqlCommand.Parameters.Add(ParTextoBuscar);

        SqlDataAdapter SqlDat = new SqlDataAdapter(SqlCmd);
        SqlDat.Fill(DtResultado);

    }
    catch (Exception ex)
    {
        DtResultado = null;
    }
    return DtResultado;
}

public DataTable MostrarArticulo_Venta_Codigo(String TextoBuscar)
{
    DataTable DtResultado = new DataTable("articulo");
    SqlConnection SqlCon = new SqlConnection();
    try
    {
        SqlCon.ConnectionString = Conexion.Cn;
        SqlCommand SqlCmd = new SqlCommand();
        SqlCmd.Connection = SqlCon;
        SqlCmd.CommandText = "spbuscararticulo_venta_codigo";
        SqlCmd.CommandType = CommandType.StoredProcedure;

        SqlParameter ParTextoBuscar = new SqlParameter();
        ParTextoBuscar.ParameterName = "@textobuscar";
```

```
        ParTextoBuscar.SqlDbType = SqlDbType.VarChar;
        ParTextoBuscar.Size = 50;
        ParTextoBuscar.Value = TextoBuscar;
        SqlCommand.Parameters.Add(ParTextoBuscar);

        SqlDataAdapter SqlDat = new SqlDataAdapter(SqlCmd);
        SqlDat.Fill(DtResultado);
    }
    catch (Exception ex)
    {
        DtResultado = null;
    }
    return DtResultado;
}
}
```

2.2. Capa de Negocios

2.2.1. NArticulo.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using CapaDatos;
using System.Data;

namespace CapaNegocio
{
    public class NArticulo
    {
        //Método Insertar
        //CapaDatos
        public static string Insertar(string codigo, string nombre, string
descripcion, byte[] imagen, int idcategoria)
        {
            Articulo Obj = new Articulo();
            Obj.Codigo = codigo;
            Obj.Nombre = nombre;
            Obj.Descripcion = descripcion;
            Obj.Imagen = imagen;
            Obj.IdCategoria = idcategoria;
            return Obj.Insertar(Obj);
        }

        //Método Editar
        //CapaDatos
        public static string Editar(int idarticulo, string codigo, string
nombre, string descripcion, byte[] imagen, int idcategoria)
        {
```

```
        Artículo Obj = new Artículo();
        Obj.Idarticulo = idarticulo;
        Obj.Codigo = codigo;
        Obj.Nombre = nombre;
        Obj.Descripcion = descripcion;
        Obj.Imagen = imagen;
        Obj.IdCategoria = idcategoria;
        return Obj.Editar(Obj);
    }

    //Método Eliminar
    //CapaDatos
    public static string Eliminar(int idarticulo)
    {
        Artículo Obj = new Artículo();
        Obj.Idarticulo= idarticulo;
        return Obj.Eliminar(Obj);
    }

    //Método Mostrar
    //CapaDatos
    public static DataTable Mostrar()
    {
        return new Artículo().Mostrar();
    }

    //Método BuscarNombre
    //CapaDatos

    public static DataTable BuscarNombre(string textobuscar)
    {
        Artículo Obj = new Artículo();
        Obj.TextoBuscar = textobuscar;
        return Obj.BuscarNombre(Obj);
    }
}
}
```

2.2.2. *NCategoria.*

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

using CapaDatos;
using System.Data;

namespace CapaNegocio
{
    public class NCategoria
    {
        //Método Insertar que llama al método Insertar de la clase
        DCategoria
    }
}
```

```
//de la CapaDatos
public static string Insertar(string nombre, string descripcion)
{
    DCategoria Obj = new DCategoria();
    Obj.Nombre = nombre;
    Obj.Descripcion = descripcion;
    return Obj.Insertar(Obj);
}

//Método Editar que llama al método Editar de la clase DCategoria
//de la CapaDatos
public static string Editar(int idcategoria,string nombre, string
descripcion)
{
    DCategoria Obj = new DCategoria();
    Obj.Idcategoria = idcategoria;
    Obj.Nombre = nombre;
    Obj.Descripcion = descripcion;
    return Obj.Editar(Obj);
}

//Método Eliminar que llama al método Eliminar de la clase
DCategoria
//de la CapaDatos
public static string Eliminar(int idcategoria)
{
    DCategoria Obj = new DCategoria();
    Obj.Idcategoria = idcategoria;
    return Obj.Eliminar(Obj);
}

//Método Mostrar que llama al método Mostrar de la clase DCategoria
//de la CapaDatos
public static DataTable Mostrar()
{
    return new DCategoria().Mostrar();
}

//Método BuscarNombre que llama al método BuscarNombre
//de la clase DCategoria de la CapaDatos

public static DataTable BuscarNombre(string textobuscar)
{
    DCategoria Obj = new DCategoria();
    Obj.TextoBuscar = textobuscar;
    return Obj.BuscarNombre(Obj);
}
}
}
```

2.2.3. NCliente.

```
using System;
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using CapaDatos;
using System.Data;

namespace CapaNegocio
{
    public class NCliente
    {
        //Método Insertar
        //de la CapaDatos
        public static string Insertar(string nombre, string apellido, string
sexo, DateTime fecha_nacimiento, string tipo_documento, string
num_documento, string direccion, string telefono, string email)
        {
            Cliente Obj = new Cliente();
            Obj.Nombre = nombre;
            Obj.Apellidos = apellido;
            Obj.Sexo = sexo;
            Obj.Fecha_Nacimiento = fecha_nacimiento;
            Obj.Tipo_Documento = tipo_documento;
            Obj.Num_Documento = num_documento;
            Obj.Direccion = direccion;
            Obj.Telefono = telefono;
            Obj.Email = email;
            return Obj.Insertar(Obj);
        }

        //Método Editar
        //de la CapaDatos
        public static string Editar(int idcliente, string nombre, string
apellido, string sexo, DateTime fecha_nacimiento, string tipo_documento,
string num_documento, string direccion, string telefono, string email)
        {
            Cliente Obj = new Cliente();
            Obj.Idcliente = idcliente;
            Obj.Nombre = nombre;
            Obj.Apellidos = apellido;
            Obj.Sexo = sexo;
            Obj.Fecha_Nacimiento = fecha_nacimiento;
            Obj.Tipo_Documento = tipo_documento;
            Obj.Num_Documento = num_documento;
            Obj.Direccion = direccion;
            Obj.Telefono = telefono;
            Obj.Email = email;
            return Obj.Editar(Obj);
        }

        //Método Eliminar
        //de la CapaDatos
        public static string Eliminar(int idcliente)
        {
            Cliente Obj = new Cliente();
            Obj.Idcliente = idcliente;
            return Obj.Eliminar(Obj);
        }
    }
}
```

```
    }

    //Método Mostrar
    //de la CapaDatos
    public static DataTable Mostrar()
    {
        return new Cliente().Mostrar();
    }

    //Método BuscarRazon_Social
    //de la clase DCategoria de la CapaDatos

    public static DataTable BuscarApellidos(string textobuscar)
    {
        Cliente Obj = new Cliente();
        Obj.TextoBuscar = textobuscar;
        return Obj.BuscarApellidos(Obj);
    }

    //Método BuscarNum_Documento
    //de la clase DCategoria de la CapaDatos

    public static DataTable BuscarNum_Documento(string textobuscar)
    {
        Cliente Obj = new Cliente();
        Obj.TextoBuscar = textobuscar;
        return Obj.BuscarNum_Documento(Obj);
    }
}
}
```

2.2.4. NIngreso.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data;
using CapaDatos;

namespace CapaNegocio
{
    public class NIngreso
    {
        //Método Insertar que llama al método Insertar de la clase Ingreso
        //de la CapaDatos
        public static string Insertar(int idtrabajador, int idproveedor,
        DateTime fecha, string tipo_comprobante, string serie, string correlativo,
        decimal iva, string estado, DataTable dtDetalles)
        {
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
Ingreso Obj = new Ingreso();
Obj.Idtrabajador = idtrabajador;
Obj.Idproveedor = idproveedor;
Obj.Fecha = fecha;
Obj.Tipo_Comprobante = tipo_comprobante;
Obj.Serie = serie;
Obj.Correlativo = correlativo;
Obj.Iva = iva;
Obj.Estado = estado;
List<Detalle_Ingreso> detalles = new List<Detalle_Ingreso>();
foreach (DataRow row in dtDetalles.Rows)
{
    Detalle_Ingreso detalle = new Detalle_Ingreso();
    detalle.Idarticulo =
Convert.ToInt32(row["idarticulo"].ToString());
    detalle.Precio_Compra =
Convert.ToDecimal(row["precio_compra"].ToString());
    detalle.Precio_Venta =
Convert.ToDecimal(row["precio_venta"].ToString());
    detalle.Stock_Inicial =
Convert.ToInt32(row["stock_inicial"].ToString());
    detalle.Stock_Actual =
Convert.ToInt32(row["stock_inicial"].ToString());
    //detalle.Fecha_Produccion =
Convert.ToDateTime(row["fecha_produccion"].ToString());
    //detalle.Fecha_Vencimiento =
Convert.ToDateTime(row["fecha_vencimiento"].ToString());
    detalles.Add(detalle);
}

return Obj.Insertar(Obj, detalles);
}

//Método Anular que llama al método Anular de la clase DIngreso
//de la CapaDatos
public static string Anular(int idingreso)
{
    Ingreso Obj = new Ingreso();
    Obj.Idingreso = idingreso;
    return Obj.Anular(Obj);
}

//Método Mostrar que llama al método Mostrar de la clase DIngreso
//de la CapaDatos
public static DataTable Mostrar()
{
    return new Ingreso().Mostrar();
}

//Método BuscarApellidos que llama al método BuscarApellidos
//de la clase DCliente de la CapaDatos
public static DataTable BuscarFechas(string textobuscar, string
textobuscar2)
{

```

```
        Ingreso Obj = new Ingreso();
        return Obj.BuscarFechas(textobuscar, textobuscar2);
    }
    public static DataTable MostrarDetalle(string textobuscar)
    {
        Ingreso Obj = new Ingreso();
        return Obj.MostrarDetalle(textobuscar);
    }
}
}
```

2.2.5. NPresentacion.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using CapaDatos;
using System.Data;

namespace CapaNegocio
{
    public class NPresentacion
    {
        //Método Insertar
        //CapaDatos
        public static string Insertar(string nombre, string descripcion)
        {
            Presentacion Obj = new Presentacion();
            Obj.Nombre = nombre;
            Obj.Descripcion = descripcion;
            return Obj.Insertar(Obj);
        }

        //Método Editar
        //CapaDatos
        public static string Editar(int idpresentacion, string nombre,
string descripcion)
        {
            Presentacion Obj = new Presentacion();
            Obj.IdPresentacion = idpresentacion;
            Obj.Nombre = nombre;
            Obj.Descripcion = descripcion;
            return Obj.Editar(Obj);
        }

        //Método Eliminar
        //CapaDatos
        public static string Eliminar(int idpresentacion)
        {
            Presentacion Obj = new Presentacion();
            Obj.IdPresentacion = idpresentacion;
            return Obj.Eliminar(Obj);
        }
    }
}
```

```
    }

    //Método Mostrar
    //CapaDatos
    public static DataTable Mostrar()
    {
        return new Presentacion().Mostrar();
    }

    //Método BuscarNombre
    //CapaDatos

    public static DataTable BuscarNombre(string textobuscar)
    {
        Presentacion Obj = new Presentacion();
        Obj.TextoBuscar = textobuscar;
        return Obj.BuscarNombre(Obj);
    }
}
}
```

2.2.6. NProveedor.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data;
using CapaDatos;

namespace CapaNegocio
{
    public class NProveedor
    {
        //Método Insertar
        //de la CapaDatos
        public static string Insertar(string razon_proveedor, string
sector_comercial, string tipo_documento, string num_documento, string
direccion, string telefono, string email, string url)
        {
            Proveedor Obj = new Proveedor();
            Obj.Razon_Social = razon_proveedor;
            Obj.Sector_Comercial = sector_comercial;
            Obj.Tipo_Documento = tipo_documento;
            Obj.Num_Documento = num_documento;
            Obj.Direccion = direccion;
            Obj.Telefono = telefono;
            Obj.Email = email;
            Obj.Url = url;
            return Obj.Insertar(Obj);
        }
    }
}
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
//Método Editar
//de la CapaDatos
public static string Editar(int idproveedor, string razon_proveedor,
string sector_comercial, string tipo_documento, string num_documento, string
direccion, string telefono, string email, string url)
{
    Proveedor Obj = new Proveedor();
    Obj.Idproveedor = idproveedor;
    Obj.Razon_Social = razon_proveedor;
    Obj.Sector_Comercial = sector_comercial;
    Obj.Tipo_Documento = tipo_documento;
    Obj.Num_Documento = num_documento;
    Obj.Direccion = direccion;
    Obj.Telefono = telefono;
    Obj.Email = email;
    Obj.Url = url;
    return Obj.Editar(Obj);
}

//Método Eliminar
//de la CapaDatos
public static string Eliminar(int idproveedor)
{
    Proveedor Obj = new Proveedor();
    Obj.Idproveedor = idproveedor;
    return Obj.Eliminar(Obj);
}

//Método Mostrar
//de la CapaDatos
public static DataTable Mostrar()
{
    return new Proveedor().Mostrar();
}

//Método BuscarRazon_Social
//de la clase Dcategoría de la CapaDatos

public static DataTable BuscarRazon_Social(string textobuscar)
{
    Proveedor Obj = new Proveedor();
    Obj.TextoBuscar = textobuscar;
    return Obj.BuscarRazon_Social(Obj);
}

//Método BuscarNum_Documento
//de la clase Dcategoría de la CapaDatos

public static DataTable BuscarNum_Documento(string textobuscar)
{
    Proveedor Obj = new Proveedor();
    Obj.TextoBuscar = textobuscar;
    return Obj.BuscarNum_Documento(Obj);
}
}
}
```

2.2.7. NTrabajador.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using CapaDatos;
using System.Data;

namespace CapaNegocio
{
    public class NTrabajador
    {
        //Método Insertar que llama al método Insertar de la clase
        DTrabajador
        //de la CapaDatos
        public static string Insertar(string nombre, string apellidos,
string sexo, DateTime fecha_nacimiento,
string num_documento, string direccion, string telefono, string
email,
string acceso, string usuario, string password)
        {
            Trabajador Obj = new Trabajador();
            Obj.Nombre = nombre;
            Obj.Apellidos = apellidos;
            Obj.Sexo = sexo;
            Obj.Fecha_Nacimiento = fecha_nacimiento;
            Obj.Num_Documento = num_documento;
            Obj.Direccion = direccion;
            Obj.Telefono = telefono;
            Obj.Email = email;
            Obj.Acceso = acceso;
            Obj.Usuario = usuario;
            Obj.Password = password;

            return Obj.Insertar(Obj);
        }

        //Método Editar que llama al método Editar de la clase DTrabajador
        //de la CapaDatos
        public static string Editar(int idtrabajador, string nombre, string
apellidos, string sexo, DateTime fecha_nacimiento,
string num_documento, string direccion, string telefono, string
email,
string acceso, string usuario, string password)
        {
            Trabajador Obj = new Trabajador();
            Obj.Idtrabajador = idtrabajador;
            Obj.Nombre = nombre;
            Obj.Apellidos = apellidos;
            Obj.Sexo = sexo;
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
        Obj.Fecha_Nacimiento = fecha_nacimiento;
        Obj.Num_Documento = num_documento;
        Obj.Direccion = direccion;
        Obj.Telefono = telefono;
        Obj.Email = email;
        Obj.Acceso = acceso;
        Obj.Usuario = usuario;
        Obj.Password = password;

        return Obj.Editar(Obj);
    }

    //Método Eliminar que llama al método Eliminar de la clase
DTrabajador
//de la CapaDatos
public static string Eliminar(int idtrabajador)
{
    Trabajador Obj = new Trabajador();
    Obj.Idtrabajador = idtrabajador;
    return Obj.Eliminar(Obj);
}

//Método Mostrar que llama al método Mostrar de la clase DTrabajador
//de la CapaDatos
public static DataTable Mostrar()
{
    return new Trabajador().Mostrar();
}

//Método BuscarApellidos que llama al método BuscarApellidos
//de la clase DTrabajador de la CapaDatos
public static DataTable BuscarApellidos(string textobuscar)
{
    Trabajador Obj = new Trabajador();
    Obj.TextoBuscar = textobuscar;
    return Obj.BuscarApellidos(Obj);
}

//Método BuscarNum_Documento que llama al método BuscarNum_Documento
//de la clase DTrabajador de la CapaDatos
public static DataTable BuscarNum_Documento(string textobuscar)
{
    Trabajador Obj = new Trabajador();
    Obj.TextoBuscar = textobuscar;
    return Obj.BuscarNum_Documento(Obj);
}

public static DataTable Login(string usuario, string password)
{
    Trabajador Obj = new Trabajador();
    Obj.Usuario = usuario;
    Obj.Password = password;
    return Obj.Login(Obj);
}
}
}
```

2.2.8. NVenta.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data;
using CapaDatos;

namespace CapaNegocio
{
    public class NVenta
    {
        //Método Insertar que llama al método Insertar de la clase DVenta
        //de la CapaDatos
        public static string Insertar(int idcliente, int idtrabajador,
DateTime fecha, string tipo_comprobante, string serie, string correlativo,
decimal iva, string estado, DataTable dtDetalles)
        {
            Venta Obj = new Venta();
            Obj.Idcliente = idcliente;
            Obj.Idtrabajador = idtrabajador;
            Obj.Fecha = fecha;
            Obj.Tipo_Comprobante = tipo_comprobante;
            Obj.Serie = serie;
            Obj.Correlativo = correlativo;
            Obj.Iva = iva;
            Obj.Estado = estado;
            List<Detalle_Venta> detalles = new List<Detalle_Venta>();
            foreach (DataRow row in dtDetalles.Rows)
            {
                Detalle_Venta detalle = new Detalle_Venta();
                detalle.Iddetalle_Ingreso =
Convert.ToInt32(row["iddetalle_ingreso"].ToString());
                detalle.Cantidad =
Convert.ToInt32(row["cantidad"].ToString());
                detalle.Precio_Venta =
Convert.ToDecimal(row["precio_venta"].ToString());
                detalle.Descuento =
Convert.ToDecimal(row["descuento"].ToString());
                detalles.Add(detalle);
            }

            return Obj.Insertar(Obj, detalles);
        }

        //Método Anular que llama al método Anular de la clase DVenta
        //de la CapaDatos
    }
}
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
public static string Eliminar(int idventa)
{
    Venta Obj = new Venta();
    Obj.Idventa = idventa;
    return Obj.Eliminar(Obj);
}

//Método Mostrar que llama al método Mostrar de la clase DVenta
//de la CapaDatos
public static DataTable Mostrar()
{
    return new Venta().Mostrar();
}

//Método BuscarFechas que llama al método BuscarFechas
//de la clase DVenta de la CapaDatos
public static DataTable BuscarFechas(string textobuscar, string
textobuscar2)
{
    Venta Obj = new Venta();
    return Obj.BuscarFechas(textobuscar, textobuscar2);
}
public static DataTable MostrarDetalle(string textobuscar)
{
    Venta Obj = new Venta();
    return Obj.MostrarDetalle(textobuscar);
}
public static DataTable MostrarArticulo_Venta_Nombre(string
textobuscar)
{
    Venta Obj = new Venta();
    return Obj.MostrarArticulo_Venta_Nombre(textobuscar);
}
public static DataTable MostrarArticulo_Venta_Codigo(string
textobuscar)
{
    Venta Obj = new Venta();
    return Obj.MostrarArticulo_Venta_Codigo(textobuscar);
}
}
}
```

2.3. Script SQL de Base de Datos.

```
USE [master]
GO
/***** Object: Database [dbventas]    Script Date: 13/02/2016 12:56:31
*****/
CREATE DATABASE [dbventas]
    CONTAINMENT = NONE
    ON PRIMARY
    ( NAME = N'dbventas', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\dbventas.mdf' , SIZE = 5120KB ,
MAXSIZE = UNLIMITED, FILEGROWTH = 1024KB )
    LOG ON
    ( NAME = N'dbventas_log', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\dbventas_log.ldf' , SIZE = 3136KB ,
MAXSIZE = 2048GB , FILEGROWTH = 10%)
GO
ALTER DATABASE [dbventas] SET COMPATIBILITY_LEVEL = 120
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [dbventas].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO
ALTER DATABASE [dbventas] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [dbventas] SET ANSI_NULLS OFF
GO
ALTER DATABASE [dbventas] SET ANSI_PADDING OFF
GO
ALTER DATABASE [dbventas] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [dbventas] SET ARITHABORT OFF
GO
ALTER DATABASE [dbventas] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [dbventas] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [dbventas] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [dbventas] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [dbventas] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [dbventas] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [dbventas] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [dbventas] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [dbventas] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [dbventas] SET DISABLE_BROKER
GO
ALTER DATABASE [dbventas] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
ALTER DATABASE [dbventas] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [dbventas] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [dbventas] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
ALTER DATABASE [dbventas] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [dbventas] SET READ_COMMITTED_SNAPSHOT OFF
GO
ALTER DATABASE [dbventas] SET HONOR_BROKER_PRIORITY OFF
GO
ALTER DATABASE [dbventas] SET RECOVERY FULL
GO
ALTER DATABASE [dbventas] SET MULTI_USER
GO
ALTER DATABASE [dbventas] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [dbventas] SET DB_CHAINING OFF
GO
ALTER DATABASE [dbventas] SET FILESTREAM( NON_TRANSACTED_ACCESS = OFF )
GO
ALTER DATABASE [dbventas] SET TARGET_RECOVERY_TIME = 0 SECONDS
GO
ALTER DATABASE [dbventas] SET DELAYED_DURABILITY = DISABLED
GO
EXEC sys.sp_db_vardecimal_storage_format N'dbventas', N'ON'
GO
USE [dbventas]
GO
/***** Object: Table [dbo].[articulo]    Script Date: 13/02/2016 12:56:32
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[articulo](
    [idarticulo] [int] IDENTITY(1,1) NOT NULL,
    [codigo] [varchar](50) NOT NULL,
    [nombre] [varchar](50) NOT NULL,
    [descripcion] [varchar](1024) NULL,
    [imagen] [image] NULL,
    [idcategoria] [int] NOT NULL,
    CONSTRAINT [PK_articulo] PRIMARY KEY CLUSTERED
(
    [idarticulo] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[categoria]    Script Date: 13/02/2016 12:56:32
*****/
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[categoria](
    [idcategoria] [int] IDENTITY(1,1) NOT NULL,
    [nombre] [varchar](50) NOT NULL,
    [descripcion] [varchar](256) NULL,
    CONSTRAINT [PK_categoria] PRIMARY KEY CLUSTERED
(
    [idcategoria] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[cliente]    Script Date: 13/02/2016 12:56:32
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[cliente](
    [idcliente] [int] IDENTITY(1,1) NOT NULL,
    [nombre] [varchar](50) NOT NULL,
    [apellidos] [varchar](50) NULL,
    [sexo] [varchar](1) NULL,
    [fecha_nacimiento] [date] NULL,
    [tipo_documento] [varchar](20) NOT NULL,
    [num_documento] [varchar](15) NOT NULL,
    [direccion] [varchar](100) NULL,
    [telefono] [varchar](10) NULL,
    [email] [varchar](50) NULL,
    CONSTRAINT [PK_cliente] PRIMARY KEY CLUSTERED
(
    [idcliente] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[detalle_ingreso]    Script Date: 13/02/2016
12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[detalle_ingreso](
    [iddetalle_ingreso] [int] IDENTITY(1,1) NOT NULL,
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
        [idingreso] [int] NOT NULL,
        [idarticulo] [int] NOT NULL,
        [precio_compra] [money] NOT NULL,
        [precio_venta] [money] NOT NULL,
        [stock_inicial] [int] NOT NULL,
        [stock_actual] [int] NOT NULL,
    CONSTRAINT [PK_detalle_ingreso] PRIMARY KEY CLUSTERED
    (
        [iddetalle_ingreso] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]

GO
/***** Object: Table [dbo].[detalle_venta]    Script Date: 13/02/2016
12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[detalle_venta](
    [iddetalle_venta] [int] IDENTITY(1,1) NOT NULL,
    [idventa] [int] NOT NULL,
    [iddetalle_ingreso] [int] NOT NULL,
    [cantidad] [int] NOT NULL,
    [precio_venta] [money] NOT NULL,
    [descuento] [money] NOT NULL,
    CONSTRAINT [PK_detalle_venta] PRIMARY KEY CLUSTERED
    (
        [iddetalle_venta] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]

GO
/***** Object: Table [dbo].[ingreso]    Script Date: 13/02/2016 12:56:32
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[ingreso](
    [idingreso] [int] IDENTITY(1,1) NOT NULL,
    [idtrabajador] [int] NOT NULL,
    [idproveedor] [int] NOT NULL,
    [fecha] [date] NOT NULL,
    [tipo_comprobante] [varchar](20) NOT NULL,
    [serie] [varchar](4) NOT NULL,
    [correlativo] [varchar](7) NOT NULL,
    [iva] [decimal](4, 2) NOT NULL,
    [estado] [varchar](7) NOT NULL,
    CONSTRAINT [PK_ingreso] PRIMARY KEY CLUSTERED
    (
        [idingreso] ASC
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[proveedor]    Script Date: 13/02/2016 12:56:32
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[proveedor](
    [idproveedor] [int] IDENTITY(1,1) NOT NULL,
    [razon_social] [varchar](150) NOT NULL,
    [sector_comercial] [varchar](50) NOT NULL,
    [tipo_documento] [varchar](20) NOT NULL,
    [num_documento] [varchar](15) NOT NULL,
    [direccion] [varchar](100) NULL,
    [telefono] [varchar](15) NULL,
    [email] [varchar](50) NULL,
    [url] [varchar](100) NULL,
    CONSTRAINT [PK_proveedor] PRIMARY KEY CLUSTERED
(
    [idproveedor] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[trabajador]    Script Date: 13/02/2016
12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[trabajador](
    [idtrabajador] [int] IDENTITY(1,1) NOT NULL,
    [nombre] [varchar](20) NOT NULL,
    [apellidos] [varchar](40) NOT NULL,
    [sexo] [varchar](1) NOT NULL,
    [fecha_nac] [date] NOT NULL,
    [num_documento] [varchar](15) NOT NULL,
    [direccion] [varchar](100) NULL,
    [telefono] [varchar](10) NULL,
    [email] [varchar](50) NULL,
    [acceso] [varchar](20) NOT NULL,
    [usuario] [varchar](20) NOT NULL,
    [password] [varchar](20) NOT NULL,
    CONSTRAINT [PK_trabajador] PRIMARY KEY CLUSTERED
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
(
    [idtrabajador] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[venta]    Script Date: 13/02/2016 12:56:32
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[venta](
    [idventa] [int] IDENTITY(1,1) NOT NULL,
    [idcliente] [int] NOT NULL,
    [idtrabajador] [int] NOT NULL,
    [fecha] [date] NOT NULL,
    [tipo_comprobante] [varchar](20) NOT NULL,
    [serie] [varchar](4) NOT NULL,
    [correlativo] [varchar](7) NOT NULL,
    [iva] [decimal](4, 2) NOT NULL,
    CONSTRAINT [PK_venta] PRIMARY KEY CLUSTERED
(
    [idventa] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
ALTER TABLE [dbo].[articulo] WITH CHECK ADD CONSTRAINT
[FK_articulo_categoria] FOREIGN KEY([idcategoria])
REFERENCES [dbo].[categoria] ([idcategoria])
GO
ALTER TABLE [dbo].[articulo] CHECK CONSTRAINT [FK_articulo_categoria]
GO
ALTER TABLE [dbo].[detalle_ingreso] WITH CHECK ADD CONSTRAINT
[FK_detalle_ingreso_articulo] FOREIGN KEY([idarticulo])
REFERENCES [dbo].[articulo] ([idarticulo])
GO
ALTER TABLE [dbo].[detalle_ingreso] CHECK CONSTRAINT
[FK_detalle_ingreso_articulo]
GO
ALTER TABLE [dbo].[detalle_ingreso] WITH CHECK ADD CONSTRAINT
[FK_detalle_ingreso_ingreso] FOREIGN KEY([idingreso])
REFERENCES [dbo].[ingreso] ([idingreso])
GO
ALTER TABLE [dbo].[detalle_ingreso] CHECK CONSTRAINT
[FK_detalle_ingreso_ingreso]
GO
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
ALTER TABLE [dbo].[detalle_venta] WITH CHECK ADD CONSTRAINT
[FK_detalle_venta_detalle_ingreso] FOREIGN KEY([iddetalle_ingreso])
REFERENCES [dbo].[detalle_ingreso] ([iddetalle_ingreso])
GO
ALTER TABLE [dbo].[detalle_venta] CHECK CONSTRAINT
[FK_detalle_venta_detalle_ingreso]
GO
ALTER TABLE [dbo].[detalle_venta] WITH CHECK ADD CONSTRAINT
[FK_detalle_venta_venta] FOREIGN KEY([idventa])
REFERENCES [dbo].[venta] ([idventa])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[detalle_venta] CHECK CONSTRAINT [FK_detalle_venta_venta]
GO
ALTER TABLE [dbo].[ingreso] WITH CHECK ADD CONSTRAINT
[FK_ingreso_proveedor] FOREIGN KEY([idproveedor])
REFERENCES [dbo].[proveedor] ([idproveedor])
GO
ALTER TABLE [dbo].[ingreso] CHECK CONSTRAINT [FK_ingreso_proveedor]
GO
ALTER TABLE [dbo].[ingreso] WITH CHECK ADD CONSTRAINT
[FK_ingreso_trabajador] FOREIGN KEY([idtrabajador])
REFERENCES [dbo].[trabajador] ([idtrabajador])
GO
ALTER TABLE [dbo].[ingreso] CHECK CONSTRAINT [FK_ingreso_trabajador]
GO
ALTER TABLE [dbo].[venta] WITH CHECK ADD CONSTRAINT [FK_venta_cliente]
FOREIGN KEY([idcliente])
REFERENCES [dbo].[cliente] ([idcliente])
GO
ALTER TABLE [dbo].[venta] CHECK CONSTRAINT [FK_venta_cliente]
GO
ALTER TABLE [dbo].[venta] WITH CHECK ADD CONSTRAINT [FK_venta_trabajador]
FOREIGN KEY([idtrabajador])
REFERENCES [dbo].[trabajador] ([idtrabajador])
GO
ALTER TABLE [dbo].[venta] CHECK CONSTRAINT [FK_venta_trabajador]
GO
/***** Object: StoredProcedure [dbo].[spanular_ingreso]    Script Date:
13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create proc [dbo].[spanular_ingreso]
@idingreso int
as
update ingreso set estado='ANULADO'
where idingreso=@idingreso

GO
/***** Object: StoredProcedure [dbo].[spb_articulo_nombre]    Script Date:
13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
GO
create proc [dbo].[spb_articulo_nombre]
@textobuscar varchar(50)
as

SELECT dbo.articulo.idarticulo, dbo.articulo.codigo, dbo.articulo.nombre,
dbo.articulo.descripcion, dbo.articulo.imagen, dbo.articulo.idcategoria,
dbo.categoria.nombre
AS Categoria
FROM dbo.articulo INNER JOIN
dbo.categoria ON dbo.articulo.idcategoria = dbo.categoria.idcategoria
where dbo.articulo.nombre like @textobuscar + '%'
order by dbo.articulo.idarticulo desc

GO
/***** Object: StoredProcedure [dbo].[spb_cliente_apellidos]      Script
Date: 13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create proc [dbo].[spb_cliente_apellidos]
@textobuscar varchar(50)
as
SELECT * FROM cliente
where apellidos like @textobuscar + '%'

GO
/***** Object: StoredProcedure [dbo].[spb_cliente_num_documento]  Script
Date: 13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create proc [dbo].[spb_cliente_num_documento]
@textobuscar varchar(8)
as
SELECT * FROM cliente
where num_documento like @textobuscar + '%'

GO
/***** Object: StoredProcedure [dbo].[spb_proveedor_num_documento]
Script Date: 13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create proc [dbo].[spb_proveedor_num_documento]
@textobuscar varchar(20)
as
SELECT * FROM proveedor
where num_documento like @textobuscar + '%'

GO
/***** Object: StoredProcedure [dbo].[spb_proveedor_razon_social]
Script Date: 13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
GO
SET QUOTED_IDENTIFIER ON
GO
create proc [dbo].[spb_proveedor_razon_social]
@textobuscar varchar(50)
as
SELECT * FROM proveedor
where razon_social like @textobuscar + '%'

GO
/***** Object: StoredProcedure [dbo].[spb_trabajador_apellidos]      Script
Date: 13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

-- Procedimiento  Buscar Trabajador Apellidos
create proc [dbo].[spb_trabajador_apellidos]
@textobuscar varchar(50)
as
SELECT * FROM trabajador
where apellidos like @textobuscar + '%'

GO
/***** Object: StoredProcedure [dbo].[spb_trabajador_num_documento]
Script Date: 13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

-- Procedimiento  Buscar Trababajador Num Documento
create proc [dbo].[spb_trabajador_num_documento]
@textobuscar varchar(8)
as
SELECT * FROM trabajador
where num_documento like @textobuscar + '%'

GO
/***** Object: StoredProcedure [dbo].[spbbuscar_categoria]      Script Date:
13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE proc [dbo].[spbbuscar_categoria]
@textobuscar varchar(50)
as
select * from categoria
where nombre like @textobuscar + '%' --Alt +39
GO
/***** Object: StoredProcedure [dbo].[spbbuscar_ingreso_fecha]      Script
Date: 13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
GO
create proc [dbo].[spbuscar_ingreso_fecha]
@textobuscar varchar(50),
@textobuscar2 varchar(50)
as
SELECT i.idingreso,
(t.apellidos + ' ' + t.nombre) as Trabajador,
p.razon_social as proveedor,
i.fecha, i.tipo_comprobante,
i.serie, i.correlativo,
i.estado, sum(d.precio_compra*
d.stock_inicial) as Total
FROM detalle_ingreso d INNER JOIN ingreso i
ON d.idingreso = i.idingreso
INNER JOIN proveedor p
ON i.idproveedor = p.idproveedor
INNER JOIN trabajador t
ON i.idtrabajador = t.idtrabajador
group by
i.idingreso,
t.apellidos + ' ' + t.nombre,
p.razon_social,
i.fecha, i.tipo_comprobante,
i.serie, i.correlativo,
i.estado
having i.fecha>=@textobuscar and i.fecha<=@textobuscar2

GO
/***** Object: StoredProcedure [dbo].[spbuscar_presentacion_nombre]
Script Date: 13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create proc [dbo].[spbuscar_presentacion_nombre]
@textobuscar varchar(50)
as
select * from presentacion
where nombre like @textobuscar + '%'

GO
/***** Object: StoredProcedure [dbo].[spbuscar_venta_fecha]      Script
Date: 13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create proc [dbo].[spbuscar_venta_fecha]
@textobuscar varchar(50),
@textobuscar2 varchar(50)
as
SELECT v.idventa,
(t.apellidos + ' ' + t.nombre) as Trabajador,
(c.apellidos + ' ' + c.nombre) as Cliente,
v.fecha, v.tipo_comprobante,
v.serie, v.correlativo, sum((d.precio_venta*
d.cantidad)-d.descuento) as Total
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
FROM detalle_venta d INNER JOIN venta v
ON d.idventa = v.idventa
INNER JOIN cliente c
ON v.idcliente = c.idcliente
INNER JOIN trabajador t
ON v.idtrabajador = t.idtrabajador
group by
v.idventa,
t.apellidos + ' ' + t.nombre,
c.apellidos + ' ' + c.nombre,
v.fecha, v.tipo_comprobante,
v.serie, v.correlativo
having v.fecha >= @textobuscar and v.fecha <= @textobuscar2

GO
/***** Object: StoredProcedure [dbo].[spbuscararticulo_venta_codigo]
Script Date: 13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create proc [dbo].[spbuscararticulo_venta_codigo]
@textobuscar varchar(50)
as
select a.Idarticulo, a.Codigo, a.Nombre, c.nombre as
Categoria, d.stock_actual, d.precio_compra,
d.precio_venta
from articulo a inner join categoria c
on a.idcategoria = c.idcategoria
inner join detalle_ingreso d
on a.idarticulo = d.idarticulo
inner join ingreso i
on i.idingreso = d.idingreso
where a.codigo = @textobuscar
and d.stock_actual > 0
and i.estado <> 'ANULADO'

GO
/***** Object: StoredProcedure [dbo].[spbuscararticulo_venta_nombre]
Script Date: 13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create proc [dbo].[spbuscararticulo_venta_nombre]
@textobuscar varchar(50)
as
select d.iddetalle_ingreso, a.Codigo, a.Nombre, c.nombre
as Categoria, d.stock_actual, d.precio_compra,
d.precio_venta
from articulo a inner join categoria c
on a.idcategoria = c.idcategoria
inner join detalle_ingreso d
on a.idarticulo = d.idarticulo
inner join ingreso i
on i.idingreso = d.idingreso
where a.nombre like @textobuscar + '%'
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
and d.stock_actual>0  
and i.estado<>'ANULADO'
```

```
GO  
/***** Object: StoredProcedure [dbo].[spdisminuir_stock]    Script Date:  
13/02/2016 12:56:32 *****/  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
create proc [dbo].[spdisminuir_stock]  
@iddetalle_ingreso int,  
@cantidad int  
as  
update detalle_ingreso set stock_actual=stock_actual-@cantidad  
where iddetalle_ingreso=@iddetalle_ingreso
```

```
GO  
/***** Object: StoredProcedure [dbo].[spe_articulo]    Script Date:  
13/02/2016 12:56:32 *****/  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
create proc [dbo].[spe_articulo]  
@idarticulo int,  
@codigo varchar(50),  
@nombre varchar(50),  
@descripcion varchar(1024),  
@imagen image,  
@idcategoria int  
as  
update articulo set codigo= @codigo, nombre = @nombre, descripcion =  
@descripcion, imagen = @imagen, idcategoria = @idcategoria  
where idarticulo = @idarticulo
```

```
GO  
/***** Object: StoredProcedure [dbo].[spe_proveedor]    Script Date:  
13/02/2016 12:56:32 *****/  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
create proc [dbo].[spe_proveedor]  
@idproveedor int,  
@razon_social varchar(150),  
@sector_comercial varchar(50),  
@tipo_documento varchar(20),  
@num_documento varchar(11),  
@direccion varchar(100),  
@telefono varchar(10),  
@email varchar(50),  
@url varchar(100)  
as  
update proveedor set razon_social=@razon_social,  
sector_comercial=@sector_comercial,  
tipo_documento=@tipo_documento,num_documento=@num_documento,
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
direccion=@direccion,telefono=@telefono,  
email=@email,url=@url  
where idproveedor=@idproveedor
```

```
GO  
/***** Object: StoredProcedure [dbo].[sped_cliente]    Script Date:  
13/02/2016 12:56:32 *****/  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
CREATE proc [dbo].[sped_cliente]  
@idcliente int,  
@nombre varchar(20),  
@apellidos varchar(40),  
@sexo varchar(1),  
@fecha_nacimiento date,  
@tipo_documento varchar(20),  
@num_documento varchar(20),  
@direccion varchar(100),  
@telefono varchar(10),  
@email varchar(50)  
as  
update cliente set nombre=@nombre,apellidos=@apellidos,sexo=@sexo,  
fecha_nacimiento=@fecha_nacimiento,  
tipo_documento=@tipo_documento,num_documento=@num_documento,  
direccion=@direccion,telefono=@telefono,email=@email  
where idcliente=@idcliente  
GO  
/***** Object: StoredProcedure [dbo].[sped_trabajador]    Script Date:  
13/02/2016 12:56:32 *****/  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
CREATE proc [dbo].[sped_trabajador]  
@idtrabajador int,  
@nombre varchar(20),  
@apellidos varchar(40),  
@sexo varchar(1),  
@fecha_nacimiento date,  
@num_documento varchar(15),  
@direccion varchar(100),  
@telefono varchar(10),  
@email varchar(50),  
@acceso varchar (20),  
@usuario varchar (20),  
@password varchar(20)  
as  
update trabajador set nombre=@nombre,apellidos=@apellidos,sexo=@sexo,  
fecha_nac=@fecha_nacimiento,  
num_documento=@num_documento,  
direccion=@direccion,telefono=@telefono,email=@email,  
acceso=@acceso,usuario=@usuario,password=@password  
where idtrabajador=@idtrabajador  
GO
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```

/***** Object: StoredProcedure [dbo].[speditar_categoria]    Script Date:
13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
--Procedimiento Editar
create proc [dbo].[speditar_categoria]
@idcategoria int,
@nombre varchar(50),
@descripcion varchar(256)
as
update categoria set nombre=@nombre,
descripcion=@descripcion
where idcategoria=@idcategoria

GO
/***** Object: StoredProcedure [dbo].[speditar_presentacion]    Script
Date: 13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create proc [dbo].[speditar_presentacion]
@idpresentacion int,
@nombre varchar (50),
@descripcion varchar (256)
as
update presentacion set nombre = @nombre, descripcion = @descripcion
where idpresentacion = @idpresentacion

GO
/***** Object: StoredProcedure [dbo].[spel_articulo]    Script Date:
13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create proc [dbo].[spel_articulo]
@idarticulo int
as
delete from articulo
where idarticulo = @idarticulo

GO
/***** Object: StoredProcedure [dbo].[spel_cliente]    Script Date:
13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create proc [dbo].[spel_cliente]
@idcliente int
as
delete from cliente
where idcliente=@idcliente

```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
GO
/***** Object: StoredProcedure [dbo].[spel_proveedor]    Script Date:
13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create proc [dbo].[spel_proveedor]
@idproveedor int
as
delete from proveedor
where idproveedor=@idproveedor

GO
/***** Object: StoredProcedure [dbo].[spel_trabajador]    Script Date:
13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

-- Procedimiento Eliminar Trabajador
create proc [dbo].[spel_trabajador]
@idtrabajador int
as
delete from trabajador
where idtrabajador=@idtrabajador

GO
/***** Object: StoredProcedure [dbo].[speliminar_categoria]    Script
Date: 13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
--Procedimiento Eliminar
create proc [dbo].[speliminar_categoria]
@idcategoria int
as
delete from categoria
where idcategoria=@idcategoria

GO
/***** Object: StoredProcedure [dbo].[speliminar_presentacion]    Script
Date: 13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create proc [dbo].[speliminar_presentacion]
@idpresentacion int
as
delete from presentacion
where idpresentacion = @idpresentacion

GO
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
/****** Object: StoredProcedure [dbo].[speliminar_venta]    Script Date:
13/02/2016 12:56:32 *****/
```

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create proc [dbo].[speliminar_venta]
@idventa int
as
delete from venta
where idventa=@idventa
```

```
GO
/****** Object: StoredProcedure [dbo].[spi_articulo]    Script Date:
13/02/2016 12:56:32 *****/
```

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create proc [dbo].[spi_articulo]
@idarticulo int output,
@codigo varchar(50),
@nombre varchar(50),
@descripcion varchar(1024),
@imagen image,
@idcategoria int
as

insert into articulo (codigo, nombre, descripcion, imagen, idcategoria)
values (@codigo, @nombre, @descripcion, @imagen, @idcategoria)
```

```
GO
/****** Object: StoredProcedure [dbo].[spi_cliente]    Script Date:
13/02/2016 12:56:32 *****/
```

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE proc [dbo].[spi_cliente]
@idcliente int output,
@nombre varchar(20),
@apellidos varchar(40),
@sexo varchar(1),
@fecha_nacimiento date,
@tipo_documento varchar(20),
@num_documento varchar(15),
@direccion varchar(100),
@telefono varchar(10),
@email varchar(50)
as
insert into
cliente(nombre,apellidos,sexo,fecha_nacimiento,tipo_documento,num_documento,
direccion,telefono,email)
values
(@nombre,@apellidos,@sexo,@fecha_nacimiento,@tipo_documento,@num_documento,@
direccion,@telefono,@email)
GO
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```

/***** Object: StoredProcedure [dbo].[spi_proveedor]    Script Date:
13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE proc [dbo].[spi_proveedor]
@idproveedor int output,
@razon_social varchar(150),
@sector_comercial varchar(50),
@tipo_documento varchar(20),
@num_documento varchar(15),
@direccion varchar(100),
@telefono varchar(15),
@email varchar(50),
@url varchar(100)
as
insert into proveedor(razon_social,sector_comercial,tipo_documento,
num_documento,direccion,telefono,email,url)
values (@razon_social,@sector_comercial,@tipo_documento,
@num_documento,@direccion,@telefono,@email,@url)

GO
/***** Object: StoredProcedure [dbo].[spi_trabajador]    Script Date:
13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE proc [dbo].[spi_trabajador]
@idtrabajador int output,
@nombre varchar(20),
@apellidos varchar(40),
@sexo varchar(1),
@fecha_nacimiento date,
@num_documento varchar(15),
@direccion varchar(100),
@telefono varchar(10),
@email varchar(50),
@acceso varchar (20),
@usuario varchar (20),
@password varchar(20)
as
insert into trabajador(nombre,apellidos,sexo,fecha_nac,num_documento,
direccion,telefono,email, acceso,usuario,password)
values (@nombre,@apellidos,@sexo,@fecha_nacimiento,
@num_documento,@direccion,@telefono,@email,@acceso,@usuario,@password)
GO
/***** Object: StoredProcedure [dbo].[spinsertar_categoria]    Script
Date: 13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
--Procedimiento Insertar
create proc [dbo].[spinsertar_categoria]
@idcategoria int output,
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
@nombre varchar(50),
@descripcion varchar(256)
as
insert into categoria (nombre,descripcion)
values (@nombre,@descripcion)

GO
/***** Object: StoredProcedure [dbo].[spinsertar_detalle_ingreso]
Script Date: 13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE proc [dbo].[spinsertar_detalle_ingreso]
@iddetalle_ingreso int output,
@idingreso int,
@idarticulo int,
@precio_compra money,
@precio_venta money,
@stock_inicial int,
@stock_actual int
as
insert into detalle_ingreso (idingreso,idarticulo,precio_compra,
precio_venta,stock_inicial,stock_actual)
values (@idingreso,@idarticulo,@precio_compra,
@precio_venta,@stock_inicial,@stock_actual)
GO
/***** Object: StoredProcedure [dbo].[spinsertar_detalle_venta]      Script
Date: 13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create proc [dbo].[spinsertar_detalle_venta]
@iddetalle_venta int output,
@idventa int,
@iddetalle_ingreso int,
@cantidad int,
@precio_venta money,
@descuento money
as
insert into detalle_venta (idventa,iddetalle_ingreso,cantidad,
precio_venta,descuento)
values (@idventa,@iddetalle_ingreso,@cantidad,
@precio_venta,@descuento)

GO
/***** Object: StoredProcedure [dbo].[spinsertar_ingreso]      Script Date:
13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create proc [dbo].[spinsertar_ingreso]
@idingreso int=null output,
@idtrabajador int,
@idproveedor int,
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
@fecha date,  
@tipo_comprobante varchar(20),  
@serie varchar(4),  
@correlativo varchar(7),  
@iva decimal(4,2),  
@estado varchar(7)  
as  
insert into  
ingreso(idtrabajador,idproveedor,fecha,tipo_comprobante,serie,correlativo,iva,  
estado)  
values  
(@idtrabajador,@idproveedor,@fecha,@tipo_comprobante,@serie,@correlativo,@iva,  
@estado)  
--Obteniendo el codigo autogenerado del ingreso  
SET @idingreso = @@IDENTITY
```

```
GO  
/***** Object: StoredProcedure [dbo].[spinsertar_presentacion] Script  
Date: 13/02/2016 12:56:32 *****/  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
create proc [dbo].[spinsertar_presentacion]  
@idpresentacion int output,  
@nombre varchar (50),  
@descripcion varchar (256)  
as  
insert into presentacion(nombre, descripcion)  
values (@nombre, @descripcion)
```

```
GO  
/***** Object: StoredProcedure [dbo].[spinsertar_venta] Script Date:  
13/02/2016 12:56:32 *****/  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO  
CREATE proc [dbo].[spinsertar_venta]  
@idventa int=null output,  
@idtrabajador int,  
@idcliente int,  
@fecha date,  
@tipo_comprobante varchar(20),  
@serie varchar(4),  
@correlativo varchar(7),  
@iva decimal(4,2)  
as  
insert into  
venta(idtrabajador,idcliente,fecha,tipo_comprobante,serie,correlativo,iva)  
values  
(@idtrabajador,@idcliente,@fecha,@tipo_comprobante,@serie,@correlativo,@iva)  
--Obteniendo el codigo autogenerado de la venta  
SET @idventa = @@IDENTITY  
GO  
/***** Object: StoredProcedure [dbo].[splogin] Script Date: 13/02/2016  
12:56:32 *****/
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create proc [dbo].[splogin]
@usuario varchar(20),
@password varchar(20)
as
select idtrabajador,apellidos,nombre,acceso from trabajador
where usuario=@usuario and
password=(select password from trabajador where usuario=@usuario)

GO
/***** Object: StoredProcedure [dbo].[spm_cliente]    Script Date:
13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create proc [dbo].[spm_cliente]
as
SELECT top 100 * FROM cliente
order by apellidos asc

GO
/***** Object: StoredProcedure [dbo].[spm_proveedor]    Script Date:
13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create proc [dbo].[spm_proveedor]
as
SELECT top 100 * FROM proveedor
order by razon_social asc

GO
/***** Object: StoredProcedure [dbo].[spm_trabajador]    Script Date:
13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
--Procedimiento Mostrar Trabajador
create proc [dbo].[spm_trabajador]
as
SELECT * FROM trabajador
order by apellidos asc

GO
/***** Object: StoredProcedure [dbo].[spmostrar_articulo]    Script Date:
13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create proc [dbo].[spmostrar_articulo]
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

as

```
SELECT top 100 dbo.articulo.idarticulo, dbo.articulo.codigo,
dbo.articulo.nombre, dbo.articulo.descripcion, dbo.articulo.imagen,
dbo.articulo.idcategoria, dbo.categoria.nombre
AS Categoria
FROM dbo.articulo INNER JOIN
                        dbo.categoria ON dbo.articulo.idcategoria =
dbo.categoria.idcategoria
                        order by dbo.articulo.idarticulo
desc
```

```
GO
/***** Object: StoredProcedure [dbo].[spmostrar_categoria]    Script Date:
13/02/2016 12:56:32 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
create proc [dbo].[spmostrar_categoria]
```

```
as
```

```
select top 200 * from categoria
```

```
order by idcategoria desc
```

```
GO
```

```
/***** Object: StoredProcedure [dbo].[spmostrar_detalle_ingreso]    Script
Date: 13/02/2016 12:56:32 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE proc [dbo].[spmostrar_detalle_ingreso]
```

```
@textobuscar int
```

```
as
```

```
select d.idarticulo,a.nombre as Articulo,d.precio_compra,
```

```
d.precio_venta,d.stock_inicial,d.stock_actual,(d.stock_inicial*d.precio_comp
ra) as Subtotal
```

```
from detalle_ingreso d inner join articulo a
```

```
on d.idarticulo=a.idarticulo
```

```
where d.idingreso=@textobuscar
```

```
GO
```

```
/***** Object: StoredProcedure [dbo].[spmostrar_detalle_venta]    Script
Date: 13/02/2016 12:56:32 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
create proc [dbo].[spmostrar_detalle_venta]
```

```
@textobuscar int
```

```
as
```

```
select d.iddetalle_ingreso,a.nombre as Articulo,
```

```
d.cantidad,d.precio_venta,d.descuento,
```

```
((d.precio_venta*d.cantidad)-d.descuento) as Subtotal
```

```
from detalle_venta d inner join detalle_ingreso di
```

```
on d.iddetalle_ingreso=di.iddetalle_ingreso
```

```
inner join articulo a
```

```
on di.idarticulo=a.idarticulo
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
where d.idventa=@textobuscar

GO
/***** Object: StoredProcedure [dbo].[spmostrar_ingreso]    Script Date:
13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create proc [dbo].[spmostrar_ingreso]
as
SELECT top 100 i.idingreso,
(t.apellidos + ' ' + t.nombre) as Trabajador,
p.razon_social as proveedor,
i.fecha, i.tipo_comprobante,
i.serie, i.correlativo,
i.estado, sum(d.precio_compra*
d.stock_inicial) as Total
FROM detalle_ingreso d INNER JOIN ingreso i
ON d.idingreso = i.idingreso
INNER JOIN proveedor p
ON i.idproveedor = p.idproveedor
INNER JOIN trabajador t
ON i.idtrabajador = t.idtrabajador
group by
i.idingreso,
t.apellidos + ' ' + t.nombre,
p.razon_social,
i.fecha, i.tipo_comprobante,
i.serie, i.correlativo,
i.estado
order by i.idingreso desc

GO
/***** Object: StoredProcedure [dbo].[spmostrar_presentacion]    Script
Date: 13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create proc [dbo].[spmostrar_presentacion]
as
select top 200 * from presentacion
order by idpresentacion desc

GO
/***** Object: StoredProcedure [dbo].[spmostrar_venta]    Script Date:
13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create proc [dbo].[spmostrar_venta]
as
SELECT top 100 v.idventa,
(t.apellidos + ' ' + t.nombre) as Trabajador,
(c.apellidos + ' ' + c.nombre) as Cliente,
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
v.fecha, v.tipo_comprobante,
v.serie, v.correlativo, sum((d.precio_venta*
d.cantidad)-d.descuento) as Total
FROM detalle_venta d INNER JOIN venta v
ON d.idventa = v.idventa
INNER JOIN cliente c
ON v.idcliente = c.idcliente
INNER JOIN trabajador t
ON v.idtrabajador = t.idtrabajador
group by
v.idventa,
t.apellidos +' '+ t.nombre,
c.apellidos+' '+c.nombre,
v.fecha, v.tipo_comprobante,
v.serie, v.correlativo
order by v.idventa desc
```

```
GO
/***** Object: StoredProcedure [dbo].[spreporte_factura]    Script Date:
13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create proc [dbo].[spreporte_factura]
@idventa int
as
SELECT v.idventa,
(t.apellidos +' '+ t.nombre) as Trabajador,
(c.apellidos + ' ' + c.nombre) as Cliente,
c.direccion,c.telefono,c.num_documento,
v.fecha, v.tipo_comprobante,
v.serie, v.correlativo,
a.nombre,v.iva,
d.precio_venta,d.cantidad,d.descuento,
(d.precio_venta*d.cantidad-d.descuento) as total_parcial
FROM detalle_venta d inner join detalle_ingreso di
on d.iddetalle_ingreso=di.iddetalle_ingreso
inner join articulo a
on di.idarticulo=a.idarticulo
INNER JOIN venta v
ON d.idventa = v.idventa
INNER JOIN cliente c
ON v.idcliente = c.idcliente
INNER JOIN trabajador t
ON v.idtrabajador = t.idtrabajador
where v.idventa=@idventa
```

```
GO
/***** Object: StoredProcedure [dbo].[spreporte_venta]    Script Date:
13/02/2016 12:56:32 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create proc [dbo].[spreporte_venta]
@idventa int
```

DESARROLLO DEL SISTEMA DE CONTROL DE ADQUISICIÓN Y VENTA DE PRODUCTOS TECNOLÓGICOS INCLUYENDO UN MÓDULO DE FACTURACIÓN ELECTRÓNICA, APLICADO A LA EMPRESA COMPUTER SOLUTION

```
as
SELECT v.idventa,
(t.apellidos + ' ' + t.nombre) as Trabajador,
(c.apellidos + ' ' + c.nombre) as Cliente,
c.direccion,c.telefono,c.num_documento,
v.fecha, v.tipo_comprobante,
v.serie, v.correlativo,a.nombre,v.iva, d.precio_venta,d.cantidad,d.descuento
FROM detalle_venta d inner join detalle_ingreso di
on d.iddetalle_ingreso=di.iddetalle_ingreso
inner join articulo a
on di.idarticulo=a.idarticulo
INNER JOIN venta v
ON d.idventa = v.idventa
INNER JOIN cliente c
ON v.idcliente = c.idcliente
INNER JOIN trabajador t
ON v.idtrabajador = t.idtrabajador
where v.idventa=@idventa

GO
USE [master]
GO
ALTER DATABASE [dbventas] SET READ_WRITE
GO
```