



**UNIVERSIDAD TÉCNICA DE AMBATO**

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E  
INDUSTRIAL**

**Carrera de Ingeniería en Sistemas Computacionales e  
Informáticos**

**TEMA:**

---

**“SISTEMA DE FACTURACIÓN E INVENTARIOS PARA EL CONTROL  
TRIBUTARIO DE COMPRA Y VENTA EN LA CORPORACIÓN VPC”**

---

**Trabajo de Graduación Modalidad TEMI.** (Trabajo Estructurado de Manera Independiente), presentado como requisito previo a la obtención del Título de Ingeniero en Sistemas Computacionales e Informáticos.

**AUTOR:** Franklin Ricardo Barrionuevo Caiza

**TUTOR:** Ing. Galo López

Ambato – Ecuador

**2012**

## **APROBACIÓN DEL TUTOR**

En calidad de Tutor del Trabajo de Investigación sobre el tema: “**SISTEMA DE FACTURACIÓN E INVENTARIOS PARA EL CONTROL TRIBUTARIO DE COMPRA Y VENTA EN LA CORPORACIÓN VPC**”, de Franklin Ricardo Barrionuevo Caiza, egresado de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato, considero que dicho informe investigativo reúne los requisitos y méritos suficientes para ser sometidos a la evaluación del Tribunal de Grado, que el Honorable Consejo Directivo designe, para su correspondiente estudio y calificación.

Ambato, Abril 2012

EL TUTOR

---

Ing. Galo López

## AUTORÍA

El presente trabajo de investigación titulado “**SISTEMA DE FACTURACIÓN E INVENTARIOS PARA EL CONTROL TRIBUTARIO DE COMPRA Y VENTA EN LA CORPORACIÓN VPC**”, es totalmente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son exclusiva responsabilidad del autor.

Ambato, Abril 2012

---

Franklin Ricardo Barrionuevo Caiza

CC.: 180350596-3

## **APROBACIÓN DE LA COMISIÓN CALIFICADORA**

La Comisión Calificadora del presente trabajo conformada por los señores docentes Dr. Mauricio Carranza y el Ing. Clay Aldás, reviso y aprobó el Informe Final del trabajo de graduación titulado: **“SISTEMA DE FACTURACIÓN E INVENTARIOS PARA EL CONTROL TRIBUTARIO DE COMPRA Y VENTA EN LA CORPORACIÓN VPC”**, presentado por el señor Franklin Ricardo Barrionuevo Caiza de acuerdo al Art. 18 del Reglamento de Graduación para Obtener el Título Terminal de Tercer Nivel de la Universidad Técnica de Ambato.

Ing. Oswaldo Paredes

**PRESIDENTE DEL TRIBUNAL**

Dr. Mauricio Carranza

**DOCENTE CALIFICADOR**

Ing. Clay Aldás

**DOCENTE CALIFICADOR**

## DEDICATORIA

*A mi madre Gladys por haberme apoyado en todo momento, por sus consejos y valores que me ha permitido ser una persona de bien.*

*A mi padre Angel quien es una persona de pocas palabras pero en lo profundo de su corazón siempre deseo lo mejor para mí.*

*A mi hermana Verónica y su esposo Santiago por el apoyo incondicional que me han brindado; a mis adorados niños Anthony, Erick y Nahomy quienes han sido fruto de inspiración para seguir adelante; a mis amigos y a todos aquellos que participaron directa o indirectamente en la elaboración de este proyecto.*

*Franklín Ricardo Barrionuevo Caiza*

## AGRADECIMIENTO

*A Dios por haberme dado la vida y colmarla de bendiciones, permitiéndome llegar hasta este punto de lograr mis objetivos.*

*A mis profesores, quienes han brindado sus conocimientos y hoy pueden ver un reflejo de lo que han formado.*

*A mi profesor guía, Ing. Galo López, por su gran apoyo y motivación en la elaboración del trabajo de graduación.*

*A la Universidad Técnica de Ambato por haberme abierto sus puertas y permitir mi formación en sus instalaciones.*

*A la empresa Ver Paint Corp, la cuál abrió sus puertas e hizo posible la elaboración de este proyecto.*

*Franklín Ricardo Barrionuevo Caíza*

# ÍNDICE

## Índice General

CONTENIDO	PÁGINA
APROBACIÓN DEL TUTOR.....	ii
AUTORÍA .....	iii
APROBACIÓN DE LA COMISIÓN CALIFICADORA .....	iv
DEDICATORIA .....	v
AGRADECIMIENTO .....	vi
ÍNDICE.....	vii
RESUMEN EJECUTIVO .....	xvii
INTRODUCCIÓN .....	xviii
CAPÍTULO I .....	1
EL PROBLEMA.....	1
1.1 Tema de investigación.....	1
1.2 Planteamiento del problema .....	1
1.2.1 Contextualización.....	1
1.2.2 Análisis Crítico.....	3
1.2.3 Prognosis.....	3
1.3 Formulación del problema.....	4
1.3.1 Preguntas directrices.....	4
1.3.2 Delimitación del problema .....	4
1.4 Justificación .....	4
1.5 Objetivos .....	5
1.5.1 Objetivo General .....	5
1.5.2 Objetivos Específicos .....	5
CAPÍTULO II.....	6
MARCO TEÓRICO.....	6
2.1 Antecedentes Investigativos .....	6
2.2 Fundamentación Legal .....	7
2.3 Categorías Fundamentales.....	11
2.3.1 Informática .....	12

2.3.2 Sistema Informático .....	12
2.3.3 Base de Datos .....	13
<b>2.3.3.1 Postgres</b> .....	13
<b>2.3.3.2 MySql</b> .....	14
<b>2.3.3.3 Oracle</b> .....	14
<b>2.3.3.4 SqlServer</b> .....	14
2.3.4 Lenguajes de Programación.....	15
<b>2.3.4.1 PHP</b> .....	15
<b>2.3.4.2 C#</b> .....	16
<b>2.3.4.3 JavaScript</b> .....	16
<b>2.3.4.4 ASP.NET</b> .....	16
2.3.5 Sistema de Facturación e Inventarios .....	16
<b>2.3.5.1 Diseño de Base de Datos</b> .....	16
<b>2.3.5.2 Diseño del Sistema</b> .....	17
<b>2.3.5.3 Creación del sistema Informático</b> .....	17
<b>2.3.5.4 Implantación</b> .....	17
2.3.6 Corporación VPC (Ver Paint Corp).....	18
2.3.7 Control de Inventarios .....	18
2.3.8 Facturación .....	18
2.3.9 Compra y Venta .....	19
2.3.10 Control tributario de compra y venta de la empresa .....	19
<b>2.3.10.1 Clientes</b> .....	19
<b>2.3.10.2 Proveedores</b> .....	19
<b>2.3.10.3 Tributación</b> .....	19
2.4 Hipótesis .....	20
2.5 Señalamiento de Variables de la Hipótesis .....	20
2.5.1 Variable Independiente.....	20
2.5.2 Variable Dependiente .....	20
CAPÍTULO III.....	21
METODOLOGÍA .....	21

3.1 Enfoque .....	21
3.2 Modalidad básica de la investigación .....	21
3.2.1 Investigación de campo .....	21
3.2.2 Investigación Bibliográfica – Documental .....	21
3.2.3 Proyecto Factible .....	22
3.3 Nivel o tipo de investigación .....	22
3.4 Población y muestra .....	22
3.4.1 Población.....	22
3.4.2 Muestra.....	22
3.5 Operacionalización de las variables.....	22
3.6. Recolección de la Información .....	25
3.6.1 Plan de recolección de información.....	25
3.6.2 Procesamiento y Análisis de la Información.....	25
3.7 Análisis e interpretación de resultados .....	25
CAPÍTULO IV .....	26
ANÁLISIS E INTERPRETACIÓN DE RESULTADOS .....	26
4.1 Análisis de la necesidad .....	26
4.2 Análisis de los resultados .....	27
4.3 Interpretación de resultados.....	35
CAPÍTULO V .....	36
CONCLUSIONES Y RECOMENDACIONES .....	36
5.1 Conclusiones .....	36
5.2 Recomendaciones.....	38
CAPÍTULO VI .....	40
PROPUESTA.....	40
6.1 Tema .....	40
6.2 Datos Informativos.....	40
6.3 Antecedentes .....	40
6.4 Justificación .....	41
6.5 Objetivos.....	42

6.5.1 Objetivo General .....	42
6.5.2 Objetivos Específicos .....	42
6.6 Análisis de factibilidad .....	43
6.6.1 Factibilidad Operativa .....	43
6.6.2 Factibilidad Económica .....	43
6.6.3 Factibilidad Técnica .....	43
6.7 Fundamentación .....	45
6.7.1 C Sharp .....	45
6.7.2 CSS .....	45
6.7.3 JavaScript .....	46
6.7.4 Internet Information Server (IIS) .....	46
6.7.5 iTextSharp .....	47
6.8. Metodología .....	47
6.9 Modelo Operativo .....	48
6.9.1 Análisis del Sistema .....	48
<b>6.9.1.1 Análisis y Requerimientos del Sistema</b> .....	48
<b>6.9.1.2 Diagramas UML</b> .....	49
<b>6.9.1.2.1 Diagramas de Caso de Uso</b> .....	49
<b>6.9.1.2.2 Diseño de la Interfaz</b> .....	53
<b>6.9.1.2.3 Diagramas de Clase</b> .....	61
<b>6.9.1.2.4 Diagramas de Secuencia</b> .....	66
6.9.2 Diseño del Sistema .....	71
<b>6.9.2.1 Diseño de la Base de Datos</b> .....	71
<b>6.9.2.2 Diccionario de Datos</b> .....	76
<b>6.9.2.3 Desarrollo del Sistema</b> .....	83
6.9.3 Pruebas .....	90
<b>6.9.3.1 Pruebas de Caja Blanca</b> .....	90
<b>6.9.3.2 Prueba de Caja Negra</b> .....	91
<b>6.9.3.3 Pruebas de Verificación y Validación</b> .....	91
6.9.4 Implantación del Sistema .....	97

<b>6.9.4.1 Cronograma de Implantación del Sistema</b> .....	98
<b>6.9.4.2 Levantamiento del Sistema</b> .....	98
6.10 Conclusiones y Recomendaciones .....	98
6.10.1 Conclusiones .....	98
6.10.2 Recomendaciones .....	99
6.11 Bibliografía .....	100
6.11.1 Información Bibliográfica de Libros .....	100
6.11.2 Información Bibliográfica Web .....	100
6.12 Glosario de Términos .....	102
ANEXOS .....	104
ANEXO 1: MODELO DE LA ENTREVISTA.....	105
ANEXO 2: MANUAL DE INSTALACIÓN DE SOFTWARE .....	107
ANEXO 3: PUBLICACIÓN DE LAS FUENTES .....	118
ANEXO 4: SCRIPT DE COMANDOS SQL .....	119
ANEXO 5: EXPLICACIÓN DEL FUNCIONAMIENTO DEL CÓDIGO FUENTE .....	140
ANEXO 6: INSTALACIÓN DEL SISTEMA.....	146
ANEXO 7: Manual de Usuario .....	148

## ÍNDICE DE FIGURAS

CONTENIDO	PÁGINA
<b>Figura 1.1</b> Inclusión Interrelacionado de las V. Independiente y Dependiente .....	11
<b>Figura 6.1</b> Caso de Uso Ingreso Sistema. ....	49
<b>Figura 6.2</b> Caso de Uso Administración Factura. ....	49
<b>Figura 6.3</b> Caso de Uso Administración Inventario.....	50
<b>Figura 6.4</b> Caso de Uso Administración Compras.....	50
<b>Figura 6.5</b> Caso de Uso Administración Créditos.....	51
<b>Figura 6.6</b> Caso de Uso Administración Clientes.....	51
<b>Figura 6.7</b> Caso de Uso Administración Color Estándar. ....	51
<b>Figura 6.8</b> Caso de Uso Administración Generación de Colores. ....	52
<b>Figura 6.9</b> Caso de Uso Administración Parámetros. ....	52
<b>Figura 6.10</b> Caso de Uso Administración Seguridad.....	53
<b>Figura 6.11</b> Diseño de la Interfaz Pantalla Principal. ....	53
<b>Figura 6.12</b> Diseño de la Interfaz Inicio Sesión.....	54
<b>Figura 6.13</b> Diseño de la Interfaz Vista de Facturas. ....	54
<b>Figura 6.14</b> Diseño de la Interfaz Administración de Facturas.....	55
<b>Figura 6.15</b> Diseño de la Interfaz Administración Compras. ....	58
<b>Figura 6.16</b> Diseño de la Interfaz Administración Clientes.....	58
<b>Figura 6.17</b> Diseño de la Interfaz Administración Color.....	59
<b>Figura 6.18</b> Diseño de la Interfaz Administración Generación Color. ....	60
<b>Figura 6.19</b> Diseño de la Interfaz Administración Usuarios.....	60
<b>Figura 6.20</b> Diagrama de Clase Facturación.....	61
<b>Figura 6.21</b> Diagrama de Clase Compras. ....	62
<b>Figura 6.22</b> Diagrama de Clase Clientes.....	63
<b>Figura 6.23</b> Diagrama de Clase Colores. ....	63
<b>Figura 6.24</b> Diagrama de Clase Créditos. ....	64
<b>Figura 6.25</b> Diagrama de Clase Históricos. ....	64
<b>Figura 6.26</b> Diagrama de Clase Generales.....	65

<b>Figura 6.27</b> Diagrama de Secuencia Nueva Factura.....	67
<b>Figura 6.28</b> Diagrama de Secuencia Ver Factura. ....	68
<b>Figura 6.29</b> Diagrama de Secuencia Ver Inventario.....	68
<b>Figura 6.30</b> Diagrama de Secuencia Reporte Inventario. ....	69
<b>Figura 6.31</b> Diagrama de Secuencia Cobro Crédito. ....	69
<b>Figura 6.32</b> Diagrama de Secuencia Consulta Crédito.....	70
<b>Figura 6.33</b> Diagrama de Secuencia Compras.....	70
<b>Figura 6.34</b> Diseño de la Base de Datos (Facturación).....	71
<b>Figura 6.35</b> Diseño de la Base de Datos (Compras). ....	72
<b>Figura 6.36</b> Diseño de la Base de Datos (Clientes).....	73
<b>Figura 6.37</b> Diseño de la Base de Datos (Colores). ....	73
<b>Figura 6.38</b> Diseño de la Base de Datos (Créditos). ....	74
<b>Figura 6.39</b> Diseño de la Base de Datos (Históricos). ....	74
<b>Figura 6.40</b> Diseño de la Base de Datos (Generales).....	75
<b>Figura 6.41</b> Pruebas Inicio Sesión.....	92
<b>Figura 6.42</b> Pruebas Pantalla principal. ....	92
<b>Figura 6.43</b> Pruebas Creación Usuario. ....	93
<b>Figura 6.44</b> Pruebas Usuario creado correctamente.....	93
<b>Figura 6.45</b> Pruebas Administración Usuarios. ....	93
<b>Figura 6.46</b> Pruebas Actualizando color estándar.....	94
<b>Figura 6.47</b> Pruebas Generar color maestro.....	94
<b>Figura 6.48</b> Pruebas Creación de Compra. ....	94
<b>Figura 6.49</b> Pruebas Ingreso Información Adicional de Compra. ....	95
<b>Figura 6.50</b> Pruebas Inventario antes de la Compra. ....	95
<b>Figura 6.51</b> Pruebas Inventario después de la Compra.....	95
<b>Figura 6.52</b> Pruebas Inicio Factura. ....	95
<b>Figura 6.53</b> Pruebas Creación Factura. ....	96
<b>Figura 6.54</b> Pruebas Impresión Factura. ....	96
<b>Figura 6.55</b> Pruebas Inventario antes de la Factura. ....	96
<b>Figura 6.56</b> Pruebas Inventario después de la Factura.....	97

## ÍNDICE DE TABLAS

CONTENIDO	PÁGINA
<b>Tabla 3.1</b> Operacionalización de variables – Variable independiente .....	23
<b>Tabla 3.2</b> Operacionalización de variables – Variable dependiente .....	24
<b>Tabla 4.1</b> Cuadro porcentual pregunta 1 .....	29
<b>Tabla 4.2</b> Cuadro porcentual pregunta 2 .....	29
<b>Tabla 4.3</b> Cuadro porcentual pregunta 3 .....	30
<b>Tabla 4.4</b> Cuadro porcentual pregunta 4 .....	31
<b>Tabla 4.5</b> Cuadro porcentual pregunta 5 .....	31
<b>Tabla 4.6</b> Cuadro porcentual pregunta 6 .....	32
<b>Tabla 4.7</b> Cuadro porcentual pregunta 7 .....	32
<b>Tabla 4.8</b> Cuadro porcentual pregunta 8 .....	33
<b>Tabla 4.9</b> Cuadro porcentual pregunta 9 .....	34
<b>Tabla 4.10</b> Cuadro porcentual pregunta 10 .....	34
<b>Tabla 6.1</b> "Facturacion"."FacturaMaestro" .....	76
<b>Tabla 6.2</b> "Facturacion"."FacturaCabecera" .....	76
<b>Tabla 6.3</b> "Facturacion"."FacturaDetalle" .....	77
<b>Tabla 6.4</b> "Facturacion"."FacturaDesglose" .....	77
<b>Tabla 6.5</b> "Facturacion"."FacturaCodigosSRI" .....	77
<b>Tabla 6.6</b> "Facturacion"."FacturaFormaPago" .....	77
<b>Tabla 6.7</b> "Compras"."CompraMaestro" .....	78
<b>Tabla 6.8</b> "Compras"."CompraDetalle" .....	78
<b>Tabla 6.9</b> "Clientes"."Cliente" .....	78
<b>Tabla 6.10</b> "Clientes"."TelefonoCliente" .....	79
<b>Tabla 6.11</b> "Clientes"."FormaDeVenta" .....	79
<b>Tabla 6.12</b> "Colores"."ColorEstandar" .....	79
<b>Tabla 6.13</b> "Colores"."GeneracionColorMaestro" .....	79
<b>Tabla 6.14</b> "Colores"."GeneracionColorDetalle" .....	80
<b>Tabla 6.15</b> "Colores"."ColorEstandarSucursal" .....	80
<b>Tabla 6.16</b> "Credito"."CreditoMaestro" .....	80

<b>Tabla 6.17</b> "Credito"."CreditoComponente" .....	81
<b>Tabla 6.18</b> "Credito"."MovimientoCreditoComponente" .....	81
<b>Tabla 6.19</b> "Históricos"."HistoricoColorEstandar" .....	81
<b>Tabla 6.20</b> "Históricos"."HistóricoSucursales" .....	81
<b>Tabla 6.21</b> "Históricos"."HistoricoColorEstandarSucursal" .....	82
<b>Tabla 6.22</b> "Generales"."Sucursales" .....	82
<b>Tabla 6.23</b> "Generales"."Calendario" .....	82
<b>Tabla 6.24</b> "Generales"."Vendedor" .....	82
<b>Tabla 6.25</b> Cronograma de Implantación del Sistema .....	98

## **RESUMEN EJECUTIVO**

El tema del presente trabajo trata sobre el desarrollo e implementación de un Sistema de Facturación e Inventarios para el control tributario de compra y venta en la Corporación VPC (Ver Paint Corp). El Trabajo Estructurado de Manera Independiente (TEMI) está separado en 6 capítulos donde se tratarán diversos aspectos de la aplicación como las bases sobre la cual parte la propuesta realizada, su diseño, desarrollo y los resultados obtenidos durante las diferentes pruebas realizadas antes y después de su implantación.

En el **CAPÍTULO I** se plantea el problema por el cual se decidió realizar el presente proyecto, se menciona los objetivos que llevaron a la realización de la investigación, así como también una justificación con argumentos claros con los que se sustenta el porqué del presente trabajo.

El **CAPÍTULO II** contiene los antecedentes investigativos, las fundamentaciones legales, las categorías fundamentales que nos guiarán en la búsqueda de una posible solución al problema planteado: así como la definición de las variables dependientes e independientes.

En el **CAPÍTULO III** se explicará la modalidad y el nivel de la investigación, la población y muestra, recolección y procesamiento de la información.

En el **CAPÍTULO IV** se realiza el análisis e interpretación de los resultados de las entrevistas y encuestas realizadas.

En el **CAPÍTULO V** se incluirán las conclusiones y recomendaciones de la investigación.

En el **CAPÍTULO VI** se describe detalladamente la propuesta planteada al problema de investigación.

## INTRODUCCIÓN

La carencia de un sistema de control de facturación e inventarios en la Corporación VPC es evidente ya que al ser distribuidores de pintura se cuenta con un sistema que realiza la mezcla de colores, este sistema es propio de los proveedores (PPG), siendo el inconveniente que este sistema no genera una factura, por lo que es necesario la creación de un sistema que pueda realizar la mezcla de colores y a la vez pueda facturar, al mismo tiempo ir afectando el inventario tanto al realizar ventas como al realizar compras. La investigación se desarrolla dentro de la corporación para conocer de una mejor manera el estado actual de la infraestructura de software, hardware y la red de datos.

El trabajo inicia con la investigación de la infraestructura de software y hardware existente, para conocer qué tipo de servidor existe, con qué sistema operativo trabaja, si la Institución tiene licencias de algún lenguaje de programación para el desarrollo del sistema o si se debe trabajar con un lenguaje de programación libre, con que motor de base de datos trabaja para el manejo de usuarios, aplicaciones soportadas por la red, acceso a Internet, entre otros.

La evolución de los lenguajes de programación y arquitecturas de software es constante lo cual conlleva a que el sistema a desarrollarse se base en tecnologías actuales.

La internet como medio de comunicación masivo orienta a que la mayor parte de aplicaciones se desarrollen en un ambiente web, siendo una de las principales ventajas la accesibilidad y multiplataforma de la aplicación, permitiendo a los usuarios del sistema acceder al mismo desde cualquier lugar del mundo sin importar la plataforma operativa que este corriendo.

Esto es la base para aplicar una metodología de análisis y diseño de nuestro sistema, con la cual se pretende dar solución a los procesos de Facturación e Inventarios que permitirá crear una herramienta que apoye a las actividades de la empresa.

# **CAPÍTULO I**

## **EL PROBLEMA**

### **1.1 Tema de investigación**

Sistema de Facturación e Inventarios para el control tributario de compra y venta en la Corporación VPC.

### **1.2 Planteamiento del problema**

#### **1.2.1 Contextualización**

Hoy en día no podemos entender a nuestra sociedad sin computadores o sin las redes de comunicación de datos que las enlazan, llegando a ser el centro y la base de información de nuestra sociedad. Son innumerables los beneficios que brinda la computación, rapidez en la obtención de resultados, almacenamiento de grandes volúmenes de información, facilidades para encontrar información adecuada y/o actualizada por parte de científicos, investigadores, profesionales y estudiantes.

La tecnología es un proceso de invención, fabricación y usos de objetivos y sistemas que contribuyen a la solución de problemas que se pueden plantear. Es uno de los motores del desarrollo de la sociedad actual, ya que ésta incorpora con asombrosa rapidez los descubrimientos tecnológicos a las acciones más triviales de la vida cotidiana.

El impacto de la tecnología en los procesos de facturación y de inventarios está fuera de toda duda. La tecnología opera como motor del cambio que permite dar respuesta a las nuevas necesidades de información. Al incorporar las tecnologías para control de facturación e inventarios se producen mejoras en velocidad que es hacer lo mismo pero más rápido.

En el Ecuador la tecnología ya no se está quedando atrás, estamos en un mundo de modernización en la cual se está mejorando los procesos en un mundo informático lo cual beneficia visiblemente en la administración de instituciones públicas y privadas. La búsqueda de un mejoramiento y la satisfacción de las necesidades de las personas es que nos ha llevado a encontrarnos con la tecnología ya que ahora en la mayoría de las instituciones encontramos sistemas informáticos que están solucionando problemas y ayudando en el crecimiento de las mismas.

En Tungurahua hoy en día la mayoría de instituciones públicas y privadas trabajan con diversos sistemas computacionales, entre estos sistemas se encuentra los sistemas de facturación e inventarios que es de gran ayuda al momento de registrar las transacciones comerciales obligatorias y aceptada por ley, con mayor rapidez y exactitud en el despliegue de resultados. De esta manera las diferentes instituciones han logrado estar involucradas en el mundo tecnológico que es una inversión pero obtienen resultados favorables ya que se a optimizado tiempo y su economía va mejorando notablemente.

En la Corporación VPC trabajan con un sistema que es muy tradicional en el mercado, este programa no cumple con las expectativas de la empresa y es conveniente crear un sistema de Facturación e Inventarios más detallado y preciso que permita conocer con exactitud los movimientos que se realiza en la empresa y ayude a tener mejores herramientas para una buena toma de decisiones en el momento oportuno.

### **1.2.2 Análisis Crítico**

Este estudio va dirigido a la corporación VPC en la cual se está llevando un control tributario de compra y venta a través del conocido programa de computación Mónica (programa de computador para negocios, permite realizar las facturas, controlar el inventario, realizar listas de precios, tener un archivo de clientes, proveedores, manejar las cuentas por cobrar, las cuentas corrientes, y la contabilidad básica de su empresa), sabiendo que este programa es muy popular en el mercado y sobre todo se va quedando en el pasado, la empresa ha optado por obtener un sistema propio y único para llevar a cabo dicho control. Con este sistema se quiere tener una óptima recepción y emisión de facturas con proveedores y clientes de la empresa, así también tener un módulo de inventarios en donde el usuario pueda tener un control exacto de los productos que se comercializa.

Se necesita también almacenar el estado del inventario diariamente para poder tener un historial de inventarios, para en lo posterior poder obtener reportes por fechas cuando el usuario lo requiera. Por otro lado la empresa tendría un adelanto tecnológico ya que hoy en día las empresas trabajan con sistemas propios más no con programas que traten de adaptarse a la empresa.

### **1.2.3 Prognosis**

Si no se toman acciones inmediatas para el desarrollo e implantación de un sistema de Facturación e Inventarios la Corporación VPC continuará trabajando con programas inadecuados teniendo insuficiencias al momento de realizar sus actividades diarias en lo que se refiere a facturación e inventarios y sobre todo teniendo un retraso tecnológico ya que hoy en día la mayoría de las empresas se están actualizando con nuevos sistemas los cuales son realizados a medida de las mismas. Si no se toman los correctivos necesarios no se podrá transparentar en el manejo de facturación e inventarios de una manera eficaz.

### **1.3 Formulación del problema**

¿Qué beneficios tiene el desarrollo de un Sistema de Facturación e Inventarios para un mejor control tributario de compra y venta en la Corporación VPC?

#### **1.3.1 Preguntas directrices**

¿La Corporación VPC posee licencias de lenguajes de programación y motor de Base de Datos?

¿Qué lenguajes de programación se utiliza para la creación del sistema?

¿Qué motor de base de datos se utiliza para almacenar la información?

¿Será factible desarrollar el sistema utilizando software de libre?

¿Cómo desarrollar un sistema único de Facturación e inventarios para la Corporación VPC?

¿Qué ventajas se obtendrán al implementar un Sistema de Facturación e Inventarios?

¿Cuáles son los reportes que debe emitir el sistema?

¿Proporcionará mejor organización en las tareas diarias de la empresa?

¿Se necesitará capacitación para el personal?

#### **1.3.2 Delimitación del problema**

Este presente trabajo se desarrollará en la Corporación VPC, ubicado en la ciudad de Ambato en la Av. Víctor Hugo y Aristides Sartorio esquina, atrás del Mall de los Andes.

### **1.4 Justificación**

La mayoría de las empresas están optando por adquirir sistemas de facturación e inventarios, las cuales han comenzado con un sistema manual (utilizando papel), luego han avanzado a una mezcla de papel y herramientas basadas en hojas de cálculo

computarizadas, y finalmente están adoptando un sistema computarizado simple. Las mejoras de hardware y software de computación facilitan la transición de una mezcla de papel y herramientas basadas en hojas de cálculo a la implementación de un sistema computarizado o la modernización de un sistema existente.

La decisión de automatizar su sistema de facturación e inventarios depende de su necesidad de contar con datos exactos, consistentes y oportunos en una diversidad de formatos para rendición de informes.

Debido a que la tecnología avanza cada día las empresas se ven obligadas a buscar las mejores alternativas para adaptarse en un mundo globalizado que nos obliga a tomar decisiones inmediatas en un cambio frecuente que se da en nuestro entorno actual. Es por eso que me ha llevado a realizar un estudio para el desarrollo e implantación de un sistema de Facturación e Inventarios para contribuir al adecuado funcionamiento del punto de venta en la Corporación VPC. Entonces podemos decir que esta investigación es factible.

## **1.5 Objetivos**

### **1.5.1 Objetivo General**

Desarrollar e implantar un Sistema de Facturación e Inventarios para un adecuado control tributario de compra y venta de la Corporación VPC.

### **1.5.2 Objetivos Específicos**

- Desarrollar una herramienta administrativa que sea capaz de adaptarse a las necesidades de la empresa.
- Tener un oportuno control de inventarios, obteniendo reportes a la fecha que el usuario lo solicite.
- Ofrecer a los usuarios de la empresa una herramienta que sea de apoyo en sus labores diarias.
- Apoyar al desarrollo de la empresa con la implantación del sistema.
- Implantar el sistema informático.

## **CAPÍTULO II**

### **MARCO TEÓRICO**

#### **2.1 Antecedentes Investigativos**

En la Corporación VPC para realizar el proceso de compra-venta y controlar el inventario se utiliza el popular programa Mónica, el inconveniente es que este programa no va de acuerdo con las necesidades de la empresa. No obstante una vez revisado las tesis y pasantías de la Facultad de Ingeniería en Sistemas Electrónica e Industrial de la Universidad Técnica de Ambato, se ha recolectado la siguiente información:

Los estudiantes Núñez Rodríguez Flor María y Vásquez Freire Edison Fabián realizaron el tema: SISTEMA CONTROL DE FACTURACIÓN CRÉDITOS Y BODEGA BAJO ARQUITECTURA CLIENTE SERVIDOR EN LA EMPRESA MERCANTIL GARZOZI SUCURSAL AMBATO.

Los estudiantes Guzmán Meléndez Bridida Gissela y Tabares Buebano Cristian Vinicio realizaron el tema: SISTEMA PARA EL CONTROL DE INVENTARIOS Y FACTURACIÓN EN UNA EMPRESA FARMACÉUTICA.

La estudiante Espejo Canseco Jessica Paola realizó el tema: IMPLEMENTACIÓN DE UN SISTEMA PARA LA AUTOMATIZACIÓN DEL CONTROL DE

## PROCESOS DE FACTURACIÓN BAJO UNA ARQUITECTURA CLIENTE-SERVIDOR EN LA EMPRESA GRABADOS OZALID.

El estudiante Freire Freire Hugo Leonidas realizó el tema: DESARROLLO E IMPLANTACIÓN DE UN SISTEMA PARA EL CONTROL Y FACTURACIÓN DEL AGUA POTABLE EN EL MUNICIPIO DE TISALEO.

### **2.2 Fundamentación Legal**

Considerando:

Que es necesario expedir las normas reglamentarias para la aplicación de la Ley para la Reforma de las Finanzas Públicas, publicada en el Suplemento del Registro Oficial 181 del 30 de abril de 1999; y, En ejercicio de la atribución que le confiere el número 5 del artículo 171 de la Constitución: Política de la República,

Decreta el siguiente:

### **REGLAMENTO DE FACTURACIÓN**

#### **DE LOS COMPROBANTES DE VENTA**

**Art. 1.- Comprobantes de venta.-** Para los fines de este Reglamento, se entiende por comprobante de venta todo documento que acredite la transferencia de bienes o la prestación de servicios. En consecuencia, quedan comprendidos bajo este concepto los siguientes documentos:

- a) Facturas;
- b) Notas de venta. Liquidaciones de compra de bienes o prestación de servicios,
- d) Tiquetes o vales emitidos por máquinas registradoras
- e) Los autorizados referidos en el artículo 10 de este Reglamento,
- f) Notas de crédito y notas de débito; y,

g) Otros que por su contenido y sistema de emisión, permitan un adecuado control por parte, del Servicio de Rentas Internas y se encuentren previa y expresamente autorizados por dicha Institución. El Servicio de Rentas Internas autorizará la impresión de los comprobantes de venta a través de los establecimientos gráficos autorizados, en los términos y bajo las condiciones del presente Reglamento. De igual forma, el Servicio de Rentas Internas podrá limitar o restringir la impresión de los comprobantes de venta, en función del grado. De cumplimiento de las obligaciones del contribuyente que haya solicitado su impresión.

**Art. 2.- Obligación de emisión de comprobantes de venta.-** Están obligados a emitir comprobantes de venta todos los sujetos pasivos de los impuestos a la renta, al valor agregado o a los consumos especiales sean sociedades o personas naturales, incluyendo las sucesiones indivisas obligados o no a llevar contabilidad, en los términos establecidos por la Ley de Régimen Tributario Interno.

**Art. 3.- Sustento del crédito tributario.-** Para ejercer el derecho al crédito tributario en el caso del impuesto al valor agregado, sólo se considerarán válidos las facturas, las liquidaciones de compras de bienes o adquisición de servicios, los pasajes expedidos por las empresas de aviación por el servicio de transporte aéreo de personas y los comprobantes emitidos por las empresas de seguros y por las empresas de telecomunicaciones.

**Art. 4.- Sustento de costos o gastos.-** Para sustentar costos o gastos a efectos de la determinación y liquidación del impuesto a la renta, sólo se considerarán como comprobantes válidos los determinados en los artículos 1 y 3, siempre que cumplan con todos los requisitos establecidos en este Reglamento.

**Art. 5.- Oportunidad para la utilización y emisión de comprobantes de venta.-** Los comprobantes de venta enumerados en el artículo 1, se utilizarán o emitirán conforme las reglas que constan en los artículos 6 al 12.

**Art. 6.- Facturas.-** Se emitirán facturas en, los siguientes casos:

a) Cuando las operaciones se realicen para transferir bienes o prestar servicios a sociedades o personas naturales que tengan derecho al uso de crédito tributario; y,

b) En operaciones de exportación.

**Art. 7.- Notas o boletas de venta.-** Se emitirán notas o boletas de venta únicamente en operaciones con consumidores o usuarios finales.

**Art. 8.- Liquidaciones de compras de bienes o prestación de servicios.-** Las liquidaciones de compras de bienes o prestación de servicios las emitirá el adquirente en los siguientes casos:

a) Cuando se trate de adquisiciones de bienes o de servicios a personas naturales no obligadas a emitir comprobantes de venta.

b) La liquidación de compras considerará, de ser el caso, el impuesto al valor agregado que será retenido y pagado por el comprobante; y,

c) En todas aquellas circunstancias en que el Servicio de Rentas Internas considere necesario autorizar a los adquirentes.

**Art. 9.- Tiquetes o vales emitidos por máquinas registradoras.-** Se emitirán tiquetes o vales emitidos por máquinas registradoras en operaciones con consumidores o usuarios finales.

**Art. 10.- Documentos autorizados.-** Son documentos autorizados, siempre que identifique al adquirente o usuario mediante su número de cédula o de Registro Único de Contribuyentes, nombres o razón social, y se discrimine el impuesto al valor agregado, los siguientes:

a) Los boletos que expidan las compañías de aviación por el servicio de transporte aéreo de personas, y,

b) Los documentos emitidos por compañías de seguros que se encuentran bajo control de la Superintendencia de Bancos y por las empresas de telecomunicaciones.

**Art. 11.- Notas de crédito.-** Las Notas de Crédito se emitirán por modificación en las condiciones de venta originalmente pactadas, es decir, para anular operaciones efectuar devoluciones, conceder descuentos y bonificaciones. Subsanan errores o casos similares deberá contener los mismos requisitos y características de los comprobantes de venta a los cuales se refieran.

**Art. 12.- Notas de débito.-** Las Notas de débito, se emitirán para recuperar costos o gastos, tales con los intereses de mora u otros, incurridos por el vendedor con posterioridad a la emisión de comprobantes de venta.

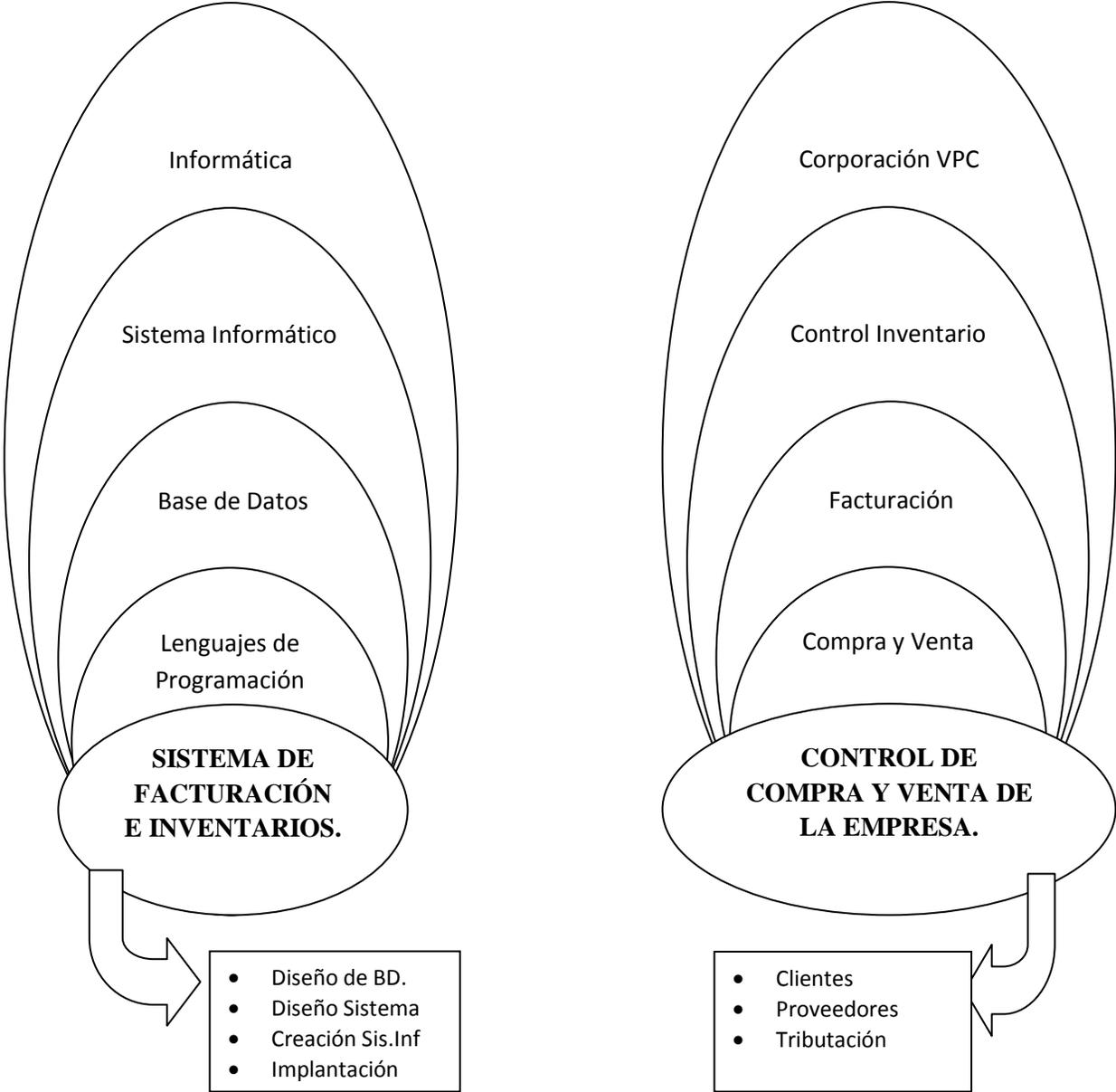
**Art. 13.- Requisitos para las notas de crédito y de débito.-** Las notas de crédito y las notas de débito deben consignar la serie y número de los comprobantes de venta que modifican.

**Art. 14.- Oportunidad de entrega de los comprobantes de venta.-** Los comprobantes de venta enumerados por el artículo 1 de este Reglamento, deberán ser entregadas en siguientes oportunidades:

- a) En la transferencia de bienes muebles en el momento en que se realiza el acto o se suscribe el contrato que tenga por objeto transferir el dominio de los mismos.
- b) Cuando el giro del negocio sea la transferencia de bienes inmuebles, en la fecha en que se perciba el ingreso o en la fecha en que se celebre el contrato lo que ocurra primero, y,
- c) En la prestación servicios, el arrendamiento mercantil, en el momento en que se realiza el acto o se suscribe el contrato que tenga por objeto la prestación de servicios o el arrendamiento.

Reglamentos de Facturación (1999, Junio 10). Recuperado el 14 de marzo de 2012, de [http://www.derechoecuador.com/index.php?option=com\\_content&task=view&id=4095](http://www.derechoecuador.com/index.php?option=com_content&task=view&id=4095)

### 2.3 Categorías Fundamentales



**Figura 1.1** Inclusión Interrelacionado de las Variables Independiente y Dependiente.

### **2.3.1 Informática**

La informática es el procesamiento automático de información mediante dispositivos electrónicos y sistemas computacionales. Los sistemas informáticos deben contar con la capacidad de cumplir tres tareas básicas: entrada (captación de la información), procesamiento y salida (transmisión de los resultados). El conjunto de estas tres tareas se conoce como algoritmo.

La informática abarca también los principales fundamentos de las ciencias de la computación, como la programación para el desarrollo de software, la arquitectura de las computadoras y del hardware, las redes como Internet y la inteligencia artificial.

Definición de Informática. (s.f) Recuperado el 14 de marzo de 2012, de <http://definicion.de/informatica/>

### **2.3.2 Sistema Informático**

Es el conjunto de partes interrelacionadas, una computadora que usa dispositivos programables para capturar, almacenar y procesar datos. La computadora personal junto con la persona que lo maneja y los periféricos que los envuelven, resultan de por sí un ejemplo de un sistema informático.

Incluso la computadora más sencilla se clasifica como un sistema informático, porque al menos dos componentes (hardware y software) tienen que trabajar unidos. Pero el genuino significado de "sistema informático" viene mediante la interconexión. Muchos sistemas informáticos pueden interconectarse, esto es, unirse para convertirse un sistema mayor. La interconexión de sistemas informáticos puede tornarse difícil debido a incompatibilidades. A veces estas dificultades ocurren a nivel de hardware, mientras que en otras ocasiones se dan entre programas informáticos que no son compatibles entre sí.

Sistema Informático. (s.f) Recuperado el 14 de marzo de 2012, de [http://es.wikipedia.org/wiki/Sistema\\_inform%C3%A1tico](http://es.wikipedia.org/wiki/Sistema_inform%C3%A1tico)

### **2.3.3 Base de Datos**

Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En la actualidad, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital (electrónico), que ofrece un amplio rango de soluciones al problema de almacenar datos.

Existen programas denominados sistemas gestores de bases de datos, abreviados SGBD (Sistema de Gestión de Bases de Datos), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas. También son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental. Entre las bases de datos que son libres tenemos a Postgres y MySQL, y entre las propietarias podemos mencionar a Oracle y SqlServer.

Base de datos. (s.f.) Recuperado el 14 de marzo de 2012, de [http://es.wikipedia.org/wiki/Base\\_de\\_datos](http://es.wikipedia.org/wiki/Base_de_datos)

#### **2.3.3.1 Postgres**

Es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD (Berkeley Software Distribution).

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una sola empresa sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

PostgreSQL. (s.f.) Recuperado el 14 de marzo de 2012, de <http://es.wikipedia.org/wiki/PostgreSQL>

### **2.3.3.2 MySql**

Es un sistema de gestión de base de datos relacional, multihilo y multiusuario. Ofrece bajo la GNU GPL (General Public License) para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C (American National Standards Institute).

MySQL. (s.f.) Recuperado el 14 de marzo de 2012, de [http://es.wikipedia.org/wiki/MySQL#Caracter.C3.ADsticas\\_de\\_la\\_versi.C3.B3n\\_5.0](http://es.wikipedia.org/wiki/MySQL#Caracter.C3.ADsticas_de_la_versi.C3.B3n_5.0).  
22

### **2.3.3.3 Oracle**

Es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), desarrollado por Oracle Corporation.

Se considera como uno de los sistemas de bases de datos más completos, destacando:

- Soporte de transacciones,
- Estabilidad,
- Escalabilidad y
- Soporte multiplataforma.

Oracle Database. (s.f.) Recuperado el 14 de marzo de 2012, de <http://es.wikipedia.org/wiki/Oracle>

### **2.3.3.4 SqlServer**

Microsoft SQL Server es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL.

Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle, Sybase ASE, PostgreSQL, Interbase, Firebird o MySQL.

Microsoft SQL Server. (s.f.) Recuperado el 14 de marzo de 2012, de [http://es.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](http://es.wikipedia.org/wiki/Microsoft_SQL_Server)

### **2.3.4 Lenguajes de Programación**

Un lenguaje de programación es aquel elemento dentro de la informática que nos permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis que pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes.

Un lenguaje es un sistema de comunicación que tiene forma, contenido y uso. La programación es el proceso de escritura del código fuente de un software. De esta forma, la programación le señala al programa informático qué tiene que hacer y cómo realizarlo.

Definición de lenguaje de programación. (s.f.) Recuperado el 14 de marzo de 2012, de <http://definicion.de/lenguaje-de-programacion/>

#### **2.3.4.1 PHP**

Lenguaje de programación usado generalmente en la creación de contenidos para sitios web. Generalmente los scripts en PHP se embeben en otros códigos como HTML, ampliando las posibilidades del diseñador de páginas web enormemente.

La interpretación y ejecución de los scripts PHP se hacen en el servidor, el cliente (un navegador que pide una página web) sólo recibe el resultado de la ejecución y jamás ve el código PHP.

#### **2.3.4.2 C#**

Es un lenguaje de propósito general orientado a objetos creado por Microsoft para su plataforma .NET. Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET el cual es similar al de Java aunque incluye mejoras derivadas de otros lenguajes. C# fue diseñado para combinar el control a bajo nivel de lenguajes como C y la velocidad de programación de lenguajes como Visual Basic.

#### **2.3.4.3 JavaScript**

Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado.

#### **2.3.4.4 ASP.NET**

Es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios web XML. ASP.NET está construido sobre el Common Language Runtime, permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework.

Definición De Lenguaje De Programación. (s.f.)Recuperado el 14 de marzo de 2012, de <http://www.mitecnologico.com/Main/DefinicionDeLenguajeDeProgramacion>

### **2.3.5 Sistema de Facturación e Inventarios**

#### **2.3.5.1 Diseño de Base de Datos**

Una base de datos correctamente diseñada permite obtener acceso a información exacta y actualizada. Puesto que un diseño correcto es esencial para lograr los objetivos fijados para la base de datos, parece lógico emplear el tiempo que sea necesario en aprender los principios de un buen diseño.

El proceso de diseño de una base de datos se guía por algunos principios. El primero de ellos es que se debe evitar la información duplicada porque malgastan el espacio y aumentan la probabilidad de que se produzcan errores e incoherencias. El segundo principio es que es importante que la información sea correcta y completa.

#### **2.3.5.2 Diseño del Sistema**

El Diseño de Sistemas se define el proceso de aplicar ciertas técnicas y principios con el propósito de definir un dispositivo, un proceso o un Sistema, con suficientes detalles como para permitir su interpretación y realización física.

#### **2.3.5.3 Creación del sistema Informático**

La creación del Sistema Informático sería que la base de datos y sistema trabaje de manera coordinada, almacenando y visualizando datos eficazmente. Un sistema informático puede ser tan simple que conste de un solo programa o tan complejo que pueda comprender cientos. Puede funcionar con una sola computadora o con sistema de maquinas gigantes.

Fernández, R. (2009, abril 8). Sistemas informáticos. Elementos y esquema. Recuperado el 14 de marzo de 2012, de <http://www.mailxmail.com/curso-sistemas-informativos-diseno-creacion-1/sistemas-informaticos-elementos-esquema>

#### **2.3.5.4 Implantación**

En la fase de implantación, las especificaciones del diseño del sistema sirven como base para la construcción del nuevo sistema. En este punto, los programadores y los analistas de sistemas asumen diferentes responsabilidades. El analista debe proveer especificaciones claras y correctas al programador. El programador codifica, prueba y documenta los módulos de programas, mientras que el analista de sistema planifica la integración de los programas y asegura que trabajen unidos para satisfacer las necesidades de la organización. Un nuevo sistema requiere planificación, construcción y prueba. Los programas y módulos deben ser diseñados, codificados, probados y documentados.

### **2.3.6 Corporación VPC (Ver Paint Corp)**

Esta corporación se dedica principalmente a la importación, venta y comercialización de pinturas y complementarios de pintura PPG (Pittsburgh Plate Glass). Son los únicos distribuidores de PPG para todo el Ecuador, siendo la número uno en acabado automotriz y líder a nivel mundial.

Los objetivos de esta Corporación a parte de brindar al mercado una excelente pintura es ofrece el mejor servicio (con capacitación continua, servicio técnico y personalizado) y ser líderes en el mercado automotriz a nivel nacional.

### **2.3.7 Control de Inventarios**

La importancia en el control de inventarios reside en el objetivo primordial de toda empresa: obtener utilidades. La obtención de utilidades obviamente reside en gran parte de Ventas, ya que éste es el motor de la empresa, sin embargo, si la función del inventario no opera con efectividad, ventas no tendrá material suficiente para poder trabajar, el cliente se inconforma y la oportunidad de tener utilidades se disuelve. Entonces, sin inventarios, simplemente no hay ventas.

Aguilar, M. (2005, abril 20). Cómo controlar tu inventario. Recuperado el 14 de marzo de 2012, de <http://www.mailxmail.com/curso-como-controlar-inventario/importancia-control-inventarios>

### **2.3.8 Facturación**

La factura comercial es un documento administrativo que refleja toda la información de una operación de compraventa. La información fundamental que aparece en una factura debe reflejar la entrega de un producto o la provisión de un servicio, junto a la fecha de devengo, además de indicar la cantidad a pagar como contraprestación.

Además, en la factura deben aparecer los datos del expedidor y del destinatario, el detalle de los productos y servicios suministrados, los precios unitarios, los precios totales, los descuentos y los impuestos.

Factura. (s.f.) Recuperado el 14 de marzo de 2012, de <http://es.wikipedia.org/wiki/Factura>

### **2.3.9 Compra y Venta**

Según el código civil la compra venta es el contrato mediante el cual el vendedor se obliga a transferir la propiedad de un bien al comprador, y este a su vez, se obliga a pagar su precio en dinero.

Alarcón, F. Compra Venta. Recuperado el 14 de marzo de 2012, de <http://www.monografias.com/trabajos16/compra-venta/compra-venta.shtml>

### **2.3.10 Control tributario de compra y venta de la empresa**

#### **2.3.10.1 Clientes**

Cliente es la persona, empresa u organización que adquiere o compra de forma voluntaria productos o servicios que necesita o desea para sí mismo, para otra persona o para una empresa u organización; por lo cual, es el motivo principal por el que se crean, producen, fabrican y comercializan productos y servicios.

Thompson, I. (2009, Julio). Definición de Cliente. Recuperado el 14 de marzo de 2012, de <http://www.promonegocios.net/clientes/cliente-definicion.html>

#### **2.3.10.2 Proveedores**

Son proveedores aquellas personas físicas o jurídicas que surten a la empresa de existencias (mercaderías, materias primas, envases, etc.), que posteriormente ésta venderá, transformará o elaborará.

#### **2.3.10.3 Tributación**

Se refiere al conjunto de obligaciones que deben realizar los ciudadanos sobre sus rentas, sus propiedades, mercancías, o servicios que prestan, en beneficio del Estado, para su sostenimiento y el suministro de servicios, tales como defensa, transportes, comunicaciones, educación, sanidad, vivienda, entre otros.

## **2.4 Hipótesis**

El correcto desarrollo e implantación de un Sistema de Facturación e Inventarios mejorará el control tributario de compra y venta en la Corporación VPC.

## **2.5 Señalamiento de Variables de la Hipótesis**

### **2.5.1 Variable Independiente**

Sistema de Facturación e Inventarios

### **2.5.2 Variable Dependiente**

Control Tributario de Compra y Venta.

## **CAPÍTULO III**

### **METODOLOGÍA**

#### **3.1 Enfoque**

La investigación se encuentra dentro de un enfoque cuali - cuantitativo porque los objetivos y el proceso de investigación a seguirse serán conocidos solo por los investigadores y las decisiones para actuar serán tomadas por los técnicos.

#### **3.2 Modalidad básica de la investigación**

##### **3.2.1 Investigación de campo**

Esta investigación permite el estudio sistemático de los hechos en el lugar que se producen los acontecimientos. En esta modalidad el investigador toma contacto en forma directa con la realidad, para tener información de acuerdo con los objetivos del proyecto.

##### **3.2.2 Investigación Bibliográfica – Documental**

Esta modalidad permite conocer, comparar, ampliar, profundizar y deducir diferentes enfoques, teorías, conceptualizaciones y criterios de diversos autores sobre una cuestión determinada basándose en internet(fuentes primarias) o en libros, revistas, periódicos y otras publicaciones(fuentes secundarias).

### **3.2.3 Proyecto Factible**

La siguiente investigación es un proyecto factible porque se elabora una propuesta en base al desarrollo de un sistema automatizado para un mejor control tributario de compra y venta, que permitirá satisfacer las necesidades de la Corporación VPC detectadas en función de un diagnóstico y sustentados con base teórica.

### **3.3 Nivel o tipo de investigación**

La investigación utilizará el nivel exploratorio para detectar las características del problema, determinar si es factible o no de solucionarse, pasaremos al nivel descriptivo para conocer con mayor profundidad las circunstancias en las que se produce el problema y la realidad en la que se desarrolla; el nivel operacional facilita la comprensión, el estudio, el análisis del fenómeno dentro de un contexto determinado para tener un conocimiento sobre el proyecto a investigar.

### **3.4 Población y muestra**

#### **3.4.1 Población**

La población está compuesta por 10 personas

#### **3.4.2 Muestra**

Como la población es pequeña pasa a constituir la muestra.

### **3.5 Operacionalización de las variables**

Para recolección de información se utilizará la técnica de entrevistas al personal que trabaja en el punto de ventas de la empresa. Esta técnica es importante por cuanto nos ha permitido conocer en forma directa la situación actual sobre el control tributario de compra y venta que se utiliza en la Corporación VPC.

Variable Independiente				
Conceptualización	Dimensión	Indicadores	Ítems	Tec-Inst
<b>Sistema de Facturación e Inventarios:</b> Son documentos electrónicos que en cuanto a las facturas detalla los artículos vendidos, así como los precios, los datos del cliente y los términos de venta. En cuanto al inventario detalla los productos que se tienen en existencia, aún no vendidos, en un momento determinado. Teniendo tras de esto un lenguaje de programación en la que se realizará el sistema en sí y un motor de base de datos que almacenara toda la información necesaria.	Lenguajes de Programación	PHP, C#, JavaScript, Visual Web Developer	¿El sistema se desarrollará en un lenguaje libre o propietario?	Entrevista
	Base de Datos	Postgres, MySql, Oracle, SqlServer	¿Cuál es la ventaja de utilizar software libre o propietario?	Entrevista
			¿El sistema se desarrollara con un motor de base de datos libre o propietario?	Entrevista
	Sistema Informático	Sistema Automatizado	¿Cuál es la ventaja de utilizar un motor de base de datos libre o propietario?	Entrevista
			¿Es necesaria la implantación de un sistema informático en la empresa?	Entrevista
	Sistema	Sistema de Facturación e Inventarios	¿Cómo influye un sistema automatizado en la empresa?	Entrevista
			¿Cómo maneja la información de la empresa actualmente?	Entrevista
	¿Cómo se manejara la información de la empresa con la implantación de un nuevo sistema?	Entrevista		

**Tabla 3.1** Operacionalización de variables – Variable independiente

Variable Dependiente				
Conceptualización	Dimensión	Indicadores	Ítems	Tec-Inst
<b>Control tributario de compra y venta:</b> Se refiere al conjunto de obligaciones que deben realizar los ciudadanos sobre sus mercancías, en beneficio del Estado, para su sostenimiento y el suministro de servicios, tales como defensa, transportes, comunicaciones, educación, sanidad, vivienda, entre otros.	Compra y Venta	Productos	¿Tienen algún inconveniente en la venta cuando se realiza el proceso de mezcla de colores?	Encuesta
	Control de Inventarios	Almacén de Bodega	¿En el proceso de facturación se realiza descuentos?	Encuesta
			¿El sistema guarda la secuencia de cada factura proporcionada por el SRI?	Encuesta
	Facturación	Facturación Automatizada	¿Se tiene un óptimo control de los productos que son vendidos a crédito?	Encuesta
			¿Tiene un adecuado control sobre inventarios?	Encuesta
			¿Pueden disponer del inventario en tiempo real?	Encuesta
	Control de los estados financieros de la empresa	Administración adecuada	¿Se puede obtener el estado del inventario de una fecha anterior?	Encuesta
			¿Tienen un óptimo control de compras?	Encuesta
			¿Tienen un adecuado control de los clientes?	Encuesta
				¿Los usuarios tienen acceso total sobre el sistema?

**Tabla 3.2** Operacionalización de variables – Variable dependiente

### **3.6. Recolección de la Información**

#### **3.6.1 Plan de recolección de información**

Recolección en fuentes bibliográficas, datos proporcionados por el personal que labora en el punto de ventas de la empresa quienes están encargados de realizar la facturación a los diferentes clientes.

#### **3.6.2 Procesamiento y Análisis de la Información**

Para analizar y procesar la información de la presente investigación se reunirá, presentará y resumirá todos los datos que han sido recogidos mediante la técnica señalada para el efecto. Luego se procederá a la clasificación de los datos recopilados, se tabularán esto en forma manual por ser un método simple, rápido y efectivo, siempre y cuando se mantenga el debido cuidado para no alterar la información.

### **3.7 Análisis e interpretación de resultados**

Los resultados se presentarán en forma gráfica de los datos obtenidos mediante una entrevista aplicada al personal que trabaja en el punto de ventas, se analizará e interpretará; seguidamente se elaborará una síntesis de los resultados obtenidos para dar una explicación al problema sujeto a investigación, por último se elaborará una síntesis general para la elaboración de conclusiones.

## **CAPÍTULO IV**

### **ANÁLISIS E INTERPRETACIÓN DE RESULTADOS**

#### **4.1 Análisis de la necesidad**

La gerencia de la Corporación VPC se encargó de realizar la propuesta para que por medio de la Universidad Técnica de Ambato los estudiantes se vinculen mediante proyectos que tienen en la Corporación, entre ellos la necesidad de un sistema de facturación e inventarios, para lo cual se solicitó que un estudiante de la carrera de Sistemas se encargue del proyecto.

Una vez aprobado el proyecto por el gerente de la Corporación VPC, se puso un acuerdo en la cual dicho proyecto servirá como trabajo de graduación del estudiante encargado.

En vista que la Corporación VPC no cuenta con un sistema que este acorde a sus necesidades, requiere del desarrollo e implantación de un sistema de facturación e inventarios que corra bajo un navegador web permitiendo una mejor atención en el punto de venta.

Hoy en día ya no solo se realizan páginas web informativas, ahora sistemas complejos están desarrollados en ambientes web, la cuál sin la necesidad de ser instalada se puede correr la aplicación desde cualquier computador solo con la utilización de un navegador, es por tal razón que se optó por desarrollo web.

## 4.2 Análisis de los resultados

### Entrevista:

La entrevista se lo realizó al gerente de la Corporación VPC quién supo manifestar que la Corporación es un Empresa dedicada a la comercialización de pintura para vehículos, casas y otros acabados, en donde con un proceso de mezcla de colores pueden obtener cualquier tipo de color (color generado), este proceso de mezcla lo realizan con un sistema y para realizar la facturación se basan de otro sistema, en fin para realizar el proceso de facturación de la venta de colores se basan de la utilización de dos sistemas por lo que se busca de un sistema que realice los dos procesos a la vez, tanto el proceso de mezcla de colores como el proceso de facturación sobre el total generado de dicha mezcla.

El gerente tiene conocimientos de software libre por lo que supo manifestar que seria bueno que el sistema en cuestión sea desarrollado con herramientas de software libre, de todas maneras se le a dado una breve explicación sobre las ventajas y desventajas que se tiene al utilizar herramientas de software libre y propietarias. En lo referente a las bases de datos se dio una explicación sobre:

- SqlServer
- Oracle
- PostgreSQL
- My Sql

Que son las bases más populares y conocidas en nuestro medio, a la vez con una gran capacidad de almacenamiento de información y recuperación de la misma de forma rápida y eficaz. De la misma manera se dio una explicación sobre los lenguajes de programación mas conocidos en nuestro medio, como son:

- Php
- C#
- Java

- Visual Basic
- C++

Luego de la explicación finalmente optó por la utilización de software libre tanto para el motor de base de datos como para la herramienta de desarrollo de software, por lo que junto con el señor Gerente se decidió utilizar Postgres como la base de datos y C# (Incluida en el paquete de Visual Studio Express Edition) como lenguaje de programación, la idea principal de utilizar software libre fueron los costos de licencia.

Adicionalmente, se le propuso a la corporación realizar el sistema mediante formularios Windows pero se manifestó diciendo que seria mejor una aplicación web, porque le facilitaría el ingreso al sistema solo con la utilización de un navegador desde cualquier computador sin la necesidad de estar instalado.

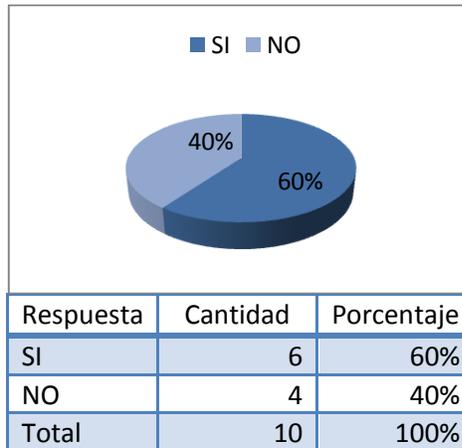
En lo referente al inventario actualmente realizan este proceso de manera manual cada mes mediante constataciones físicas, este proceso es incomodo para la corporación porque no se tiene los datos a tiempo real y se debe asignar a un personal pudiendo este haber ocupado ese tiempo en otras actividades.

La idea con el desarrollo del nuevo sistema es tener el stock de cada producto en tiempo real, es decir que cada producto vaya disminuyendo o aumentando de acuerdo con las ventas y compras realizadas diariamente, entonces al momento de consultar el stock poder conocer el estado si el producto esta por agotarse para realizar el proceso respectivo de reposición.

La Encuesta realizada al personal institucional de la Corporación VPC. Cabe indicar que se han seleccionado para el análisis las preguntas más relevantes; las preguntas tabuladas son las siguientes:

**Encuesta:**

**1. Tienen algún inconveniente en la venta cuando se realiza el proceso de mezcla de colores?**

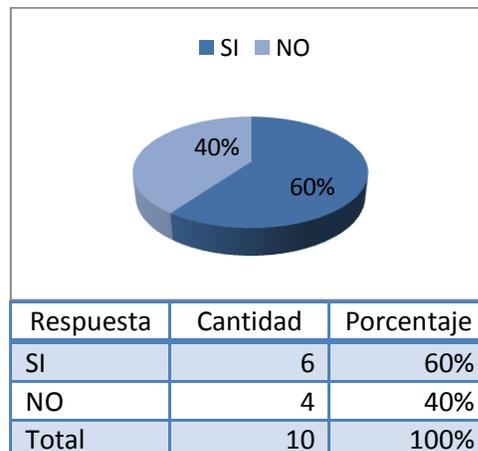


**Tabla 4.1** Cuadro porcentual pregunta 1

SI, porque el mayor inconveniente es que una vez realizada la mezcla de color toca acudir a un segundo sistema para realizar la factura. NO, porque una vez mezclado se obtiene el color deseado.

Observación: Se verificó que efectivamente se utiliza dos sistemas para poder realizar el proceso de facturación.

**2. En el proceso de facturación se realiza descuentos?**

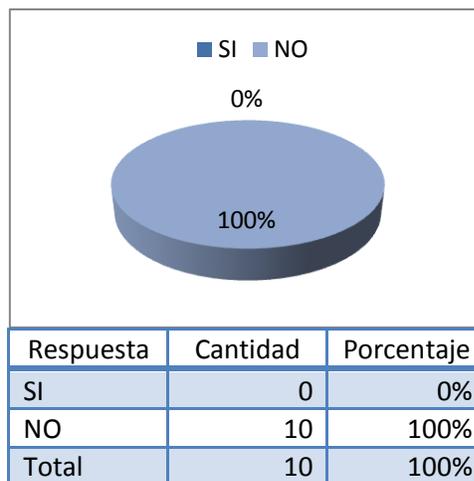


**Tabla 4.2** Cuadro porcentual pregunta 2

SI, cuando se realizan promociones en ciertos productos. NO, porque no en todos los productos se aplican descuentos.

Observación: Se pudo comprobar que realizan descuentos en determinadas facturas, el inconveniente es que no se puede aplicar dos descuentos, una por cada producto y otra por el total de la factura.

### 3. El sistema guarda la secuencia de cada factura proporcionada por el SRI?

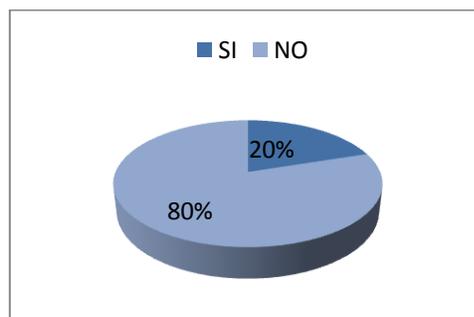


**Tabla 4.3** Cuadro porcentual pregunta 3

NO, porque el sistema no lo permite.

Observación: Se pudo verificar que el sistema no está guardando ningún dato de la factura proporcionado por el SRI.

### 4. Se tiene un óptimo control de los productos que son vendidos a crédito?



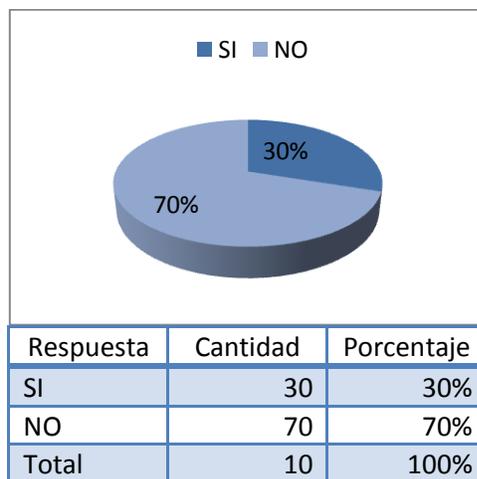
Respuesta	Cantidad	Porcentaje
SI	20	20%
NO	80	80%
Total	10	100%

**Tabla 4.4** Cuadro porcentual pregunta 4

SI, porque vamos registrando manualmente en hojas de Excel. NO, porque el sistema no guarda una tabla donde nos indique cuando el cliente tiene que acercarse a cancelar su crédito.

Observación: Se pudo verificar que no se tiene un óptimo control sobre los productos que son vendidos a crédito.

#### 5. Tiene un adecuado control sobre inventarios?

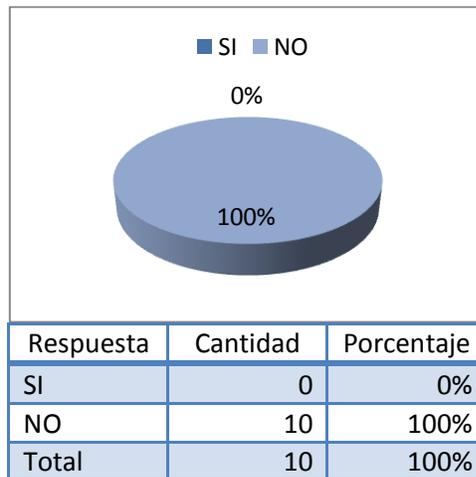


**Tabla 4.5** Cuadro porcentual pregunta 5

SI, porque lo realizamos mediante verificaciones físicas. NO, porque es un pasar de tiempo que cada vez que se necesite el inventario una persona tiene que estar haciendo verificaciones en la bodega.

Observación: Se pudo verificar que el inventario lo realizan de manera manual mensual cuando lo óptimo sería que el sistema vaya actualizando el stock de cada producto y una contatación física como práctica sana anual.

## 6. Pueden disponer del inventario en tiempo real?

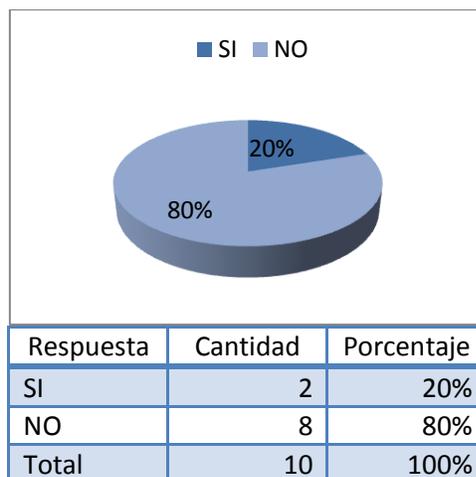


**Tabla 4.6** Cuadro porcentual pregunta 6

NO, porque el inventario se lo realiza de forma manual.

Observación: Se pudo verificar que el sistema no guarda el inventario diario por lo cual no puede recuperar el estado del inventario actual.

## 7. Se puede obtener el estado del inventario de una fecha anterior?

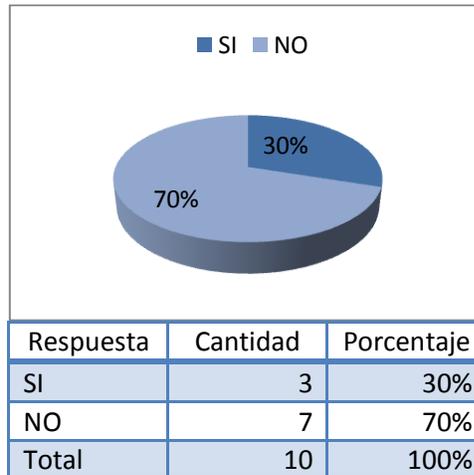


**Tabla 4.7** Cuadro porcentual pregunta 7

SI, porque los inventarios manuales que se realizan se archivan. NO, porque el inventario se lo realiza cada mes por lo que no se puede saber el estado que estaba en un día específico.

Observación: Se pudo verificar el inventario manual no se realiza todos los días por lo que es imposible saber el estado de cualquier día anterior a la fecha actual.

### 8. Tienen un óptimo control de compras?

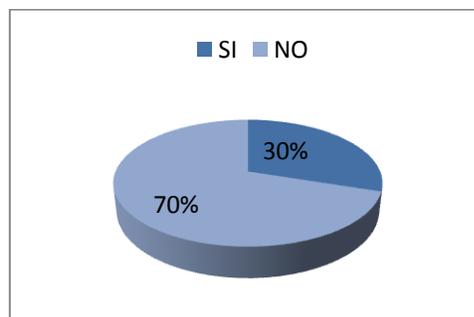


**Tabla 4.8** Cuadro porcentual pregunta 8

SI, porque se lo va ingresando en el sistema. NO, porque a pesar que el sistema lo registra no almacena todos los datos necesarios como descuentos y otros cargos si son necesarios.

Observación: Se pudo verificar el proceso de compras el cual no se adapta con los requerimientos de la corporación, como IVA, descuentos y otros cargos.

### 9. Tienen un adecuado control de los clientes?



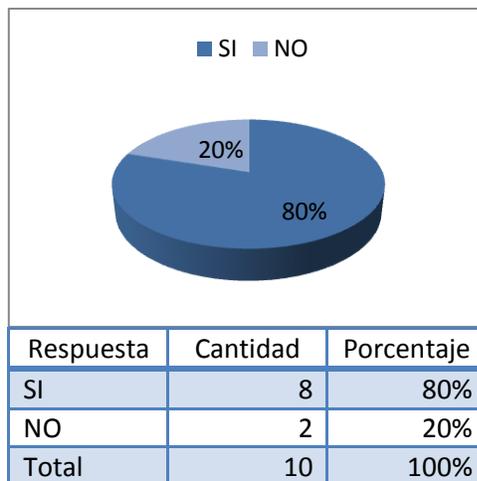
Respuesta	Cantidad	Porcentaje
SI	3	30%
NO	7	70%
Total	10	100%

**Tabla 4.9** Cuadro porcentual pregunta 9

SI, porque el control lo llevamos desde el sistema. NO, porque existen datos que el sistema no los registra (como varios números telefónicos, un vendedor asociado al cliente, formas de pago, etc).

Observación: Se pudo verificar que los datos de los clientes guardados en el sistema no todos son necesarios y que existen otros datos que si son necesarios pero no son guardados.

**10. Los usuarios tienen acceso total sobre el sistema?**



**Tabla 4.10** Cuadro porcentual pregunta 10

SI, porque el sistema no tiene un control sobre quien puede acceder. NO, porque al sistema solo ingresan usuarios autorizados.

Observación: Se pudo observar que como el sistema no tiene un control sobre usuarios entonces no todos acceden al sistema, pero deberían poder realizar consultas.

### **4.3 Interpretación de resultados**

La corporación actualmente para la comercialización de pintura se basa de dos sistemas, el uno realiza el proceso de combinación de colores generando un nuevo color, mientras que el otro sistema se encarga de realizar la facturación sobre el total del color generado, es por tal razón que la corporación requiere de un sistema que realice los dos procesos a la vez.

También se pudo determinar que el gerente a optado que el sistema sea desarrollado mediante herramientas de software libre ya que la principal ventaja de esta son los costos de licencia a la vez que son herramientas de igual capacidad de las herramientas de software propietario, entonces una vez analizado las ventajas y desventajas de las herramientas de software libre se decidió que para el desarrollo del sistema se ocupara PostgreSQL como motor de base de datos y C# como lenguaje de programación la cual esta diseñada para generar programas sobre la plataforma .NET por lo que se utilizará Microsoft Visual Studio Express.

El sistema debe ser desarrollado bajo un ambiente web, con el objetivo que la aplicación pueda arrancar desde cualquier computador conectado a la red, sin la necesidad que este instalado, para esto se utilizara Visual Web Developer que es una herramienta para diseñar aplicaciones web y que forma parte Microsoft Visual Studio Express.

También se pudo verificar que el control del inventario no es el óptimo porque no se puede obtener el stock de un producto en tiempo real, con el sistema en cuestión se plantea poder obtener el stock exacto de cada producto en tiempo real cada vez que fuese necesario, para esto el sistema deberá aumentar o disminuir el stock de acuerdo a los movimientos que se va realizado.

## **CAPÍTULO V**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **5.1 Conclusiones**

- La corporación requiere de un sistema para realizar procesos de facturación e inventarios, de esta manera se estaría optimizando tiempo y se daría un mejor servicio a los usuario.
- El proceso de comercialización de pintura actualmente se lo está realizando con la utilización de dos sistemas, este proceso se deberá realizar con un mismo sistema.
- Las herramientas para el desarrollo de la aplicación deberán ser de software Open Source y en un ambiente web.
- Se deberá obtener el inventario real de cada producto cada vez que el usuario del sistema lo creyere conveniente.
- El sistema con el ingreso de parámetros debe ser capaz de poder realizar la mezcla de colores, obteniendo un nuevo color con el total de la cual se deberá realizar la factura con diferentes formas de pago.
- Mediante el desarrollo e implementación del sistema informático que trabaje con una base de datos, la información tiene mayor integridad y seguridad, los datos son más confiables y aportan con mayor eficiencia para recuperar la información.

De la encuesta realizada al personal de la corporación se pudo concluir que:

1. Al realizar el proceso de combinación de colores el inconveniente es que luego de generar la mezcla tienen que realizar la facturación en otro sistema.
2. Actualmente se realiza facturas en las cuales se hace un descuento por el total, el inconveniente es que se no pueden aplicar dos descuentos de ser necesario, un descuento por cada producto y otro por el total.
3. El sistema actual no guarda los datos de la factura proporcionados por el SRI como son el secuencial y la autorización.
4. No se tiene un adecuado control sobre los productos que son vendidos a crédito ya que el sistema actual no guarda información para luego ser recuperado y le indique al usuario cuando un cliente tiene que acercarse a cancelar su crédito.
5. No se tiene un adecuado control del inventario ya que este se lo realiza de una forma manual la cual es muy tedioso y lleva mucho tiempo.
6. No se puede disponer del inventario en tiempo real ya que el sistema actual no lo almacena y si lo necesitan tienen que realizarlo de manera manual.
7. Por motivo que el sistema actual no tiene un control sobre el inventario diario no se puede recuperar el estado en la que estaba en una fecha determinada.
8. El control de compras de productos no es el óptimo ya que existen compras de las cuales es necesario guardar datos adicionales (como descuento y otros cargos).
9. El control de clientes no es el óptimo ya que el sistema actual registra datos innecesarios (como dirección uno dirección dos) y no registra otros datos que son necesarios para un mejor control (como varios números telefónicos, un vendedor asociado al cliente, formas de pago)
10. El control de acceso de usuarios al sistema no es el adecuado ya que no se cuenta con un módulo de seguridades donde cada usuario tenga acceso al sistema de acuerdo a lo que necesite realizar.

## 5.2 Recomendaciones

- Diseñar correctamente la estructura de la base de datos para garantizar la información que será almacenada en la misma.
- Diseñar la interfaz de la aplicación con la utilización de estilos en cascada CSS, obteniendo una interfaz amigable para el usuario.
- Desarrollar la aplicación utilizando Visual Studio Express y JavaScript para la validación de los datos, garantizando de esta manera que los datos ingresados a la base sean los correctos.
- Realizar las pruebas necesarias para garantizar el correcto funcionamiento de la aplicación.
- Realizar los respectivos manuales para la fácil manipulación de la aplicación por parte de los usuarios.
- Finalmente se recomienda implementar políticas de uso del sistema para su correcto y eficaz funcionamiento.

Además de la encuesta realizada se recomienda:

1. Realizar un sistema que unifique el proceso de mezcla de colores y a la vez realice el proceso de facturación.
2. Que el módulo de facturación debe tener la opción de aplicar dos descuentos una en cada producto ingresada y otra sobre el total de la factura.
3. Que en el proceso de facturación se almacene los datos de cada factura como son la autorización y el secuencial, que el secuencial sea de manera automática (parametrizable) y manual.
4. Realizar un módulo de créditos que al momento de realizar una factura de la opción de pago a crédito en donde se ingrese parámetros como monto, el tiempo y un recargo adicional si fuese necesario, en la que el sistema automáticamente guarde una modalidad de rubros por fechas para que cliente vaya cancelando de acuerdo a una tabla de pagos generado.

5. Realizar un módulo de inventarios con el objetivo de que ya no se realice de forma manual y que se pueda consultar el estado cada vez que sea necesario en cuestión de segundos.
6. Que el sistema debe ir aumentando y disminuyendo el stock automáticamente de cada producto de acuerdo a las ventas y compras realizadas con el objetivo de obtener un inventario en tiempo real cada vez que sea necesario.
7. Guardar el estado del inventario diario, es decir generar un histórico de inventarios con el objetivo de ser recuperado y saber como estaba el estado en una fecha cualquiera antes de la fecha actual.
8. Realizar un módulo de compras de acuerdo a las necesidades de la corporación en la que guarde solo datos necesarios.
9. Realizar un módulo de clientes en la cual el sistema almacene solo información necesaria.
10. Realizar un módulo de seguridades en donde se pueda crear y editar usuarios a la vez establecer permisos de accesos de acuerdo a funciones que puede tener cada usuario.

## **CAPÍTULO VI**

### **PROPUESTA**

#### **6.1 Tema**

SISTEMA DE FACTURACIÓN E INVENTARIOS PARA EL CONTROL TRIBUTARIO DE COMPRA Y VENTA EN LA CORPORACIÓN VPC.

#### **6.2 Datos Informativos**

**Institución:** Corporación VPC.

**Ciudad:** Ambato

**Dirección:** Pilahuín

**Investigador:** Franklin Ricardo Barrionuevo Caiza

**Tutor:** Ing. Galo López

#### **6.3 Antecedentes**

La implantación de un Sistema de Facturación e Inventarios surge por el crecimiento que ha tenido la Corporación, viéndose en la necesidad de tener un Sistema propio para realizar las transacciones en el punto de venta y tener un adecuado control del inventario, al mismo tiempo con la necesidad de estar actualizándose con las nuevas tecnologías existentes en estos tiempos.

Estando en una época donde se está optando por software libre y aprovechando todas las ventajas que nos da las mismas a un bajo costo, se ha optado por realizar el desarrollo e implantación de nuestro proyecto en dicha tecnología.

Teniendo en cuenta la exigencia en cuanto a seguridad, fiabilidad y veracidad de la información de la Corporación se ha decidido establecer como motor de base de datos al gestor llamado Postgres y para el desarrollo de la aplicación se estableció Visual Studio 2008 Express Edition (Visual C# y Visual Web Developer), que son versiones libres de la línea de Visual Studio de Microsoft. Para las validaciones utilizaremos JavaScript y para la reporteria iTextSharp.

#### **6.4 Justificación**

Hoy en día la tecnología ha tenido un avance notable, cada día viene mejoras y la mayoría de las empresas buscan estar al tanto de la tecnología existente, una de las novedades son las aplicaciones Web, pero ya no solo portales web se realizan en esta plataforma sino que ahora se están desarrollando sistemas complejos que serán utilizados solo con la utilización de un navegador, de esta manera con la utilización de internet se podría utilizar el sistema en varias sucursales de una forma rápida y económica permitiendo el acceso simultáneo de varios usuarios al servidor de datos.

El Software Libre hoy en día es una de las mejores alternativas al momento de realizar aplicaciones Windows y Web ya que proporciona oportunidades para profesionales en el campo de desarrollo, ya que el costo de licencias ha sido uno de los impedimentos para realizar proyectos de este tipo.

Al desarrollar con un lenguaje (C#) de versiones Express como lo es Visual Studio 2008 Express Edition (conjunto de programas gratuitos desarrollado por Microsoft) nos permite aprender este lenguaje de desarrollo y poderlo reutilizar cuando sea necesario sin la necesidad de adquirir licencias.

Es importante recalcar la capacidad que tiene Visual Studio 2008 Express Edition, para adaptarse con otros lenguajes de software libre como es JavaScript que nos

brinda posibilidades de mejor desarrollo de nuestras aplicaciones de forma segura, rápida y confiable a la hora de la manipulación de los datos.

También debemos tener en cuenta que tanto el software libre como el software propietario son herramientas poderosas para el desarrollo de aplicaciones, deberemos escoger dependiendo para donde se realizará el proyecto, ya que en otras ocasiones se deberá utilizar software propietario, todo dependiendo del desarrollador y del que lo necesita.

## **6.5 Objetivos**

### **6.5.1 Objetivo General**

Desarrollar e implantar un Sistema de Facturación e Inventarios utilizando Visual Studio 2008 Express Edition y Postgres para un adecuado control tributario de compra y venta de la Corporación VPC.

### **6.5.2 Objetivos Específicos**

- Analizar e interpretar los requerimientos facilitados por la Corporación para el posterior desarrollo del sistema.
- Diseñar correctamente la estructura de la base de datos para garantizar la información almacenada.
- Diseñar la interfaz de la aplicación con la utilización de estilos en cascada CSS.
- Desarrollar la aplicación con la utilizando lenguaje de programación C# con conexión a base de datos con la seguridad necesaria para garantizar la correcta manipulación de la información.
- Realizar las pruebas necesarias para garantizar el correcto funcionamiento de la aplicación.
- Realizar los respectivos manuales para la fácil utilización de la aplicación por parte de los usuarios.
- Capacitar al personal que estará a cargo de la manipulación del sistema.

## **6.6 Análisis de factibilidad**

### **6.6.1 Factibilidad Operativa**

La necesidad y deseo de un cambio del sistema actual, expresada por los usuarios de la Corporación, llevo la aceptación de un nuevo sistema que sea de una manera más sencilla y amigable, que cubra todos sus requerimientos, expectativas y proporcione la información en forma oportuna y confiable. Basándose en las entrevistas y conversaciones sostenidas con el personal se demostró que estos no representan ninguna oposición al cambio, por lo que el sistema es factible operacionalmente.

Con la finalidad de garantizar el buen funcionamiento del sistema y que este impactará de forma positiva a los usuarios, el sistema presentará una interfaz amigable al usuario, lo que se traduce en una herramienta de fácil manejo y comprensión.

Dependiendo del usuario logueado en el sistema podrá manipular el contenido de los diferentes módulos. Para que el usuario pueda conectarse al sistema debe tener una cuenta creada por el administrador, una vez conectado puede cambiar su clave y realizar las tareas mencionadas.

### **6.6.2 Factibilidad Económica**

Se ha determinado los recursos que se va a utilizar para el desarrollo e implantación de la aplicación, puesto que se utilizará lenguajes de programación y motor de base de datos Open Source, estaríamos fuera de costos de licencias. Es un proyecto factible puesto que el sistema aportará significativamente en las actividades diarias en el punto de venta, aportando a los usuarios con información de acuerdo a lo solicitado de manera rápida y oportuna.

### **6.6.3 Factibilidad Técnica**

La factibilidad Técnica consistió en realizar una evaluación de la tecnología existente en la Corporación, este estudio estuvo destinado a recolectar información sobre los

componentes técnicos que posee la Corporación y la posibilidad de hacer uso de los mismos en el desarrollo e implementación del sistema propuesto y de ser necesario, los requerimientos tecnológicos que deben ser adquiridos para el desarrollo y puesta en marcha del sistema.

De acuerdo a la tecnología necesaria para la implantación del sistema de Facturación e Inventarios en la Corporación VPC, se evaluó bajo dos enfoques:

### **Software**

- Lenguaje de programación C#
- Herramienta de Programación Visual Studio Express.
- Motor de base de datos Postgres
- Web Server IIS

### **Hardware**

Evaluando el hardware existente y tomando en cuenta la configuración mínima necesaria, la Corporación no requirió realizar inversión para la adquisición de nuevos equipos, tampoco actualizar los equipos existentes, ya que los mismos satisfacen los requerimientos establecidos tanto para el desarrollo y puesta en funcionamiento del sistema propuesto.

Para la puesta en marcha del sistema propuesto no requiere de computadoras de última generación, bastará que cuente con las características que se detalla a continuación:

- Sistema Operativo: Windows XP o Windows 7.
- Procesador: Pentium IV o superior.
- Memoria RAM: 1GB o superior.
- Disco Duro: 10GB libres o superior porque la base crecerá con el pasar del tiempo.
- Servidor Web: Internet Information Server (IIS).

## **6.7 Fundamentación**

### **6.7.1 C Sharp**

C# (pronunciado ci sharp en inglés) es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET. Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma.NET, similar al de Java aunque incluye mejoras derivadas de otros lenguajes (entre ellos Delphi).

La sintaxis y estructuración de C# es muy parecida a la de C++ o Java, puesto que la intención de Microsoft es facilitar la migración de códigos escritos en estos lenguajes a C# y facilitar su aprendizaje a los desarrolladores habituados a ellos. Sin embargo, su sencillez y el alto nivel de productividad son comparables con los de Visual Basic. C# es un lenguaje de programación que toma las mejores características de lenguajes preexistentes como Visual Basic, Java o C++ y las combina en uno solo.

### **6.7.2 CSS**

CSS (Cascading Style Sheets) es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML (HyperText Markup Language) y XHTML (eXtensible HyperText Markup Language). CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "documentos semánticos"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

Al crear una página web, se utiliza en primer lugar el lenguaje HTML/XHTML para marcar los contenidos, es decir, para designar la función de cada elemento dentro de

la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc. Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc.

### **6.7.3 JavaScript**

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

### **6.7.4 Internet Information Server (IIS)**

#### **¿Qué es un servidor web?**

Un servidor web es un programa que se está ejecutando en un equipo, normalmente un servidor y que proporciona páginas web a los "clientes" que le piden. Los clientes son los navegadores web como Internet Explorer, Firefox o NetScape.

Cuando llamamos a un equipo "servidor" nos referimos a un equipo que va a proporcionar determinados servicios a los usuarios de una red local o de Internet. Por tanto si en un navegador o explorador web escribimos una dirección o una página intentará localizar el servidor escrito y buscar la página solicitada. Para que ocurra esto debemos tener instalado en nuestro servidor el programa Internet Information Server. Internet Information Server, que llamaremos IIS, es el servidor de páginas web de Microsoft. Éste viene de forma gratuita con el sistema operativo Windows NT, 2000 y XP y descargable para los demás.

### **6.7.5 iTextSharp**

iText es una biblioteca Open Source que permite crear y manipular documentos PDF. Permite a los desarrolladores que buscan mejorar las aplicaciones web y otros con la generación dinámica de documentos PDF. Cuenta con beneficios como Mostrar los PDFs en un Navegador, Agregar marcadores de libros, números de página, marcas de agua, etc.

### **6.8. Metodología**

En la actualidad, para desarrollar un producto de software es necesario contar con un plan bien estructurado. El cliente tiene que comprender que es lo que hará el investigador; además tiene que ser capaz de señalar cambios si no han sido claramente asimiladas sus necesidades.

La clave está en organizar el proceso de diseño de tal forma que los analistas, clientes, desarrolladores y otras personas involucradas en el desarrollo del sistema lo comprendan y convengan con él.

Para el desarrollo de la propuesta se utilizó pasos ordenados con un ciclo de vida basado en el Análisis, Diseño, Pruebas e Implantación tomando como ejemplo una metodología orientada a objetos basada en el lenguaje unificado de modelado UML (Unified Modeling Language - Lenguaje Unificado de Modelado), que es un popular lenguaje de modelado de sistemas de software. Se trata de un lenguaje gráfico para construir, documentar, visualizar y especificar un sistema de software. Entre otras palabras, UML se utiliza para definir un sistema de software.

Posee la riqueza suficiente como para crear un modelo del sistema, pudiendo modelar los procesos de negocios, funciones, esquemas de bases de datos, expresiones de lenguajes de programación, etc.

## **6.9 Modelo Operativo**

### **6.9.1 Análisis del Sistema**

#### **6.9.1.1 Análisis y Requerimientos del Sistema**

La corporación VPC no cuenta con un sistema de facturación que esté al alcance y cumpla con todas las necesidades que se presentan, es por tal razón que se decide implantar un Sistema que sea creado de acuerdo a los requerimientos de la Corporación a fin de brindar un mejor servicio a todos sus clientes. De acuerdo a las necesidades de la corporación VPC para el desarrollo del sistema se toma como referencia los siguientes aspectos:

- Creación de Facturas
- Revisión de Facturas por estado y fecha.
- Control de Inventarios
- Administración de productos vendidos a Crédito.
- Administración de Compras.
- Administración de Clientes.
- Administración de Proveedores.
- Administración de Nuevas Sucursales.
- Administración de Usuarios por Sucursales.
- Administración de Seguridades (modificar, eliminar y administrar usuarios).
- Administración de datos generales (País, Provincias, Cantones, Teléfonos, entre otros).
- Reportearía en general.

Referente a las medidas de seguridad, cada usuario tendrá su clave para el ingreso al sistema la misma que será encriptado y almacenado en la base de datos, con esto lograríamos que ningún otro usuario pueda descifrar la clave.

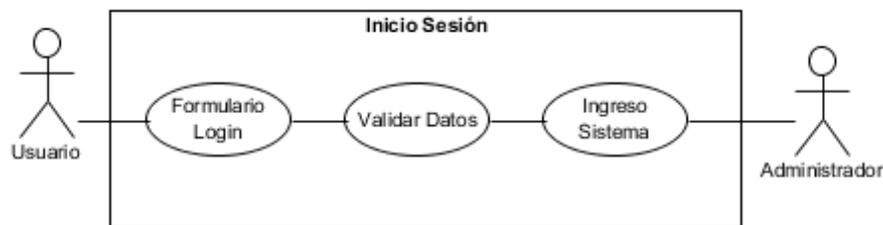
La tecnología que se utilizará para el desarrollo e implementación del sistema propuesto será todo referente a lo que tenga que ver con software libre. La idea de software libre es dejar a un lado el costo de licencias.

## 6.9.1.2 Diagramas UML

### 6.9.1.2.1 Diagramas de Caso de Uso

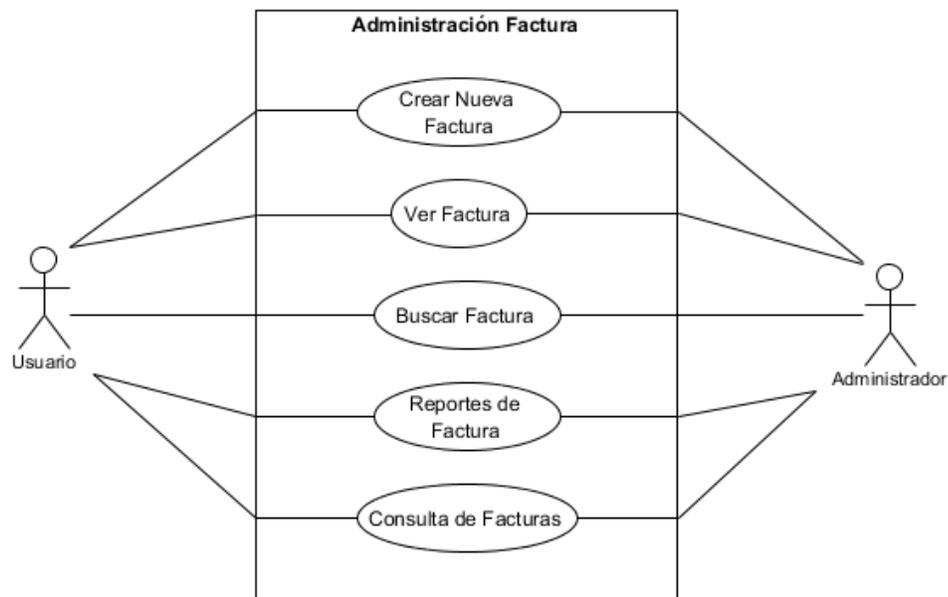
Los diagramas de casos de uso documentan el comportamiento de un sistema desde el punto de vista del usuario. Determinan los requisitos funcionales del sistema, es decir, representan las funciones que un sistema puede ejecutar. Su ventaja principal es la facilidad para interpretarlos, lo que hace que sean especialmente útiles en la comunicación con el cliente.

#### Ingreso Sistema



**Figura 6.1** Caso de Uso Ingreso Sistema

#### Administración Factura



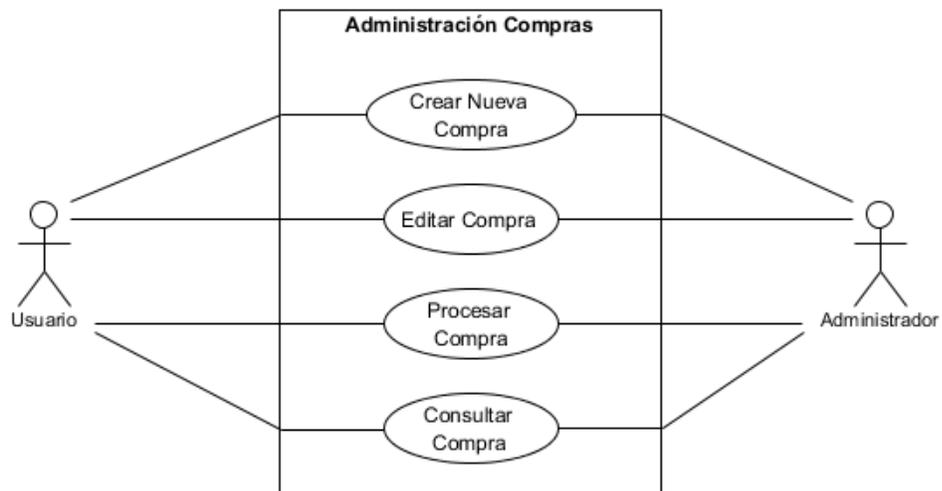
**Figura 6.2** Caso de Uso Administración Factura

## Administración Inventario



**Figura 6.3** Caso de Uso Administración Inventario

## Administración Compras



**Figura 6.4** Caso de Uso Administración Compras

## Administración Créditos

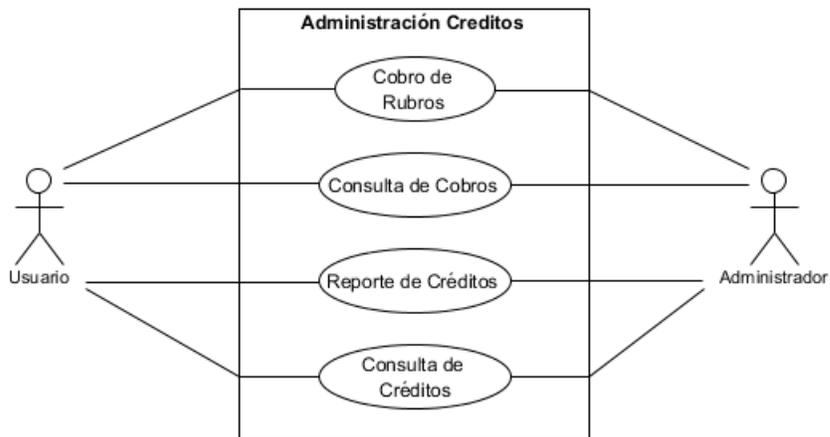
Ver **Figura 6.5** Caso de Uso Administración Créditos

## Administración Clientes

Ver **Figura 6.6** Caso de Uso Administración Clientes

## Administración Color Estándar

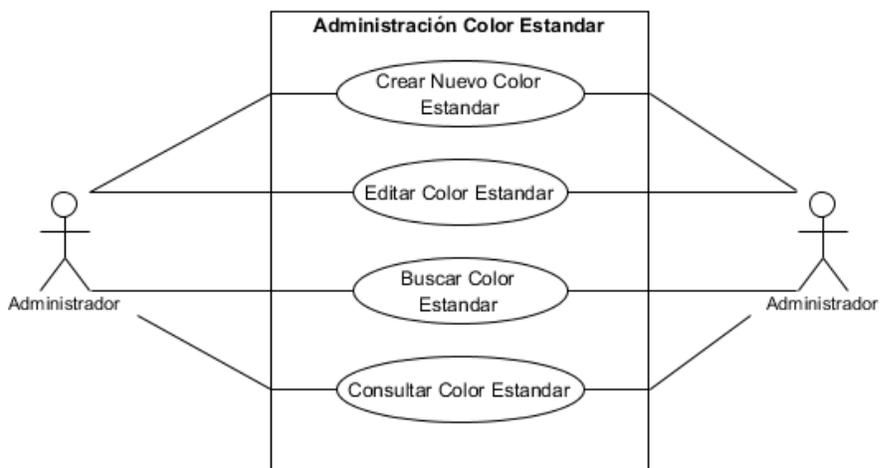
Ver **Figura 6.7** Caso de Uso Administración Color Estándar



**Figura 6.5** Caso de Uso Administración Créditos



**Figura 6.6** Caso de Uso Administración Clientes



**Figura 6.7** Caso de Uso Administración Color Estandar

## Administración Generación de Colores

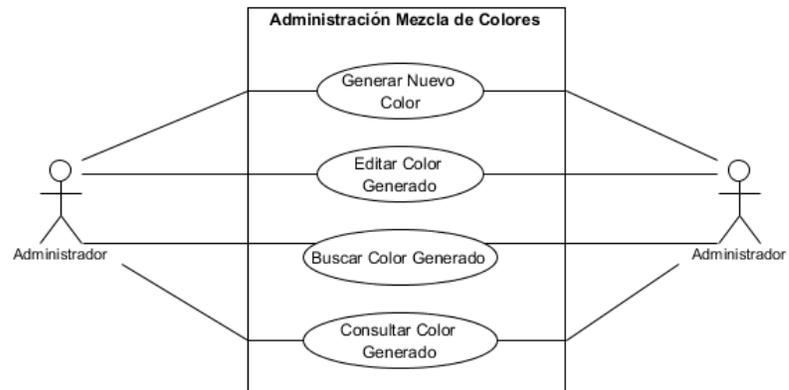


Figura 6.8 Caso de Uso Administración Generación de Colores

## Administración Parámetros

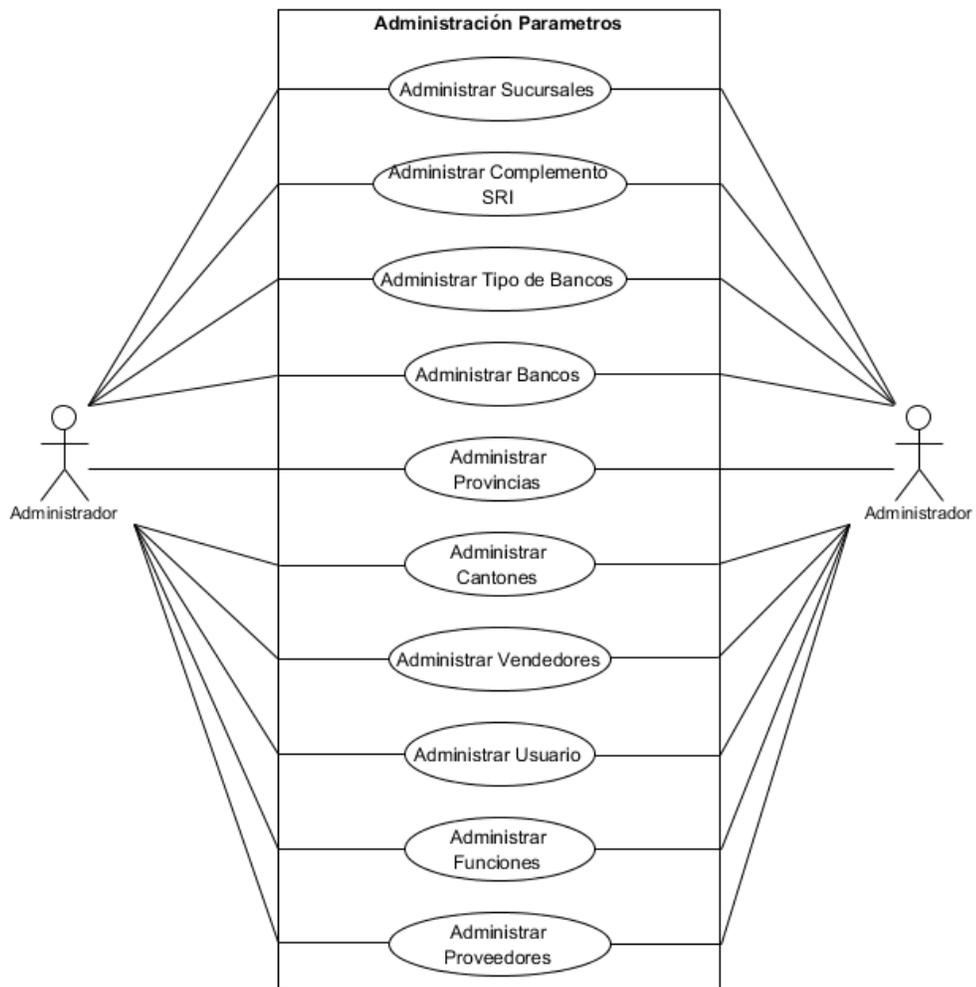
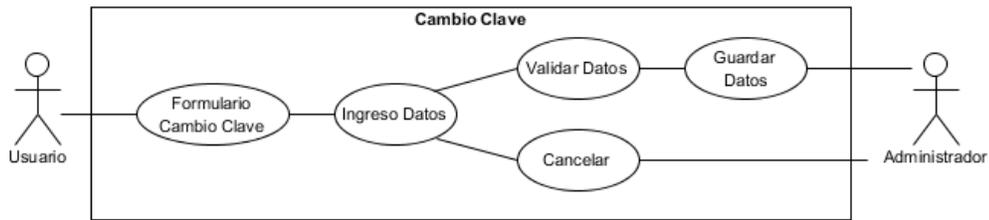


Figura 6.9 Caso de Uso Administración Parámetros

## Administración Seguridad



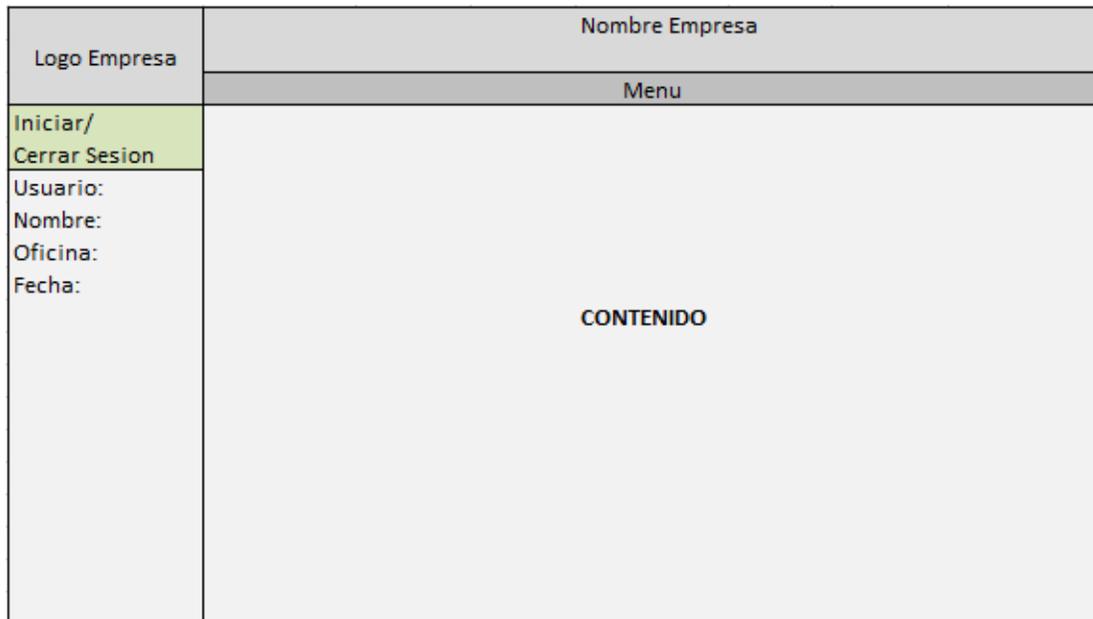
**Figura 6.10** Caso de Uso Administración Seguridad

### 6.9.1.2.2 Diseño de la Interfaz

Para el diseño se la interfaz del sistema informático propuesto se a tomado como referencia los requerimientos proporcionados por la Corporación.

#### Pantalla Principal

Esta es la pantalla principal, la cual aparecerá en todo momento a excepción del contenido que será nuestro espacio de trabajo.



**Figura 6.11** Diseño de la Interfaz Pantalla Principal

#### Inicio de Sesión

Esta pantalla aparecerá cuando inicie sesión en la cual le pedirá que ingrese el nombre de usuario y la contraseña, estos datos tienen que ser ingresados previamente, en esta pantalla se validará los datos para permitir o rechazar el acceso al sistema.

**Figura 6.12** Diseño de la Interfaz Inicio Sesión

1. Caja de texto para el ingreso del Nombre de Usuario.
2. Caja de texto para el ingreso de la contraseña.
3. Botón para el Inicio de Sesión, aquí estará la validación, si los datos son correctos podrá ingresar al sistema de lo contrario le cotara un error.

### Vista de Facturas

En esta pantalla aparecerán todas las facturas que han sido ingresadas al sistema, en la cual para una mejor visualización se podrá filtrar por Sucursal, Fecha y Estado.

Listado de Facturas Ingresadas					
Factura 1					
Factura 2					
Factura 3					

**Figura 6.13** Diseño de la Interfaz Vista de Facturas

1. Botón para la creación de una nueva Factura.
2. Botón para ver una Factura creada anteriormente.
3. Botón para empezar una búsqueda de Facturas.
4. Botón para eliminar una Factura creada, al ejecutar este botón el inventario será restaurado.

5. Caja DropDownList para desplegar las sucursales existentes en el sistema y poder filtrar las Facturas por Sucursales.
6. Caja de texto para ingresar la fecha y poder filtrar las facturas por fecha.
7. Caja DropDownList para desplegar los diferentes estados en que puede estar la factura y poder filtrar las Facturas por estado.
8. GridView para mostrar las Facturas según los filtros ingresados.

## Administración Facturas

En esta pantalla se podrá realizar el proceso de facturación, esta compuesta por el encabezado, detalle, desglose (detalle de los colores que se ocuparon para generar un nuevo color).

<b>Guardar</b> 1					
Código:	<input type="text"/>	2	Fecha:	<input type="text"/>	5
Identificación:	<input type="text"/>	3	Dirección:	<input type="text"/>	6
Nombre:	<input type="text"/>	4	Teléfono:	<input type="text"/>	7
<div style="border: 1px solid black; padding: 5px; display: inline-block;"> <b>DATOS</b>  <b>FACTURA</b>  <b>SRI</b> </div>					8
<b>DETALLE</b>					
Código:	<input type="text"/>	Nombre:	<input type="text"/>	Cantidad:	<input type="text"/>
	9		10		11
Descuento:	<input type="text"/>				
					12
<div style="border: 1px solid black; padding: 2px; display: inline-block;"> <b>Guardar</b> </div>					13
<div style="border: 1px solid black; padding: 2px; display: inline-block;"> <b>Eliminar</b> </div>					14
<b>Datos del Detalle</b>					15
COLOR 1	<input type="text"/>				
COLOR 2	<input type="text"/>				
Comentario:	<input type="text"/>				
	16				
				Subtotal:	<input type="text"/>
				Desto.:	<input type="text"/>
				Iva:	<input type="text"/>
				Total:	<input type="text"/>
					17
					18
					19
					20
<b>DESGLOSE</b>					
<b>Datos del Desglose</b>					21
Color 1	<input type="text"/>				
Color 2	<input type="text"/>				
				Subtotal:	<input type="text"/>
					22

**Figura 6.14** Diseño de la Interfaz Administración de Facturas

1. Botón para guardar la Factura, este botón será el principal de esta pantalla ya que una vez pasado todas las validaciones se procederá a ejecutar procesos adicionales como a disminuir el inventario.
2. Caja de texto para el ingreso del Código del Cliente.
3. Caja de texto para el ingreso de la Identificación del Cliente.
4. Caja de texto para la visualización del nombre del Cliente, este aparecerá de acuerdo al código o la identificación ingresada.

5. Caja de texto para la visualización de la Fecha, esta aparecerá de acuerdo a la fecha del sistema.
6. Caja de texto para la visualización de la dirección, esta aparecerá de acuerdo al código o la identificación ingresada.
7. Caja de texto para la visualización del teléfono, este aparecerá de acuerdo al código o la identificación ingresada.
8. Cajas de texto que contendrá la información proporcionada por el SRI, como son el ruc de la empresa, la autorización y el secuencial de la factura.
9. Caja de texto para ingresar el Código de un color Generado.
10. Caja de texto para ingresar el Nombre de un color Generado.
11. Caja de texto para ingresar la cantidad de color que se va adquirir.
12. Caja de texto para ingresar un descuento en caso de que lo hubiese.
13. Botón para Guardar el color y visualizar en el listado del detalle.
14. Botón para Eliminar un color que ya no se desea del listado del detalle.
15. GridView con el detalle de los Colores Ingresados para la Factura.
16. Caja de texto para algún comentario de la Factura.
17. Caja de texto que mostrara el subtotal de la Factura.
18. Caja de texto para un segundo descuento, pero esta vez del subtotal.
19. Caja de texto para el IVA.
20. Caja de Texto para visualizar el total de la Factura.
21. GridView que mostrara el desglose de los colores que se necesito para generar cada color del Detalle de la Factura.
22. Caja de texto para visualizar el subtotal de cada Detalle de Factura.

### **Administración Compras**

En esta pantalla se podrá ingresar una nueva compra o editar una que haya sido ingresada anteriormente, se podrá editar siempre y cuando la compra aun no sea procesada es decir cuando aun no haya afectado el inventario, de lo contrario solo se podrá visualizar datos.

Ver **Figura 6.15** Diseño de la Interfaz Administración Compra

1. Botón para Guardar la Compra, guardara siempre y cuando haya pasado las validaciones.
2. Caja de texto para visualizar el Secuencial, este campo será generado automáticamente por el Sistema.
3. Caja de texto para ingresar el proveedor.

4. Caja de texto para visualizar la fecha, esta fecha lo dará el sistema automáticamente.
5. Caja de texto para visualizar el estado en la que se encuentra la compra.
6. Caja de texto para visualizar el total de la Compra.
7. Botón para Guardar un ítem en el detalle de la compra.
8. Botón para eliminar un ítem seleccionado del detalle de la compra.
9. Caja de texto para la visualización de secuencial del detalle, este campo será generado automáticamente por el sistema.
10. Caja de texto para visualizar el color que será ingresado al detalle.
11. Caja de texto para visualizar la sucursal a la que se va a realizar la compra.
12. Caja de texto para ingresar la cantidad en litros para la compra.
13. Caja de texto para ingresar el costo por cada litro.
14. Caja de texto para visualizar el Total por cada ítem de la compra.
15. GridView para visualizar el detalle de la compra.
16. Caja de texto para visualizar el subtotal de la compra.
17. Caja de texto para ingresar el descuento por la compra.
18. Caja de texto para visualizar el Total Neto de la compra.
19. Caja de texto para ingresar los valores de Otros Cargos si lo hubiese.
20. Caja de texto para visualizar el valor del IVA.
21. Caja de texto para visualizar el total de la compra.

### **Administración Clientes**

En esta pantalla se podrá ingresar un nuevo cliente o editar uno existente.

Ver **Figura 6.16** Diseño de la Interfaz Administración Clientes.

1. Botón para Guardar al Cliente, guardara siempre u cuando haya pasado todas las validaciones.
2. Caja de texto para el tipo de Identificación del Cliente, que podría se Cedula, Ruc, Pasaporte, etc.
3. Caja de texto para ingresar la Identificación del Cliente.
4. Caja de texto para ingresar un Código al Cliente.
5. Caja de texto para Ingresar la ciudad en la que vive el Cliente.
6. Caja de texto para ingresar el nombre del Cliente.
7. Caja de texto para ingresar la dirección del Cliente.
8. Caja de texto para ingresar el correo del Cliente.
9. Caja de texto para ingresar al vendedor que será responsable para este Cliente.
10. Ingreso de los teléfonos que puede tener el Cliente.
11. Ingreso de las formas de venta que puede tener este Cliente.

<b>Guardar</b> 1	
Secuencial:	<input type="text"/> 2
Proveedor:	<input type="text"/> 3
Fecha:	<input type="text"/> 4
Estado:	<input type="text"/> 5
Total:	<input type="text"/> 6
<b>DETALLE</b>	
7 <b>Guardar</b>	<b>Eliminar</b> 8
Secuencial:	<input type="text"/> 9
Color Estandar:	<input type="text"/> 10
Sucursal:	<input type="text"/> 11
Cantidad:	<input type="text"/> 12
Costo:	<input type="text"/> 13
Total:	<input type="text"/> 14
<b>Detalle Compra</b> 15	
COLOR 1	<input type="text"/>
COLOR 2	<input type="text"/>
COLOR 3	<input type="text"/>
SubTotal:	<input type="text"/> 16
Descuento:	<input type="text"/> 17
Total Neto:	<input type="text"/> 18
Otros Cargos:	<input type="text"/> 19
Iva:	<input type="text"/> 20
Total Compra:	<input type="text"/> 21

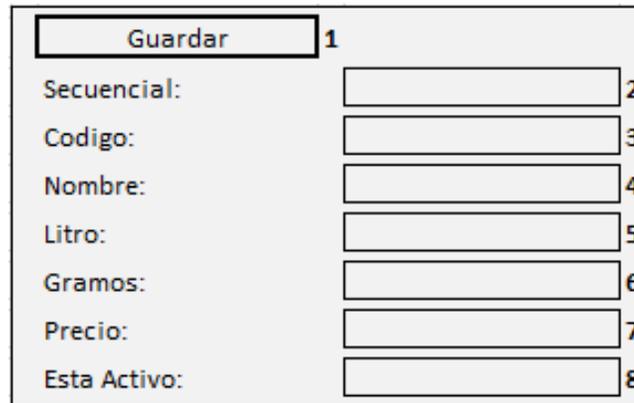
**Figura 6.15** Diseño de la Interfaz Administración Compras

<b>Guardar</b> 1	
Tipo Identificación:	<input type="text"/> 2
Identificación:	<input type="text"/> 3
Código:	<input type="text"/> 4
Ciudad:	<input type="text"/> 5
Nombre:	<input type="text"/> 6
Dirección:	<input type="text"/> 7
Correo:	<input type="text"/> 8
Vendedor:	<input type="text"/> 9
Telefono 10	Forma de Venta 11
<input type="text"/> 1	<input type="text"/> 1
<input type="text"/> 2	<input type="text"/> 2
<input type="text"/> 3	<input type="text"/> 3

**Figura 6.16** Diseño de la Interfaz Administración Clientes

## Administración Color

En esta pantalla se podrá ingresar un nuevo Color o Editar uno existente. Es importante ingresar los datos correctos ya que esta información será la referencia para la generación de colores.



La interfaz de usuario para la administración de colores se muestra en un cuadro gris con los siguientes elementos:

- Botón "Guardar" etiquetado con el número 1.
- Caja de texto "Secuencial:" etiquetada con el número 2.
- Caja de texto "Codigo:" etiquetada con el número 3.
- Caja de texto "Nombre:" etiquetada con el número 4.
- Caja de texto "Litro:" etiquetada con el número 5.
- Caja de texto "Gramos:" etiquetada con el número 6.
- Caja de texto "Precio:" etiquetada con el número 7.
- Caja de texto "Esta Activo:" etiquetada con el número 8.

**Figura 6.17** Diseño de la Interfaz Administración Color

1. Botón Guardar Color, se guardara siempre y cuando haya pasado todas las validaciones.
2. Caja de texto secuencial, este campo será generado automáticamente por el sistema.
3. Caja de texto para ingresar el Código del Color.
4. Caja de texto para ingresar el Nombre del Color.
5. Caja de texto que visualiza el valor en litros.
6. Caja de texto para ingresar el equivalente de un litro pero en gramos.
7. Caja de texto para ingresar el precio por litro.
8. CheckBox para activar o inactivar este color.

## Administración Generación de Color

En esta pantalla se va a proceder a mezclar los colore y generar uno nuevo, será el color generado el que aparecerá para ingresar en el detalle de la factura al momento de un nuevo ingreso.

**Figura 6.18** Diseño de la Interfaz Administración Generación Color

1. Botón Guardar Color Generado, guardara siempre y cuando haya pasado todas las validaciones.
2. Caja de texto para visualizar el Secuencial, este campo será generado automáticamente por el sistema.
3. Caja de texto para ingresar el código del color que será generado.
4. Caja de texto para ingresar el nombre del color que será generado.
5. CheckBox para activar o inactivar este color generado.
6. Botón para un nuevo ítem.
7. Botón para ingresar el nuevo ítem al detalle.
8. Botón para editar un ítem que ya ha sido ingresado.
9. Botón para eliminar un ítem seleccionado del detalle.
10. GridView para mostrar los ítems que han sido ingresados al detalle.
11. Caja de texto para mostrar el total de color ingresado en Litros.
12. Caja de texto para mostrar el total de color ingresado en Gramos.

### Administración Usuarios

En esta pantalla se podrá administrar a los usuarios que ya han sido ingresados al sistema, esto es parte de la seguridad del sistema.

**Figura 6.19** Diseño de la Interfaz Administración Usuarios

1. GridView, muestra todos los usuarios ingresados al sistema.
2. Botón para proceder a la edición del usuario seleccionado en el GridView.
3. Botón para proceder a la eliminación del usuario seleccionado en el GridView.
4. Botón para proceder a la edición de funciones (roles) que podrá tener el usuario seleccionado en el GridView.

### 6.9.1.2.3 Diagramas de Clase

Un Diagrama de Clases representa las clases que serán utilizadas dentro del sistema y las relaciones que existen entre ellas.

### Facturación

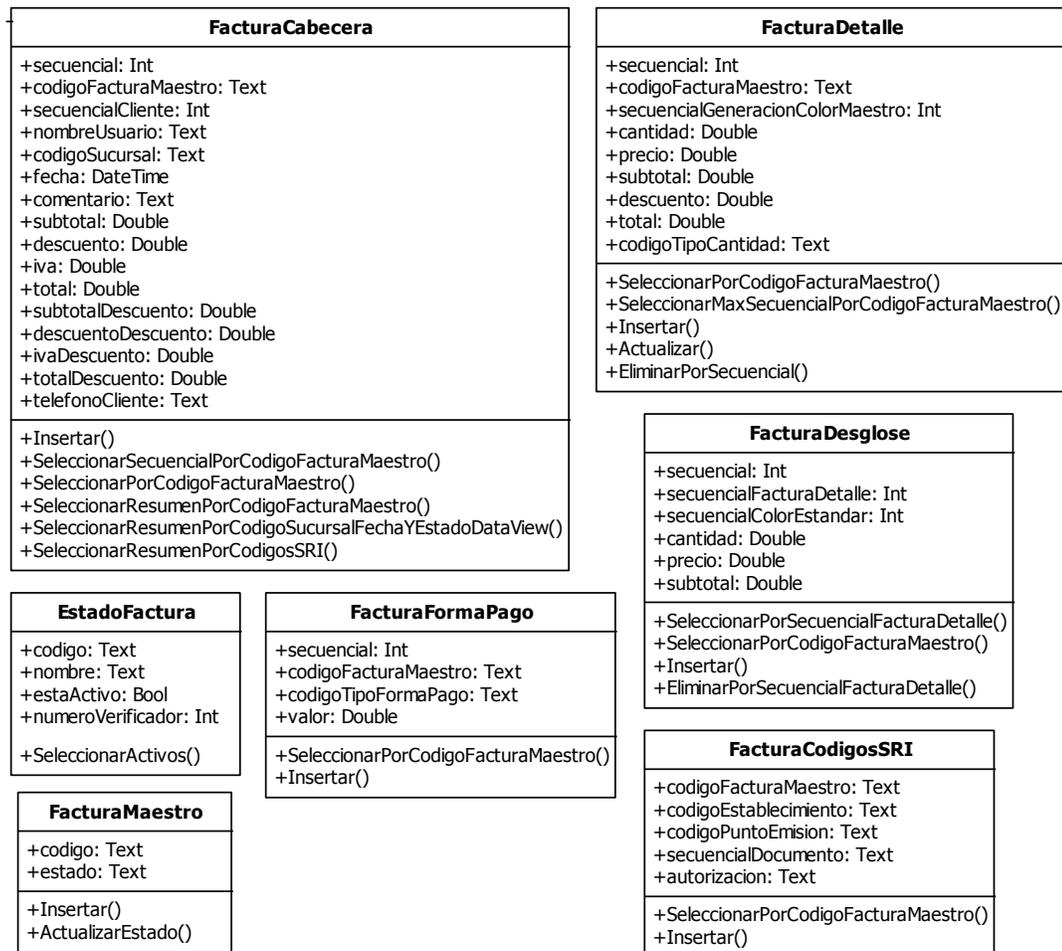


Figura 6.20 Diagrama de Clase Facturación

## Compras

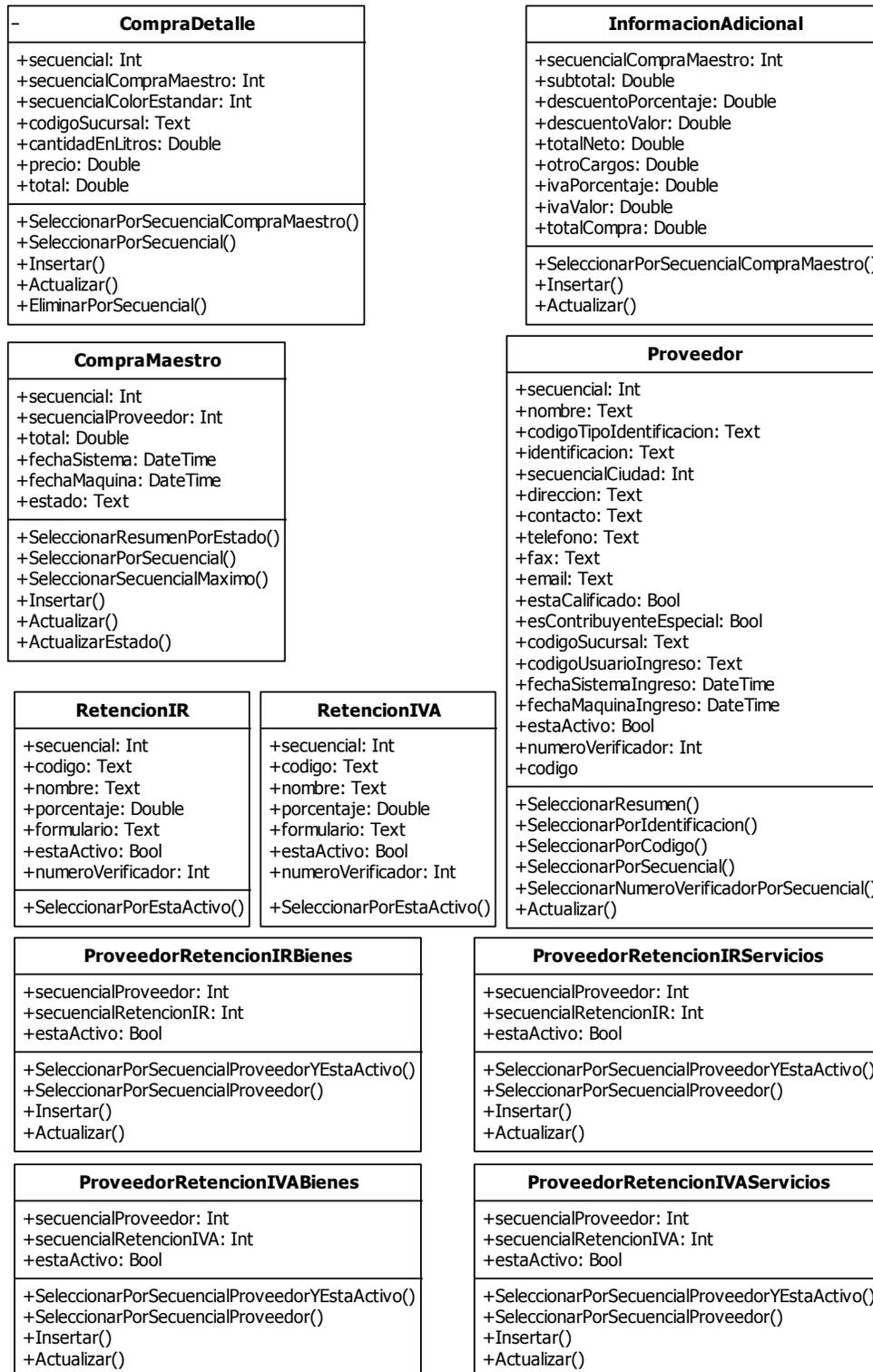


Figura 6.21 Diagrama de Clase Compras

## Cientes

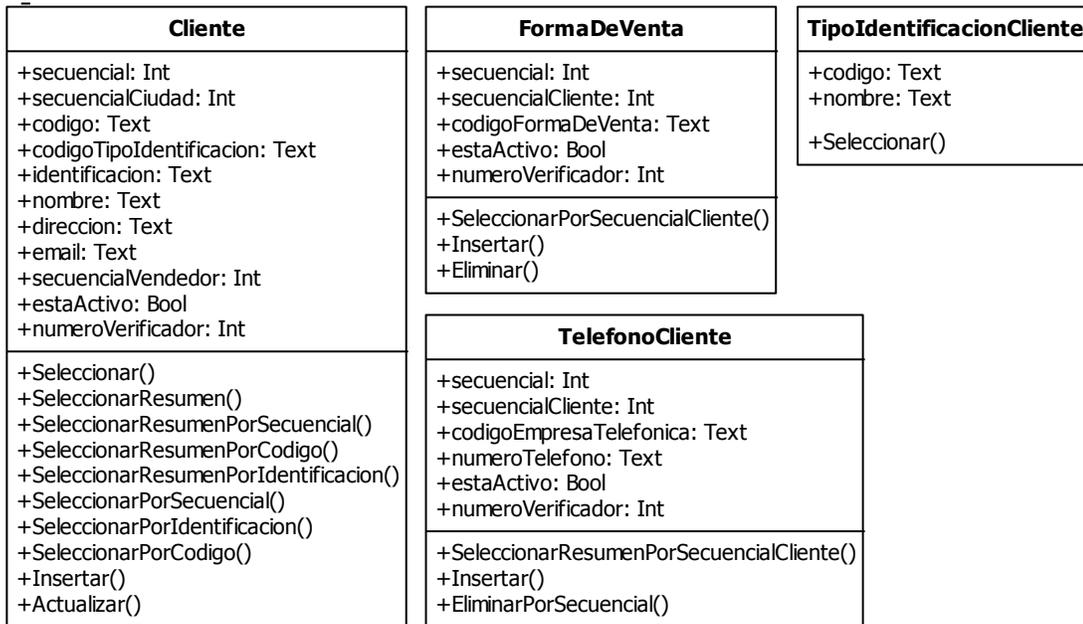


Figura 6.22 Diagrama de Clase Cientes

## Colores

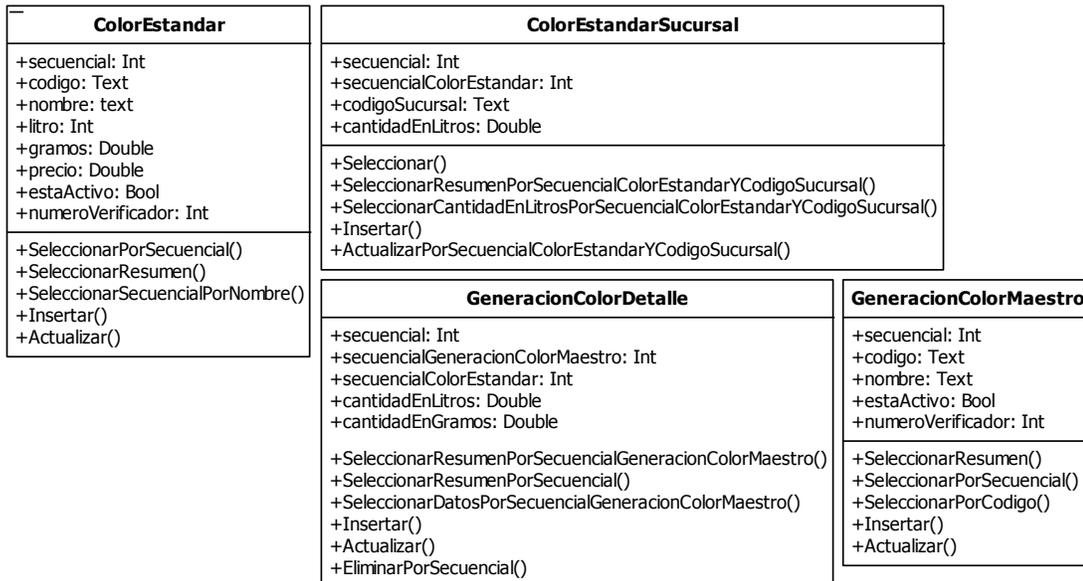


Figura 6.23 Diagrama de Clase Colores

## Créditos

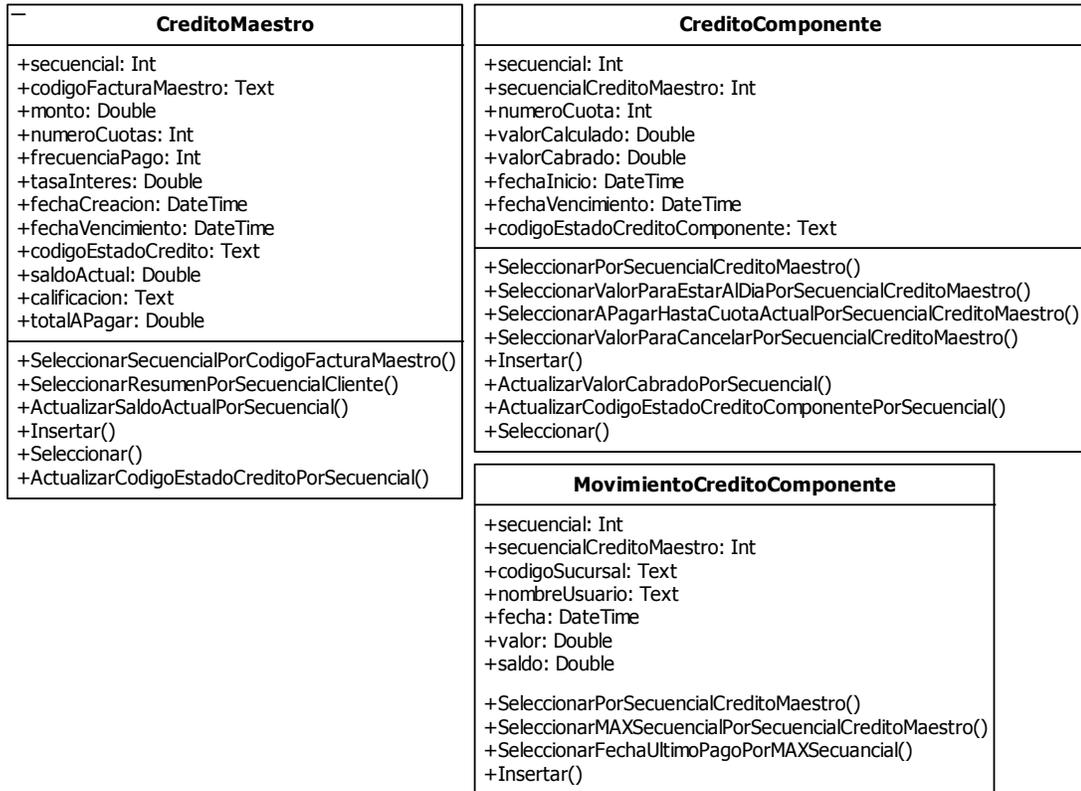


Figura 6.24 Diagrama de Clase Créditos

## Históricos

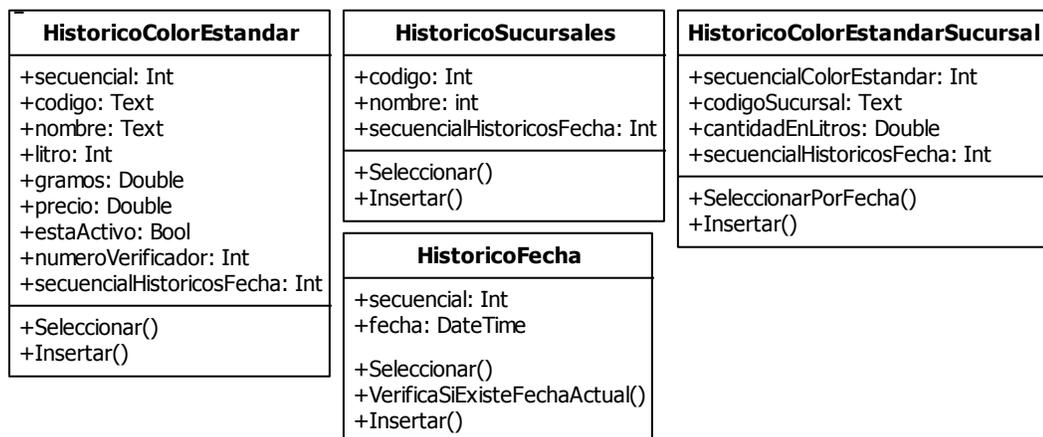


Figura 6.25 Diagrama de Clase Históricos

## Generales

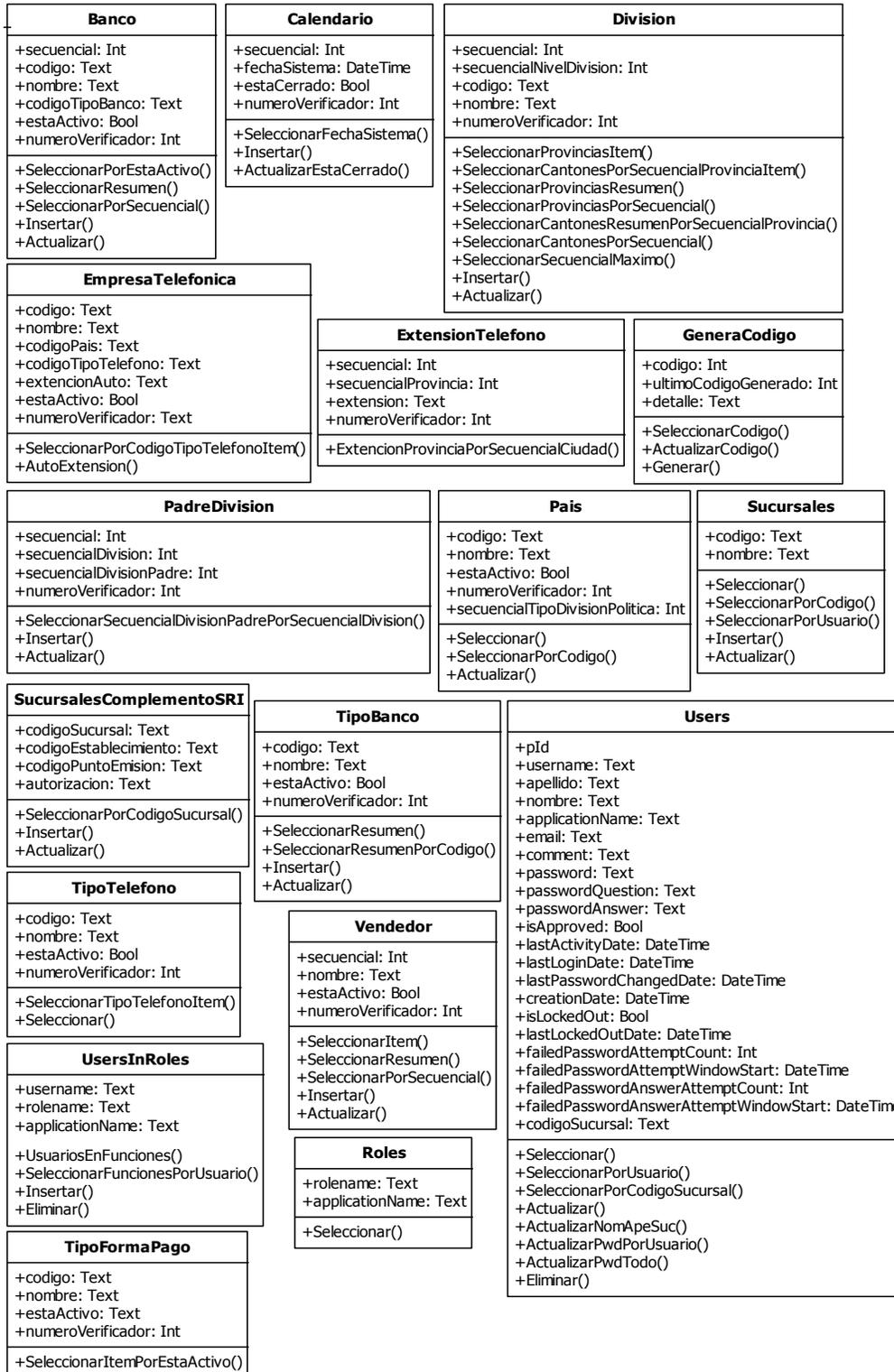


Figura 6.26 Diagrama de Clase Generales

#### **6.9.1.2.4 Diagramas de Secuencia**

Un Diagrama de Secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso. Mientras que el diagrama de casos de uso permite el modelado de una vista business del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes intercambiados entre los objetos.

#### **Nueva Factura**

Ver **Figura 6.27** Diagrama de Secuencia Nueva Factura

#### **Ver Factura**

Ver **Figura 6.28** Diagrama de Secuencia Ver Factura

#### **Ver Inventario**

Ver **Figura 6.29** Diagrama de Secuencia Ver Inventario

#### **Reporte Inventario**

Ver **Figura 6.30** Diagrama de Secuencia Reporte Inventario

#### **Cobro Crédito**

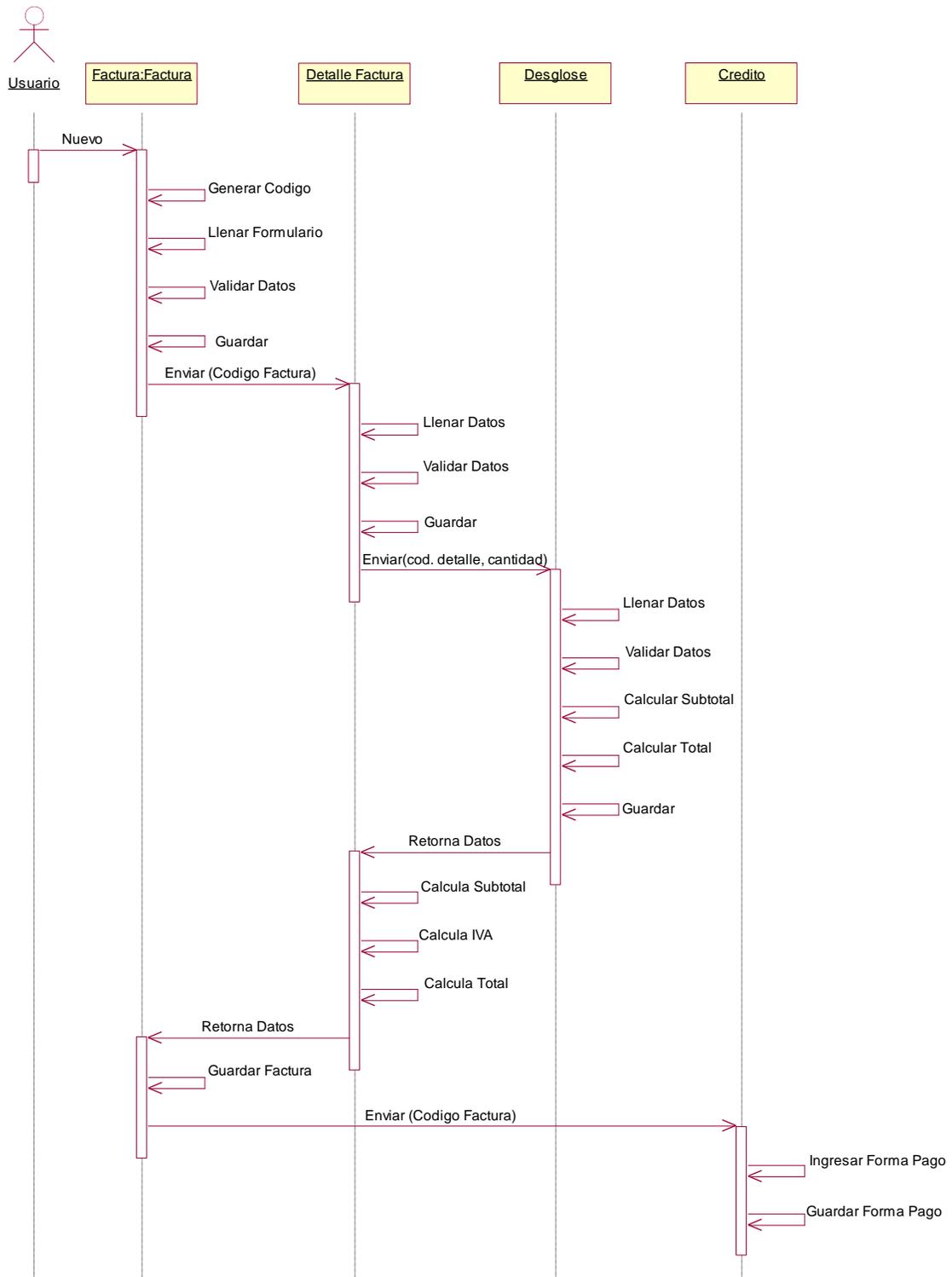
Ver **Figura 6.31** Diagrama de Secuencia Cobro Crédito

#### **Consulta Crédito**

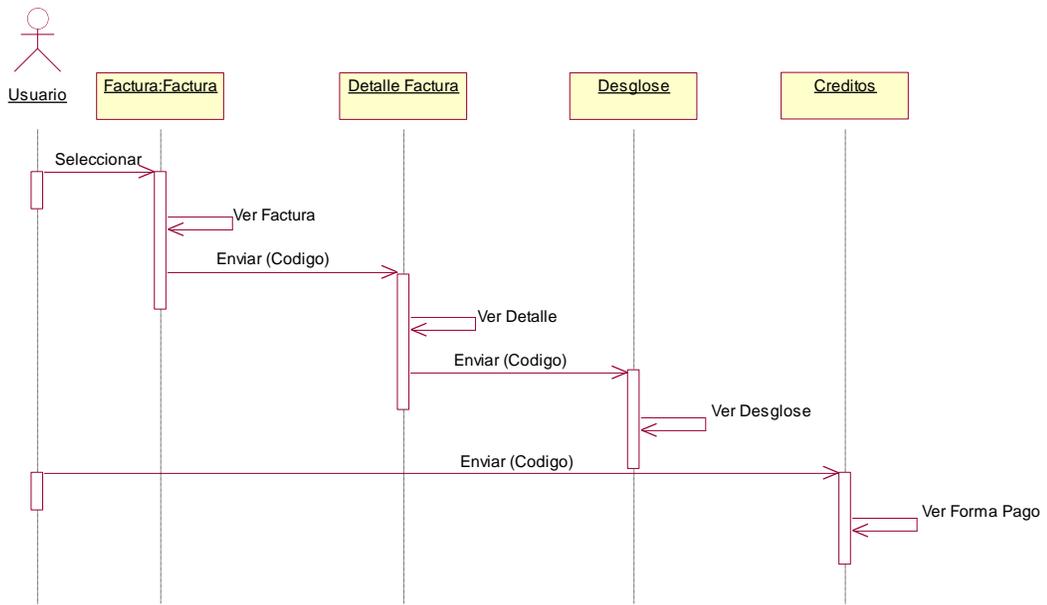
Ver **Figura 6.32** Diagrama de Secuencia Consulta Crédito

#### **Compras**

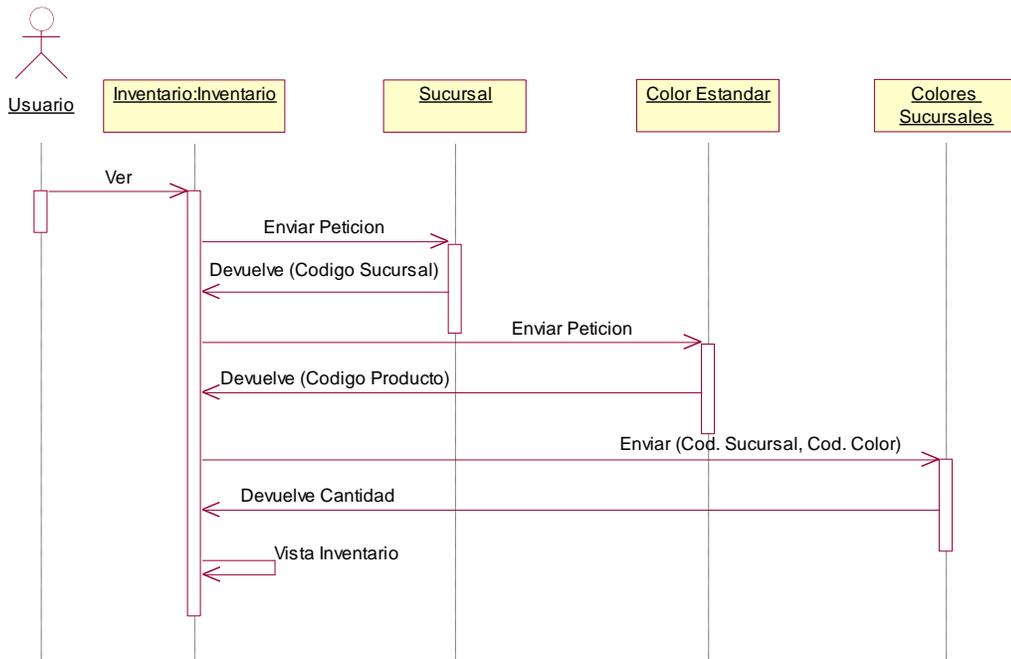
Ver **Figura 6.33** Diagrama Secuencia Compras



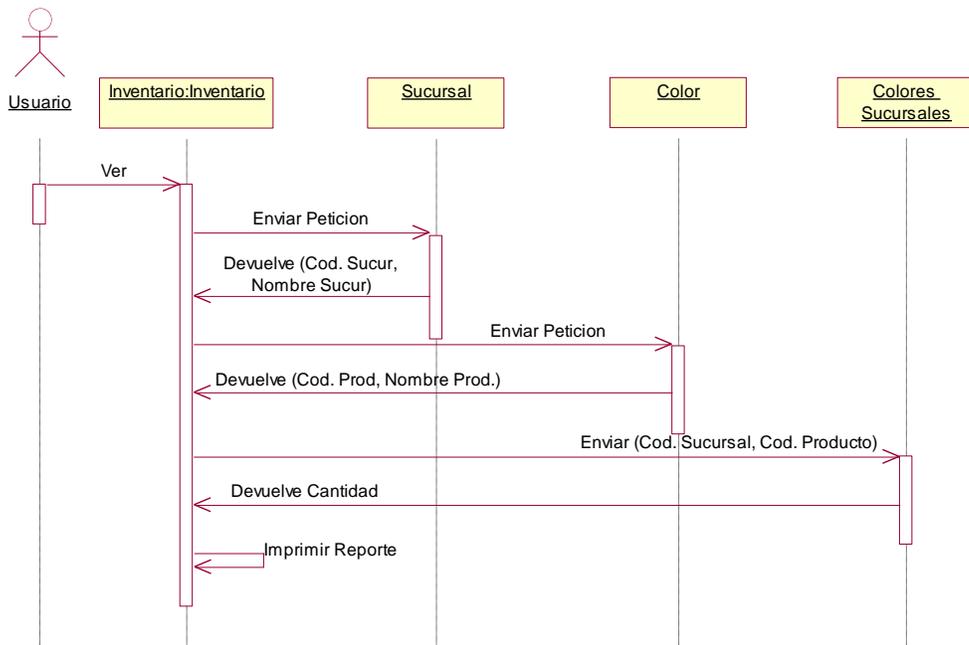
**Figura 6.27** Diagrama de Secuencia Nueva Factura



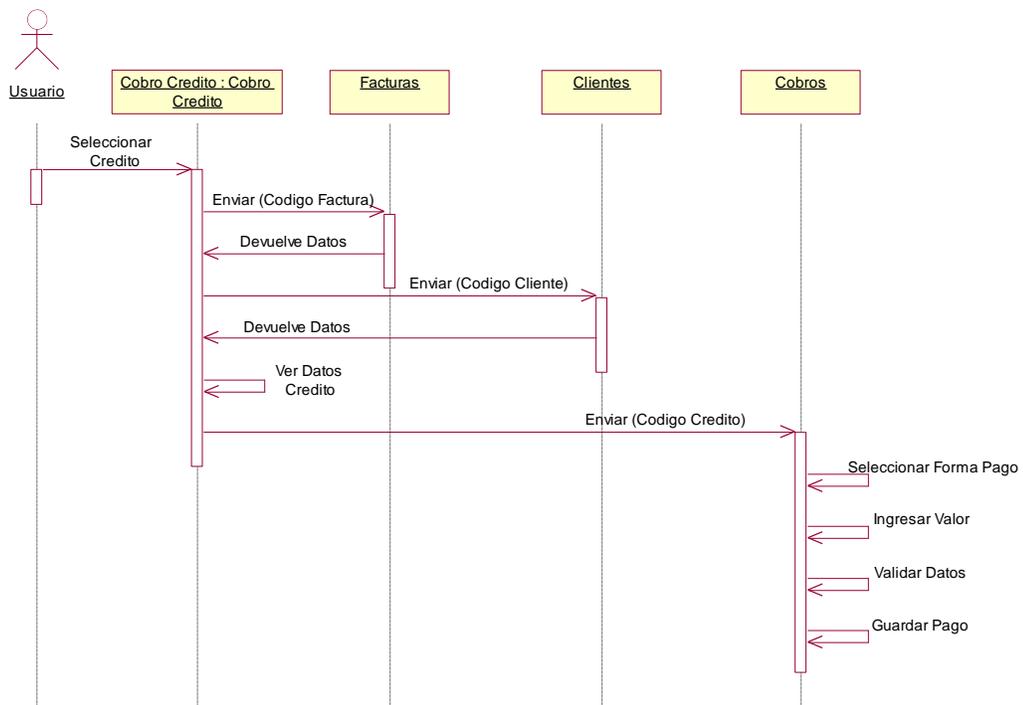
**Figura 6.28** Diagrama de Secuencia Ver Factura



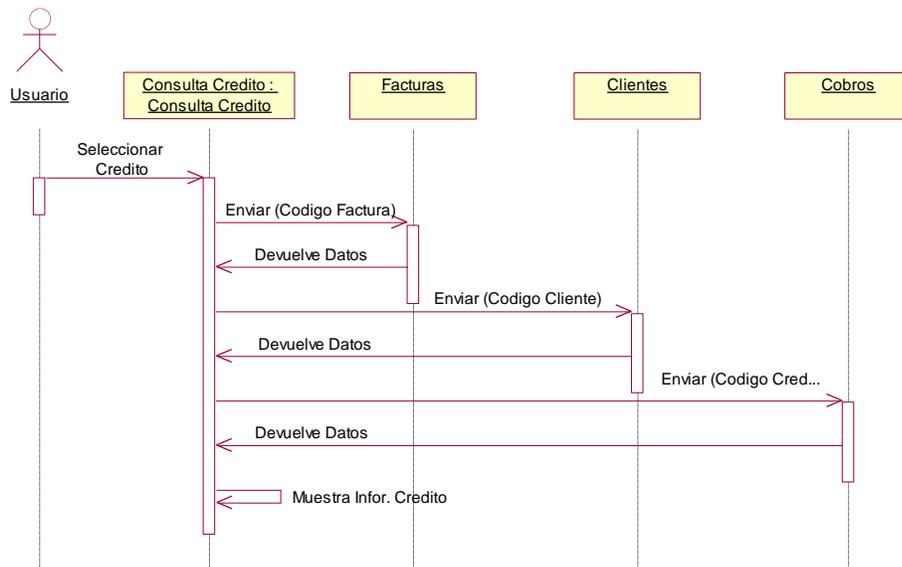
**Figura 6.29** Diagrama de Secuencia Ver Inventario



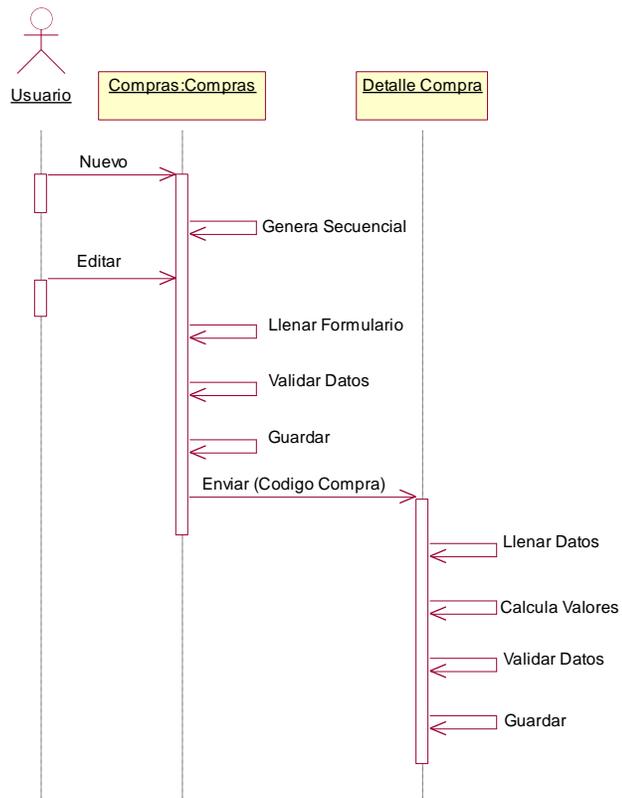
**Figura 6.30** Diagrama de Secuencia Reporte Inventario



**Figura 6.31** Diagrama de Secuencia Cobro Crédito



**Figura 6.32** Diagrama de Secuencia Consulta Crédito



**Figura 6.33** Diagrama de Secuencia Compras

## 6.9.2 Diseño del Sistema

### 6.9.2.1 Diseño de la Base de Datos

Una base de datos correctamente diseñada permite obtener acceso a información exacta y actualizada. Puesto que un diseño correcto es esencial para lograr los objetivos fijados, es lógico emplear el tiempo que sea necesario para diseñar y que la base de datos termine adaptándose a nuestras necesidades y pueda modificarse fácilmente. Una vez analizado la información otorgada por la Corporación se ha modelado la base de datos de la siguiente manera, la cual esta diseñada mediante esquemas para una mejor administración.

#### Esquema: Facturación

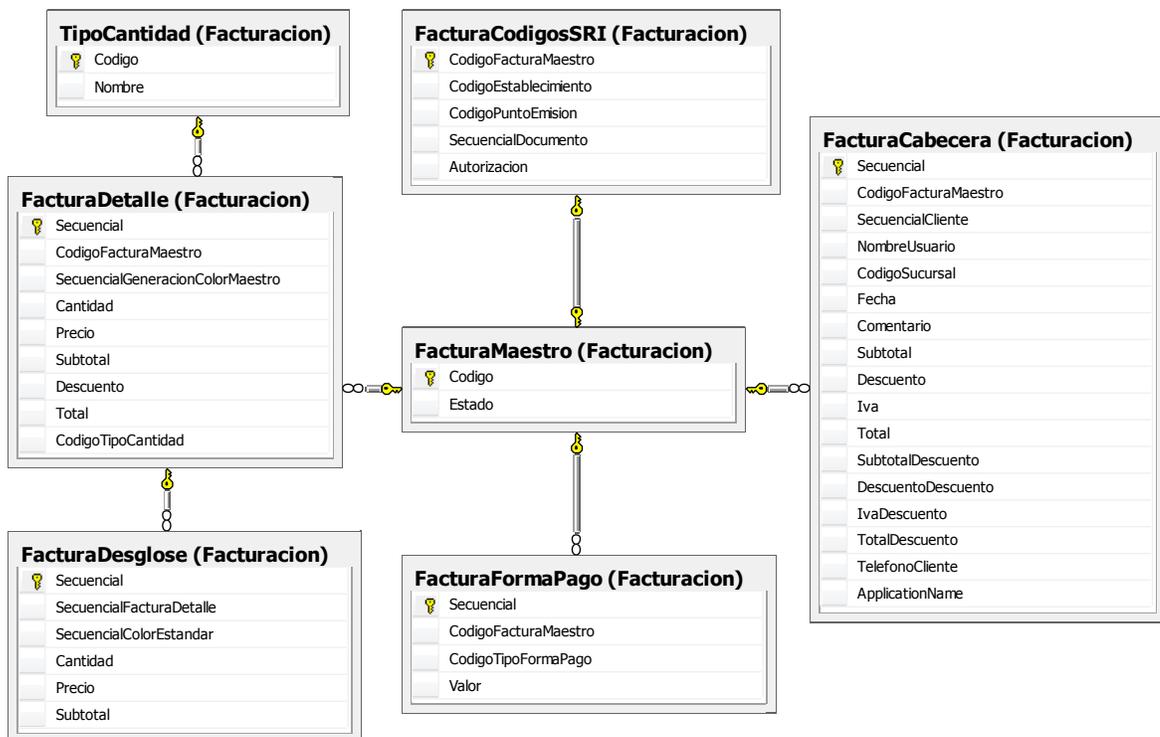


Figura 6.34 Diseño de la Base de Datos (Facturación)

#### Esquema: Compras

Ver Figura 6.35 Diseño de la Base de Datos (Compras)

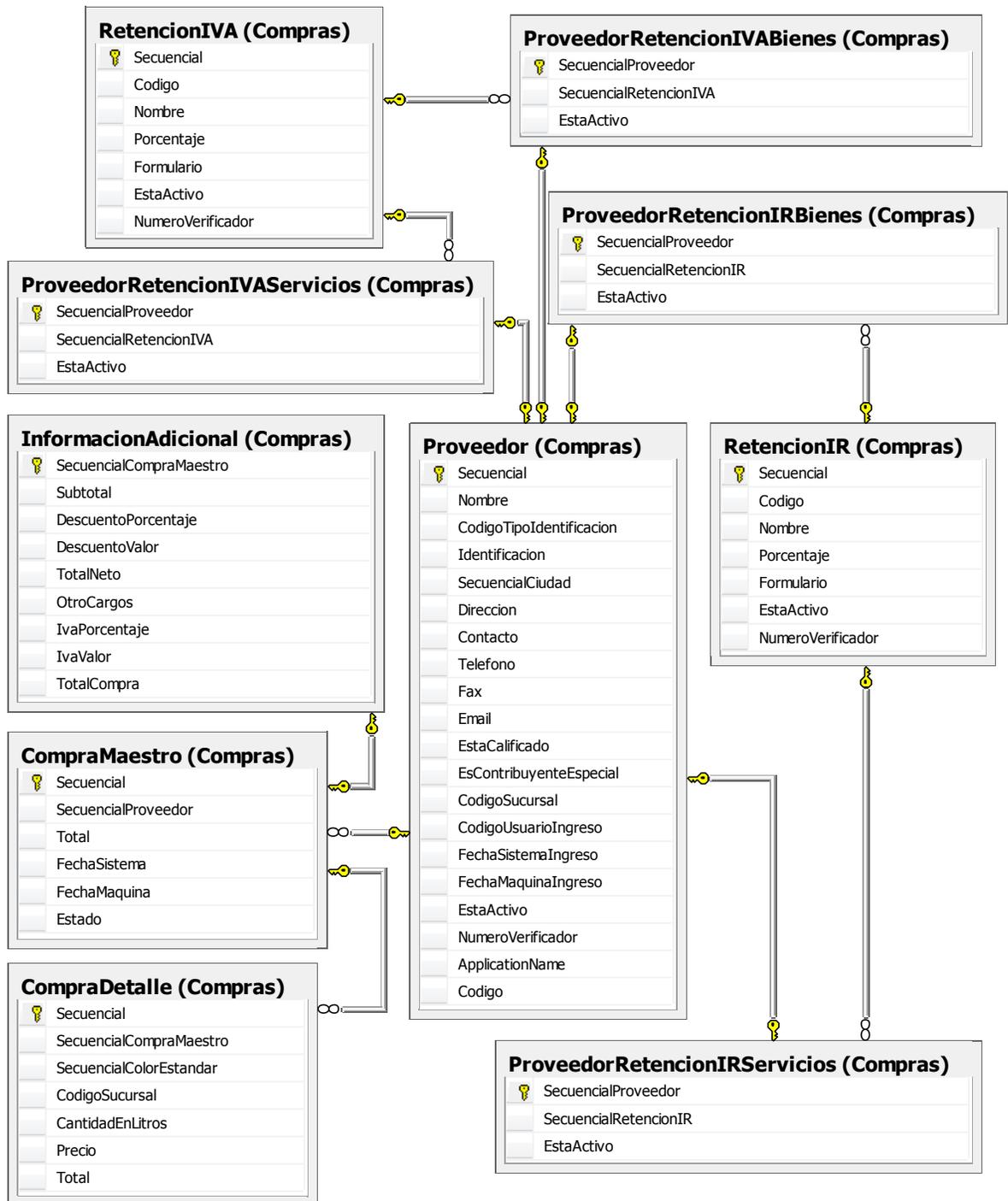


Figura 6.35 Diseño de la Base de Datos (Compras)

### Esquema: Clientes



Figura 6.36 Diseño de la Base de Datos (Clientes)

### Esquema: Colores

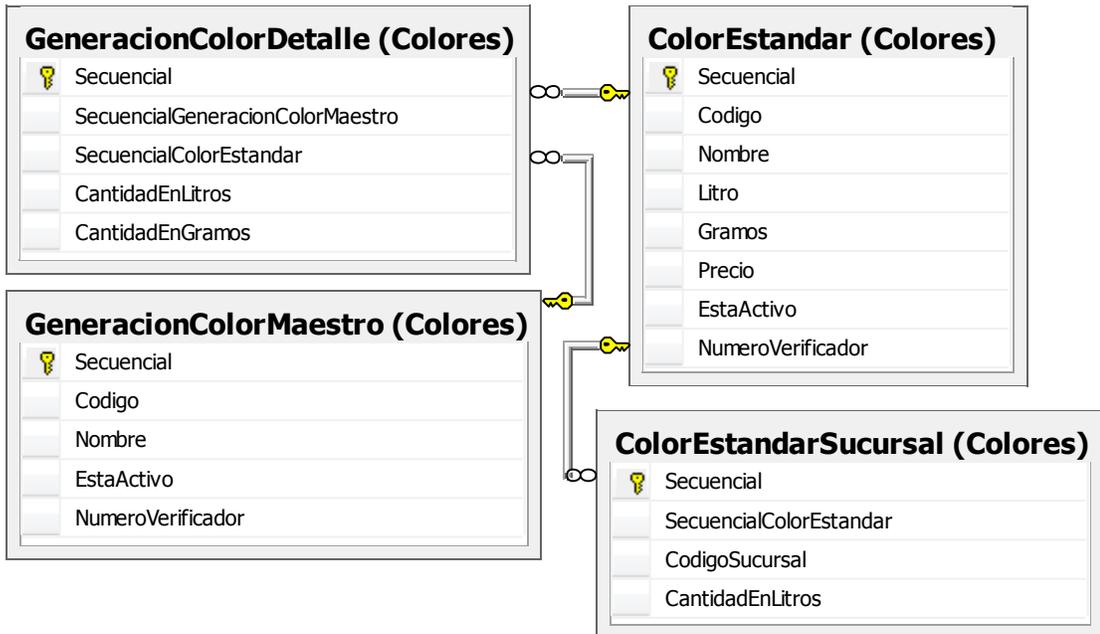


Figura 6.37 Diseño de la Base de Datos (Colores)

## Esquema: Créditos

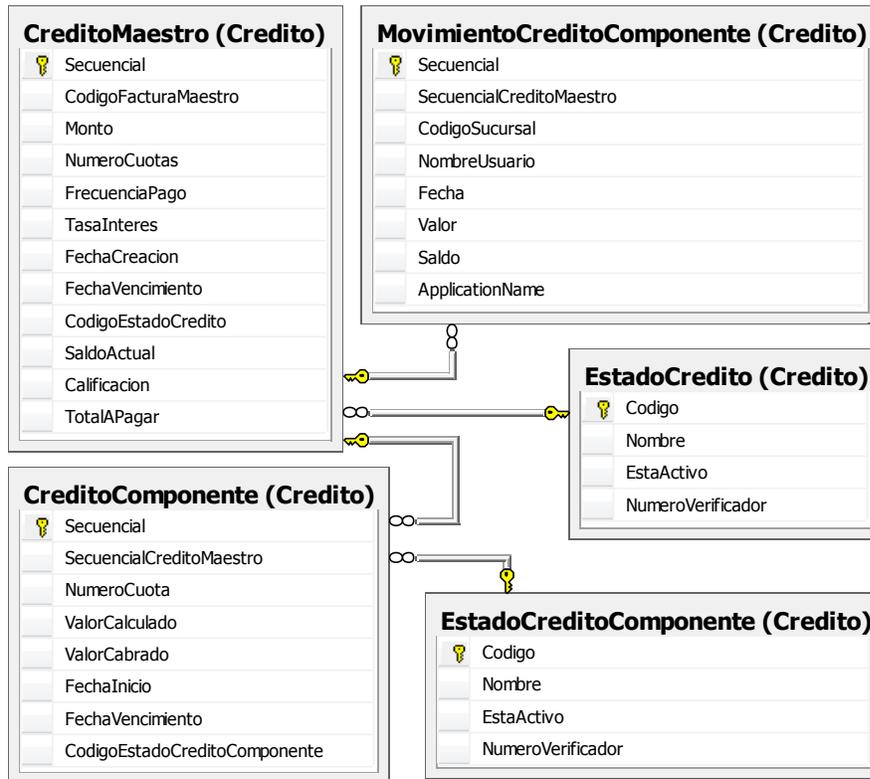


Figura 6.38 Diseño de la Base de Datos (Créditos)

## Esquema: Históricos



Figura 6.39 Diseño de la Base de Datos (Históricos)

## Esquema: Generales

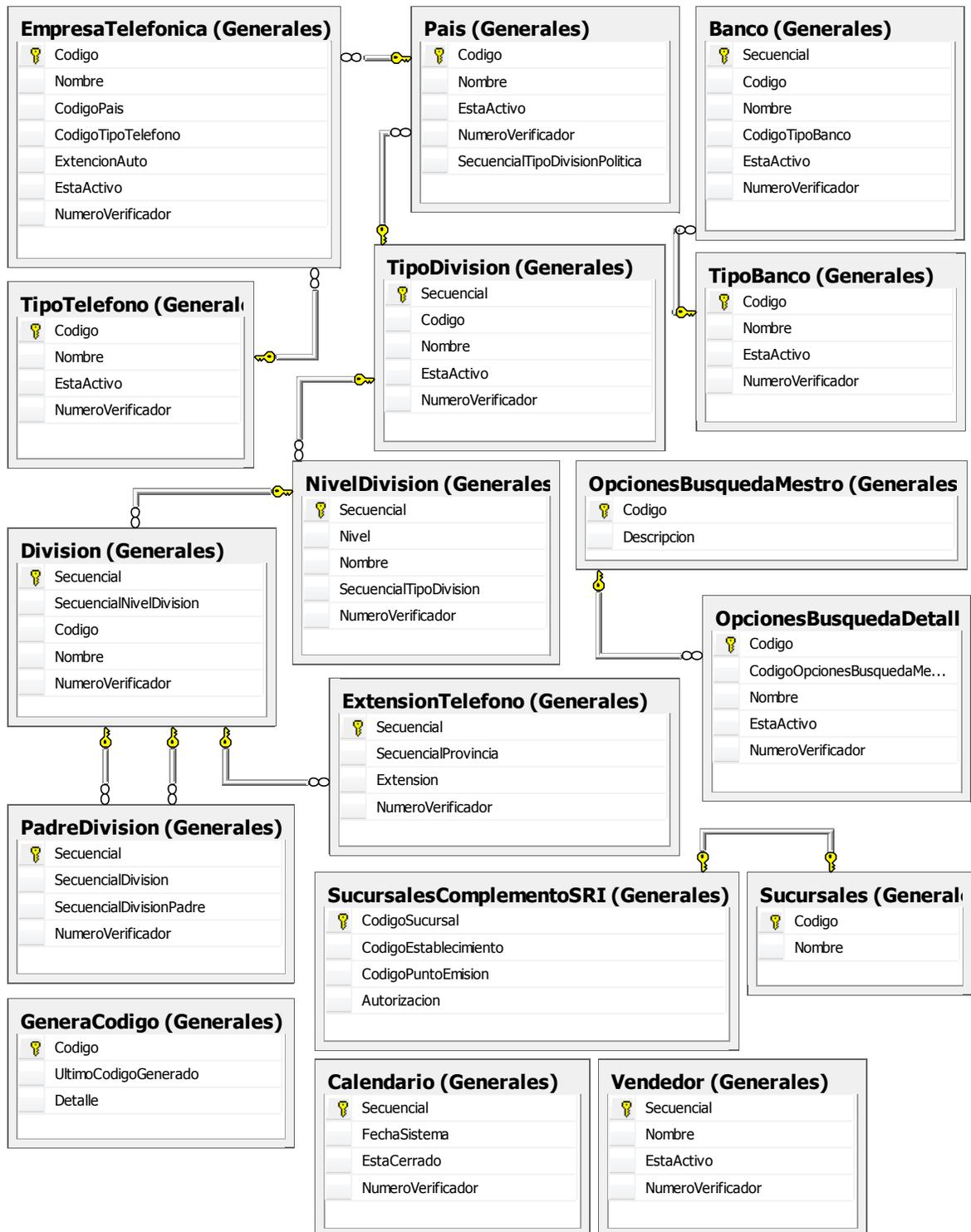


Figura 6.40 Diseño de la Base de Datos (Generales)

### 6.9.2.2 Diccionario de Datos

TABLA		
<b>Nombre:</b> "Facturacion"."FacturaMaestro"		
<b>Descripción:</b> Tabla principal para la factura		
Campo	Tipo	Observación
Codigo	character varying(12)	PK
Estado	character varying(20)	

**Tabla 6.1** "Facturacion"."FacturaMaestro"

TABLA		
<b>Nombre:</b> "Facturacion"."FacturaCabecera"		
<b>Descripción:</b> Guarda los datos de la cabecera de la factura		
Campo	Tipo	Observación
Secuencial	Integer	PK
CodigoFacturaMaestro	character varying(12)	FK "Facturacion"."FacturaMaestro"(Codigo)
SecuencialCliente	Integer	FK "Clientes"."Cliente" ("Secuencial")
NombreUsuario	character varying(255)	FK "Users" ("Username")
CodigoSucursal	character varying(10)	FK "Generales"."Sucursales" ("Codigo")
Fecha	Date	
Comentario	character varying(250)	
Subtotal	numeric(12,2)	
Descuento	numeric(12,2)	
Iva	numeric(12,2)	
Total	numeric(12,2)	
SubtotalDescuento	numeric(12,2)	
DescuentoDescuento	numeric(12,2)	
IvaDescuento	numeric(12,2)	
TotalDescuento	numeric(12,2)	
TelefonoCliente	character varying(40)	

**Tabla 6.2** "Facturacion"."FacturaCabecera"

TABLA		
<b>Nombre:</b> "Facturacion"."FacturaDetalle"		
<b>Descripción:</b> Guarda los datos del detalle de la factura		
Campo	Tipo	Observación
Secuencial	Integer	PK
CodigoFacturaMaestro	character varying(12)	FK "Facturacion"."FacturaMaestro" ("Codigo")
SecGeneColorMaestro	Integer	FK "Colores"."GeneraciColorMaestro" ("Secuencial")
Cantidad	numeric(12,2)	
Precio	numeric(12,2)	
Subtotal	numeric(12,2)	
Descuento	numeric(12,2)	
Total	numeric(12,2)	<i>continúa</i>

CodigoTipoCantidad	character varying(1)	FK "Facturacion"."TipoCantidad" ("Codigo")
--------------------	----------------------	--

**Tabla 6.3** "Facturacion"."FacturaDetalle"

TABLA		
<b>Nombre:</b> "Facturacion"."FacturaDesglose"		
<b>Descripción:</b> Guarda los datos de los colores que se ocupo en la factura		
Campo	Tipo	Observación
Secuencial	Integer	PK
SecuencialFacturaDetalle	Integer	FK Facturacion."FacturaDetalle" ("Secuencial")
SecuencialColorEstandar	Integer	FK "Colores"."ColorEstandar" ("Secuencial")
Cantidad	numeric(12,6)	
Precio	numeric(12,2)	
Subtotal	numeric(12,2)	

**Tabla 6.4** "Facturacion"."FacturaDesglose"

TABLA		
<b>Nombre:</b> "Facturacion"."FacturaCodigosSRI"		
<b>Descripción:</b> Guarda Información adicional de la factura		
Campo	Tipo	Observación
CodigoFacturaMaest	character varying(12)	PK, FK "Facturacion"."FacturaMaestro" ("Codigo")
CodigoEstablecimien	character varying(6)	
CodigoPuntoEmision	character varying(6)	
SecuencialDocument	character varying(14)	
Autorizacion	character varying(100)	

**Tabla 6.5** "Facturacion"."FacturaCodigosSRI"

TABLA		
<b>Nombre:</b> "Facturacion"."FacturaFormaPago"		
<b>Descripción:</b> Guarda las formas de pago que se realizó la factura		
Campo	Tipo	Observación
Secuencial	Integer	PK
CodigoFacturaMaestro	character varying(12)	FK "Facturacion"."FacturaMaestro" ("Codigo")
CodigoTipoFormaPago	character varying(5)	FK "Generales"."TipoFormaPago" ("Codigo")
Valor	numeric(12,2)	

**Tabla 6.6** "Facturacion"."FacturaFormaPago"

TABLA		
<b>Nombre:</b> "Compras"."CompraMaestro"		
<b>Descripción:</b> Guarda el encabezado de la compra		
Campo	Tipo	Observación
Secuencial	Integer	PK
SecuencialProveedor	Integer	FK "Compras"."Proveedor" ("Secuencial")
Total	numeric(12,2)	<i>continúa</i>

FechaSistema	Date	
FechaMaquina	Date	
Estado	character varying(20)	

**Tabla 6.7** "Compras"."CompraMaestro"

TABLA		
<b>Nombre:</b> "Compras"."CompraDetalle"		
<b>Descripción:</b> Guarda el detalle de la compra		
Campo	Tipo	Observación
Secuencial	Integer	PK
SecuencialCompraMaest	Integer	FK "Compras"."CompraMaestro" ("Secuencial")
SecuencialColorEstandar	Integer	FK "Colores"."ColorEstandar" ("Secuencial")
CodigoSucursal	character varying(10)	FK "Generales"."Sucursales" ("Codigo")
CantidadEnLitros	numeric(12,2)	
Precio	numeric(12,2)	
Total	numeric(12,2)	

**Tabla 6.8** "Compras"."CompraDetalle"

TABLA		
<b>Nombre:</b> "Clientes"."Cliente"		
<b>Descripción:</b> Guarda la información de los clientes		
Campo	Tipo	Observación
Secuencial	Integer	PK
SecuencialCiudad	Integer	FK "Generales"."Division" ("Secuencial")
Codigo	character varying(100)	
CodigoTipIdentifica	character varying(5)	FK "Clientes"."TipIdentificacionCliente" ("Codigo")
Identificacion	character varying(40)	UNIQUE
Nombre	character varying(100)	
Direccion	character varying(100)	
Email	character varying(100)	
SecuencialVendedor	Integer	FK "Generales"."Vendedor" ("Secuencial")
EstaActivo	Boolean	
NumeroVerificador	Integer	

**Tabla 6.9** "Clientes"."Cliente"

TABLA		
<b>Nombre:</b> "Clientes"."TelefonoCliente"		
<b>Descripción:</b> Guarda los números telefónicos del cliente		
Campo	Tipo	Observación
Secuencial	Integer	PK
SecuencialCliente	Integer	FK "Clientes"."Cliente" ("Secuencial")
CodigoEmpresaTelefonica	character varying(6)	FK "Generales"."EmpresaTelefonica" ("Codigo")
NumeroTelefono	character varying(40)	<i>continúa</i>

EstaActivo	Boolean	
NumeroVerificador	Integer	

**Tabla 6.10** "Clientes"."TelefonoCliente"

TABLA		
<b>Nombre:</b> "Clientes"."FormaDeVenta"		
<b>Descripción:</b> Guarda las formas de pago permitidas para cada cliente		
Campo	Tipo	Observación
Secuencial	Integer	PK
SecuencialCliente	Integer	FK "Clientes"."Cliente" ("Secuencial")
CodigoFormaDeVenta	character varying(5)	FK "Generales"."TipoFormaPago" ("Codigo")
EstaActivo	Boolean	
NumeroVerificador	Integer	

**Tabla 6.11** "Clientes"."FormaDeVenta"

TABLA		
<b>Nombre:</b> "Colores"."ColorEstandar"		
<b>Descripción:</b> Guarda la información de todos los colores		
Campo	Tipo	Observación
Secuencial	Integer	PK
Codigo	character varying(20)	
Nombre	character varying(100)	
Litro	Integer	
Gramos	numeric(12,2)	
Precio	numeric(12,2)	
EstaActivo	Boolean	
NumeroVerificador	Integer	

**Tabla 6.12** "Colores"."ColorEstandar"

TABLA		
<b>Nombre:</b> "Colores"."GeneracionColorMaestro"		
<b>Descripción:</b> Guarda los colores que se han generado		
Campo	Tipo	Observación
Secuencial	Integer	PK
Codigo	character varying(20)	UNIQUE
Nombre	character varying(100)	
EstaActivo	Boolean	
NumeroVerificador	Integer	

**Tabla 6.13** "Colores"."GeneracionColorMaestro"

TABLA		
<b>Nombre:</b> "Colores"."GeneracionColorDetalle"		
<b>Descripción:</b> Guarda la info de los colores que se utilizaron para generar un nuevo color <i>continúa</i>		

Campo	Tipo	Observación
Secuencial	Integer	PK
SecGeneracionColorMaestr	Integer	FK "Colores"."GeneracionColorMaestro" ("Secuencial")
SecuencialColorEstandar	Integer	FK "Colores"."ColorEstandar" ("Secuencial")
CantidadEnLitros	numeric(12,6)	
CantidadEnGramos	numeric(12,6)	

**Tabla 6.14 "Colores"."GeneracionColorDetalle"**

TABLA		
<b>Nombre:</b> "Colores"."ColorEstandarSucursal"		
<b>Descripción:</b> Guarda la relación entre los colores y las sucursales		
Campo	Tipo	Observación
Secuencial	Integer	PK
SecuencialColorEstandar	Integer	FK "Colores"."ColorEstandar" ("Secuencial")
CodigoSucursal	character varying(10)	FK "Generales"."Sucursales" ("Codigo")
CantidadEnLitros	numeric(12,6)	

**Tabla 6.15 "Colores"."ColorEstandarSucursal"**

TABLA		
<b>Nombre:</b> "Credito"."CreditoMaestro"		
<b>Descripción:</b> Guarda la información principal de un crédito		
Campo	Tipo	Observacion
Secuencial	Integer	PK
CodigoFacturaMaestro	character varying(12)	FK "Facturacion"."FacturaMaestro" ("Codigo")
Monto	numeric(12,2)	
NumeroCuotas	Integer	
FrecuenciaPago	Integer	
TasaInteres	numeric(6,2)	
FechaCreacion	Date	
FechaVencimiento	Date	
CodigoEstadoCredito	character varying(4)	FK "Credito"."EstadoCredito" ("Codigo")
SaldoActual	numeric(12,2)	
Calificacion	character varying(2)	
TotalAPagar	numeric(12,2)	

**Tabla 6.16 "Credito"."CreditoMaestro"**

TABLA		
<b>Nombre:</b> "Credito"."CreditoComponente"		
<b>Descripción:</b> Guarda los pagos que tendrá que realizar el cliente		
Campo	Tipo	Observación
Secuencial	Integer	PK
SecuencCreditoMaes	Integer	FK "Credito"."CreditoMaestro" ("Secuencial")
NumeroCuota	Integer	<i>continúa</i>

ValorCalculado	numeric(12,2)	
ValorCabrado	numeric(12,2)	
FechaInicio	Date	
FechaVencimiento	Date	
CodiEstadoCreditoCo	character varying(4)	FK "Credito"."EstadoCreditoComponente" ("Codigo")

**Tabla 6.17** "Credito"."CreditoComponente"

TABLA		
<b>Nombre:</b> "Credito"."MovimientoCreditoComponente"		
<b>Descripción:</b> Guarda los pagos que va realizando los clientes por los créditos obtenidos		
Campo	Tipo	Observación
Secuencial	Integer	PK
SecuencialCreditoMaest	Integer	FK "Credito"."CreditoMaestro" ("Secuencial")
CodigoSucursal	character varying(10)	FK "Generales"."Sucursales" ("Codigo")
NombreUsuario	character varying(255)	FK "Users" ("Username")
Fecha	Date	
Valor	numeric(12,2)	
Saldo	numeric(12,2)	

**Tabla 6.18** "Credito"."MovimientoCreditoComponente"

TABLA		
<b>Nombre:</b> "Historicos"."HistoricoColorEstandar"		
<b>Descripción:</b> Guarda los datos de histórico de los colores estándar		
Campo	Tipo	Observación
Secuencial	Integer	
Codigo	character varying(20)	
Nombre	character varying(100)	
Litro	Integer	
Gramos	numeric(12,2)	
Precio	numeric(12,2)	
EstaActivo	Boolean	
NumeroVerificador	Integer	
SecuenciHistoricosFecha	Integer	FK "Historicos"."HistoricoFecha" ("Secuencial")

**Tabla 6.19** "Historicos"."HistoricoColorEstandar"

TABLA		
<b>Nombre:</b> "Historicos"."HistoricoSucursales"		
<b>Descripción:</b> Guarda los datos de históricos de las sucursales		
Campo	Tipo	Observación
Codigo	character varying(10)	
Nombre	character varying(50)	
SecuencialHistoricosFech	Integer	FK "Historicos"."HistoricoFecha" ("Secuencial")

**Tabla 6.20** "Historicos"."HistoricoSucursales"

TABLA		
<b>Nombre:</b> "Historicos"."HistoricoColorEstandarSucursal"		
<b>Descripción:</b> Guarda los históricos de las relaciones entre colores estándar y sucursales		
Campo	Tipo	Observación
SecuencialColorEstandar	Integer	
CodigoSucursal	character varying(10)	
CantidadEnLitros	numeric(12,2)	
SecuencialHistoricosFecha	Integer	FK "Historicos"."HistoricoFecha" ("Secuencial")

**Tabla 6.21** "Historicos"."HistoricoColorEstandarSucursal"

TABLA		
<b>Nombre:</b> "Generales"."Sucursales"		
<b>Descripción:</b> Guarda las sucursales		
Campo	Tipo	Observación
Codigo	character varying(10)	PK
Nombre	character varying(50)	

**Tabla 6.22** "Generales"."Sucursales"

TABLA		
<b>Nombre:</b> "Generales"."Calendario"		
<b>Descripción:</b> Muestra la fecha en la que esta trabajando el sistema		
Campo	Tipo	Observación
Secuencial	Integer	PK
FechaSistema	Date	
EstaCerrado	Boolean	
NumeroVerificador	Integer	

**Tabla 6.23** "Generales"."Calendario"

TABLA		
<b>Nombre:</b> "Generales"."Vendedor"		
<b>Descripción:</b> Guarda información de los usuarios vendedores		
Campo	Tipo	Observación
Secuencial	Integer	PK
Nombre	character varying(100)	
EstaActivo	Boolean	
NumeroVerificador	Integer	

**Tabla 6.24** "Generales"."Vendedor"

### 6.9.2.3 Desarrollo del Sistema

El código fuente del presente sistema será desarrollado en C# mediante el uso de Visual Studio 2008 Express Edition.

#### Código Fuente

Archivo de configuración web.config.

```
Conexión a la Base de Datos
<connectionStrings>
<add name="conexion" connectionString="DATABASE=FactInvePPG;
SERVER=localhost; PORT=5432; User Id=postgres; Password=latino;
Encoding=UNICODE; Sslmode=Prefer; Pooling=true;"/>
</connectionStrings>
```

```
Administración de Roles
<membership defaultProvider="ClaseUsuarios">
<providers>
<add connectionStringName="conexion" applicationName="Sistema"
requiresUniqueEmail="true" enablePasswordRetrieval="false"
enablePasswordReset="true" requiresQuestionAndAnswer="true"
passwordFormat="Hashed" name="ClaseUsuarios"
type="PostgreSQL.Web.PostgreSQLMembershipProvider"/>
</providers>
</membership>

<roleManager enabled="true" cacheRolesInCookie="true"
cookieName=".AspNetRoles" defaultProvider="ClaseRoles">
<providers>
<clear/>
<add connectionStringName="conexion" applicationName="Sistema"
name="ClaseRoles" type="PostgreSQL.Web.PostgreSQLRoleProvider"/>
</providers>
</roleManager>
```

En vista que el código fuente escrito es demasiado grande, tomaremos una parte del código, en este caso se toma la parte que utilizamos para mezclar un color, para lo demás el código es similar, detallaremos la parte del Servidor como la parte del Cliente.

A continuación se extrae parte del código fuente de la parte del servidor (DAL).

```
Clase: ColoreEstandar.cs (DAL)
using System;
using System.Collections.Generic;
using System.Linq;
```

continúa

```

using System.Text;
using Npgsql;
using System.Data;

namespace VPCServidorDAL.Colores
{
    public class ColorEstandar
    {
        #region Datos
        private int _secuencial;
        private string _codigo;
        private string _nombre;
        private int _litro;
        private double _gramos;
        private double _precio;
        private bool _estaActivo;
        private int _numeroVerificador;
        #endregion

        #region Propiedades
        public int Secuencial
        {
            get { return this._secuencial; }
            set { this._secuencial = value; }
        }
        public stringCodigo
        {
            get { return this._codigo; }
            set { this._codigo = value; }
        }
        public string Nombre
        {
            get { return this._nombre; }
            set { this._nombre = value; }
        }
        public int Litro
        {
            get { return this._litro; }
            set { this._litro = value; }
        }
        public double Gramos
        {
            get { return this._gramos; }
            set { this._gramos = value; }
        }
        public double Precio
        {
            get { return this._precio; }
            set { this._precio = value; }
        }
        public bool EstaActivo
        {
            get { return this._estaActivo; }
            set { this._estaActivo = value; }
        }
        public int NumeroVerificador
        {
            get { return this._numeroVerificador; }
            set { this._numeroVerificador = value; }
        }
    }
}

```

*continúa*

```

#endregion

public DataTable SeleccionarPorSecuencial(int secuencial)
{
    NpgsqlConnection cnBaseDatos = new NpgsqlConnection();
    try
    {
        cnBaseDatos.ConnectionString = ConectarBD.CadenaDeConexion;
        string sentenciaSelect =
            "SELECT \"Secuencial\", \"Codigo\", \"Nombre\", \"Litro\", \"Gramos\",
            \"Precio\", \"EstaActivo\", \"NumeroVerificador\" " +
            "FROM \"Colores\".\"ColorEstandar\" " +
            "WHERE \"Secuencial\"=@secuencial ";

        NpgsqlCommand cmdSelect = new NpgsqlCommand(sentenciaSelect, cnBaseDatos);
        cmdSelect.CommandType = CommandType.Text;
        cmdSelect.Parameters.Add("@secuencial",
NpgsqlTypes.NpgsqlDbType.Integer).Value = secuencial;

        NpgsqlDataAdapter daColorEstandar = new NpgsqlDataAdapter();
        daColorEstandar.SelectCommand = cmdSelect;
        DataTable dtColorEstandar = new DataTable();
        daColorEstandar.Fill(dtColorEstandar);
        return dtColorEstandar;
    }
    catch (NpgsqlException errorPostgres)
    {
        throw errorPostgres;
    }
    catch (Exception errorCSharp)
    {
        throw errorCSharp;
    }
    finally
    {
        cnBaseDatos.Close(); //Cerrar la conexion.
        cnBaseDatos.Dispose(); //Liberar recursos.
    }
}

public bool Insertar(ColorEstandar colorEstandar)
{
    bool insertoRegistro = false;
    NpgsqlConnection cnBaseDatos = new NpgsqlConnection();
    try
    {
        cnBaseDatos.ConnectionString = ConectarBD.CadenaDeConexion;
        string sentenciaInsert =
            " INSERT INTO \"Colores\".\"ColorEstandar\"(\"Codigo\", \"Nombre\",
            \"Litro\", \"Gramos\", \"Precio\", \"EstaActivo\", \"NumeroVerificador\") " +
            " VALUES (@codigo, @nombre, @litro, @gramos, @precio, @estaActivo,
@numeroVerificador) " ;

        NpgsqlCommand cmdInsert = new NpgsqlCommand(sentenciaInsert, cnBaseDatos);
        cmdInsert.CommandType = CommandType.Text;
        cmdInsert.Parameters.Add("@codigo", NpgsqlTypes.NpgsqlDbType.Varchar,
20).Value = colorEstandar.Codigo;
        cmdInsert.Parameters.Add("@nombre", NpgsqlTypes.NpgsqlDbType.Varchar,
100).Value = colorEstandar.Nombre;
        cmdInsert.Parameters.Add("@litro", NpgsqlTypes.NpgsqlDbType.Integer).Value

```

continúa

```

= colorEstandar.Litro;
    cmdInsert.Parameters.Add("@gramos", NpgsqlTypes.NpgsqlDbType.Double).Value
= colorEstandar.Gramos;
    cmdInsert.Parameters.Add("@precio", NpgsqlTypes.NpgsqlDbType.Double).Value
= colorEstandar.Precio;
    cmdInsert.Parameters.Add("@estaActivo",
NpgsqlTypes.NpgsqlDbType.Boolean).Value = colorEstandar.EstaActivo;
    cmdInsert.Parameters.Add("@numeroVerificador",
NpgsqlTypes.NpgsqlDbType.Integer).Value = colorEstandar.NumeroVerificador;

    //Abrir la coneccion
    cnBaseDatos.Open();
    //Ejecutar el comando SQL contra la BD.
    int filasAfectadas = cmdInsert.ExecuteNonQuery();
    if (filasAfectadas == 1)
    {
        insertoRegistro = true;
    }
    return insertoRegistro;
}
catch (NpgsqlException errorPostgres)
{
    throw errorPostgres;
}
catch (Exception errorCSharp)
{
    throw errorCSharp;
}
finally
{
    cnBaseDatos.Close(); //Cerrar la conexion.
    cnBaseDatos.Dispose(); //Liberar recursos.
}
}

public bool Actualizar(ColorEstandar colorEstandar)
{
    bool actualizoRegistro = false;
    NpgsqlConnection cnBaseDatos = new NpgsqlConnection();
    try
    {
        cnBaseDatos.ConnectionString = ConectarBD.CadenaDeConexion;
        string sentenciaUpdate =
            " UPDATE \"Colores\".\"ColorEstandar\" " +
            " SET \"Codigo\"=@codigo, " +
            " \"Nombre\"=@nombre, " +
            " \"Litro\"=@litro, " +
            " \"Gramos\"=@gramos, " +
            " \"Precio\"=@precio, " +
            " \"EstaActivo\"=@estaActivo " +
            " WHERE \"Secuencial\"=@secuencial ";

        NpgsqlCommand cmdUpdate = new NpgsqlCommand(sentenciaUpdate, cnBaseDatos);
        cmdUpdate.CommandType = CommandType.Text;
        cmdUpdate.Parameters.Add("@secuencial",
NpgsqlTypes.NpgsqlDbType.Integer).Value = colorEstandar.Secuencial;
        cmdUpdate.Parameters.Add("@codigo", NpgsqlTypes.NpgsqlDbType.Varchar,
20).Value = colorEstandar.Codigo;
        cmdUpdate.Parameters.Add("@nombre", NpgsqlTypes.NpgsqlDbType.Varchar,
100).Value = colorEstandar.Nombre;

```

continúa



```

{
    ColorEstandar colorEstandar = new ColorEstandar();
    DataTable dtColorEstandar = new DataTable();
    dtColorEstandar =
colorEstandar.SeleccionarPorSecuencial(Convert.ToInt32(tbSecuencial.Text));
    if (dtColorEstandar.Rows.Count == 1)
    {
        tbSecuencial.Text = dtColorEstandar.Rows[0]["Secuencial"].ToString();
        tbCodigo.Text = dtColorEstandar.Rows[0]["Codigo"].ToString();
        tbNombre.Text = dtColorEstandar.Rows[0]["Nombre"].ToString();
        tbLitro.Text = dtColorEstandar.Rows[0]["Litro"].ToString();
        tbGramos.Text = dtColorEstandar.Rows[0]["Gramos"].ToString();
        tbPrecio.Text = dtColorEstandar.Rows[0]["Precio"].ToString();
        cbEstaActivo.Checked =
Convert.ToBoolean(dtColorEstandar.Rows[0]["EstaActivo"].ToString());
    }
}
catch (NpgsqlException errorPostgres)
{
    lblMensaje.Text = "Error Postgres, <br />" + errorPostgres.Message + "<br
/>" + errorPostgres.StackTrace;
}
catch (Exception errorCSharp)
{
    lblMensaje.Text = "Error CSharp, <br />" + errorCSharp.Message + "<br />" +
errorCSharp.StackTrace;
}
}
#endregion

#region Metodos DML
public void InsertarColorEstandar()
{
    try
    {
        ColorEstandar colorEstandar = new ColorEstandar();
        colorEstandar.Codigo = tbCodigo.Text;
        colorEstandar.Nombre = tbNombre.Text;
        colorEstandar.Litro = Convert.ToInt32(tbLitro.Text);
        colorEstandar.Gramos = Convert.ToDouble(tbGramos.Text);
        colorEstandar.Precio = Convert.ToDouble(tbPrecio.Text);
        colorEstandar.EstaActivo = cbEstaActivo.Checked;
        if (colorEstandar.Insertar(colorEstandar))
        {
            lblMensaje.Text = "Color Estandar Ingresado con Exito";
            Response.Redirect("VistaColorEstandar.aspx");
        }
        else
        {
            lblMensaje.Text = "Error al Ingresar Color Estandar";
        }
    }
}
catch (NpgsqlException errorPostgres)
{
    lblMensaje.Text = "Error Postgres, <br />" + errorPostgres.Message + "<br
/>" + errorPostgres.StackTrace;
}
catch (Exception errorCSharp)
{
    lblMensaje.Text = "Error CSharp, <br />" + errorCSharp.Message + "<br />" +

```

continúa

```

errorCSharp.StackTrace;
    }
}
public void ActualizarColorEstandar()
{
    try
    {
        ColorEstandar colorEstandar = new ColorEstandar();
        colorEstandar.Secuencial = Convert.ToInt32(tbSecuencial.Text);
        colorEstandar.Codigo = tbCodigo.Text;
        colorEstandar.Nombre = tbNombre.Text;
        colorEstandar.Litro = Convert.ToInt32(tbLitro.Text);
        colorEstandar.Gramos = Convert.ToDouble(tbGramos.Text);
        colorEstandar.Precio = Convert.ToDouble(tbPrecio.Text);
        colorEstandar.EstaActivo = cbEstaActivo.Checked;

        if (colorEstandar.Actualizar(colorEstandar))
        {
            lblMensaje.Text = "Color Estandar Actualizado con Exito";
        }
        else
        {
            lblMensaje.Text = "Error al Actualizar Color Estandar";
        }
    }
    catch (NpgsqlException errorPostgres)
    {
        lblMensaje.Text = "Error Postgres, <br />" + errorPostgres.Message + "<br />" + errorPostgres.StackTrace;
    }
    catch (Exception errorCSharp)
    {
        lblMensaje.Text = "Error CSharp, <br />" + errorCSharp.Message + "<br />" + errorCSharp.StackTrace;
    }
}
#endregion

protected void Page_Load(object sender, EventArgs e)
{
    try
    {
        lblMensaje.Text = "";

        if (!Page.IsPostBack)
        {
            if (ClaseEstatica.EsNuevoColorEstandar)
            {
                tbSecuencial.Text = "0";
                tbLitro.Text = "1";
                cbEstaActivo.Checked = true;
                cbEstaActivo.Enabled = false;
            }
            else //es edicion
            {
                tbSecuencial.Text = ClaseEstatica.SecuencialColorEstandar;
                ObtenerDatosColorEstandar();
            }
        }
    }
}

```

continúa

```

catch (NpgsqlException errorPostgres)
{
    lblMensaje.Text = "Error Postgres, <br />" + errorPostgres.Message + "<br
/>" + errorPostgres.StackTrace;
}
catch (Exception errorCSharp)
{
    lblMensaje.Text = "Error CSharp, <br />" + errorCSharp.Message + "<br />" +
errorCSharp.StackTrace;
}
}

protected void ibtnGuardar_Click(object sender, ImageClickEventArgs e)
{
    try
    {
        ColorEstandar colorEstandar = new ColorEstandar();
        DataTable dtColorEstandar = new DataTable();

        dtColorEstandar =
colorEstandar.SeleccionarPorSecuencial(Convert.ToInt32(tbSecuencial.Text));

        if (dtColorEstandar.Rows.Count == 1)
        {
            //Es edicion
            ActualizarColorEstandar();
        }
        else
        {
            //Es nuevo ingreso
            InsertarColorEstandar();
        }
    }
    catch (NpgsqlException errorPostgres)
    {
        lblMensaje.Text = "Error Postgres, <br />" + errorPostgres.Message + "<br
/>" + errorPostgres.StackTrace;
    }
    catch (Exception errorCSharp)
    {
        lblMensaje.Text = "Error CSharp, <br />" + errorCSharp.Message + "<br />" +
errorCSharp.StackTrace;
    }
}
}
}

```

## 6.9.3 Pruebas

### 6.9.3.1 Pruebas de Caja Blanca

Las pruebas de caja blanca se denominan a un tipo de pruebas de software que se realiza sobre las funciones internas de un módulo, para lo cual se diseñaron casos de prueba para examinar la lógica del programa y para evaluar la estructura del control del diseño procedimental, es decir:

- Se evaluaron los posibles caminos de ejecución independientes de cada uno de los módulos.
- Se evaluaron las decisiones lógicas-aritméticas.
- Se ejecutaron las estructuras de datos internos (definición-uso de variables)
- Se comprobaron todos los bucles.

### **6.9.3.2 Prueba de Caja Negra**

Se denomina a aquel elemento que es estudiado desde el punto de vista de las entradas que recibe y las salidas o respuestas que produce, sin tener en cuenta su funcionamiento interno. En otras palabras, de una caja negra nos interesará su forma de interactuar con el medio que le rodea entendiendo qué es lo que hace, pero sin dar importancia a cómo lo hace.

Esta prueba se realizó sobre la interfaz del sistema, por lo tanto estas son completamente indiferentes al comportamiento interno del mismo. Con estas pruebas se demostró que:

- El usuario ingresa al sistema de forma adecuada y sencilla.
- Los datos ingresan correctamente al sistema.
- Los datos de salida a la interfaz del usuario son los deseados.
- Las funciones del sistema son operativas y factibles.
- La integridad del sistema se mantiene.

Para lograr esta prueba se realizó entrada de datos obteniendo salidas deseadas en todos los módulos reiteradas veces.

### **6.9.3.3 Pruebas de Verificación y Validación**

Es el proceso de control que asegura que el software cumple con su especificación y satisface las necesidades del usuario. Muchas veces se confunde “verificación” con validación”. *Verificación:* ¿Estamos construyendo correctamente el producto? implica controlar que el producto conforma su especificación inicial. *Validación:*

¿Estamos construyendo el producto correcto? Se ocupa de controlar si el producto satisface los requerimientos del usuario.

**Verificación.-** Este proceso determino que el sistema está de acuerdo a las especificaciones iniciales y que el sistema concuerda y cumple con las especificaciones planteadas. Como:

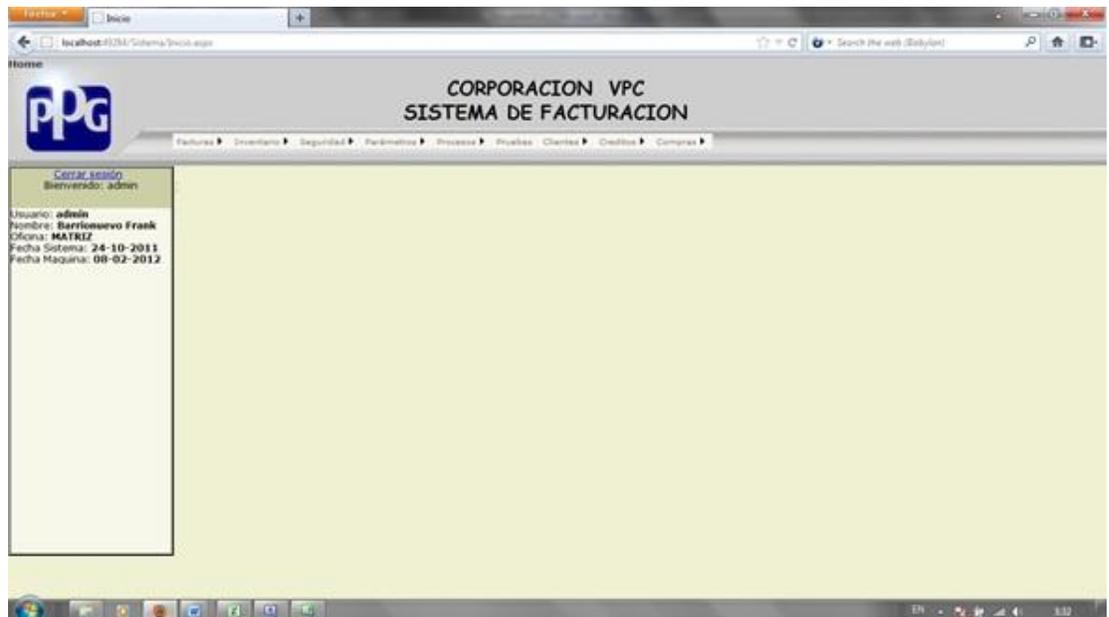
1. Inicio de Sesión, se valida tanto el nombre de usuario y la contraseña.



The image shows a login form with a blue header bar containing the text 'Iniciar sesión'. Below the header, there are two input fields: 'Nombre de usuario:' with the text 'admin' and 'Contraseña:' with a masked password of seven dots. Below these fields is a checkbox labeled 'Recordármelo la próxima vez.' and a button labeled 'Inicio de sesión'.

**Figura 6.41** Pruebas Inicio Sesión

2. Una vez pasada la validación de inicio de sesión, ingresa al sistema en la cual le aparece el menú de acuerdo a los permisos (Roles), junto con la información del usuario.



**Figura 6.42** Pruebas Pantalla principal

### 3. Creación de Usuarios.

Regístrese para obtener una nueva cuenta

Nombre de usuario: vero

Apellido: Barrionuevo

Nombre: Veronica

Contraseña: ●●●●

Confirmar contraseña: ●●●●

Correo electrónico: vero\_ajcb@yahoo.es

Pregunta de seguridad: color

Respuesta de seguridad: blanco

Sucursal: MATRIZ

Crear usuario

Figura 6.43 Pruebas Creación Usuario

Completar

La cuenta se ha creado correctamente.

Continuar

Figura 6.44 Pruebas Usuario creado correctamente

### 4. Dar permisos de acceso al usuario creado

Administrar Usuarios

	Username	Apellido	Nombre
<a href="#">Seleccionar</a>	usuario1	Caguana	Nahomy
<a href="#">Seleccionar</a>	usuario3	Barrionuevo	Erick
<a href="#">Seleccionar</a>	usuario2	Caguana	Anthony
<a href="#">Seleccionar</a>	vero	Barrionuevo	Veronica
<a href="#">Seleccionar</a>	admin	Barrionuevo	Frank
<a href="#">Seleccionar</a>	majo	Santana	Maria Jose

Usuario Seleccionado: vero

Editar Usuario   Eliminar Usuario   Editar Funciones

Editar Funciones

- funAdministrador
- funFacturacion
- funParametros
- funSeguridad
- funCuentasPorCobrar
- funInventario
- funClientes
- funCreditos
- funCompras

Aceptar   Cancelar

Figura 6.45 Pruebas Administración Usuarios

5. Actualización de un Color Estándar

**Color Estandar Actualizado con Exito**

Secuencial: 14

Codigo: DUL 603

Nombre: Clear Bicope

Litro: 1

Gramos: 980,00

Precio: 17,61

Esta Activo:

Figura 6.46 Pruebas Actualizando color estándar

6. Actualización de la mezcla de un color (Generación Color).

**GeneracionColorMaestro Actualizado con Exito**

Secuencial: 7

Codigo: BS 2

Nombre: Blanco Saint2

Esta Activo:

	Secuencial	SecColor	Codigo	Nombre	CantidadEnLitros	CantidadEnGramos
Seleccionar	23	14	DUL 603	Clear Bicope	0,458469	449,300000
Seleccionar	24	15	DUL 637	Branco Puro	0,537607	629,000000
Seleccionar	25	16	DUL 690	Liso Matalico Azul	0,001275	1,300000
Seleccionar	26	17	DUL 692	Liso Matalico Preto	0,002673	2,700000
<b>Total Litros: 1,000024</b>					<b>Total Gramos: 1082,3</b>	

Figura 6.47 Pruebas Generar color maestro

7. Ingreso de una nueva compra.

**CompraMaestro Ingresado con Exito**

Secuencial: 10

Proveedor: 1803505963001  Ident. >> VER PAINT CORP

Fecha: 24-10-2011

Estado: Ingresando

Total: 0

**Detalle**

Secuencial: 0 Cantidad(L):

Color Estandar: D753 - Base Coatwhile Costo:

Sucursal: MATRIZ Total: 0

Figura 6.48 Pruebas Creación de Compra

**Información Adicional ingresada con éxito**

Secuencial: 10  
 Proveedor: 1803505963001  Ident. >>  
 Fecha: 24-10-2011  
 Estado: Ingresado  
 Total: **443,64**

**Detalle**

Secuencial: 0 Cantidad(L):  
 Color Estandar: D753 - Base Coat white Costo:  
 Sucursal: MATRIZ Total:

Seleccionar	Sec	SecColor	CodColor	Color	CodSucursal	Sucursal	CantidadEnLitros	Precio	Total
	16	14	DUL 603	Clear Bicope	5001	MATRIZ	12,00	36,97	443,64

Calculos Adicionales **Sub Total:** 443,64  
 **Descuento:** 0 % 0  
**Total Neto:** 443,64  
**Otros Cargos:** 0  
**Iva:** 12%  - 0%  0  
**Total Compra:** **443,64**

**Figura 6.49** Pruebas Ingreso Información Adicional de Compra

8. Afectacion del Inventario por la compra

Antes de procesar la compra

DUL 603	Clear Bicope	1017,008122
---------	--------------	-------------

**Figura 6.50** Pruebas Inventario antes de la Compra

Después de procesar la compra, como podemos ver se incremento los 12 litros del color DUL 603.

DUL 603	Clear Bicope	1029,008122
---------	--------------	-------------

**Figura 6.51** Pruebas Inventario después de la Compra

9. Creación de una nueva Factura

**FACTURA**  
 1891732593001  
 001-001  
 0000005  
 1108267976

Código: VEROS Fecha: 24-10-2011  
 Identificación: 1803505948 Dirección: Centro  
 Nombre: Veronica Barrionuevo Teléfono: 03458723

**DETALLE**

Codigo: BS 2 Nombre: Blanco Saint Cantidad: 500 L Descuento: 0 %  
 Comentario: Subtotal: 0 0  
 Dto: % 0 0  
 Iva 12%: 0 0  
 Total: 0 **0**

**FORMA PAGO**

Cheque  
 Credito  
 Efectivo

**DESGLOSE**  
 Subtotal: 13789,41

**Figura 6.52** Pruebas Inicio Factura

Código:	VEROS	Fecha:	24-10-2011	<b>FACTURA</b>	
Identificación:	1803505948	Dirección:	Centro	1891732593001	
Nombre:	Veronica Barrionuevo	Teléfono:	03458723	001-001	
				0000005	
				1108267976	

<b>DETALLE</b>								
Código:	Nombre:	Cantidad:	Descuento:					
BS 2	Blanco Saint	500	0 %					
Cantidad	TipoCant	Codigo	Nombre	Precio	Subtotal	Descuento	Total	Secuencial
500,00	G	BS 2	Blanco Saint 2	0,03	12,74	0,00	12,74	97
Comentario:					Subtotal:	12,74	12,74	
					Dto: %	0	0	
					Iva 12%:	1,53	1,53	
					Total:	14,27	14,27	

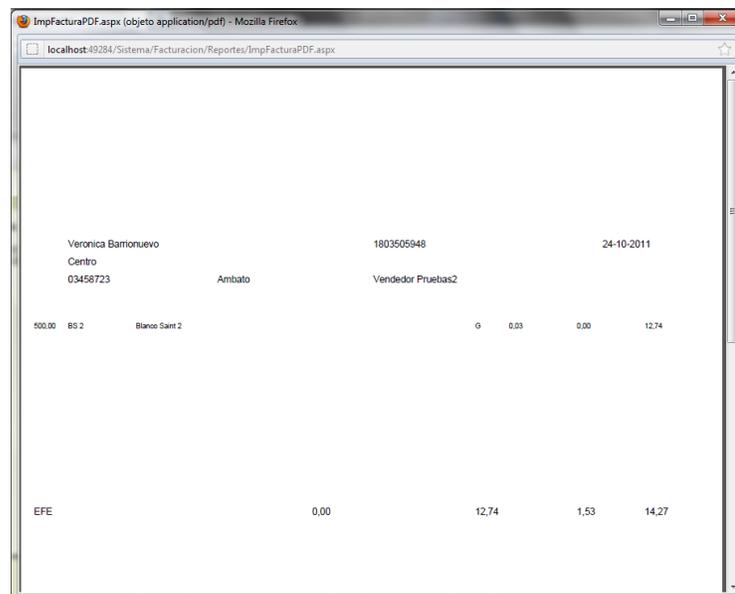
  

<b>DESGLOSE</b>				
Codigo	Nombre	Cantidad	Precio	Subtotal
DUL 603	Clear Bicope	207,567218	0,02	3,73
DUL 637	Branco Puro	290,584866	0,03	8,94
DUL 690	Liso Matalico Azul	0,600573	0,04	0,02
DUL 692	Liso Matalico Preto	1,247344	0,04	0,05
Subtotal:				12,74

<b>FORMA PAGO</b>	
<input type="radio"/> Cheque	Valor: 14,27
<input type="radio"/> Credito	
<input checked="" type="radio"/> Efectivo	

**Figura 6.53** Pruebas Creación Factura



**Figura 6.54** Pruebas Impresión Factura

10. Afectación al inventario por la creación de la Factura.

Antes de realizar la Factura.

DUL 603	Clear Bicope	1017,217925
DUL 637	Branco Puro	1432,155867
DUL 690	Liso Matalico Azul	16,997547
DUL 692	Liso Matalico Preto	13,744857

**Figura 6.55** Pruebas Inventario antes de la Factura

Después de crear la Factura, como podemos ver el inventario se ha disminuido la cantidad detallada en el desglose de la Factura.

DUL 803	Clear Bicope	1017,008122
DUL 837	Branco Puro	1431,907304
DUL 890	Liso Matalico Azul	16,998958
DUL 892	Liso Matalico Preto	13,743622

**Figura 6.56** Pruebas Inventario después de la Factura

**Validación:** Con esta prueba se verificó que el sistema satisface los requerimientos del usuario, para lo cual se realizó pruebas alfa y beta, *alfa* realizada por el usuario y con el desarrollador como observador; *beta* realizada por el usuario y sin observadores.

#### **6.9.4 Implantación del Sistema**

Luego de que el Sistema de Facturación e Inventarios para el control tributario de compra y venta en la Corporación VPC pasó satisfactoriamnte las pruebas se procedió a su implantación.

##### **Primer Paso: Base de Datos**

Instalar el motor de base de datos, crear la base de datos junto a las tablas que necesita el Sistema Informático para su funcionamiento.

##### **Segundo Paso: Instalación del Sistema en la Institución**

Se procedió a la publicación de las fuentes del sistema en el servidor proporcionado por la institución realizando primero la instalación del Framework 3.5 necesario para el funcionamiento del sistema.

Mediante los pasos anteriormente mencionados se concluyó que el método utilizado fue la Implantación Directa.

#### 6.9.4.1 Cronograma de Implantación del Sistema.

Nombre de Tarea	Duración
<b>Implantación</b>	
Instalación de las Bases de datos	1 Días
Creación de las tablas en la Base de Datos	1 Días
Ejecución de scripts de inicialización de la Base	1 Días
Verificación de operaciones de la BD	1 Días
Acceso al sistema, e ingreso de datos a la base	1 Días
Verificación de todos los módulos del sistema	3 Días
<b>Pruebas</b>	
Pruebas de caja negra, blanca y validación	15 Días
<b>Capacitación</b>	
Capacitación al funcionario encargado del sistema	2 Días

**Tabla 6.25** Cronograma de Implantación del Sistema

#### 6.9.4.2 Levantamiento del Sistema

Para la implantación del Sistema, se requiere de la instalación de:

- Instalación del Motor de Base de Datos PostgreSQL.
- Microsoft net framework (componente integral de Windows que admite la compilación y la ejecución de aplicaciones y servicios Web)
- Instalación del Navegador Web Firefox.
- Publicación del Sistema utilizando IIS.

Finalmente el sistema estará listo, se puede empezar con la utilización de un navegador Web, de preferencia Firefox.

### 6.10 Conclusiones y Recomendaciones

#### 6.10.1 Conclusiones

- Se dio solución al problema de no contar con un sistema adecuado para llevar a cabo las ventas y emitir facturas de las mismas, además de almacenar toda la información en una base centralizada.

- Se logro unificar el proceso de mezcla de colores y el de facturación de la misma en un solo sistema, a la vez realizando procesos adicionales como la generación de facturas a crédito y almacenamiento de información del SRI.
- Se tiene un mejor control del inventario a la vez se tiene un histórico de inventarios diarios, pudiendo ser consultado en cualquier momento obteniendo datos exactos en un tiempo oportuno.
- En cuanto al desarrollo Web en ASP.NET fue muy interesante ya que es una herramienta muy poderosa y que está en pleno avance tecnológico que ayuda a los programadores a desarrollar cualquier tipo de aplicaciones de una manera entendible. Y ahora gracias a Microsoft que libero la versión de Visual Studio en versiones express, cualquier programador tiene acceso a la herramienta y desarrollar cualquier aplicación.
- En cuanto a la base de datos POSTGRES, es una herramienta Open Source, capaz de administrar toda la información proporcionada por el sistema, brindando seguridad y fiabilidad al momento de almacenar y recuperar los datos.
- El sistema está desarrollado con una interfaz amigable de fácil utilización, permitiendo a los usuarios navegar por el sistema y realizar sus actividades sin ningún problema.

### **6.10.2 Recomendaciones**

- Se recomienda capacitar a los usuarios encargados de la utilización del sistema para que puedan utilizar de una manera óptima, de esta manera se evitara posibles inconvenientes.
- El sistema cuenta con un manual, por lo que se recomienda acudir a él en caso de no estar seguro del proceso que se quiere realizar, el manual ayudara a que los procesos realizados en el sistema sean los correctos.
- Ingresar datos coherentes por la interfaz del sistema, para que cuando se recupere los mismos tener la idea clara de lo que se guardo, evitar en lo posible guardar datos basura.

- Contar con una persona que tenga conocimientos básicos de base de datos para que pueda llevar la administración de la misma, quien deberá respaldar la base de datos diariamente para poder salvaguardar la información de la empresa por cualquier inconveniente que pudiese suceder.
- Para mayor seguridad respaldar la base en algún medio de almacenamiento que este fuera del área donde se encuentra el servidor de base de datos.
- Para el adecuado funcionamiento del sistema de preferencia utilizar navegadores Firefox, ya que se ocupó este navegador en el desarrollo de proyecto.

## **6.11 Bibliografía**

### **6.11.1 Información Bibliográfica de Libros**

- PAZMAY, Galo (2004) *Guía práctica para la elaboración de tesis y trabajos de investigación.*
- GONZALES, Argenis *Comparación entre sistemas de gestión de bases de datos (SGBD) bajo licenciamiento libre y comercial*
- MEDINA, Washington (2008) *Guía para el desarrollo de trabajos de graduación.*

### **6.11.2 Información Bibliográfica Web**

- Tipos de datos – Constantes – Variables – Operadores. (s.f.) Recuperado el 08 de Febrero del 2012 desde [http://es.wikipedia.org/wiki/C\\_Sharp](http://es.wikipedia.org/wiki/C_Sharp)
- González, J. (2006, Octubre 03). Características de C#. Recuperado el 08 de Febrero del 2012 desde <http://www.devjoker.com/contenidos/Tutorial-C/125/Introduccion-a-C.aspx>
- ¿Qué es CSS?. (s.f.) Recuperado el 08 de Febrero del 2012 desde <http://www.librosweb.es/css/capitulo1.html>
- ¿Qué es JavaScript?. (s.f.) Recuperado el 08 de Febrero del 2012 desde <http://www.librosweb.es/javascript/capitulo1.html>

- Internet Information Server. (s.f.) Recuperado el 08 de Febrero del 2012 desde <http://www.adrformacion.com/curso/intranet/leccion1/InternetInformationServer.htm>
- Alvarez, M. (2003, Febrero 26). Breve descripción de la forma de abrir ventanas secundarias con el lenguaje Javascript. Recuperado el 08 de Febrero del 2012 desde <http://www.desarrolloweb.com/articulos/1075.php>
- Alvarez, M. (2005, Enero 05). Validar un formulario en una página web, en el lado del cliente con Javascript. Recuperado el 08 de Febrero del 2012 desde <http://www.desarrolloweb.com/articulos/1767.php>
- Introducción a CSS. (s.f.) Recuperado el 08 de Febrero del 2012 desde <http://www.librosweb.es/css/>
- Manual CSS. (s.f.) Recuperado el 08 de Febrero del 2012 desde <http://www.webestilo.com/css/>
- Ojeda, J. (2011, Febrero 11). Creando Menús Dinámicos con ASP.NET. Recuperado el 08 de Febrero del 2012 desde <http://jvsoft.wordpress.com/2011/02/11/creando-menus-dinamicos-con-asp-net/>
- Cómo funciona la paginación en el control GridView. (s.f.) Recuperado el 08 de Febrero del 2012 desde <http://msdn.microsoft.com/es-es/library/5aw1xfh3%28v=vs.80%29.aspx>
- Martinez, R. (2009, Septiembre 09). Versiones de PostgreSQL disponibles. Recuperado el 08 de Febrero del 2012 desde <http://www.postgresql.org.es/node/343>
- Últimas versiones de PostgreSQL. (s.f.) Recuperado el 08 de Febrero del 2012 desde <http://www.postgresql.org.es/aggregator/sources/1>
- Martinez, R. (2010, Octubre 02). Sobre PostgreSQL. Recuperado el 08 de Febrero del 2012 desde [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql)
- Wikipedia enciclopedia libre. PostgreSQL. (s.f.) Recuperado el 08 de Febrero del 2012 desde <http://es.wikipedia.org/wiki/PostgreSQL>

- Microsoft Visual Studio. (s.f.) Recuperado el 08 de Febrero del 2012 desde [http://es.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](http://es.wikipedia.org/wiki/Microsoft_Visual_Studio)
- Julio, J. (2008, Noviembre 08). El rincón del .NET. Recuperado el 08 de Febrero del 2012 desde <http://jjdotnet.blogspot.com/>

## 6.12 Glosario de Términos

**ASP.NET:** Es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios web XML.

**C#.-** Es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET

**JavaScript:** Es un lenguaje de programación utilizado principalmente de lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas

**CSS: (*Cascading Style Sheets*)** Es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML.

**IIS: (*Internet Information Server*)** Es un servidor web y un conjunto de servicios para el sistema operativo Microsoft Windows.

**Postgres:** Es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD.

**Interfaz:** Es el medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo.

**Usuario:** Es la persona que utiliza o trabaja con algún objeto, la cual tendrá un conjunto de permisos para su acceso.

**UML (*Unified Modeling Language*):** Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.

**Diagramas de Caso de Uso:** Es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso. Los personajes o entidades que participarán en un caso de uso se denominan actores.

**Diagramas de Secuencia:** Muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso.

**Diagramas de Clases:** Describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos.

**Visual Studio Express:** Es un conjunto de programas gratuitos en entornos de desarrollo integrado (IDE) desarrollado por Microsoft.

**Visual Web Developer 2008 Express:** Software de desarrollo web, su función principal es la creación de páginas ASP.NET.

**Script:** Es un programa usualmente simple, que por lo regular se almacena en un archivo de texto plano.

**Framework:** Es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

**HTML: (HyperText Markup Language)** Lenguaje de Marcación de Hipertexto, es un lenguaje se utiliza comúnmente para establecer la estructura y contenido de un sitio web, tanto de texto, objetos e imágenes.

**OPEN SOURCE:** Código abierto es el término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de compartir el código que a las cuestiones éticas y morales las cuales destacan en el llamado software libre.

# ANEXOS

**ANEXO 1: MODELO DE LA ENTREVISTA**

**UNIVERSIDAD TECNICA DE AMBATO**

**F.I.S.E.I**

Entrevista enfocada al recurso humano de la Corporación VPC.

**Nota:** La información que se recopila es para uso exclusivo del trabajo de investigación a efectuarse en la Corporación VPC por lo tanto se recomienda que la información sea verídica.

**Cuestionario**

1. ¿El sistema se desarrollará en un lenguaje libre o propietario?

.....

2. ¿Cuál es la ventaja de utilizar software libre o propietario?

.....

3. ¿El sistema se desarrollara con un motor de base de datos libre o propietario?

.....

4. ¿Cuál es la ventaja de utilizar un motor de base de datos libre o propietario?

.....

5. ¿Es necesaria la implantación de un sistema informático en la empresa?

.....

6. ¿Cómo influye un sistema automatizado en la empresa?

.....

7. ¿Cómo maneja la información de la empresa actualmente?

.....

8. ¿Cómo se manejara la información de la empresa con la implantación de un nuevo sistema?

.....

9. ¿Tienen algún inconveniente en la venta cuando se realiza el proceso de mezcla de colores?

SI  NO  Porqué? .....

10. ¿En el proceso de facturación se realiza descuentos?

SI  NO  Porqué? .....

11. ¿El sistema guarda la secuencia de cada factura proporcionada por el SRI?

SI  NO  Porqué? .....

12. ¿Se tiene un óptimo control de los productos que son vendidos a crédito?

SI  NO  Porqué? .....

13. ¿Tiene un adecuado control sobre inventarios?

SI  NO  Porqué? .....

14. ¿Pueden disponer del inventario en tiempo real?

SI  NO  Porqué? .....

15. ¿Se puede obtener el estado del inventario de una fecha anterior?

SI  NO  Porqué? .....

16. ¿Tienen un óptimo control de compras?

SI  NO  Porqué? .....

17. ¿Tienen un adecuado control de los clientes?

SI  NO  Porqué? .....

18. ¿Los usuarios tienen acceso total sobre el sistema?

SI  NO  Porqué? .....

## ANEXO 2: MANUAL DE INSTALACIÓN DE SOFTWARE

### REQUERIDO

#### Anexo 2.1: Manual de Instalación de PostgreSQL.

1. En caso de no tener el instalador podemos descargar de la siguiente pagina <http://www.enterprisedb.com/products-services-training/pgdownload#windows>

Seleccionamos la versión que se desee, en nuestro caso para Windows.

#### Download PostgreSQL

*Please Note: Cookies should be enabled for the download process to function correctly*

Installer version *Version 9.1.2-1*



Installer version *Version 9.0.6-1*



Installer version *Version 8.4.10-1*



Installer version *Version 8.3.17-1*



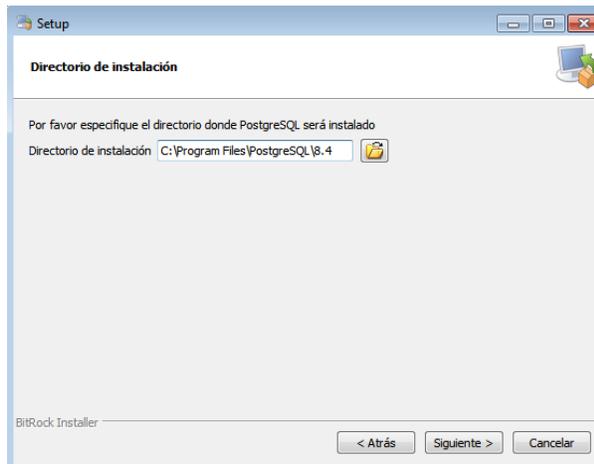
**Figura A2.1** Versiones de PostgreSQL

2. Una vez contemos con el archivo.exe, pulsar doble para iniciar con la instalación.
3. Aparecera el asistente para la instalación.



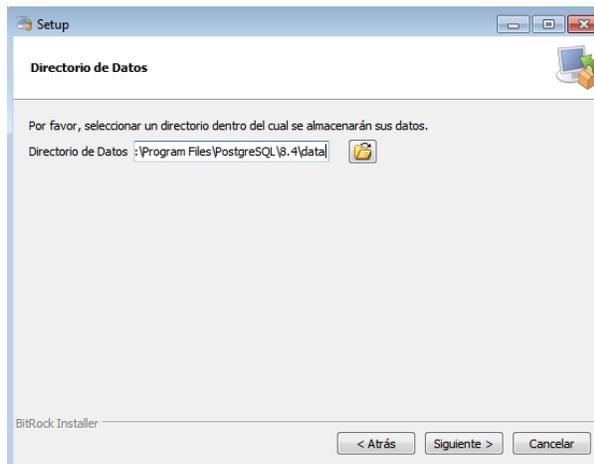
**Figura A2.2** Asistente instalación de PostgreSQL

4. Especificamos el directorio donde se instalara la base de datos.



**Figura A2.3** Directorio de instalación PostgreSQL

5. Especificamos el directorio donde serán almacenados nuestros datos.



**Figura A2.4** Directorio de Datos PostgreSQL

- Ingresamos una contraseña la cual nos servirá para poder acceder a la base de datos una vez instalada.



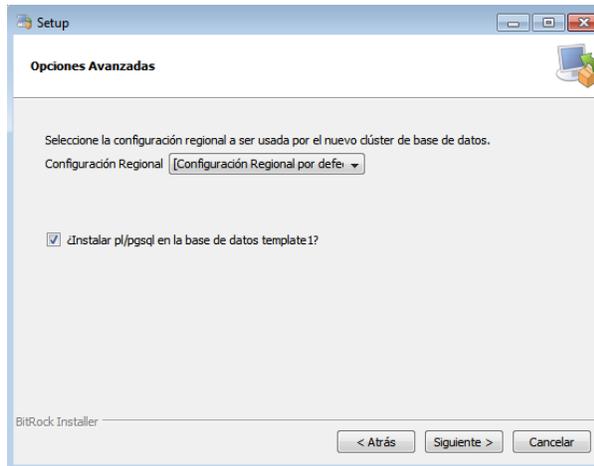
**Figura A2.5** Clave y contraseña PostgreSQL

- Seleccionamos el puerto por la cual el servidor escuchara, el puerto por defecto es 5432.



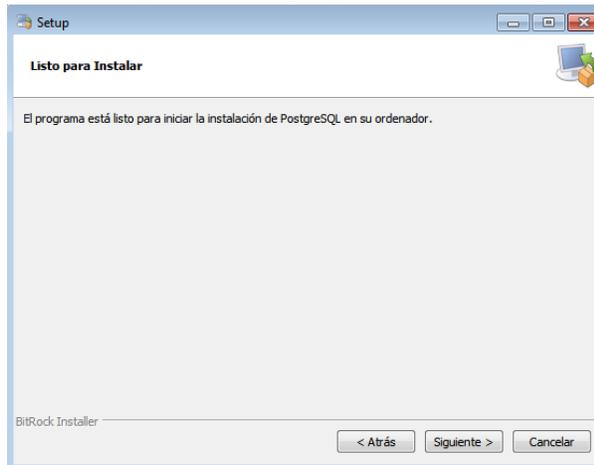
**Figura A2.6** Puerto PostgreSQL

- Seleccionamos la configuración Regional



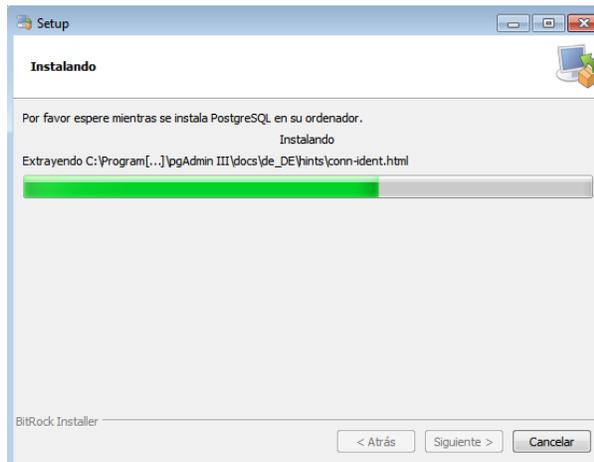
**Figura A2.7** Configuración Regional PostgreSQL

9. Una vez realizada la configuración el programa esta listo para ser instalado.



**Figura A2.8** Inicio de instalación PostgreSQL

10. Proceso de instalación



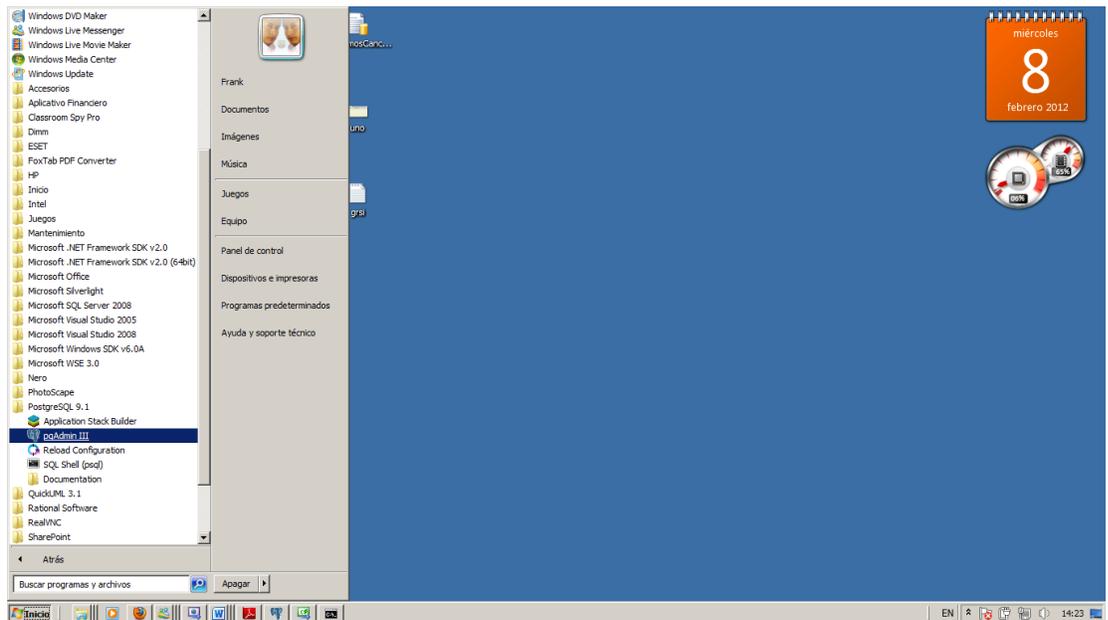
**Figura A2.9** Proceso de instalación PostgreSQL

## 11. Finalización de la instalación.



**Figura A2.10** Termino de instalación PostgreSQL

## 12. Una vez terminada la instalación verificamos si esta instalada correctamente.



**Figura A2.11** PostgreSQL instalado

## **Anexo 2.2 Manual de instalación de Microsoft Visual Studio 2008 Express**

Es el paquete sin costo del Entorno de Desarrollo Integrado de Microsoft, que contiene versiones ligeras, pero a la vez productivas en el tema de desarrollo de software.

Visual Studio Express 2008 contiene los siguientes productos:

- Visual Basic 2008 Express Edition
- Visual C# 2008 Express Edition
- Visual C++ 2008 Express Edition
- Visual Web Developer 2008 Express Edition.

Estas versiones incluyen:

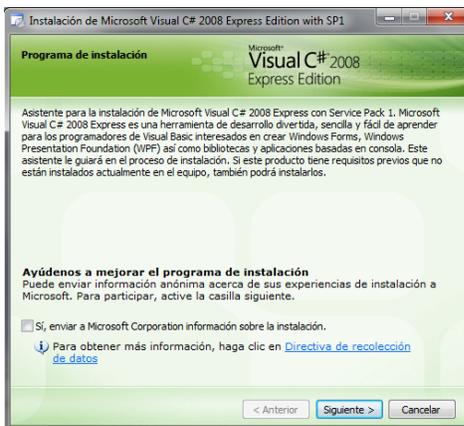
- Intellisense, corrigiendo errores de sintaxis y sugiriendo alternativas en el instante que se codifica.
- Soporte para Windows Presentation Foundation (solo version VB 2008).
- Depuración en tiempo de ejecución.
- Publicación mediante ClickOnce.

### **Instalación de Visual C# 2008 Express Edition**

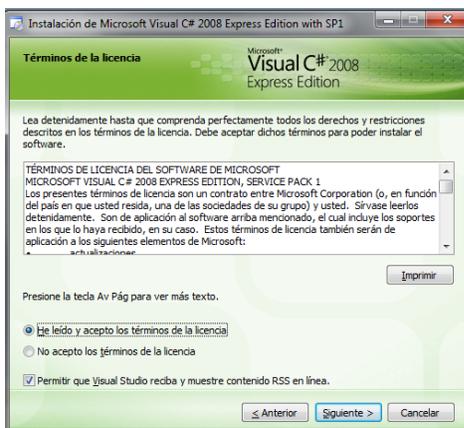
Es el lenguaje "estrella" del .NET, ya que nació con la primera versión del framework. Es un lenguaje sencillo de aprender (tiene una cierta similitud en la sintaxis con Java y C++), y sobretodo muy potente para realizar aplicaciones para Windows.



**Figura A2.12** Seleccionar Visual C# 2008



**Figura A2.13** Asistente de Instalación de Visual C#



**Figura A2.14** Términos de Licencia Visual C#

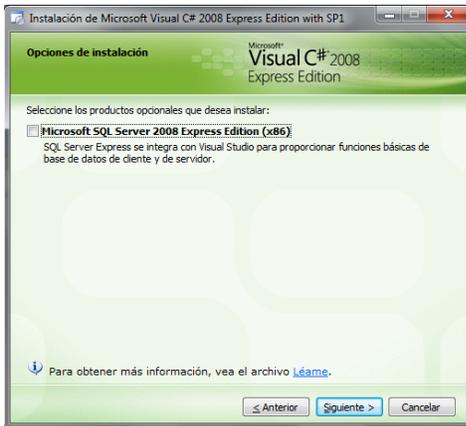


Figura A2.15 Opciones de Instalacion Visual C#



Figura A2.16 Carpeta de Instalación Visual C#

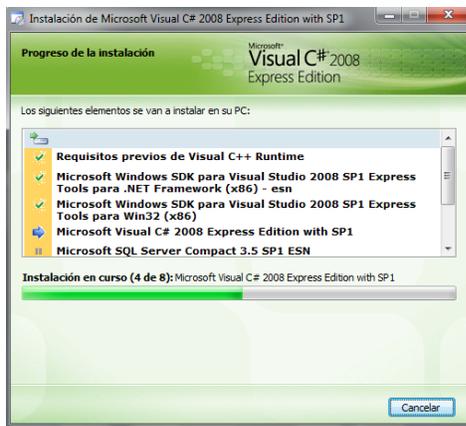
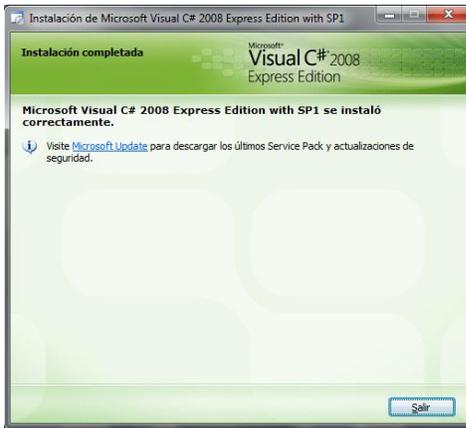


Figura A2.17 Progreso de Instalacion Visual C#



**Figura A2.18** Instalacion completa Visual C#

### **Instalación de Visual Web Developer 2008 Express Edition**

Es el paquete para desarrollo de sitios web ASP.NET con Visual Basic .NET o C#, tiene las funciones de arrastrar y soltar elementos de la barra de herramientas en el diseñador, separacion de codigo del lado del cliente y del servidor, un limitado explorador de base de datos; apoyo a otras tecnologías web (por ejemplo, CSS, JavaScript, XML); y la opcion de integrarse con AJAX, aunque para la version 2008, ajax ya viene incluido.



**Figura A2.19** Seleccionar Visual Web Developer 2008

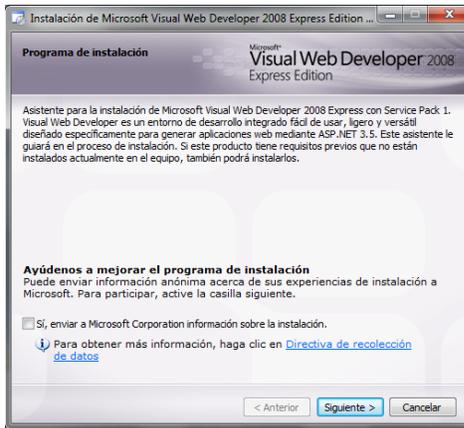


Figura A2.20 Asistente de Instalacion Visual W. D.

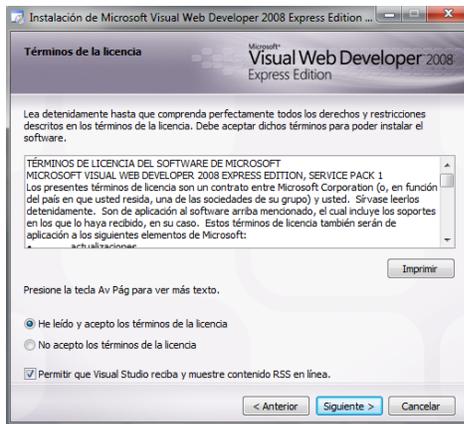


Figura A21.18 Terminos de Licencia Visual W. D.

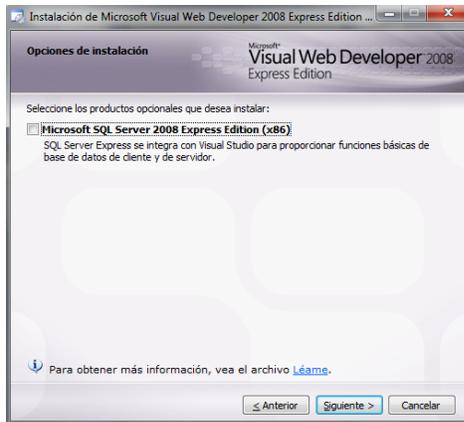
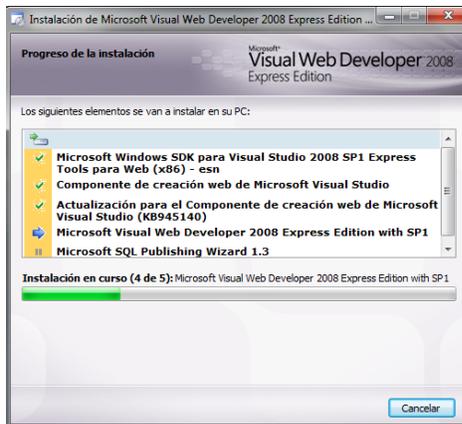


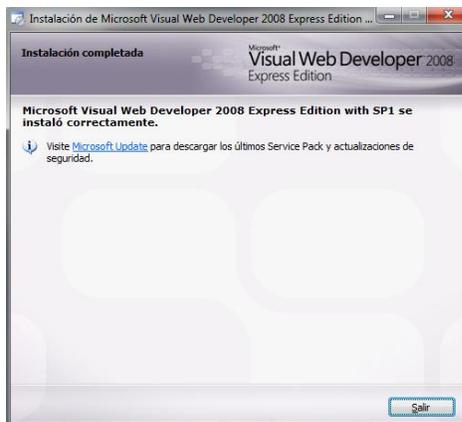
Figura A22.18 Opciones de Instalación Visual W. D.



**Figura A23.18** Carpeta de Instalación Visual W. D.



**Figura A24.18** Progreso de la Instalación Visual W. D.

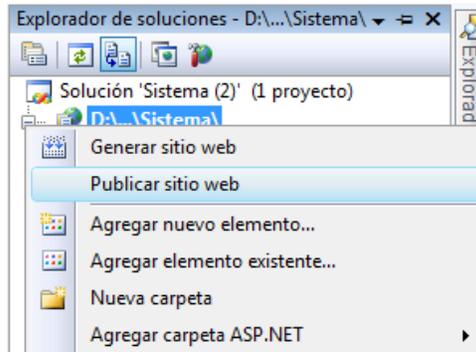


**Figura A25.18** Instalacion completa Visual W. D.

### ANEXO 3: PUBLICACIÓN DE LAS FUENTES

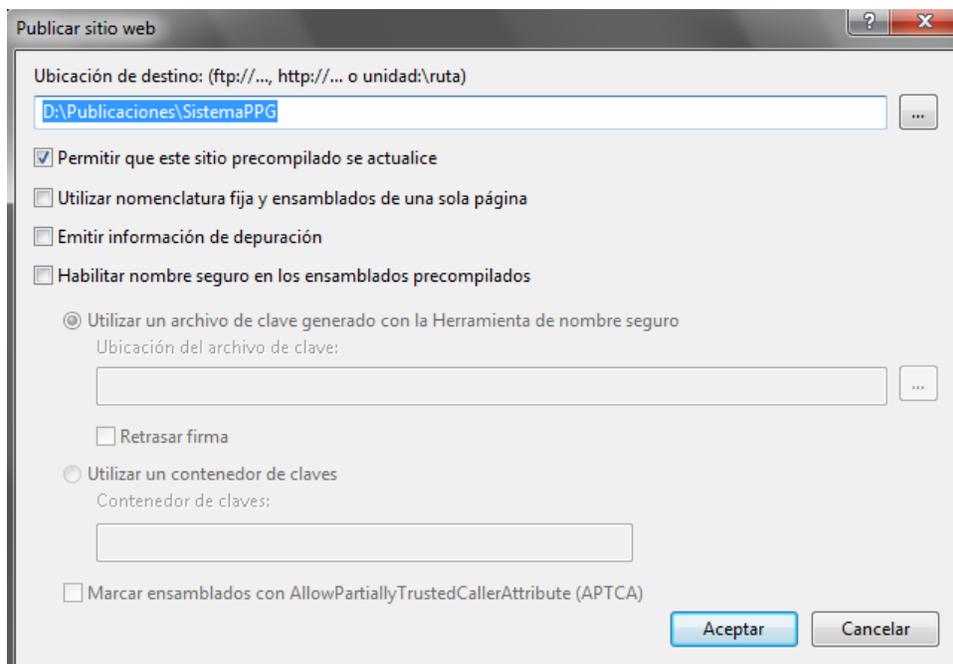
Como la aplicación esta orientada a Web entonces una vez que las fuentes estén libres de errores procedemos a realizar la publicación.

- En el explorador de soluciones click derecho sobre el nombre de la aplicación y presionar en Publicar Sitio Web.



**Figura A3.1** Publicación Fuentes 1

- Ponemos la ubicación donde se guardara la publicación y aceptamos.



**Figura A3.2** Publicación Fuentes 2

## ANEXO 4: SCRIPT DE COMANDOS SQL

Una vez instalada el gestor de base de datos y creada la base de datos ejecutaremos el siguiente Script:

```
CREATE TABLE "Users"
(
    "pId"                character(36)          NOT NULL,
    "Username"           character varying(255) NOT NULL,
    "Apellido"           character varying(100)  NULL,
    "Nombre"             character varying(100)  NULL,
    "ApplicationName"    character varying(255) NOT NULL,
    "Email"              character varying(128)  NULL,
    "Comment"           character varying(128)  NULL,
    "Password"          character varying(255) NOT NULL,
    "PasswordQuestion"  character varying(255)  NULL,
    "PasswordAnswer"    character varying(255)  NULL,
    "IsApproved"        boolean
    NULL,
    "LastActivityDate"  timestampz          NULL,
    "LastLoginDate"    timestampz
    NULL,
    "LastPasswordChangedDate" timestampz          NULL,
    "CreationDate"     timestampz          NULL,
    "IsOnLine"         boolean
    NULL,
    "IsLockedOut"      boolean
    NULL,
    "LastLockedOutDate" timestampz          NULL,
    "FailedPasswordAttemptCount" integer
    NULL,
    "FailedPasswordAttemptWindowStart" timestampz          NULL,
    "FailedPasswordAnswerAttemptCount" integer          NULL,
    "FailedPasswordAnswerAttemptWindowStart" timestampz          NULL,
    "CodigoSucursal"   character varying(10)  NULL,
    CONSTRAINT users_pkey PRIMARY KEY ("pId"),
    CONSTRAINT users_username_application_unique UNIQUE ("Username",
"ApplicationName"),
    CONSTRAINT fk_usuarios_01 FOREIGN KEY ("CodigoSucursal") REFERENCES
"Generales"."Sucursales" ("Codigo")
);
CREATE INDEX users_email_index ON "Users" ("Email");
CREATE INDEX users_islockedout_index ON "Users" ("IsLockedOut");
```

**CREATE TABLE "Roles"**

```
(
    "Rolename"          character varying(255) NOT NULL,
    "ApplicationName"   character varying(255) NOT NULL,
    CONSTRAINT roles_pkey PRIMARY KEY ("Rolename", "ApplicationName")
);
```

**CREATE TABLE "UsersInRoles"**

```
(
    "Username"          character varying(255) NOT NULL,
    "Rolename"          character varying(255) NOT NULL,
    "ApplicationName"   character varying(255) NOT NULL,
    CONSTRAINT usersinroles_pkey PRIMARY KEY ("Username", "Rolename",
"ApplicationName"),
    CONSTRAINT usersinroles_username_fkey FOREIGN KEY ("Username",
"ApplicationName") REFERENCES "Users" ("Username", "ApplicationName") ON DELETE
CASCADE,
    CONSTRAINT usersinroles_rolename_fkey FOREIGN KEY ("Rolename",
"ApplicationName") REFERENCES "Roles" ("Rolename", "ApplicationName") ON DELETE
CASCADE
);
```

**CREATE TABLE "Generales"."TipoDivision"**

```
(
    "Secuencial" serial NOT NULL,
    "Codigo" character varying(20) NOT NULL,
    "Nombre" character varying(100) NOT NULL,
    "EstaActivo" boolean NOT NULL,
    "NumeroVerificador" integer NOT NULL,
    CONSTRAINT pk_TipoDivision PRIMARY KEY ("Secuencial")
);
```

**CREATE TABLE "Generales"."NivelDivision"**

```
(
    "Secuencial" serial NOT NULL,
    "Nivel" integer NOT NULL,
    "Nombre" character varying(100) NOT NULL,
    "SecuencialTipoDivision" integer NOT NULL,
    "NumeroVerificador" integer NOT NULL,
    CONSTRAINT pk_NivelDivision PRIMARY KEY ("Secuencial"),
    CONSTRAINT fk_NivelDivision_01 FOREIGN KEY ("SecuencialTipoDivision") REFERENCES
"Generales"."TipoDivision" ("Secuencial")
);
```

**CREATE TABLE "Generales"."Division"**

```
(
  "Secuencial" serial NOT NULL,
  "SecuencialNivelDivision" integer NOT NULL,
  "Codigo" character varying(40),
  "Nombre" character varying(200) NOT NULL,
  "NumeroVerificador" integer NOT NULL,
  CONSTRAINT pk_Division PRIMARY KEY ("Secuencial"),
  CONSTRAINT fk_Division_01 FOREIGN KEY ("SecuencialNivelDivision") REFERENCES
  "Generales"."NivelDivision" ("Secuencial")
);
```

**CREATE TABLE "Generales"."PadreDivision"**

```
(
  "Secuencial" serial NOT NULL,
  "SecuencialDivision" integer NOT NULL,
  "SecuencialDivisionPadre" integer NOT NULL,
  "NumeroVerificador" integer NOT NULL,
  CONSTRAINT pk_PadreDivision PRIMARY KEY ("Secuencial"),
  CONSTRAINT fk_PadreDivision_01 FOREIGN KEY ("SecuencialDivision") REFERENCES
  "Generales"."Division" ("Secuencial"),
  CONSTRAINT fk_PadreDivision_02 FOREIGN KEY ("SecuencialDivisionPadre") REFERENCES
  "Generales"."Division" ("Secuencial")
);
```

**CREATE TABLE "Generales"."Pais"**

```
(
  "Codigo" character varying(20) NOT NULL,
  "Nombre" character varying(100) NOT NULL,
  "EstaActivo" boolean NOT NULL,
  "NumeroVerificador" integer NOT NULL,
  "SecuencialTipoDivisionPolitica" integer NOT NULL,
  CONSTRAINT pk_Pais PRIMARY KEY ("Codigo"),
  CONSTRAINT fk_Pais_01 FOREIGN KEY ("SecuencialTipoDivisionPolitica") REFERENCES
  "Generales"."TipoDivision" ("Secuencial")
);
```

**CREATE TABLE "Generales"."TipoTelefono"**

```
(
  "Codigo" character varying(2) NOT NULL,
  "Nombre" character varying(100) NOT NULL,
  "EstaActivo" boolean NOT NULL,
  "NumeroVerificador" integer NOT NULL,
  CONSTRAINT pk_TipoTelefono PRIMARY KEY ("Codigo")
);
```

```

CREATE TABLE "Generales"."EmpresaTelefonica"
(
  "Codigo" character varying(6) NOT NULL,
  "Nombre" character varying(100) NOT NULL,
  "CodigoPais" character varying(20) NOT NULL,
  "CodigoTipoTelefono" character varying(2) NOT NULL,
  "ExtencionAuto" boolean NOT NULL,
  "EstaActivo" boolean NOT NULL,
  "NumeroVerificador" integer NOT NULL,
  CONSTRAINT pk_EmpresaTelefonica PRIMARY KEY ("Codigo"),
  CONSTRAINT fk_EmpresaTelefonica_01 FOREIGN KEY ("CodigoPais") REFERENCES
"Generales"."Pais" ("Codigo"),
  CONSTRAINT fk_EmpresaTelefonica_02 FOREIGN KEY ("CodigoTipoTelefono") REFERENCES
"Generales"."TipoTelefono" ("Codigo")
);

```

```

CREATE TABLE "Generales"."TipoBanco"
(
  "Codigo" character varying(4) NOT NULL,
  "Nombre" character varying(100) NOT NULL,
  "EstaActivo" boolean NOT NULL,
  "NumeroVerificador" integer NOT NULL,
  CONSTRAINT pk_TipoBanco PRIMARY KEY ("Codigo")
);

```

```

CREATE TABLE "Generales"."Banco"
(
  "Secuencial" serial NOT NULL,
  "Codigo" character varying(20) NOT NULL,
  "Nombre" character varying(100) NOT NULL,
  "CodigoTipoBanco" character varying(4) NOT NULL,
  "EstaActivo" boolean NOT NULL,
  "NumeroVerificador" integer NOT NULL,
  CONSTRAINT pk_Banco PRIMARY KEY ("Secuencial"),
  CONSTRAINT fk_Banco_01 FOREIGN KEY ("CodigoTipoBanco") REFERENCES
"Generales"."TipoBanco" ("Codigo")
);

```

```

CREATE TABLE "Generales"."Vendedor"
(
  "Secuencial" serial NOT NULL,
  "Nombre" character varying(100) NOT NULL,
  "EstaActivo" boolean NOT NULL,
  "NumeroVerificador" integer NOT NULL,
  CONSTRAINT pk_Vendedor PRIMARY KEY ("Secuencial")
);

```

```

CREATE TABLE "Generales"."TipoFormaPago"
(
  "Codigo" character varying(5) NOT NULL,
  "Nombre" character varying(20) NOT NULL,
  "EstaActivo" boolean NOT NULL,
  "NumeroVerificador" integer NOT NULL,
  CONSTRAINT pk_TipoFormaPago PRIMARY KEY ("Codigo")
);

```

```

CREATE TABLE "Generales"."Sucursales"
(
  "Codigo" character varying(10) NOT NULL,
  "Nombre" character varying(50) NOT NULL,
  CONSTRAINT pk_Sucursales PRIMARY KEY ("Codigo")
);

```

```

CREATE TABLE "Generales"."Calendario"
(
  "Secuencial" serial NOT NULL,
  "FechaSistema" date NOT NULL,
  "EstaCerrado" boolean NOT NULL,
  "NumeroVerificador" integer NOT NULL,
  CONSTRAINT pk_Calendario PRIMARY KEY ("Secuencial")
);

```

```

CREATE TABLE "Generales"."SucursalesComplementoSRI"
(
  "CodigoSucursal" character varying(10) NOT NULL,
  "CodigoEstablecimiento" character varying(6) NOT NULL,
  "CodigoPuntoEmision" character varying(6) NOT NULL,
  "Autorizacion" character varying(100) NOT NULL,
  "SecuencialInicio" integer NOT NULL,
  "SecuencialFin" integer NOT NULL,
  "EsAutomatico" boolean NOT NULL,
  "SecuencialAutomatico" integer NOT NULL,
  CONSTRAINT pk_SucursalesComplementoSRI PRIMARY KEY ("CodigoSucursal"),
  CONSTRAINT fk_SucursalesComplementoSRI_01 FOREIGN KEY ("CodigoSucursal")
REFERENCES "Generales"."Sucursales" ("Codigo"));

```

```

CREATE TABLE "Generales"."ExtensionTelefono"
(
  "Secuencial" serial NOT NULL,
  "SecuencialProvincia" integer NOT NULL,
  "Extension" character varying(5) NOT NULL,
  "NumeroVerificador" integer NOT NULL,
  CONSTRAINT pk_extensiotelefono PRIMARY KEY ("Secuencial" ),

```

```

CONSTRAINT fk_extensiotelefono_01 FOREIGN KEY ("SecuencialProvincia")
REFERENCES "Generales"."Division" ("Secuencial")
);

```

**CREATE TABLE "Generales"."Empresa"**

```

(
"Secuencial" serial NOT NULL,
"Siglas" character varying(50) NOT NULL,
"Nombre" character varying(100) NOT NULL,
"Ruc" character varying(50) NOT NULL,
"CodigoPais" character varying(20) NOT NULL,
"EstaActivo" boolean NOT NULL,
"NumeroVerificador" integer NOT NULL,
CONSTRAINT pk_Empresa PRIMARY KEY ("Secuencial"),
CONSTRAINT fk_Empresa_01 FOREIGN KEY ("CodigoPais")
REFERENCES "Generales"."Pais" ("Codigo")
);

```

**CREATE TABLE "Clientes"."TipoidentificacionCliente"**

```

(
"Codigo" character varying(5) NOT NULL,
"Nombre" character varying(20) NOT NULL,
CONSTRAINT pk_tic PRIMARY KEY ("Codigo")
);

```

**CREATE TABLE "Clientes"."Cliente"**

```

(
"Secuencial" serial NOT NULL,
"SecuencialCiudad" integer NOT NULL,
"Codigo" character varying(100) NOT NULL,
"CodigoTipoidentificacion" character varying(5) NOT NULL,
"Identificacion" character varying(40) NOT NULL,
"Nombre" character varying(100) NOT NULL,
"Direccion" character varying(100),
"Email" character varying(100),
"SecuencialVendedor" integer NOT NULL,
"EstaActivo" boolean NOT NULL,
"NumeroVerificador" integer NOT NULL,
CONSTRAINT pk_Cliente PRIMARY KEY ("Secuencial"),
CONSTRAINT fk_Cliente_01 FOREIGN KEY ("SecuencialCiudad") REFERENCES
"Generales"."Division" ("Secuencial"),
CONSTRAINT fk_cliente_02 FOREIGN KEY ("CodigoTipoidentificacion") REFERENCES
"Clientes"."TipoidentificacionCliente" ("Codigo"),
CONSTRAINT fk_cliente_03 FOREIGN KEY ("SecuencialVendedor") REFERENCES
"Generales"."Vendedor" ("Secuencial"),
CONSTRAINT unica_cliente_01 UNIQUE ("Identificacion"));

```

```

CREATE TABLE "Clientes"."TelefonoCliente"
(
  "Secuencial" serial NOT NULL,
  "SecuencialCliente" integer NOT NULL,
  "CodigoEmpresaTelefonica" character varying(6) NOT NULL,
  "NumeroTelefono" character varying(40) NOT NULL,
  "EstaActivo" boolean NOT NULL,
  "NumeroVerificador" integer NOT NULL,
  CONSTRAINT pk_TelefonoCliente PRIMARY KEY ("Secuencial"),
  CONSTRAINT fk_TelefonoCliente_01 FOREIGN KEY ("SecuencialCliente") REFERENCES
"Clientes"."Cliente" ("Secuencial"),
  CONSTRAINT fk_TelefonoCliente_02 FOREIGN KEY ("CodigoEmpresaTelefonica")
REFERENCES "Generales"."EmpresaTelefonica" ("Codigo")
);

```

```

CREATE TABLE "Clientes"."FormaDeVenta"
(
  "Secuencial" serial NOT NULL,
  "SecuencialCliente" integer NOT NULL,
  "CodigoFormaDeVenta" character varying(5) NOT NULL,
  "EstaActivo" boolean NOT NULL,
  "NumeroVerificador" integer NOT NULL,
  CONSTRAINT pk_FormaDeVenta PRIMARY KEY ("Secuencial"),
  CONSTRAINT fk_FormaDeVenta_01 FOREIGN KEY ("SecuencialCliente") REFERENCES
"Clientes"."Cliente" ("Secuencial"),
  CONSTRAINT fk_FormaDeVenta_02 FOREIGN KEY ("CodigoFormaDeVenta") REFERENCES
"Generales"."TipoFormaPago" ("Codigo")
);

```

```

CREATE TABLE "Colores"."ColorEstandar"
(
  "Secuencial" serial NOT NULL,
  "Codigo" character varying(20) NOT NULL,
  "Nombre" character varying(100) NOT NULL,
  "Litro" integer NOT NULL,
  "Gramos" numeric(12,2) NOT NULL,
  "Precio" numeric(12,2) NOT NULL,
  "EstaActivo" boolean NOT NULL,
  "NumeroVerificador" integer NOT NULL,
  CONSTRAINT pk_ColorEstandar PRIMARY KEY ("Secuencial")
);

```

```

CREATE TABLE "Colores"."GeneracionColorMaestro"
(
  "Secuencial" serial NOT NULL,
  "Codigo" character varying(20) NOT NULL,
  "Nombre" character varying(100) NOT NULL,
  "EstaActivo" boolean NOT NULL,
  "NumeroVerificador" integer NOT NULL,
  CONSTRAINT pk_GeneracionColorMaestro PRIMARY KEY ("Secuencial"),
  CONSTRAINT unica_GeneracionColorMaestro_01 UNIQUE ("Codigo")
);

```

```

CREATE TABLE "Colores"."GeneracionColorDetalle"
(
  "Secuencial" serial NOT NULL,
  "SecuencialGeneracionColorMaestro" integer NOT NULL,
  "SecuencialColorEstandar" integer NOT NULL,
  "CantidadEnLitros" numeric(12,6) NOT NULL,
  "CantidadEnGramos" numeric(12,6) NOT NULL,
  CONSTRAINT pk_GeneracionColorDetalle PRIMARY KEY ("Secuencial"),
  CONSTRAINT fk_GeneracionColorDetalle_01 FOREIGN KEY
("SecuencialGeneracionColorMaestro") REFERENCES "Colores"."GeneracionColorMaestro"
("Secuencial"),
  CONSTRAINT fk_GeneracionColorDetalle_02 FOREIGN KEY ("SecuencialColorEstandar")
REFERENCES "Colores"."ColorEstandar" ("Secuencial")
);

```

```

CREATE TABLE "Colores"."ColorEstandarSucursal"
(
  "Secuencial" serial NOT NULL,
  "SecuencialColorEstandar" integer NOT NULL,
  "CodigoSucursal" character varying(10) NOT NULL,
  "CantidadEnLitros" numeric(12,6) NOT NULL,
  CONSTRAINT pk_ColorEstandarSucursal PRIMARY KEY ("Secuencial"),
  CONSTRAINT fk_ColorEstandarSucursal_01 FOREIGN KEY ("SecuencialColorEstandar")
REFERENCES "Colores"."ColorEstandar" ("Secuencial"),
  CONSTRAINT fk_ColorEstandarSucursal_02 FOREIGN KEY ("CodigoSucursal") REFERENCES
"Generales"."Sucursales" ("Codigo")
);

```

```

CREATE TABLE "Facturacion"."TipoCantidad"
(
  "Codigo" character varying(1) NOT NULL,
  "Nombre" character varying(10) NOT NULL,
  CONSTRAINT pk_TipoCantidad PRIMARY KEY ("Codigo")
);

```

**CREATE TABLE "Facturacion"."FacturaMaestro"**

```
(  
  "Codigo" character varying(12) NOT NULL,  
  "Estado" character varying(20) NOT NULL,  
  CONSTRAINT pk_FacturaMaestro PRIMARY KEY ("Codigo")  
);
```

**CREATE TABLE "Facturacion"."FacturaCabecera"**

```
(  
  "Secuencial" serial NOT NULL,  
  "CodigoFacturaMaestro" character varying(12) NOT NULL,  
  "SecuencialCliente" integer NOT NULL,  
  "NombreUsuario" character varying(255) NOT NULL,  
  "CodigoSucursal" character varying(10) NOT NULL,  
  "Fecha" date NOT NULL,  
  "Comentario" character varying(250),  
  "Subtotal" numeric(12,2) NOT NULL,  
  "Descuento" numeric(12,2) NOT NULL,  
  "Iva" numeric(12,2) NOT NULL,  
  "Total" numeric(12,2) NOT NULL,  
  "SubtotalDescuento" numeric(12,2) NOT NULL,  
  "DescuentoDescuento" numeric(12,2) NOT NULL,  
  "IvaDescuento" numeric(12,2) NOT NULL,  
  "TotalDescuento" numeric(12,2) NOT NULL,  
  "TelefonoCliente" character varying(40),  
  "ApplicationName" character varying(255),  
  CONSTRAINT pk_FacturaCabecera PRIMARY KEY ("Secuencial"),  
  CONSTRAINT fk_FacturaCabecera_01 FOREIGN KEY ("CodigoFacturaMaestro") REFERENCES  
"Facturacion"."FacturaMaestro" ("Codigo"),  
  CONSTRAINT fk_FacturaCabecera_02 FOREIGN KEY ("SecuencialCliente") REFERENCES  
"Clientes"."Cliente" ("Secuencial"),  
  CONSTRAINT fk_FacturaCabecera_03 FOREIGN KEY ("NombreUsuario",  
"ApplicationName") REFERENCES "Users" ("Username", "ApplicationName"),  
  CONSTRAINT fk_FacturaCabecera_04 FOREIGN KEY ("CodigoSucursal") REFERENCES  
"Generales"."Sucursales" ("Codigo")  
);
```

**CREATE TABLE "Facturacion"."FacturaDetalle"**

```
(  
  "Secuencial" serial NOT NULL,  
  "CodigoFacturaMaestro" character varying(12) NOT NULL,  
  "SecuencialGeneracionColorMaestro" integer NOT NULL,  
  "Cantidad" numeric(12,2) NOT NULL,  
  "Precio" numeric(12,2) NOT NULL,  
  "Subtotal" numeric(12,2) NOT NULL,  
  "Descuento" numeric(12,2) NOT NULL,
```

```

"Total" numeric(12,2) NOT NULL,
"CodigoTipoCantidad" character varying(1) NOT NULL,
CONSTRAINT pk_FacturaDetalle PRIMARY KEY ("Secuencial"),
CONSTRAINT fk_FacturaDetalle_01 FOREIGN KEY ("CodigoFacturaMaestro") REFERENCES
"Facturacion"."FacturaMaestro" ("Codigo"),
CONSTRAINT fk_FacturaDetalle_02 FOREIGN KEY ("SecuencialGeneracionColorMaestro")
REFERENCES "Colores"."GeneracionColorMaestro" ("Secuencial"),
CONSTRAINT fk_FacturaDetalle_03 FOREIGN KEY ("CodigoTipoCantidad") REFERENCES
"Facturacion"."TipoCantidad" ("Codigo")
);

```

**CREATE TABLE "Facturacion"."FacturaDesglose"**

```

(
"Secuencial" serial NOT NULL,
"SecuencialFacturaDetalle" integer NOT NULL,
"SecuencialColorEstandar" integer NOT NULL,
"Cantidad" numeric(12,6) NOT NULL,
"Precio" numeric(12,2) NOT NULL,
"Subtotal" numeric(12,2) NOT NULL,
CONSTRAINT pk_FacturaDesglose PRIMARY KEY ("Secuencial"),
CONSTRAINT fk_FacturaDesglose_01 FOREIGN KEY ("SecuencialFacturaDetalle")
REFERENCES "Facturacion"."FacturaDetalle" ("Secuencial"),
CONSTRAINT fk_FacturaDesglose_02 FOREIGN KEY ("SecuencialColorEstandar")
REFERENCES "Colores"."ColorEstandar" ("Secuencial")
);

```

**CREATE TABLE "Facturacion"."FacturaFormaPago"**

```

(
"Secuencial" serial NOT NULL,
"CodigoFacturaMaestro" character varying(12) NOT NULL,
"CodigoTipoFormaPago" character varying(5) NOT NULL,
"Valor" numeric(12,2),
CONSTRAINT pk_FacturaFormaPago PRIMARY KEY ("Secuencial"),
CONSTRAINT fk_FacturaFormaPago_01 FOREIGN KEY ("CodigoFacturaMaestro")
REFERENCES "Facturacion"."FacturaMaestro" ("Codigo"),
CONSTRAINT fk_FacturaFormaPago_02 FOREIGN KEY ("CodigoTipoFormaPago")
REFERENCES "Generales"."TipoFormaPago" ("Codigo")
);

```

**CREATE TABLE "Facturacion"."FacturaRetencion"**

```

(
"Secuencial" serial NOT NULL,
"SecuencialFacturaCabecera" integer NOT NULL,
"FechaEmision" date NOT NULL,
"BaseImponible" numeric(12,2) NOT NULL,
"MontoSobreImpuesto" numeric(12,2) NOT NULL,

```

```

"PorcentajeRendimiento" numeric(12,2) NOT NULL,
"ValorRetenido" numeric(12,2) NOT NULL,
CONSTRAINT pk_FacturaRetencion PRIMARY KEY ("Secuencial"),
CONSTRAINT fk_FacturaRetencion_01 FOREIGN KEY ("SecuencialFacturaCabecera")
REFERENCES "Facturacion"."FacturaCabecera" ("Secuencial")
);

```

**CREATE TABLE "Facturacion"."FacturaRetencionImpresion"**

```

(
"SecuencialFacturaRetencion" integer NOT NULL,
"CodigoEstablecimiento" character varying(6) NOT NULL,
"CodigoPuntoEmision" character varying(6) NOT NULL,
"SecuencialDocumentoRetencion" character varying(14) NOT NULL,
"Autorizacion" character varying(100) NOT NULL,
CONSTRAINT pk_FacturaRetencionImpresion PRIMARY KEY
("SecuencialFacturaRetencion"),
CONSTRAINT fk_FacturaRetencionImpresion_01 FOREIGN KEY
("SecuencialFacturaRetencion") REFERENCES "Facturacion"."FacturaRetencion"
("Secuencial")
);

```

**CREATE TABLE "Facturacion"."FacturaCodigosSRI"**

```

(
"CodigoFacturaMaestro" character varying(12) NOT NULL,
"CodigoEstablecimiento" character varying(6) NOT NULL,
"CodigoPuntoEmision" character varying(6) NOT NULL,
"SecuencialDocumento" character varying(14) NOT NULL,
"Autorizacion" character varying(100) NOT NULL,
CONSTRAINT pk_FacturaCodigosSRI PRIMARY KEY ("CodigoFacturaMaestro"),
CONSTRAINT fk_FacturaCodigosSRI_01 FOREIGN KEY ("CodigoFacturaMaestro")
REFERENCES "Facturacion"."FacturaMaestro" ("Codigo")
);

```

**CREATE TABLE "Facturacion"."EstadoFactura"**

```

(
"Codigo" character varying(4) NOT NULL,
"Nombre" character varying(100) NOT NULL,
"EstaActivo" boolean NOT NULL,
"NumeroVerificador" integer NOT NULL,
CONSTRAINT pk_EstadoFactura PRIMARY KEY ("Codigo")
);

```

```

CREATE TABLE "Credito"."EstadoCredito"
(
  "Codigo" character varying(4) NOT NULL,
  "Nombre" character varying(100) NOT NULL,
  "EstaActivo" boolean NOT NULL,
  "NumeroVerificador" integer NOT NULL,
  CONSTRAINT pk_EstadoCredito PRIMARY KEY ("Codigo")
);

```

```

CREATE TABLE "Credito"."EstadoCreditoComponente"
(
  "Codigo" character varying(4) NOT NULL,
  "Nombre" character varying(100) NOT NULL,
  "EstaActivo" boolean NOT NULL,
  "NumeroVerificador" integer NOT NULL,
  CONSTRAINT pk_EstadoCreditoComponente PRIMARY KEY ("Codigo")
);

```

```

CREATE TABLE "Credito"."CreditoMaestro"
(
  "Secuencial" serial NOT NULL,
  "CodigoFacturaMaestro" character varying(12) NOT NULL,
  "Monto" numeric(12,2) NOT NULL,
  "NumeroCuotas" integer NOT NULL,
  "FrecuenciaPago" integer NOT NULL,
  "TasaInteres" numeric(6,2) NOT NULL,
  "FechaCreacion" date NOT NULL,
  "FechaVencimiento" date NOT NULL,
  "CodigoEstadoCredito" character varying(4) NOT NULL,
  "SaldoActual" numeric(12,2) NOT NULL,
  "Calificacion" character varying(2) NOT NULL,
  "TotalAPagar" numeric(12,2) NOT NULL,
  CONSTRAINT pk_CreditoMaestro PRIMARY KEY ("Secuencial"),
  CONSTRAINT fk_CreditoMaestro_01 FOREIGN KEY ("CodigoFacturaMaestro") REFERENCES
"Facturacion"."FacturaMaestro" ("Codigo"),
  CONSTRAINT fk_CreditoMaestro_02 FOREIGN KEY ("CodigoEstadoCredito") REFERENCES
"Credito"."EstadoCredito" ("Codigo")
);

```

```

CREATE TABLE "Credito"."CreditoComponente"
(
  "Secuencial" serial NOT NULL,
  "SecuencialCreditoMaestro" integer NOT NULL,
  "NumeroCuota" integer NOT NULL,
  "ValorCalculado" numeric(12,2) NOT NULL,
  "ValorCabrado" numeric(12,2) NOT NULL,

```

```

"FechaInicio" date NOT NULL,
"FechaVencimiento" date NOT NULL,
"CodigoEstadoCreditoComponente" character varying(4) NOT NULL,
CONSTRAINT pk_CreditoComponente PRIMARY KEY ("Secuencial"),
CONSTRAINT fk_CreditoComponente_01 FOREIGN KEY ("SecuencialCreditoMaestro")
REFERENCES "Credito"."CreditoMaestro" ("Secuencial"),
CONSTRAINT fk_CreditoComponente_02 FOREIGN KEY
("CodigoEstadoCreditoComponente") REFERENCES "Credito"."EstadoCreditoComponente"
("Codigo")
);

```

**CREATE TABLE "Credito"."CreditoFormaPago"**

```

(
"Secuencial" serial NOT NULL,
"CodigoTipoFormaPago" character varying(5) NOT NULL,
"EstaActivo" boolean NOT NULL,
"NumeroVerificador" integer NOT NULL,
CONSTRAINT pk_CreditoFormaPago PRIMARY KEY ("Secuencial"),
CONSTRAINT fk_CreditoFormaPago_01 FOREIGN KEY ("CodigoTipoFormaPago")
REFERENCES "Generales"."TipoFormaPago" ("Codigo")
);

```

**CREATE TABLE "Credito"."MovimientoCreditoComponente"**

```

(
"Secuencial" serial NOT NULL,
"SecuencialCreditoMaestro" integer NOT NULL,
"CodigoSucursal" character varying(10) NOT NULL,
"NombreUsuario" character varying(255) NOT NULL,
"Fecha" date NOT NULL,
"Valor" numeric(12,2) NOT NULL,
"Saldo" numeric(12,2) NOT NULL,
"ApplicationName" character varying(255),
CONSTRAINT pk_MovimientoCreditoComponente PRIMARY KEY ("Secuencial"),
CONSTRAINT fk_MovimientoCreditoComponente_01 FOREIGN KEY
("SecuencialCreditoMaestro") REFERENCES "Credito"."CreditoMaestro" ("Secuencial"),
CONSTRAINT fk_MovimientoCreditoComponente_02 FOREIGN KEY ("CodigoSucursal")
REFERENCES "Generales"."Sucursales" ("Codigo"),
CONSTRAINT fk_MovimientoCreditoComponente_03 FOREIGN KEY ("NombreUsuario",
"ApplicationName") REFERENCES "Users" ("Username", "ApplicationName")
);

```

```

CREATE TABLE "Credito"."MovimientoDetalleCreditoComponente"
(
  "Secuencial" serial NOT NULL,
  "SecuencialMovimientoCreditoComponente" integer NOT NULL,
  "NumeroCuota" integer NOT NULL,
  "Valor" numeric(12,2) NOT NULL,
  CONSTRAINT pk_MovimientoDetalleCreditoComponente PRIMARY KEY ("Secuencial")
);

```

```

CREATE TABLE "Remesas"."EstadoCheque"
(
  "Codigo" character varying(40) NOT NULL,
  "Nombre" character varying(100) NOT NULL,
  "EstaActivo" boolean NOT NULL,
  "NumeroVerificador" integer NOT NULL,
  CONSTRAINT pk_EstadoCheque PRIMARY KEY ("Codigo")
);

```

```

CREATE TABLE "Remesas"."Cheque"
(
  "Secuencial" serial NOT NULL,
  "CodigoCuentaCorriente" character varying(40) NOT NULL,
  "CodigoCheque" character varying(40) NOT NULL,
  "SecuencialBancoEmisor" integer NOT NULL,
  "Valor" numeric(12,2) NOT NULL,
  "NombreUsuario" character varying(255) NOT NULL,
  "EstaEnBodega" boolean NOT NULL,
  "FechaSistemaIngreso" date NOT NULL,
  "FechaMaquinaIngreso" date NOT NULL,
  "CodigoEstadoCheque" character varying(40) NOT NULL,
  "NumeroVerificador" integer NOT NULL,
  "ApplicationName" character varying(255),
  CONSTRAINT pk_Cheque PRIMARY KEY ("Secuencial"),
  CONSTRAINT fk_Cheque_01 FOREIGN KEY ("SecuencialBancoEmisor") REFERENCES
"Generales"."Banco" ("Secuencial"),
  CONSTRAINT fk_Cheque_02 FOREIGN KEY ("NombreUsuario", "ApplicationName")
REFERENCES "Users" ("Username", "ApplicationName"),
  CONSTRAINT fk_Cheque_03 FOREIGN KEY ("CodigoEstadoCheque") REFERENCES
"Remesas"."EstadoCheque" ("Codigo")
);

```

```

CREATE TABLE "ArchivosAT"."RetencionIvaBienes"
(
  "Secuencial" serial,
  "Codigo" character varying(10),
  "Porcentaje" numeric(12,2),
  "FechaInicio" date,
  "FechaFin" date,
  "EstaActivo" boolean,
  CONSTRAINT pk_RetencionIvaBienes PRIMARY KEY ("Secuencial")
);

```

```

CREATE TABLE "ArchivosAT"."RetencionIvaServicios"
(
  "Secuencial" serial,
  "Codigo" character varying(10),
  "Porcentaje" numeric(12,2),
  "FechaInicio" date,
  "FechaFin" date,
  "EstaActivo" boolean,
  CONSTRAINT pk_RetencionIvaServicios PRIMARY KEY ("Secuencial")
);

```

```

CREATE TABLE "ArchivosAT"."TiposIdentificacion"
(
  "Codigo" character varying(5),
  "Descripcion" character varying(100),
  CONSTRAINT pk_TiposIdentificacion PRIMARY KEY ("Codigo")
);

```

```

CREATE TABLE "ArchivosAT"."TiposTransacciones"
(
  "Codigo" character varying(5),
  "Descripcion" character varying(100),
  CONSTRAINT pk_TiposTransacciones PRIMARY KEY ("Codigo")
);

```

```

CREATE TABLE "ArchivosAT"."SecuenciasTransaccion"
(
  "Codigo" character varying(5),
  "CodigoTiposIdentificacion" character varying(5),
  "CodigoTiposTransacciones" character varying(5),
  CONSTRAINT pk_SecuenciasTransaccion PRIMARY KEY ("Codigo"),
  CONSTRAINT fk_SecuenciasTransaccion_01 FOREIGN KEY ("CodigoTiposIdentificacion")
REFERENCES "ArchivosAT"."TiposIdentificacion" ("Codigo"),
  CONSTRAINT fk_SecuenciasTransaccion_02 FOREIGN KEY ("CodigoTiposTransacciones")
REFERENCES "ArchivosAT"."TiposTransacciones" ("Codigo"));

```

```

CREATE TABLE "Compras"."Proveedor"
(
  "Secuencial" serial NOT NULL,
  "Nombre" character varying(120) NOT NULL,
  "CodigoTipoidentificacion" character varying(20) NOT NULL,
  "Identificacion" character varying(26) NOT NULL,
  "SecuencialCiudad" integer NOT NULL,
  "Direccion" character varying(500),
  "Contacto" character varying(200),
  "Telefono" character varying(40),
  "Fax" character varying(40),
  "Email" character varying(70),
  "EstaCalificado" boolean,
  "EsContribuyenteEspecial" boolean,
  "CodigoSucursal" character varying(10),
  "CodigoUsuarioIngreso" character varying(255),
  "FechaSistemaIngreso" date,
  "FechaMaquinaIngreso" date,
  "EstaActivo" boolean NOT NULL,
  "NumeroVerificador" integer NOT NULL,
  "ApplicationName" character varying(255),
  "Codigo" character varying(100) NOT NULL,
  CONSTRAINT pk_Proveedor PRIMARY KEY ("Secuencial"),
  CONSTRAINT fk_Proveedor_01 FOREIGN KEY ("SecuencialCiudad") REFERENCES
"Generales"."Division" ("Secuencial"),
  CONSTRAINT fk_Proveedor_02 FOREIGN KEY ("CodigoSucursal") REFERENCES
"Generales"."Sucursales" ("Codigo"),
  CONSTRAINT fk_Proveedor_03 FOREIGN KEY ("CodigoUsuarioIngreso", "ApplicationName")
REFERENCES "Users" ("Username", "ApplicationName")
);

```

```

CREATE TABLE "Compras"."RetencionIR"
(
  "Secuencial" serial NOT NULL,
  "Codigo" character varying(10) NOT NULL,
  "Nombre" character varying(500) NOT NULL,
  "Porcentaje" numeric(12,2),
  "Formulario" integer NOT NULL,
  "EstaActivo" boolean NOT NULL,
  "NumeroVerificador" integer NOT NULL,
  CONSTRAINT pk_RetencionIR PRIMARY KEY ("Secuencial")
);

```

```

CREATE TABLE "Compras"."RetencionIVA"
(
  "Secuencial" serial NOT NULL,
  "Codigo" character varying(10) NOT NULL,
  "Nombre" character varying(500) NOT NULL,
  "Porcentaje" numeric(12,2),
  "Formulario" integer NOT NULL,
  "EstaActivo" boolean NOT NULL,
  "NumeroVerificador" integer NOT NULL,
  CONSTRAINT pk_RetencionIVA PRIMARY KEY ("Secuencial")
);

```

```

CREATE TABLE "Compras"."ProveedorRetencionIRBienes"
(
  "SecuencialProveedor" integer NOT NULL,
  "SecuencialRetencionIR" integer NOT NULL,
  "EstaActivo" boolean NOT NULL,
  CONSTRAINT pk_ProveedorRetencionIRBienes PRIMARY KEY ("SecuencialProveedor"),
  CONSTRAINT fk_ProveedorRetencionIRBienes_01 FOREIGN KEY ("SecuencialProveedor")
REFERENCES "Compras"."Proveedor" ("Secuencial"),
  CONSTRAINT fk_ProveedorRetencionIRBienes_02 FOREIGN KEY ("SecuencialRetencionIR")
REFERENCES "Compras"."RetencionIR" ("Secuencial")
);

```

```

CREATE TABLE "Compras"."ProveedorRetencionIRServicios"
(
  "SecuencialProveedor" integer NOT NULL,
  "SecuencialRetencionIR" integer NOT NULL,
  "EstaActivo" boolean NOT NULL,
  CONSTRAINT pk_ProveedorRetencionIRServicios PRIMARY KEY ("SecuencialProveedor"),
  CONSTRAINT fk_ProveedorRetencionIRServicios_01 FOREIGN KEY ("SecuencialProveedor")
REFERENCES "Compras"."Proveedor" ("Secuencial"),
  CONSTRAINT fk_ProveedorRetencionIRServicios_02 FOREIGN KEY
("SecuencialRetencionIR") REFERENCES "Compras"."RetencionIR" ("Secuencial")
);

```

```

CREATE TABLE "Compras"."ProveedorRetencionIVABienes"
(
  "SecuencialProveedor" integer NOT NULL,
  "SecuencialRetencionIVA" integer NOT NULL,
  "EstaActivo" boolean NOT NULL,
  CONSTRAINT pk_ProveedorRetencionIVABienes PRIMARY KEY ("SecuencialProveedor"),
  CONSTRAINT fk_ProveedorRetencionIVABienes_01 FOREIGN KEY ("SecuencialProveedor")
REFERENCES "Compras"."Proveedor" ("Secuencial"),
  CONSTRAINT fk_ProveedorRetencionIVABienes_02 FOREIGN KEY
("SecuencialRetencionIVA") REFERENCES "Compras"."RetencionIVA" ("Secuencial"));

```

```

CREATE TABLE "Compras"."ProveedorRetencionIVAServicios"
(
  "SecuencialProveedor" integer NOT NULL,
  "SecuencialRetencionIVA" integer NOT NULL,
  "EstaActivo" boolean NOT NULL,
  CONSTRAINT pk_ProveedorRetencionIVAServicios PRIMARY KEY ("SecuencialProveedor"),
  CONSTRAINT fk_ProveedorRetencionIVAServicios_01 FOREIGN KEY
("SecuencialProveedor") REFERENCES "Compras"."Proveedor" ("Secuencial"),
  CONSTRAINT fk_ProveedorRetencionIVAServicios_02 FOREIGN KEY
("SecuencialRetencionIVA") REFERENCES "Compras"."RetencionIVA" ("Secuencial")
);

```

```

CREATE TABLE "Compras"."TipoDocumentoCompra"
(
  "Codigo" character varying(4) NOT NULL,
  "Nombre" character varying(100) NOT NULL,
  "TipoComprobanteAnexo" character varying(6) NOT NULL,
  "EstaActivo" boolean NOT NULL,
  "NumeroVerificador" integer NOT NULL,
  CONSTRAINT pk_TipoDocumentoCompra PRIMARY KEY ("Codigo")
);

```

```

CREATE TABLE "Compras"."CompraMaestro"
(
  "Secuencial" serial NOT NULL,
  "SecuencialProveedor" integer NOT NULL,
  "Total" numeric(12,2) NOT NULL,
  "FechaSistema" date NOT NULL,
  "FechaMaquina" date NOT NULL,
  "Estado" character varying(20) NOT NULL,
  CONSTRAINT pk_CompraMaestro PRIMARY KEY ("Secuencial"),
  CONSTRAINT fk_CompraMaestro_01 FOREIGN KEY ("SecuencialProveedor") REFERENCES
"Compras"."Proveedor" ("Secuencial")
);

```

```

CREATE TABLE "Compras"."CompraDetalle"
(
  "Secuencial" serial NOT NULL,
  "SecuencialCompraMaestro" integer NOT NULL,
  "SecuencialColorEstandar" integer NOT NULL,
  "CodigoSucursal" character varying(10) NOT NULL,
  "CantidadEnLitros" numeric(12,2) NOT NULL,
  "Precio" numeric(12,2) NOT NULL,
  "Total" numeric(12,2) NOT NULL,
  CONSTRAINT pk_CompraDetalle PRIMARY KEY ("Secuencial"),

```

```

    CONSTRAINT fk_CompraDetalle_01 FOREIGN KEY ("SecuencialCompraMaestro")
REFERENCES "Compras"."CompraMaestro" ("Secuencial"),
    CONSTRAINT fk_CompraDetalle_02 FOREIGN KEY ("SecuencialColorEstandar") REFERENCES
"Colores"."ColorEstandar" ("Secuencial"),
    CONSTRAINT fk_CompraDetalle_03 FOREIGN KEY ("CodigoSucursal") REFERENCES
"Generales"."Sucursales" ("Codigo")
);

```

**CREATE TABLE "Compras"."InformacionAdicional"**

```

(
    "SecuencialCompraMaestro" integer NOT NULL,
    "Subtotal" numeric(12,2) NOT NULL,
    "DescuentoPorcentaje" numeric(12,2) NOT NULL,
    "DescuentoValor" numeric(12,2) NOT NULL,
    "TotalNeto" numeric(12,2) NOT NULL,
    "OtroCargos" numeric(12,2) NOT NULL,
    "IvaPorcentaje" numeric(12,2) NOT NULL,
    "IvaValor" numeric(12,2) NOT NULL,
    "TotalCompra" numeric(12,2) NOT NULL,
    CONSTRAINT pk_InformacionAdicional PRIMARY KEY ("SecuencialCompraMaestro"),
    CONSTRAINT fk_InformacionAdicional_01 FOREIGN KEY ("SecuencialCompraMaestro")
REFERENCES "Compras"."CompraMaestro" ("Secuencial")
);

```

**CREATE TABLE "Negocios"."MovimientoEfectivo"**

```

(
    "Secuencial" serial NOT NULL,
    "ValorIngreso" numeric(12,2) NOT NULL,
    "ValorEgreso" numeric(12,2) NOT NULL,
    "EsIngreso" boolean NOT NULL,
    "NombreUsuario" character varying(255) NOT NULL,
    "FechaSistema" date NOT NULL,
    "FechaMaquina" date NOT NULL,
    "NumeroVerificador" integer NOT NULL,
    "ApplicationName" character varying(255),
    CONSTRAINT pk_MovimientoEfectivo PRIMARY KEY ("Secuencial"),
    CONSTRAINT fk_MovimientoEfectivo_01 FOREIGN KEY ("NombreUsuario",
"ApplicationName") REFERENCES "Users" ("Username", "ApplicationName")
);

```

**CREATE TABLE "Generales"."GeneraCodigo"**

```

(
    "Codigo" integer,
    "UltimoCodigoGenerado" integer,
    "Detalle" character varying(50),
    CONSTRAINT pk_GeneraCodigo PRIMARY KEY ("Codigo"));

```

**CREATE TABLE "Credito"."AuxTablaPagos"**

```
(  
  "CodigoFacturaMaestro" character varying(12),  
  "NumeroPago" integer,  
  "FechaPago" date,  
  "ValorPago" numeric(12,2)  
);
```

**CREATE TABLE "Generales"."OpcionesBusquedaMestro"**

```
(  
  "Codigo" integer NOT NULL,  
  "Descripcion" character varying(100) NOT NULL,  
  CONSTRAINT pk_OpcionesBusquedaMestro PRIMARY KEY ("Codigo")  
);
```

**CREATE TABLE "Generales"."OpcionesBusquedaDetalle"**

```
(  
  "Codigo" integer NOT NULL,  
  "CodigoOpcionesBusquedaMestro" integer NOT NULL,  
  "Nombre" character varying(50) NOT NULL,  
  "EstaActivo" boolean NOT NULL,  
  "NumeroVerificador" integer NOT NULL,  
  CONSTRAINT pk_OpcionesBusquedaDetalle PRIMARY KEY ("Codigo"),  
  CONSTRAINT fk_OpcionesBusquedaDetalle_01 FOREIGN KEY  
  ("CodigoOpcionesBusquedaMestro") REFERENCES "Generales"."OpcionesBusquedaMestro"  
  ("Codigo")  
);
```

**CREATE TABLE "Historicos"."HistoricoFecha"**

```
(  
  "Secuencial" serial NOT NULL,  
  "Fecha" date,  
  CONSTRAINT pk_hf PRIMARY KEY ("Secuencial"),  
  CONSTRAINT "unica_FechaHistorico" UNIQUE ("Fecha")  
);
```

**CREATE TABLE "Historicos"."HistoricoColorEstandar"**

```
(  
  "Secuencial" integer NOT NULL,  
  "Codigo" character varying(20) NOT NULL,  
  "Nombre" character varying(100) NOT NULL,  
  "Litro" integer NOT NULL,  
  "Gramos" numeric(12,2) NOT NULL,  
  "Precio" numeric(12,2) NOT NULL,  
  "EstaActivo" boolean NOT NULL,  
  "NumeroVerificador" integer NOT NULL,
```

```
"SecuencialHistoricosFecha" int NOT NULL,  
CONSTRAINT fk_HistoricoColorEstandar_01 FOREIGN KEY ("SecuencialHistoricosFecha")  
REFERENCES "Historicos"."HistoricoFecha" ("Secuencial")  
);
```

**CREATE TABLE "Historicos"."HistoricoSucursales"**

```
(  
"Codigo" character varying(10) NOT NULL,  
"Nombre" character varying(50) NOT NULL,  
"SecuencialHistóricosFecha" int NOT NULL,  
CONSTRAINT fk_Históricossucursales_01 FOREIGN KEY ("SecuencialHistóricosFecha")  
REFERENCES "Historicos"."HistoricoFecha" ("Secuencial")  
);
```

**CREATE TABLE "Historicos"."HistoricoColorEstandarSucursal"**

```
(  
"SecuencialColorEstandar" integer NOT NULL,  
"CodigoSucursal" character varying(10) NOT NULL,  
"CantidadEnLitros" numeric(12,2) NOT NULL,  
"SecuencialHistóricosFecha" int NOT NULL,  
CONSTRAINT fk_HistoricoColorEstandarSucursal_01 FOREIGN KEY  
("SecuencialHistoricosFecha") REFERENCES "Historicos"."HistoricoFecha" ("Secuencial")  
);
```

## ANEXO 5: EXPLICACIÓN DEL FUNCIONAMIENTO DEL CÓDIGO FUENTE

El sistema esta creado en dos capas: Cliente-Servidor

**Servidor:** Es la capa de acceso a datos (DAL), tiene una comunicación directa con la base de datos ya que contiene toda la sentencia SQL para poder ejecutarse en la base. Esta capa tiene una clase por cada tabla creada en la base de datos. Cada clase esta compuesta por: Librerias, Datos, Propiedades y Metodos. Los métodos serán llamados desde el cliente de acuerdo a los tipos de datos.

```
Clase: ColorEstandar.cs
        throw errorCSharp;
    }
    finally
    {
        cnBaseDatos.Close(); //Cerrar la conexion.
        cnBaseDatos.Dispose(); //Liberar recursos.
    }
}

public bool Insertar(ColorEstandar colorEstandar)
{
    bool insertoRegistro = false;
    NpgsqlConnection cnBaseDatos = new NpgsqlConnection();

    try
    {
        cnBaseDatos.ConnectionString = ConectarBD.CadenaDeConexion;

        string sentenciaInsert =
            " INSERT INTO \"Colores\".\"ColorEstandar\"(\"Codigo\", \"Nombre\",
            \"Litro\", \"Gramos\", \"Precio\", \"EstaActivo\", \"NumeroVerificador\") " +
            " VALUES (@codigo, @nombre, @litro, @gramos, @precio, @estaActivo,
            @numeroVerificador) " ;

        NpgsqlCommand cmdInsert = new NpgsqlCommand(sentenciaInsert,
cnBaseDatos);
        cmdInsert.CommandType = CommandType.Text;

        cmdInsert.Parameters.Add("@codigo", NpgsqlTypes.NpgsqlDbType.Varchar,
20).Value = colorEstandar.Codigo;
        cmdInsert.Parameters.Add("@nombre", NpgsqlTypes.NpgsqlDbType.Varchar,
100).Value = colorEstandar.Nombre;
        cmdInsert.Parameters.Add("@litro",
NpgsqlTypes.NpgsqlDbType.Integer).Value = colorEstandar.Litro;
        cmdInsert.Parameters.Add("@gramos",
NpgsqlTypes.NpgsqlDbType.Double).Value = colorEstandar.Gramos;
        cmdInsert.Parameters.Add("@precio",
NpgsqlTypes.NpgsqlDbType.Double).Value = colorEstandar.Precio;
        cmdInsert.Parameters.Add("@estaActivo",
NpgsqlTypes.NpgsqlDbType.Boolean).Value = colorEstandar.EstaActivo;
        cmdInsert.Parameters.Add("@numeroVerificador",
NpgsqlTypes.NpgsqlDbType.Integer).Value = colorEstandar.NumeroVerificador;

        //Abrir la coneccion
    }
}
```

continúa

```

        cnBaseDatos.Open();

        //Ejecutar el comando SQL contra la BD.
        int filasAfectadas = cmdInsert.ExecutesNonQuery();

        if (filasAfectadas == 1)
        {
            insertoRegistro = true;
        }
        return insertoRegistro;
    }
    catch (NpgsqlException errorPostgres)
    {
        throw errorPostgres;
    }
    catch (Exception errorCSharp)
    {
        throw errorCSharp;
    }
    finally
    {
        cnBaseDatos.Close(); //Cerrar la conexion.
        cnBaseDatos.Dispose(); //Liberar recursos.
    }
}

public bool Actualizar(ColorEstandar colorEstandar)
{
    bool actualizoRegistro = false;
    NpgsqlConnection cnBaseDatos = new NpgsqlConnection();

    try
    {
        cnBaseDatos.ConnectionString = ConectarBD.CadenaDeConexion;

        string sentenciaUpdate =
            " UPDATE \"Colores\".\"ColorEstandar\" " +
            " SET \"Codigo\"=@codigo, " +
            " \"Nombre\"=@nombre, " +
            " \"Litro\"=@litro, " +
            " \"Gramos\"=@gramos, " +
            " \"Precio\"=@precio, " +
            " \"EstaActivo\"=@estaActivo " +
            " WHERE \"Secuencial\"=@secuencial ";

        NpgsqlCommand cmdUpdate = new NpgsqlCommand(sentenciaUpdate,
cnBaseDatos);
        cmdUpdate.CommandType = CommandType.Text;

        cmdUpdate.Parameters.Add("@secuencial",
NpgsqlTypes.NpgsqlDbType.Integer).Value = colorEstandar.Secuencial;
        cmdUpdate.Parameters.Add("@codigo", NpgsqlTypes.NpgsqlDbType.Varchar,
20).Value = colorEstandar.Codigo;
        cmdUpdate.Parameters.Add("@nombre", NpgsqlTypes.NpgsqlDbType.Varchar,
100).Value = colorEstandar.Nombre;
        cmdUpdate.Parameters.Add("@litro",
NpgsqlTypes.NpgsqlDbType.Integer).Value = colorEstandar.Litro;
        cmdUpdate.Parameters.Add("@gramos",
NpgsqlTypes.NpgsqlDbType.Double).Value = colorEstandar.Gramos;
        cmdUpdate.Parameters.Add("@precio",
NpgsqlTypes.NpgsqlDbType.Double).Value = colorEstandar.Precio;
        cmdUpdate.Parameters.Add("@estaActivo",
NpgsqlTypes.NpgsqlDbType.Boolean).Value = colorEstandar.EstaActivo;

        //Abrir la coneccion
        cnBaseDatos.Open();

        //Ejecutar el comando SQL contra la BD.

```

continúa



```

        tbNombre.Text = dtColorEstandar.Rows[0]["Nombre"].ToString();
        tbLitro.Text = dtColorEstandar.Rows[0]["Litro"].ToString();
        tbGramos.Text = dtColorEstandar.Rows[0]["Gramos"].ToString();
        tbPrecio.Text = dtColorEstandar.Rows[0]["Precio"].ToString();
        cbEstaActivo.Checked =
Convert.ToBoolean(dtColorEstandar.Rows[0]["EstaActivo"].ToString());
    }
}
catch (NpgsqlException errorPostgres)
{
    lblMensaje.Text = "Error Postgres, <br />" + errorPostgres.Message + "<br
/>" + errorPostgres.StackTrace;
}
catch (Exception errorCSharp)
{
    lblMensaje.Text = "Error CSharp, <br />" + errorCSharp.Message + "<br />"
+ errorCSharp.StackTrace;
}
}
#endregion

#region Metodos DML
public void InsertarColorEstandar()
{
    try
    {
        ColorEstandar colorEstandar = new ColorEstandar();
        colorEstandar.Codigo = tbCodigo.Text;
        colorEstandar.Nombre = tbNombre.Text;
        colorEstandar.Litro = Convert.ToInt32(tbLitro.Text);
        colorEstandar.Gramos = Convert.ToDouble(tbGramos.Text);
        colorEstandar.Precio = Convert.ToDouble(tbPrecio.Text);
        colorEstandar.EstaActivo = cbEstaActivo.Checked;
        if (colorEstandar.Insertar(colorEstandar))
        {
            lblMensaje.Text = "Color Estandar Ingresado con Exito";
            Response.Redirect("VistaColorEstandar.aspx");
        }
        else
        {
            lblMensaje.Text = "Error al Ingresar Color Estandar";
        }
    }
    catch (NpgsqlException errorPostgres)
    {
        lblMensaje.Text = "Error Postgres, <br />" + errorPostgres.Message + "<br
/>" + errorPostgres.StackTrace;
    }
    catch (Exception errorCSharp)
    {
        lblMensaje.Text = "Error CSharp, <br />" + errorCSharp.Message + "<br />"
+ errorCSharp.StackTrace;
    }
}
public void ActualizarColorEstandar()
{
    try
    {
        ColorEstandar colorEstandar = new ColorEstandar();
        colorEstandar.Secuencial = Convert.ToInt32(tbSecuencial.Text);
        colorEstandar.Codigo = tbCodigo.Text;
        colorEstandar.Nombre = tbNombre.Text;
        colorEstandar.Litro = Convert.ToInt32(tbLitro.Text);
        colorEstandar.Gramos = Convert.ToDouble(tbGramos.Text);
        colorEstandar.Precio = Convert.ToDouble(tbPrecio.Text);
        colorEstandar.EstaActivo = cbEstaActivo.Checked;
        if (colorEstandar.Actualizar(colorEstandar))
        {

```

continúa

```

        lblMensaje.Text = "Color Estandar Actualizado con Exito";
    }
    else
    {
        lblMensaje.Text = "Error al Actualizar Color Estandar";
    }
}
catch (NpgsqlException errorPostgres)
{
    lblMensaje.Text = "Error Postgres, <br />" + errorPostgres.Message + "<br
/>" + errorPostgres.StackTrace;
}
catch (Exception errorCSharp)
{
    lblMensaje.Text = "Error CSharp, <br />" + errorCSharp.Message + "<br />"
+ errorCSharp.StackTrace;
}
}
#endregion

protected void Page_Load(object sender, EventArgs e)
{
    try
    {
        lblMensaje.Text = "";
        if (!Page.IsPostBack)
        {
            if (ClaseEstatica.EsNuevoColorEstandar)
            {
                tbSecuencial.Text = "0";
                tbLitro.Text = "1";
                cbEstaActivo.Checked = true;
                cbEstaActivo.Enabled = false;
            }
            else //es edicion
            {
                tbSecuencial.Text = ClaseEstatica.SecuencialColorEstandar;
                ObtenerDatosColorEstandar();
            }
        }
    }
    catch (NpgsqlException errorPostgres)
    {
        lblMensaje.Text = "Error Postgres, <br />" + errorPostgres.Message + "<br
/>" + errorPostgres.StackTrace;
    }
    catch (Exception errorCSharp)
    {
        lblMensaje.Text = "Error CSharp, <br />" + errorCSharp.Message + "<br />"
+ errorCSharp.StackTrace;
    }
}

protected void ibtnGuardar_Click(object sender, ImageClickEventArgs e)
{
    try
    {
        ColorEstandar colorEstandar = new ColorEstandar();
        DataTable dtColorEstandar = new DataTable();
        dtColorEstandar =
colorEstandar.SeleccionarPorSecuencial(Convert.ToInt32(tbSecuencial.Text));
        if (dtColorEstandar.Rows.Count == 1)
        {
            //Es edicion
            ActualizarColorEstandar();
        }
        else
        {
            //Es nuevo ingreso

```

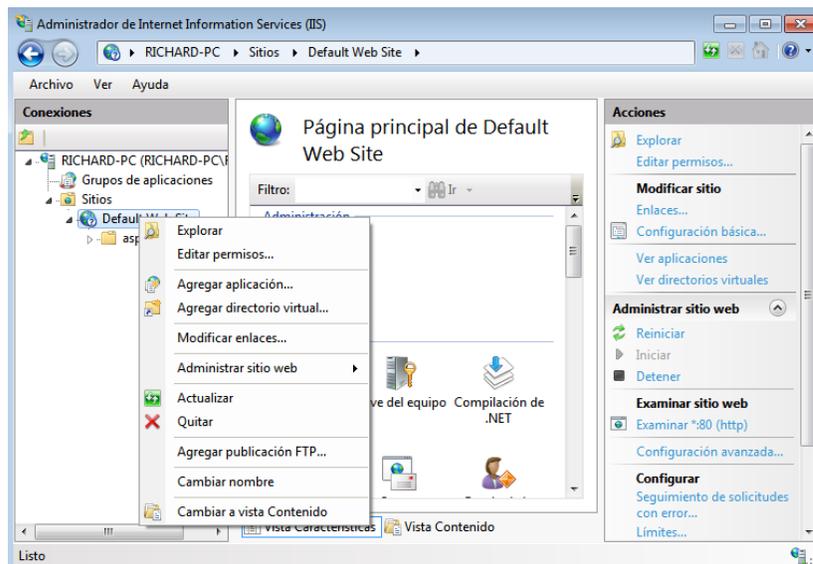
continúa

```
        InsertarColorEstandar();
    }
}
catch (NpgsqlException errorPostgres)
{
    lblMensaje.Text = "Error Postgres, <br />" + errorPostgres.Message + "<br />" + errorPostgres.StackTrace;
}
catch (Exception errorCSharp)
{
    lblMensaje.Text = "Error CSharp, <br />" + errorCSharp.Message + "<br />" + errorCSharp.StackTrace;
}

}
protected void ibtnAtras_Click(object sender, ImageClickEventArgs e)
{
    Response.Redirect("VistaColorEstandar.aspx");
}
}
```

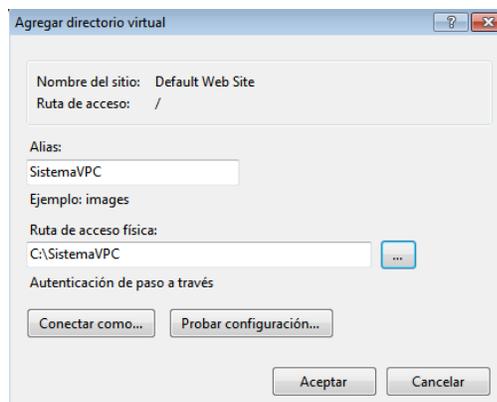
## ANEXO 6: INSTALACIÓN DEL SISTEMA

- En el panel de control, en Activar o desactivar las características de Windows, seleccionar *Internet Information Services*.
- Para abrir el Administrador de Internet Information Services, en Ejecutar escribimos *inetmgr*.
- En el administrador de Internet Information Services agregamos un directorio virtual.



**Figura A6.1** Agregar Directorio Virtual

- Escribimos el Alias y la Ruta de acceso física (La ruta contendrá la carpeta con los archivos de publicación del las fuentes)



**Figura A6.2** Alias y Ruta de Acceso Física

- Convertimos en Aplicación

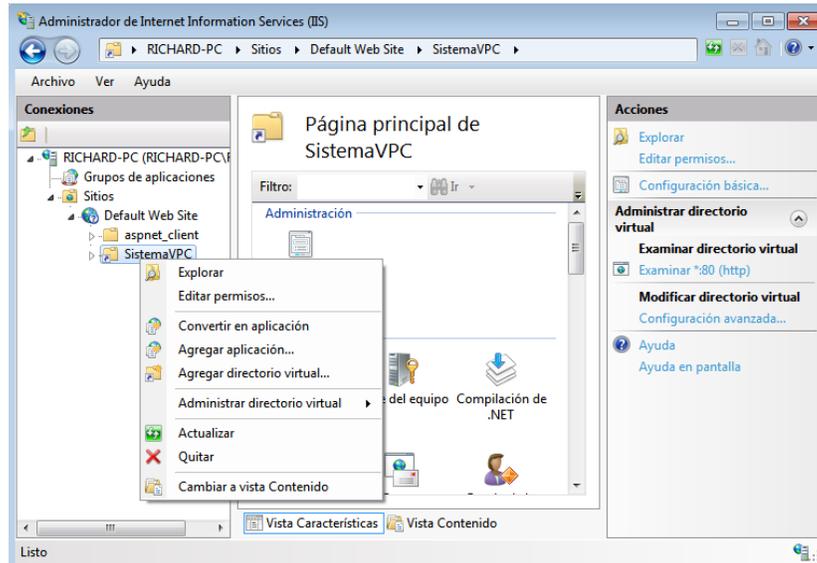


Figura A6.3 Convertir en Aplicación

### Configuración web.config

Este paso es importante para el funcionamiento del sistema, para lograr la comunicación con la base de datos configuraremos la cadena de conexión que se encuentra dentro de este archivo, para lo cual podemos utilizar cualquier editor de archivos planos.

```
<connectionStrings>
  <add name="conexion"
    connectionString="DATABASE=FactInvePPG;SERVER=localhost;PORT=54
32;User
Id=postgres;Password=latino;Encoding=UNICODE;Sslmode=Prefer;Pool
ing=true;"/>
</connectionStrings>
```

## ANEXO 7: Manual de Usuario

Esta documentación servirá al cliente como una guía práctica para la correcta utilización del sistema facilitando su comprensión y manejo de la misma.

Es recomendable que el usuario del sistema tenga conocimientos básicos de computación, para evitar cualquier inconveniente es necesario leer este manual antes de empezar a operar sobre la misma.

El diseño de la interfaz consta de lo siguiente:

1. Logo Empresa
2. Nombre Empresa
3. Nombre Sistema
4. Mapa de Sitio
5. Menú
6. Iniciar Sesión – Cerrar Sesión
7. Información Usuario
8. Espacio de Trabajo

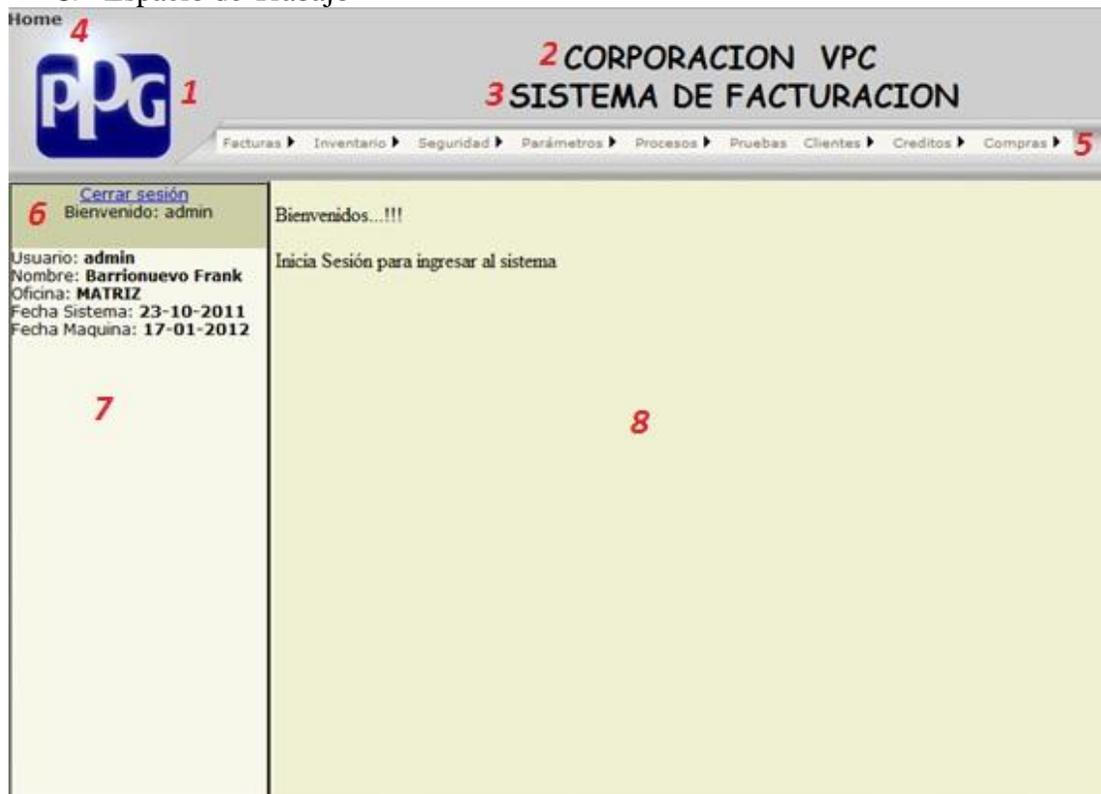
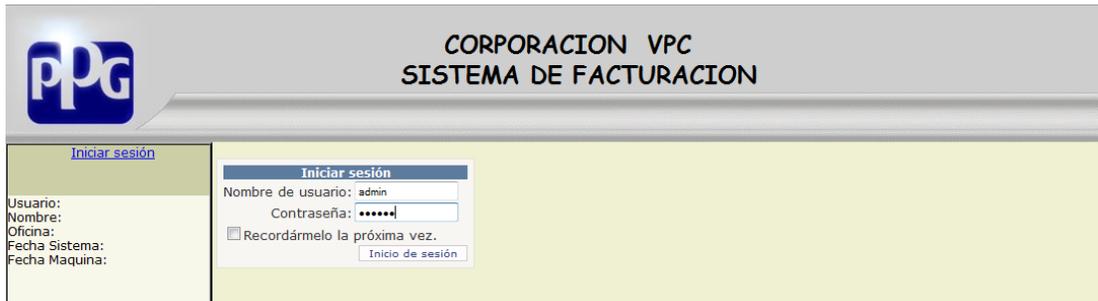


Figura A7.1 Portada Inicial

## Pantalla Inicio de Sesión

Ingreso al sistema y una vez pasado la validación podrá realizar diferentes actividades de acuerdo a los privilegios (roles) que tenga.



**Figura A7.2** Formulario Inicio Sesión

## Iconos que aparecerán en la mayoría de las pantallas.

Al pasar el mouse por el icono le mostrara cual es la función de la misma, en la mayoría realizan los mismos procesos. Los iconos más comunes que encontraremos en el sistema son los siguientes:

-  Nuevo registro.
-  Ver registros.
-  Buscar.
-  Eliminar.
-  Guardar registro.
-  Actualizar.
-  Imprimir.
-  Editar.
-  Procesar.
- 

## Facturas

Podrá ver todas las facturas ingresadas, se podrá filtrar por Sucursal, Fecha y Estado (Ingresadas y Pendientes).

1. Opciones de factura: Nuevo, Ver, Buscar y Eliminar
2. Filtros e información de Factura.
3. Listado de las Facturas ingresadas.



Figura A7.3 Formulario Vista Facturas

### Crear Nueva Factura

Podrá ingresar una nueva Factura, el desglose de cada detalle será calculado de acuerdo a la parametrización de los Colores Estándar y la Generación de Color.

1. Cabecera de la Factura, contienen los botones (...) para la búsqueda de clientes.
2. Información Adicional para el SRI, donde el secuencial será automático o manual y junto con los demás datos serán parametrizables.
3. Detalle de la Factura, con los cálculos incluido el descuento.
4. Opciones para la administración del ingreso del detalle, donde se podrá buscar por Código (en caso de varios colores permitirá una selección) o Nombre.
5. Desglose, colores que se utilizaron para la creación de los colores del detalle.
6. Ingreso de la forma de pago. (Cheque, Efectivo y Crédito). Si es por Crédito el sistema creará automáticamente un sistema de Crédito donde el cliente pagará sus distintas letras, en diferentes fechas.
7. Con este botón realiza los cálculos faltantes, guarda la factura definitiva, actualiza el stock, imprime el documento de la Factura, si el pago es a crédito genera automáticamente una modalidad de crédito.

**7**

Código:	VEROS	Fecha:	24-10-2011	<b>FACTURA</b> 1891732593001 001-001 0000010 1108267976
Identificación:	1803505948	Dirección:	Centro	
Nombre:	Veronica Barrionuevo	Teléfono:	03458723 <b>1</b>	

**2**

---

**DETALLE**

Código: BS 2      Nombre: Blanco Saint 2      Cantidad: 500 L      Descuento: 0 % **4**

Cantidad	TipoCant	Codigo	Nombre	Precio	Subtotal	Descuento	Total	Secuencial
500,00	G	BS 2	Blanco Saint 2	0,03	12,74	0,00	12,74	92

Comentario:

**3**

Subtotal:	12,74	12,74
Dto: %	0	0
Iva 12%:	1,53	1,53
<b>Total:</b>	<b>14,27</b>	<b>14,27</b>

---

DESGLOSE					FORMA PAGO	
Codigo	Nombre	Cantidad	Precio	Subtotal		
DUL 603	Clear Bicope <b>5</b>	207,567218	0,02	3,73	<input checked="" type="radio"/> Cheque	Valor: 14,27
DUL 637	Branco Puro	290,584866	0,03	8,94	<input type="radio"/> Credito <b>6</b>	Banco: Pichincha
DUL 690	Liso Matalico Azul	0,600573	0,04	0,02	<input type="radio"/> Efectivo	Cuenta: 123456
DUL 692	Liso Matalico Preto	1,247344	0,04	0,05		Cheque: 654321
Subtotal:				12,74		

**Figura A7.4** Formulario Nueva Factura

### Consulta Factura Ingresada

Contiene los datos de una Factura creada con anterioridad, muestra lo siguiente:

1. Cabecera de la Factura.
2. Información adicional para el SRI.
3. Detalle de la Factura.
4. Desglose del detalle de la Factura.
5. Formas de pago que se realizó en la Factura.
6. Reimpresión de la Factura

**Consulta Factura**

Código:	FRANK	Fecha:	23-10-2011	<b>FACTURA</b> <b>1803505963001</b> <b>001-001</b> <b>0000005</b> <b>1108267976</b>
Identificación:	1803505963	Dirección:	CENTRO	
Nombre:	FRANKLIN BARRIONUI	Teléfono:	095914148	

**DETALLE**

	Cantidad	Codigo	Tipo	Cantidad	Codigo	Nombre	Precio	Subtotal	Descuento	Total	Secuencial	CodigoFacturaMaestro	
Seleccionar	10,00	L			G0001	Gris	0,89	8,86	0,09	8,77	9	201110234	
Comentario:											Subtotal:	8,86	8,77
											Dcto:	0,18	0,18
											Iva 12%:	1,06	1,05
											Total:	9,74	9,64

**DESGLOSE**

Secuencial	FacturaDetalle	Codigo	Nombre	Cantidad	Precio	Subtotal
9		B0001	Blanco	2,80	1,20	3,36
9		N0001	Negro	5,00	1,10	5,50
Subtotal:						8,86

**FORMA PAGO**

Factura	Pago	Valor
201110234	CHE	9,64

**Figura A7.5** Formulario Consulta Factura Ingresada

## Buscar Factura

Podrá buscar factura por Numero Factura, Nombre, Código e Identificación, también se podrá filtrar por Sucursal, Fecha y Estado (Ingresado y Pendiente).

1. Opciones de la Factura
2. Opciones de Búsqueda
3. Filtros e información
4. Listado de Facturas, resultado de la búsqueda.

**Facturas**

Numero Factura
  Nombre
  Código
  Identificación

Código: FRANK

Sucursal: MATRIZ    Fecha: 23-10-2011    Estado: Ingresado    Factura Seleccionada: 201110234

	Factura	Nombre	Codigo	Identificación	Subtotal	Descuento	Iva	Total
Seleccionar	201110231	FRANKLIN BARRIONUEVO	FRANK	1803505963	8,77	0,09	1,05	9,73
Seleccionar	201110234	FRANKLIN BARRIONUEVO	FRANK	1803505963	8,77	0,18	1,05	9,64

**Figura A7.6** Formulario Buscar Factura



## INVENTARIOS

### Vista del Inventario

Muestra el inventario filtrado por Sucursal y Color Estándar:

1. Filtros para la consulta.
2. Resultado de la consulta.
3. Ejecutar la consulta.

CódigoSucursal	Sucursal	CódigoColor	Color	Precio	Cantidad
S001	MATRIZ	B0001	Blanco	1,20	82,60
S004	PELILEO	B0001	Blanco	1,20	10,00
S003	SALCEDO	B0001	Blanco	1,20	400,00
S002	QUITO	B0001	Blanco	1,20	389,00
S001	MATRIZ	R0001	Rojo	2,22	305,40
S003	SALCEDO	R0001	Rojo	2,22	621,00
S002	QUITO	R0001	Rojo	2,22	700,00
S003	SALCEDO	A0001	Azul	0,99	435,00
S002	QUITO	A0001	Azul	0,99	432,00
S001	MATRIZ	A0001	Azul	0,99	329,00
S001	MATRIZ	V0001	Verde	1,35	183,60
S003	SALCEDO	V0001	Verde	1,35	398,00
S002	QUITO	V0001	Verde	1,35	576,00
S001	MATRIZ	N0001	Negro	1,10	126,00
S003	SALCEDO	N0001	Negro	1,10	456,00
S002	QUITO	N0001	Negro	1,10	500,00

Figura A7.10 Formulario Inventario

### Reporte de Inventario

Muestra la Sucursal y el Color Estándar que serán utilizados como filtro para el reporte.

1. Muestra el detalle de cada color generado
2. Muestra los colores que se utilizo para generar el nuevo color

Sucursal: MATRIZ    Color Estándar: TODOS

Figura A7.11 Formulario Reporte de Inventario

## Resultado del Reporte de Inventario

**CORPORACION VPC**  
 REPORTE INVENTARIO

Sucursal: MATRIZ Color: TODOS

COD.SUCURSAL	SUCURSAL	COD.COLOR	COLOR	PRECIO	CANTIDAD
S001	MATRIZ	B0001	Blanco	1,20	82,80
S001	MATRIZ	R0001	Rojo	2,22	305,40
S001	MATRIZ	A0001	Azul	0,99	329,00
S001	MATRIZ	V0001	Verde	1,35	183,60
S001	MATRIZ	N0001	Negro	1,10	126,00

**Figura A7.12** Formulario resultado del Reporte de Inventario

## Reporte de Históricos de Inventario

Mediante filtros como la Fecha, la Sucursal, y el Color Estándar se podrá obtener el estado del inventario de una fecha específica.

1. Parametros para el reporte de Históricos de Inventario
2. Ejecutar el reporte

**Figura A7.13** Formulario Reporte de Históricos de Inventario

## Resultado del reporte de Históricos de Inventario

**CORPORACION VPC**  
 REPORTE INVENTARIO

Sucursal: TODOS Color: Blanco Fecha: 22-10-2011

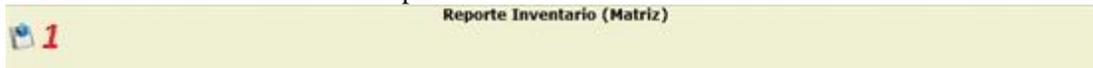
COD.SUCURSAL	SUCURSAL	COD.COLOR	COLOR	PRECIO	CANTIDAD
S002	QUITO	B0001	Blanco	1,20	389,00
S003	SALCEDO	B0001	Blanco	1,20	400,00
S001	MATRIZ	B0001	Blanco	1,20	25,40
S004	PELILEO	B0001	Blanco	1,20	10,00

**Figura A7.14** Formulario resultado del reporte de Históricos de Inventario

## Reporte de Invenrario resumida en una matriz

Muestra el botón para ejecutar el reporte.

1. Encabezado de la Cuenta por Cobrar



**Figura A7.15** Formulario reporte de Invenrario resumida en una matriz

## Resultado del reporte de Inventario resumida en una matriz

Muestra el reporte con los filtros del formulario anterior, este reporte muestra la información mas ordenada dentro de una matriz, mostrando los Colores vs las Sucursales.

CORPORACION VPC					
REPORTE INVENTARIO MATRIZ					
Color	MATRIZ	QUITO	SALCEDO	PELILEO	Total
Bianco	82,6	389	400	10	881,6
Rojo	305,4	700	621	0	1626,4
Azul	329	432	435	0	1196
Verde	183,6	578	398	0	1157,6
Prueba1	0	0	0	0	0
Prueba2	0	0	0	0	0
Prueba3	0	0	0	0	0
Prueba4	0	0	0	0	0
Negro	128	500	456	0	1082

**Figura A7.16** Formulario resultado del reporte de Inventario resumida en una matriz

## COMPRAS

### Vista de compras Ingresadas

Muestra todas las compras ingresadas no procesadas, una vez procesada afectara el stock del inventario y ya no aparecerá en este listado.

1. Opciones de compra: Nuevo, Editar y Procesar
2. Vista de compras ingresadas, una vez proceadas ya no se verán en este listado.

	Secuencial	Identificacion	Nombre	Total	Fecha	Estado
Seleccionar	3	1803505963001	VER PAINT CORP	200,00	22-10-2011	Ingresado
Seleccionar	5	1803505963001	VER PAINT CORP	0,00	22-10-2011	Ingresado
Seleccionar	6	1803505963001	VER PAINT CORP	120,00	23-10-2011	Ingresado

**Figura A7.17** Formulario Vista de compras Ingresadas

## Administración de Compras

Se podrá ingresar una nueva compra, adicionalmente se podrá realizar cálculos adicionales.

1. Información de la cabecera de la compra.
2. Botón para la búsqueda del proveedor, se puede buscar por Código o por Identificación.
3. Detalle de la Compra.
4. Opciones para el detalle de la compra (Guardar, Blanquear y Eliminar).
5. Cálculos adicionales no obligados para todas las compras.
6. Botón para guardar la compra

Administración Compra

Secuencial: 7  
Proveedor: 1803505963001  Ident. >> VER PAINT CORP  
Fecha: 23-10-2011  
Estado: Ingresando  
Total: 120

Detalle

Secuencial: 7 Cantidad(L): 12.00  
Color Estandar: V0001 - Verde Costo: 10.00  
Sucursal: MATRIZ Total: 120.00

Sec	SecCompra	SecColor	Color	CodSucursal	Sucursal	CantidadEnLitros	Precio	Total
7	7	6	Verde	S001	MATRIZ	12,00	10,00	120,00

Calculos Adicionales Sub Total: 120  
Guardar Descuento: 2 % 2.4  
Total Neto: 117.6  
Otros Cargos: 0  
Iva: 12%  - 0% 14,112  
Total Compra: 131.712

Figura A7.18 Formulario Administración de Compras

## Consulta de Compras Procesadas

Muestra una lista de todas las compras que han sido procesadas filtradas por fechas, estas compras ya no se podrán editar por ya afectaron el stock del inventario.

1. Fecha en la que se realizó la compra.
2. Opciones: Procesar consulta y Ver compra procesada.
3. Listado de compras procesadas.

Compras Procesadas

Fecha: 22-10-2011

Secuencial	Identificación	Nombre	Total	Fecha	Estado
2	1803505963001	VER PAINT CORP	200,00	22-10-2011	Procesado
4	1803505963001	VER PAINT CORP	50,00	22-10-2011	Procesado

**Figura A7.19** Formulario Consulta de Compras Procesadas.

### Ver Compra Procesada

Muestra los detalles de una compra que fue procesada, esta compra ya no será editable.

1. Encabezado de la compra.
2. Detalle de la compra.
3. Información adicional, no obligada en todas las compras.

The screenshot shows a web application interface for viewing a processed purchase. The title is 'Consulta Compra'. The form contains the following fields and data:

- Secuencial: 4
- Proveedor: 1803505963001
- Fecha: 22-10-2011
- Estado: Procesado
- Total: 50,00

A table with the following columns: Sec, SecCompra, SecColor, Color, CodSucursal, Sucursal, CantidadEnLitros, Precio, Total. The data row is: 4, 4, 1, Blanco, S001, MATRIZ, 5,00, 10,00, 50,00.

Summary fields:

- Sub Total: 50,00
- Descuento: 2,00 % (1,00)
- Total Neto: 49,00
- Otros Cargos: 0,00
- Iva: 12% (checked) - 0% (unchecked) (5,88)
- Total Compra: 54,88

Red numbers 1, 2, and 3 are overlaid on the image to indicate the sections described in the list above.

**Figura A7.20** Formulario Ver Compra Procesada

## CREDITOS

### Vista Cobro de Crédito

Muestra un resumen del crédito que fue generado por la compra de una Factura con la forma de pago Crédito, detallara el valor que tendrá que pagar el cliente.

1. Informacin del Cliente
2. Informacion del Crédito
3. Detalle del Pago
4. Forma de pago

**Cobro de Credito**

Código:  Identificación:  Nombre:  **1**  
 Cliente: FRANK      1803505963     

Secuencial	CodigoFacturaMaestro	Monto	SaldoActual	FechaCreacion	CodigoEstadoCredito
Seleccionar 2	201110221	20,00	12,00	22-10-2011	A

**2**

Valor para estar al Día:       Usuario:   
 A pagar hasta cuota actual:       Sucursal:  **3**  
 Valor para cancelar:       Ultimo pago:

Cheque **4**  
 Efectivo

**Figura A7.21** Formulario Vista Cobro de Crédito

### Recibo del cobro del Crédito

Nombre: FRANKLIN BARRIONUEVO	Fecha Sistema: 23-10-2011
Identificación: 1803505963	Fecha Maquina: 17-01-2012
Sucursal: MATRIZ	Usuario: admin
Saldo Anterior: 12	Credito #: 2
Valor Pago: 0,5	
Saldo Actual: 11,5	

**Figura A7.22** Formulario Recibo del cobro del Crédito

### Consulta de Cobros de Crédito

Muestra la información del préstamo junto con los movimientos y los componetes.

1. Datos del Cliente.
2. Informacion del Crédito.
3. Informacion del próximo pago.
4. Detalle de los movimientos.
5. Detalle de los componentes.

**Consulta de Creditos**

Codigo:  Identificacion:  Nombre:

Cliente: FRANK 1803505963 FRANKLIN BARRIONUI 1

Secuencial	CodigoFacturaMaestro	Monto	SaldoActual	FechaCreacion	CodigoEstadoCredito
2	201110221	20,00	11,50	22-10-2011	A

Secuencial Credito:  Estado:

Valor para estar al Dia:  Usuario:

A pagar hasta cuota actual:  Sucursal:

Valor para cancelar:  Ultimo pago:

2

Movimientos

CodigoSucursal	NombreUsuario	Fecha	Valor	Saldo
S001	admin	11-01-2012	7,00	15,00
S001	admin	16-01-2012	1,00	14,00
S001	admin	16-01-2012	1,00	13,00
S001	admin	16-01-2012	0,50	12,50
S001	admin	16-01-2012	0,50	12,00
S001	admin	17-01-2012	0,50	11,50

4

Componentes

NumeroCuota	ValorCalculado	ValorCabrado	FechaInicio	FechaVencimiento	Estado
1	5,50	5,50	22-10-2011	21-11-2011	C
2	5,50	5,00	21-11-2011	21-12-2011	P
3	5,50	0,00	21-12-2011	20-01-2012	P
4	5,50	0,00	20-01-2012	19-02-2012	P

5

**Figura A7.23** Formulario Consulta de Cobros de Crédito

## CLIENTES

### Vista Clientes

Muestra el listado de los clientes registrados en el sistema.

1. Opciones de Clientes: Nuevo, Editar, Buscar y Ver
2. Listado de Clientes

**Clientes**

1

Secuencial	Codigo	Identificacion	Nombre	Ciudad	Vendedor
3	VEROS	1803505948	Veronica	Ambato	Vendedor Pruebas2
2	FRANK	1803505963	FRANKLIN BARRIONUEVO	Ambato	Vendedor Pruebas2

2

**Figura A7.24** Formulario Vista Clientes

### Administración de Clientes

Muestra la información de los clientes.

1. Opciones: Guardar Cliente y Regresar a la Vista Clientes.
2. Datos del Cliente.
3. Telefono y las formas de venta permitidas para los Clientes.

**Administrar Cliente**

Tipo Identificación: Cedula  
 Identificación: 1803505948  
 Código: VEROS  
 Ciudad: Tungurahua  
 Nombre: Veronica  
 Dirección: Barrionuevo  
 Correo: vero@yahoo.es  
 Vendedor: Vendedor Pruebas2

Seleccionar	Secuencial	Operadora	NumeroTelefono
3	CNT		03458723

**Figura A7.25** Formulario Administración de Clientes

### Ver Cliente

Muestra la información de los clientes, en este formulario no se podrá editar los datos.

1. Datos del cliente.
2. Telefonos y Formas de venta de los Clientes.
3. Regresar a la pagina Vista Clientes.

**Información Cliente**

Tipo Identificación: Cedula  
 Identificación: 1803505948  
 Código: VEROS  
 Ciudad: Ambato  
 Nombre: Veronica  
 Dirección: Barrionuevo  
 Correo: vero@yahoo.es  
 Vendedor: Vendedor Pruebas2

Forma Venta
<input checked="" type="checkbox"/> Cheque
<input checked="" type="checkbox"/> Credito
<input checked="" type="checkbox"/> Efectivo

**Figura A7.26** Formulario Ver Cliente

## Buscar Cliente

Se podrá buscar a un cliente por el código, la identificación, el nombre, la ciudad y vendedor.

1. Opciones del cliente
2. Opciones para la búsqueda del Cliente
3. Listado de Clientes, resultado de la búsqueda

	Secuencial	Codigo	Identificación	Nombre	Ciudad	Vendedor
Seleccionar	3	VEROS	1803505948	Veronica	Ambato	Vendedor Pruebas2
Seleccionar	2	FRANK	1803505963	FRANKLIN BARRIONUEVO	Ambato	Vendedor Pruebas2

Figura A7.27 Formulario Buscar Cliente

## COLORES

### Vista Colores Estándar

Muestra los colores ingresados.

1. Listado de los Colores Estándar existentes.
2. Opciones para la administración de Colores Estándar (Nuevo, Editar, Buscar y Ver uno existente).

	Secuencial	Codigo	Nombre	Litro	Gramos	Precio
Seleccionar	1	B0001	Blanco	1	453,60	1,20
Seleccionar	5	R0001	Rojo	1	453,87	2,22
Seleccionar	4	A0001	Azul	1	356,65	0,99
Seleccionar	6	V0001	Verde	1	453,65	1,35
Seleccionar	7	P0001	Prueba1	1	459,23	1,89
Seleccionar	8	P0002	Prueba2	1	459,54	2,03
Seleccionar	9	P0003	Prueba3	1	459,87	1,54
Seleccionar	10	P0004	Prueba4	1	479,54	2,43
Seleccionar	2	N0001	Negro	1	528,50	1,10

Figura A7.28 Formulario Vista Colores Estándar

## Administración de Color Estándar

Muestra información de los colores que han sido ingresados al sistema.

1. Espacio para ingresar un nuevo Color o editar uno existente.
2. Botones Guardar Color y regresar al formulario anterior.



Administración de Colores

Secuencial: 1  
Codigo: B0001  
Nombre: Blanco 1  
Litro: 1  
Gramos: 453,60  
Precio: 1,20  
Esta Activo:

**Figura A7.29** Formulario Administración de Color Estándar

## Información Color

Muestra los detalles del color, en este formulario no se podrán editar, solo ver.

1. Información del color seleccionado, estos campos no son editables.
2. Botón para regresar al formulario anterior.



Información Color

Secuencial: 1  
Codigo: B0001  
Nombre: Blanco 1  
Litro: 1  
Gramos: 453,60  
Precio: 1,20  
Esta Activo:

**Figura A7.30** Formulario Información Color

## Búsqueda Color

Permite la búsqueda de uno o varios colores ingresados al sistema, filtrado por Código o Nombre.

1. Listado de colores, resultado de la búsqueda.
2. Opciones para la búsqueda, por Código y Nombre
3. Opciones de Administración de Colores

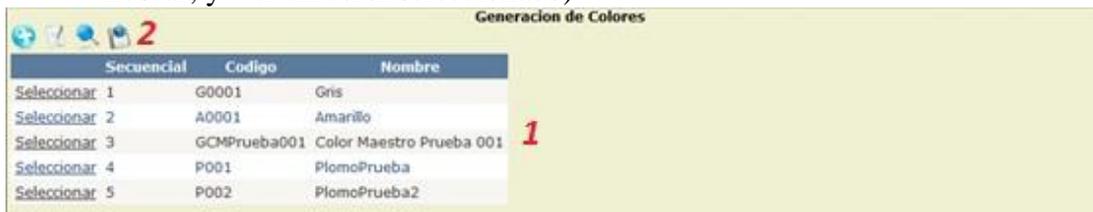


**Figura A7.31** Formulario Búsqueda Color

### Vista Generación de Colores

Muestra la mezcla de un nuevo color que se ha generado mediante la combinación de los colores estándar.

1. Listado de los Colores Generados existentes.
2. Opciones para la Administración de un Color Generado (Nuevo, Editar, Buscar, y Ver un color seleccionado).



**Figura A7.32** Formulario Vista Generación de Colores

### Administración de Colores Generados

Podemos ver los datos del color generado junto con el detalle en la que estarán los colore estándar.

1. Botones Guardar un Color Generado y regresar al formulario anterior.
2. Espacio para ingresar un nuevo Color Generado o editar uno existente.
3. Opciones para la Administración del detalle de un Color Generado (Nuevo, Guardar, Blanquear y Eliminar).
4. Listado del detalle del Color Generado.
5. Suma total de las cantidades en litros y gramos.

Administrar Generacion de Colores

Secuencial: 1

Codigo: G0001

Nombre: Gris

Esta Activo:

Seleccionar	Secuencial	Nombre	CantidadEnLitros	CantidadEnGramos
1	1	Blanco	0,28	127,01
2	2	Negro	0,50	226,58

Total Litros: 0,78

Total Gramos: 353,59

**Figura A7.33** Formulario Administración de Colores Generados

### Información de Colores Generados

Podemos ver los datos del color generado junto con el detalle, en este formulario no se podrá editar datos.

1. Información de la cabecera del Color Generado, estos datos no serán editables.
2. Listado del detalle que se utilizo para generar el Color
3. Boton para regresar al formulario anterior.

Información Generacion Color

Secuencial: 1

Codigo: G0001

Nombre: Gris

Esta Activo:

Seleccionar	Secuencial	Nombre	CantidadEnLitros	CantidadEnGramos
1	1	Blanco	0,28	127,01
2	2	Negro	0,50	226,58

**Figura A7.34** Formulario Información de Colores Generados

### Búsqueda de los Colores Generados

Podremos buscar uno o vario colores, dependiendo del filtro que se ingrese.

1. Opciones para la Administración de Colores Generados
2. Opciones para el filtro para la búsqueda.
3. Listado de los Colores Generados, resultado de la búsqueda.

Generacion de Colores

Codigo  Nombre

G0001 Aceptar

Seleccionar	Secuencial	Codigo	Nombre
1	1	G0001	Gris

**Figura A7.35** Formulario Búsqueda de los Colores Generados

## GENERALES

### Procesos Internos

El sistema realizará procesos automáticos internamente.

1. Botón procesar (Se ejecutara varias cosas como revisar los estados de los créditos, guardar la información necesaria en históricos, cambiar la fecha del sistema).
2. Información de los Procesos Internos que se han ejecutado.



**Figura A7.36** Formulario Procesos Internos

### Cambio de Clave

Aquí podremos cambiar nuestra clave personal.

1. Ingreso y la clave anterior y confirmación de la nueva clave
2. Botones para guardar la nueva contraseña o Cancelar



**Figura A7.37** Formulario Cambio de Clave

### Administración de Sucursales

Muestra las sucursales y nos permitirá su administración.

1. Opciones para la administración de las sucursales (Nuevo, Editar, Guardar e Información Adicional).
2. Campos para el ingreso de nueva sucursal o modificación.
3. Listado de las sucursales Ingresadas.



Administración Sucursales

Codigo: S001

Nombre: MATRIZ

	Codigo	Nombre
Seleccionar	S001	MATRIZ
Seleccionar	S002	QUITO
Seleccionar	S003	SALCEDO
Seleccionar	S004	PELILEO

**Figura A7.38** Formulario Administración de Sucursales

### Información Adicional de las Sucursales

Nos permite administrar información adicional proporcionada por el SRI.

1. Campos para el Ingreso o Modificación, si la opción Es automatico esta marcado el sistema dara un secuencial automatico de Factura al momento de realizar la Factura.
2. Opciones de Guardar y Regresar al formulario anterior.



Sucursales Complemento SRI

Sucursal: MATRIZ

Codigo Establecimiento: 001

Codigo Punto Emision: 001

Autorizacion: 1108267976

Secuencial Inicio: 1

Secuencial Fin: 1000

Es Automatico:

Secuencial Automatico: 38

**Figura A7.39** Formulario Información Adicional de las Sucursales

### Vista Tipos de Bancos

Muestra los diferentes tipos de bancos

1. Listado de los Tipos de Bancos existentes.
2. Opciones para crear un Nuevo Tipo de Banco o editar un existente.



Vista Tipo Bancos

	Codigo	Nombre	EstaActivo
Seleccionar	E	Extranjero	<input checked="" type="checkbox"/>
Seleccionar	N	Nacional	<input checked="" type="checkbox"/>
Seleccionar	0	Prueba	<input checked="" type="checkbox"/>
Seleccionar	P	Prueba	<input checked="" type="checkbox"/>

**Figura A7.40** Formulario Vista Tipos de Bancos

## Administración Tipo de Banco

Aquí se podrá ingresar un nuevo Banco o modificar uno existente.

1. Campos para el ingreso o modificación de un Tipo de Banco
2. Opciones para guardar un Tipo de Banco.

Administración Tipo Banco

Codigo: E

Nombre: Extranjero

Esta Activo:

**Figura A7.41** Formulario Administración Tipo de Banco

## Vista Bancos

Muestra los bancos ingresados al sistema.

1. Listado de los Bancos existentes.
2. Opciones para crear un nuevo Banco y Editar uno existentes.

Vista Bancos

	Secuencial	Codigo	Nombre	EstaActivo
Seleccionar	1	01	Central	<input checked="" type="checkbox"/>
Seleccionar	2	02	Pichincha	<input checked="" type="checkbox"/>
Seleccionar	3	03	Guayaquil	<input checked="" type="checkbox"/>
Seleccionar	4	04	Pacifico	<input checked="" type="checkbox"/>
Seleccionar	5	05	Austro	<input checked="" type="checkbox"/>

**Figura A7.42** Formulario Vista Bancos

## Administración de Bancos

Muestra los detalles del Banco.

1. Espacio para ingresar un nuevo Banco o editar uno existente.
2. Botones Guardar Banco y regresar al formulario anterior.

Administración Banco

Secuencial: 1

Codigo: 01

Nombre: Central

Tipo Banco: Nacional

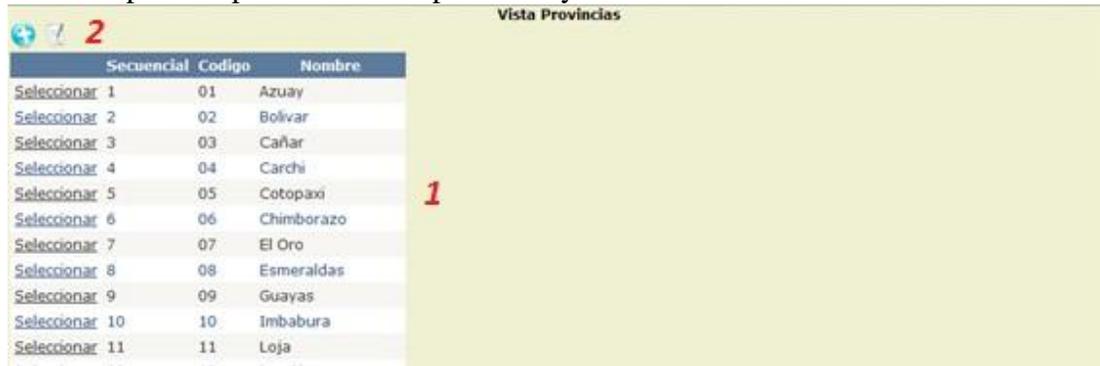
Esta Activo:

**Figura A7.43** Formulario Administración de Bancos

## Vista Provincias

Muestra las provincias ingresadas.

1. Listado de las provincias.
2. Opciones para una nueva provincia y editar uno existente.



	Secuencial	Codigo	Nombre
Seleccionar	1	01	Azuay
Seleccionar	2	02	Bolivar
Seleccionar	3	03	Cañar
Seleccionar	4	04	Carchi
Seleccionar	5	05	Cotopaxi
Seleccionar	6	06	Chimborazo
Seleccionar	7	07	El Oro
Seleccionar	8	08	Esmeraldas
Seleccionar	9	09	Guayas
Seleccionar	10	10	Imbabura
Seleccionar	11	11	Loja

**Figura A7.44** Formulario Vista Provincias

## Administración Provincia

Muestra los detalles de la provincia.

1. Espacio para ingresar una nueva Provincia o editar una existente.
2. Botones para Guardar la Provincia y regresar al formulario anterior.



Administración Provincia

Secuencial:

Nivel Division:

Codigo:

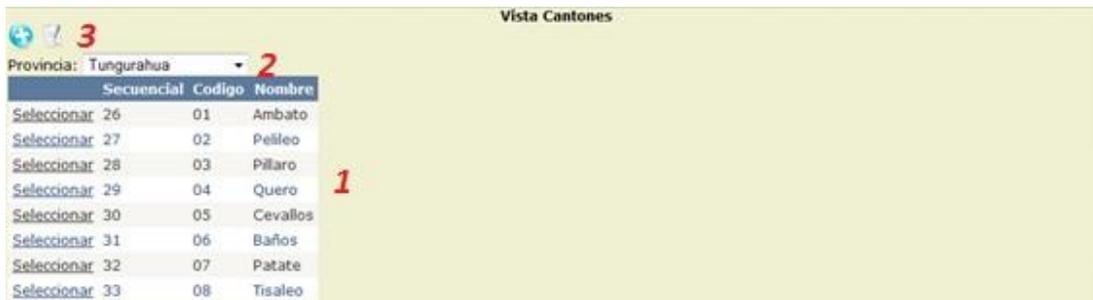
Nombre:

**Figura A7.45** Formulario Administración Provincia

## Vista Cantones

Muestra los cantones que han sido ingresados al sistema, la cual se podrá filtra por Provincia.

1. Listado de los Cantones existentes.
2. Combo para seleccionar la Provincia.
3. Opciones para crear un nuevo Canton y Editar uno existente.



**Figura A7.46** Formulario Vista Cantones

### Administración de Cantones

Muestras los detalles del Canton.

1. Espacio para ingresar un nuevo Canton o editar uno existente.
2. Botones Guardar el Canton y regresar al formulario anterior.



**Figura A7.47** Formulario Administración de Cantones

### Vista Vendedores

Muestra los vendedores ingresados.

1. Listado de los Vendedores existentes.
2. Opciones para crear un nuevo Vendedor y Editar uno existente.



**Figura A7.48** Formulario Vista Vendedores

### Administración Vendedor

Muestra los detalles del vendedor.

1. Espacio para ingresar un nuevo Vendedor o editar uno existente.
2. Botones Guardar un Vendedor y regresar al formulario anterior.

Administración Vendedor

Secuencial: 1

Nombre: Vendedor Pruebas1

Esta Activo:

**Figura A7.49** Formulario Administración Vendedor

### Estados de Crédito

Muestra los estados en los que puede estar un crédito.

Codigo	Nombre	EstaActivo
A	Al Dia	<input checked="" type="checkbox"/>
M	Moroso	<input checked="" type="checkbox"/>
Z	Cancelado	<input checked="" type="checkbox"/>
V	Vencido	<input type="checkbox"/>

**Figura A7.50** Formulario Vista de los estados de Crédito

### Vista de Estados de los Componentes de Crédito

Muestra los estados en los que pueden estar los Componentes de Crédito.

Codigo	Nombre	EstaActivo
A	Activo	<input checked="" type="checkbox"/>
C	Cancelado	<input checked="" type="checkbox"/>
E	Exigible	<input checked="" type="checkbox"/>
P	Pendiente	<input checked="" type="checkbox"/>
V	Vendido	<input checked="" type="checkbox"/>

**Figura A7.51** Formulario Vista de Estados de los Componentes de Crédito

### Tipos de Identificación de los Clientes

Muestra los tipos de identificación que puede tener un cliente.

Codigo	Nombre
CED	Cedula
RUC	Ruc
SEC	Secuencial

**Figura A7.52** Formulario Vista de los Tipos de Identificación de los Clientes

### Vista de los Proveedores

Muestra los proveedores ingresados al sistema.

1. Listado de los proveedores existentes
2. Botones para un Nuevo Proveedor y Editar uno ya existente.

Vista Proveedores

	Secuencial	Nombre	Codigo	Identificacion	FechaSistemaIngreso	EstaActivo
Seleccionar	2	VER PAINT CORP	VPC	1803505963001	22-10-2011	<input checked="" type="checkbox"/>
Seleccionar	3	PINTULAG CNIA	PINTU	1803505948001	23-10-2011	<input checked="" type="checkbox"/>

**Figura A7.53** Formulario Vista de los Proveedores

## Administración de Proveedores

Permitira la administración de un nuevo proveedor o de uno ya existente.

1. Espacio para ingresar datos de un nuevo proveedor o editar uno existente.
2. Botones para Guardar Proveedor y regresar a al formulario anterior

Administracion Proveedor

Secuencial:   
 Nombre:   
 Codigo:   
 Tipo Identificacion:   
 Identificacion:   
 Ubicacion:    
 Direccion:   
 Contacto:   
 Telefono:   
 Fax:   
 Email:   
 Sucursal:   
 Es Contrib. Especial:   
 Esta Calificado:   
 Usuario Ingreso:   
 Fecha Ingreso:   
 Esta Activo:   
 Retiene IR Bienes:    
 Retiene IR Servicios:   
 Retiene IVA Bienes:    
 Retiene IVA Servicios:

**Figura A7.54** Formulario Administración de Proveedores

## Resetear la Clave

Podremos resetear las claves ya sea de uno o de todos los usuarios, la clave reseteada será 12345.

Resetear Clave

Usuario:

**Figura A7.55** Formulario para Resetear la Clave

## Crear un Nuevo Usuario

Este formulario nos permite la creación de un nuevo Usuario para el sistema, por Sucursal.



Regístrese para obtener una nueva cuenta

Nombre de usuario:

Apellido:

Nombre:

Contraseña:

Confirmar contraseña:

Correo electrónico:

Pregunta de seguridad:

Respuesta de seguridad:

Sucursal: MATRIZ

**Figura A7.56** Formulario para Crear un Nuevo Usuario

## Administrar Usuarios

Muestra los usuarios ingresados al sistema, de las cuales se podrá editar los datos del usuario, eliminar un Usuario y editar las funciones (permisos de acceso) del usuario.

1. Listado de los Usuarios existentes en el sistema.
2. Editar Usuario seleccionado.
3. Eliminar Usuario seleccionado.
4. Editar las funciones (permisos), de los Usuarios.



Administración de Usuarios

	Username	Apellido	Nombre
<a href="#">Seleccionar</a>	usuario1	Caguana	Nahomy
<a href="#">Seleccionar</a>	usuario3	Barrionuevo	Erick
<a href="#">Seleccionar</a>	usuario2	Caguana	Anthony
<a href="#">Seleccionar</a>	admin	Barrionuevo	Frank
<a href="#">Seleccionar</a>	majo	Saritana	Maria Jose

Usuario Seleccionado: **admin**

**Figura A7.57** Formulario Administrar Usuarios

## Vista Empresa

Muestra la empresa, ingresadas al sistema, en nuestro caso solo estará la corporación VPC.

1. Listado de las empresas existentes.
2. Boton para editar la empresa.



**Figura A7.58** Formulario Vista Empresa

## Administración de Empresa

Se podrá editar la información de la empresa.

1. Espacio para la edición de los datos de la Empresa.
2. Boton Guardar y regresar al formulario Anterior

Administracion Empresa

Secuencial: 1

Siglas: PPG

Nombre: Ver Paint Corp PPG

Ruc: 1891732593001

Codigo Pais: EC

Esta Activo:

**Figura A7.59** Formulario Administración de Empresa