

UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA

UNAN-LEÓN

FACULTAD DE CIENCIAS Y TECNOLOGÍA

DEPARTAMENTO DE COMPUTACIÓN



**MONOGRAFÍA PARA OPTAR AL TÍTULO DE:
INGENIERO EN SISTEMAS DE INFORMACIÓN**

**AUTOMATIZACIÓN DE REGISTROS E INVENTARIO DE LA LIBRERÍA GUADALUPE CORINTO
DE LA CIUDAD DE CHINANDEGA (SILIGU)**

INTEGRANTES:

- BR. CRISTHIAM OSMAR ALVARADO FIGUEROA.
- BR. CARLOS ANDRÉS CASTELLÓN VALLE.
- BR. CRISTHIAM BENITO PACHECO CENTENO.

TUTOR: MSC. ÁLVARO ALTAMIRANO

LEÓN, 07 DE MAYO DEL 2013

“A LA LIBERTAD POR LA UNIVERSIDAD”



AGRADECIMIENTO





A Dios.

Por haberme permitido llegar hasta este punto y haberme dado la salud para lograr mis objetivos, por haberme dado la vida y haber llegado esta etapa de la misma, y las fuerzas para seguir siempre adelante superando obstáculos de todo tipo, gracias por el amor, bendiciones y gracias que has derramado sobre mí y cada persona que me rodea

“El Dios en quien yo creo no nos manda el problema, sino la fuerza para sobrellevarlo”

A mi madre Paola Lorena Figueroa Pérez

Por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien, pero más que nada por su amor.

A mi padre Simón Eduardo Alvarado Quintero

Por los ejemplos de perseverancia y constancia que lo caracterizan y que me ha infundado siempre, por el valor mostrado para salir adelante y por su amor.

A mi hermano Carlos.

Por ser un ejemplo de un hermano mayor del cual aprendí aciertos y de momentos difíciles.

A nuestro tutor Msc. Álvaro Altamirano

Por su gran apoyo y motivación para la culminación de nuestros estudios profesionales y la elaboración de esta Tesis

A mis compañeros de Tesis Carlos Castellón y Cristhiam Pacheco

Sin ellos este trabajo no hubiese sido posible.

Y por último a mis amigos

Que nos apoyamos mutuamente en nuestra formación profesional y que hasta ahora seguimos siendo amigos: Milton Padilla, Oswaldo Rivas, Yader Castillo, Edwin Amaya, Ernesto Bravo, Daryl Cruz y todos los que han estado presentes en este tiempo.

“La gratitud en silencio no sirve de mucho a nadie, por ello, aunque sepas que les estoy agradecido quiero dejar constancia de ello, gracias por estar siempre a mi lado”

Cristhiam Osmar Alvarado Figueroa.





Le agradezco con mucho amor primeramente a **Dios** por darme la sabiduría, fuerza, salud y la oportunidad de haber culminado mis metas de nuestras vidas.

A mi **Madre Anselma Valle Cantillanoya** mi **padre Carlos Javier Castellón González**

Gracias por darme la vida, por su ayuda incondicional, con su apoyo moral y económico con su esmerada dedicación, confianza que me ha brindado a lo largo de mi vida y la oportunidad que nos dieron para culminar y coronar nuestra carrera, a la Señora Natalia Ramos Ríos por haberme brindado el cariño y hospedaje en su casa, a mi madrina Lidia Martínez de Blanford.

A mis hermanos por el apoyo moral en todo momento, y a personas que aunque sus nombres no estén plasmados en este momento me han brindado sus valiosos aportes.

A mis **familiares** y **amigos** por su apoyo y motivación en los momentos de dificultades.

A mis compañeros de Tesis **Cristhiam Alvarado** y **Cristhiam Pacheco**

A nuestro **tutor Msc. Álvaro Altamirano**, por brindarnos parte de sus conocimientos, tiempo, paciencia y colaboración para realizar el presente trabajo.

“Hay una fuerza motriz más poderosa que el vapor, la electricidad y la energía atómica: la voluntad.” (Albert Einstein)

Carlos Andrés Castellón Valle





A Dios,

Doy gracias a **Dios** primeramente por haberme permitido culminar este trabajo monográfico ya que si no fuera por su amor y ayuda yo no estaría hoy aquí escribiendo estas líneas de agradecimientos.

A mi Madre,

Esta tesis la he dedicado de manera muy especial, con mucho cariño y amor a mi madre **Reyna Patricia Centeno**, por la educación que me ha dado. Sin sus maternales consejos no me habría convertido en la persona que soy hoy y no habría podido completar este ciclo de mi vida. No tengo como pagarle todo lo hermosa que ha sido conmigo, siempre estaré agradecido con ella.

A mi Padre,

También dedico a mi padre **G. Salvador Q. Martínez**, Gracias a su apoyo incondicional. Sin sus paternales consejos no me habría convertido en la persona que soy hoy. Siempre estaré agradecido con él.

A mi Hermana,

Gracias a **Gabriela Patricia**, Que sin tu apoyo, tan especial nunca hubiera podido ser la persona en la que ahora me convertí, este corto tiempo que llevo de tenerte a mi lado ha sido muy especial para mí.

A mi novia,

Gracias **Cinthy Carolina Membreño** sin tu apoyo incondicional no podría ser el hombre de bien que soy ahora.

A mis compañeros de Tesis **Cristhiam Alvarado y Carlos Castellón**, sin su apoyo no hubiera finalizado esta etapa de mi vida.

Y por último a mis amigos,

Que nos apoyamos mutuamente en nuestra formación profesional y que hasta ahora seguimos siendo amigos: **Milton Padilla, Oswaldo Rivas, Dylan Otero** y todos los que han estado presentes en todo el transcurso de mi formación.

Gracias a todos por su ayuda hoy puedo decir que mis amigos Cristhiam, Carlos y Yo hemos terminado nuestra tesis.

“Para poder triunfar en la vida, se necesita el apoyo de las personas importantes, ya que sin ella no se podría superar uno como persona” (Cristhiam P. Centeno)

Cristhiam Benito Pacheco Centeno





INDICE

AGRADECIMIENTO	2
RESUMEN	9
1. INTRODUCCIÓN	10
1.1 ANTECEDENTES	11
1.2 PLANTEAMIENTO DEL PROBLEMA	12
1.3 JUSTIFICACIÓN.....	13
1.3.1 Originalidad	13
1.3.2 Alcance	13
1.3.3 Producto.....	13
1.3.4 Impacto	14
1.4 OBJETIVOS	15
Objetivo General	15
Objetivos Específicos.....	15
2. METODOLOGÍA.....	16
Ciclo de vida en espiral.....	17
3. MARCO TEÓRICO.....	19
3.1 Tecnologías Empleadas	20
3.1.1 Visual estudio 2008	20
3.1.2 .NET	21
3.1.3 Estructura en capas	23
3.1.4 LINQ (Language-Integrated Query).....	26
3.1.5 DevExpress v10.2.....	28
3.1.6 SQL Server Express 2008	35
4. ANÁLISIS DE REQUISITOS	40
Especificación de requisitos de software	41
1. Introducción.	41
1.1 Propósito:	41
1.2 Alcance:	41
1.3 Definiciones Acrónimos y Abreviaturas:	42
1.4 Visión General:	43





1.5 Referencias:	43
2. Descripción general:	43
2.1 Relaciones del Producto.	43
2.2 Funciones del Producto.	44
2.3 Características del Usuario.	44
2.4 Restricciones Generales.	44
3. Requisitos Específicos.	45
3.1 Requisitos Funcionales.	45
3.2 Requisitos del funcionamiento.	76
3.3 Restricción de diseño.	76
3.4 Atributo de seguridad	76
3.5 Otros requisitos	76
5. CASOS DE USOS DOCUMENTADOS	80
6. DIAGRAMA DE CLASES	88
7. DIAGRAMAS DE SECUENCIA	93
8. DISEÑO DE DATOS	98
9. DISEÑO ARQUITECTÓNICO	103
10. DISEÑO DE INTERFACES	104
11. REQUERIMIENTOS DEL SISTEMA	112
❖ Requerimiento Software	112
12. CONCLUSIONES	113
13. RECOMENDACIONES	114
14. BIBLIOGRAFÍA	115
15. ANEXOS	116
Glosario	117
DALUsuarios	118
DALFacturaVenta	119





INDICE DE IMÁGENES

Figura No.1 Ciclo de vida en espiral	17
Figura No.2 Arquitectura de Capas	25
Figura No.3 LinQ.....	27
Figura No.4 DevExpress.....	28
Figura No.5 Diseño XtraGrid.....	31
Figura No.6 Diseño XtraReport	32
Figura No.7 Cuadro de Herramientas de XtraChart	33
Figura No.8 Diseño LookUp Editor	33
Figura No.9 Búsqueda LookUp Editor	34
Figura No.10 Casos de Usos Generalizado	77
Figura No.11 Diagrama de Casos de Uso del Administrador	78
Figura No.12 Diagrama de Casos de Uso del Usuario Normal	79
Figura No.13 Diagrama de Clases Cliente	88
Figura No.14 Diagrama de Clases Usuario	89
Figura No.15 Diagrama de Clases FacturaCompra	90
Figura No.16 Diagrama de Clases FacturaVenta	91
Figura No.17 Diagrama de Clases General del Sistema de SILIGU	92
Figura No.18 Diagrama de Secuencia Modificar Producto	93
Figura No.19 Diagrama de Secuencia Agregar Usuario.....	94
Figura No.20 Diagrama de Secuencia Agregar FacturaCompra	94
Figura No.21 Diagrama de Secuencia Agregar FacturaVenta	95
Figura No.22 Diagrama de Secuencia Visualizar Historial.....	95
Figura No.23 Diagrama de Secuencia Eliminar Cliente	96
Figura No.24 Diagrama Entidad Relación.....	97
Figura No.25 Diagrama de Entidades.....	99
Figura No.26 Diseño Arquitectónico	103
Figura No.27 Inicio de la Aplicación	105
Figura No.28 Interfaz de Configuración del Servidor	105
Figura No.29 Interfaz de Inicio de Sesión.....	106
Figura No.30 Interfaz Principal del Programa (Menú)	106
Figura No.31 Interfaz Principal del Programa (Reportes)	107
Figura No.32 Interfaz Registrar Cliente	107
Figura No.33 Interfaz Registrar Usuario.....	108
Figura No.34 Interfaz Registrar Factura Compra	108
Figura No.35 Interfaz Actualizar Proveedor.....	109
Figura No.36 Interfaz Detalle Factura Venta.....	109
Figura No.37 Interfaz Modificar Usuario.....	110
Figura No.38 Interfaz Modificar Producto	110
Figura No.39 Interfaz Reporte Inventario	111
Figura No.40 Interfaz Consultar Inventario Producto	111





Figura No.41 Interfaz Consultar Inventario FacturaVenta 111

RESUMEN

El presente trabajo monográfico se trata sobre el desarrollo de un sistema para gestionar la información de la Librería Guadalupe Corinto con la finalidad de lograr un mejor desempeño y agilizar los procesos, garantizando un mejor manejo de la información.

Al finalizar la elaboración de este sistema puede afirmarse que se cumplieron todos los objetivos planteados, adicionalmente para el desarrollo de la aplicación se utilizaron diversas tecnologías como los controles DevExpress para .NET, el gestor de base de datos SQL Express 2008 y Visual Estudio 2008.

Dentro de esta aplicación se crearon varios módulos a los cuales se encargan de agilizar los procesos de gestión de la Librería Guadalupe entre los cuales tenemos: Clientes, Proveedores, Productos.

Este documento ayuda a formar una idea general de las diferentes etapas que se llevaron a cabo para el desarrollo de este sistema.





1. INTRODUCCIÓN





1.1 ANTECEDENTES

Actualmente la Librería Guadalupe Corinto realiza todos los procesos de manera manual usando hojas de Excel, documentos de Word o libretas, lo que provoca que el trabajo se realice de una manera muy lenta y las operaciones tarden más tiempo.

Los procesos como Registrar un cliente, proveedor, productos se en archivos de Excel. En este punto se llenan unas hojas de Excel que contienen las listas de los proveedores, clientes y productos.

La creación de facturas se hace usando libretas con la cantidad de productos, nombre de los productos y la fecha.

El control de los productos comprados a los proveedores, son guardados en hojas de Excel y en archivos de Word, haciendo esto que se dificulte mucho cuando quieran realizar un inventario de los productos comprados a los proveedores.

Ya que los procesos realizados en la Librería Guadalupe Corinto son muy importantes para el perfecto funcionamiento de la misma, se hace imprescindible la creación de un software de Automatización de Inventario que permita a los trabajadores de la Librería Guadalupe Corinto que su proceso sea lo más rápido posible.

En la actualidad existen una gran variedad de software de Automatización de Inventario que se han elaborado para muchas distribuidoras del municipio de León y que han venido a beneficiar a las distribuidores que se le han diseñados este tipo de software.





1.2 PLANTEAMIENTO DEL PROBLEMA

La Librería Guadalupe Corinto tiene problemas con sus procesos de gestión de compra y venta de productos, así como el almacenamiento de los productos ofertados, proveedores y clientes, ya que este es muy rudimentario. Algunos de estos problemas se describen a continuación:

Para almacenar los clientes, este proceso se realiza de una manera no muy segura, ya que lo hacen en hojas de Excel, y se necesita poder acceder a la información del cliente de una manera rápida y segura.

Para almacenar los proveedores, este proceso también se realiza de manera no muy segura, ya que lo hacen en hojas de Excel, y se necesita poder acceder a la información del proveedor de una manera rápida y segura, ya que su información es de vital importancia.

El proceso de vender un producto se hace en una libreta, la persona que compra el producto en la librería recibe un papel con los productos que comprará, para luego tener que ir a caja a pagar sus productos para que se los puedan entregar.

El proceso de compra de productos a los proveedores, son anotados en una libreta para esa manera tener un control de los productos que fueron comprados para la Librería, haciendo eso que se les dificulte el trabajo, ya que es tedioso realizar búsquedas en libretas físicas.

De todo lo anteriormente descrito, se desglosan una serie de preguntas generales y específicas para nuestro trabajo:

¿Podrá crearse un sistema que permita automatizar todos los procesos relacionados con la gestión de La Librería Guadalupe, de la ciudad de Chinandega?

¿Se puede agilizar el proceso de almacenamiento de los datos de clientes, proveedores y productos, para que pueda ser accedida desde el puesto de los operadores de manera más rápida?

¿Se puede mejorar la forma de generar las facturas de venta de productos?

¿Habrán una forma de realizar informes de manera rápida y precisa, así como el informe del inventario de productos?





1.3 JUSTIFICACIÓN

Debido a las dificultades con la que cuenta la Librería Guadalupe Corinto, con respecto a los procesos de la gestión de ventas de productos, y almacenamiento de datos de clientes, proveedores y productos, se decidió desarrollar un software que permita una gestión de todos los procesos de una manera ágil, eficiente y segura.

1.3.1 Originalidad

Es el primer sistema que la Librería empleará. Se han utilizado para su creación herramientas de las más actuales.

1.3.2 Alcance

En las funcionalidades que tiene nuestro sistema de Automatización e Inventario tenemos Registrar Clientes, Proveedores, Productos, Usuarios, Ventas, así como Modificar los datos anteriores con excepción de las Ventas y las Compras realizadas, también se obtendrá la información del historial de operaciones y reportes de los datos registrados, y se obtendrá un reporte de inventario de clientes, factura de ventas, factura de compras, productos, productos faltantes.

1.3.3 Producto

Dentro de las características principales del sistema tenemos que resaltar varias dentro de ellas tenemos autenticación de usuarios y sus roles, la generación de reportes de los datos registrados por el sistema. También se destaca la posibilidad de realizar respaldos de la base de datos, así siempre los datos estarán respaldados. También se podrá restaurar la base de datos antes respaldada en caso de problemas.





1.3.4 Impacto

Se obtendrán los resultados esperados para una mejor utilización de la información referente a los clientes, proveedores, productos. Esto ayudará a mejorar que el proceso de la obtención sea más rápida y confiables que evitara algunos errores que son provocados por los usuarios que interactúan con la información detallada referente al usuario final.





1.4 OBJETIVOS

Objetivo General

- ✓ Desarrollar un Sistema para la Automatización e Inventario de los procesos relacionados con la gestión de venta y compra de productos de la Librería Guadalupe Corinto de la ciudad de Chinandega.

Objetivos Específicos

- ✓ Agilizar el proceso de almacenamiento de clientes, proveedores y productos, de manera que toda su información pueda ser accedida desde el puesto de los operadores o usuarios finales, de forma rápida y eficiente.
- ✓ Optimizar el tiempo de creación del Inventario de productos, manteniendo todas las garantías que se tienen actualmente.
- ✓ Mejorar el almacenamiento y procesamiento de las ventas y compras realizadas de tal forma que se permita una sencilla generación de reportes.





2. METODOLOGÍA





Ciclo de vida en espiral

Para el desarrollo del presente proyecto se realizaron una serie de fases, teniendo como modelo a seguir el ciclo de vida en espiral. Las actividades de este modelo se conforman en una espiral, en la que cada bucle o iteración representa un conjunto de actividades. Las actividades no están fijadas a priori, sino que las siguientes se eligen en función del análisis de riesgo, comenzando por el bucle interior y se sigue con los que están más arriba.

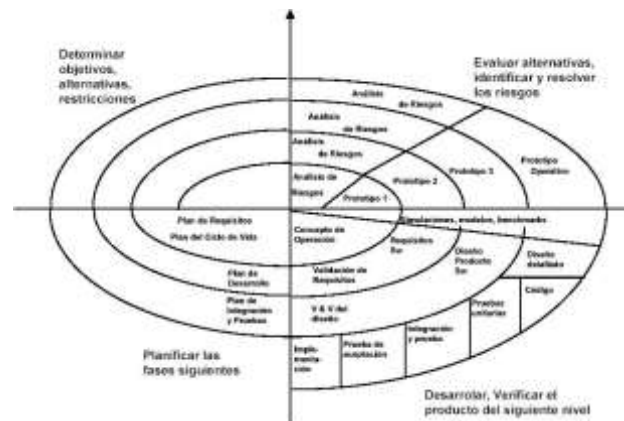


Figura No.1 Ciclo de vida en espiral

Para cada ciclo habrá cuatro actividades:

Determinar o fijar objetivos

Se realizaron visitas a la pequeña empresa donde será empleado el sistema con el objetivo de conocer su funcionamiento, forma en que tramitan, gestionan, se desempeña el personal, conflictos internos con el fin de conocer a profundidad las necesidades de la institución.

Análisis del riesgo

Uno de los principales riesgos al que nos enfrentamos es la inconsistencia del sistema debido a ese problema presentamos las siguientes alternativas:

1. El uso de contraseña para cada usuario.
2. Uso de diversas herramientas para mayor seguridad en la aplicación.
3. Realización de reportes acerca de los datos registrados.

Desarrollar, verificar y validar (probar)





Basándonos en la Información recolectada se procedió a determinar lo que el sistema requiere para su elaboración, obteniendo de esta etapa:

- Diagrama de Casos de Uso.
- Diagrama de Clases.
- Diagrama de Secuencia.
- Diagrama Entidad Relación.
- Diseño de Datos.
- Diseño de las Interfaces
- Diseño Arquitectónico.
- Diseño Procedimental.

Con la información antes recolectada, se procedió al desarrollo de esta parte del software. Luego se procedió a probar el sistema antes de implementarlo con el objetivo de comprobar que el resultado fue el esperado. La fase se culmina solo si las anteriores finalizaron con éxito entonces se procederá a la puesta en ejecución del sistema final.

Planificar

Revisar todo lo que sea hecho, evaluándolo y con ello decidir si se continúa con las fases siguientes y planificar la próxima actividad.





3. MARCO TEÓRICO





3.1 Tecnologías Empleadas

3.1.1 Visual estudio 2008

3.1.1.1 Qué es Visual Estudio 2008

Visual estudio es un entorno de desarrollo integrado (**IDE**), visual estudio permite crear aplicaciones en varios lenguajes de programación y estos tipos de aplicaciones diseñadas con este entorno se pueden intercomunicar con estaciones de trabajo, además que permite mejorar la interoperabilidad entre código nativo y código manejado por .NET. Esta integración más profunda simplifica el trabajo del diseño y la codificación.

Es un completo conjunto de herramientas para la creación tanto de aplicaciones de escritorio como de aplicaciones web empresariales para trabajo en equipo. Aparte de generar aplicaciones de escritorio de alto rendimiento, se pueden utilizar las eficaces herramientas de desarrollo basado en componentes y otras tecnologías de Visual Studio para simplificar el diseño, desarrollo e implementación en equipo de soluciones empresariales. (Microsoft.com, s.f.)

3.1.1.2 Características de Visual Estudio 2008

- Permite la creación de aplicaciones en una variedad de plataformas
- Provee de un nuevo lenguaje de consultas integrado para el manejo de los datos, es el Microsoft LanguageIntegratedQuery (LINQ), este nuevo tipo de lenguaje hace fácil la manipulación de los datos y hace más rápido el proceso de los datos ya que no se va a la base de datos a traer la información sino que desde donde la tenemos ya como es el caso de una lista donde estén los datos.
- Provee a desarrolladores la habilidad de poder escoger entre múltiples versiones del Framework con el mismo entorno de desarrollo, así podemos desarrollar en la versión que queramos ya sea en .NET Framework 2.0, 3.0 o 3.5, y con esto podremos realizar proyectos diferentes en el mismo entorno. (Microsoft T. , 2012)





3.1.2 .NET

3.1.2.1 ¿Qué es .NET?

.NET es una parte integral de muchas aplicaciones que se ejecutan en Windows y proporciona funcionalidad común para aquellas aplicaciones que se ejecutan. Esta descarga es para la gente que necesita .NET para ejecutar una aplicación en su ordenador. Para los desarrolladores, el Marco .NET proporciona un modelo de programación completo y coherente para la creación de aplicaciones que tienen experiencias de usuario visualmente impresionantes y una comunicación segura y sin problemas. (<http://www.microsoft.com/net>, 2012)

Es un framework de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones. Basado en ella, la empresa intenta desarrollar una estrategia horizontal que integre todos sus productos, desde el sistema operativo hasta las herramientas de mercado.

.NET podría considerarse una respuesta de Microsoft al creciente mercado de los negocios en entornos Web, como competencia a la plataforma Java de Oracle Corporación y a los diversos framework de desarrollo web basados en PHP. Su propuesta es ofrecer una manera rápida y económica, a la vez que segura y robusta, de desarrollar aplicaciones –o como la misma plataforma las denomina, soluciones– permitiendo una integración más rápida y ágil entre empresas y un acceso más simple y universal a todo tipo de información desde cualquier tipo de dispositivo.

(http://es.wikipedia.org/wiki/Microsoft_.NET, 2012)

El Framework .NET es un componente integral de Windows que admite la creación y ejecución de la próxima generación de aplicaciones y servicios Web XML. .NET Framework está diseñado para cumplir los siguientes objetivos:

- Para proporcionar un consistente entorno orientado a objetos de programación si el código objeto se almacena y se ejecuta a nivel local, ejecutado a nivel local pero distribuida en Internet o ejecutar de forma remota.
- Proporcionar un entorno de ejecución de código que reduzca al mínimo la implementación de software y los conflictos de versiones.
- Proporcionar un entorno de ejecución de código que promueve la ejecución segura del código, incluyendo el código creado por un tercero desconocido o semi-confianza.
- Proporcionar un entorno de ejecución de código que elimine los problemas de rendimiento de los entornos de secuencias de comandos o interpretado.





- Para hacer que la experiencia del desarrollador consistente a través de muy diversos tipos de aplicaciones, como aplicaciones basadas en Windows y aplicaciones basadas en Web.
- Para desarrollar toda la comunicación en estándares de la industria para asegurar que el código basado en el Marco. NET puede integrarse con cualquier otro código.

3.1.2.2 Las características de CommonLanguageRuntime

El CommonLanguageRuntime administra la ejecución de la memoria, el hilo, la ejecución de código, verificación de la seguridad del código, compilación, y otros servicios del sistema. Estas características son intrínsecas al código administrado que se ejecuta en el tiempo de ejecución de lenguaje común.

Con respecto a la seguridad, los componentes administrados se otorgan diversos grados de confianza, dependiendo de un número de factores que incluyen su origen (tal como la Internet, red de la empresa, o equipo local). Esto significa que un componente administrado podría o no podría ser capaz de realizar las operaciones de acceso a archivos, operaciones de acceso de registro-, u otras funciones sensibles, incluso si se está utilizando en la aplicación activa mismo.

El tiempo de ejecución aplica la seguridad de acceso a código. Por ejemplo, los usuarios pueden confiar en que un archivo ejecutable incrustado en una página Web puede reproducir una animación en la pantalla o cantar una canción, pero no pueden acceder a sus datos personales, sistema de archivos, o la red. Las características de seguridad del tiempo de ejecución por lo tanto permitir que el software legítimo Internet desplegada a ser excepcionalmente rico en características.

3.1.2.3 .NET Framework

La biblioteca de Framework. NET es una colección de tipos reutilizables que se integran estrechamente con CommonLanguageRuntime. La biblioteca de clases está orientada a objetos, proporcionando los tipos de los que su propio código administrado puede derivar funcionalidad. Esto no sólo hace que los tipos de Marco. NET fácil de usar, sino que también reduce el tiempo asociado con el aprendizaje de las nuevas características del Marco. NET. Además, los componentes de terceros se pueden integrar a la perfección con las clases del Framework. NET.

Por ejemplo, las clases de Marco. NET de recogida de poner en práctica un conjunto de interfaces que se pueden utilizar para desarrollar sus propias clases de colección. Sus clases de colección se combinan a la perfección con las clases del Framework. NET.



Como era de esperar de una biblioteca de clases orientada a objetos, los tipos de .NET Framework permiten realizar una serie de tareas de programación comunes, incluyendo tareas como la gestión de cadena, recolección de datos, conectividad de base de datos y acceso a archivos. Además de estas tareas comunes, la biblioteca de clases incluye tipos que admiten una gran variedad de escenarios de desarrollo especializados. Por ejemplo, puede utilizar el .NET Framework para desarrollar los siguientes tipos de aplicaciones y servicios.

- Las aplicaciones de consola.
- Applications GUI de Windows (Windows Forms).
- Windows PresentationFoundation (WPF).
- Aplicaciones ASP.NET.
- Servicios Web. Los servicios de Windows.

(Microsoft,msdn, 2012)

3.1.3 Estructura en capas

El tener una arquitectura n-capas bien definida reducirá los hoyos de vulnerabilidad que evidentemente se dan cuando no desarrollamos con esta arquitectura. Dentro de esta arquitectura la capa intermedia o la capa lógica de negocios ocupa un lugar angular en la construcción de una infraestructura de software que nos permitirá el crecimiento y la extensibilidad de servicios para todas las aplicaciones existentes y futuras.

La definición de los linderos de cada capa nos permitirá definir correctamente la colaboración que proveerá cada una de ellas y descubriremos que la capa intermedia es inevitablemente la lógica de negocios, esto hará una infraestructura robusta y lista para la extensibilidad y el crecimiento como proveedora de servicios. Veámoslo más a detalle:

3.1.3.1 Capa de acceso a datos (DAL)

Para el acceso a datos, comúnmente en las aplicaciones se crea una clase que instancia una conexión a la base de datos y otras clases en las cuales se generan manualmente las sentencias SQL. Luego se realizan procedimientos para pasar los parámetros respectivos, abrir la conexión a la base de datos, ejecutar la sentencia, capturar los datos si es que aplica y luego volver a cerrar la conexión.





3.1.3.2 La capa lógica de negocios (BLL)

La capa lógica de negocios (BLL) es donde se comprueban las reglas de negocio y donde se manipulan los datos devueltos por la capa de acceso a datos antes de enviarlos a la capa de objeto de negocio (BO).

Los objetos de negocio (BO) y las clases encargadas de trabajar con dichos objetos (BLL) se encargan de la obtención de datos para después ser manipulados por esta capa. Cada clase BLL contiene los mismos métodos que su clase análoga en la capa inferior (clase DAL). Estos métodos realizarán funciones de validación y de filtrado de datos antes de llamar a sus métodos parejos en la capa de acceso a datos.

3.1.3.3 Capa de objeto de negocio (BO)

En esta capa se maneja los objetos de las clases creadas por cada una de las tablas que se van a ocupar que estas sean ocupas para obtener los datos temporalmente y además de esta se crea otra capa más dentro de esta

3.1.3.4 Capa Lista (List)

En esta se crean las listas dependiendo de cada una de las clase BO que se hayan creado estas lista son encargadas de realizar la obtención de todos los datos solicitadas a través de las capas Lógica de negocio (BLL) y la de acceso a datos (DAL).

Estas dos últimas capas son utilizadas para la obtención y manipulación de los datos ellas se encargan de tener los datos para luego ser utilizados.





3.1.3.5 Diseño de la Arquitectura en capas



Figura No2 Arquitectura de Capas

Utilizar la arquitectura en capas no necesariamente se basa en utilizar estrictamente 3 capas sino que se utilizan más en el caso de uno de nuestros objetos de negocio utilizamos capas multinivel ya que dentro de la capa acceso a datos(DAL) se hacen llamados a otras capas.



3.1.3.6 *Justificación del uso de la estructura capas*

Existe una gran cantidad de ventajas que se obtienen de usar la arquitectura en capas para los desarrollos, pero las que justifican el uso de esta tecnología para el proyecto actual son:

Compatibilidad con herramientas: este tipo de programación simplifica muchas operaciones por que hace que las herramientas de visual estudio sean compatibles con este tipo de programación.

Eficacia y flexibilidad: la estructura en capas es muy eficiente ya que reduce líneas de código y maximiza la interacción con los datos además de ser muy flexibles con los datos y su utilización.

Simplicidad: la programación en capas simplifica la utilización de código muy redundante y minimiza el código que se utiliza para el acceso y obtención de datos.

Seguridad: este modelo puedes integran módulos en los que se crea las autenticaciones de usuarios para obtener una mayor seguridad con el acceso a la información y manipulación de dicha información.

Menos líneas de código: se ahorran líneas de código ya que se programa solo las clases que se quieren manipular con el acceso a los datos y se hace una petición de datos.

(wikipedia, 2012)

3.1.4 LINQ (Language-Integrated Query)

Language-IntegratedQuery (LINQ) define un conjunto de operadores de consulta estándar generales que se pueden usar en lenguajes de programación de .NET Framework 3.0. Estos operadores de consulta estándar permiten proyectar, filtrar y recorrer recopilaciones en memoria o tablas de una base de datos. Tenga en cuenta que las consultas de LINQ se expresan en el lenguaje de programación mismo y no como literales de cadena incrustados en el código de la aplicación. Se trata de un cambio significativo respecto a la forma en que la mayoría de aplicaciones se han escrito en versiones anteriores de .NET Framework. Escribir consultas desde el lenguaje de programación ofrece varias ventajas fundamentales. Simplifica las consultas eliminando la necesidad de usar un lenguaje de consulta independiente. Y si utiliza el IDE de Visual Studio 2008, LINQ también permite aprovechar la comprobación en tiempo de compilación, los tipos estáticos e IntelliSense.





LINQ se integra en varios aspectos del acceso a datos de .NET Framework, incluyendo el modelo de programación desconectada del DataSet y los esquemas de base de datos de SQL Server existentes. En esta sección se describe LINQ to ADO.NET, la implementación ADO.NET de LINQ.

El siguiente diagrama proporciona una visión general de cómo se relaciona LINQ to ADO.NET con lenguajes de programación de alto nivel, diferentes de las tecnologías LINQ y orígenes de datos compatibles con LINQ.

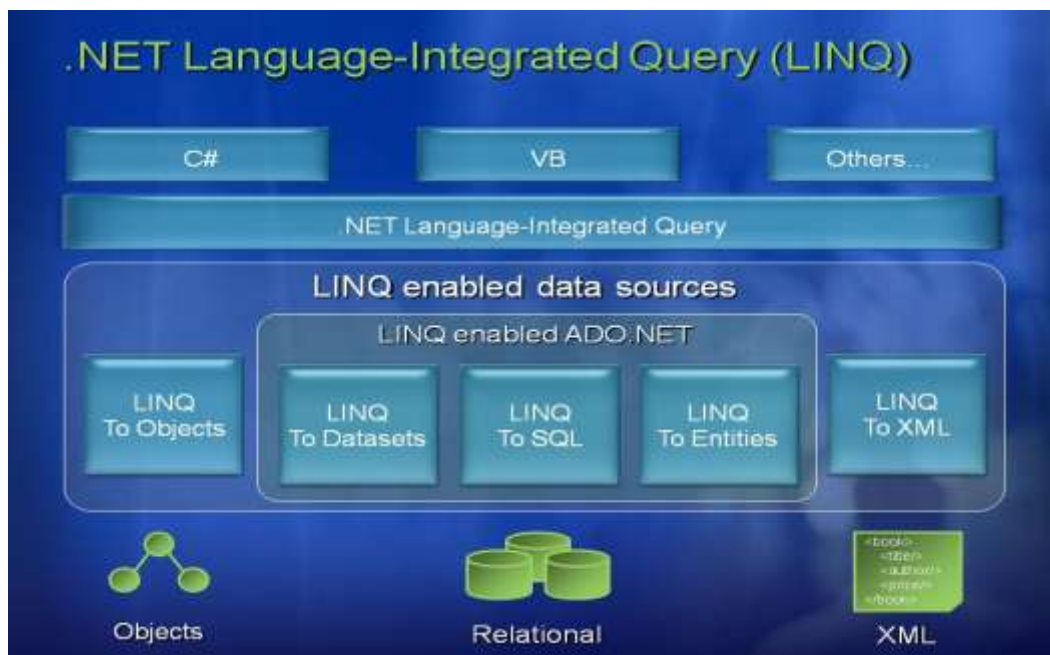


Figura No.3 LinQ

3.1.4.1 Ventajas de utilizar LINQ

- Sintaxis familiar para escribir consultas.
- Comprobación en tiempo de compilación de errores de sintaxis y seguridad de tipos.
- Compatibilidad mejorada con el depurador.
- Compatibilidad con IntelliSense.
- Capacidad para trabajar directamente con elementos XML en lugar de crear un documento XML contenedor, que es lo que se requiere en W3C DOM.
- Modificación de documentos XML en memoria de gran eficacia, aún más fácil de usar que XPath o XQuery.
- Funciones de filtrado, ordenación y agrupación eficaces.



- Modelo coherente para trabajar con datos en varios tipos de formatos y orígenes de datos. (Microsoft)

3.1.5 DevExpress v10.2

DevExpress controles de Windows Forms comparten muchas características comunes. Ellos utilizan los mismos métodos para controlar la apariencia configuraciones, datos, enfoques vinculantes mismos valores de formato y localización de recursos, etc.



Figura No.4 DevExpress

3.1.5.1 XtraChart

XtraCharts suites un conjunto de controles de gráficos profesionales para .NET (para las formas tanto en plataformas Windows como ASP.NET, así como para XtraReports).

Meticulosamente diseñado con una impresionante variedad de tipos de gráficos diferentes, un conjunto de indicadores específicos listo visuales, monitoreo en tiempo real, análisis comparativo / contrastivo, mientras que también es capaz de gráficos financieros, XtraCharts es la solución a todas sus necesidades de creación de gráficos.

Independientemente de su tipo de origen de datos (ya sea una base de datos, colección o red de pivote) o la cantidad y el tipo de datos de entrada, XtraCharts es la mejor herramienta para la visualización de todos los datos con capacidades de personalización casi infinitas. Con funciones inteligentes de automatización de alto nivel, XtraCharts hace de gráficos 2D y 3D más fácil que nunca, mientras que permanece flexible y eficiente mediante la manifestación de una interfaz amigable e intuitiva.



3.1.5.1.1 Características generales

Soporte para múltiples plataformas de destino : XtraCharts Suite incluye el ChartControl (para Windows Forms) y WebChartControl (para ASP.NET). Ambos ofrecen una funcionalidad similar, mientras que ofrece todas las ventajas de sus plataformas específicas.

Profundo apoyo para los productos DevExpress otros: PivotGridControl (ASPxPivotGrid) y XtraReports .

Soporte para múltiples tipos de gráficos diferentes: 2D y 3D tipos de gráficos desde básico (por ejemplo, barras y columnas , Punto y línea , Pie y redireccionamiento, o Radar y Polar) a lo específico (por ejemplo, Burbuja , Gantt o Parcela Swift que es especialmente diseñado para su procesamiento inmediato de cantidades muy grandes de puntos de datos).

3.1.5.1.2 Enlace de Datos

- XtraCharts soporta una variedad de tipos de orígenes de datos comunes, que también incluye soporte completo + ADO.
- Modo independiente - Una serie puede ser rellena con datos manualmente por especifican valores para los (X) e (Y) las coordenadas de los puntos. También es posible combinar unido y sin unir puntos de vista de la serie en un solo diagrama.
- Series Encuadernación Individual - XtraCharts puede mostrar datos de múltiples fuentes de datos, ya que cada serie se puede enlazar a su propia tabla de datos.
- Uso de las plantillas de la Serie - En este modo el ChartControl crea Visto serie automáticamente, y no hay necesidad de rellenar la colección Serie manualmente.
- Serie de datos de filtrado - Esta característica le permite aplicar criterios de filtro para los datos mostrados por series específicas, lo que puede evitar la inundación del control de gráfico con datos irrelevantes.
- Datos de la Serie Clasificación - Puede ordenar los datos dentro de una serie, ya sea por sus argumentos o valores.
- Serie de datos de resumen - Puede aplicar diferentes funciones de resumen para los datos de una serie.
- Impresión y Exportación - Si usted es dueño de la Biblioteca XtraPrinting , usted puede exportar el gráfico a: un archivo de imagen (múltiples formatos de archivo son compatibles), un archivo PDF; una imagen incrustada en una página HTML, una imagen incrustada en una hoja de cálculo de Microsoft Excel, una página impresa.





3.1.5.2 XtraGrid

Completo ADO + Soporte - XtraGrid aprovecha al máximo ADO, Usando esta arquitectura de acceso a datos y la separación de los módulos de datos internos de presentación de datos, XtraGrid no utiliza ningún tipo de buffer adicionales en cualquiera de sus modos. Incluso cuando se agrupan las columnas, XtraGrid usará memoria mínima y realizar la operación tan rápido como el origen de datos puede proporcionar los datos.

Soporte para listas de datos - XtraGrid funciona con cualquier fuente que admita el IList, IListed o interfaz IBindingList, además de todas las interfaces heredadas.

Columnas sin enlazar - Es posible combinar la envolvente y los modos de datos no consolidados. Sólo tiene que añadir columnas sin enlazar y suministra datos a través de un evento. Los usuarios finales pueden editar los valores en las columnas sin enlazar y los cambios se pueden guardar en un origen de datos personalizado.

Verdadero Maestro-Detalle Support - XtraGrid le permite establecer y presentar maestro-detalle la información ya sea en el modo dependiente o independiente. Se puede incluso utilizar el **taladro** para hacer zoom en el modo de cada nivel de detalle, permitiendo que incluso los conjuntos más complicados de las relaciones para ser manipulado fácilmente.

Del lado del servidor de Gestión de Datos - Controles GridControlGridLookUpEdit y apoyar un nuevo modo de enlace de datos, llamado Modo de servidor, que ha sido específicamente diseñado para soportar grandes volúmenes de datos (que consisten en 50.000 registros y más aún). El modo de servidor de datos en función de las cargas pequeñas porciones y sólo cuando sea necesario. Todas las operaciones de reconocimiento de datos se realizan en el servidor de base de datos que se ha optimizado para estas consultas.

Las características clave del modo de servidor de enlace de datos garantizar el acceso rápido a los datos, independientemente de los datos de operación, tanto si la red utiliza ordenar, agrupar, filtrar o funciones de resumen.

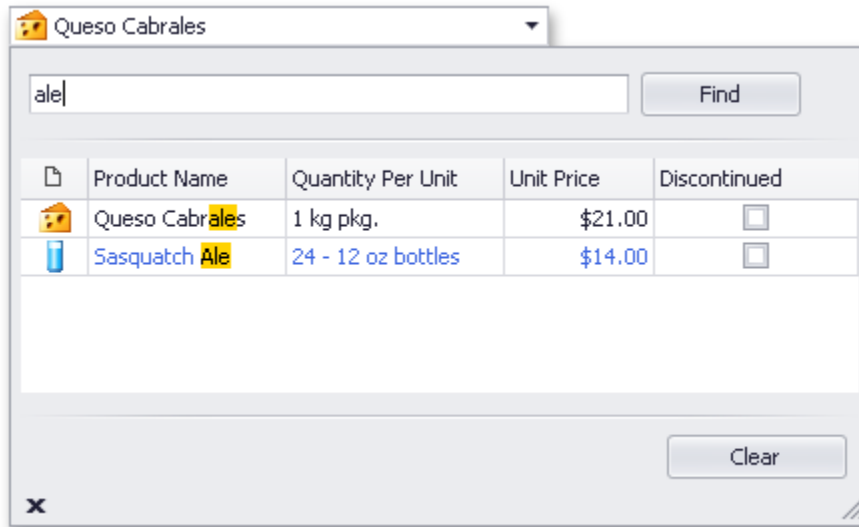


Figura No.5 Diseño XtraGrid

3.1.5.3 XtraReport

Se ha mejorado el Diseñador de informes para apoyar la interfaz de múltiples documentos (MDI). Ahora se puede cargar tantos informes como desee en la misma instancia del diseñador de informes (lo que le permite cambiar entre ellos y copiar / pegar datos y los controles a través de múltiples diseños de informe).

El Diseñador de informes para el usuario final utiliza una interfaz de barra de herramientas del menú, puede elegir entre "clásico" o MDI con pestañas. Cuando se utiliza una cinta, la única opción MDI con pestañas disponibles.

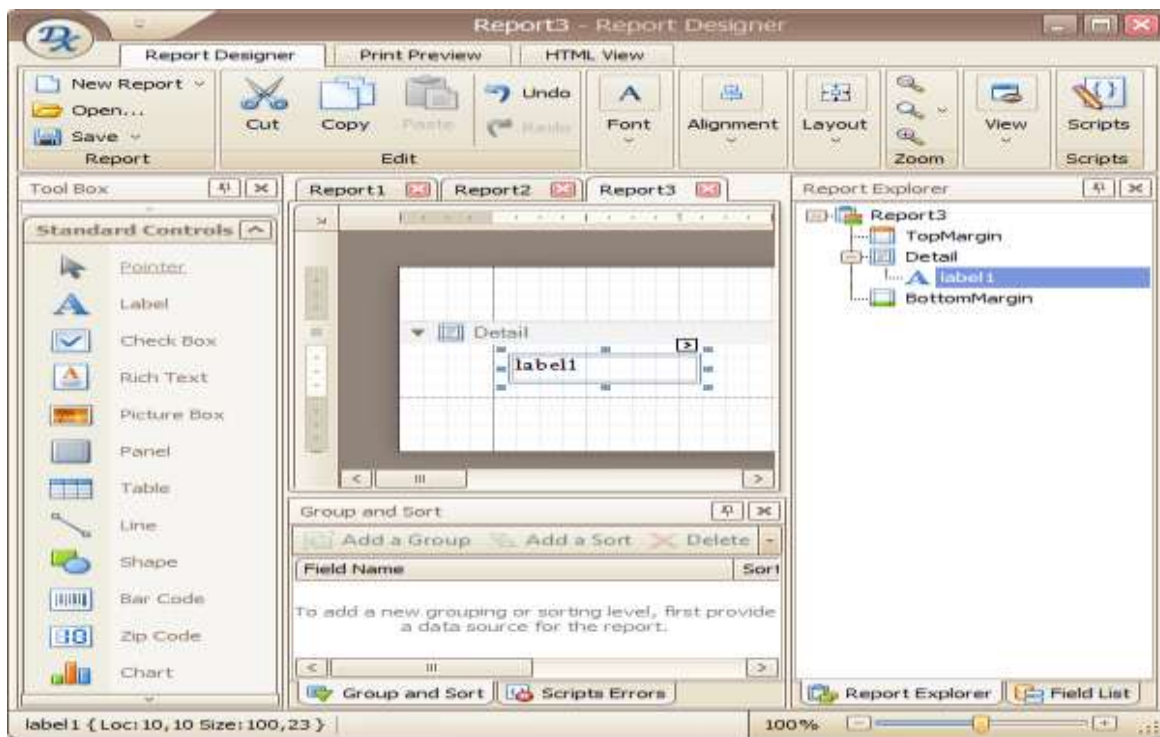


Figura No.6 Diseño XtraReport

En esta versión se ha hecho más fácil de guardar y cargar informes a cualquier fuente de datos en el diseñador de informes del usuario final. Dependiendo de la arquitectura de la aplicación, se puede definir un informe personalizado de almacenamiento para guardar los informes como archivos en un disco, un archivo zip, o directamente a una base de datos.

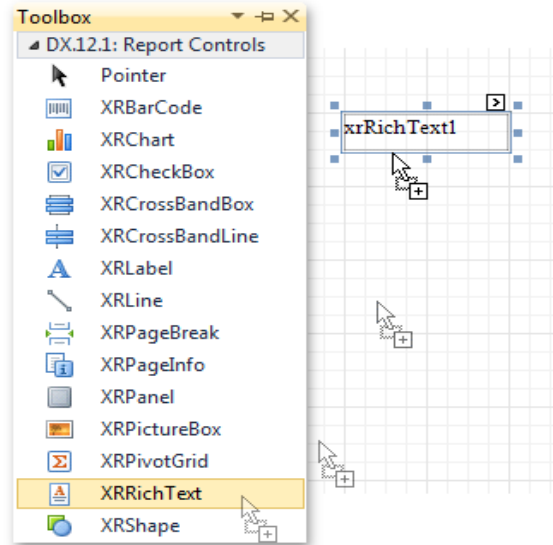


Figura No.7 Cuadro de Herramientas de XtraChart

3.1.5.4 LookUp Editor

Este nuevo editor combina la funcionalidad de búsqueda con una nueva función de búsqueda. Al igual que el control de GridLookUpEdit, el Editor de Mapas Buscar incrusta un control de cuadrícula en el menú desplegable, y ofrece numerosas opciones de personalización de la cuadrícula. El built-in cuadro Buscar le permite localizar rápidamente los datos.

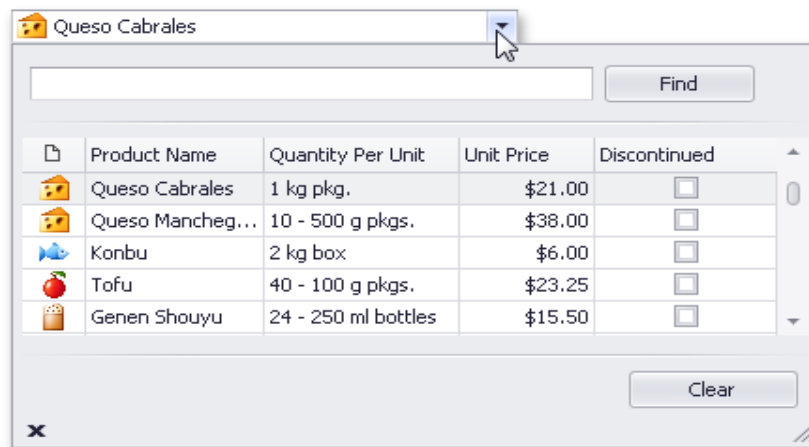


Figura No.8 Diseño LookUp Editor

Para localizar los datos de la consulta, basta con introducir caracteres en el cuadro Buscar y el control filtrar registros en consecuencia (aquellos que cumplan los criterios de búsqueda):



El control proporciona un conjunto de opciones y eventos que le permiten:

- Seleccione el modo de búsqueda automática o manual
- Utilice los Empieza con Contiene u operador de comparación
- Especificar columnas de búsqueda
- Implementar soporte para agregar nuevos registros a través de la ventana desplegable, etc.

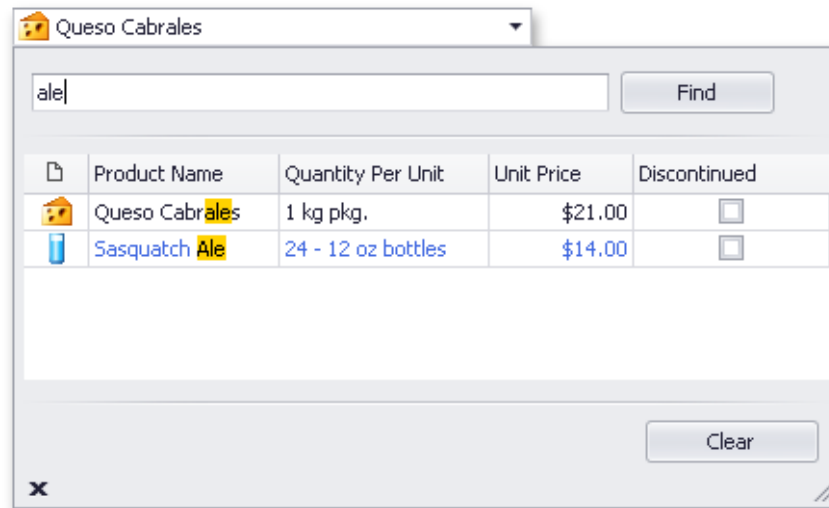


Figura No.9 BusquedaLookUp Editor



3.1.6 SQL Server Express 2008

Es una plataforma global de base de datos, que ofrece administración de datos empresariales con herramientas integradas de inteligencia empresarial (BI). El motor de la base de datos SQL Server 2008 ofrece almacenamiento más seguro y confiable tanto para datos relacionales como estructurados, lo que le permite crear y administrar aplicaciones de datos altamente disponibles y con mayor rendimiento.

El motor de datos SQL Server 2008 constituye el núcleo de esta solución de administración de datos empresariales. Asimismo, SQL Server 2008 combina lo mejor en análisis, información, integración y notificación. Esto permite crear soluciones de BD rentables que ayuden a su equipo a incorporar datos.

La integración directa con Microsoft Visual Studio, el Microsoft Office System y un conjunto de nuevas herramientas de desarrollo, incluido el Business IntelligenceDevelopment Studio, distingue al SQL Server 200.

3.1.6.1 Características de SQL Server Express 2008

- ✓ **Mirroring de Bases de Datos:** amplía las posibilidades de duplicación de logs (“log shipping”) proporcionando a los administradores de BBDD la opción de mirroring. Los administradores pueden usar esta funcionalidad para garantizar la disponibilidad de sus sistemas SQL mediante la configuración de un servidor en espera para su activación automática en caso de fallo (failover).
- ✓ **Backups duplicados (“mirroredbackup”):** SQL Server 2008 incluye un nuevo soporte para volúmenes de backup espejados, aumentando la disponibilidad de las copias de seguridad de SQL Server. La posibilidad de replicar el backup permite resolver posibles problemas de corrupción del medio físico de copia.
- ✓ **Mejoras en Transact-SQL:** introduce muchas posibilidades nuevas para el desarrollo de aplicaciones de bases de datos escalables. Estas mejoras incluyen el manejo de errores, nuevas posibilidades de consultas recursivas y soporte para nuevas funcionalidades del motor de SQL Server.
- ✓ **Conexión de Administrador dedicada:** introduce la conexión de administración dedicada, que pueden utilizar los administradores de BBDD para acceder a un servidor en explotación aun cuando el servidor está bloqueado o no disponible por cualquier motivo. Así, los administradores podrán ejecutar funciones de





diagnóstico, o sentencias Transact SQL, a fin de poder resolver problemas en el servidor. (Server, 2012)

3.1.6.2 Procedimientos almacenados (storeprocedure)

Un procedimiento almacenado (store procedure) no es más que una secuencia ordenada de instrucciones T-SQL, que pueden recibir y proporcionar parámetros provistos por el usuario y se pueden guardar en el servidor con un nombre, para posteriormente se invocados y ejecutados, por lo regular desde una aplicación (Escritorio o Web). Desde la versión 2008, se incorpora la posibilidad de utilizar procedimientos almacenados usando el CLR de .NET. Es decir tenemos dos tipos de procedimientos almacenados.

Un procedimiento almacenado CLR es una referencia a un método de un ensamble (dll) de .NET Framework que puede aceptar y devolver parámetros suministrados por el usuario. (<http://mspnor.wordpress.com/2008/10/31/sqlprocedimientos-almacenados-paso-a-paso/>, 2012)

Cuando crea una aplicación con Microsoft SQL Server 2008, el lenguaje de programación Transact-SQL es la principal interfaz de programación entre las aplicaciones y la base de datos de Microsoft SQL Server. Cuando utiliza programas Transact-SQL, dispone de dos métodos para almacenar y ejecutar los programas.

- Puede almacenar los programas localmente y crear aplicaciones que envían los comandos a SQL Server y procesan los resultados.
- Puede almacenar los programas como procedimientos almacenados en SQL Server y crear aplicaciones que ejecutan los procedimientos almacenados y procesan los resultados. (microsoft, 2012)

Los procedimientos almacenados de Microsoft SQL Server son similares a los procedimientos de otros lenguajes de programación en el sentido de que pueden:

- Aceptar parámetros de entrada y devolver varios valores en forma de parámetros de salida al lote o al procedimiento que realiza la llamada.
- Contener instrucciones de programación que realicen operaciones en la base de datos, incluidas las llamadas a otros procedimientos.
- Devolver un valor de estado a un lote o a un procedimiento que realiza una llamada para indicar si la operación se ha realizado correctamente o se han producido errores (y el motivo de éstos).





Puede utilizar la instrucción EXECUTE de Transact-SQL para ejecutar un procedimiento almacenado. Los procedimientos almacenados difieren de las funciones en que no devuelven valores en lugar de sus nombres ni pueden utilizarse directamente en una expresión.

Utilizar procedimientos almacenados en SQL Server en vez de programas Transact-SQL almacenados localmente en equipos cliente presenta las siguientes ventajas:

- Se registran en el servidor.
- Pueden incluir atributos de seguridad (como permisos) y cadenas de propiedad, además se les pueden asociar certificados. Los usuarios pueden disponer de permiso para ejecutar un procedimiento almacenado sin necesidad de contar con permisos directos en los objetos a los que se hace referencia en el procedimiento.
- Mejoran la seguridad de la aplicación. Los procedimientos almacenados con parámetros pueden ayudar a proteger la aplicación ante ataques por inyección de código SQL.
- Permiten una programación modular. Puede crear el procedimiento una vez y llamarlo desde el programa tantas veces como desee. Así, puede mejorar el mantenimiento de la aplicación y permitir que las aplicaciones tengan acceso a la base de datos de manera uniforme.
- Constituyen código con nombre que permite el enlace diferido. Esto proporciona un nivel de direccionamiento indirecto que facilita la evolución del código.
- Pueden reducir el tráfico de red. Una operación que necesite centenares de líneas de código Transact-SQL puede realizarse mediante una sola instrucción que ejecute el código en un procedimiento, en vez de enviar cientos de líneas de código por la red. (microsoft, 2012)





----Agregar FacturaVenta

go

createprocedure guardarfacturaventa

@idcliente varchar(50),

@fechaventa varchar(50),

@totalventa varchar(50)

Asbegin

insertinto Tbl_FacturaVenta(IdCliente, FechaVenta, TotalVenta)values (@idcliente, @fechaventa, @totalventa)

end

----Agregar DetalleFV

go

createprocedure guardardetallefacturaventa

@idventa varchar(50),

@idproducto varchar(50),

@existenciaproducto int,

@cantidad varchar(50),

@costototal varchar(60)

as

begintransaction

begintry

insertinto Tbl_DetalleFacturaVenta(IdVenta, IdProducto, Cantidad, CostoTotal)

values (@idventa, @idproducto, @cantidad, @costototal)

update Tbl_Productos set ExistenciaProducto = @existenciaproducto where IdProducto = @idproducto

committransaction

endtry

begincatch

rollbacktransaction

print'Ocurrio un Error'

endcatch





--- VISUALIZAR FACTURAVENTA

```
go
CREATEprocedure Consultarfacturaventa

as
begin
select IdVenta, NombreCliente, FechaVenta, TotalVenta from Tbl_Clientes innerjoin
Tbl_FacturaVenta on
Tbl_Clientes.IdCliente = Tbl_FacturaVenta.IdCliente;
end
```

---MODIFICAR CLIENTE

```
go
CREATEprocedure Modificarcliente
@idcliente int,
@nombrecliente varchar(50),
@direccioncliente varchar(50),
@telefonocliente varchar(50),
@departamentocliente varchar(100),
@emailcliente varchar(60)

as
begin
update Tbl_Clientes setNombreCliente = @nombrecliente, DireccionCliente =
@direccioncliente, DepartamentoCliente = @departamentocliente, TelefonoCliente =
@telefonocliente, EmailCliente = @emailcliente
WHERE IdCliente = @idcliente
End
```

-----Obtener Inventario FacturaVenta entre fechas

```
go
createprocedure obtenerinventarioFV

@fechai datetime,
@fechaf datetime

Asbegin
select IdVenta, NombreCliente, FechaVenta, TotalVenta from Tbl_Clientes innerjoin
Tbl_FacturaVenta on Tbl_Clientes.IdCliente = Tbl_FacturaVenta.IdCliente where
FechaVenta Between @fechai and @fechaf;
end
```





4. ANÁLISIS DE REQUISITOS





Especificación de requisitos de software

1. Introducción.

1.1 Propósito:

Definición del conjunto de especificaciones de requisitos de software que debe cumplir la aplicación que gestione el sistema de control de inventario de una Librería.

La aplicación de control de inventario ha sido encargada por la Librería Guadalupe Corinto, para devaluar sus carencias de mercaderías con el fin de fijar su posible masa de producción. Cuya funcionalidad se centra en la automatización de las entradas y salidas de productos, la actualización de la información y generar reportes de los clientes, productos, ventas, etc.

Este documento está dirigido a la dirección de la Empresa y usuarios finales que deberán estudiarlo para su aprobación o desacuerdo antes de abordar la fase de análisis.

1.2 Alcance:

El producto que vamos a describir es un programa que desempañara el papel del administrador de la Librería.

Este producto debe ser capaz de adaptarse a cualquier distribuidora o tienda con el mismo fin, o sea, según el número de productos, estantes, vitrinas, vendedores, etc. El programa deberá funcionar de igual manera.

Además que el programa debe ser capaz de brindarnos la información del estado global de la Librería en cualquier momento, avisándonos todos los sucesos que han ocurrido en el día.

Referente a la parte de aplicación, las funciones que realizara la aplicación serán:

1. Registrar Usuario.
2. Registrar Proveedor.
3. Registrar Cliente.
4. Registrar Producto.
5. Registrar Factura Venta.
6. Registrar Detalle Factura Venta.
7. Registrar Factura Compra.
8. Registrar Detalle Factura Compra.
9. Modificar Producto.
10. Modificar Proveedor.
11. Modificar Usuario.
12. Modificar Cliente.
13. Eliminar Producto.
14. Eliminar proveedor.
15. Eliminar Usuario.





16. Eliminar Cliente.
17. Buscar Producto.
18. Buscar Proveedor.
19. Buscar Usuario.
20. Buscar Cliente.
21. Buscar Factura Venta.
22. Buscar Factura Compra.
23. Generar Factura Venta.
24. Reporte Factura Venta.
25. Reporte Factura Compra.
26. Reporte Clientes.
27. Reporte Proveedores.
28. Reporte Usuarios.
29. Reporte Productos.
30. Reporte Productos Faltantes.
31. Respaldo Base de Datos.
32. Restaurar Base de Datos.

En esta aplicación no se abarca la nómina de pago de la Librería.

1.3 Definiciones Acrónimos y Abreviaturas:

Conlleve: llevar a un lugar o propósito.

Activos Corrientes: La comprenden las cuentas que registran los montos de efectivos y bancos de la empresa.

Proveedor: Empresa que suministra los productos para abastecer a la librería. Lo llamaremos PROVEEDOR.

Ventas: Procesos por el cual se satisfacen las necesidades del comprador para el beneficio de ambas partes. En el resto del ERS se le llamara VENTAS.

Compra: El costo de los productos que adquiere la librería para revenderla a los clientes.

Pago: Se le conoce como el abono correspondiente que realiza la Librería a los PROVEEDORES por una COMPRA.

Cliente: Son aquellas personas que realizan una compra en la librería.

Producto: Artículos que vende la librería a los clientes.

Factura: Documento con el cual se comprueba una acción, normalmente una compraventa que proporciona información importante tanto para comprador como vendedor.





Inventario: Cuenta de activo utilizado para representar todo aquello que es objeto de compra o venta del negocio.

Librería: Es un establecimiento comercial cuyo principal producto a la venta son libros, etc.

Corinto: es un municipio del Departamento de Chinandega, en Nicaragua.

Niveles de Operatividad: El acceso a los sistemas de información debe ser restringida al usuario que no cuente con el permiso o autorización de operar el sistema.

De manera general estos se pueden clasificar:

- Nivel de Administración.
- Nivel de Usuario.

Base de Datos: es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

Software: Es todo el conjunto intangible de datos y programas de la computadora.

1.4 Visión General:

Como primer paso se realizará una visión general del producto que se desea estudiar y detallar cada uno de los requisitos específicos de forma individual.

1.5 Referencias:

Pressman Roger S. Ingeniería del software, un enfoque práctico. 3ra edición.

2. Descripción general:

2.1 Relaciones del Producto.

El equipo en el que se desarrollara e implementara la aplicación tiene las siguientes características:

Procesador	Pentium IV
Marca	Dell
Velocidad del procesador	1.6 GHZ.
Capacidad del disco duro	60 GB.
Memoria RAM	512 MB.





Requisitos Óptimos para el funcionamiento del sistema.

Procesador	Dual Core
Marca	Dell
Velocidad del procesador	2.2 GHZ.
Capacidad del disco duro	80 GB.
Memoria RAM	1 GB.

2.2 Funciones del Producto.

El producto software tiene la facilidad de brindar información que necesite el administrador de la librería.

Funciones de la aplicación:

- ✓ El usuario debe introducir los datos al ordenador para actualizar la base de datos del sistema.
- ✓ Cuando se necesita agregar un proveedor, el usuario deberá llenar en el ordenador el formulario de ingreso del mismo para posteriormente almacenarlo en la base de datos.
- ✓ Cuando se desea anexar un nuevo cliente en nuestros registros. El usuario deberá llenar en el ordenador los datos del mismo para luego almacenarlo en la base de datos.
- ✓ Si se compra un nuevo producto el usuario deberá ser capaz de registrar los datos del nuevo producto en el ordenador.
- ✓ Al tener los datos de los clientes en orden y analizados se debe presentar factura.
- ✓ Cada vez que se requiera un Reporte del cliente, tanto crédito como contado, el operador determina la aplicación para generar el listado.
- ✓ El operador dispone de los siguientes Reportes en forma de diario.
 - Reporte de las Compras.
 - Reporte de las Ventas.
 - Reporte de inventario actualizado.

2.3 Características del Usuario.

Los usuarios finales de la aplicación serán personas que tienen que ser entrenadas en el manejo del sistema operativo Windows y del Sistema de inventario del diseño de la librería.

2.4 Restricciones Generales.

- El lenguaje de programación a utilizar será Visual Studio 2008.
- Sistema gestor de base de datos Microsoft SQL Server 2008.
- Sistema Operativo: Windows XP, Vista y 7.





Se deben seguir los estándares de la programación C#.

3. Requisitos Específicos.

3.1 Requisitos Funcionales.

3.1.1 Registrar Usuario

3.1.1.1 Especificación

3.1.1.1.1 Introducción

Esta función deberá capturar los datos de un nuevo usuario para el sistema.

3.1.1.1.2 Entradas

Pantalla:

- NombreUsuario
- NombreCompletoUsuario
- DireccionUsuario
- TelefonoUsuario
- CedulaUsuario
- EmailUsuario
- FechaIngreso_Usuario
- Contraseña_Usuario
- Rol_Usuario

Por el sistema:

- Id_Usuario.

3.1.1.1.3 Procesos

Se mostrará por pantalla una interfaz donde se introduzcan los datos del nuevo usuario, luego los nuevos datos del usuario serán almacenados en la base de datos.

3.1.1.1.4 Salidas

Los datos mencionados anteriormente se almacenaran en la base de datos.

3.1.1.2 Interfaces externas.

3.1.1.2.1 Interfaces de usuario.

La captura de los datos del usuario se realizará de forma interactiva por pantalla.





3.1.1.2.2 Interfaces hardware

No existen ningunas interfaces hardware.

3.1.1.2.3 Interfaces software

El proceso interactúa con la base de datos de la librería.

3.1.1.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.

3.1.2 Registrar Proveedor.

3.1.2.1 Especificación.

3.1.2.1.1 Introducción.

Este proceso capturará los datos de cada proveedor y realizará la actualización de la base de datos.

3.1.2.1.2 Entradas.

Pantalla:

- NombreProveedor
- ApellidosProveedor
- DireccionProveedor
- TelefonoProveedor

Datos proporcionados por el sistema:

- IdProveedor.

3.1.2.1.3 Proceso:

Se mostrará una interfaz gráfica donde el usuario introducirá los datos del proveedor y los almacenará en la base de datos.

Los datos necesarios a introducir:

- NombreProveedor: es un dato obligatorio.
- ApellidoProveedor: es un dato obligatorio.
- DirecciónProveedor: es un dato obligatorio.
- TeléfonoProveedor: es un dato obligatorio.



3.1.2.1.4 Salidas

Los datos antes mencionados se guardarán en la base de datos.

3.1.2.2 Interfaces externas.

3.1.2.2.1 Interfaces de usuario.

La captura de los datos del proveedor se realizará de forma interactiva por pantalla.

3.1.2.2.2 Interfaces hardware

No existen ningunas interfaces hardware.

3.1.2.2.3 Interfaces software

El proceso interactúa con la base de datos de la empresa.

3.1.2.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.

3.1.3 Registrar Cliente

3.1.3.1 Especificación.

3.1.3.1.1 Introducción.

Este proceso capturará los datos del cliente y luego realizará la actualización de la base de datos.

3.1.3.1.2 Entradas.

Pantalla:

NombreCliente

DireccionCliente

TelefonoCliente

DepartamentoCliente

EmailCliente

Datos proporcionados por el sistema:

IdCliente





3.1.3.1.3 Proceso:

Se mostrará una interfaz gráfica donde el usuario introducirá los datos del cliente y los almacenará en la base de datos.

3.1.3.1.4 Salidas

Los datos antes mencionados se guardarán en la base de datos.

3.1.3.2 Interfaces externas.

3.1.3.2.1 Interfaces de usuario.

La captura de los datos del cliente se realizara de forma interactiva por pantalla.

3.1.3.2.2 Interfaces hardware

No existen ningunas interfaces hardware.

3.1.3.2.3 Interfaces software

El proceso interactúa con la base de datos de la Librería.

3.1.3.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.

3.1.4 Registrar Producto.

3.1.4.1 Especificación.

3.1.4.1.1 Introducción.

Este proceso deberá capturar los datos de los productos adquiridos de los proveedores.

3.1.4.1.2 Entradas.

Pantalla:

NombreProducto

ExistenciaProducto

PrecioCompra

PrecioUnidad

Observaciones





Datos proporcionados por el sistema:

IdProducto

3.1.4.1.3 Proceso:

Se mostrará una interfaz gráfica donde el usuario introducirá los datos del producto y lo almacenará en la base de datos.

3.1.4.1.4 Salidas

Se guardaran en la base de datos los datos antes mencionados.

3.1.4.2 Interfaces externas.

3.1.4.2.1 Interfaces de usuario.

La captura de los datos del producto se realizara de forma interactiva.

3.1.4.2.2 Interfaces hardware

No existen ningunas interfaces hardware.

3.1.4.2.3 Interfaces software

El proceso interactúa con la base de datos de la Librería.

3.1.4.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.

3.1.5 Registrar Factura Venta.

3.1.5.1 Especificación.

3.1.5.1.1 Introducción.

Este proceso realiza la captura de los datos de los datos de las ventas a realizar a los clientes y actualizara la base de datos.

3.1.5.1.2 Entradas.

Datos proporcionados por el sistema:

IdVenta





IdCliente

TotalVenta

3.1.5.1.3 Proceso:

Se mostrará por pantalla una interfaz gráfica de la venta donde el usuario introducirá los datos correspondientes. Después almacenara los datos antes mencionados en la base de datos.

3.1.5.1.4 Salidas

Los datos mencionados anteriormente se almacenaran en la base de datos y se restara al inventario la cantidad de productos vendidos.

3.1.5.2 Interfaces externas.

3.1.5.2.1 Interfaces de usuario.

La captura de los datos de la venta se realizara de forma interactiva por pantalla.

3.1.5.2.2 Interfaces hardware

No existen ningunas interfaces hardware.

3.1.5.2.3 Interfaces software

El proceso interactúa con la base de datos de la librería.

3.1.5.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.

3.1.6 Registrar Detalle Factura Venta.

3.1.6.1 Especificación.

3.1.6.1.1 Introducción.

Este proceso realiza la captura de los datos de los datos de las ventas a realizar a los clientes y actualizara la base de datos.

3.1.6.1.2 Entradas.

Datos proporcionados por el usuario:

Cantidad

Datos proporcionados por el sistema:





IdVenta

IdProducto

IdProducto

CostoTotal

3.1.6.1.3 Proceso:

Se mostrará por pantalla una interfaz gráfica del detalle de la venta donde el usuario introducirá los datos correspondientes. Después se almacenarán los datos antes mencionados en la base de datos.

3.1.6.1.4 Salidas

Los datos mencionados anteriormente se almacenarán en la base de datos y se restará al inventario la cantidad de productos vendidos.

3.1.6.2 Interfaces externas.

3.1.6.2.1 Interfaces de usuario.

La captura de los datos de la venta se realizará de forma interactiva por pantalla.

3.1.6.2.2 Interfaces hardware

No existen ningunas interfaces hardware.

3.1.6.2.3 Interfaces software

El proceso interactúa con la base de datos de la librería.

3.1.6.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.

3.1.7 Registrar Factura Compra

3.1.7.1 Especificación.

3.1.7.1.1 Introducción.

Este proceso realizará la captura de los datos de las compras realizadas a los proveedores, para luego actualizar la base de datos.

3.1.7.1.2 Entradas.

Datos Proporcionados por el Sistema:





IdCompra

IdProveedor

IdProveedor

FechaCompra

TotalCompra

3.1.7.1.3 Proceso:

Se mostrará por pantalla una interfaz gráfica donde el usuario introducirá los datos correspondientes a la compra realizada a los proveedores. Después se almacenarán los datos antes mencionados en la base de datos.

3.1.7.1.4 Salidas

Los datos mencionados anteriormente se almacenarán en la base de datos y se sumará al inventario la cantidad de productos comprados.

3.1.7.2 Interfaces externas.

3.1.7.2.1 Interfaces de usuario.

La captura de los datos de la compra se realizará de forma interactiva por pantalla.

3.1.7.2.2 Interfaces hardware

No existen ningunas interfaces hardware.

3.1.7.2.3 Interfaces software

El proceso interactúa con la base de datos de la Librería.

3.1.7.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.

3.1.8 Registrar Detalle Factura Compra.

3.1.8.1 Especificación.

3.1.8.1.1 Introducción.

Este proceso realiza la captura de los datos de los datos de las ventas a realizar a los clientes y actualizara la base de datos.





3.1.8.1.2 Entradas.

Datos proporcionados por el usuario:

Cantidad

Datos proporcionados por el sistema:

IdCompra

IdProducto

CostoTotal

3.1.8.1.3 Proceso:

Se mostrará por pantalla una interfaz gráfica del detalle de la venta donde el usuario introducirá los datos correspondientes. Después se almacenarán los datos antes mencionados en la base de datos.

3.1.8.1.4 Salidas

Los datos mencionados anteriormente se almacenarán en la base de datos y se restará al inventario la cantidad de productos vendidos.

3.1.8.2 Interfaces externas.

3.1.8.2.1 Interfaces de usuario.

La captura de los datos de la venta se realizará de forma interactiva por pantalla.

3.1.8.2.2 Interfaces hardware

No existen ningunas interfaces hardware.

3.1.8.2.3 Interfaces software

El proceso interactúa con la base de datos de la librería.

3.1.8.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.

3.1.9 Modificar Producto.

3.1.9.1 Especificación.





3.1.9.1.1 Introducción.

Este proceso realizará la modificación del producto para luego actualizar la información en la base de datos.

3.1.9.1.2 Entradas.

Datos Proporcionados por el Usuario:

NombreProducto

ExistenciaProducto

PrecioCompra

PrecioUnidad

Observaciones

Datos proporcionados por el sistema:

IdProducto

3.1.9.1.3 Proceso:

Se mostrara por pantalla una interfaz donde se introduzca los datos de la información del producto a modificar.

3.1.9.1.4 Salidas

Los datos mencionados anteriormente se almacenaran en la base de datos, modificando la información que anteriormente contenía la base de datos.

3.1.9.2 Interfaces externas.

3.1.9.2.1 Interfaces de usuario.

La captura de los datos de la modificación se realizara de forma interactiva por pantalla.

3.1.9.2.2 Interfaces hardware

No existen ningunas interfaces hardware.

3.1.9.2.3 Interfaces software

El proceso interactúa con la base de datos de la Librería.





3.1.9.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.

3.1.10 Modificar Proveedor

3.1.10.1 Especificación.

3.1.10.1.1 Introducción.

Este proceso realizará la modificación de un proveedor para luego actualizar la información de la base de datos.

3.1.10.1.2 Entradas.

Datos proporcionado por el usuario.

NombreProveedor

ApellidosProveedor

DireccionProveedor

TelefonoProveedor

Datos proporcionados por el sistema:

IdProveedor.

3.1.10.1.3 Proceso:

Se mostrara por pantalla una interfaz donde se introduzca los nuevos datos del proveedor a modificar, para luego actualizar los datos introducidos a la base de datos.

3.1.10.1.4 Salidas

Los datos mencionados anteriormente se almacenaran en la base de datos, modificando la información que anteriormente contenía la base de datos.

3.1.10.2 Interfaces externas.

3.1.10.2.1 Interfaces de usuario.

La captura de los datos de la modificación se realizara de forma interactiva por pantalla.

3.1.10.2.2 Interfaces hardware





No existen ningunas interfaces hardware.

3.1.10.2.3 Interfaces software

El proceso interactúa con la base de datos de la Librería.

3.1.10.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.

3.1.11 Modificar Usuario

3.1.11.1 Especificación.

3.1.11.1.1 Introducción.

Este proceso realizará la modificación de un usuario para luego actualizar la información de la base de datos.

3.1.11.1.2 Entradas.

Pantalla:

NombreUsuario

NombreCompletoUsuario

DireccionUsuario

TelefonoUsuario

CedulaUsuario

EmailUsuario

FechaIngreso_Usuario

Contraseña_Usuario

Rol_Usuario

Por el sistema:

Id_Usuario.



3.1.11.1.3 Proceso:

Se mostrará por pantalla una interfaz donde se introduzca los nuevos datos del usuario a modificar, para luego actualizar los datos introducidos a la base de datos.

3.1.11.1.4 Salidas

Los datos mencionados anteriormente se almacenarán en la base de datos, modificando la información que anteriormente contenía la base de datos.

3.1.11.2 Interfaces externas.

3.1.11.2.1 Interfaces de usuario.

La captura de los datos de la modificación del usuario se realizará de forma interactiva por pantalla.

3.1.11.2.2 Interfaces hardware

No existen ningunas interfaces hardware.

3.1.11.2.3 Interfaces software

El proceso interactúa con la base de datos de la Librería.

3.1.11.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.

3.1.12 Modificar Cliente

3.1.12.1 Especificación

3.1.12.1.1 Introducción

Esta función deberá modificar un cliente registrado anteriormente en el sistema.

3.1.12.1.2 Entradas

Pantalla:

NombreCliente

DireccionCliente

TelefonoCliente

DepartamentoCliente





EmailCliente

Datos proporcionados por el sistema:

IdCliente

3.1.12.1.3 Procesos

Se mostrará por pantalla una interfaz donde se introduzca los nuevos datos del cliente a modificar.

3.1.12.1.4 Salidas

No Hay ninguna salida.

3.1.12.2 Interfaces externas.

3.1.12.2.1 Interfaces de usuario.

La captura de los datos introducidos por el usuario o administrador se realizará de forma interactiva por pantalla.

3.1.12.2.2 Interfaces hardware

No existen ningunas interfaces hardware.

3.1.12.2.3 Interfaces software

El proceso interactúa con la base de datos de la Librería.

3.1.12.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.

3.1.13 Eliminar Producto

3.1.13.1 Especificación

3.1.13.1.1 Introducción

Esta función deberá realizar la eliminación de los productos especificados.

3.1.13.1.2 Entradas

Pantalla:

NombreProducto





3.1.13.1.3 Procesos

Se mostrará por pantalla una interfaz donde se seleccione el producto a eliminar se dará click sobre la opción eliminar registro, luego se actualizará la base de datos.

3.1.13.1.4 Salidas

No hay salida.

3.1.13.2 Interfaces externas.

3.1.13.2.1 Interfaces de usuario.

La eliminación de datos de los productos se realizará de forma interactiva por pantalla.

3.1.13.2.2 Interfaces hardware

No existen ningunas interfaces hardware.

3.1.13.2.3 Interfaces software

El proceso interactúa con la base de datos de la librería.

3.1.13.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.

3.1.14 Eliminar Proveedor

3.1.14.1 Especificación

3.1.14.1.1 Introducción

Esta función deberá realizar la eliminación del proveedor especificado.

3.1.14.1.2 Entradas

Pantalla:

NombreProveedor

3.1.14.1.3 Procesos

Se mostrará por pantalla una interfaz donde se seleccione el proveedor a eliminar se dará click sobre la opción eliminar registro, luego se actualizará la base de datos.



3.1.14.1.4 Salidas

No hay salida.

3.1.14.2 Interfaces externas.

3.1.14.2.1 Interfaces de usuario.

La eliminación de datos del proveedor se realizará de forma interactiva por pantalla.

3.1.14.2.2 Interfaces hardware

No existen ningunas interfaces hardware.

3.1.14.2.3 Interfaces software

El proceso interactúa con la base de datos de la librería.

3.1.14.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.

3.1.15 Eliminar Usuario

3.1.15.1 Especificación

3.1.15.1.1 Introducción

Esta función deberá eliminar un usuario del sistema.

3.1.15.1.2 Entradas

Pantalla:

NombreUsuario.

Por el sistema:

Id_Usuario

3.1.15.1.3 Procesos

Se mostrará por pantalla una interfaz donde se seleccione el usuario a eliminar se dará click sobre la opción eliminar registro, luego se actualizará la base de datos.



3.1.15.1.3 Salidas

Los datos mencionados anteriormente se modificarán y se actualizará el contenido de la base de datos.

3.1.15.2 Interfaces externas.

3.1.15.2.1 Interfaces de usuario.

La eliminación de los datos del usuario se realizará de forma interactiva por pantalla.

3.1.15.2.2 Interfaces hardware

No existen ningunas interfaces hardware.

3.1.15.2.3 Interfaces software

El proceso interactúa con la base de datos de la librería.

3.1.15.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.

3.1.16 Eliminar Cliente

3.1.16.1 Especificación

3.1.16.1.1 Introducción

Esta función deberá eliminar un cliente del sistema.

3.1.16.1.2 Entradas

No Hay

3.1.16.1.3 Procesos

Se mostrará por pantalla una interfaz donde se seleccione el cliente a eliminar se dará click sobre la opción eliminar registro, luego se actualizará la base de datos.

3.1.16.1.4 Salidas

Los datos mencionados anteriormente se modificarán y se actualizará el contenido de la base de datos.



3.1.16.2 Interfaces externas.

3.1.16.2.1 Interfaces de usuario.

La eliminación de los datos del cliente se realizará de forma interactiva por pantalla.

3.1.16.2.2 Interfaces hardware

No existen ningunas interfaces hardware.

3.1.16.2.3 Interfaces software

El proceso interactúa con la base de datos de la librería.

3.1.16.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.

3.1.17 Buscar Producto

3.1.17.1 Especificación

3.1.17.1.1 Introducción

Esta función deberá realizar la búsqueda de los productos que se encuentran en la Base de Datos.

3.1.17.1.2 Entradas

Pantalla:

NombreProducto

3.1.17.1.3 Procesos

Se mostrará por pantalla una interfaz donde se introduzca el dato a buscar, por NombreProducto.

3.1.17.1.4 Salidas

No hay salida.

3.1.17.2 Interfaces externas.

3.1.17.2.1 Interfaces de usuario.

La búsqueda de datos del producto realizados con anterioridad se realizará de forma interactiva por pantalla.



3.1.17.2.2 Interfaces hardware

No existen ningunas interfaces hardware.

3.1.17.2.3 Interfaces software

El proceso interactúa con la base de datos de la librería.

3.1.17.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.

3.1.18 Buscar Proveedor

3.1.18.1 Especificación

3.1.18.1.1 Introducción

Esta función deberá realizar la búsqueda de los proveedores que se encuentran en el sistema.

3.1.18.1.2 Entradas

Pantalla:

NombreProveedor

3.1.18.1.3 Procesos

Se mostrará por pantalla una interfaz donde se introduzca el dato a buscar, por NombreProveedor.

3.1.18.1.4 Salidas

No hay salida.

3.1.18.2 Interfaces externas.

3.1.18.2.1 Interfaces de usuario.

La búsqueda de datos del proveedor realizados con anterioridad se realizará de forma interactiva por pantalla.

3.1.18.2.2 Interfaces hardware

No existen ningunas interfaces hardware.



3.1.18.2.3 Interfaces software

El proceso interactúa con la base de datos de la librería.

3.1.18.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.

3.1.19 Buscar Usuario

3.1.19.1 Especificación

3.1.19.1.1 Introducción

Esta función deberá realizar la búsqueda de los usuarios que se encuentran en el sistema.

3.1.19.1.2 Entradas

Pantalla:

NombreUsuario

3.1.19.1.3 Procesos

Se mostrará por pantalla una interfaz donde se introduzca el dato a buscar, por NombreUsuario.

3.1.19.1.4 Salidas

No hay salida.

3.1.19.2 Interfaces externas.

3.1.19.2.1 Interfaces de usuario.

La búsqueda de datos del usuario realizados con anterioridad se realizará de forma interactiva por pantalla.

3.1.19.2.2 Interfaces hardware

No existen ningunas interfaces hardware.

3.1.19.2.3 Interfaces software

El proceso interactúa con la base de datos de la librería.

3.1.19.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.





3.1.20 Buscar Cliente

3.1.20.1 Especificación

3.1.20.1.1 Introducción

Esta función deberá realizar la búsqueda de los clientes que se encuentran en el sistema.

3.1.20.1.2 Entradas

Pantalla:

NombreCliente

3.1.20.1.3 Procesos

Se mostrará por pantalla una interfaz donde se introduzca el dato a buscar, por NombreCliente.

3.1.20.1.4 Salidas

No hay salida.

3.1.20.2 Interfaces externas.

3.1.20.2.1 Interfaces de usuario.

La búsqueda de datos del cliente realizados con anterioridad se realizará de forma interactiva por pantalla.

3.1.20.2.2 Interfaces hardware

No existen ningunas interfaces hardware.

3.1.20.2.3 Interfaces software

El proceso interactúa con la base de datos de la librería.

3.1.20.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.

3.1.21 Buscar Factura Venta

3.1.21.1 Especificación





3.1.21.1.1 Introducción

Esta función deberá realizar la búsqueda de las ventas realizadas de la empresa a los clientes, es decir, buscar facturas de las ventas que se hicieron con anterioridad, por NombreCliente.

3.1.21.1.2 Entradas

Pantalla:

NombreCliente

3.1.21.1.3 Procesos

Se mostrará por pantalla una interfaz donde se introduzca el dato a buscar, ya sea NombreCliente.

3.1.21.1.4 Salidas

No hay salida.

3.1.21.2 Interfaces externas.

3.1.21.2.1 Interfaces de usuario.

La búsqueda de datos de las facturas realizadas con anterioridad se realizará de forma interactiva por pantalla.

3.1.21.2.2 Interfaces hardware

No existen ningunas interfaces hardware.

3.1.21.2.3 Interfaces software

El proceso interactúa con la base de datos de la librería.

3.1.21.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.

3.1.22 *Buscar Factura Compra*

3.1.22.1 *Especificación*

3.1.22.1.1 *Introducción*

Esta función deberá realizar la búsqueda de las compras realizadas de la empresa a los proveedores, es decir, buscar facturas de las compras que se hicieron con anterioridad, por NombreProveedor.



3.1.22.1.2 Entradas

Pantalla:

NombreProveedor

3.1.22.1.3 Procesos

Se mostrará por pantalla una interfaz donde se introduzca el dato a buscar, ya sea NombreProveedor.

3.1.22.1.4 Salidas

No hay salida.

3.1.22.2 Interfaces externas.

3.1.22.2.1 Interfaces de usuario.

La búsqueda de datos de las facturas realizadas con anterioridad se realizará de forma interactiva por pantalla.

3.1.22.2.2 Interfaces hardware

No existen ningunas interfaces hardware.

3.1.22.2.3 Interfaces software

El proceso interactúa con la base de datos de la librería.

3.1.22.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.

3.1.23 Generar Factura Venta

3.1.23.1 Especificación

3.1.23.1.1 Introducción

Esta función deberá generar la factura de la venta realizada a los clientes de la empresa.

3.1.23.1.2 Entradas

Las entradas son las capturadas de la venta realizada a los clientes anteriormente.



3.1.23.1.3 Procesos

Se mostrará por pantalla una interfaz donde se generará la factura de la venta realizada a los clientes.

3.1.23.1.4 Salidas

No tiene salida.

3.1.23.2 Interfaces externas.

3.1.23.2.1 Interfaces de usuario.

No tiene interfaz de usuario.

3.1.23.2.2 Interfaces hardware

No existen ningunas interfaces hardware.

3.1.23.2.3 Interfaces software

El proceso interactúa con la base de datos de la librería.

3.1.23.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.

3.1.24 Reporte Factura Venta.

3.1.24.1 Especificación

3.1.24.1.1 Introducción

Esta función deberá generar el reporte de las ventas de productos realizadas a los clientes de la empresa.

3.1.24.1.2 Entradas

No tiene entradas.

3.1.24.1.3 Procesos

Se mostrará por pantalla una interfaz donde se generará el reporte de las ventas realizadas a los clientes.

3.1.24.1.4 Salidas

No tiene salida.





3.1.24.2 Interfaces externas.

3.1.24.2.1 Interfaces de usuario.

No tiene interfaz de usuario.

3.1.24.2.2 Interfaces hardware

No existen ningunas interfaces hardware.

3.1.24.2.3 Interfaces software

El proceso interactúa con la base de datos de la librería.

3.1.24.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.

3.1.25 Reporte Factura Compra.

3.1.25.1 Especificación

3.1.25.1.1 Introducción

Esta función deberá generar el reporte de las compras de productos realizadas a los proveedores de la empresa.

3.1.25.1.2 Entradas

No tiene entradas.

3.1.25.1.3 Procesos

Se mostrará por pantalla una interfaz donde se generará el reporte de las compras de productos realizada a los proveedores.

3.1.25.1.4 Salidas

No tiene salida.

3.1.25.2 Interfaces externas.

3.1.25.2.1 Interfaces de usuario.

No tiene interfaz de usuario.

3.1.25.2.2 Interfaces hardware





No existen ningunas interfaces hardware.

3.1.25.2.3 Interfaces software

El proceso interactúa con la base de datos de la librería.

3.1.25.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.

3.1.26 Reporte Clientes.

3.1.26.1 Especificación

3.1.26.1.1 Introducción

Esta función deberá generar el reporte de todos los datos de los clientes registrados en el sistema.

3.1.26.1.2 Entradas

No tiene entradas.

3.1.26.1.3 Procesos

Se mostrará por pantalla una interfaz donde se generará el reporte de todos los clientes registrados en el sistema.

3.1.26.1.4 Salidas

No tiene salida.

3.1.26.2 Interfaces externas.

3.1.26.2.1 Interfaces de usuario.

No tiene interfaz de usuario.

3.1.26.2.2 Interfaces hardware

No existen ningunas interfaces hardware.

3.1.26.2.3 Interfaces software

El proceso interactúa con la base de datos de la librería.

3.1.26.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.





3.1.27 Reporte Proveedores.

3.1.27.1 Especificación

3.1.27.1.1 Introducción

Esta función deberá generar el reporte de todos los datos de los proveedores registrados en el sistema.

3.1.27.1.2 Entradas

No tiene entradas.

3.1.27.1.3 Procesos

Se mostrará por pantalla una interfaz donde se generará el reporte de todos los proveedores registrados en el sistema.

3.1.27.1.4 Salidas

No tiene salida.

3.1.27.2 Interfaces externas.

3.1.27.2.1 Interfaces de usuario.

No tiene interfaz de usuario.

3.1.27.2.2 Interfaces hardware

No existen ningunas interfaces hardware.

3.1.27.2.3 Interfaces software

El proceso interactúa con la base de datos de la librería.

3.1.27.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.

3.1.28 Reporte Usuarios.

3.1.28.1 Especificación

3.1.28.1.1 Introducción

Esta función deberá generar el reporte de todos los datos de los usuarios registrados en el sistema.





3.1.28.1.2 Entradas

No tiene entradas.

3.1.28.1.3 Procesos

Se mostrará por pantalla una interfaz donde se generará el reporte de todos los usuarios registrados en el sistema.

3.1.28.1.4 Salidas

No tiene salida.

3.1.28.2 Interfaces externas.

3.1.28.2.1 Interfaces de usuario.

No tiene interfaz de usuario.

3.1.28.2.2 Interfaces hardware

No existen ningunas interfaces hardware.

3.1.28.2.3 Interfaces software

El proceso interactúa con la base de datos de la librería.

3.1.28.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.

3.1.29 Reporte Productos.

3.1.29.1 Especificación

3.1.29.1.1 Introducción

Esta función deberá generar el reporte de todos los datos de los productos registrados en el sistema.

3.1.29.1.2 Entradas

No tiene entradas.

3.1.29.1.3 Procesos

Se mostrará por pantalla una interfaz donde se generará el reporte de todos los productos registrados en el sistema.





3.1.29.1.4 Salidas

No tiene salida.

3.1.29.2 Interfaces externas.

3.1.29.2.1 Interfaces de usuario.

No tiene interfaz de usuario.

3.1.29.2.2 Interfaces hardware

No existen ningunas interfaces hardware.

3.1.29.2.3 Interfaces software

El proceso interactúa con la base de datos de la librería.

3.1.29.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.

3.1.30 Reporte Productos Faltantes.

3.1.30.1 Especificación

3.1.30.1.1 Introducción

Esta función deberá generar el reporte de todos los datos de los productos que se agotaron para luego reabastecerse de nuevos productos proporcionados por el proveedor.

3.1.30.1.2 Entradas

No tiene entradas.

3.1.30.1.3 Procesos

Se mostrará por pantalla una interfaz donde se generará el reporte de todos los productos agotados.

3.1.30.1.4 Salidas

No tiene salida.

3.1.30.2 Interfaces externas.





3.1.30.2.1 Interfaces de usuario.

No tiene interfaz de usuario.

3.1.30.2.2 Interfaces hardware

No existen ningunas interfaces hardware.

3.1.30.2.3 Interfaces software

El proceso interactúa con la base de datos de la librería.

3.1.30.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.

3.1.31 Respaldo Base de Datos

3.1.31.1 Especificación

3.1.31.1.1 Introducción

Esta función deberá realizar un backup de la base de datos.

3.1.31.1.2 Entradas

No hay entradas.

3.1.31.1.3 Procesos

Se mostrará por pantalla una interfaz donde se pueda respaldar la base de datos en una específica ubicación.

3.1.31.1.4 Salidas

No hay salidas.

3.1.31.2 Interfaces externas.

3.1.31.2.1 Interfaces de usuario.

No hay interfaz de usuario.

3.1.31.2.2 Interfaces hardware

No existen ningunas interfaces hardware.



3.1.31.2.3 Interfaces software

El proceso interactúa con la base de datos de la librería.

3.1.31.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.

3.1.32 Restaurar Base de Datos.

3.1.32.1 Especificación

3.1.32.1.1 Introducción

Esta función deberá restaurar la base de datos utilizada anteriormente, por si ocurrió un problema.

3.1.32.1.2 Entradas

No hay entradas.

3.1.32.1.3 Procesos

Se mostrará por pantalla una interfaz donde se pueda restaurar la base de datos del sistema, en caso de fallas.

3.1.32.1.4 Salidas

No hay salidas.

3.1.32.2 Interfaces externas.

3.1.32.2.1 Interfaces de usuario.

No hay interfaz de usuario.

3.1.32.2.2 Interfaces hardware

No existen ningunas interfaces hardware.

3.1.32.2.3 Interfaces software

El proceso interactúa con la base de datos de la librería.

3.1.32.2.4 Interfaces de comunicaciones.

No existen ningunas interfaces de comunicaciones en la aplicación.



3.2 Requisitos del funcionamiento

3.2.1 Requisitos estáticos.

Solo se puede trabajar en una computadora ya que nuestro Sistema no está basado en redes.

3.2.2 Requisitos dinámicos.

El funcionamiento en cuanto al tiempo de respuesta no debe aumentar a pesar de la cantidad de datos.

3.3 Restricción de diseño.

El formato de respuesta de la aplicación debe contener información acerca del nombre de la librería a la que se realizara la aplicación.

3.4 Atributo de seguridad

3.4.1 Seguridad.

Todos los programas deben de estar protegidos mediante Autenticación de los usuarios al sistema.

3.4.2 Mantenimiento.

Cualquier modificación que afecte a los requerimientos mencionados en este documento deberá ser reflejada en el mismo, así como la documentación obtenida en las fases de análisis, diseño e implementación.

3.4.3 Ayuda en línea

No habrá ayuda en línea. Debido a la carencia de una base informática de los usuarios finales, el proceso llevado a cabo por nosotros constará de un manual de usuario.

3.5 Otros requisitos

3.5.1 Base de datos.

El almacenamiento de información se realizará por medio de una base de datos relacional.

3.5.2 Operaciones.

Todas las operaciones de la base de datos se realizaran según el apartado de Seguridad.



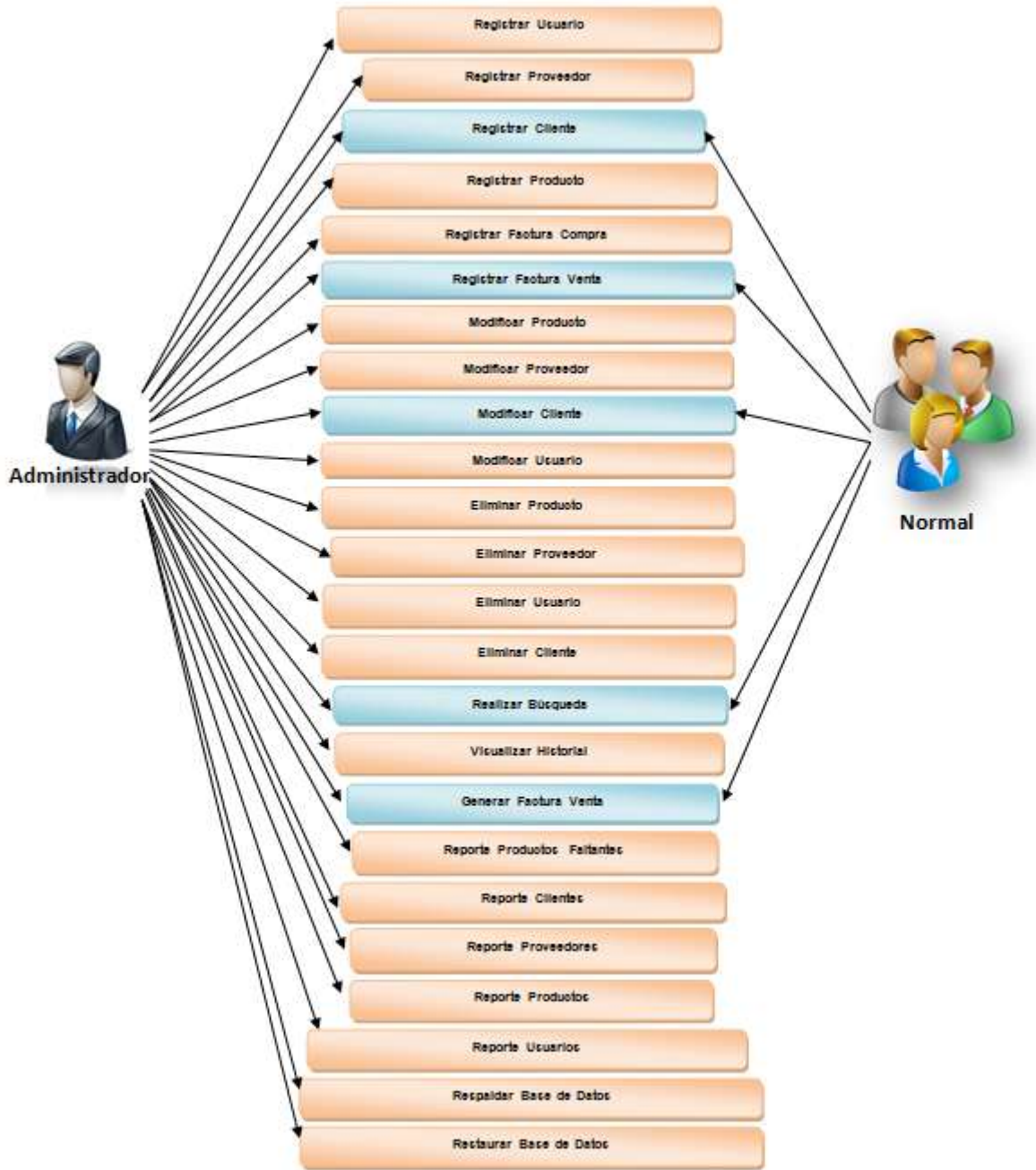


Figura No.10 Casos de Usos Generalizado

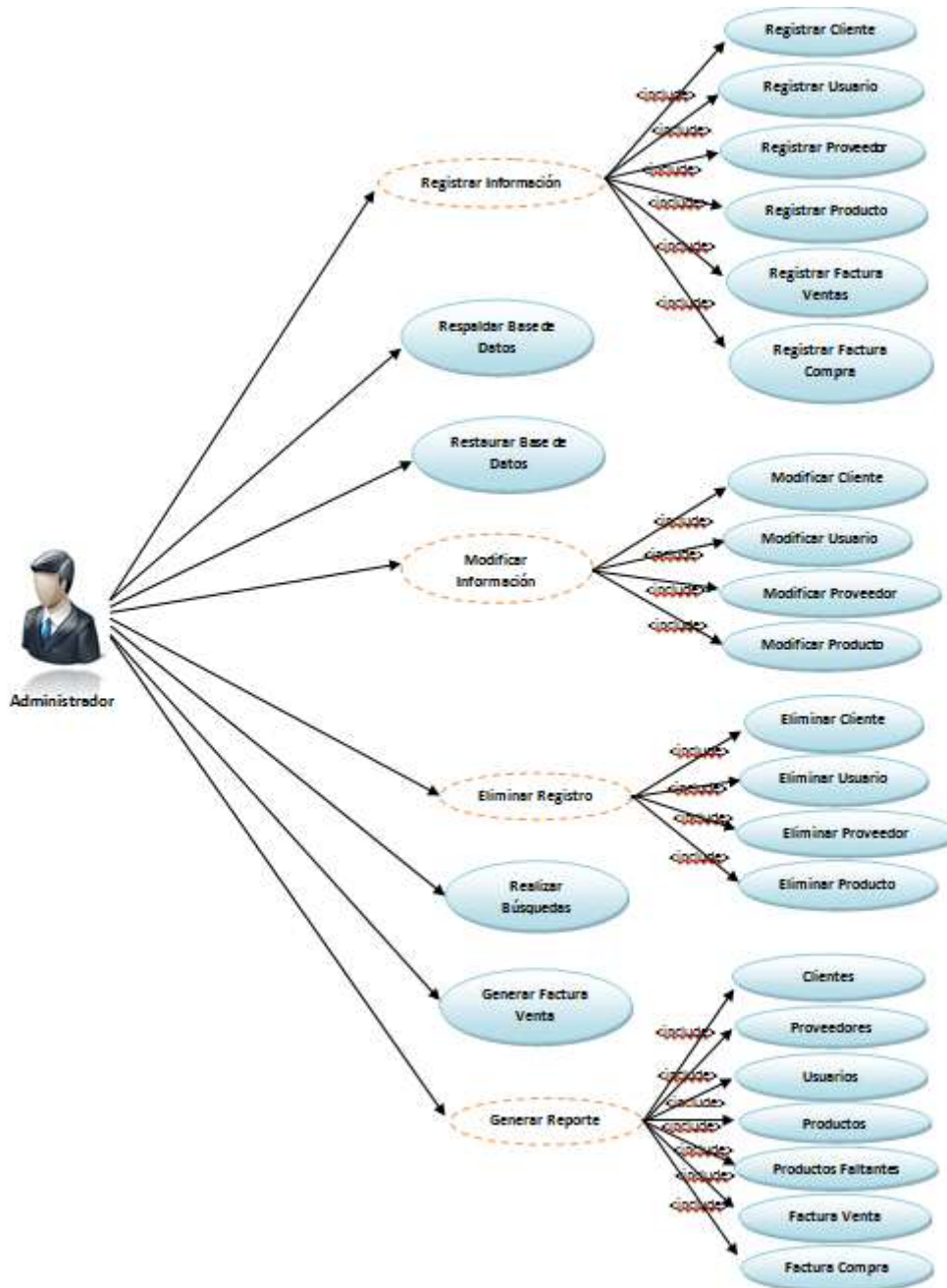


Figura No.11 Diagrama de Casos de Uso del Administrador

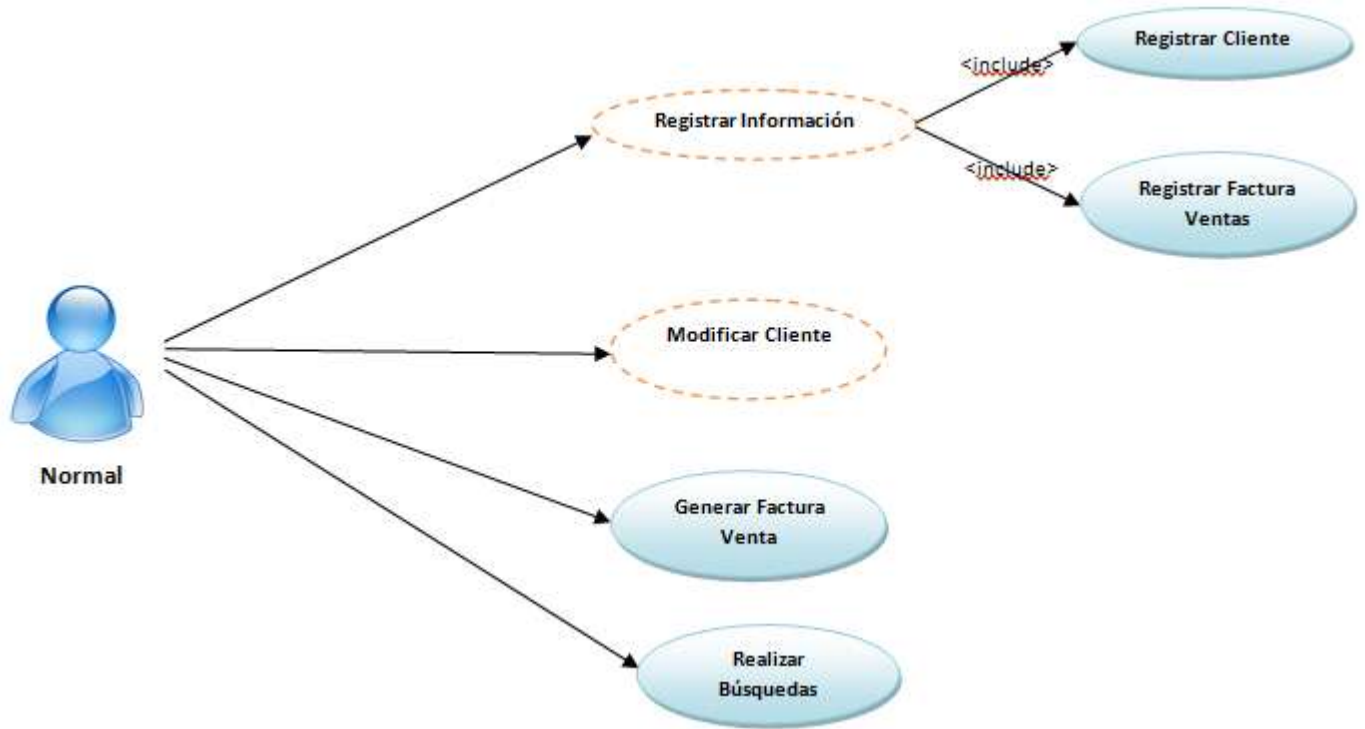


Figura No.12 Diagrama de Casos de Uso del Usuario Normal



5. CASOS DE USOS DOCUMENTADOS

Caso de uso número 1

Caso de Uso:	Registrar Usuario
Actores:	Administrador
Descripción	Registras Usuario
Pre-Condición:	El usuario previamente autenticado por el administrador
Flujo Alternativo	No tiene
Post Condición	Registra, actualiza cliente y factura venta.

Caso de uso número 2

Caso de Uso:	Registrar Proveedor
Actores:	Administrador
Descripción	Se debe registrar un proveedor
Pre-Condición:	El proveedor previamente autenticado por el administrador
Flujo Alternativo	No tiene
Post-Condición:	Registra y Actualiza proveedor

Caso de uso número 3

Caso de Uso:	Registrar Cliente
Actores:	Administrador, Usuario normal
Descripción	Se debe registrar un cliente
Pre-Condición:	El cliente previamente autenticado por el administrador
Flujo Alternativo	No tiene
Post-Condición:	Registra y actualiza cliente

Caso de uso número 4

Caso de Uso:	Registrar Producto
Actores:	Administrador
Descripción	Permite registrar los productos en el sistema
Pre-Condición:	El producto previamente autenticado por el





	administrador
Flujo Alternativo	No tiene
Post-Condición:	Registra y Actualiza producto

Caso de uso número 5

Caso de Uso:	Registrar Facturas Ventas
Actores:	Administrador, Usuario Normal
Descripción:	Los Usuarios registran las facturas ventas del día.
Pre-Condición:	Los Productos y Clientes deben haber sido registrados previamente
Flujo Alternativo	No tiene
Post Condición	No tiene

Caso de uso número 6

Caso de Uso:	Registrar Detalle Factura Venta
Actores:	Sistema
Descripción:	El Sistema registra los detalle de la Factura Venta
Pre-Condición:	Los Usuarios deben haber registrado una nueva transacción
Flujo Alternativo	No tiene
Post Condición	No tiene

Caso de uso número 7

Caso de Uso:	Registrar Factura Compra
Actores:	Administrador
Descripción	El administrador registra factura compra
Pre-Condición:	Los Productos y los Proveedores deben haber sido registrados previamente
Flujo Alternativo	No tiene
Post Condición	No tiene

Caso de uso número 8

Caso de Uso:	Registrar Detalle Factura Compra
Actores:	Sistema
Descripción:	El Sistema registra los detalle de la Factura Compra
Pre-Condición:	Los Usuarios deben haber registrado una nueva transacción





Flujo Alternativo	No tiene
Post Condición	No tiene

Caso de uso número 9

Caso de Uso:	Modificar Producto
Actores:	Administrador
Descripción	El administrador modifica los producto y los faltantes de los estantes
Pre-Condición:	El Producto debe estar previamente registrado
Flujo Alternativo	No tiene
Post Condición	No tiene

Caso de uso número 10

Caso de Uso:	Modificar Proveedor
Actores:	Administrador
Descripción	El administrador modifica el proveedor
Pre-Condición:	El Proveedor debe estar previamente registrado
Flujo Alternativo	No tiene
Post Condición	No tiene

Caso de uso número 11

Caso de Uso:	Modificar Usuario
Actores:	Administrador
Descripción:	El administrador modifica el usuario
Pre-Condición:	El Usuario debe estar previamente registrado
Flujo Alternativo	No tiene
Post Condición	No tiene

Caso de uso número 12

Caso de Uso:	Modificar Cliente
Actores:	Administrador
Descripción:	El administrador modifica el cliente
Pre-Condición:	El Cliente debe estar previamente registrado
Flujo Alternativo	No tiene
Post Condición	No tiene





Caso de uso número 13

Caso de Uso:	Eliminar Producto
Actores:	Administrador
Descripción:	El administrador elimina producto
Pre-Condición:	El Producto debe estar previamente registrado
Flujo Alternativo	No tiene
Post Condición	No tiene

Caso de uso número 14

Caso de Uso:	Eliminar Proveedor
Actores:	Administrador
Descripción:	El administrador elimina proveedor
Pre-Condición:	El Proveedor debe estar previamente registrado
Flujo Alternativo	No tiene
Post Condición	No tiene

Caso de uso número 15

Caso de Uso:	Eliminar Usuario
Actores:	Administrador
Descripción:	El administrador realiza la eliminación de los usuario
Pre-Condición:	El Usuario debe estar previamente registrado
Flujo Alternativo	No tiene
Post Condición	No tiene

Caso de uso número 16

Caso de Uso:	Eliminar Cliente
Actores:	Administrador
Descripción::	El administrador elimina el cliente
Pre-Condición:	El Cliente debe estar previamente registrado
Flujo Alternativo	No tiene
Post Condición	No tiene

Caso de uso número 17

Caso de Uso:	Buscar Producto
Actores:	Administrador
Descripción:	El administrador busca el producto
Pre-Condición:	El Producto debe estar previamente registrado





Flujo Alternativo	No tiene
Post Condición	No tiene

Caso de uso número 18

Caso de Uso:	Buscar Proveedor
Actores:	Administrador
Descripción:	El administrador busca el proveedor
Pre-Condición:	El Proveedor debe estar previamente registrado
Flujo Alternativo:	No tiene
Post Condición	No tiene

Caso de uso número 19

Caso de Uso:	Buscar Usuario
Actores:	Administrador
Descripción:	El administrador busca al usuario
Pre-Condición:	El Usuario debe estar previamente registrado
Flujo Alternativo	No tiene
Post Condición	No tiene

Caso de uso número 20

Caso de Uso:	Buscar Cliente
Actores:	Administrador
Descripción:	El administrador busca el cliente
Pre-Condición:	El Cliente debe estar previamente registrado
Flujo Alternativo	No tiene
Post Condición	No tiene

Caso de uso número 21

Caso de Uso:	Buscar Factura Venta
Actores:	administrador
Descripción::	El administrador busca la factura venta
Pre-Condición:	La transacción debe estar previamente registrada
Post-Condición:	El administrador selecciona uno de los caso de uso
Flujo Alternativo	No tiene
Post Condición	No tiene



Caso de uso número 22

Caso de Uso:	Buscar Factura Compra
Actores:	Administrador
Descripción:	Administrador busca las facturas compras
Pre-Condición:	La transacción debe estar previamente registrada
Flujo Alternativo	No tiene
Post Condición	No tiene

Caso de uso número 23

Caso de Uso:	Generar Factura Venta
Actores:	Administrador, Usuario Normal
Descripción:	Los Usuarios generan la Factura Venta
Pre-Condición:	La transacción debe haber sido registrada previamente la impresión de la misma
Flujo Alternativo	No tiene
Post Condición	No tiene

Caso de uso número 24

Caso de Uso:	Reporte Factura Venta
Actores:	Administrador
Descripción:	El administrador realiza el reporte factura venta
Pre-Condición:	Las transacciones deben estar registradas
Flujo Alternativo	No tiene
Post Condición	No tiene

Caso de uso número 25

Caso de Uso:	Reporte Factura Compra
Actores:	Administrador
Descripción:	El administrador realiza reporte factura Compra
Pre-Condición:	Las transacciones deben estar registradas
Flujo Alternativo	No tiene
Post Condición	No tiene





Caso de uso número 26

Caso de Uso:	Reporte Cliente
Actores:	Administrador
Descripción:	El administrador realiza reporte cliente
Pre-Condición:	Los Clientes deben estar registrados
Flujo Alternativo	No tiene
Post Condición	No tiene

Caso de uso número 27

Caso de Uso:	Reporte Proveedores
Actores:	Administrador
Descripción:	El administrador realiza reporte de proveedores
Pre-Condición:	Los Proveedores deben estar registrados
Flujo Alternativo	No tiene
Post Condición	No tiene

Caso de uso número 28

Caso de Uso:	Reporte Usuario
Actores:	Administrador
Descripción:	El administrador realiza reporte factura Compra
Pre-Condición:	Los Usuarios deben estar registrados
Flujo Alternativo	No tiene
Post Condición	No tiene

Caso de uso número 29

Caso de Uso:	Reporte Productos
Actores:	Administrador
Descripción:	El administrador realiza reporte productos
Pre-Condición:	Los Productos deben estar registrados
Flujo Alternativo	No tiene
Post Condición	No tiene



Caso de uso número 30

Caso de Uso:	Reporte Productos Faltantes
Actores:	Administrador
Descripción:	El administrador realiza reporte de productos faltantes
Pre-Condición:	El número de existencia de estos productos debe ser cero (0)
Flujo Alternativo	No tiene
Post Condición	No tiene

Caso de uso número 31

Caso de Uso:	Respaldar Base de Datos
Actores:	Administrador
Descripción:	El administrador realiza el respaldo de la base de datos
Pre-Condición:	El administrador debe acceder al sistema mediante su login de servidor
Flujo Alternativo	No tiene
Post Condición	No tiene

Caso de uso número 32

Caso de Uso:	Restaurar Base de Datos
Actores:	Administrador
Descripción:	El administrador restaura la basa de datos
Pre-Condición:	El administrador debe haber creado un respaldo previamente de la Base de Datos
Flujo Alternativo	No tiene
Post Condición	No tiene



6. DIAGRAMA DE CLASES

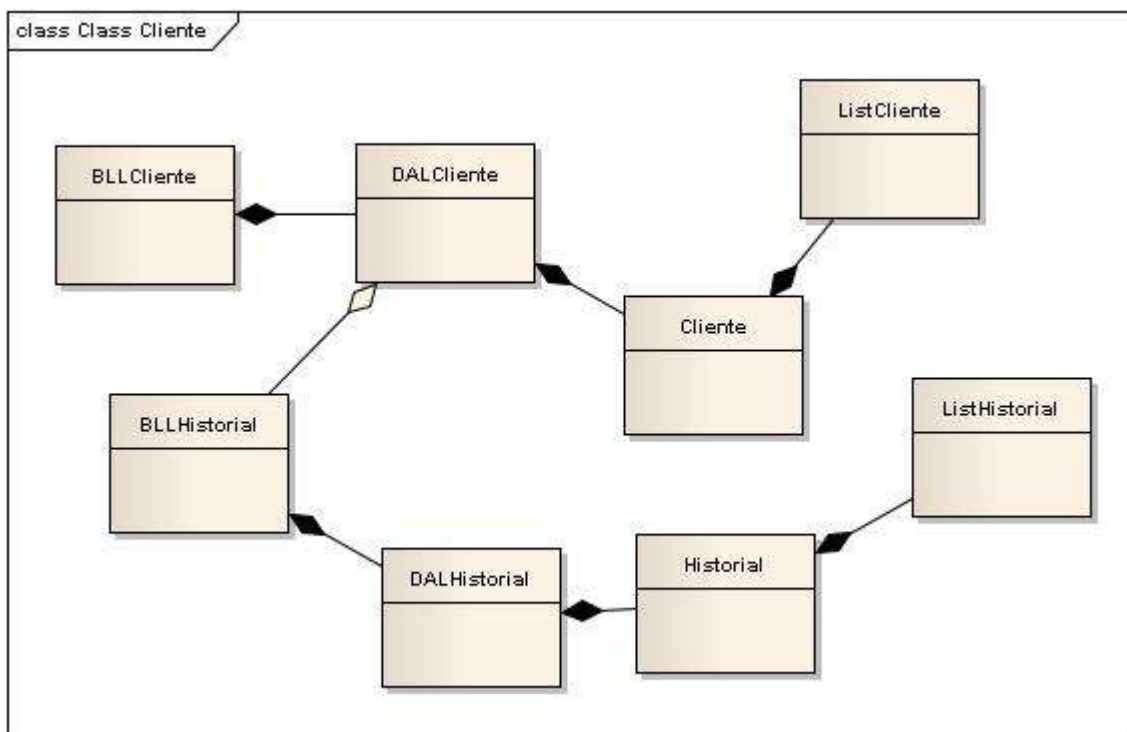


Figura No.13 Diagrama de Clases Cliente

Clases encargadas de la manipulación de la información de los clientes. A continuación explicamos la utilidad de algunas de ellas:

DALCliente: Este se encarga del control de la inserción, eliminación, modificación y visualización de los datos a cada una de las tablas a través de procedimientos almacenados mediante los cuales accede a cada una de las tablas para la debida inserción de los datos, además esta clase utiliza transacciones para la debida eliminación, modificación y visualización, si ocurre un error no se realiza la operación.

Cliente: Esta clase se encarga de la asignación de los datos obtenidos por la clase **DALCliente** que es la que le envía los datos para su debida manipulación por la clase **ListCliente** para la inserción con el procedimiento que le corresponde a dicha clase.

BLLCliente: Esta clase que es la capa de presentación que es la que se encarga de enlazar las demás clases para la debida utilización y manipulación de los datos.



ListHistorial: Es una lista que almacena los datos de la información de haberlo obtenidos por la clase **DALCliente** este se utiliza para la manipulación de los datos y luego ser insertado con su procedimiento respectivo de la clase.

DALHistorial: Se encarga del control de la inserción y visualización de los datos a cada una de las tablas a través de procedimientos almacenados, esta clase utiliza transacciones para la debida inserción y visualización, además si ocurre un error no se realiza la operación.

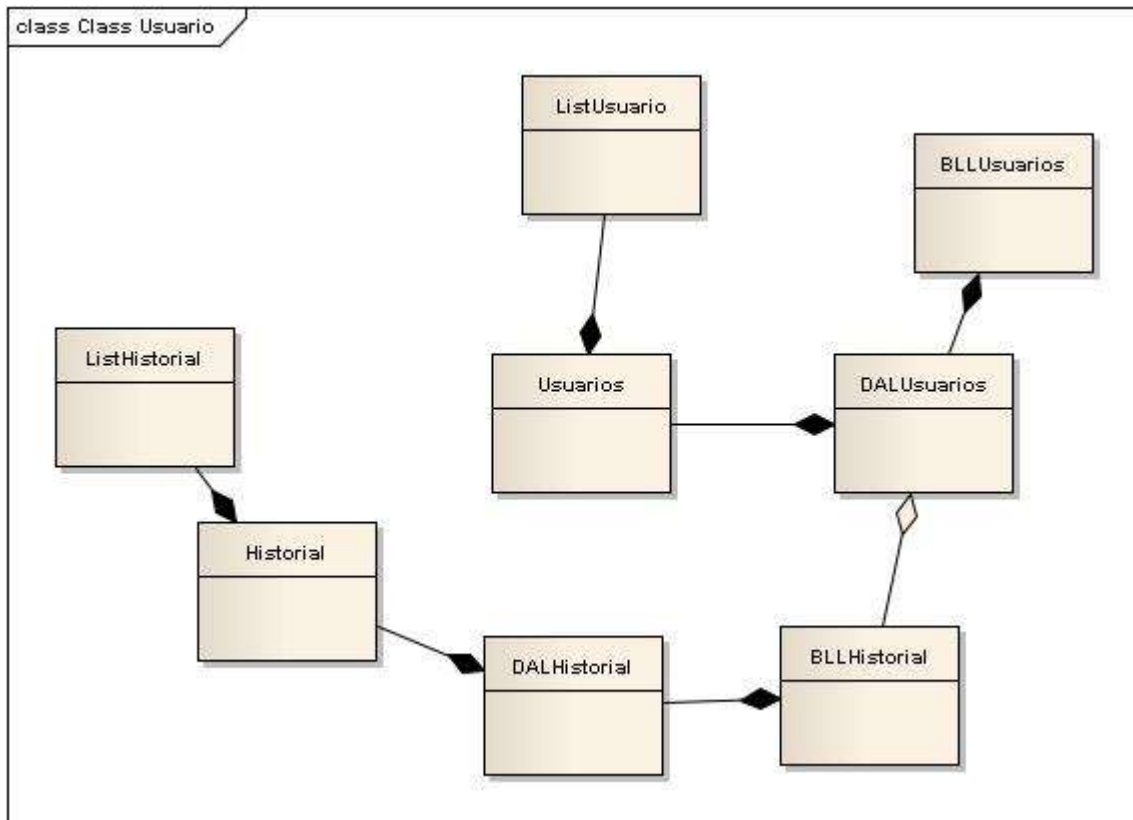


Figura No.14 Diagrama de Clases Usuario

Estas son todas las clases que se encargan de la manipulación de la información de los Usuarios.

DALUsuario: En esta clase se realiza la captura de todos los usuario que van a poder acceder a la aplicación en esta se encarga de también de asignarle el rol que tendrá el usuario que va a acceder a la aplicación y a cada módulo que tiene acceso para su manipulación.

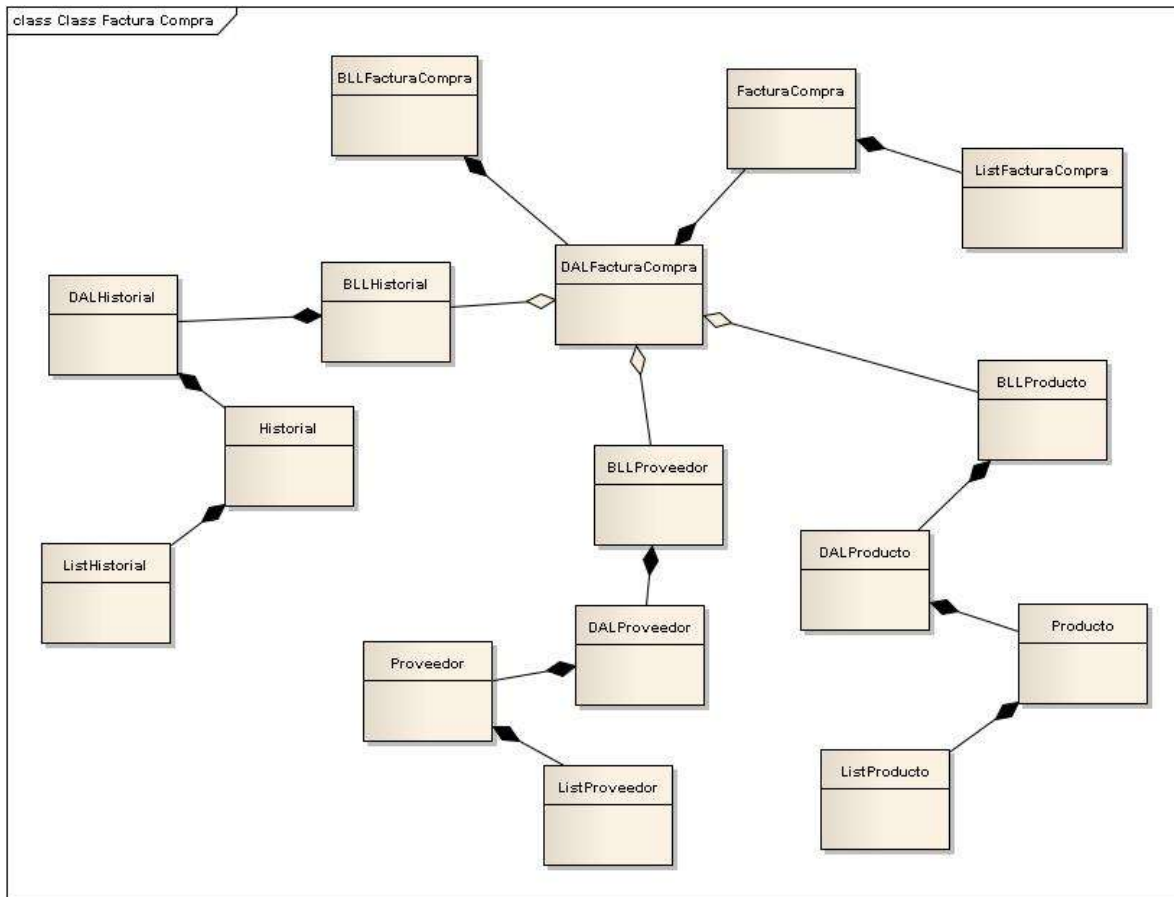


Figura No.15 Diagrama de Clases FacturaCompra

Estas son todas las clases que se encargan de la manipulación de la información de la FacturaCompra. A continuación explicamos la utilidad de algunas de ellas:

DALProducto: Esta clase es encargada del registro, visualización y actualizaciones de los productos con ayuda de las clases para la debida creación de los productos y sus facturas.

FacturaCompra: La clase se encarga de la obtención de los datos de las respectivas Facturas de Compras al proveedor para su debida utilización de ellas.

BLLProveedores: Esta clase obtiene los datos relacionados de los proveedores para su debida asignación a sus respectivas facturas.

ListHistorial: Se realiza la manipulación de los datos referente al Historial de la aplicación.

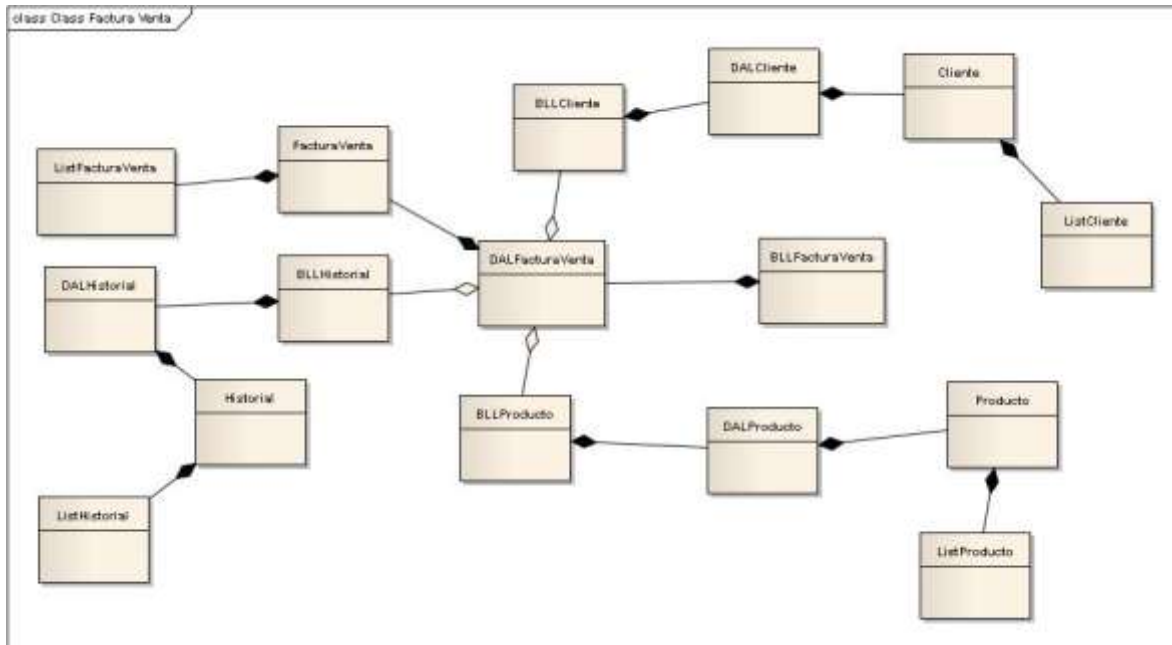


Figura No.16 Diagrama de Clases FacturaVenta

Estas son todas las clases que se encargan de la manipulación de la información de las Facturas de Venta al Cliente. A continuación explicamos la utilidad de algunas de ellas:

DALFacturaCompra: Esta clase se encarga de realizar las facturas a los clientes y asignarles los productos a razón del pedido realizado por ellos (clientes).

FacturaVenta: Con esta realizamos la debida inserción de las facturas a los clientes para su procedimiento y se le asigna el id factura automáticamente a cada registro realizado.

BBLCliente: En esta clase se realiza la operación de la extracción de los clientes debido a la necesidad de obtener distintos datos de estos al realizar la factura.



7. DIAGRAMAS DE SECUENCIA

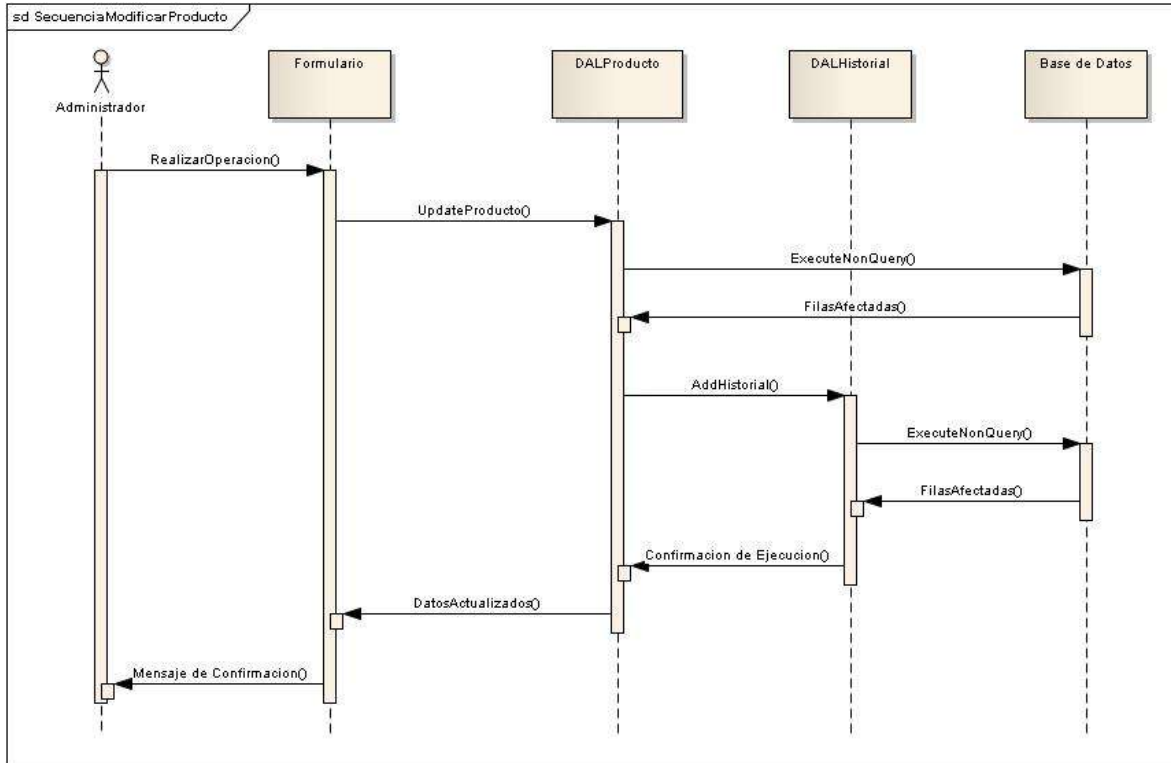


Figura No.18 Diagrama de Secuencia Modificar Producto

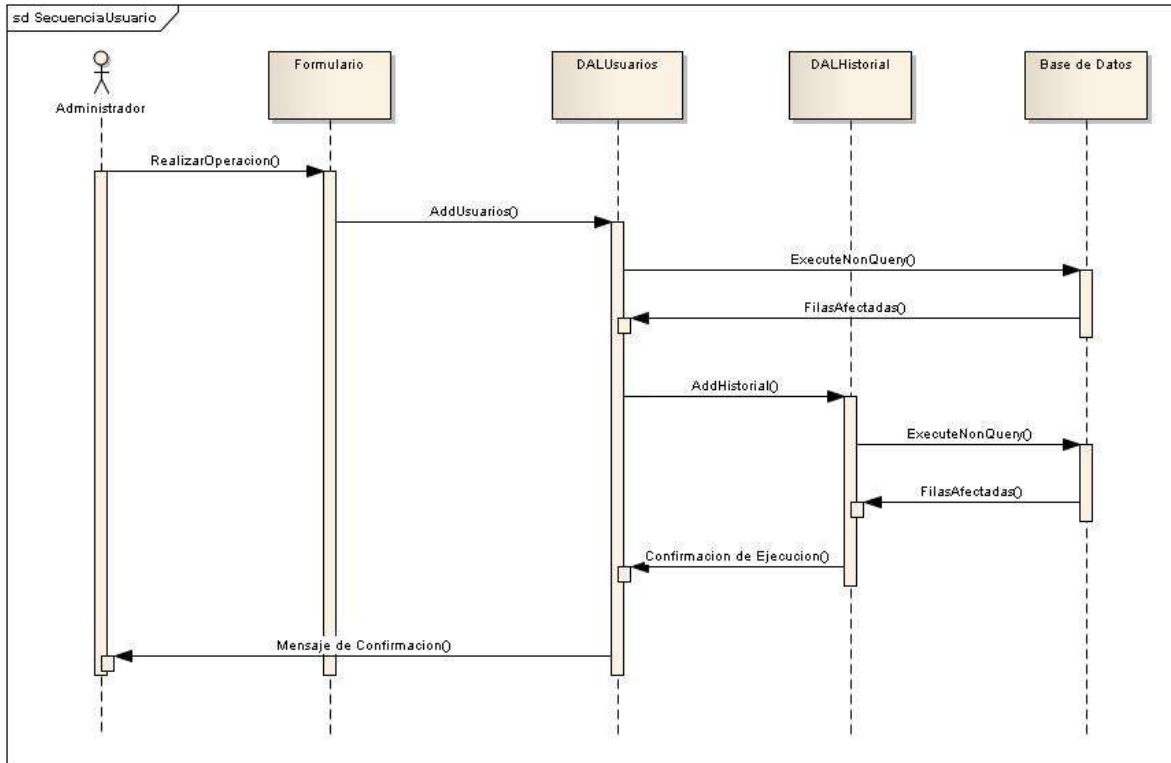


Figura No.19 Diagrama de Secuencia Agregar Usuario

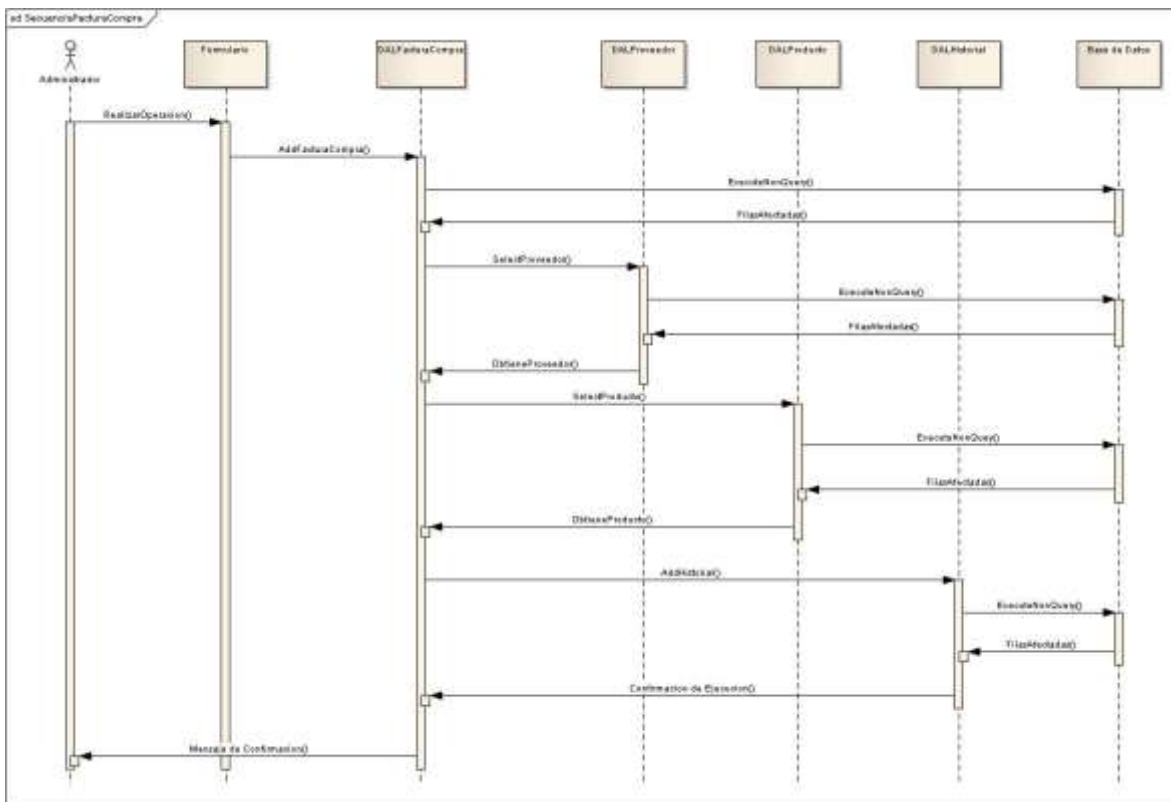


Figura No.20 Diagrama de Secuencia Agregar FacturaCompra

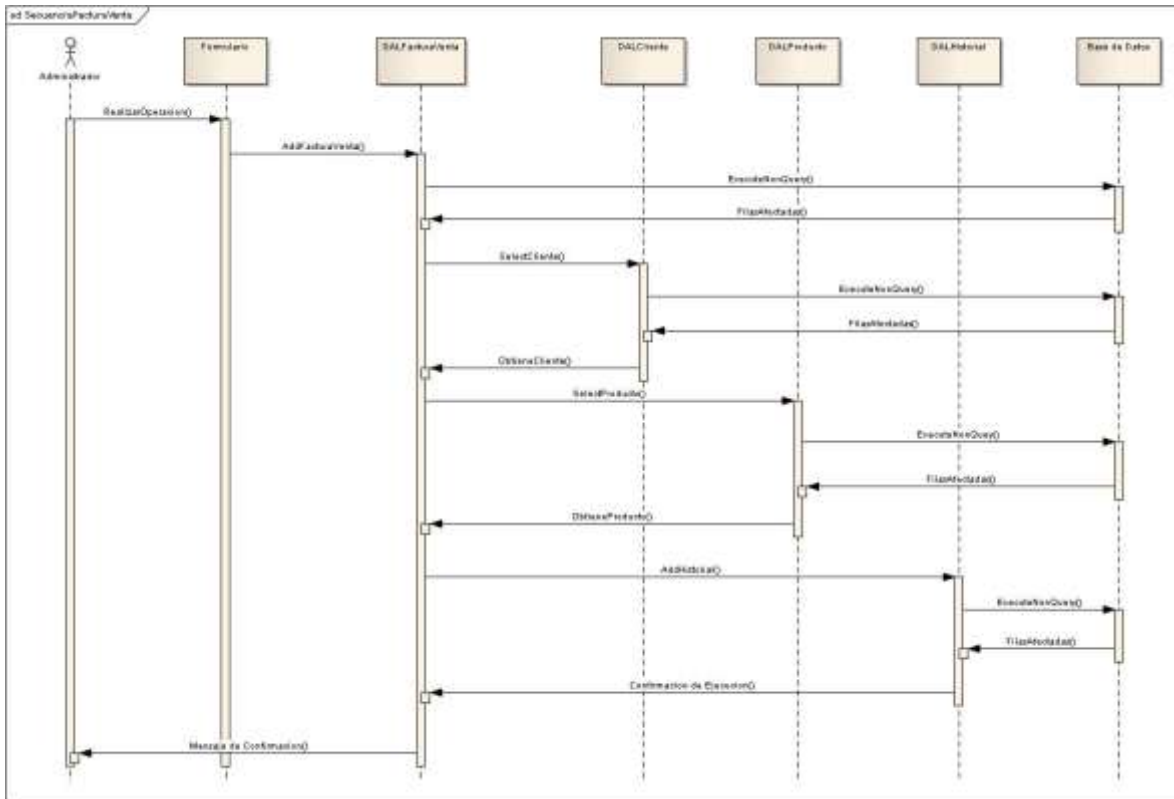


Figura No.21 Diagrama de Secuencia Agregar FacturaVenta

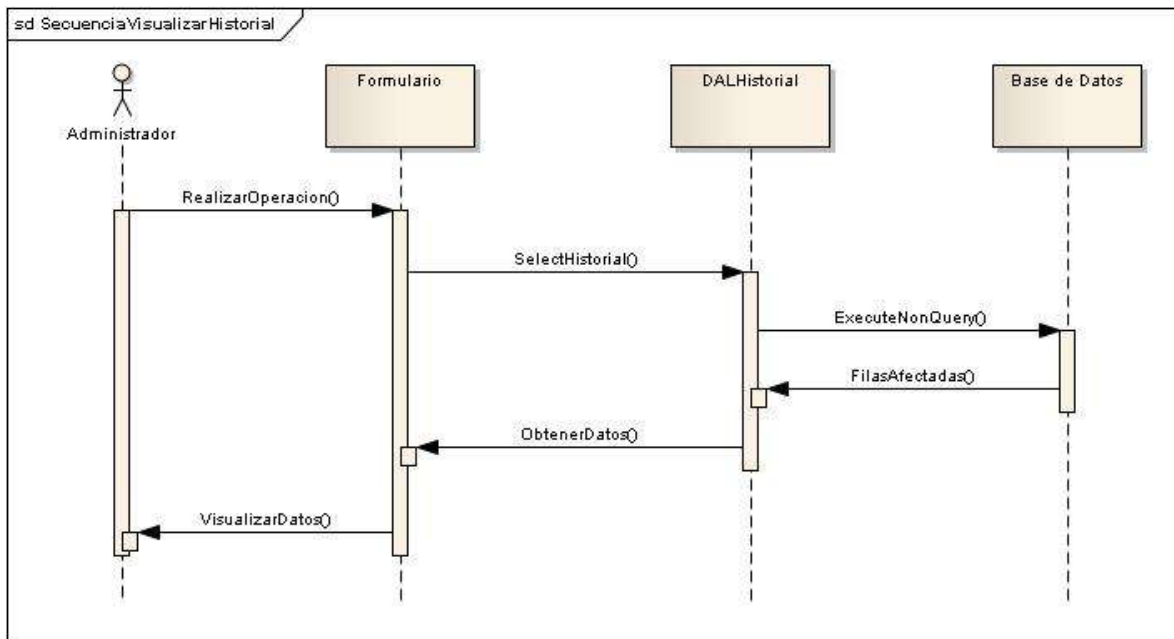


Figura No.22 Diagrama de Secuencia Visualizar Historial

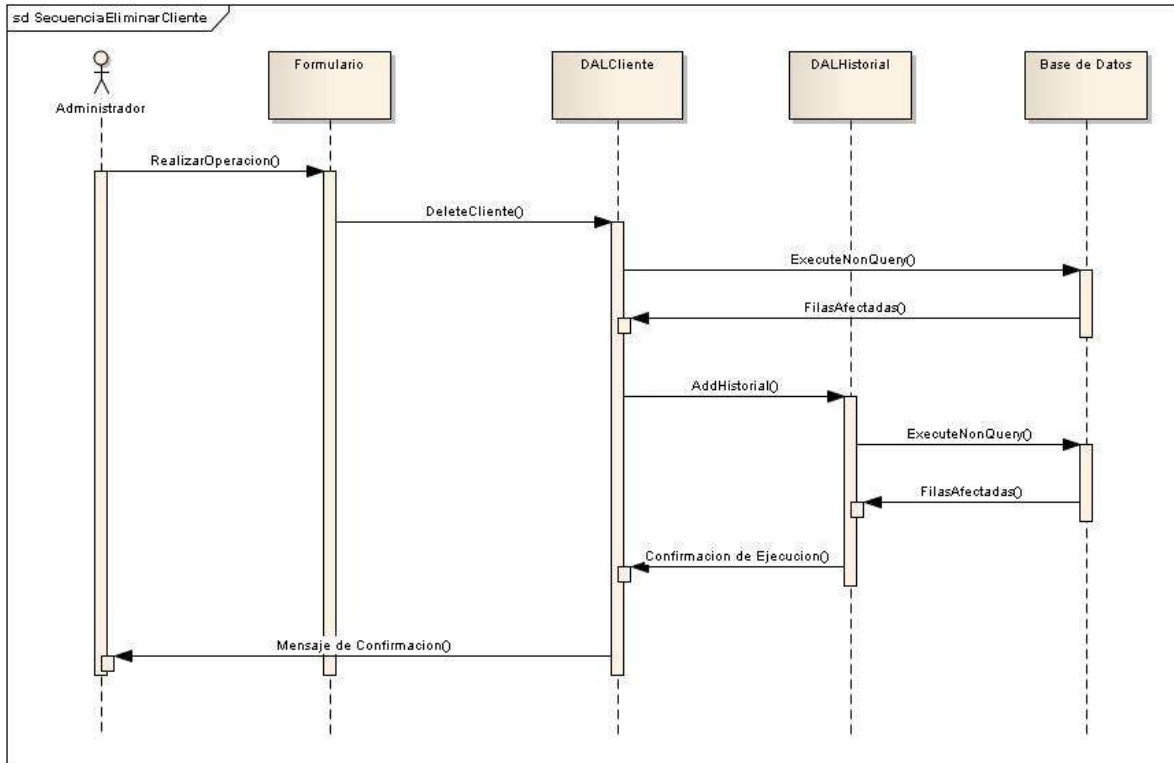


Figura No.23 Diagrama de Secuencia Eliminar Cliente

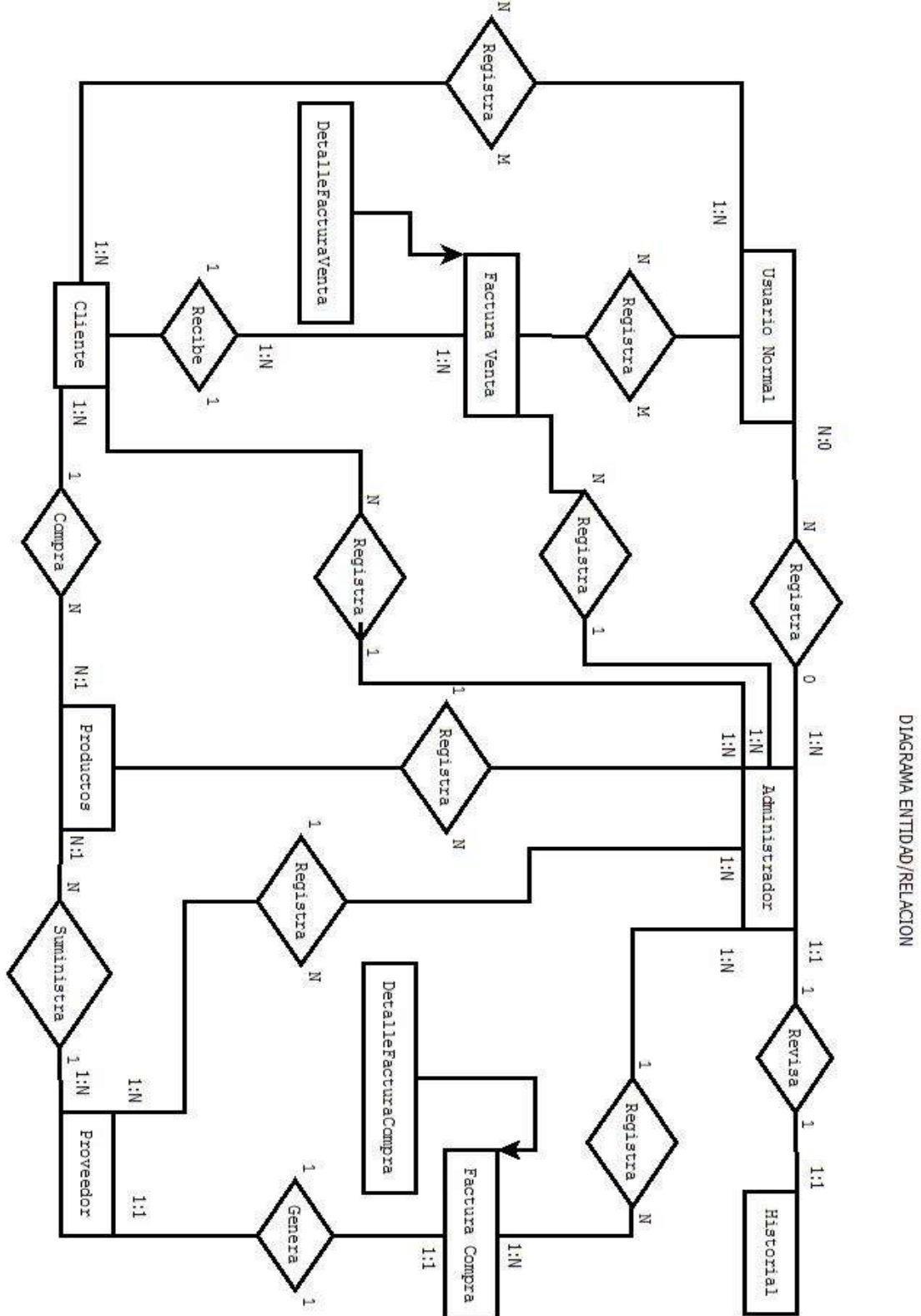


Figura No.24 Diagrama Entidad Relación



8. DISEÑO DE DATOS



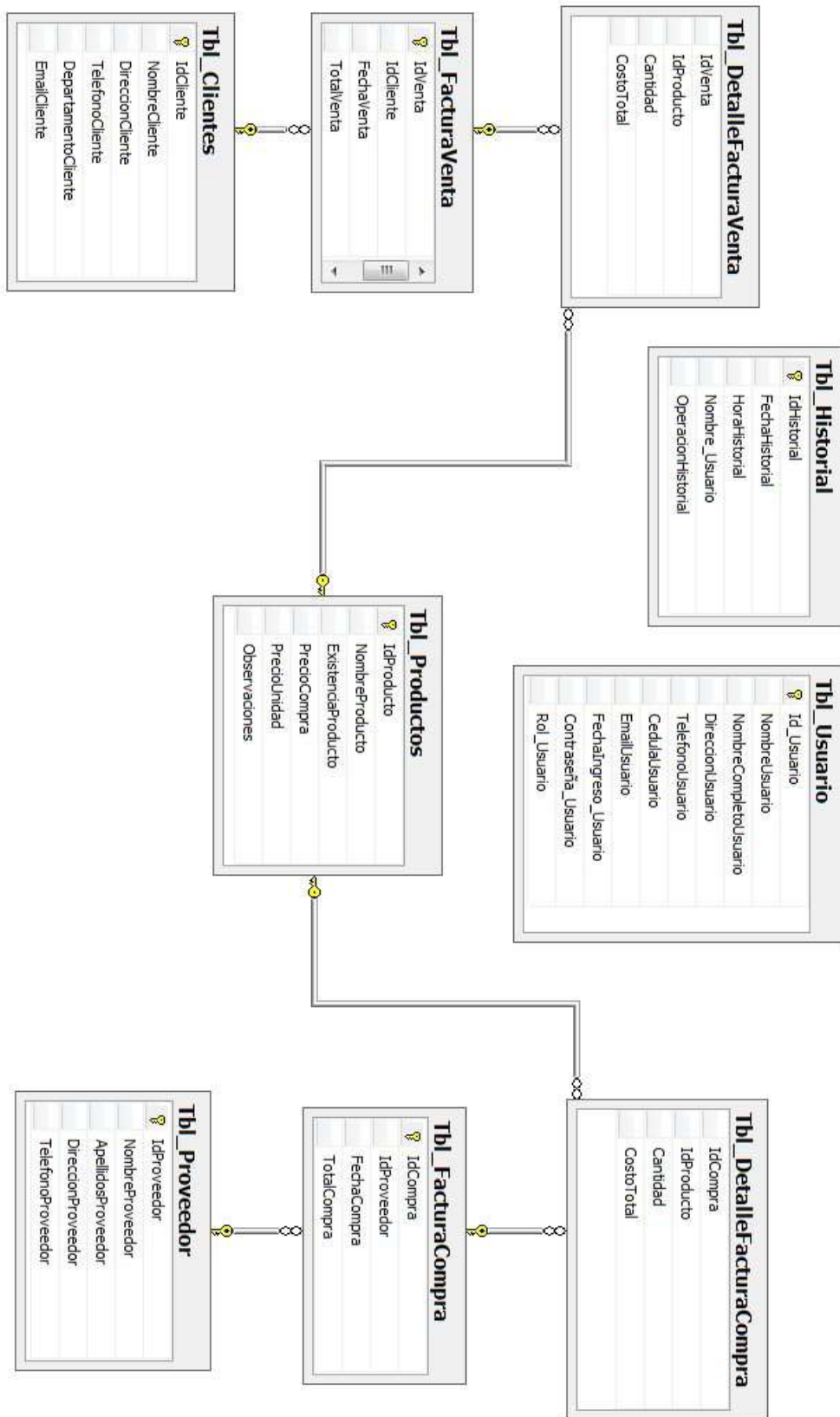


Figura No.25 Diagrama de Entidades



Tabla Usuario

Almacena todos los datos de los Usuarios que se encuentran registrados en Librería Guadalupe Corinto.

Nombre	Descripción	Tipo
Id_Usuario	Llave primaria	Int
NombreUsuario	Nombre de Login	Varchar(20)
NombreCompletoUsuario	Nombre Completo	Varchar(100)
DireccionUsuario	Dirección	Varchar(150)
TelefonoUsuario	Teléfono	Varchar(20)
CedulaUsuario	Cedula	Varchar(30)
EmailUsuario	Correo Electrónico	varchar(60)
FechaIngreso_Usuario	Fecha de ingreso del Usuario	Datetime
Contraseña_Usuario	Contraseña de Login	varchar(70)
Rol_Usuario	Tipo de Usuario	varchar(20)

Tabla Cliente

Almacena todos los datos de los Clientes que se encuentran registrados en Librería Guadalupe Corinto.

Nombre	Descripción	Tipo
IdCliente	Llave primaria	Int
NombreCliente	Nombre del Cliente	varchar(100)
DireccionCliente	Dirección	varchar(50)
TelefonoCliente	Teléfono	varchar(50)
DepartamentoCliente	Departamento	varchar(100)
EmailCliente	Correo Electrónico	varchar(60)

Tabla Productos

Almacena todos los datos de los Productos que se encuentran registrados en Librería Guadalupe Corinto.

Nombre	Descripción	Tipo
IdProducto	Llave primaria	Int
NombreProducto	Nombre del Producto	varchar(50)
ExistenciaProducto	Existencia	Int
PrecioCompra	Precio de Compra	Float
PrecioUnidad	Precio por Unidad	Float
Observaciones	Estante Ubicado	varchar(100)





Tabla Proveedor

Almacena todos los datos de los Proveedores que se encuentran registrados en Librería Guadalupe Corinto.

Nombre	Descripción	Tipo
IdProveedor	Llave primaria	Int
NombreProveedor	Nombre del Proveedor	varchar(50)
EncargadoProveedor	Nombre del responsable de la Empresa Proveedora	varchar(50)
DireccionProveedor	Dirección	varchar(50)
TelefonoProveedor	Telefono	varchar(10)

Tabla Factura Compra

Registra los datos de las Facturas recibidas por el proveedor que se encuentran registrados en Librería Guadalupe Corinto.

Nombre	Descripción	Tipo
IdCompra	Llave primaria	Int
IdProveedor	Nombre del Proveedor (Llave Foránea)	Int
FechaCompra	Fecha en que se realizó la Compra	Datetime
TotalCompra	Monto Total de la Factura	Float

Tabla Detalle Factura Compra

Registra los productos y las cantidades recibidas por el proveedor en cada Factura Compra que se encuentran registrados en Librería Guadalupe Corinto.

Nombre	Descripción	Tipo
IdCompra	Id Factura Compra	Int
IdProducto	Nombre del Producto (Llave Foránea)	Int
Cantidad	Cantidad del Producto	Int
CostoTotal	Costo Total por producto	Float

Tabla Factura Venta

Registra los datos de las Facturas realizadas a un cliente que se encuentran registrados en Librería Guadalupe Corinto.





Nombre	Descripción	Tipo
IdVenta	Llave primaria	Int
IdCliente	Nombre del Cliente (Llave Foránea)	Int
FechaVenta	Fecha en que se realizó la Venta	Datetime
TotalVenta	Monto Total de la Factura	Float

Tabla Detalle Factura Venta

Registra los productos y las cantidades vendidas a un cliente en cada Factura Venta que se encuentran registrados en Librería Guadalupe Corinto.

Nombre	Descripción	Tipo
IdVenta	Id Factura Venta	Int
IdProducto	Nombre del Producto (Llave Foránea)	Int
Cantidad	Cantidad del Producto	Int
CostoTotal	Costo Total por producto	Float

Tabla Historial

Registra las operaciones realizadas por un Usuario en el Sistema de Librería Guadalupe Corinto.

Nombre	Descripción	Tipo
IdHistorial	Llave primaria	Int
FechaHistorial	Fecha en que fue realizada la operación	Date
HoraHistorial	Hora en que fue realizada la operación	time(7)
Nombre_Usuario	Nombre del Usuario que realizo la operación	varchar(20)
OperacionHistorial	Operación Realizada	varchar(100)



9. DISEÑO ARQUITECTÓNICO

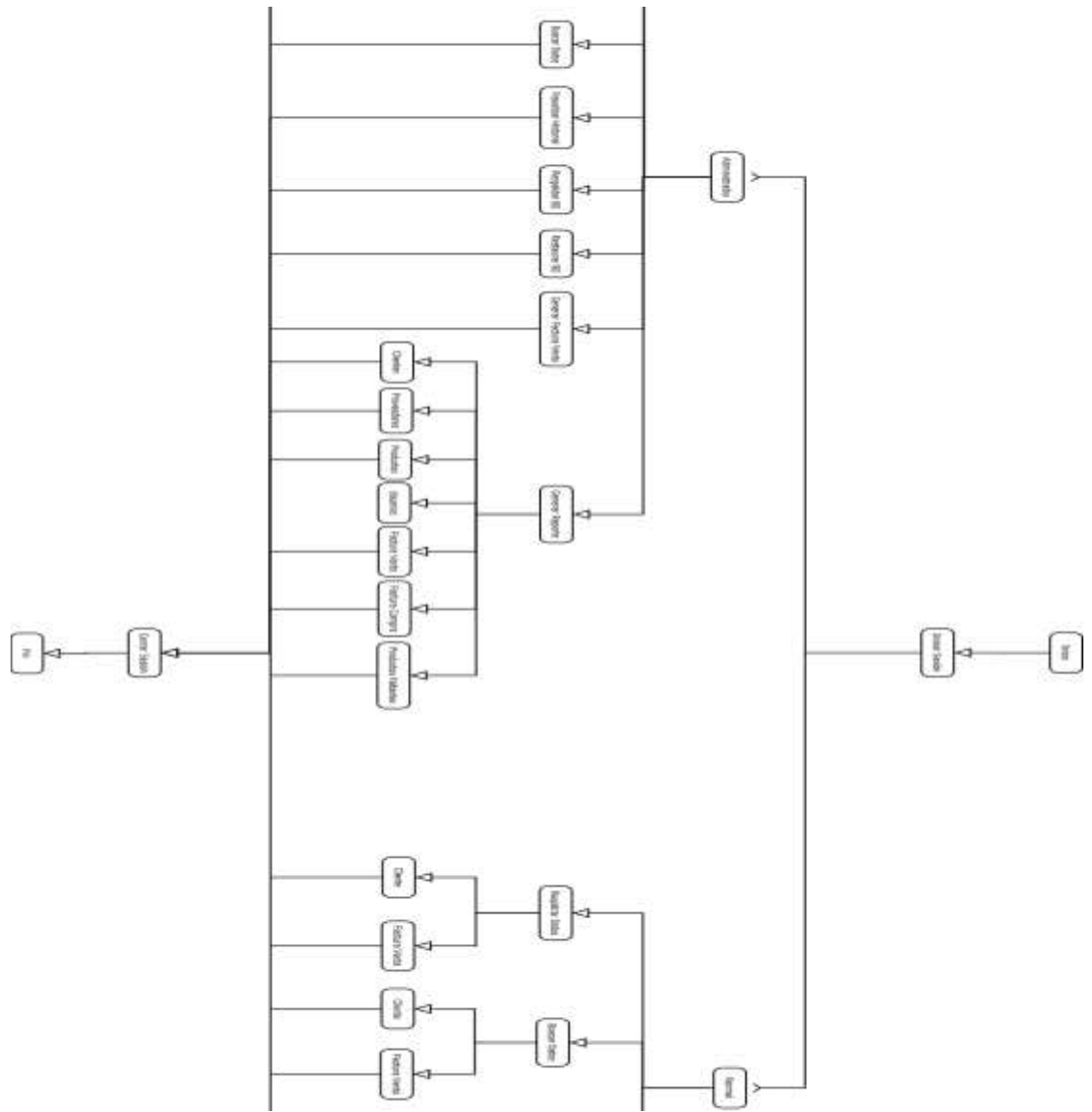


Figura No.26 Diseño Arquitectónico



10. DISEÑO DE INTERFACES



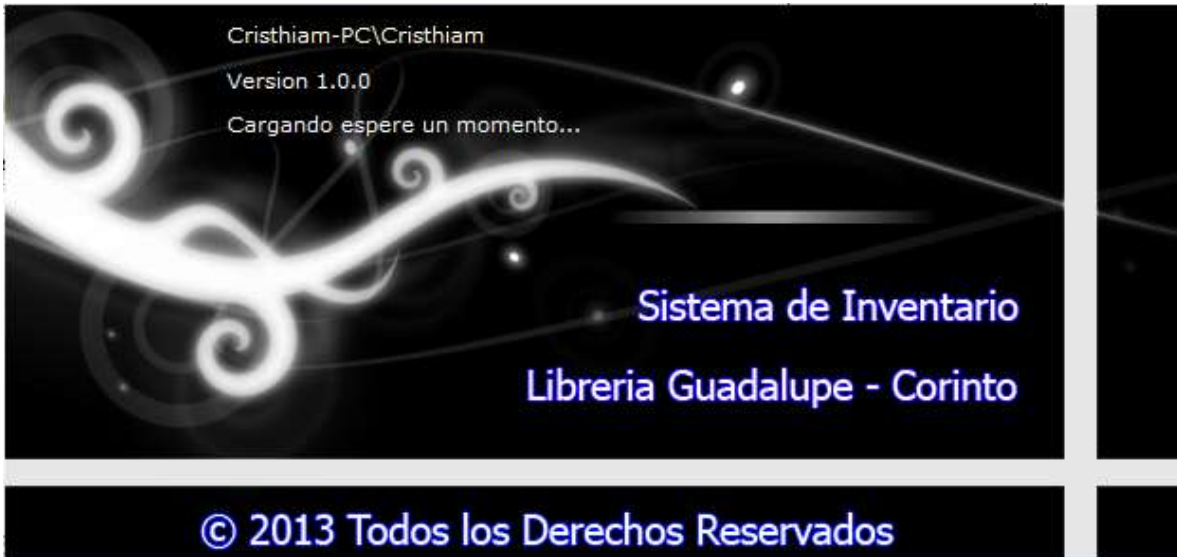


Figura No.27 Inicio de la Aplicación

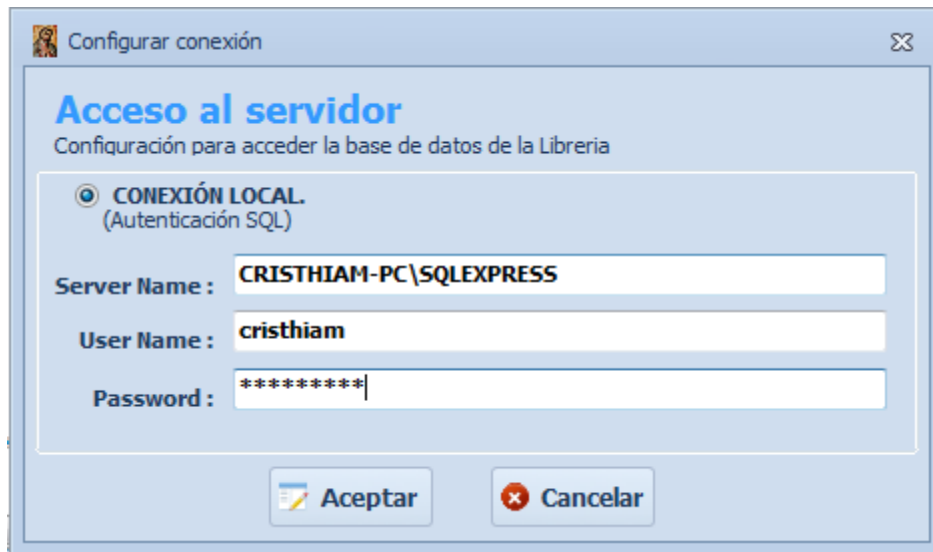


Figura No.28 Interfaz de Configuración del Servidor



Figura No.29 Interfaz de Inicio de Sesión

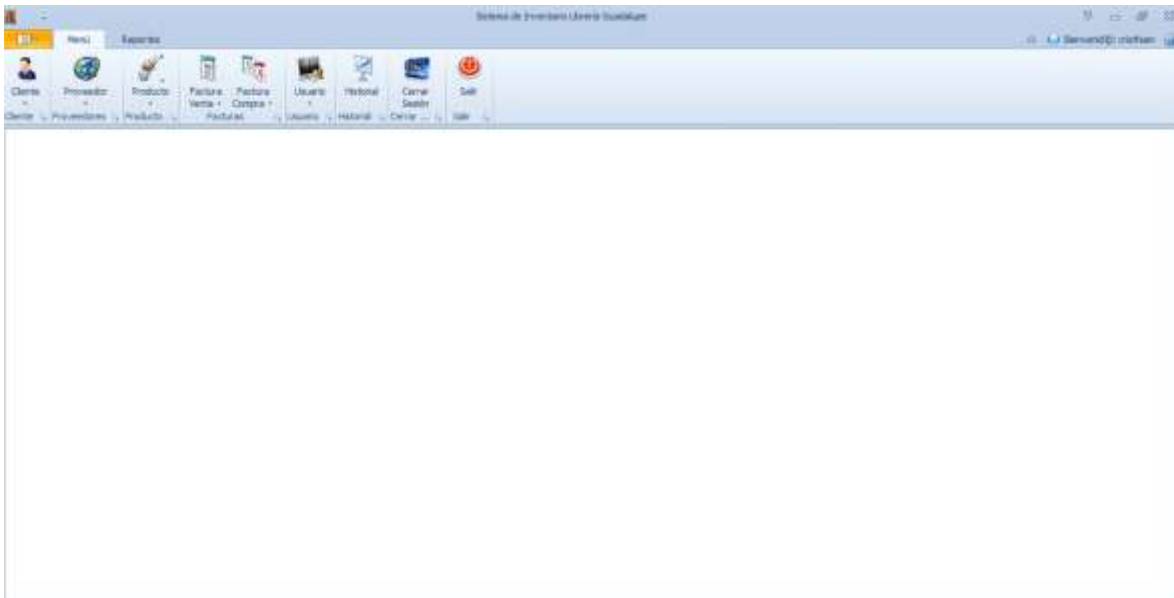


Figura No.30 Interfaz Principal del Programa (Menú)

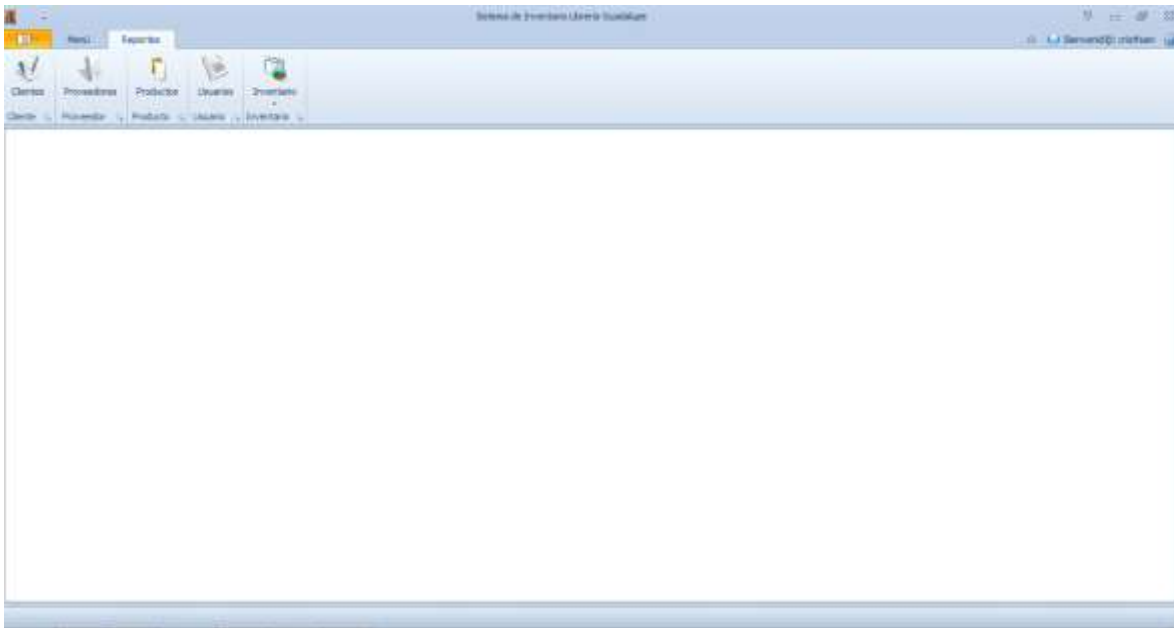


Figura No.31 Interfaz Principal del Programa (Reportes)

Registrar Cliente

Nombre Completo:

Dirección:

Teléfono:

Departamento:

Email:

Guardar Cancelar

Figura No.32 Interfaz Registrar Cliente



Registrar Usuario

Nombre de Usuario:

Nombre Completo:

Dirección:

Teléfono: Cédula:

Email: Contraseña:

Fecha de Ingreso: Rol Usuario:

Figura No.33 Interfaz Registrar Usuario

Registrar Factura Compra

Proveedor: Id Factura:

Producto: Cantidad:

Producto	Existencia	Cantidad	Precio Unidad	Total
----------	------------	----------	---------------	-------

Total Compra:

Figura No.34 Interfaz Registrar Factura Compra

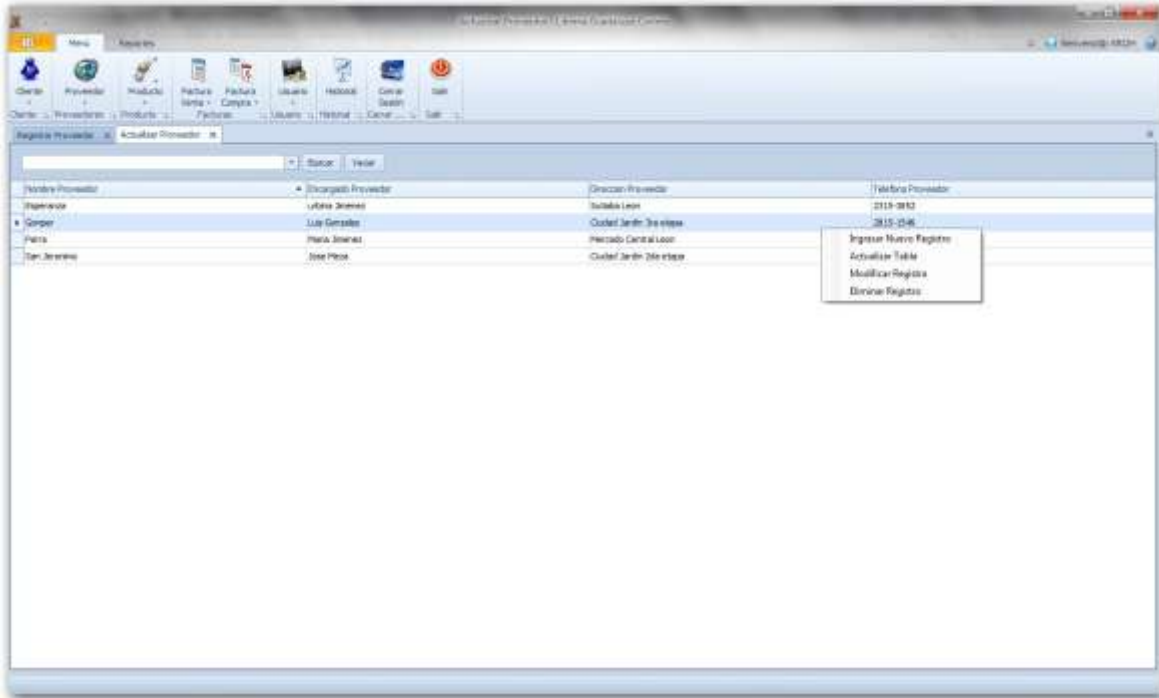


Figura No.35 Interfaz Actualizar Proveedor

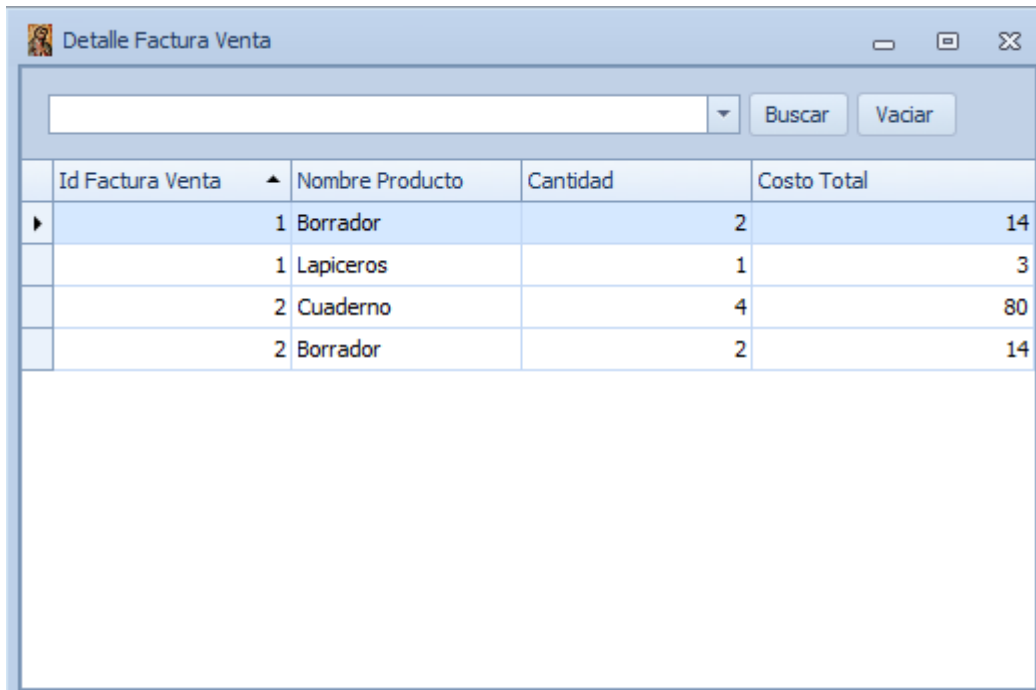


Figura No.36 Interfaz Detalle Factura Venta



Modificar Usuario

Nombre de Usuario:

Nombre Completo:

Dirección:

Teléfono: Cédula:

Email: Contraseña:

Fecha de Ingreso: Rol Usuario:

Figura No.37 Interfaz Modificar Usuario

Modificar Producto

Nombre Producto:

Existencia Producto:

Precio Compra:

Precio Unidad:

Observaciones:

Figura No.38 Interfaz Modificar Producto

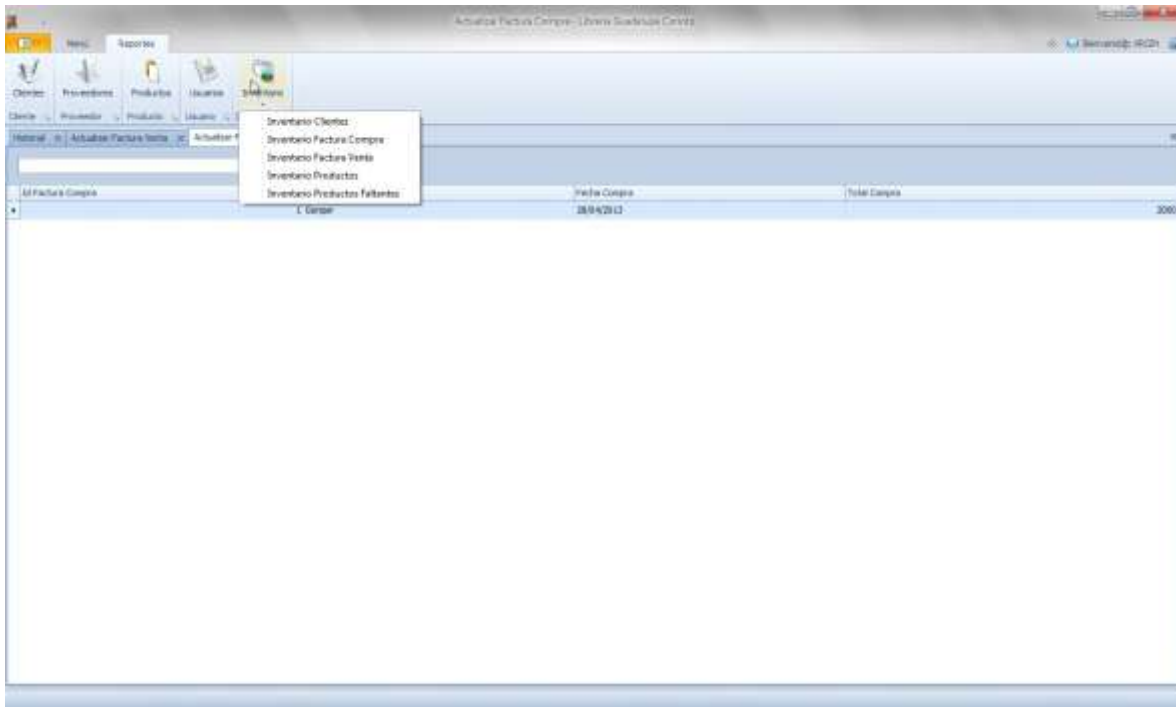


Figura No.39 Interfaz Reporte Inventario

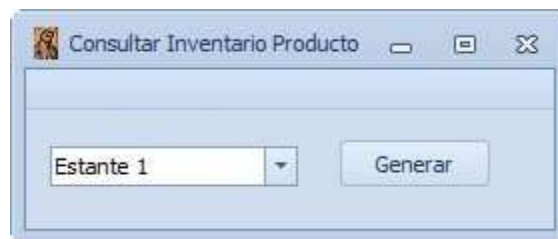


Figura No.40 Interfaz Consultar Inventario Producto

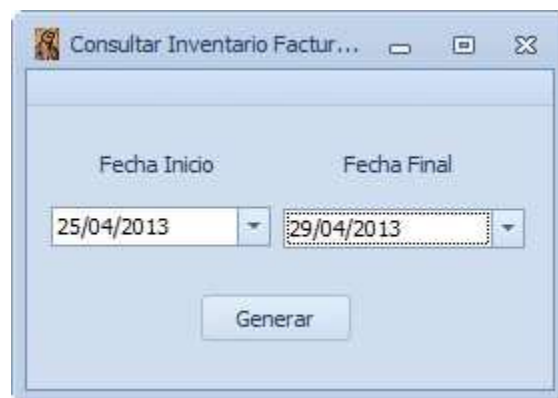


Figura No.41 Interfaz Consultar Inventario FacturaVenta



11. REQUERIMIENTOS DEL SISTEMA

❖ Requerimiento Hardware

Procesador Dual-Core.

Disco Duro de 250GB.

Memoria Ram 2GB.

❖ Requerimiento Software

Windows XP, Windows 7

Microsoft SQL Server 2008.

.Net Framework 3.5





12. CONCLUSIONES

En conclusión se ha obtenido un sistema que permitirá minimizar el tiempo de respuesta para la generación de reportes y la automatización de todas las operaciones de la Librería Guadalupe Corinto a través de los módulos creados para agilizar la manipulación de los datos de todos los productos, clientes y proveedores, logrando que la manipulación de la información sea fácil y rápida de utilizar.

Por ende podemos afirmar que se ha logrado alcanzar el objetivo de nuestro trabajo al proveer a la Librería Guadalupe Corinto de un software capaz de solventar los problemas que se generaban en él y a nosotros como desarrolladores nos permitió aplicar los conocimientos adquiridos a lo largo de nuestra formación como Ingenieros en Sistemas de Información.





13. RECOMENDACIONES

Terminado nuestro trabajo consideramos los siguientes aspectos interesantes para la continuación y mejora de esta aplicación.

- ❖ Desarrollar una aplicación web de la Librería para que los clientes y futuros clientes puedan ver los productos que se ofrecen, promociones, etc.
- ❖ Desarrollar una aplicación en red para la Librería Guadalupe Corinto y la revisión de los productos ofrecidos en dicho lugar.
- ❖ Realizar un blog para mantener informado a los clientes sobre las actividades de la Librería.





14. BIBLIOGRAFÍA

- <http://documentation.devexpress.com/>. (03 de 03 de 2013).
<http://documentation.devexpress.com/>.
- http://es.wikipedia.org/wiki/Microsoft_.NET. (26 de 03 de 2013).
http://es.wikipedia.org/wiki/Microsoft_.NET.
- <http://mspnor.wordpress.com/2008/10/31/sqlprocedimientos-almacenados-paso-a-paso/>. (26 de 03 de 2013).
<http://mspnor.wordpress.com/2008/10/31/sqlprocedimientos-almacenados-paso-a-paso/>.
- <http://www.microsoft.com/net>. (26 de 03 de 2013). *<http://www.microsoft.com/net>*.
- Microsoft. (05 de 04 de 2013). *[http://msdn.microsoft.com/es-es/library/bb385795\(v=vs.90\).aspx](http://msdn.microsoft.com/es-es/library/bb385795(v=vs.90).aspx)*.
- microsoft. (06 de 04 de 2013). *[http://msdn.microsoft.com/es-es/library/ms190782\(v=sql.90\).aspx](http://msdn.microsoft.com/es-es/library/ms190782(v=sql.90).aspx)*.
- Microsoft. (s.f.). *LINQ to ADO.NET*. Obtenido de ms-help://MS.VSCC.v90/MS.MSDNQTR.v90.es/wd_linqadonet/html/be3297b9-1b54-4d4c-82a8-add0d79c2006.htm
- Microsoft,msdn. (26 de 03 de 2013). *[http://msdn.microsoft.com/en-us/library/zw4w595w\(v=vs.90\).aspx](http://msdn.microsoft.com/en-us/library/zw4w595w(v=vs.90).aspx)*. Obtenido de [http://msdn.microsoft.com/en-us/library/zw4w595w\(v=vs.90\).aspx](http://msdn.microsoft.com/en-us/library/zw4w595w(v=vs.90).aspx)
- Microsoft.com. (s.f.). *[http://msdn.microsoft.com/es-es/library/52f3sw5c\(v=vs.90\).aspx](http://msdn.microsoft.com/es-es/library/52f3sw5c(v=vs.90).aspx)*.
- Server, M. S. (12 de 03 de 2013).
<http://www.microsoft.com/spain/sql/productinfo/features/top30features.msp>.
- wikipedia. (10 de 03 de 2013). *http://es.wikipedia.org/wiki/Desarrollo_en_espiral*.





15. ANEXOS





Glosario

- ❖ **CommonLanguageRuntime (CLR):** "Entorno en tiempo de ejecución de lenguaje común" es un entorno de ejecución para los códigos de los programas que corren sobre la plataforma Microsoft .NET. El CLR es el encargado de compilar una forma de código intermedio llamada CommonIntermediateLanguage (CIL, anteriormente conocido como MSIL, por Microsoft IntermediateLanguage), al código de máquina nativo, mediante un compilador en tiempo de ejecución. No debe confundirse el CLR con una máquina virtual, ya que una vez que el código está compilado, corre nativamente sin intervención de una capa de abstracción sobre el hardware subyacente.

- ❖ **Interfaz:** La **interfaz de usuario** es el medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo. Normalmente suelen ser fáciles de entender y fáciles de accionar.

- ❖ **Framework:** Un framework o infraestructura digital, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.





DALUsuarios

```
publicstaticbool delete(string id)
{
    Comando = newSqlCommand { Connection = Conexion, CommandText =
"Eliminarusuario", CommandType = CommandType.StoredProcedure };
    Comando.Parameters.Add(newSqlParameter("@id", SqlDbType.Int)).Value = id;
returncgenerica.RealizarOperacion(Comando);
}

publicstaticbool Add(CrearUsuario user)
{
    Comando = newSqlCommand { CommandText = "crearusuario", CommandType =
CommandType.StoredProcedure };
    Comando.Parameters.Add(newSqlParameter("@nombreusuario",
SqlDbType.VarChar)).Value = user.nombre_usuario;
    Comando.Parameters.Add(newSqlParameter("@nombrecompletousuario",
SqlDbType.VarChar)).Value = user.nombre_completo_usuario;
    Comando.Parameters.Add(newSqlParameter("@direccionusuario",
SqlDbType.VarChar)).Value = user.direccion_usuario;
    Comando.Parameters.Add(newSqlParameter("@telefonousuario",
SqlDbType.VarChar)).Value = user.telefono_usuario;
    Comando.Parameters.Add(newSqlParameter("@cedulausuario",
SqlDbType.VarChar)).Value = user.cedula_usuario;
    Comando.Parameters.Add(newSqlParameter("@emailusuario",
SqlDbType.VarChar)).Value = user.email_usuario;
    Comando.Parameters.Add(newSqlParameter("@fechaingresausuario",
SqlDbType.DateTime)).Value = user.fechaingresa_usuario;
    Comando.Parameters.Add(newSqlParameter("@contraseñausuario",
SqlDbType.VarChar)).Value = user.contraseña_usuario;
    Comando.Parameters.Add(newSqlParameter("@rolusuario",
SqlDbType.VarChar)).Value = user.rol_usuario;

returncgenerica.RealizarOperacion(Comando);
}

publicstaticbool Update(CrearUsuario user)
{
    Comando = newSqlCommand {CommandText = "Modificarusuario",
CommandType = CommandType.StoredProcedure };
    Comando.Parameters.Add(newSqlParameter("@idusuario", SqlDbType.Int)).Value
= user.id_usuario;
    Comando.Parameters.Add(newSqlParameter("@nombreusuario",
SqlDbType.VarChar)).Value = user.nombre_usuario;
```





```
Comando.Parameters.Add(newSqlParameter("@nombrecompletousuario",
SqlCommandType.VarChar)).Value = user.nombre_completo_usuario;
Comando.Parameters.Add(newSqlParameter("@direccionusuario",
SqlCommandType.VarChar)).Value = user.direccion_usuario;
Comando.Parameters.Add(newSqlParameter("@telefonousuario",
SqlCommandType.VarChar)).Value = user.telefono_usuario;
Comando.Parameters.Add(newSqlParameter("@cedulausuario",
SqlCommandType.VarChar)).Value = user.cedula_usuario;
Comando.Parameters.Add(newSqlParameter("@emailusuario",
SqlCommandType.VarChar)).Value = user.email_usuario;
Comando.Parameters.Add(newSqlParameter("@fechaingresousuario",
SqlCommandType.DateTime)).Value = user.fechaingresa_usuario;
Comando.Parameters.Add(newSqlParameter("@contraseñausuario",
SqlCommandType.VarChar)).Value = user.contraseña_usuario;
Comando.Parameters.Add(newSqlParameter("@rolusuario",
SqlCommandType.VarChar)).Value = user.rol_usuario;

return generica.RealizarOperacion(Comando);
}
```

DALFacturaVenta

```
public static bool AddDetalle(ListFacturaVenta fventa)
{
    Conexion = ConexionSQL.GetConexion();
    Conexion.Open();
    trans = Conexion.BeginTransaction();
    try
    {
        for (int i = 0; i < fventa.Count; i++)
        {
            Comando = new SqlCommand { CommandText = "guardardetallefacturaventa",
            CommandType = CommandType.StoredProcedure };
            Comando.Parameters.Add(new SqlParameter("@idventa",
            SqlDbType.VarChar)).Value = fventa[i].id_venta;
            Comando.Parameters.Add(new SqlParameter("@idproducto",
            SqlDbType.VarChar)).Value = fventa[i].id_producto;
            Comando.Parameters.Add(new SqlParameter("@cantidad",
            SqlDbType.VarChar)).Value = fventa[i].cantidad;
            Comando.Parameters.Add(new SqlParameter("@costototal",
            SqlDbType.VarChar)).Value = fventa[i].costo_total;
            Comando.Parameters.Add(new SqlParameter("@existenciaproducto",
            SqlDbType.Int)).Value = fventa[i].existencia_producto;
        }
    }
}
```





```
cgenerica.RealizarOperacion(Comando, Conexion, trans);
}
    trans.Commit();

returntrue;
}
catch (SqlException Ex)
{
    trans.Rollback();
XtraMessageBox.Show(Ex.Message, ConexionSQL.xTitlename, MessageBoxButtons.OK,
MessageBoxIcon.Information);
returnfalse;
}
finally
{
    Conexion.Close();
}
}

publicstaticbool Add(FacturaVenta fventa)
{
    Comando = newSqlCommand { CommandText = "guardarfacturaventa",
CommandType = CommandType.StoredProcedure };
    Comando.Parameters.Add(newSqlParameter("@idcliente",
SqlDbType.VarChar)).Value = fventa.id_cliente;
    Comando.Parameters.Add(newSqlParameter("@fechaventa",
SqlDbType.DateTime)).Value = fventa.fecha_venta;
    Comando.Parameters.Add(newSqlParameter("@totalventa",
SqlDbType.VarChar)).Value = fventa.total_venta;

returncgenerica.RealizarOperacion(Comando);
}

publicstaticDataTable obtenerfacturaventa()
{
    Comando = newSqlCommand { CommandType = CommandType.StoredProcedure,
CommandText = "Consultarfacturaventa" };
returncgenerica.ObtenerDatos(Comando);
}
}
```

