

PROYECTO SIMPLE DATA

FABIAN STEVEN NOVOA PALOMARES

**CORPORACIÓN UNIVERSITARIA MINUTO DE DIOS
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INFORMÁTICA, Y ELECTRÓNICA
PROGRAMA TECNOLOGÍA EN INFORMÁTICA
BOGOTÁ 2011**

PROYECTO SIMPLE DATA

FABIAN STEVEN NOVOA PALOMARES

**Proyecto de Grado Para optar Al título
de Tecnólogo en informática.**

**CORPORACIÓN UNIVERSITARIA MINUTO DE DIOS
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INFORMÁTICA, Y ELECTRÓNICA
PROGRAMA TECNOLOGÍA EN INFORMÁTICA
BOGOTÁ 2011**

Nota De Aceptación

Firma Presidente del Jurado

Firma de Jurado

Firma de Jurado

Bogotá D.C. ____ de Enero de 2012

AGRADECIMIENTOS

En primera instancia el agradecimiento es para mi familia, ya que ellos nunca siempre han sido de ayuda en momentos difíciles.

A mis compañeros de clase, que me han brindado su amistad, y su apoyo durante estos tres años.

A la Corporación Universitaria Minuto de Dios, y a la mayoría de los docentes, por su ayuda.

A todos ellos muchas gracias.

DEDICATORIA

A las cuatro personas más importantes en mi vida:

A mi madre María Etelvina Palomares, que día a día a luchado por la familia, y es una motivación constante en todos los ámbitos de mi vida.

A mi hermano Eduin Yesid Montaña Palomares, ya que durante toda mi vida, ha sido un ejemplo a seguir y modelo de persona.

A mi hermana Francy Julieth Novoa Palomares, que es una gran motivación para mi vida, y ha sido una gran compañía siempre que lo he necesitado.

A mi novia Ingrid Álvarez Gómez, que en los últimos años siempre ha estado a mi lado, sin importar las circunstancias, y ha sido una motivación permanente.

Fabián Steven Novoa Palomares

TABLA DE CONTENIDO

| | Pág. |
|--|------|
| INTRODUCCIÓN..... | 10 |
| 1.1. TÍTULO DEL PROYECTO | 11 |
| 1.2. PLANTEAMIENTO DEL PROBLEMA | 11 |
| 1.3. ALCANCES Y JUSTIFICACIÓN | 12 |
| 1.4. OBJETIVOS DEL PROYECTO..... | 13 |
| 1.4.1. General. | 13 |
| 1.4.2. Específicos. | 13 |
| 2. INGENIERÍA DEL PROYECTO..... | 14 |
| 2.1. MODELO DE DESARROLLO..... | 14 |
| 2.2. PATRÓN DE ARQUITECTURA..... | 16 |
| 3. ANÁLISIS Y DISEÑO | 18 |
| 3.1. DEFINICIÓN DE REQUERIMIENTOS..... | 18 |
| 3.2. DESCRIPCIÓN DEL SISTEMA PROPUESTO | 20 |
| 3.2.1. Proceso Común (CRUD). | 20 |
| 3.2.2. Acceso al sistema..... | 23 |
| 3.2.3. Prototipo Gráfico General. | 23 |
| 3.2.4. Clientes. | 24 |
| 3.2.5. Productos | 25 |
| 3.2.6. Transacciones | 26 |
| 3.2.7. Consultas. | 29 |
| 3.2.8. Administración | 30 |
| 3.3. Diseño del Sistema propuesto | 36 |
| 3.3.1. Diagramas Dinámicos..... | 36 |
| 3.3.2. Diagramas Estáticos..... | 45 |
| 4. DESARROLLO | 54 |
| 4.1. Especificaciones Técnicas. | 55 |
| 4.1.1. Software | 55 |
| 5. GLOSARIO | 57 |
| 6. CONCLUSIONES | 58 |
| 7. BIBLIOGRAFÍA | 59 |
| 8. ANEXOS..... | 60 |

LISTA DE TABLAS

| | Pág. |
|---|------|
| Tabla 1: Casos de Uso Administrador..... | 37 |
| Tabla 2: Casos de Uso Administrador..... | 38 |
| Tabla 3: Casos de Uso Administrador..... | 39 |
| Tabla 4: Casos de Uso Administrador..... | 40 |
| Tabla 5: Casos de Uso Administrador..... | 41 |
| Tabla 6: Casos de Uso Administrador..... | 42 |
| Tabla 7: Diccionario de Datos Tabla Abono: | 48 |
| Tabla 8: Diccionario de Datos Tabla Categoría_Producto..... | 48 |
| Tabla 9: Diccionario de Datos Tabla Cliente | 49 |
| Tabla 10: Diccionario de Datos Tabla Lista_Negra | 49 |
| Tabla 11: Diccionario de Datos Tabla Perfil | 49 |
| Tabla 12: Diccionario de Datos Tabla Permiso. | 50 |
| Tabla 13. Diccionario de Datos Tabla Politica_Reabastecimiento..... | 50 |
| Tabla 14: Diccionario de Datos Tabla Producto. | 51 |
| Tabla 15: Diccionario de Datos Tabla Registro | 51 |
| Tabla 16: Diccionario de Datos Tabla Registro_Perfil | 52 |
| Tabla 17: Diccionario de Datos Tabla Tipo Cliente | 52 |
| Tabla 18: Diccionario de Datos Tabla Tipo_Credito | 52 |
| Tabla 19: Diccionario de Datos Tabla Tipo_Venta | 52 |
| Tabla 20: Diccionario de Datos Tabla Usuario | 53 |
| Tabla 21: Diccionario de Datos Tabla Venta: | 53 |

LISTA DE ILUSTRACIONES

| | Pág. |
|--|------|
| Ilustración 1: Modelo XP | 15 |
| Ilustración 2: Modelo MVC | 16 |
| Ilustración 3: Prototipo Gráfico General / Proceso Común | 21 |
| Ilustración 4: Login..... | 23 |
| Ilustración 5: Administración de Clientes | 24 |
| Ilustración 6: Administración de Productos | 25 |
| Ilustración 7: Venta | 26 |
| Ilustración 8: Recibo de Venta | 27 |
| Ilustración 9: Abono | 28 |
| Ilustración 10: Recibo de Abono | 29 |
| Ilustración 11: Consultas Generales | 29 |
| Ilustración 12: Administrtación de Perfiles | 30 |
| Ilustración 13: Administración de Usuarios | 32 |
| Ilustración 14: Lista Negra | 33 |
| Ilustración 15: Gestión de Créditos | 34 |
| Ilustración 16: Políticas de reabastecimiento. | 35 |
| Ilustración 17: Diagrama de Casos de Uso Administrador | 43 |
| Ilustración 18: Diagrama de Casos de Uso del Sistema..... | 44 |
| Ilustración 19: Diagrama de Actividades del Sistema..... | 45 |
| Ilustración 20: Diagrama de Actividades del Sistema..... | 46 |
| Ilustración 21: Diagrama de Despliegue | 46 |
| Ilustración 22: Modelo Relacional | 47 |

LISTA DE ANEXOS

| | Pág. |
|--|------|
| Anexo 1: Manual técnico de simple data..... | 60 |
| Anexo 2: Manual de usuario simple data | 68 |
| Anexo 3: modelo entidad relación | 82 |

INTRODUCCIÓN

El presente proyecto está enfocado al diseño y el desarrollo de un software básico, el cual se centra en el manejo de ventas e inventarios para una microempresa llamada WinCore¹ que se dedica a la venta de equipos de computación. El proyecto recrea todos los procesos necesarios en un contexto real, es decir, se identifica el problema que tiene una microempresa en específico, se realizan una serie de entrevistas y reuniones con el cliente donde se definen los requerimientos y el alcance del proyecto.

El proyecto se realiza utilizando en un 100%, herramientas de software libre, esto en un contexto real ayuda a reducir costos, lo cual a cualquier cliente le puede resultar interesante, en un artículo publicado por el ingeniero Manuel Dávila, afirma “El software libre está introduciendo mejoras que con seguridad influirán en los precios del software propietario” (Dávila, 2005), algo que es notorio actualmente, el software libre adquiere buen terreno en el campo empresarial, por esta razón se ha decidido desarrollar el proyecto utilizando el lenguaje de Programación PHP² y el gestor de bases de datos mySQL³, y para la programación en el lado del cliente se utiliza JavaScript⁴.

El presente trabajo tiene como objetivo al desarrollo de un software con la capacidad de administrar de manera fácil los inventarios, clientes, ventas y deudas por cobrar de una microempresa.

El modelo de desarrollo utilizado es XP (eXtreme Programming), esto debido a que es una metodología de desarrollo ágil, y es conveniente ya se busca poder entregarle al cliente avances en el desarrollo en periodos de tiempo cortos y XP ofrece las herramientas ideales para lograrlo.

¹ La microempresa se prestó para la implementación del software durante un ciclo de pruebas.

² Lenguaje de Programación interpretado basado en Scripts muy utilizado en el desarrollo de aplicativos web

³ Gestor de Bases de datos relacionales.

⁴ Lenguaje de programación interpretado, que se ejecuta directamente en un navegador web

A continuación se muestra de manera más específica todo lo relacionado con el proyecto Simple Data.

1.1. TÍTULO DEL PROYECTO

Simple Data: El nombre del proyecto es debido a la capacidad que tiene el aplicativo de manejar la información de una forma simple y muy fácil de comprender para el usuario, además de ser intuitivo.

1.2. PLANTEAMIENTO DEL PROBLEMA

La problemática está basada en la situación actual de una microempresa dedicada a la venta de equipos de cómputo y suministros para los mismos, ya que no cuenta con un sistema de información eficiente que tenga la capacidad de llevar adecuadamente los registros básicos que necesita cualquier microempresa, dentro de estos registros se pueden citar los siguientes: Clientes, Productos, Ventas diarias, semanales o mensuales, inventarios, abonos, recibos, deudas por cobrar, donde se presentan problemas tales como tiempos elevados para la generación de reportes de ventas o de deudores, las copias de las facturas o recibos ocasionalmente se pierden o se dañan ya que están en medio físico y dificultan la obtención de datos reales, la falta de un control de existencias dificulta la adquisición de las mismas, los datos no se encuentran seguros, los valores obtenidos manualmente son datos estimados y no exactos, la información no está bajo una buena seguridad y en cualquier momento puede perderse o dañarse, no hay ningún control eficiente para dar manejo a las deudas de los clientes a quienes se les ofrece crédito.

Por lo general se mantiene mucha información registrada en hojas de cálculo y en ocasiones la información no es registrada o está en medio físicos, donde la empresa se exponen a tener los datos desorganizados, perder tiempo o incluso hasta a llegar a perder la información ya que no se cuenta con respaldo de la misma.

1.3. ALCANCES Y JUSTIFICACIÓN

Para las microempresas es importante, migrar su información a un servicio tecnológico que ofrezca seguridad y confiabilidad, una microempresa a pesar de ser pequeña puede llegar a manejar un gran volumen de datos y para cualquier organización sin importar el tamaño es de vital importancia el manejo seguro de la información, por lo cual es conveniente generar este proyecto, teniendo en cuenta que el aplicativo puede ser funcional para otras microempresas.

El aplicativo objeto de este proyecto no es un software contable, se podría definir mejor con un sistema de información básico que da al usuario la facilidad de registrar y consultar lo necesario para su empresa.

Este software permitirá, llevar registros de manera organizada, otorgando a la información un factor de alta disponibilidad y fiabilidad, mejorando los tiempos de consulta de la información y la forma de trabajar de todos los integrantes de la organización, que utilicen el sistema, además evitar duplicidad en datos y pérdida de los mismos por degradación de papel.

El software ofrece la posibilidad de ser utilizado por varios usuarios de manera simultánea, lo cual permite que una microempresa que tenga varias sucursales, pueda tener toda su información integrada, generando así un canal de información constante.

En general el aplicativo hará posible que la empresa pueda tener realmente un control, y un respaldo, además el software ofrece la posibilidad de integrarse con otros sistemas, por medio de conexiones a bases de datos, y es posible crear nuevas funcionalidades, adaptando y reutilizando el código ya existente, por lo cual es un aplicativo que puede estar en constante progreso, y evolucionar de manera muy rápida, dependiendo de las necesidades que experimente la empresa con el paso del tiempo, es decir, el software está preparado para el crecimiento de la microempresa, y puede llegar a convertirse en una herramienta CRM que permita controlar diferentes áreas de la empresa.

1.4. OBJETIVOS DEL PROYECTO

Para el presente proyecto se definen los siguientes objetivos:

1.4.1. General.

- Desarrollar un sistema de información básico, teniendo en cuenta las fases de análisis, diseño e implementación, orientado a una empresa dedicada a la venta de equipos de cómputo y suministros.

1.4.2. Específicos.

- Ofrecer a la microempresa la posibilidad de manejar su información de una manera rápida y eficiente, de tal forma que sea fácil de registrar y consultar la información.
- Generar un software que permita mejor manejo de la información, evitar pérdida de la misma, ofrezca acceso de manera oportuna y evite la redundancia de los datos.
- Darle al usuario las herramientas de administración dentro del software para que tenga control total sobre todos sus datos.

2. INGENIERÍA DEL PROYECTO

Para cualquier proyecto de software es importante seleccionar algún modelo de desarrollo que permita a los integrantes del proyecto tener una idea clara acerca de las etapas puntuales que el proyecto contemplara de inicio a fin.

Debido a que el presente proyecto está enfocado a desarrollar una herramienta que sea útil dentro de un contexto real, en este capítulo se detalla de manera general el modelo y las etapas que se llevarán a cabo durante todas las fases del proyecto, para iniciar teniendo unas buenas bases de tal manera que se pueda llevar el proyecto a un fin exitoso de manera ágil.

2.1. MODELO DE DESARROLLO.

El modelo de desarrollo que se utilizará en este proyecto es XP (Programación Extrema), se decide esto teniendo en cuenta que es un proceso ágil de desarrollo, tiene una fuerte orientación hacia la interacción con el cliente, y es muy aplicable a grupos pequeños de desarrollo, como en este caso donde se involucran únicamente un desarrollador, con este modelo de desarrollo se busca la calidad y la agilidad para desarrollar el proyecto, teniendo en cuenta los principios básicos del modelo como: simplicidad, comunicación, y retroalimentación, además del hecho de que todo se realiza en conjunto con el cliente quien es muy importante durante todo el ciclo de vida del proyecto. En la página extremeprogramming.org, se habla sobre el éxito de este modelo de desarrollo “Ya se ha demostrado ser un gran éxito en muchas empresas de todos los tamaños e industrias en todo el mundo. Extreme Programming es un éxito porque hace hincapié en la satisfacción del cliente” (Wells, 2009)

Es importante resaltar la posibilidad que ofrece este modelo de definir nuevos requerimientos en la marcha, lo cual se ajusta mucho a la vida real, y a los principios de la programación orientada a objetos (POO), donde el cliente es muy importante durante toda la etapa de desarrollo, se hará énfasis en la comunicación con él, ya que permitirá la realización de pruebas de implementación del proyecto en su microempresa.

En el presente proyecto se aplicarán las etapas de XP de la siguiente forma, inicialmente una reunión con el cliente para plantear de manera global el esquema que manejará el proyecto, además de esto tomar los requerimientos del sistema que propone el cliente, XP define que el cliente debe generar las denominadas “Historias de Usuario”, pero para efectos prácticos de este proyecto se definirán los requerimientos junto con el usuario en la primera reunión.

Una segunda reunión donde se muestra al cliente la lista de requerimientos, definidos, según la primera reunión y allí se decidirá cuantas iteraciones va a tener el proyecto, los tiempos de entrega de cada versión, y el orden de desarrollo de los requerimientos. Luego de esto se empieza con el desarrollo como tal del proyecto, siguiendo el esquema planteado en las reuniones y haciendo entregas periódicas de versiones al cliente, hasta culminar con las iteraciones y hacer la entrega final en los tiempos estipulados.

Debido a que el modelo XP genera poca documentación, se trabajarán los diagramas con UML (Lenguaje de modelamiento Unificado), ya que este lenguaje agrupa todos los modelos que se necesitan para la documentación del proyecto en todas las fases del proyecto, esto teniendo en cuenta que UML sirve para especificar, diseñar y documentar, el uso de UML dentro del proyecto es una de las razones por cual no se utilizan las “Historias de Usuario” de XP.

Teniendo en cuenta el modelo XP, se ha realizado un gráfico donde se contemplan las etapas del proyecto.

Étapas Del Desarrollo Del Proyecto

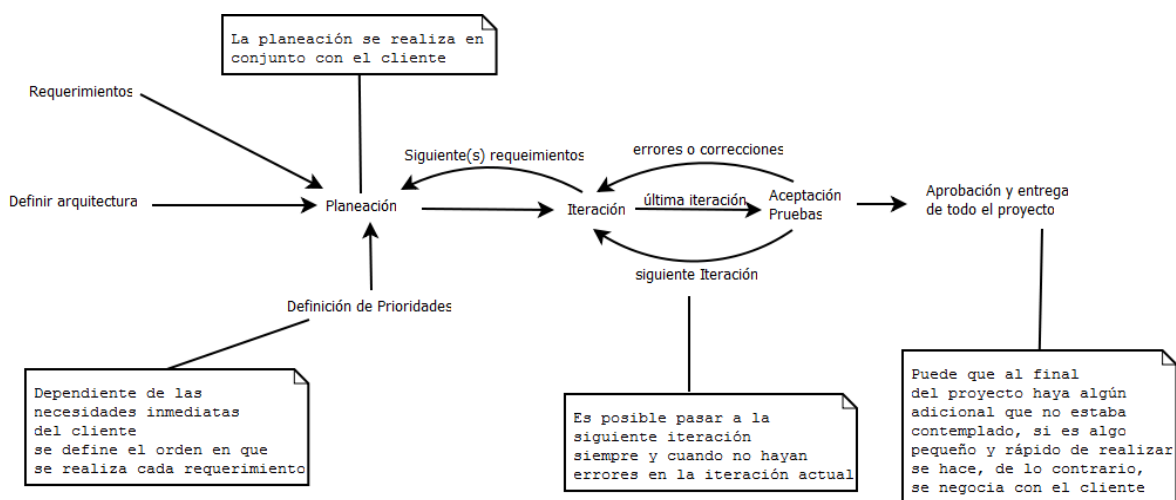


Ilustración 1: Modelo XP

Fuente: El autor

2.2. PATRÓN DE ARQUITECTURA

Para el diseño del aplicativo como tal se utilizará el patrón MVC⁵, que separa el la Vista de la Lógica, esto teniendo en cuenta que se habla de una aplicación totalmente orientada Web, es una buena práctica separar el código HTML, del código de lógica de negocio. En el siguiente grafico se muestra la forma en que se implementa el patrón MVC en el presente proyecto.

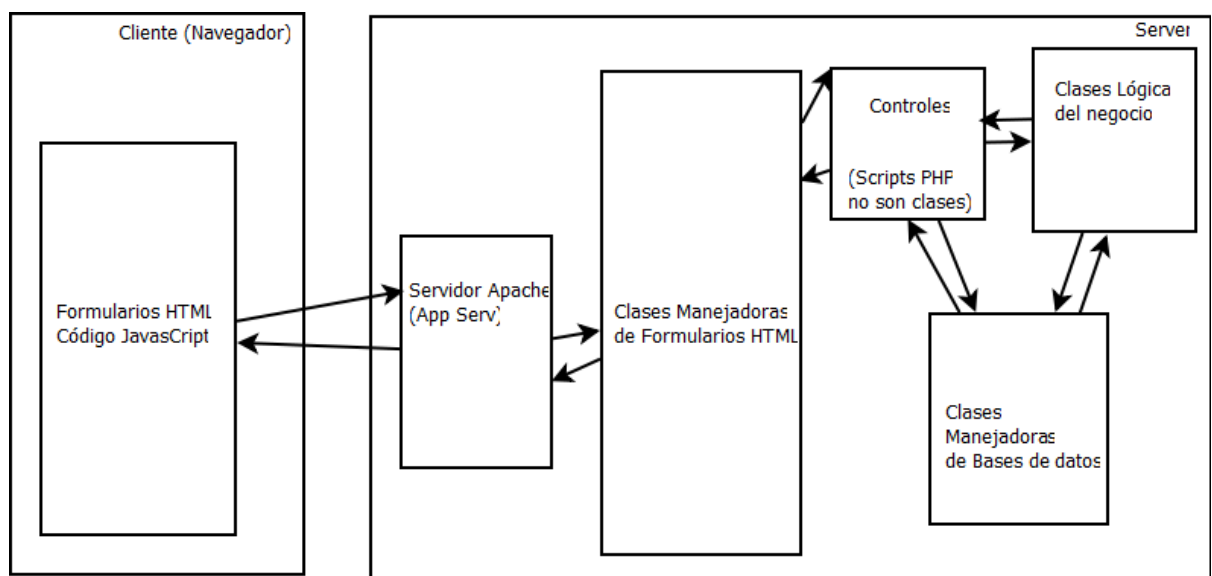


Ilustración 2: Modelo MVC

Fuente: El autor

En el gráfico se muestra la comunicación que realiza el Modelo, la Vista y los controles, inicialmente el cliente puede enviar una petición, la cual recibe el servidor web en este caso el Servidor Apache, la primera comunicación que hay se da con las clases manejadores de formularios, es ella la que decide qué control debe recibir la petición, y enviar los datos de la petición, allí el control designado recibe los parámetros GET o POST, y decide qué hacer con ellos, es decir decide si debe comunicarse con una de las clases de la lógica del negocio, o si puede obtener los datos directamente desde las clases manejadoras de base de datos, en caso de que la petición sea dirigida a una clase de negocio, ésta clase también puede consultar las clases manejadoras de bases de datos, luego de esto se da una respuesta al control, el control se comunica con la clase manejadora de

⁵ Patrón de diseño Modelo Vista Controlador

formulario a la que va dirigida la respuesta, y finalmente el servidor arroja un resultado al cliente.

3. ANÁLISIS Y DISEÑO

Antes de iniciar la codificación del proyecto, se hace necesario realizar una etapa de análisis y diseño, ya que esto garantiza en gran parte que el proyecto concluya de manera exitosa.

Esta etapa asegura que han sido entendidos todos los requerimientos del software, todos los actores del software y de igual manera sus características gráficas; esto teniendo en cuenta que aunque es importante tener los conocimientos técnicos, también es de suma importancia entender la lógica del negocio que plantea el cliente, asimismo es importante que el cliente conozca sus posibles interacciones con el aplicativo.

3.1. DEFINICIÓN DE REQUERIMIENTOS

Luego de las reuniones sostenidas con el cliente, se definen los requerimientos del software en base las especificaciones del cliente:

3.1.1. Requerimientos Funcionales del Software.

Se han clasificado los requerimientos funcionales en varios grupos de acuerdo al uso propuesto dentro de la aplicación.

Globales y de seguridad

- Permitir la creación de usuarios (Nombre de Usuario, Contraseña).
- Administrar permisos sobre cada funcionalidad del sistema.
- Permitir el cambio de contraseña de usuarios.
- Permitir la creación de perfiles de usuario
- Permitir la modificación de los perfiles de usuario.
- Permitir ingreso al sistema

Clientes

- Permitir el ingreso de clientes con datos básicos como nombre, cédula, teléfono y dirección.
- Permitir la modificación de los datos de los clientes.
- Manejar tipos de crédito asignados a los clientes, donde se debe registrar un tope.
- Tener un sistema de abonos.

Productos

- Permitir el registro de productos (Nombre de producto, precio de compra, precio de venta especial, precio de venta general, cantidad en inventario)
- Permitir la modificación de los datos de un producto.
- La actualización de productos en inventario se puede realizar por interfaz o por medio de una hoja de calculo que se importa al software
- El software debe manejar una política de reabastecimiento, que le informe al usuario la falta de determinado producto de acuerdo a la demanda del mismo.

Mezclados

- Permitir dos tipos de venta a crédito y contado.
- Permitir consultas acerca de las deudas de los clientes, las ventas diarias, semanales, mensuales, debe permitir visualizar el estado de los clientes.
- Implementar una lista negra, donde se registren los clientes a quienes no se les permite realizar compras a crédito.
- La lista negra se debe manejar manualmente, si un cliente queda a paz y salvo, esto no significa que se elimine automáticamente de la lista negra.

Reportes

- Generar recibos de venta.
- Permitir la consulta de los recibos de venta luego de generados.
- Proporcionar la forma de consultar el inventario de los productos existentes.

3.1.2. Requerimientos No Funcionales del Software.

- El aplicativo debe brindar una disponibilidad mayor al 90%.
- Los tiempos de generación de reportes deben estar entre 3 y 5 segundos por máximo.

- La aplicación debe ser totalmente orientada a la Web.
- Los datos proporcionados por la aplicación deben ser confiables y precisos.

3.2. DESCRIPCIÓN DEL SISTEMA PROPUESTO

A continuación se muestra una aproximación gráfica a lo que sería el sistema, los siguientes prototipos gráficos, están sujetos a cambios, a medida que se desarrollan las iteraciones.

El sistema puede asumir tantos roles de usuario como sean necesarios, por tanto es el mismo usuario quien define los perfiles o roles, y es el mismo usuario quien otorga permisos sobre las funciones específicas del sistema a cada perfil, el sistema se entrega al usuario con un único usuario de nombre adminest y Perfil Administrador, el cual tiene acceso a todas las funciones del sistema.

3.2.1. Proceso Común (CRUD)⁶.

Antes de iniciar con la descripción del sistema es necesario definir un procedimiento común que es posible encontrar en varias funciones del sistema, es el procedimiento para la creación, consulta, modificación y eliminación de registros, lo cual se ve reflejado directamente en la base de datos; éste proceso es utilizado para Clientes, Productos, Usuarios y Créditos, éste proceso puede ser comprendido fácilmente, mediante la ilustración 3, que contiene la estructura general de las pantallas del software, ilustrando cada una de sus partes.

⁶ En computación CRUD es el acrónimo de Crear, Obtener, Actualizar y Borrar (del original en inglés: Create, Read, Update and Delete). Es usado para referirse a las funciones básicas en bases de datos o la capa de persistencia en un sistema de software. (Wikipedia, 2011)

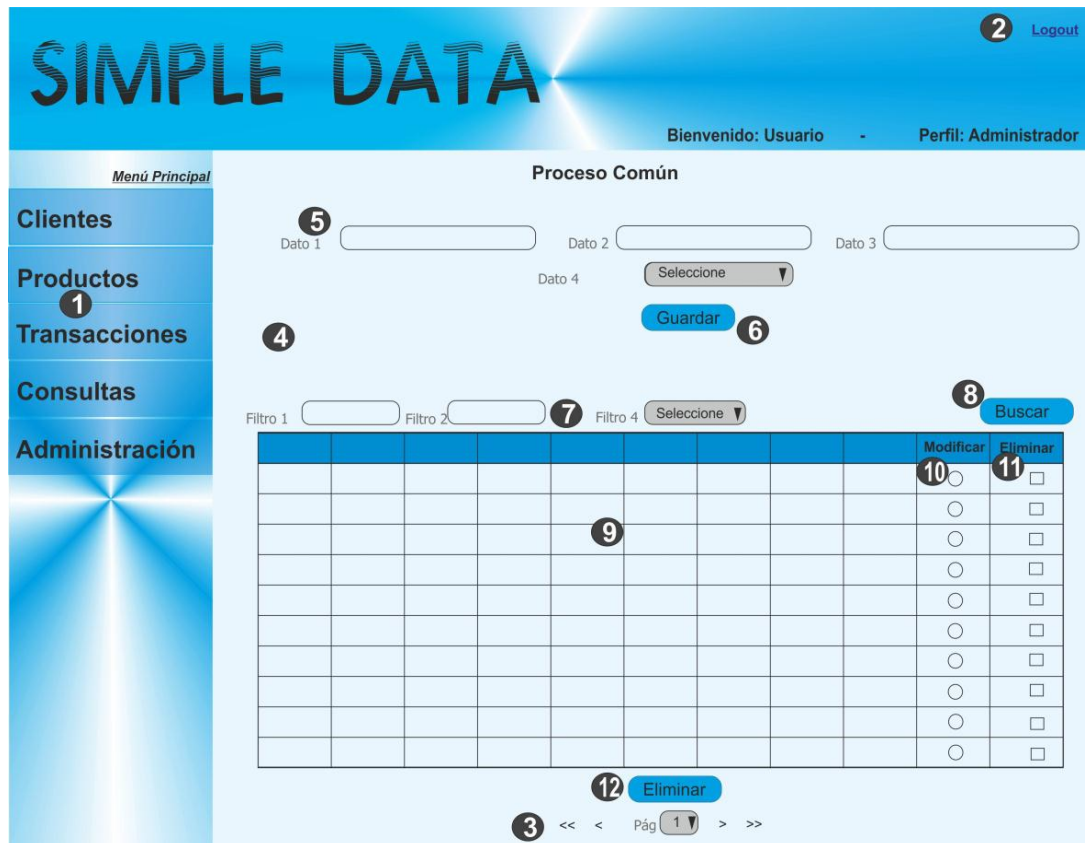


Ilustración 3: Prototipo Gráfico General / Proceso Común
Fuente. El autor

- Para crear un registro únicamente debe diligenciar los datos solicitados en el formulario de la parte superior (Ilustración 3 – Item 5) y luego hacer clic en el botón *guardar* (Ilustración 3 – Item 6).
- Para Modificar un registro ya existente se debe realizar una búsqueda del registro por medio de los filtros (Ilustración 3 – Item 7), y se da clic en el botón buscar (Ilustración 3 – Item 8), se mostrarán los datos obtenidos en la búsqueda en la tabla (Ilustración 1 – Item 9), luego de haber ubicado el registro que se desea modificar se debe hacer clic en la penúltima casilla de la tabla (Ilustración 3 – Item 10), y aparecerán los datos en el formulario ubicado en la parte superior (Ilustración 3 – Item 5), allí se podrán hacer las modificaciones, y luego hacer clic en el botón guardar (Ilustración 3 Item 6).
- Para eliminar uno o más registros, se realiza la búsqueda por medio de los filtros (Ilustración 3 – Item 7), y se hace clic en el botón buscar (Ilustración 3 – Item 8), para encontrar los registros que se desean eliminar, se seleccionan los deseados en la última casilla de cada registro de la tabla

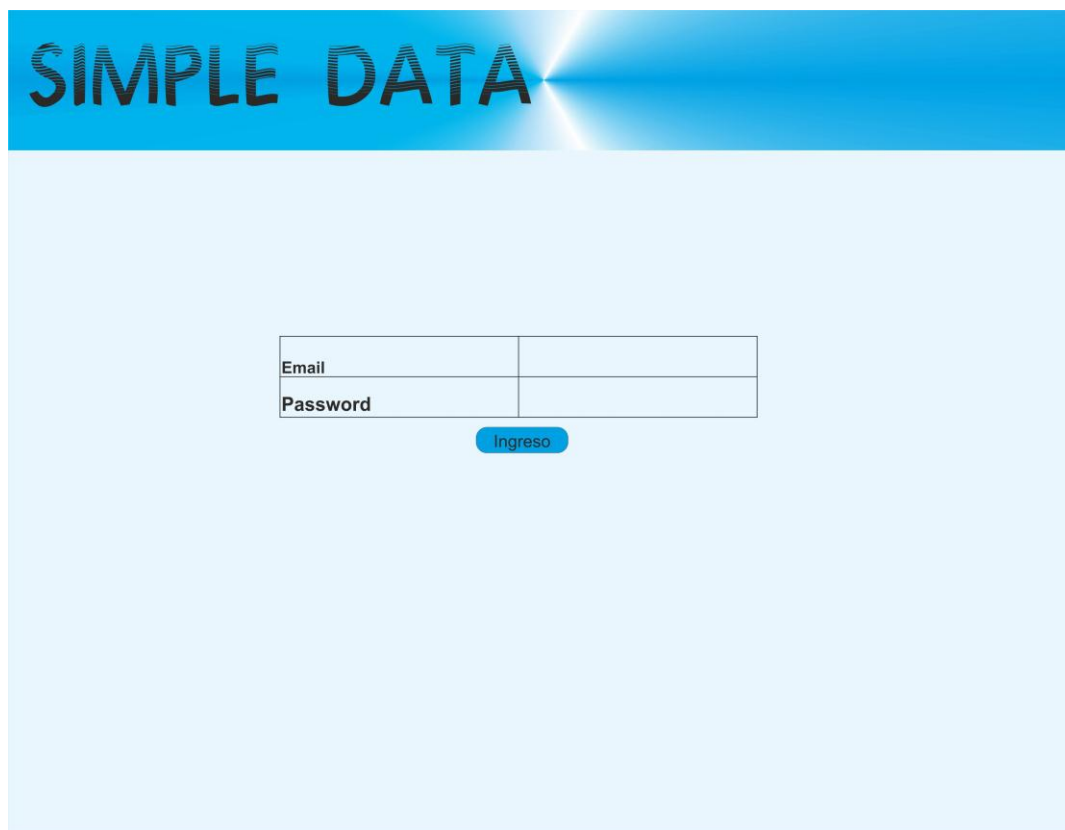
que se desee eliminar (Ilustración 3 – Ítem 11), y por último se hace clic en el botón eliminar ubicado en la parte inferior del formulario (Ilustración 3 – Ítem 12).

Notas:

Cuando hay tablas con más de 10 registros, se realiza paginación de los datos, y por tanto es necesario utilizar el control de paginación (Ilustración 3 – Ítem 3), que se encuentra en la parte inferior.

3.2.2. Acceso al sistema.

Inicialmente el acceso al sistema se realiza por medio de una pantalla donde usuario debe diligenciar los campos nombre de usuario y contraseña.



The image shows a login interface for a system named 'SIMPLE DATA'. At the top, there is a blue header with the text 'SIMPLE DATA' in white, bold, sans-serif font. Below the header, the background is a light blue gradient. In the center, there is a login form consisting of two input fields stacked vertically. The top field is labeled 'Email' and the bottom field is labeled 'Password'. Below these fields is a blue button with the text 'Ingreso' in white. The entire form is centered on the page.

Ilustración 4: Login
Fuente: El autor

3.2.3. Prototipo Gráfico General.

Luego de ingresar al sistema se muestra una interfaz con un menú vertical ubicado en la parte izquierda (Ilustración 3 – Item 1), el cual siempre será visible. En la parte superior derecha de la ventana el usuario podrá encontrar un link llamado *logout* para salir del sistema (Ilustración 3 – Item 2). La parte derecha de la pantalla luego del menú, se definirá como el espacio de trabajo del usuario donde podrá realizar las tareas asociadas a la aplicación (Ilustración 3 – Item 4).

3.2.4. Clientes.

Logout

BIENVENIDO: Usuario - Perfil: Administrador

Menú Principal

Cientes

Administrar Clientes

Productos

Transacciones

Consultas

Administración

Administración de Clientes

Nombre Apellido Telefono

Dirección E-mail Cédula

Tipo de Cliente Tipo de Crédito

Nombre Cédula Tipo de Cliente Tipo de Crédito

| Item # | Nombre | Apellido | Teléfono | Dirección | Email | Cédula | T. Cliente | T. Crédito | Modificar | Eliminar |
|--------|--------|----------|----------|-----------|-------|--------|------------|------------|-----------------------|--------------------------|
| | | | | | | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | | | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | | | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | | | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | | | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | | | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | | | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | | | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | | | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | | | | | | <input type="radio"/> | <input type="checkbox"/> |

<< < Pág 1 > >>

Ilustración 5: Administración de Clientes
Fuente: El autor

En esta pestaña el cliente puede encontrar la opción de administración de clientes:

En administración del clientes se pueden realizar las tareas mencionadas en el *Proceso Común*: Crear, Modificar y Eliminar Clientes

Notas:

Por defecto se tiene cargado un cliente con nombre *Genérico*, en caso de que hagan ventas, donde no se hace necesario hacer registro del cliente, pero sí de la transacción, se le cargan a este cliente. Al cliente genérico sólo se le pueden realizar ventas de contado.

3.2.5. Productos

SIMPLE DATA Logout

Bienvenido: Usuario - Perfil: Administrador

Menú Principal

- Cientes
- Productos**
 - Administrar Productos
- Transacciones
- Consultas
- Administración

Administración de Productos

Código (Autogenerado) Nombre Cant. Inventario

Precio Compra Precio Especial Precio General

Código Nombre

| Item # | Código | Nombre | Precio Compra | Precio Especial | Precio General | Cant. Inv. | Modificar | Eliminar |
|--------|--------|--------|---------------|-----------------|----------------|------------|-----------------------|--------------------------|
| | | | | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | | | | <input type="radio"/> | <input type="checkbox"/> |

<< < Pág 1 > >>

Exportar Inventario

Nombre de Archivo

Cargar Inventario

Ubicación

Ilustración 6: Administración de Productos
Fuente: El autor

Cuando el usuario haga clic sobre la opción productos podrá realizar las operaciones del *Proceso Común*: Crear, Modificar y Eliminar productos.

- Exportar Inventario:

Para exportar el inventario se debe diligenciar la casilla nombre de archivo, con el nombre del archivo que se desee, y luego se debe dar clic en exportar, con lo cual se descargará un archivo en Excel el cual tendrá el resultado del filtro realizado.

- Cargar Inventario:

Para cargar un inventario se debe hacer clic en el botón (...), esto cargará la ruta del archivo en la casilla *Ubicación*, luego de esto se hace clic en el botón *Cargar*. Es importante utilizar la estructura de un archivo antes exportado, de lo contrario se corre el riesgo de arruinar la información actual del inventario, si en el archivo CVS, se encuentra un código de producto ya existente, este se actualizará, si tiene

un código no existente, éste se ignorará y el sistema autogenerará uno nuevo creando un nuevo registro, si no tiene código, el sistema creará un nuevo registro y autogenerará un nuevo código. (Ilustración 6)

3.2.6. Transacciones

Las transacciones disponibles en la aplicación son Venta y Abono.

- Venta

The screenshot shows the 'SIMPLE DATA' application interface. At the top, there is a blue header with the text 'SIMPLE DATA' and a 'Logout' link. Below the header, it says 'Bienvenido: Usuario - Perfil: Administrador'. On the left, there is a 'Menú Principal' sidebar with options: Clientes, Productos, Transacciones (highlighted), Venta, Abono, Consultas, and Administración. The main content area is titled 'Realizar Venta' and contains the following elements:

- Form fields: 'Cliente' (dropdown menu), 'Tipo venta' (dropdown menu), 'Producto' (dropdown menu), and 'Cantidad' (text input).
- Buttons: 'Adicionar' (blue button).
- Table with 6 columns: 'Item #', 'Código', 'Producto', 'Cantidad', 'Total', and 'Eliminar'. The table is currently empty.
- Form field: 'Total A pagar' (text input).
- Buttons: 'Guardar', 'Exportar A word', and 'Vista previa de Impresión' (all blue buttons).

Ilustración 7: Venta

Fuente: El autor

Para realizar una venta el Usuario debe inicialmente diligenciar los campos *Cliente* y *Tipo Venta*, si estos datos no se encuentran diligenciados, no se muestran los campos *Producto*, *Cantidad* ni el botón *Adicionar*. Cuando los campos *Cliente* y

Tipo Venta, estén diligenciados se procede a ingresar uno a uno cada producto objeto de la venta con sus respectivas cantidades, haciendo clic en el botón *Adicionar*, cada vez que se diligencien los datos de un producto. Cada vez que se hace clic en el botón *Adicionar*, aparecerá un nuevo registro en la tabla, en caso de equivocaciones, se puede eliminar algún cualquier registro haciendo clic en la casilla que aparece en la celda *Eliminar* de cada registro que se desee eliminar. A medida que se ingresan o se eliminan productos el campo *Total A pagar* se actualiza de manera automática. (Ilustración 7)

Una vez se haya finalizado de adicionar los productos se procede a hacer clic en el botón *Guardar*, esto finaliza la venta, y hace inmodificable la factura. Una vez se hace clic en el botón *Guardar*, se habilitan los botones *Exportar A Word*, y *Vista Previa de Impresión*, los cuales mostrarán un imprimible del *Recibo de Venta*, ya sea en formato .DOC, o en una salida por WEB. El formato generado se muestra a continuación (Ilustración 7)

SIMPLE DATA

Recibo de Venta Nº ??????

| | |
|---------------------------|--------------------------|
| Nombre Cliente: | Cedula Cliente: |
| Dirección Cliente: | Teléfono Cliente: |
| Fecha: | Vendedor: |

| Item # | Nombre | Cantidad | V/Unitario | V/Total |
|--------|--------|----------|------------|---------|
| | | | | |
| | | | | |
| | | | | |

Total a pagar

| | |
|---|--|
| Espacio Para firma y sello Vendedor | Espacio Para firma y sello Cliente |
|---|--|

Ilustración 8: Recibo de Venta

Fuente: El autor

- Abono

The screenshot shows a web application interface for 'SIMPLE DATA'. The header is blue and contains the logo 'SIMPLE DATA' on the left, 'Logout' on the right, and user information 'Bienvenido: Usuario - Perfil: Administrador' in the center. A sidebar menu on the left is titled 'Menú Principal' and includes options: 'Clientes', 'Productos', 'Transacciones', 'Venta', 'Abono', 'Consultas', and 'Administración'. The main content area is titled 'Realizar Abono' and contains a form with a 'Cliente' dropdown menu (currently showing 'Seleccione'), a 'Valor abono' text input field, and three buttons: 'Guardar', 'Exportar A word', and 'Vista previa de Impresión'.

Ilustración 9: Abono
Fuente: El autor

Al hacer Clic en la opción abonos del menú Transacciones, se muestra un formulario como el de la Ilustración 9, allí únicamente se debe seleccionar el cliente que realiza el abono, y se debe registrar un valor real de abono, es decir, debe ser un número positivo, y no debe superar el valor total de la deuda que tiene el cliente.

Una vez se diligencien los datos, se debe hacer clic en el botón *Guardar*, luego de esto se habilitarán los botones *Exportar A Word*, y *Vista Previa de Impresión*, los cuales mostrarán un imprimible del *Recibo de Abono*, ya sea en formato .DOC, o en una salida por WEB. El formato generado se muestra a continuación (Ilustración 10).

SIMPLE DATA

Recibo de Abono N° ??????

| | |
|--------------------|-------------------|
| Nombre Cliente: | Cedula Cliente: |
| Dirección Cliente: | Teléfono Cliente: |
| Fecha: | Vendedor: |

| Valor Abono | Saldo Anterior | Nuevo Saldo |
|-------------|----------------|-------------|
| | | |

| | |
|--|---------------------------------------|
| Espacio Para firma y sello Vendedor | Espacio Para firma y sello Cliente |
|--|---------------------------------------|

Ilustración 10: Recibo de Abono

Fuente: El autor

3.2.7. Consultas.

- Consultas Generales

The screenshot displays the 'SIMPLE DATA' web application interface. At the top, the title 'SIMPLE DATA' is prominent, with a 'Logout' link on the right. Below the title, the user is logged in as 'Usuario' and has the 'Administrador' profile. A 'Menú Principal' (Main Menu) is located on the left side, listing various system functions. The main content area is titled 'Consultas Generales' (General Queries). It contains several search filters: 'Cliente' (dropdown), 'Tipo venta' (dropdown), 'Tipo Cliente' (dropdown), and 'Vendedor' (dropdown). There are also input fields for 'Fecha Ini' (Start Date) and 'Fecha Fin' (End Date), a 'Tipo Transacción' (dropdown), and a 'Buscar' (Search) button. Below the filters, there are two data tables. The first table is labeled 'Ventas' (Sales) and has columns: 'Item #', 'No Recibo Venta', 'Cliente', 'Fecha', 'Vendedor', 'Tipo Venta', and 'Valor'. The second table is labeled 'Abonos' (Payments) and has columns: 'Item #', 'No Recibo Abono', 'Cliente', 'Fecha', 'Vendedor', and 'Valor'. Both tables are currently empty. Navigation controls for the tables include 'Pág 1' and arrows for page navigation. At the bottom of the page, there is a button labeled 'Exportar A excel' (Export to Excel).

Ilustración 11: Consultas Generales

Fuente: El autor

Al hacer clic en la opción *Consultas Generales*, del menú consultas, se mostrará un menú como el de la Ilustración 11, allí se podrán consultar las transacciones de acuerdo a la combinación de filtros utilizada.

Ejemplo:

Es posible realizar una consulta de las transacciones hechas a crédito de un determinado cliente, en un determinado rango de tiempo.

Notas:

La información que se muestra en la casilla *No Recibo Venta y Número Recibo Abono*, es un link, este link a la pantalla de Transacciones -> Ventas o Transacciones -> Abonos, y muestra los datos de la recibo en cuestión, dando la posibilidad de Exportar a Word, o ver la salida Web.

En la parte inferior del formulario, se puede exportar el resultado de la consulta, en un archivo XLS. Los datos mostrados inicialmente en la tabla, son los datos encontrados sin haber aplicado ningún filtro.

3.2.8. Administración

- Perfiles:

The screenshot shows the 'SIMPLE DATA' web application interface. The top header includes the logo 'SIMPLE DATA', a 'Logout' link, and a user status bar: 'Bienvenido: Usuario - Perfil: Administrador'. A sidebar menu on the left lists navigation options: 'Menú Principal', 'Clientes', 'Productos', 'Transacciones', 'Consultas', 'Administración', 'Perfiles', 'Usuarios', 'Lista Negra', 'Créditos', and 'Políticas de reabastecimiento'. The main content area is titled 'Administración de Perfiles' and contains a 'Creación de Perfil' form with a 'Nombre Perfil' input field and a 'Guardar' button. Below the form is a 'Gestión de Permisos' section with a table listing permissions for various modules.

| Permiso | Ingreso | Crea | Modifica | Elimina |
|----------------|--------------------------|--------------------------|--------------------------|--------------------------|
| Clientes | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Productos | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Cargar Inv. | <input type="checkbox"/> | | | |
| Transacciones | <input type="checkbox"/> | <input type="checkbox"/> | | |
| Consultas | <input type="checkbox"/> | | | |
| Generales | <input type="checkbox"/> | | | |
| Ganancias | <input type="checkbox"/> | | | |
| Administración | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Perfiles | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Usuarios | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Lista Negra | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Créditos | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Políticas | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Ilustración 12: Administratción de Perfiles

Fuente: El autor

Al hacer clic en la opción *Perfiles* del menú *Administración*, el usuario podrá Crear, Modificar y Eliminar Perfiles.

Un perfil consta de un nombre y unos permisos sobre la aplicación. Para crear un perfil únicamente se debe diligenciar el campo *Nombre Perfil*, y hacer clic en el Botón *Guardar*, de esta manera se crea un perfil, que no tiene ningún permiso sobre la aplicación, inmediatamente se crea el perfil, aparecerá el nombre del Perfil abajo del título *Gestión de Permisos*, para asignar los permisos correspondientes, únicamente se debe hacer clic en el nombre del perfil, seleccionar los permisos del perfil, y luego hacer clic en Guardar Permisos.

Para eliminar un perfil se debe hacer clic en el nombre del perfil, y luego hacer clic en el botón *Eliminar* que se habilita una vez se selecciona el perfil, si no se ha dado clic en ningún perfil el botón no aparece.

Notas:

No es posible eliminar un perfil que tiene usuarios asignados, el sistema no lo permitirá.

Si se dan permisos de Crear, Modificar y Eliminar a un perfil en la opción Administración, se están dando estos permisos sobre todas las opciones de administración.

Si se selecciona Ingreso únicamente, el usuario tendrá permisos de sólo lectura sobre la función.

Si el usuario no tiene el permiso, es decir el primer check, la funcionalidad no aparecerá en el menú

- Usuarios

The screenshot shows the 'SIMPLE DATA' web application interface. The header features the application name 'SIMPLE DATA' and a 'Logout' link. Below the header, the user is logged in as 'Usuario' with the role of 'Administrador'. The main content area is titled 'Administración de Usuarios' and contains a form for adding a new user with fields for 'Nombre Usuario', 'Email', 'Password', and a 'Perfil' dropdown menu. A 'Guardar' button is located below the form. Below the form is a table with the following structure:

| # | Usuario | Email | Perfil | Modificar | Eliminar |
|---|---------|-------|--------|-----------------------|--------------------------|
| | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | <input type="radio"/> | <input type="checkbox"/> |

Below the table is an 'Eliminar' button and a pagination control showing 'Pág 1'.

Ilustración 13: Administración de Usuarios
Fuente: El autor

Si se hace clic en la opción *Usuarios* del menú *Administración*, se pueden realizar las acciones del proceso común: Crear, Modificar y Eliminar Usuarios.

Notas:

El email del usuario, representará el *username* para la aplicación.

Solamente un Administrador puede cambiar la contraseña de un usuario.

- Lista Negra

The screenshot shows the 'SIMPLE DATA' web application interface. At the top, there is a blue header with the application name 'SIMPLE DATA' and a 'Logout' link. Below the header, the user is logged in as 'Usuario' with the profile of 'Administrador'. A main navigation menu on the left lists various options: Clientes, Productos, Transacciones, Consultas, Administración, Perfiles, Usuarios, Lista Negra, Créditos, and Políticas de reabastecimiento. The 'Lista Negra' option is selected.

The main content area is titled 'Lista Negra'. It features a form with a 'Cliente' dropdown menu (currently showing 'Seleccione'), an 'Observaciones' text input field, and an 'Agregar' button. Below the form is a table with the following structure:

| # | Cliente | Observaciones | Quitar de esta lista |
|---|---------|---------------|--------------------------|
| | | | <input type="checkbox"/> |
| | | | <input type="checkbox"/> |
| | | | <input type="checkbox"/> |
| | | | <input type="checkbox"/> |
| | | | <input type="checkbox"/> |
| | | | <input type="checkbox"/> |
| | | | <input type="checkbox"/> |
| | | | <input type="checkbox"/> |
| | | | <input type="checkbox"/> |
| | | | <input type="checkbox"/> |
| | | | <input type="checkbox"/> |

Below the table, there is a 'Quitar' button and a pagination control showing 'Pág 1' with navigation arrows.

Ilustración 14: Lista Negra

Fuente: El autor

Si se hace clic en la opción *Lista Negra*, del menú *Administración*, el usuario podrá ingresar clientes a la lista negra. Cuando un cliente está en una lista negra, significa que a éste cliente no se le puede realizar ningún tipo de transacción a crédito.

Para ingresar in cliente en la Lista Negra basta con seleccionarlo en el campo Cliente, y agregar algún tipo de observación, esta observación se hará visible en la tabla que se muestra en el formulario, también se hará visible en el *Estado del cliente*, ubicado en el menú de consultas, y en transacciones cuando se selecciona un cliente que está en esta lista.

Para eliminar un cliente de esta lista basta con seleccionarlo en última celda del registro, y hacer clic en el botón *Quitar*

- Créditos

The screenshot shows the 'SIMPLE DATA' web application interface. The header features the application name 'SIMPLE DATA' and a 'Logout' link. Below the header, the user is logged in as 'Usuario' with the profile of 'Administrador'. The main navigation menu on the left includes 'Clientes', 'Productos', 'Transacciones', 'Consultas', and 'Administración'. The 'Administración' menu is expanded, showing options like 'Perfiles', 'Usuarios', 'Lista Negra', 'Créditos', and 'Políticas de reabastecimiento'. The 'Créditos' option is selected, leading to the 'Gestión de Créditos' page. This page contains a form with three input fields: 'Código(Autogenerado)', 'Nombre', and 'Tope', followed by a 'Guardar' button. Below the form is a table with columns for 'Item #', 'Nombre', 'Código', 'Tope', 'Modificar', and 'Eliminar'. The table is currently empty. At the bottom of the table, there is an 'Eliminar' button and a pagination control showing 'Pág 1'.

Ilustración 15: Gestión de Créditos

Fuente: El autor

Al hacer clic en la opción *Créditos* del menú *Administración*, se podrán realizar las acciones del proceso común: Crear, Modificar y Eliminar créditos.

Cuando se crea un cliente, se debe seleccionar un crédito, de los que se han creado con anterioridad en este formulario, el crédito define un nombre y un tope, este tope se ve reflejado en las transacciones, cuando se está realizando una venta a crédito, y se está intentando ingresar un producto con el cual se superaría el tope de crédito autorizado ya que el sistema no lo permite.

Notas:

El sistema tiene por defecto un Crédito llamado sin Crédito donde el tope es 0, este se selecciona para usuario que no tienen ningún crédito.

- Políticas de Reabastecimiento

SIMPLE DATA Logout

Bienvenido: Usuario - Perfil: Administrador

Menú Principal

- Clientes
- Productos
- Transacciones
- Consultas
- Administración
 - Perfiles
 - Usuarios
 - Lista Negra
 - Créditos
 - Políticas de reabastecimiento

Políticas de Reabastecimiento

Producto Cantidad Mínima Observaciones

| # | Producto | Observaciones | Modificar | Eliminar |
|---|----------|---------------|-----------------------|--------------------------|
| | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | <input type="radio"/> | <input type="checkbox"/> |

<< < Pág 1 > >>

Ilustración 16: Políticas de reabastecimiento.

Fuente: El autor

Al hacer clic en la opción *Políticas de Reabastecimiento*, el usuario podrá crear políticas de reabastecimiento para algún producto. La política de reabastecimiento indica que cuando quede cierta cantidad en inventario, esto generará una alerta, que indica la cantidad en inventario de los productos que tienen políticas, mostrando también la observación. Estas alertas serán visibles en un banner ubicado en la parte inferior de la aplicación.

3.3. Diseño del Sistema propuesto

Debido a que el modelo de desarrollo XP, es un modelo de desarrollo ágil, éste no genera mucha documentación, lo cual no quiere decir que no sea importante la ésta etapa, también es necesario diseñar el sistema, para generar una vista previa del proyecto, y tener claro el resultado que se desea obtener a nivel de software.

Para el diseño del sistema se ha propuesto realizar diagramas de casos de uso, y diagrama de actividades a nivel dinámico, y a nivel estático se decide realizar el diagrama de clases, y diagrama de despliegue.

3.3.1. Diagramas Dinámicos

Los diagramas dinámicos se usan para expresar y modelar el comportamiento del sistema a lo largo del tiempo, con el que describe las relaciones temporales entre objetos. Muestran las interacciones entre objetos ocurridas en un escenario del sistema. (Camacho, 2011)

- **Casos de Uso**

Luego de haber realizado dos reuniones con el cliente se han definido todos los requerimientos del sistema, y se han aclarado todas las dudas, por lo cual se procede a la realización de casos de uso. Se concibe un caso de uso por cada funcionalidad del sistema.

Los casos de uso son una excelente ayuda para resolver todo tipo de dudas, que se tengan a nivel de negocio con el cliente, ya que dan una perspectiva más clara acerca de las necesidades que tiene la organización

- **Casos de Uso Usuario Administrador**

Teniendo en cuenta que en los requerimientos se define que el usuario está en capacidad de crear perfiles y asignar permisos en el sistema, el sistema se entregará únicamente con un perfil por defecto, el perfil administrador, y por tanto todos los casos de uso que se muestran a continuación se relacionan directamente con el rol de administrador, y en general con todos los roles del sistema que el usuario defina una vez se encuentre en producción el aplicativo.

| | |
|---|--|
| <p>Caso de Uso creación de usuario</p> <p>Los datos se refieren a Nombre de Usuario y contraseña</p> <p>Administrador</p> <p>Los Perfiles deben haber sido creados con anterioridad</p> <p>Detailed description: The diagram shows an actor 'Administrador' connected to a use case 'Crear Usuario'. This use case includes two other use cases: 'Digitar Datos' and 'Seleccionar Perfil'. Two notes are present: one at the top stating 'Los datos se refieren a Nombre de Usuario y contraseña' and one at the bottom stating 'Los Perfiles deben haber sido creados con anterioridad'.</p> | <p>1. Nombre: Crear usuario</p> <p>2. Descripción: Se digitan los datos Usuario y Contraseña, se selecciona un perfil de sistema, y se guarda</p> <p>3. Flujo de Eventos:</p> <p>3.1. <i>Flujo Básico:</i> Ingresa al sistema → Administración→ Usuarios, → Diligencia Datos → Selecciona Perfil → Guardar</p> <p>3.2. <i>Flujo alternativo:</i> Datos incompletos → Error // Contraseña < 6 Caracteres → Error</p> <p>4. Precondiciones:</p> <ul style="list-style-type: none"> - Estar Registrado, Haber ingresado, Tener Privilegios <p>5. Post condiciones:</p> <ul style="list-style-type: none"> - Usuario almacenado, Ingreso con nuevos datos |
| <p>Caso de Uso Modificación de Usuario</p> <p>Los datos se refieren a Nombre de Usuario y contraseña</p> <p>Administrador</p> <p>Los Perfiles deben haber sido creados con anterioridad</p> <p>Detailed description: The diagram shows an actor 'Administrador' connected to a use case 'Modificar Usuario'. This use case includes three other use cases: 'Busca Usuario', 'Actualizar Datos', and 'Cambiar Perfil'. Two notes are present: one at the top stating 'Los datos se refieren a Nombre de Usuario y contraseña' and one at the bottom stating 'Los Perfiles deben haber sido creados con anterioridad'.</p> | <p>1. Nombre: Modificar usuario</p> <p>2. Descripción: Se actualizan los datos Usuario y Contraseña, se actualiza perfil de sistema, y se guarda.</p> <p>3. Flujo de Eventos:</p> <p>3.1. <i>Flujo Básico:</i> Ingresa al sistema → Administración→ Usuarios, → Buscar Usuario → Diligencia Datos → Selecciona Perfil → Guardar</p> <p>3.2. <i>Flujo alternativo:</i> Datos incompletos → Error / Contraseña < 6 Caracteres → Error</p> <p>4. Precondiciones:</p> <ul style="list-style-type: none"> - Estar Registrado, Haber ingresado, Tener Privilegios <p>5. Post condiciones:</p> <ul style="list-style-type: none"> - Usuario modificado, Ingreso con nuevos datos. |
| <p>Caso de Uso Eliminar Usuario</p> <p>Administrador</p> <p>Detailed description: The diagram shows an actor 'Administrador' connected to a single use case 'Eliminar Usuario'.</p> | <p>1. Nombre: Eliminar usuario</p> <p>2. Descripción: Se busca usuario y se elimina</p> <p>3. Flujo de Eventos:</p> <p>3.1. <i>Flujo Básico:</i> Ingresa al sistema → Administración→ Usuarios, → Buscar Usuario → Eliminar</p> <p>4. Precondiciones:</p> <ul style="list-style-type: none"> - Estar Registrado, Haber ingresado, Tener Privilegios <p>5. Post condiciones:</p> <ul style="list-style-type: none"> - Usuario eliminar, sin ingreso al sistema |

Tabla 1: Casos de Uso Administrador

Fuente: El autor

| | |
|--|--|
| <p style="text-align: center;">Crear Perfil de Usuario</p> <pre> graph LR Admin[Administrador] --- CP((Crear Perfil)) CP -.-> <<include>> DN((Digitar Nombre)) CP -.-> <<include>> SP((Seleccionar Permisos)) </pre> | <p>1. Nombre: Crear Perfil de Usuario</p> <p>2. Descripción: Se digita nombre, y se seleccionan permisos.</p> <p>3. Flujo de Eventos:</p> <p>3.1. <i>Flujo Básico:</i> Ingresa al sistema → Administración→ Perfiles, → Digita Nombre→ Selecciona Permisos</p> <p>3.2. <i>Flujo Alternativo:</i> Nombre vacío → Error</p> <p>4. Precondiciones:</p> <ul style="list-style-type: none"> - Estar Registrado, Haber ingresado, Tener Privilegios <p>5. Post condiciones:</p> <ul style="list-style-type: none"> - Perfil creado, disponible para asignar |
| <p style="text-align: center;">Crear Cliente</p> <pre> graph LR Admin[Administrador] --- CC((Crear Cliente)) CC -.-> <<include>> DD((Digitar datos)) CC -.-> <<include>> STC((Seleccionar Tipo Cliente)) CC -.-> <<include>> STCrd((Seleccionar Tipo Crédito)) </pre> <p>Los tipos de crédito deben haber sido creados con anterioridad</p> | <p>1. Nombre: Crear Cliente</p> <p>2. Descripción: Se digita datos básicos, y se seleccionan tipo cliente y tipo crédito.</p> <p>3. Flujo de Eventos:</p> <p>3.1. <i>Flujo Básico:</i> Ingresa al sistema → Clientes→ Digita Datos→ Selecciona Tipo Cliente y Tipo Crédito</p> <p>3.2. <i>Flujo Alternativo:</i> Datos incompletos → Error</p> <p>4. Precondiciones:</p> <ul style="list-style-type: none"> - Estar Registrado, Haber ingresado, Tener Privilegios <p>5. Post condiciones:</p> <ul style="list-style-type: none"> - Cliente creado, disponible para transacciones |
| <p style="text-align: center;">Modificar Cliente</p> <pre> graph LR Admin[Administrador] --- MC((Modificar Cliente)) MC -.-> <<include>> BC((Buscar Cliente)) MC -.-> <<include>> AD((Actualizar datos)) MC -.-> <<include>> CTC((Cambiar Tipo Cliente)) MC -.-> <<include>> CTCrd((Cambiar Tipo Crédito)) </pre> <p>Los tipos de crédito deben haber sido creados con anterioridad</p> | <p>1. Nombre: Modificar Cliente</p> <p>2. Descripción: Se busca cliente, Se digita datos básicos, y se seleccionan tipo cliente y tipo crédito.</p> <p>3. Flujo de Eventos:</p> <p>3.1. <i>Flujo Básico:</i> Ingresa al sistema → Clientes→ Buscar Cliente → Actualizar Datos→ Cambiar Tipo Cliente y Tipo Crédito</p> <p>3.2. <i>Flujo Alternativo:</i> Datos incompletos → Error</p> <p>4. Precondiciones:</p> <ul style="list-style-type: none"> - Estar Registrado, Haber ingresado, Tener Privilegios <p>5. Post condiciones:</p> <ul style="list-style-type: none"> - Cliente modificado, disponible para transacciones |

Tabla 2: Casos de Uso Administrador
Fuente: El autor

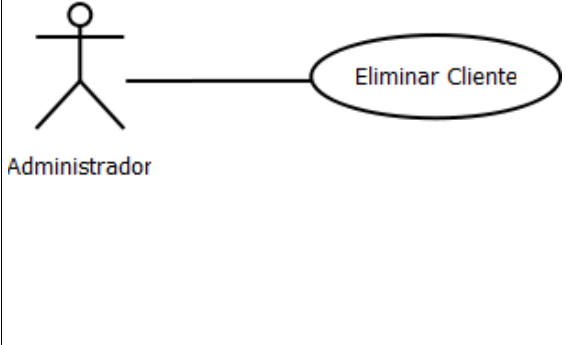
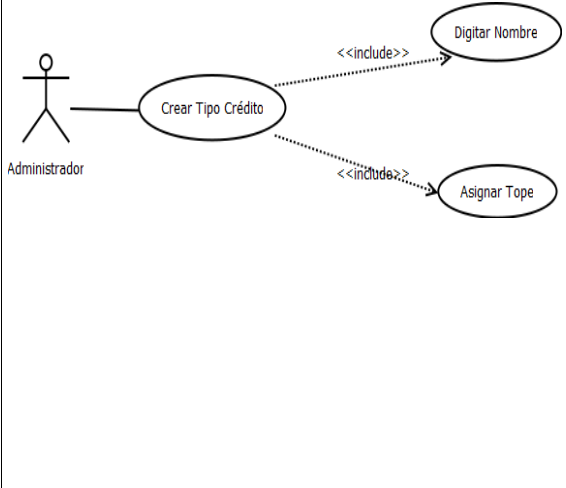
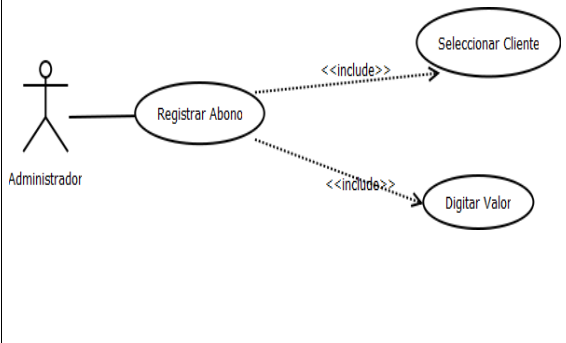
| | |
|--|---|
| <p style="text-align: center;">Eliminar Cliente</p>  | <p>1. Nombre: Eliminar Cliente</p> <p>2. Descripción: Se busca cliente, y se eliminar</p> <p>3. Flujo de Eventos:</p> <p>3.1. Flujo Básico: Ingresa al sistema → Clientes→ Buscar Cliente → Eliminar</p> <p>3.1. Flujo Alternativo: Cliente con pendientes → No elimina</p> <p>4. Precondiciones:</p> <ul style="list-style-type: none"> - Estar Registrado, Haber ingresado, Tener Privilegios <p>5. Post condiciones:</p> <ul style="list-style-type: none"> - Cliente eliminado, no disponible para transacciones |
| <p style="text-align: center;">Crear Tipo de Crédito</p>  | <p>1. Nombre: Crear Tipo de Crédito</p> <p>2. Descripción: Se digita el nombre del tipo de crédito, se asigna tope.</p> <p>3. Flujo de Eventos:</p> <p>3.1. Flujo Básico: Ingresa al sistema → Administración→ Digitar Nombre→ Asignar Tope → Guardar</p> <p>3.1. Flujo Alternativo: Cliente con pendientes → No elimina</p> <p>4. Precondiciones:</p> <ul style="list-style-type: none"> - Estar Registrado, Haber ingresado, Tener Privilegios <p>5. Post condiciones:</p> <ul style="list-style-type: none"> - Tipo de Crédito, disponible para asignar a cliente |
| <p style="text-align: center;">Registrar Abono</p>  | <p>1. Nombre: Crear Tipo de Crédito</p> <p>2. Descripción: Se selecciona cliente y se digita valor</p> <p>3. Flujo de Eventos:</p> <p>3.1. Flujo Básico: Ingresa al sistema → Administración→ Seleccionar Cliente→ Digitar Valor→ Guardar</p> <p>3.1. Flujo Alternativo: Cantidad Negativa, Cantidad Mayor a decuda→ Error</p> <p>4. Precondiciones:</p> <ul style="list-style-type: none"> - Estar Registrado, Haber ingresado, Tener Privilegios <p>5. Post condiciones:</p> <ul style="list-style-type: none"> - Abono se resta de deuda de cliente |

Tabla 3: Casos de Uso Administrador
Fuente: El autor

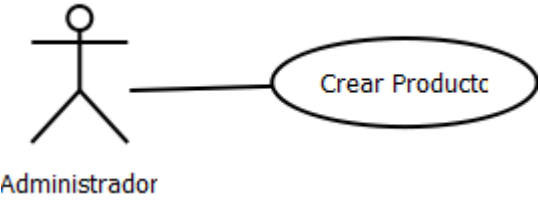
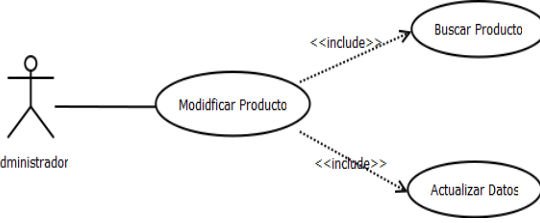
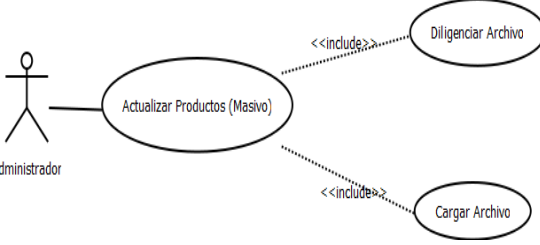
| | |
|---|--|
| <p style="text-align: center;">Crear Producto</p>  <p>Administrador</p> | <p>1. Nombre: Crear Producto</p> <p>2. Descripción: Se digitan datos básicos</p> <p>3. Flujo de Eventos:</p> <p>3.1. Flujo Básico: Ingresa al sistema → Productos→ Digita datos básicos→ Guardar</p> <p>3.1. Flujo Alternativo: Datos incompletos-→ Error / Cantidades Negativas → Error</p> <p>4. Precondiciones:</p> <ul style="list-style-type: none"> - Estar Registrado, Haber ingresado, Tener Privilegios <p>5. Post condiciones:</p> <ul style="list-style-type: none"> - Producto disponible para la venta |
| <p style="text-align: center;">Modificar Producto</p>  <p>Administrador</p> | <p>1. Nombre: Modificar Producto</p> <p>2. Descripción: Se busca producto, y se actualizan datos</p> <p>3. Flujo de Eventos:</p> <p>3.1. Flujo Básico: Ingresa al sistema → Productos→ Buscar Producto → Actualiza Datos básicos→ Guardar</p> <p>3.1. Flujo Alternativo: Datos incompletos-→ Error / Cantidades Negativas → Error</p> <p>4. Precondiciones:</p> <ul style="list-style-type: none"> - Estar Registrado, Haber ingresado, Tener Privilegios |
| <p style="text-align: center;">Actualiza Productos (Carga Masiva)</p>  <p>Administrador</p> | <p>1. Nombre: Actualiza Productos (Carga Masiva)</p> <p>2. Descripción: Se crear y diligencia archivo, se carga al sistema</p> <p>3. Flujo de Eventos:</p> <p>3.1. Flujo Básico: Ingresa al sistema → Productos→ Generar Archivo → Diligenciar Archivo → Cargar Archivo</p> <p>3.1. Flujo Alternativo: Archivo incompatible</p> <p>4. Precondiciones:</p> <ul style="list-style-type: none"> - Estar Registrado, Haber ingresado, Tener Privilegios |

Tabla 4: Casos de Uso Administrador
Fuente: El autor

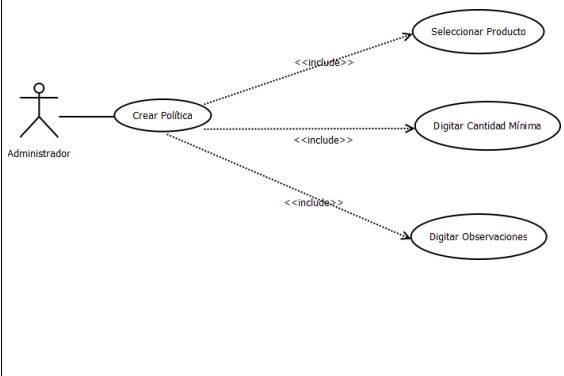
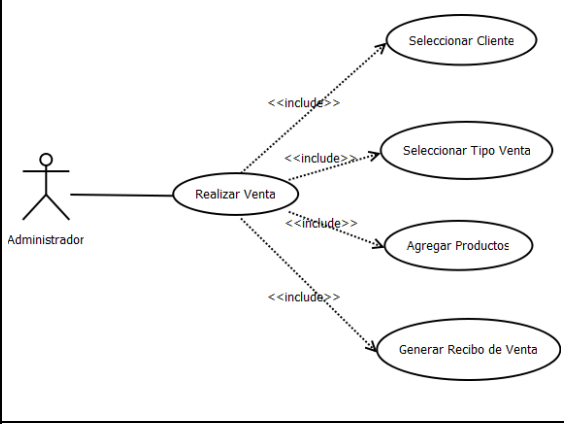
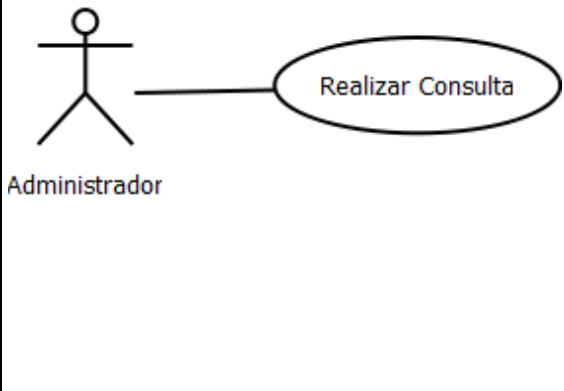
| | |
|--|---|
| <p>Crear Política de Reabastecimiento</p>  | <p>1. Nombre: Crear Política de Reabastecimiento 2. Descripción: Se selecciona producto, se digita cantidad mínima y observaciones 3. Flujo de Eventos: 3.1. <i>Flujo Básico:</i> Ingresar al sistema → Administración → Políticas de Reabastecimiento → Seleccionar Productos → Digitar Cantidad Mínima y Observaciones. 3.1. <i>Flujo Alternativo:</i> Cantidades Negativas → Error 4. Precondiciones: - Estar Registrado, Haber ingresado, Tener Privilegios 5. Pos condiciones: - Política genera alerta de reabastecimiento</p> |
| <p>Realizar Venta</p>  | <p>1. Nombre: Realizar Venta 2. Descripción: Se selecciona Cliente, Tipo Venta, se agregan Productos, se genera Recibo 3. Flujo de Eventos: 3.1. <i>Flujo Básico:</i> Ingresar al sistema → Transacciones → Realizar Transacción → Selecciona Cliente y Tipo Venta → Agrega Productos → Genera Recibo 3.1. <i>Flujo Alternativo:</i> Clientes en lista Negra en venta a crédito → Error / Clientes sin Saldo venta a crédito → Error 4. Precondiciones: - Estar Registrado, Haber ingresado, Tener Privilegios 5. Pos condiciones: - Se registra venta</p> |
| <p>Consultar</p>  | <p>1. Nombre: Consultar 2. Descripción: El cliente filtra y obtiene resultados 3. Flujo de Eventos: 3.1. <i>Flujo Básico:</i> Ingresar al sistema → Consultas → Consultas Generales → Selecciona Tipo Consulta → Filtra → Obtiene Resultados 4. Precondiciones: - Estar Registrado, Haber ingresado, Tener Privilegios 5. Pos condiciones: - Se registra venta</p> |

Tabla 5: Casos de Uso Administrador

Fuente: El autor

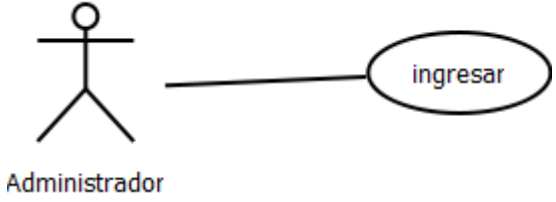
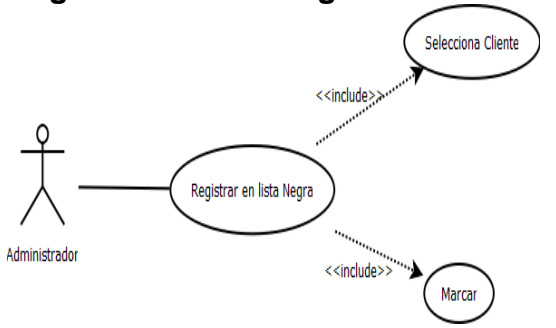
| | |
|---|--|
| <p>Ingreso al Sistema</p>  | <p>1. Nombre: Ingresar al sistema</p> <p>2. Descripción: Usuario digita datos y entra al sistema</p> <p>3. Flujo de Eventos:</p> <p>3.1. <i>Flujo Básico:</i> Digita Usuario y Password → Accede a la página de inicio del sistema</p> <p>3.1. <i>Flujo Alternativo:</i> Password o usuario errados → Error</p> <p>4. Precondiciones:</p> <ul style="list-style-type: none"> - Tener un usuario para ingreso al sistema <p>5. Pos condiciones:</p> <ul style="list-style-type: none"> - Acceso al sistema |
| <p>Registrar en lista Negra</p>  | <p>1. Nombre: Registrar en lista negra</p> <p>2. Descripción: Se selecciona el cliente, y se envía a la lista</p> <p>3. Flujo de Eventos:</p> <p>3.1. <i>Flujo Básico:</i> Ingresa al sistema → Administración → Lista Negra → Cliente → Enviar a la lista (Marcar)</p> <p>3.1. <i>Flujo Alternativo:</i> Cliente ya existente en la lista negra → Error</p> <p>4. Precondiciones:</p> <ul style="list-style-type: none"> - Estar Registrado, Haber ingresado, Tener Privilegios <p>5. Pos condiciones:</p> <ul style="list-style-type: none"> - Cliente no puede comprar a crédito |

Tabla 6: Casos de Uso Administrador

Fuente: El autor

- **Diagrama Casos de Uso Usuario Administrador**

El administrador puede utilizar todas las funcionalidades del sistema

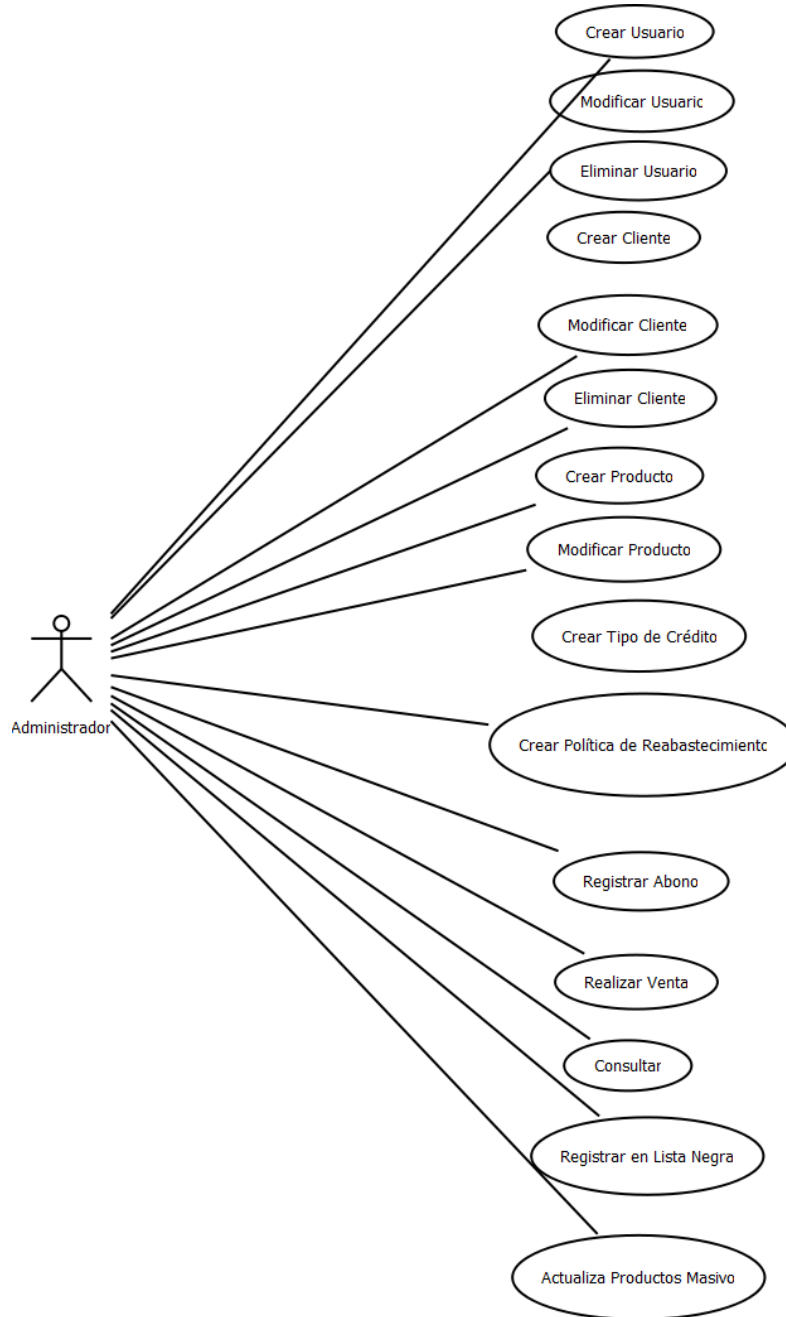


Ilustración 17: Diagrama de Casos de Uso Administrador
Fuente: El autor

- **Diagrama Casos de Uso del Sistema**

Los casos de uso Alertar Política y Restringir son propios del sistema, y es el sistema el que utiliza esas funciones en el momento necesario.

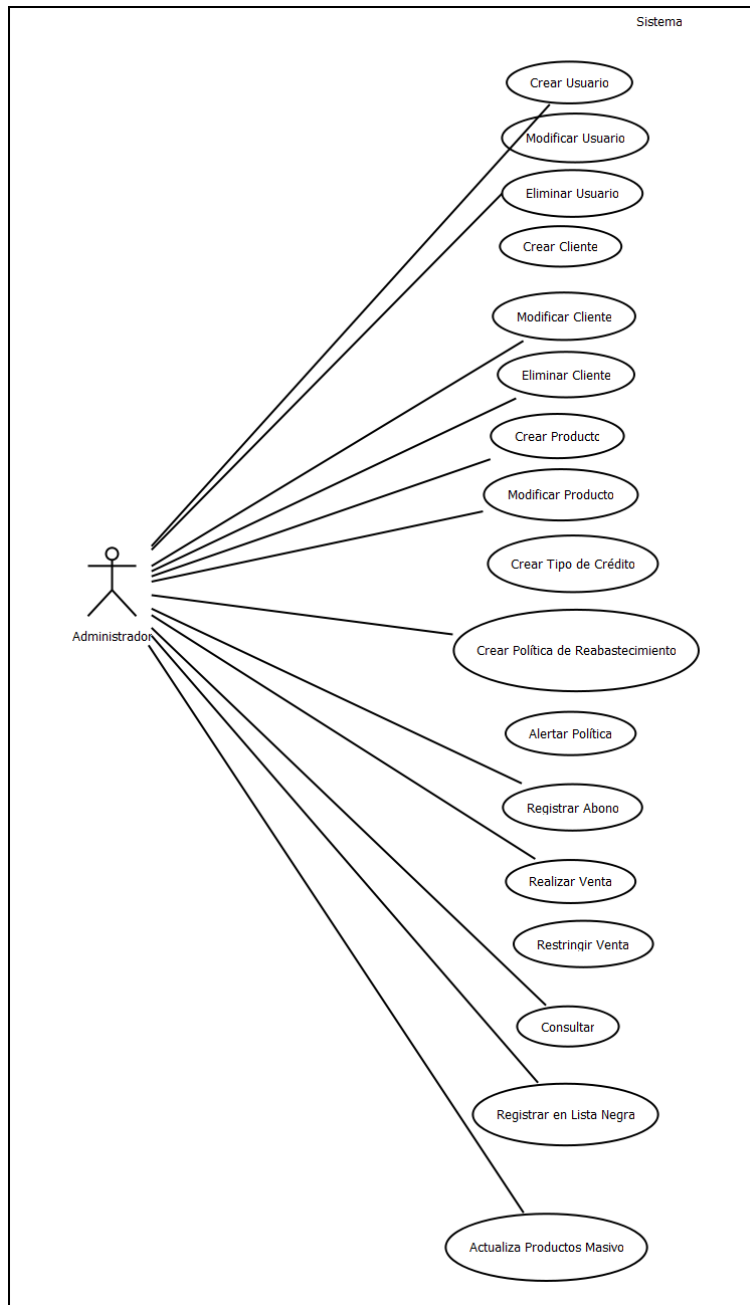


Ilustración 18: Diagrama de Casos de Uso del Sistema

Fuente: El autor

- **Diagrama de Actividades**

En el siguiente diagrama se muestra el flujo de actividades para la realización de una venta, desde el momento en que se selecciona el cliente a quien se le vende, hasta el momento en que finaliza la venta.

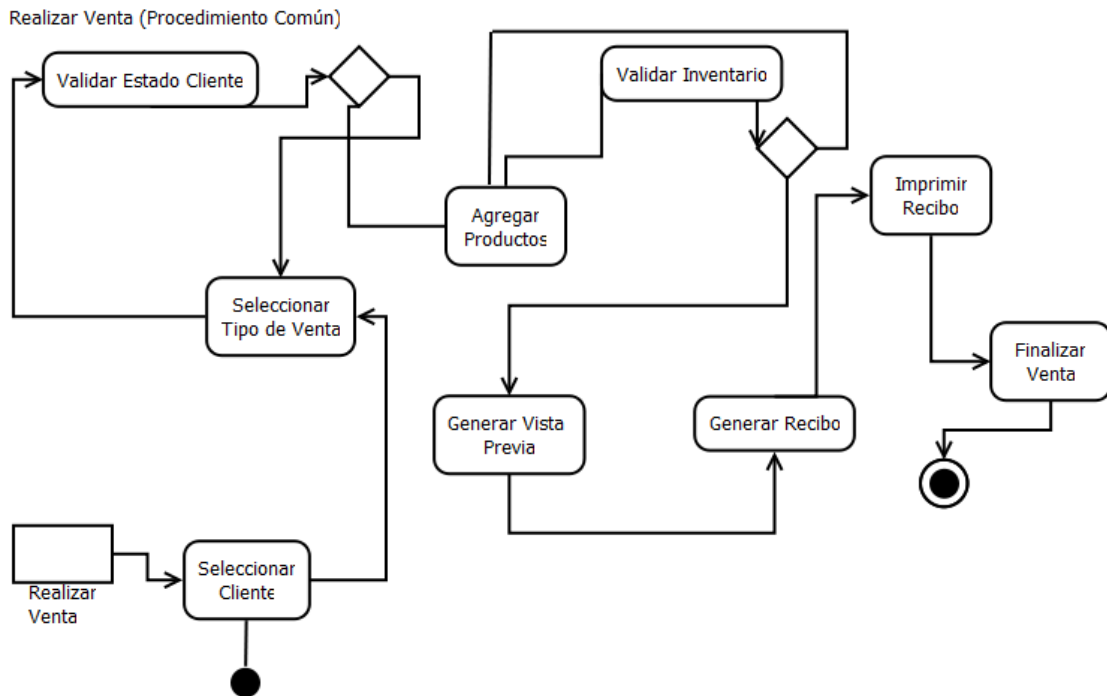


Ilustración 19: Diagrama de Actividades del Sistema
Fuente: El autor

3.3.2. Diagramas Estáticos

Los diagramas estáticos se encargan de definir todos los elementos estructurales dentro del sistema, por ejemplo clases, objetos, componentes, paquetes e.t.c. y muestran cual es la estructura interna de cada uno de ellos. Para el presente proyecto se definieron los siguientes:

- **Diagrama de Clases**

En el siguiente diagrama se define la estructura y la interrelación que hay entre las clases de negocio, en el diagrama se muestran los atributos y los métodos que implementa cada clase.

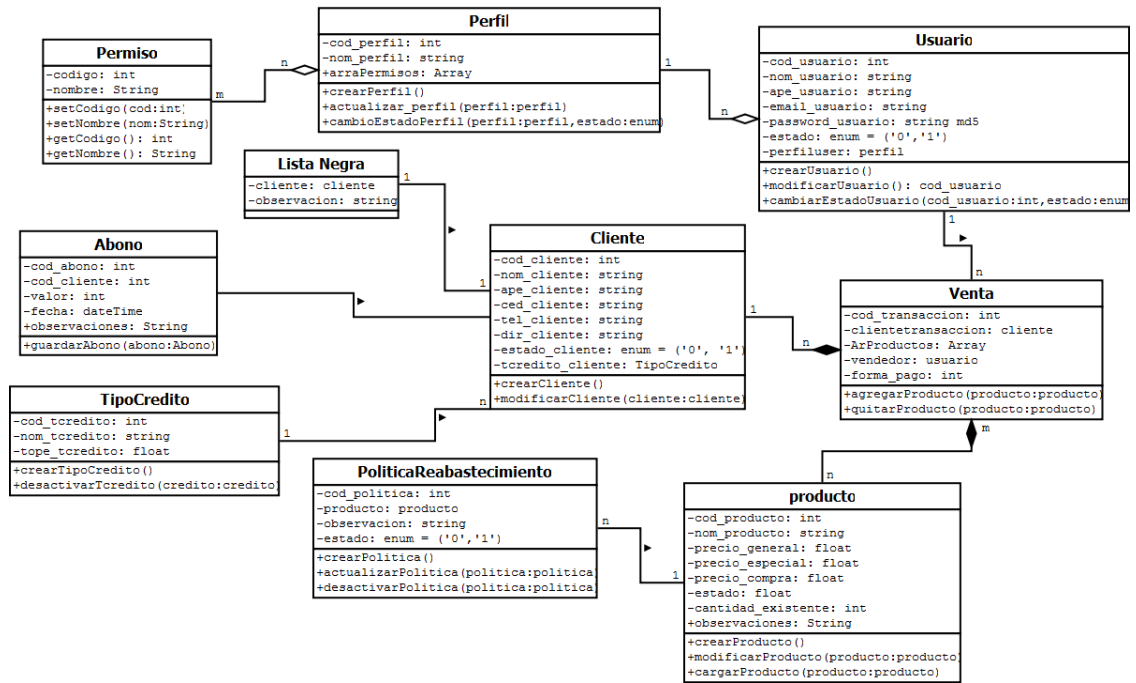


Ilustración 20: Diagrama de Actividades del Sistema

Fuente: El autor

- Diagrama de despliegue

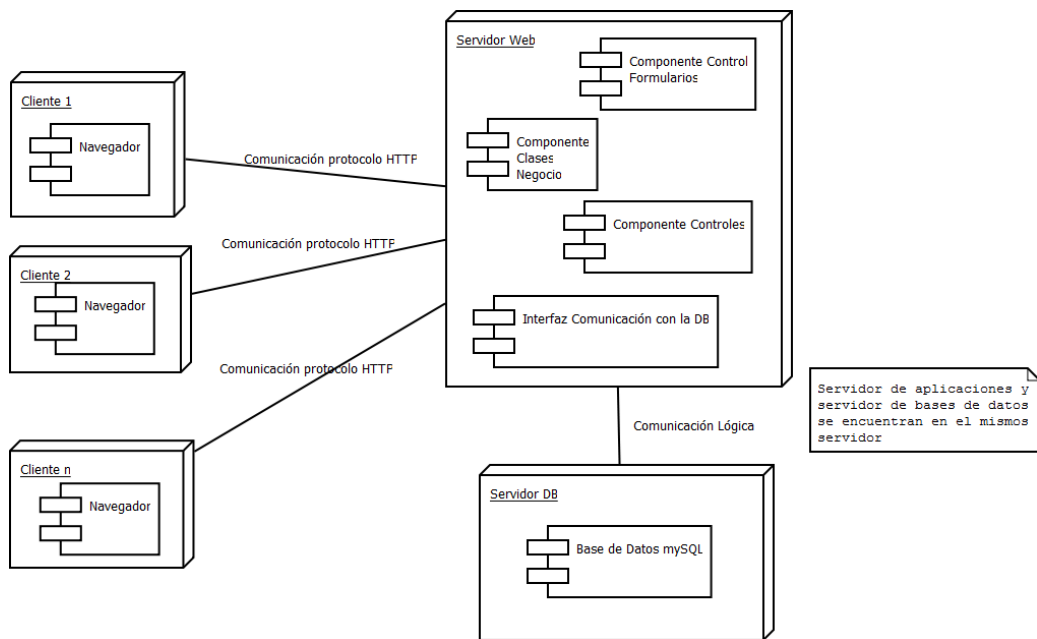


Ilustración 21: Diagrama de Despliegue

Fuente: El autor

- **Modelo Entidad Relación y Modelo Relacional**

Nota: El modelo Entidad relación lo puede encontrar en el anexo 3.

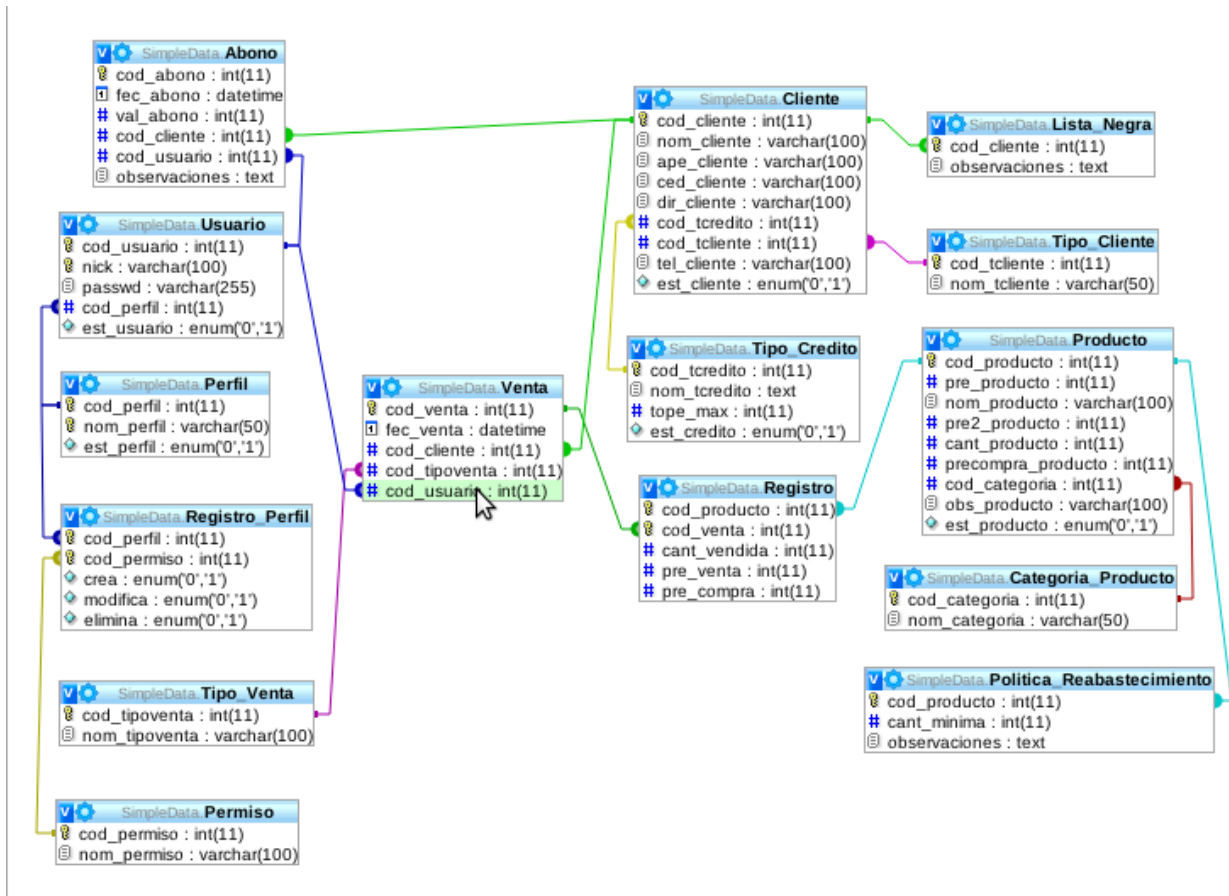


Ilustración 22: Modelo Relacional

Fuente: El autor

- **Diccionario de Datos:**

Es un conjunto de metadatos el cual contiene las características lógicas de los datos que se van a utilizar en el aplicativo, a continuación se documentan todas las tablas, que contiene el aplicativo simple data, para la generación de éste diccionario de datos se utilizó la herramienta Modelador, de phpMyAdmin

- **Abono**

Comentarios de la tabla: En esta tabla se registran los abonos que hacen los clientes

| Columna | Tipo | Nulo | Predeterminado | Enlaces a | Comentarios |
|------------------|----------|------|----------------|---------------------------|---|
| <u>cod_abono</u> | int(11) | No | | | Llave primaria, número consecutivo autogenerado de manera incremental de uno en uno, por cada abono que se almacena |
| fec_abono | datetime | No | | | La fecha en que se realiza el abono |
| val_abono | int(11) | No | | | El valor en pesos del abono recibido |
| cod_cliente | int(11) | No | | Cliente -> cod_cliente | El código del cliente que realiza el abono |
| cod_usuario | int(11) | No | | Usuario -> cod_usuario | El código del usuario del sistema, que recibe y registra el abono |
| Observaciones | text | No | | | Observaciones acerca del abono que realiza un cliente. |

Tabla 7: Diccionario de Datos Tabla Abono:

Fuente: El autor

- **Categoría_Producto**

Comentarios de la tabla: Tabla con datos, sobre las categorías de los productos.

| Columna | Tipo | Nulo | Predeterminado | Comentarios |
|----------------------|-------------|------|----------------|--|
| <u>cod_categoria</u> | int(11) | No | | Llave primaria, valor autoincrementable. |
| nom_categoria | varchar(50) | No | | Nombre de la categoría |

Tabla 8: Diccionario de Datos Tabla Categoría_Producto

Fuente: El autor

- **Cliente**

Comentarios de la tabla: Tabla que almacena los datos básicos de los clientes

| Columna | Tipo | Nulo | Predeterminado | Enlaces a | Comentarios |
|--------------------|----------------|------|----------------|------------------------------|--|
| <u>cod_cliente</u> | int(11) | No | | | Llave primaria, valor autoincrementable. Identifica cada cliente de manera única |
| nom_cliente | varchar(100) | No | | | El primer nombre del cliente (O por el nombre que se le identifique) |
| ape_cliente | varchar(100) | No | | | El primer apellido del cliente |
| ced_cliente | varchar(100) | No | | | Llave única, la cedula de ciudadanía de cliente. |
| dir_cliente | varchar(100) | No | | | La dirección de residencia del cliente |
| cod_tcredito | int(11) | No | | Tipo_Credito -> cod_tcredito | El tipo de crédito que tiene asociado el cliente a su registro |
| cod_tcliente | int(11) | No | | Tipo_Cliente -> cod_tcliente | El código de tipo de cliente que tiene asignado el cliente en su registro |
| tel_cliente | varchar(100) | No | | | Números de teléfono para contactar al cliente |
| est_cliente | enum('0', '1') | No | 1 | | Indica si el cliente es activo (1) o si se encuentra inactivo (0) |

Tabla 9: Diccionario de Datos Tabla Cliente

Fuente: El autor

- **Lista_Negra**

Comentarios de la tabla: Tabla donde se registran clientes a quienes, para restringir ventas a crédito.

| Columna | Tipo | Nulo | Predeterminado | Enlaces a | Comentarios |
|--------------------|---------|------|----------------|------------------------|---|
| <u>cod_cliente</u> | int(11) | No | | Cliente -> cod_cliente | El código del cliente, que se encuentra registrado en la lista negra - Relacionado con la tabla Cliente |
| Observaciones | text | No | | | Las observaciones realizadas en el momento de registrar el cliente en la lista. |

Tabla 10: Diccionario de Datos Tabla Lista_Negra

Fuente: El autor

- **Perfil**

Comentarios de la tabla: Tabla que almacena los perfiles creados dentro de la aplicación

| Columna | Tipo | Nulo | Predeterminado | Comentarios |
|-------------------|----------------|------|----------------|--|
| <u>cod_perfil</u> | int(11) | No | | llave primaria, valor autoincrementable |
| nom_perfil | varchar(50) | No | | nombre del perfil |
| est_perfil | enum('0', '1') | No | 1 | Indica si un perfil se encuentra activo o no |

Tabla 11: Diccionario de Datos Tabla Perfil

Fuente: El autor

- **Permiso**

Comentarios de la tabla: Tabla que almacena los permisos que aplican en el software.

| Columna | Tipo | Nulo | Predeterminado | Comentarios |
|--------------------|--------------|------|----------------|---|
| <u>cod_permiso</u> | int(11) | No | | llave primaria, valor autoincrementable |
| nom_permiso | varchar(100) | No | | Nombre del permiso |

Tabla 12: Diccionario de Datos Tabla Permiso.

Fuente: El autor

- **Politica_Reabastecimiento**

Comentarios de la tabla: En esta tabla se almacenan, las políticas establecidas para productos específicos.

| Columna | Tipo | Nulo | Predeterminado | Enlaces a | Comentarios |
|---------------------|---------|------|----------------|--------------------------|--|
| <u>cod_producto</u> | int(11) | No | | Producto -> cod_producto | llave primaria, y forenea a la vez. |
| cant_minima | int(11) | No | | | indica la cantidad minima que debe haber de un producto para mostrar la alerta |
| Observaciones | text | No | | | Las observaciones que se muestran en la alerta |

Tabla 13. Diccionario de Datos Tabla Politica_Reabastecimiento

Fuente: El autor

- **Producto**

Comentarios de la tabla: Tabla que almacena datos sobre productos, y existencias.

| Columna | Tipo | Nulo | Predeterminado | Enlaces a | Comentarios |
|---------------------|--------------|------|----------------|-----------------------|--|
| <u>cod_producto</u> | int(11) | No | | | Llave primaria, valor autoincrementable, Se usa como referencia de los productos |
| pre_producto | int(11) | No | | | Precio general del producto |
| nom_producto | varchar(100) | No | | | El nombre comercial del producto |
| pre2_producto | int(11) | No | | | Precio del producto, dirigido a clientes de tipo mayorista |
| cant_producto | int(11) | No | | | Cantidad existente en inventario del producto |
| precompra_producto | int(11) | No | | | Precio de compra del producto (Para calcular ganancia posteriormente) |
| cod_categoria | int(11) | No | | Categoria_Producto -> | Categoria del producto, relacionada con la tabla |

| | | | | | |
|--------------|----------------|----|---|---------------|--|
| | | | | cod_categoria | Categoría_Producto |
| obs_producto | varchar(100) | No | | | Observaciones realizadas al producto |
| est_producto | enum('0', '1') | No | 1 | | Indica si el producto está activo (1) o inactivo (0) |

Tabla 14: Diccionario de Datos Tabla Producto.

Fuente: El autor

- Registro

Comentarios de la tabla: Tabla intermedia, entre ventas y productos.

| Columna | Tipo | Nulo | Predeterminado | Enlaces a | Comentarios |
|---------------------|---------|------|----------------|--------------------------|--|
| <u>cod_producto</u> | int(11) | No | | Producto -> cod_producto | Llave primaria compuesta junto a cod_venta (No se puede repetir el mismo producto en una venta más de una vez) - Relacionado con la tabla Producto |
| <u>cod_venta</u> | int(11) | No | | Venta -> cod_venta | Llave primaria compuesta junto a cod_producto (No se puede repetir el mismo producto en una venta más de una vez) - Relacionado con la tabla Venta |
| cant_vendida | int(11) | No | | | Cantidad vendida del producto (Cuando se registra venta, disminuye esta cantidad de inventario Tabla Productos)) |
| pre_venta | int(11) | No | | | Precio del producto en el momento de la venta, se registra porque los valores en Producto están sujetos a cambios, y se necesita el valor del momento de la venta, sin importar si posteriormente cambia en Producto (Fidelidad de la información)) |
| pre_compra | int(11) | No | | | Precio de compra del producto en el momento de la venta, se registra porque los valores en Producto están sujetos a cambios, y se necesita el valor del momento de la venta, sin importar si posteriormente cambia en Producto (Fidelidad de la información) |

Tabla 15: Diccionario de Datos Tabla Registro

Fuente: El autor

- Registro_Perfil

Comentarios de la tabla: Tabla para registrar, la relación entre un perfil y sus permisos.

| Columna | Tipo | Nulo | Predeterminado | Enlaces a | Comentarios |
|--------------------|----------------|------|----------------|------------------------|--|
| <u>cod_perfil</u> | int(11) | No | | Perfil -> cod_perfil | llave primaria, y forenea a la vez. |
| <u>cod_permiso</u> | int(11) | No | | Permiso -> cod_permiso | llave primaria, y forenea a la vez. |
| Crea | enum('0', '1') | No | | | indica si el perfil tiene permisos de escritura |
| Modifica | enum('0', '1') | No | | | indica si el perfil tiene permisos de modificación |

| | | | | | |
|---------|----------------|----|--|--|---|
| Elimina | enum('0', '1') | No | | | indica si el perfil tiene permisos de eliminacion |
|---------|----------------|----|--|--|---|

Tabla 16: Diccionario de Datos Tabla Registro_Perfil

Fuente: El autor

- Tipo_Cliente

Comentarios de la tabla: Tabla estática con los tipos de cliente válidos para el negocio

| Columna | Tipo | Nulo | Predeterminado | Comentarios |
|---------------------|-------------|------|----------------|--|
| <u>cod_tcliente</u> | int(11) | No | | Llave primaria, valor autoincrementable. |
| nom_tcliente | varchar(50) | No | | El nombre que tiene el tipo de cliente |

Tabla 17: Diccionario de Datos Tabla Tipo Cliente

Fuente: El autor

- Tipo_Credito

Comentarios de la tabla: tabla que almacena los tipos de crédito validos

| Columna | Tipo | Nulo | Predeterminado | Comentarios |
|---------------------|----------------|------|----------------|---|
| <u>cod_tcredito</u> | int(11) | No | | Llave primaria, valor autoincrementable. |
| nom_tcredito | Text | No | | El nombre que recibe el tipo de crédito |
| tope_max | int(11) | No | | El tope máximo (Es decir el valor máximo de crédito que tiene un cliente) |
| est_credito | enum('0', '1') | No | 1 | indica si es tipo de crédito está activo o no |

Tabla 18: Diccionario de Datos Tabla Tipo_Credito

Fuente: El autor

- Tipo_Venta

Comentarios de la tabla: tabla estática con los tipos de venta válidos para el negocio

| Columna | Tipo | Nulo | Predeterminado | Comentarios |
|----------------------|--------------|------|----------------|---|
| <u>cod_tipoventa</u> | int(11) | No | | Llave primaria, valor autoincrementable |
| nom_tipoventa | varchar(100) | No | | Nombre del tipo de venta. |

Tabla 19: Diccionario de Datos Tabla Tipo_Venta

Fuente: El autor

- **Usuario**

Comentarios de la tabla: Tabla que almacena los datos de los usuarios de la aplicación

| Columna | Tipo | Nulo | Predeterminado | Enlaces a | Comentarios |
|--------------------|----------------|------|----------------|----------------------|--|
| <u>cod_usuario</u> | int(11) | No | | | Llave primaria, valor autoincrementable, código de uso interno dentro del aplicativo |
| Nick | varchar(100) | No | | | Nombre para acceder al aplicativo |
| Passwd | varchar(255) | No | | | Contraseña codificación MD5. |
| cod_perfil | int(11) | No | | Perfil -> cod_perfil | El código del perfil, al que el usuario está asociado |
| est_usuario | enum('0', '1') | No | 1 | | Indica si un usuario, está activo dentro del sistema o no |

Tabla 20: Diccionario de Datos Tabla Usuario

Fuente: El autor

- **Venta**

Comentarios de la tabla: Tabla que almacena datos básicos sobre las ventas realizadas

| Columna | Tipo | Nulo | Predeterminado | Enlaces a | Comentarios |
|------------------|----------|------|----------------|-----------------------------|---|
| <u>cod_venta</u> | int(11) | No | | | Llave primaria, valor autoincrementable, se utiliza como consecutivo para los recibos generados |
| fec_venta | datetime | No | | | Fecha y hora en que se realizó la venta |
| cod_cliente | int(11) | No | | Cliente -> cod_cliente | codigo del cliente que compró, relacionado con la tabla Cliente |
| cod_tipoventa | int(11) | No | | Tipo_Venta -> cod_tipoventa | Codigo del tipo de venta realizada, relacionado con la tabla Tipo_Venta |
| cod_usuario | int(11) | No | | Usuario -> cod_usuario | Código del usuario que realizó la venta, relacionado con la tabla Usuario |

Tabla 21: Diccionario de Datos Tabla Venta:

Fuente: El autor

4. DESARROLLO

Fue necesario tener cuatro reuniones con el cliente, en las cuales el cliente pudo interactuar con los requerimientos desarrollados, se creó un formato de acta, el cual se utilizó en todas las reuniones (Ver anexo), donde se especificaban los siguientes requerimientos a desarrollar, de la misma manera en cada reunión se detectaban errores en el funcionamiento, y se planeaba el respectivo plan de corrección. Durante la etapa de desarrollo no surgieron nuevos requerimientos, por lo cual el desarrollo de las iteraciones se pudo dar de una manera fluida, utilizando para ellos 90 días de desarrollo planeados de la siguiente manera:

- Iteración 1:

Persistencia: Construcción de la Base de Datos, utilizando el gestor de bases de datos MySQL⁷, la base de datos fue construida teniendo en cuenta las especificaciones hechas en el capítulo de diseño.

Interfaces y Estilos: Construcción del ambiente gráfico, y creación de las hojas de estilo (CSS)

- Iteración 2:

Sistema de Seguridad: Construcción del sistema de autenticación (Login), e implementación de Perfiles y Permisos (PHP y HTML).

CRUD: Desarrollo del CRUD, para la realización del proceso común (Crea, Consulta, Modifica, Elimina/Desactiva), para esto se utilizó Ajax⁸

- Iteración 3.

Lógica del Negocio: Construcción de toda la lógica relacionada con el negocio, implementación de todos los métodos y clases más importantes dentro de la aplicación. (Todo esto utilizando los diagramas realizados en la etapa de diseño).

⁷ Gestor de Bases de Datos Relacionales de Propiedad de Oracle Corporation

⁸ JavaScript Asíncrono y XML (“*Ajax no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes.*” (Garrett, 2005))

- Iteración final.

Ajustes: Verificación de toda la lista de requerimientos, realización de pruebas relacionadas con la funcionalidad del software,

- *Implementación:* del software en un hosting gratuito Zymic.com⁹, con el fin de realizar pruebas y poder mostrar un demo de la aplicación al cliente.

Teniendo en cuenta que la metodología de desarrollo XP, no implica realizar una documentación muy extensa, por el contrario está más enfocada a los resultados hacia el cliente, es decir al desarrollo del código fuente como tal, que es lo que finalmente el cliente quiere ver funcionando. Los esfuerzos para la realización del proyecto estuvieron siempre más enfocados hacia la codificación, y a mostrar avances significativos al cliente para que éste evaluara si realmente lo que se estaba codificando, era lo que el cliente había solicitado.

Para el desarrollo es importante contemplar todas las herramientas que se utilizan tanto a nivel de hardware como a nivel de software, debido a que esto causa un impacto directo en el correcto desempeño del aplicativo

4.1. Especificaciones Técnicas.

El aplicativo debe tener unas condiciones determinadas, para que su funcionamiento sea óptimo, y para garantizar la alta disponibilidad de la información, por lo cual se establecen unos requisitos mínimos en su instalación, los cuales se especifican a continuación.

4.1.1. Software

Para el funcionamiento óptimo del aplicativo se requiere instalar un servidor con las siguientes características:

- Servidor web HTTP Apache,
- Gestor de bases de datos relacionales mySQL,

⁹ Hosting, que ofrece servicios gratuitos (PHP, mySQL, FTP) para alojar aplicaciones web.

- Si utiliza un sistema operativo Windows, puede utilizar alguna de las siguientes herramientas Xampp, Wamp, Appserv.¹⁰ (Si no conoce las herramientas puede consultar el anexo de documentación técnica)
- Se recomienda un sistema operativo Linux, y la instalación de las herramientas Apache2 y MySQL de manera independiente, para obtener sus últimas versiones. (Si no conoce la instalación puede consultar el anexo de documentación técnica)
- Una conexión a internet (Superior a 1mb)

Requerimientos Mínimos:

- Navegador Web.
- Una conexión a la red. (Internet, Intranet) dependiendo de donde se encuentre alojado el aplicativo.
- Teclado y Mouse

Requerimientos Recomendados:

- Cualquier versión Google Chrome
- Conexión a la red de 1mb
- Teclado y Mouse

¹⁰ Herramientas libres, que permiten realizar la instalación por defecto del servidor Apache, junto con MySQL y PHP

5. GLOSARIO

PHP: PHP (acrónimo de *PHP: Hypertext Preprocessor*) es un lenguaje de código abierto muy popular especialmente adecuado para desarrollo web y que puede ser incrustado en HTML. (Mehdi Achour, 06)

Apache: es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.12 y la noción de sitio virtual. (Wikipedia, 2011)

MySQL: Gestor de bases de datos relacionales propiedad de Oracle Corporation

HTML: HTML es el lenguaje que se emplea para el desarrollo de páginas de internet.

XP: Metodología de desarrollo utilizada para crear proyectos de software de una manera ágil.

AJAX: AJAX son las siglas de Asynchronous JavaScript And XML. No es un lenguaje de programación sino un conjunto de tecnologías (HTML-JavaScript-CSS-DHTML-PHP/ASP.NET/JSP-XML) que permiten hacer páginas de internet más interactivas. (Ajax Ya)

POO: La programación orientada a objetos o POO (OOP según sus siglas en inglés) es un paradigma de programación que usa objetos y sus interacciones, para diseñar aplicaciones y programas informáticos. (Wikipedia, 2011)

UML: Lenguaje Unificado de Modelado TM (UML [®]) es un lenguaje visual para especificar, construir y documentar los artefactos de un sistema. (IBM, 2011)

Hosting: Es un servicio que provee a los usuario, almacenar información, para que pueda acceder a ella desde cualquier lugar.

6. CONCLUSIONES

Se logró realizar el ejercicio de realizar el proceso de un desarrollo de software en un contexto real, utilizando un cliente real, de esta manera fue posible realizar un acercamiento a problema de la vida real, al cual fue posible dar una solución tecnológica a la medida, que pudo suplir las necesidades del cliente en parte considerable.

El uso de una metodología de desarrollo ágil como XP, permitió tener una buena interacción con el cliente, a tal punto, que el cliente se considera una parte más del equipo de trabajo, ya que sus sugerencias o correcciones pudieron ser analizadas y ejecutadas sin causar un alto impacto en las tareas programadas.

El uso de un patrón de diseño, y unas reglas propias a la hora de codificar permite tener un código fuente organizado, funcional y reutilizable, además de generar una estructura que permite realizar mantenimiento sin mucho esfuerzo.

Teniendo en cuenta lo anterior es posible afirmar lo siguiente:

- Fue posible entregar al cliente un software que le brindara seguridad y confiabilidad en el manejo de la información.
- El software garantiza una disponibilidad alta de la información, comparada con la disponibilidad que proporciona tener la información en medios físicos u hojas de cálculo.
- En todo momento la información se encuentra respaldada, en caso de que haya alguna eventualidad con la base de datos, es posible restaurarla debido a los respaldos que se pueden hacer de la misma regularmente.
- La utilización de herramientas libres tanto para el desarrollo, como para el diseño, es una gran ventaja debido a que los costos se reducen de manera considerable.

7. BIBLIOGRAFÍA

Ajax Ya. (s.f.). Recuperado el 2011 de 12 de 18, de <http://www.ajaxya.com.ar/temarios/descripcion.php?cod=8&punto=1>

Camacho, D. (2011). *ingenieriasoftwaredos.wikispaces.com*. Recuperado el 11 de 21 de 2011, de *ingenieriasoftwaredos.wikispaces.com*:
<http://ingenieriasoftwaredos.wikispaces.com/Diagramas+Dinamicos>

Dávila, M. (2005). Los costos del software libre. En M. Dávila, *Software Libre Una visión* (pág. 62). Bogotá.

Garrett, J. J. (18 de 02 de 2005). <http://www.adaptivepath.com>. Recuperado el 15 de Diciembre de 15, de <http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications>

IBM. (s.f.). Recuperado el 2012 de Enero de 03, de <http://www-01.ibm.com/software/rational/uml/>

Mehdi Achour, F. B. (2011 de 12 de 06). *php.net*. Recuperado el 2012 de Diciembre de 18, de <http://www.php.net/manual/es/intro-what-is.php>

Wells, D. (28 de 09 de 2009). <http://www.extremeprogramming.org/>. Recuperado el 13 de 11 de 2011, de <http://www.extremeprogramming.org/>:
<http://www.extremeprogramming.org/>

Wikipedia. (07 de 06 de 2011). *Wikipedia*. Recuperado el 20 de 11 de 2011, de Wikipedia:
<http://es.wikipedia.org/wiki/CRUD>

Wikipedia. (30 de 12 de 2011). *Wikipedia*. Recuperado el 2011 de Enero de 02, de http://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos

Wikipedia. (2011 de 11 de 28). *Wikipedia*. Recuperado el 2011 de 12 de 18, de http://es.wikipedia.org/wiki/Servidor_HTTP_Apache

8. ANEXOS

ANEXO 1: MANUAL TÉCNICO DE SIMPLE DATA

1. INSTALACIÓN

- En Windows:

Para realizar la respectiva instalación del aplicativo en un equipo con sistema operativo Windows, se requiere una de las siguientes herramientas

Xampp: Puede obtenerlo en <http://www.apachefriends.org/es/xampp.html> (Ésta herramienta también está disponible para linux);

Wamp: Puede obtenerlo en <http://www.wampserver.com/en/>

Appserv: Puede obtenerlo en <http://www.appservnetwork.com/>

En cada una de las páginas se encuentra la respectiva documentación acerca de la instalación de éstas herramientas.

- En Linux:

Si utiliza un sistema operativo de tipo Linux por ejemplo Ubuntu o Ubuntu Server, usted podrá instalar todo lo que necesita escribiendo los siguientes comandos en la terminal:

```
Sudo apt-get install apache2  
Sudo apt-get install mysql-server  
Sudo apt-get install phpmyadmin
```

Una vez instalada la herramienta seleccionada, debe descomprimir el archivo SimpleData.zip, lo encontrará en el disco marcado como Simple Data Source, una vez lo haya descomprimido, debe copiar el directorio Simple Data a la carpeta web de su servidor, en el caso de Xampp, es la carpeta htdocs que se encuentra en C://xampp/htdocs, en el caso Wamp y AppServ, es la carpeta www que se encuentra en C://appserv o C://wamp. En Linux ésta carpeta se encuentra en var/www/.

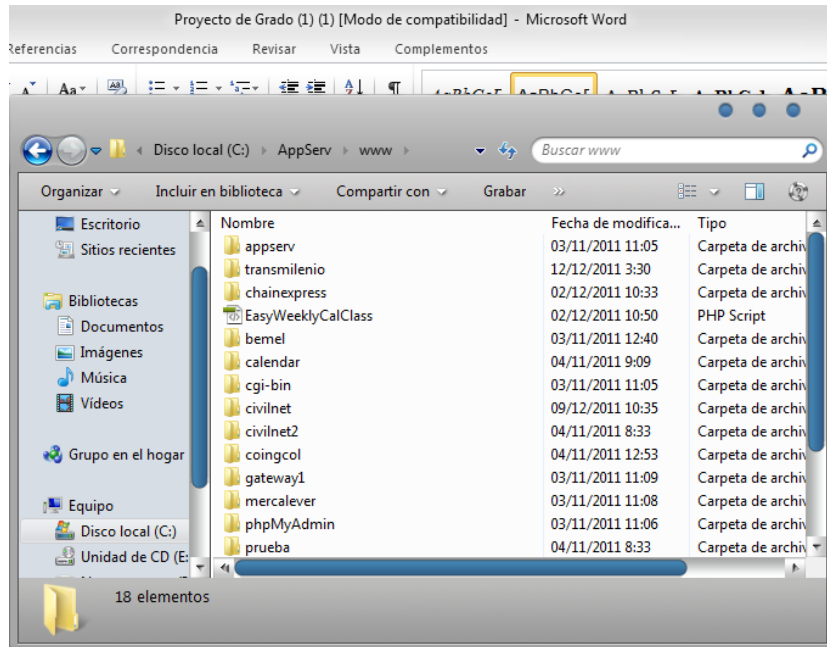


Figura 1: Ubicación del Proyecto en la carpeta web del servidor

Luego debe utilizar la herramienta phpmyAdmin que por defecto se encuentra instalada junto con el servidor instalado en Windows, para ingresar debe digitar en la barra de direcciones de su navegador: localhost/phpmyadmin.

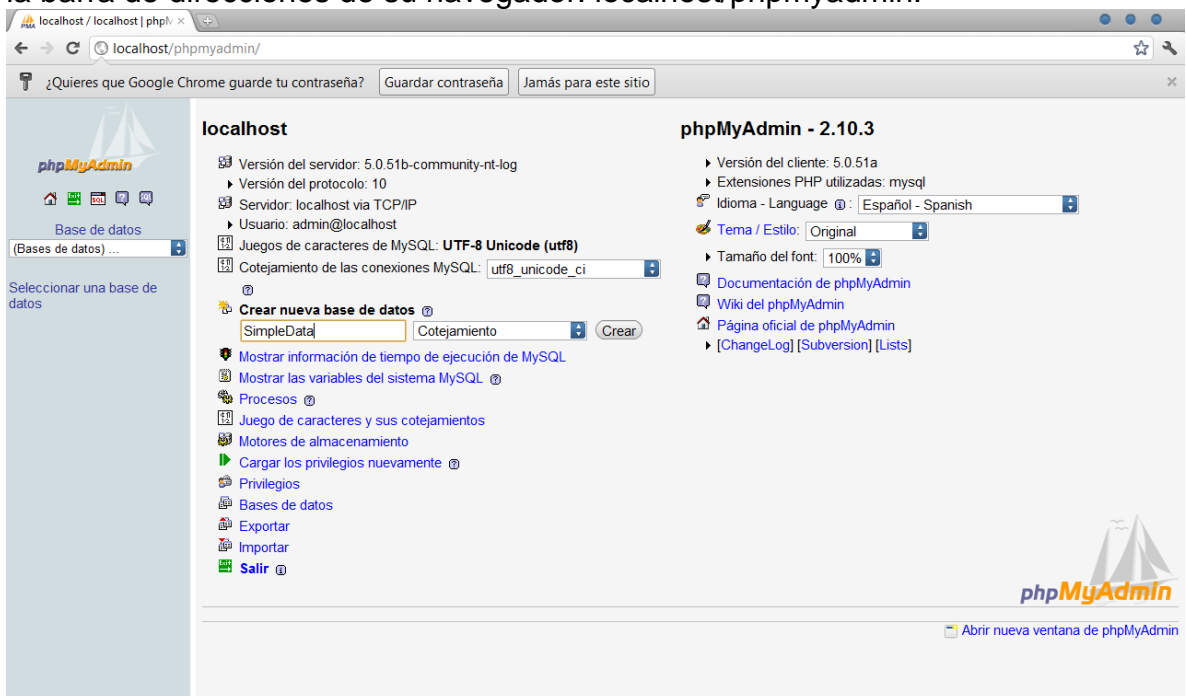


Figura 2: Creación de la base de datos

Una vez haya ingresado al aplicativo, que por defecto no debe tener ninguna contraseña, debe crear la base de datos SimpleData, e importar el archivo SimpleData.sql que se encuentra dentro del disco marcado como Simple Data Source.

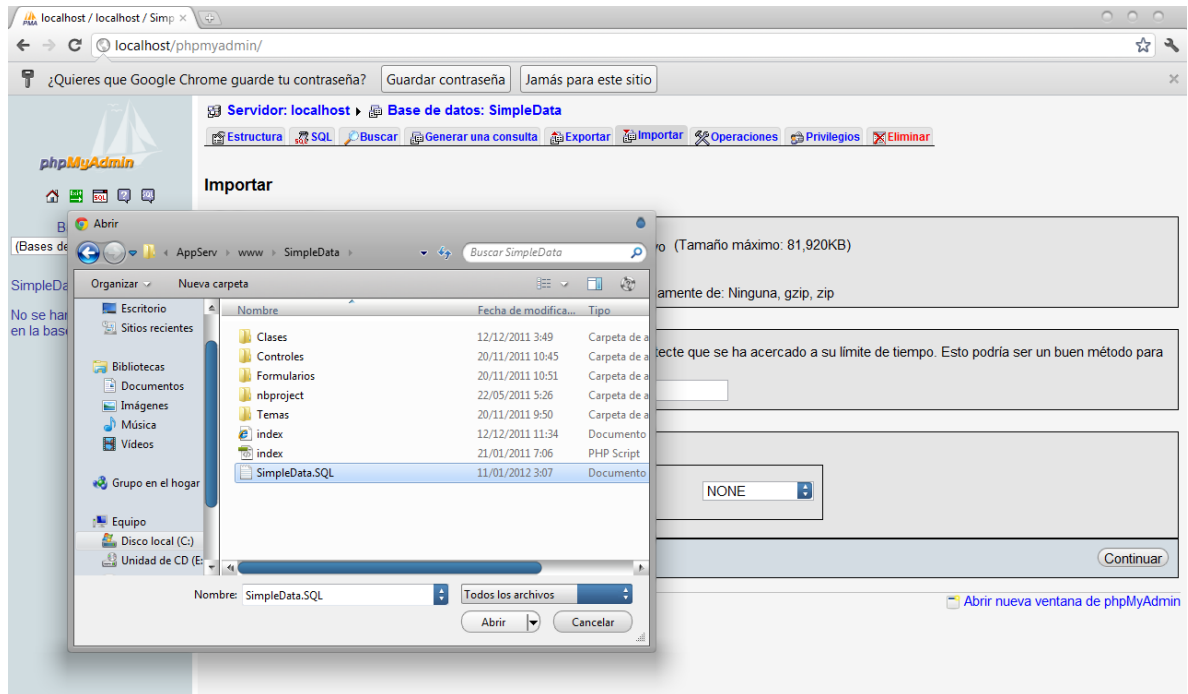


Figura 3: Importando la base de datos

Debe asignar una contraseña para el acceso del aplicativo a la base de datos, en el aplicativo por defecto el usuario es la palabra “admin” y la contraseña es la palabra “admin”, usted puede crear el usuario que desee y usar la contraseña que desee, pero recuerde que debe también configurar éstos parámetros es el archivo DbOperator.php, que se encuentra en la carpeta de clases, dentro de la carpeta del aplicativo.

Una vez ha realizado los pasos anteriores, usted puede ingresar por medio de su navegador a: localhost/SimpleData, y se abrirá una página de login.

Allí usted debe digitar los siguientes datos:

Usuario: adminest

Contraseña: 123456

Una vez haya ingresado, debe inmediatamente cambiar esta contraseña, éste procedimiento está documentado en el manual de usuario.

2. MANTENIMIENTO

Gracias a que el aplicativo se ha desarrollado utilizando el patrón de diseño MVC, es muy fácil realizar mantenimiento al código fuente a continuación se explica la forma correcta de revisar las clases y los scripts.

2.1. ESTRUCTURA:

La estructura básica de la mayoría de los archivos en el aplicativo es la siguiente:
Un Control: Un archivo PHP que maneja todas las peticiones `$_GET`, `$_POST` y `$_REQUEST`, todas éstas peticiones son enviadas siempre desde un formulario.

Una vez una petición llega al control, el control debe decidir si puede dar respuesta a la petición sin acudir a terceros, o si es necesario realizar una consulta, instanciar un objeto, invocar un método etc.

Una Formulario: Contiene toda la parte gráfica con la que se puede afectar la clase de negocio, es decir todo el código html, se encuentra alojado en los formularios, los formularios a su vez también son clases que contienen atributos y métodos, pero éstos atributos y métodos, se utilizan únicamente para manejar el código html del formulario, y son los controles quienes hacen uso de estos métodos siempre.

Una Clase: Actúa como un repositorio de información, un puente entre los controles y los datos propios de la clase, que son extraídos de la base de datos. Las clases de negocio también son usadas para realizar el respectivo CRUD, de las tablas.

2.1.1. Las clases para la conexión con la base de datos.

Adicional a ésta estructura se encuentran otras clases que también manejan lógica de negocio, pero están estrictamente relacionadas con la persistencia, esas clases son `Operador.php`, y `DbOperator.php`, ninguna clase, ni control, puede acceder directamente a la base de datos, las Clases de negocio deben acudir a `DbOperator`, y los controles deben acudir a `Operador`.

La clase `Operador` es únicamente un repositorio de consultas, por lo cual los controles la utilizan únicamente para acceder a la información, pero desde allí la información de la base de datos, no se modifica ni se elimina, además `Operador` es quien abre las conexiones y las cierra, los controles en ningún momento deben preocuparse de estas acciones. Por el contrario desde la clase `DbOperator` se puede realizar cualquier tipo de consulta, esto es idea, para las clases de negocio, que manejan todo tipo de consultas.

2.1.2. La Clase Guardian:

Ésta clase se encarga de la seguridad dentro del aplicativo, contiene un método que aprueba el ingreso al aplicativo, y otro método que aprueba el ingreso a las funciones específicas del sistema.

La administración de permisos se hace utilizando los permisos que se encuentran en la base de datos en la Tabla Permiso, los cuales se relacionan por medio de una tabla intermedia con un la tabla perfil, lo cual permite que el sistema de permisos sea muy flexible.

Gracias a la estructura planteada, es posible realizar mejoras al software, de manera muy sencilla, o agregar nuevas funcionalidades. Sí que sea necesario alterar grandes trozos de código, a continuación se muestra ejemplos de los trozos de código más comunes que se pueden encontrar dentro de la aplicación.

2.1.3. Clases Manejadores de Formularios.

Dentro de las clases manejadoras de formularios, hay una característica común los métodos:

El método, setTheme, es utilizado para importar el código html, que inicialmente se encuentra en la base de datos, pero es importado a la sesión ya que es un valor usado constantemente, esta funcionalidad se agregó con el fin de poder cambiar el tema CSS desde la base de datos sin ningún tipo de complicación.

```
public function setTheme(){  
    $this->theme=$_SESSION['theme'];  
}
```

Éste método contiene el código html que contiene la base del formulario, no es todo el código, debido a que la plantilla que se escribe en este método, es alimentada por los demás métodos de la clase.

```
public function updateHtml ( ) {  
  
    //Aquí se escribe el código html de la página  
  
}
```


Este método se implementa de la misma forma en todos los formularios, se utiliza para autorizar al formulario, que muestre su código.

```
public function setVisible($visible){
    $this->setTheme();
    if($visible==TRUE){
        $this->updateHtml();
        echo $this->html;
    }
    else{
        echo "";
    }
}
```

Todos los métodos restantes, se encargan únicamente de retornar trozos de html, en los formularios por cuestiones de diseño, únicamente es recomendable implementar métodos con la siguiente forma

```
Public function retornarHtml($array, $datos){
    $tabla.="<codigohtml></codigohtml>";
    return table;
}
```

Esto facilita mucho la implementación de peticiones Ajax, todos los formularios por lo general utilizan éste tipo de peticiones que la página web sea dinámica, como era de esperarse las respuestas a las peticiones Ajax, también se encuentran administradas por el controlador correspondiente al formulario.

Una de las peticiones más importantes que se realiza utilizando Ajax, es la petición de paginación, en la mayoría de los formularios se encuentra un método que inicia con la palabra update, éste método utiliza una estructura como la siguiente:

```
updatePrueba(pag, fFiltro1, fFiltro2,){
//Html que contiene campos para realizar filtros

//Seccion que contiene una tabla con la información

//Paginación
}
```

Una vez realizada la petición por vía Ajax, se invoca al controlador, el cual recibe los parámetros, y los envía como parámetros para no perder los filtros, y mostrar la consulta solicitada.

Los controladores no contienen métodos cómo tal, un controlador es únicamente un script PHP, codificado utilizando el paradigma de programación estructurada, esto se hace de esta manera debido a que resulta cómodo realizar un archivo simple, que tiene la capacidad de controlar clases, que utilizan el paradigma orientado a objetos, además de esto PHP es un lenguaje que soporta ambos paradigmas implementados de manera simultánea, y es una ventaja que decidió aprovecharse.

Un ejemplo de una petición atendida por un controlador sería el siguiente:

```
If($_REQUEST['imprimir']=='si'){
    $operador= new Operador();
    $frmImprimir=new FrmImprimir();
    $frmImprimir->setFormato($operador->darFormato($_GET['cod']));
    $frmImprimir->setVisible(TRUE);
}
```

Como se puede observar en el código de ejemplo, se recibe un parámetro ya sea por POST, GET, o REQUEST, y se crea una instancia de formulario, del cual se utilizan sus métodos internos, para entregarle una respuesta, ésta respuesta se obtiene utilizando el repositorio de consultas Operador, absolutamente todos los formularios implementan éstas política, es decir un formulario por sí sólo no puede dar una respuesta por pantalla, es necesario instanciar el formulario como objeto, y utilizar sus métodos para generar un formulario dinámico.

3. El modelo

Como se mencionó anteriormente, las clases del modelo actúan como repositorios de datos, así que la mayoría de las clases, implementan únicamente métodos comunes, entre ellos setters y getters, y métodos específicos para:

- Guardar la información en la base de datos
- Eliminar o desactivar
- Actualizar los datos
- Obtener un registro de la base de datos como un objeto

La estructura de los tres primeros métodos es algo como lo siguiente.

```
Public Function guardarPrueba($obj){
```

```
Return this->myConect->executeQueryOnly(Insert into ...);  
}
```

Gracias a la clase DbOperator se crea el objeto myConect, que sigue éste nombre estándar en todas las clases, en el método únicamente se encarga de confeccionar un query, utilizando los datos del objeto que llega como parámetro, en este caso el método executeQueryOnly, se encarga de realizar la conexión y de cerrarla también.

La estructura del método de obtener un registro es algo parecido a lo que se muestra en el siguiente código:

```
Public function retornarPruebaObj($id){  
$sql="select * ... where id=$id";  
$obj= new EsteObjeto();  
  
While ($consult=mysql_fetch_array($this->myConect->consult($sql))){  
$obj->setNomber($consult['nombre']);  
}  
Return $obj;  
}
```

En el ejemplo anterior se realiza una consulta a la base de datos, se recorre el array de resultado, y se asignan los datos al objeto creado, y se devuelve el objeto.

El ejemplo anterior es el caso más sencillo, pero hay casos mucho más complejos como por ejemplos el método de guardarVenta, de la clase venta, pero éstos métodos se encuentran documentados directamente en el código fuente, así que si desea conocer más a profundidad el código, se recomienda consultar directamente los archivos del código fuente.

ANEXO 2: MANUAL DE USUARIO SIMPLE DATA

Simple Data es una aplicación web gráfica, la cual permite gestionar de una manera fácil y ágil las ventas de un establecimiento.

A continuación se explican paso a paso cada una de sus funcionalidades.

1. Proceso Común (CRUD)¹¹

Menú Principal

- Clientes
- Productos
- Transacciones
- Consultas
- Administración

SIMPLE DATA 2 Logout

Bienvenido: Usuario - Perfil: Administrador

Proceso Común

Dato 1 Dato 2 Dato 3

Dato 4 Seleccione ▼

Guardar

Filtro 1 Filtro 2 Filtro 4 Seleccione ▼

Buscar

| | | | | | | | Modificar | Eliminar |
|--|--|--|--|--|--|--|-----------------------|--------------------------|
| | | | | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | | | | <input type="radio"/> | <input type="checkbox"/> |
| | | | | | | | <input type="radio"/> | <input type="checkbox"/> |

Eliminar

<< < Pág 1 > >>

Figura 1. Prototipo de Proceso Común

El proceso común es el proceso que se debe realizar para cada una de las siguientes operaciones:

¹¹ En computación CRUD es el acrónimo de Crear, Obtener, Actualizar y Borrar (del original en inglés: Create, Read, Update and Delete). Es usado para referirse a las funciones básicas en bases de datos o la capa de persistencia en un sistema de software. (Wikipedia, 2011)

1. Crear registros
2. Modificar registros
3. Eliminar o desactivar registros
4. Consultar registros.

A continuación se explica detalladamente la forma de realizar las acciones antes mencionadas utilizando como base el prototipo mostrado en la figura 1.

- Para crear un registro únicamente debe diligenciar los datos solicitados en el formulario de la parte superior (Figura 1 – Item 5) y luego hacer clic en el botón *guardar* (Figura 1 – Item 6).
- Para Modificar un registro ya existente se debe realizar una búsqueda del registro por medio de los filtros (Figura 1 – Item 7), y se da clic en el botón buscar (Figura 1 – Item 8), se mostrarán los datos obtenidos en la búsqueda en la tabla (Figura 1 – Item 9), luego de haber ubicado el registro que se desea modificar se debe hacer clic en la penúltima casilla de la tabla (Figura 1– Item 10), y aparecerán los datos en el formulario ubicado en la parte superior (Figura 1– Item 5), allí se podrán hacer las modificaciones, y luego hacer clic en el botón guardar (Figura 1 Item 6).
- Para eliminar uno o más registros, se realiza la búsqueda por medio de los filtros (Figura 1 – Item 7), y se hace clic en el botón buscar (Figura 1– Item 8), para encontrar los registros que se desean eliminar, se seleccionan los deseados en la última casilla de cada registro de la tabla que se desee eliminar (Figura 1– Item 11), y por último se hace clic en el botón eliminar ubicado en la parte inferior del formulario (Figura 1 – Item 12).

Notas:

Cuando hay tablas con más de 10 registros, se realiza paginación de los datos, y por tanto es necesario utilizar el control de paginación (Figura 1 – Item 3), que se encuentra en la parte inferior.

2. Login

Inicialmente el acceso al sistema se realiza por medio de una pantalla donde se debe diligenciar los campos nombre de usuario y contraseña.

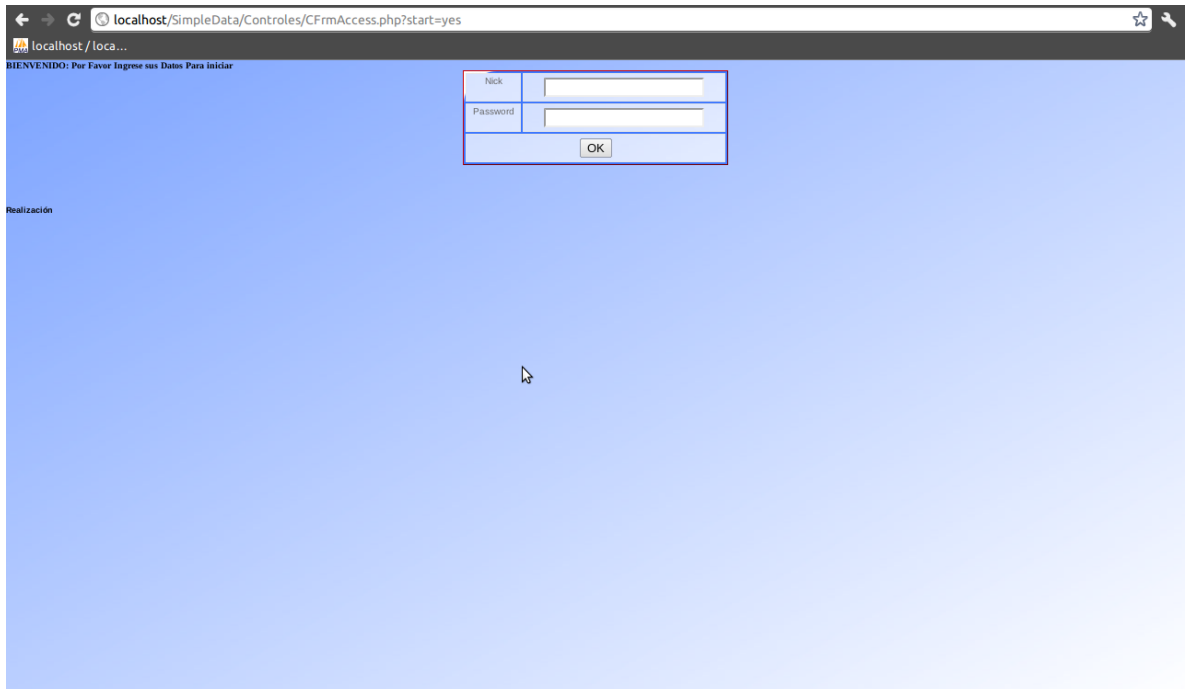


Figura 2. Ingreso al sistema

Luego de ingresar al sistema se muestra una interfaz con un menú vertical ubicado en la parte izquierda (Figura 1 – Item 1), el cual siempre será visible. En la parte superior derecha de la ventana el usuario podrá encontrar un link llamado *salir* para salir del sistema (Figura 1 – Item 2). La parte derecha de la pantalla luego del menú, se definirá como el espacio de trabajo del usuario donde podrá realizar las tareas asociadas a la aplicación (Figura 1 – Item 4).

3. Clientes

Su Logo
AQUI

Bienvenido: ADMINEST - Perfil: Administrador [Salir](#)

INGRESAR CLIENTE

Diligencie los datos requeridos: (Los campos marcados con * son obligatorios)

| | | | |
|-----------|----------------------|--------------------|----------------------|
| Nombre(*) | <input type="text"/> | Apellido(*) | <input type="text"/> |
| Cedula | <input type="text"/> | Teléfono | <input type="text"/> |
| Dirección | <input type="text"/> | Tipo de Crédito(*) | Seleccione ▼ |
| | | Tipo Cliente(*) | Seleccione ▼ |

[Crear Cliente](#) [Cancelar](#)

LISTA DE CLIENTES

Filtros

| | | | | | | | |
|----------------------|----------------------|----------------------|----------------------|--------------|--------------|-------------------|---------------------------------------|
| Nombre o Apellido | Telefono | Direccion | Cedula | Tipo Credito | Tipo Cliente | Activos/Inactivos | Buscar |
| <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> | Seleccione ▼ | Seleccione ▼ | Activos ▼ | <input type="button" value="Buscar"/> |

| # | Nombre | Apellido | Cedula | Telefono | Direccion | Tipo de Credito | Tipo de Cliente | Modificar | Activar/Desactivar |
|---|----------|----------|------------|------------|-----------------------|-----------------|-----------------|-----------------------|--------------------------|
| 1 | Eduin | Montaño | 80253322 | 3114789808 | Kra_2_A_No_136_17_Sur | Normal | General | <input type="radio"/> | <input type="checkbox"/> |
| 2 | Fabian | Novoa | 1022961788 | 3134681601 | Kra_2_A_N_136_17_Sur | Normal | General | <input type="radio"/> | <input type="checkbox"/> |
| 3 | Enrique | Novoa | 2598456 | 7708792 | Kra_2_A_N_136_17_Sur | SinCredito | General | <input type="radio"/> | <input type="checkbox"/> |
| 4 | Ingrid | Alvarez | 1031128048 | 5633404 | Calle_50_N_31-00 | Max | General | <input type="radio"/> | <input type="checkbox"/> |
| 5 | Edison | Cerquera | 1043243432 | 4343534 | Kra_2_A_N_136_17_Sur | Normal | General | <input type="radio"/> | <input type="checkbox"/> |
| 6 | Juliana | Rojas | 123456789 | 12345667 | kra_4_fff | Alto | General | <input type="radio"/> | <input type="checkbox"/> |
| 7 | Santiago | Salazar | 534 | 65654656 | | Alto | General | <input type="radio"/> | <input type="checkbox"/> |

Figura 3. Clientes

En esta pestaña el cliente puede encontrar la opción de administración de clientes:

En administración del clientes se pueden realizar las tareas mencionadas en el *Proceso Común*: Crear, Modificar y Eliminar Clientes

Notas:

Por defecto se tiene cargado un cliente con nombre *Genérico*, en caso de que hagan ventas, donde no se hace necesario hacer registro del cliente, pero sí de la transacción, se le cargan a este cliente. Al cliente genérico sólo se le pueden realizar ventas de contado.

4. Productos

Bienvenido: ADMINEST - Perfil: Administrador [Salir](#)

LISTA DE PRODUCTOS

Filtros

Nombre: Categoría: Activos/Inactivos:

| # | Nombre | Categoría | Precio 1 | Precio 2 | Precio Compra | Cantidad Disponible | Observaciones | Modificar | Desactivar |
|----|---------------------|-----------|----------|----------|---------------|---------------------|-----------------------------------|-----------------------|--------------------------|
| 1 | ImpresionFull | Partes | 900 | 800 | 100 | 5 | observacion1 | <input type="radio"/> | <input type="checkbox"/> |
| 2 | MinutoCelular | Insumos | 150 | 0 | 0 | 0 | bien | <input type="radio"/> | <input type="checkbox"/> |
| 3 | MinutoLocal | Partes | 100 | 89 | 10 | 999 | Minuto se saca del telefono negro | <input type="radio"/> | <input type="checkbox"/> |
| 4 | ImpresionFotoGrande | Partes | 3000 | 0 | 0 | 0 | dasdasda | <input type="radio"/> | <input type="checkbox"/> |
| 5 | ArregloPC | Partes | 35000 | 0 | 0 | 0 | renewed43 | <input type="radio"/> | <input type="checkbox"/> |
| 6 | CarpetaPresentación | Partes | 300 | 0 | 0 | 0 | dda | <input type="radio"/> | <input type="checkbox"/> |
| 7 | SobreManilaCarta | Partes | 250 | 0 | 0 | 0 | dasdsades | <input type="radio"/> | <input type="checkbox"/> |
| 8 | SobreManilaOficio | Partes | 300 | 0 | 0 | 0 | dasdasda | <input type="radio"/> | <input type="checkbox"/> |
| 9 | Digitación | Partes | 1000 | 0 | 0 | 0 | dasdsadas | <input type="radio"/> | <input type="checkbox"/> |
| 10 | Fotocopia | Partes | 100 | 0 | 0 | 0 | dasdasda | <input type="radio"/> | <input type="checkbox"/> |

Paginación:

Figura 4. Productos

Cuando el usuario haga clic sobre la opción productos podrá realizar las operaciones del *Proceso Común*: Crear, Modificar y Eliminar productos.

- Exportar Inventario:
Para exportar el inventario únicamente debe hacer clic sobre el botón *Descargar Estructura*, el sistema le arrojará un archivo en Excel, con los datos de los productos, puede obtener distintos archivos, utilizando los filtros de la parte superior,
- Cargar Inventario: Para cargar un inventario se debe hacer clic en el botón *Crear Producto* inmediatamente se mostrarán dos tablas una para crear un producto como se define en el proceso común y la otra le permitirá cargar un archivo, éste archivo debe tener exactamente la misma estructura que el archivo que se obtiene con el botón *Descargar Estructura*, usted puede diligenciar este archivo con nuevos productos, o modificar los que había descargado, y hacer clic en el botón *Cargar*. Es importante utilizar la estructura de un archivo antes exportado, de lo contrario se corre el riesgo de arruinar la información actual del inventario, si en el archivo CVS, se encuentra un código de producto ya existente, este se actualizará, si tiene un código no existente, éste se ignorará y el sistema autogenerará uno

nuevo creando un nuevo registro, si no tiene código, el sistema creará un nuevo registro y autogenerará un nuevo código.

5. Transacciones

a. Venta

Una venta tiene tres pasos el primero de ellos es seleccionar un cliente y una forma de pago, allí se validará que el cliente no esté en la lista negra, si es una venta a crédito

Su Logo
AQUI

Bienvenido: ADMINEST - Perfil: Administrador [Salir](#)

TRANSACCION

Por favor siga las instrucciones para realizar la transacción:

Seleccionar Cliente: Seleccionar Tipo de Venta:

Figura 5: Venta Paso 1

Luego de esto se debe agregar los productos a la venta, allí se validará que la cantidad de producto solicitada no supere la cantidad en el inventario, que no se supere el tope de crédito en caso de que la venta sea a crédito. (Figura 6)

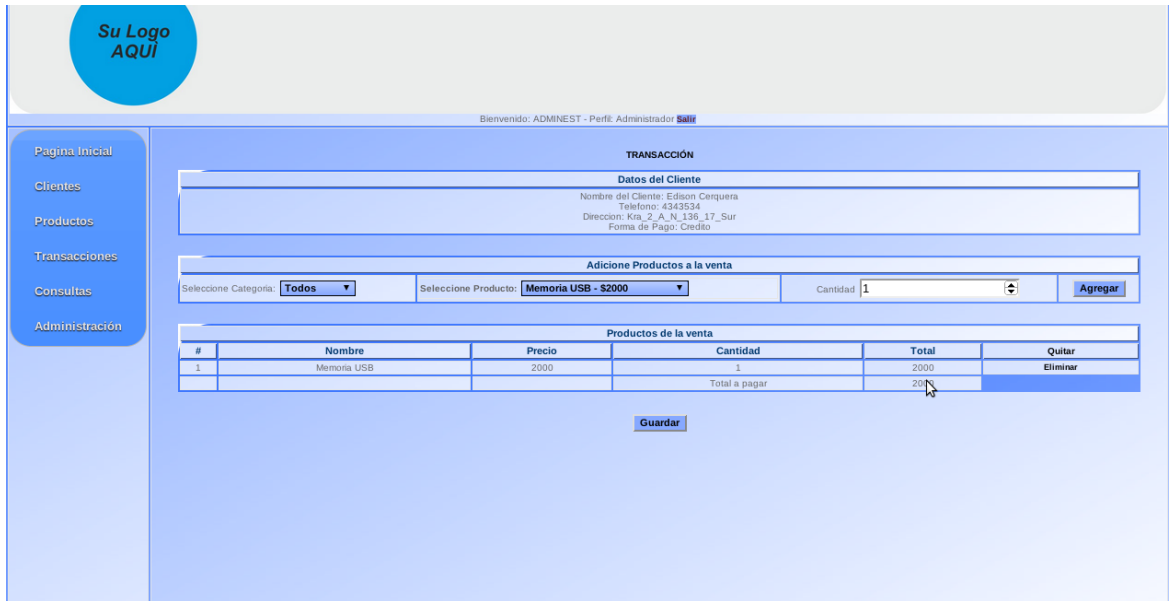


Figura 6. Venta Paso 2

Cuando la venta se haya confirmado con el botón guardar, éste botón se deshabilita, se deshabilita la opción eliminar de la columna quitar, y se habilitan dos botones Exportar y Vista de Impresión, (Figura 7)

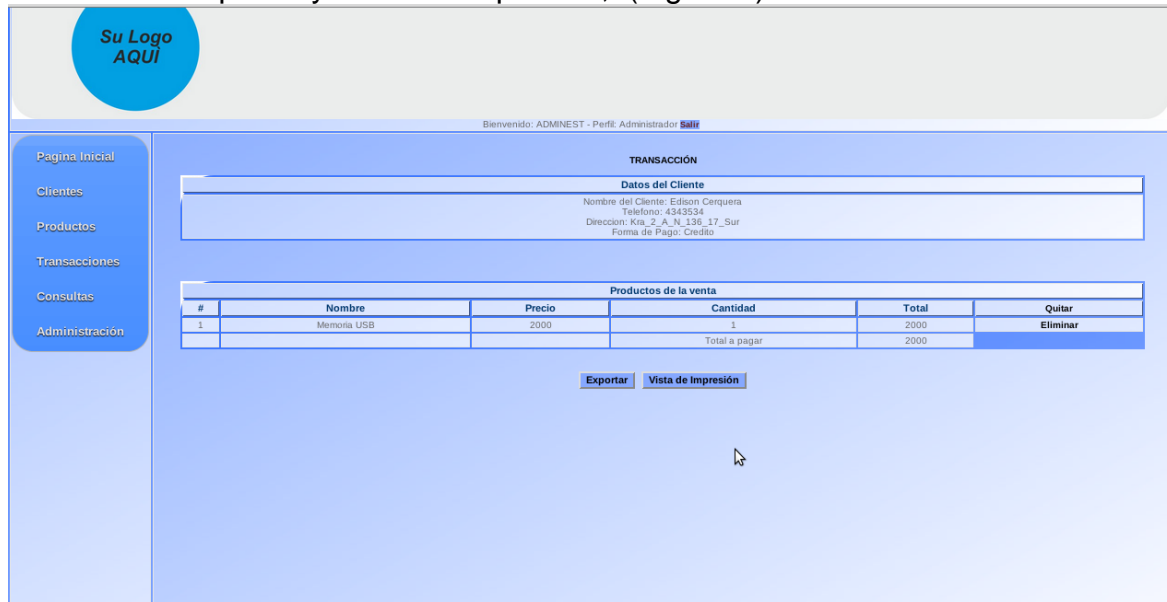


Figura 7. Venta Paso 3.

A medida que se ingresan o se eliminan productos el campo *Total A pagar* se actualiza de manera automática. (Figura 7)

Una vez se haya finalizado de adicionar los productos se procede a hacer clic en el botón *Guardar*, esto finaliza la venta, y hace inmodificable la factura. Una vez se hace clic en el botón *Guardar*, se habilitan los botones *Exportar*, y *Vista Previa de Impresión*, los cuales mostrarán un imprimible del *Recibo de Venta*, ya sea en formato .DOC, o en una salida por WEB. El formato generado se muestra a continuación (Figura 7)

| Señor (es): Edilson Cerquera | | Fecha: 2012-01-10 23:47:37 | | |
|---|----------|----------------------------|--------------------------|-------------|
| C.C.: 1043243432 | | Telefono: 4343534 | | |
| Dirección: Kra_2_A_N_136_17_Sur | | Forma de Pago: Credito | Tipo de Cliente: General | |
| Código del Producto | Cantidad | Descripción del Producto | Valor Unitario | Valor Total |
| 18 | 1 | Memoria USB | 2000 | 2000 |
| | | | Total | 2000 |
| Nota: Si este recibo no tiene la firma y el sello de aceptación, no tiene ninguna validez | | Aceptada (Firma y sello) | | |
| | | Vendedor: adminest | | |

Figura 8. Estructura Recibo de Venta

b. Abono

Su Logo
AQUI

Bienvenido: ADMINEST - Perfil: Administrador **Salir**

ABONOS

Ingresar abono

Seleccione el cliente que realiza el abono **Seleccione** Digite el valor del abono Observaciones

Ingresar Abono

Figura 9. Abono

Al hacer Clic en la opción abonos del menú Transacciones, se muestra un formulario como el de la Figura 9, allí únicamente se debe seleccionar el cliente que realiza el abono, registrar un valor real de abono, y hacer alguna observación, el valor debe ser un número positivo, y no debe superar el valor total de la deuda que tiene el cliente.

Una vez se diligencien los datos, se debe hacer clic en el botón *Guardar*, luego de esto se habilitarán los botones *Exportar A Word*, y *Vista Previa de Impresión*, los cuales mostrarán un imprimible del *Recibo de Abono*, ya sea en formato .DOC, o en una salida por WEB, con una estructura similar al recibo de venta (Figura 8)

6. Consultas

BIENVENIDO: ADMINEST - Perfil: Administrador [Salir](#)

LISTA DE TRANSACCIONES

Filtros

Nombre Cliente: Tipo Cliente: Tipo Venta: Vendedor: Tipo Transaccion: Fecha Ini: Fin:

| # | No Recibo | Cliente | Fecha | Vendedor | Tipo Venta | Tipo Cliente | Valor | Ver Recibo |
|----|-----------|------------------|---------------------|----------|---------------|--------------|-------|--|
| 1 | 4 | Ingrid Alvarez | 2011-01-29 00:00:00 | fabian | VENTA/Contado | General | 0 | <input checked="" type="checkbox"/> Ver <input type="button" value="Descargar"/> |
| 2 | 5 | Eduin Montaño | 2011-02-04 00:00:00 | fabian | VENTA/Contado | General | 0 | <input checked="" type="checkbox"/> Ver <input type="button" value="Descargar"/> |
| 3 | 6 | Santiago Salazar | 2011-05-25 00:00:00 | adminest | VENTA/Credito | General | 0 | <input checked="" type="checkbox"/> Ver <input type="button" value="Descargar"/> |
| 4 | 7 | Santiago Salazar | 2011-05-25 00:00:00 | adminest | VENTA/Contado | General | 0 | <input checked="" type="checkbox"/> Ver <input type="button" value="Descargar"/> |
| 5 | 8 | Edison Cerquera | 2011-05-28 00:00:00 | adminest | VENTA/Contado | General | 0 | <input checked="" type="checkbox"/> Ver <input type="button" value="Descargar"/> |
| 6 | 9 | Santiago Salazar | 2011-06-04 00:00:00 | adminest | VENTA/Contado | General | 0 | <input checked="" type="checkbox"/> Ver <input type="button" value="Descargar"/> |
| 7 | 54 | Francy Novoa | 2012-01-01 19:23:07 | adminest | VENTA/Credito | General | 31128 | <input checked="" type="checkbox"/> Ver <input type="button" value="Descargar"/> |
| 8 | 55 | Francy Novoa | 2012-01-01 19:23:08 | adminest | VENTA/Credito | General | 31128 | <input checked="" type="checkbox"/> Ver <input type="button" value="Descargar"/> |
| 9 | 56 | Santiago Salazar | 2012-01-01 19:30:39 | adminest | VENTA/Credito | General | 12600 | <input checked="" type="checkbox"/> Ver <input type="button" value="Descargar"/> |
| 10 | 57 | Fabian Novoa | 2012-01-01 19:31:54 | adminest | VENTA/Contado | General | 3760 | <input checked="" type="checkbox"/> Ver <input type="button" value="Descargar"/> |

Paginación

Figura 10. Consultas

Al hacer clic en la opción *Consultas Generales*, del menú consultas, se mostrará un menú como el de la Gráfica 11, allí se podrán consultar las transacciones de acuerdo a la combinación de filtros utilizada.

Ejemplo:

Es posible realizar una consulta de las transacciones hechas a crédito de un determinado cliente, en un determinado rango de tiempo.

Notas:

EN la casilla Ver Recibo aparecen 2 links, con ellos se pueden obtener los recibos en una salida por Web en una nueva pestaña, o en un archivo .doc.

7. Administración

a. Perfiles:

CREAR PERFIL

Diligencie los datos requeridos: (Los campos marcados con * son obligatorios)

Nombre(*) Administrador

| # | Acceso | Nombre del Permiso | Crea | Modifica | Bona |
|----|-------------------------------------|-----------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 1 | <input checked="" type="checkbox"/> | administra_clientes | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 2 | <input checked="" type="checkbox"/> | administrar_productos | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 3 | <input checked="" type="checkbox"/> | ventas | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 4 | <input checked="" type="checkbox"/> | abonos | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 5 | <input checked="" type="checkbox"/> | consultas_generales | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 6 | <input checked="" type="checkbox"/> | reporte_ganancias | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 7 | <input checked="" type="checkbox"/> | estado_cliente | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 8 | <input checked="" type="checkbox"/> | perfiles | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 9 | <input checked="" type="checkbox"/> | usuarios | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 10 | <input checked="" type="checkbox"/> | lista_negra | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 11 | <input checked="" type="checkbox"/> | creditos | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 12 | <input checked="" type="checkbox"/> | politicar | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

Modificar Perfil Cancelar

LISTA DE PERFILES

Filtros

Nombre Activos/Inactivos Buscar

Figura 11. Perfiles

Al hacer clic en la opción *Perfiles* del menú *Administración*, el usuario podrá Crear, Modificar y Desactivar Perfiles. (Figura 11)

El sistema puede asumir tantos roles de usuario como sean necesarios, por tanto es el mismo usuario quien define los perfiles o roles, y es el mismo usuario quien otorga permisos sobre las funciones específicas del sistema a cada perfil, el sistema se entrega al usuario con un único usuario de nombre adminest y Perfil Administrador, el cual tiene acceso a todas las funciones del sistema.

Un perfil consta de un nombre y unos permisos sobre la aplicación. Para crear un perfil únicamente se debe hacer clic en el botón crear perfil, diligenciar el campo *Nombre Perfil*, y seleccionar los permisos que el perfil tiene sobre la aplicación.

Para desactivar uno o varios perfiles, se deben seleccionar en la ultima casilla, y hacer clic en el botón *Activar/Desactivar*.

Notas:

Si se dan permisos de Crear, Modificar y Eliminar a un perfil en la opción Administración, se están dando estos permisos sobre todas las opciones de administración.

Si se selecciona Ingreso únicamente, el usuario tendrá permisos de sólo lectura sobre la función.

b. Usuarios

| # | Nombre | Perfil | Modificar | Desactivar |
|---|---------------|---------------|-----------------------|--------------------------|
| 1 | fabian | dsdsdd | <input type="radio"/> | <input type="checkbox"/> |
| 2 | EduinMontaño | Administrador | <input type="radio"/> | <input type="checkbox"/> |
| 3 | Cosmo | Administrador | <input type="radio"/> | <input type="checkbox"/> |
| 4 | Ingrid | myPerfil | <input type="radio"/> | <input type="checkbox"/> |
| 5 | adminest | Administrador | <input type="radio"/> | <input type="checkbox"/> |
| 6 | guilvert | Administrador | <input type="radio"/> | <input type="checkbox"/> |
| 7 | Nestor_Suarez | Administrador | <input type="radio"/> | <input type="checkbox"/> |
| 8 | usemuevo | Administrador | <input type="radio"/> | <input type="checkbox"/> |
| 9 | fabi | Geedadas | <input type="radio"/> | <input type="checkbox"/> |

Figura 12. Usuarios

Si se hace clic en la opción *Usuarios* del menú *Administración*, se pueden realizar las acciones del proceso común: Crear, Modificar y Eliminar Desactivar. (Figura 12)

c. Lista Negra

BIENVENIDO: ADMINEST - Perfil: Administrador [Salir](#)

LISTA NEGRA

Agregar cliente a lista negra

Seleccione el cliente (*) Indique una observación(*)

LISTA DE CLIENTES

Filtros

| Nombre o Apellido | Telefono | Direccion | Cedula | Tipo Credito | Tipo Cliente | Activos/inactivos | Buscar |
|----------------------|----------------------|----------------------|----------------------|---|---|--------------------------------------|---------------------------------------|
| <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text" value="Seleccione"/> | <input type="text" value="Seleccione"/> | <input type="text" value="Activos"/> | <input type="button" value="Buscar"/> |

| # | Nombre | Apellido | Cedula | Telefono | Direccion | Tipo de Credito | Tipo de Cliente | Observaciones | Quitar de la lista |
|---|--------|----------|------------|------------|-----------------------|-----------------|-----------------|---------------|--------------------------|
| 1 | Eduin | Montaño | 80253322 | 3114789808 | Kra_2_A_No_136_17_Sur | Normal | General | Me debe 5000 | <input type="checkbox"/> |
| 2 | Ingrid | Alvarez | 1031128048 | 5633404 | Calle_50_N_31-80 | Max | General | mamasita | <input type="checkbox"/> |

Paginación

Figura 13. Lista Negra

Si se hace clic en la opción *Lista Negra*, del menú *Administración*, el usuario podrá ingresar clientes a la lista negra. Cuando un cliente está en una lista negra, significa que a éste cliente no se le puede realizar ningún tipo de transacción a crédito.

Para ingresar in cliente en la Lista Negra basta con seleccionarlo en el campo Cliente, y agregar algún tipo de observación, esta observación se hará visible en la tabla que se muestra en el formulario, también se hará visible en el *Estado del cliente*, ubicado en el menú de consultas, y en transacciones cuando se selecciona un cliente que está en esta lista.

Para eliminar clientes de está lista basta con seleccionar uno o varios en última celda del registro, y hacer clic en el botón *Quitar*

d. Créditos

BIENVENIDO: ADMINEST - Perfil: Administrador [Salir](#)

LISTA DE CREDITOS

Filtros
Nombre: Activos/inactivos: **Activos**

| # | Nombre | Tope | Modificar | Desactivar |
|----|------------|--------|-----------------------|--------------------------|
| 1 | Normal | 10501 | <input type="radio"/> | <input type="checkbox"/> |
| 2 | Medio | 20000 | <input type="radio"/> | <input type="checkbox"/> |
| 3 | Alto | 30000 | <input type="radio"/> | <input type="checkbox"/> |
| 4 | Max | 50000 | <input type="radio"/> | <input type="checkbox"/> |
| 5 | Super | 100000 | <input type="radio"/> | <input type="checkbox"/> |
| 6 | SinCredito | 0 | <input type="radio"/> | <input type="checkbox"/> |
| 7 | Superato | 70000 | <input type="radio"/> | <input type="checkbox"/> |
| 8 | Áspero | 999999 | <input type="radio"/> | <input type="checkbox"/> |
| 9 | sisisisi | 1 | <input type="radio"/> | <input type="checkbox"/> |
| 10 | ewewewq | 32312 | <input type="radio"/> | <input type="checkbox"/> |

Paginación

Figura 14. Créditos

Al hacer clic en la opción *Créditos* del menú *Administración*, se podrán realizar las acciones del proceso común: Crear, Modificar y Eliminar créditos.

Cuando se crea un cliente, se debe seleccionar un crédito, de los que se han creado con anterioridad en este formulario, el crédito define un nombre y un tope, este tope se ve reflejado en las transacciones, cuando se está realizando una venta a crédito, y se está intentando ingresar un producto con el cual se superaría el tope de crédito autorizado ya que el sistema no lo permite.

Notas:

El sistema tiene por defecto un Crédito llamado sin Crédito donde el tope es 0, este se selecciona para usuario que no tienen ningún crédito.

e. Políticas de Reabastecimiento

Figura 15 Políticas de Reabastecimiento

| # | Nombre | Categoria | Cantidad Minima | Observaciones | Modificar | Desactivar |
|---|------------|-----------|-----------------|----------------------------------|-----------------------|--------------------------|
| 1 | Cd | Partes | 20 | Tengo que ir al centro a comprar | <input type="radio"/> | <input type="checkbox"/> |
| 2 | MemoriaUSB | Partes | 20 | toca traer 30 | <input type="radio"/> | <input type="checkbox"/> |
| 3 | Estuche | Insumos | 1 | asds | <input type="radio"/> | <input type="checkbox"/> |

Al hacer clic en la opción *Políticas de Reabastecimiento*, el usuario podrá crear políticas de reabastecimiento para algún producto. La política de reabastecimiento indica que cuando quede cierta cantidad en inventario, esto generará una alerta, que indica la cantidad en inventario de los productos que tienen políticas, mostrando también la observación. Estas alertas serán visibles en la página principal.

ANEXO 3: MODELO ENTIDAD RELACIÓN

