

UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA, LEÓN
FACULTAD DE CIENCIAS Y TECNOLOGÍA

DEPARTAMENTO DE COMPUTACIÓN



TESIS MONOGRÁFICA PARA OPTAR AL TÍTULO DE
INGENIERIA EN SISTEMAS DE INFORMACIÓN.

TEMA:

Automatización de los Procesos de Ventas, Compras, Facturación e Inventario de
La Farmacia ***“LA ASUNCION” del Municipio de Telica, León.***

Autores:

Br. Francisco Javier Altamirano Mayorga.
Br. Laysa Mariela Betancourt Silva.
Br. Inelia del Carmen Espinoza Sevilla.

Tutor

M.Sc. Álvaro Altamirano.

León 2013.

AGRADECIMIENTO.

Tener un corazón que se desborda de agradecimiento no es algo que surge con naturaleza. La mayoría de nosotros raramente nos despertamos agradecidos y alegres. Es mucho más fácil quejarse y criticar, para la mayoría de nosotros nuestro estado automático es la murmuración. Tenemos que empezar a trabajar en una actitud de gratitud, escojamos ser agradecidos, y con el tiempo nuestras emociones nos seguirán. Las emociones siguen a nuestras acciones y no al revés.

A **Dios** porque siempre ha estado con nosotros protegiéndonos en cada paso que damos dándonos sabiduría e inteligencia para culminar nuestros estudios profesionales.

A nuestros **Padres** por brindarnos su apoyo y amor incondicional siempre, alentándonos para seguir adelante y superarnos.

A nuestro tutor **MSc. Álvaro Altamirano** por la ayuda y paciencia brindada en nuestro trabajo monográfico.

A una persona muy especial para nosotros el **Lic. Andrés Acetuno Sánchez** ya que él ha estado presente con su apoyo en nuestras necesidades y siempre brindándonos una mano amiga incondicional.

Son muchas las personas que han formado parte de nuestra vida profesional a las que nos encantaría agradecerles su amistad, consejos, apoyo, ánimo y compañía en los momentos más difíciles de nuestras vidas. Algunas están aquí con nosotros y otras en nuestros recuerdos y en nuestros corazones, sin importar en donde estén queremos darles las gracias por formar parte de nuestras vidas, por todo lo que nos han brindado y por todas sus bendiciones.

Para ellos: Muchas gracias y que Dios los bendiga.

Francisco Altamirano, Laysa Betancourt, Inelia Espinoza.

DEDICATORIA.

A Dios:

Primeramente le dedico este trabajo a Dios Todopoderoso, que me brindo sabiduría e inteligencia para realizarlo, y la oportunidad de alcanzar una meta más en la vida.

A mis Padres:

Rosa Mayorga y Mauricio Altamirano, mis dos grandes pilares, que me brindaron su apoyo incondicional, amor, fortaleza y han sido inspiración.

A mis Amigos:

Los cuales, me apoyaron, ayudaron y enseñaron a ser perseverante.

Francisco Javier Altamirano Mayorga.

DEDICATORIA.

Dedico este proyecto de tesis primeramente a Dios, a mis padres y a mi hija.

A **Dios** porque siempre está conmigo donde quiera que vaya ayudándome y protegiéndome, él me ha regalado la fuerza, fortaleza y sabiduría para seguir adelante día a día.

A mi padre **José Guillermo Betancourt G.** y mi madre **Marcia Silva R.** quienes a lo largo de mi vida han velado por mi bienestar y educación siendo mi apoyo en todo momento, depositando su entera confianza en cada reto que se me presentaba sin dudar ni un solo momento en mi inteligencia y capacidad.

A mi hija Laysa Montserrat Betancourt la cual es un gran motor en mi vida para seguir superándome.

A todas las personas que de una u otra manera han aportado un granito de arena en mi preparación profesional.

Laysa Mariela Betancourt Silva.

DEDICATORIA

Este trabajo Monográfico se lo dedico especialmente a:

Dios Todopoderoso Por Haberme dado la sabiduría y la capacidad de vencer cada obstáculo que se me presento durante mis años de estudio al darme las fuerzas necesarias para culminar mi carrera profesional sueño que siempre he anhelado y gracias a su presencia cada día de mi vida esto va ser posible.

A mis Padres: **Gersán Iván Espinoza y Francisca Sevilla** por ser los seres que me dieron la vida y brindarme su apoyo incondicional e incentivado para seguir adelante siempre, son el pilar fundamental de mi vida.

A mis hermanos: **Mildred, Odilia, Francis y Gersán** por estar ahí siempre que los necesitaba y brindarme su ayuda en los momentos difíciles.

Inelia del Carmen Espinoza Sevilla.



INDICE

INTRODUCCIÓN	3
ANTECEDENTES	4
JUSTIFICACIÓN.....	5
OBJETIVOS	6
GENERAL:.....	6
ESPECIFICOS:.....	6
MARCO TEORICO	7
HERRAMIENTAS DE DESARROLLO	7
C SHARP	7
MICROSOFT VISUAL STUDIO 2010.	11
DEVCOMPONENTS DOTNETBAR	12
MYSQL SERVER	17
MYSQL CONNECTOR/ NET 6.6.4.	18
MYSQL DUMP.	19
LIBRERÍAS DE ENLACE DINÁMICO:	21
MYSQLBACKUP.	22
HELP N-DOC	30
LA CRIPTOGRAFÍA COMO ELEMENTO DE LA SEGURIDAD INFORMÁTICA.	30
SHA-1.....	32
CONCURRENCIA Y TRANSACCIONES.	32
MODELO EN TRES CAPAS:	35
RED CLIENTE/ SERVIDOR	38
ESPECIFICACIÓN DE REQUISITO SOFTWARE.....	40
CASOS DE USO EXTENDIDO.	73
DIAGRAMA DE CASOS DE USO.....	87
DIAGRAMAS DE SECUENCIA.....	89
METODOLOGIA	102
RECURSOS.....	107
DIAGRAMA ENTIDAD-RELACION.	108



DICCIONARIO DE DATOS.....	109
DISEÑO ARQUITECTONICO	123
CONCLUSIONES.....	127
RECOMENDACIONES.	128
ANEXOS.	130



INTRODUCCIÓN

En la actualidad, la tecnología ha ido avanzando a pasos agigantados y se encuentra en constante crecimiento, debido a esto automatizar los procesos en una empresa son cada vez más necesarios, por medio de esta se ha podido adquirir diversas herramientas que han logrado que la productividad aumente y que el trabajo del día a día se vuelva más eficaz.

Farmacia La Asunción está ubicada en el Municipio de Telica, Casa Cural 1/2c al Sur bajo la regencia de la Lic. Denia María Espinoza Mayorga, en dicha farmacia se realizan distintas tareas como ventas de medicamentos, Compras a los proveedores, facturación a los clientes dependiendo la modalidad de pago y se realiza Inventarios de dichos productos, además de informes detallados de las ventas efectuadas en el día, cuentas por pagar a proveedores y cuentas por Cobrar a los clientes. Con el propósito de tener un mayor control de los procesos que se realizan en dicha Farmacia surge la necesidad de automatizarlos para tener un manejo rápido de la información requerida.

De modo que realizamos un Sistema para la Automatización de los procesos de Ventas, Compras, Facturación e Inventario de la farmacia “LA ASUNCION” del municipio de Telica, León, así como también las entradas y salidas de los productos, además el control de cuentas por cobrar y cuentas por pagar.



ANTECEDENTES

Actualmente en la Farmacia La Asunción del municipio de Telica no existe ningún sistema automatizado que lleve un control sobre la información de los procesos realizados tales como: ventas, compras, facturación e inventario. Todo el proceso de almacenamiento de la información se realiza de forma manual ingresando la venta realizada en una hoja de cálculo de Excel, etiquetas colocadas en los productos, facturas y proformas que son archivadas en un folder para luego ser utilizadas.

Por estas razones la Farmacia espera contar con un sistema automatizado que permita manipular la información adecuadamente y por consiguiente una mayor eficiencia en los procesos antes mencionados, así como también reducir el tiempo de proceso de la información ya que estos se realizaban anteriormente de forma manual. De esta manera los procesos realizados en dicha farmacia estarán bien organizados y abra un control más eficaz para el desarrollo de dicho negocio.



JUSTIFICACIÓN

Debido a que Farmacia La Asunción del municipio de Telica no consta con un Sistema Automatizado que lleve el control de los procesos que en dicha se manejan y a las dificultades presentadas en cuanto al manejo rápido de la información por realizar las operaciones de forma manual ya que en dicha farmacia se manipulan alrededor de 500 productos y esto conlleva a distintos inconvenientes surgió la necesidad de crear este sistema.

Con este proyecto queremos mostrar la importancia de implementar un Sistema Automatizado y las ventajas que este aporta tales como: seguridad, fiabilidad y rapidez al momento de capturar y buscar la información. Además será de gran ayuda a la empresa porque las operaciones y los procesos que se realizan en dicha Farmacia serán mejor controlados y por lo consiguiente se evitaban inconvenientes.



OBJETIVOS

GENERAL:

Automatizar los Procesos de Ventas, Compras, Facturación e Inventario de la Farmacia La Asunción del Municipio de Telica, León.

ESPECIFICOS:

- Desarrollar una Aplicación con Visual Studio 2010, empleando como lenguaje de programación Visual C#.
- Generar reportes de productos controlados, ventas al día, inventario, ventas al mes, productos vencidos, cuentas por cobrar y cuentas por pagar.
- Realizar facturación a los clientes en la modalidad de pago: crédito, contado.
- Realizar respaldo y restauración de la Base de Datos.
- Implementar la aplicación en una red Cliente- Servidor.



MARCO TEORICO

HERRAMIENTAS DE DESARROLLO

Microsoft.NET es el conjunto de nuevas tecnologías en las que Microsoft ha estado trabajando durante los últimos años con el objetivo de obtener una plataforma sencilla y potente para distribuir el software en forma de servicios que puedan ser suministrados remotamente y que puedan comunicarse y combinarse unos con otros de manera totalmente independientes de la plataforma, lenguajes de programación y modelo de componentes con los que hayan sido desarrollados.

Esta es la llamada plataforma .NET y a los servicios antes comentados se les denomina servicios Web.

Para crear aplicaciones para la plataforma .NET, tanto servicios Web como aplicaciones tradicionales (aplicaciones de consola, aplicaciones de ventanas, servicios de Windows NT, etc.). Microsoft ha publicado el denominado Kit de desarrollo de software conocido como .NET Framework SDK, que incluye las herramientas necesarias tanto para su desarrollo como para su distribución y ejecución y Visual Studio.NET que permite hacer todo lo anterior desde una interfaz visual basada en ventanas.

El concepto de Microsoft .NET también incluye al conjunto de nuevas aplicaciones que Microsoft y terceros están desarrollando para ser utilizadas en la plataforma .NET. Entre ellas podemos destacar aplicaciones desarrolladas por Microsoft tales como Windows.NET, Hailstorm, Visual Studio.NET, MSN.NET, Office.NET y los nuevos servidores para empresas de Microsoft (SQL Server.NET, Exchange.NET. etc.)

C SHARP

C Sharp es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un



estándar por la ECMA (ECMA-334) e ISO (ISO/IEC 23270). C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común.

Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes.

El nombre C Sharp fue inspirado por la notación musical, donde '#' (sostenido, en inglés *Sharp*) indica que la nota (C es la nota do en inglés) es un semitono más alta, sugiriendo que C# es superior a C/C++.

Aunque C# forma parte de la plataforma .NET, ésta es una API, mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Ya existe un compilador implementado que provee el marco Mono - DotGNU, el cual genera programas para distintas plataformas como Windows, Unix, Android, iOS, Windows Phone, Mac OS y GNU/Linux.

Características de C Sharp

Algunas de estas características no son propias del lenguaje, sino de la Plataforma .NET, aunque se listan aquí ya tienen una implicación directa en el lenguaje.

- **Sencillez de Uso:** C# elimina muchos elementos añadidos por otros lenguajes y que facilitan su uso y comprensión, como por ejemplo los ficheros de cabecera o ficheros fuentes. Además no se incorporan al lenguaje elementos poco útiles como por ejemplo macros, herencia múltiple u operadores diferentes al operador de acceso a métodos (operador punto) para acceder a miembros de espacios de nombres.
- **Modernidad:** Al ser C# un lenguaje de última generación, incorpora elementos que se han demostrado a lo largo del tiempo que son muy útiles para el programador, como tipos decimales o booleanos, un tipo básico string, así como una instrucción



que permita recorrer colecciones con facilidad (Instrucción foreach). Estos elementos hay que simularlos en otros lenguajes.

- Orientado a objetos: C# como lenguaje de última generación y de propósito general es orientado a objetos. C# no permite la inclusión de funciones y variables globales que no estén incluidos en la definición de tipos, por lo que la orientación a objetos es más pura y clara que en otros lenguajes como C++. Además C# soporta las características del paradigma de la programación orientada a objetos como son la encapsulación, la herencia y el polimorfismo.
- Orientado a Componentes: La propia sintaxis de C# incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular. La sintaxis de C# incluye por ejemplo formas de definir propiedades, eventos o atributos.
- Recolección de Basura: Como ya se comentó todo lenguaje incluido en la plataforma .NET tiene a su disposición el recolector de basura del CLR. Esto implica que no es necesario incluir instrucciones de destrucción de objetos en el lenguaje.
- Seguridad de Tipos: C# incluye mecanismos de control de accesos a tipos de datos, lo que garantiza que no se produzcan errores difíciles de detectar como un acceso a memoria de ningún objeto por ejemplo. Para ello el lenguaje provee de una serie de normas de sintaxis como por ejemplo no realizar conversiones entre tipos que no sean compatibles. Además no se pueden usar variables no inicializadas previamente y en acceso a tablas se hace una comprobación de rangos para que no accedan ninguno de los índices de la misma. Se puede controlar así mismo los desbordamientos en operaciones aritméticas produciéndose excepciones.



- Instrucciones Seguras: Para evitar errores comunes como se producían programando en otros lenguajes en C# se han impuesto una serie de restricciones en el uso de instrucciones más comunes.
- Unificación de Tipos: En C# todos los tipos derivan de una superclase común llamada System.Object por lo que automáticamente heredaran todos los miembros definidos en esta clase.
- Extensión de operadores Básicos: Para facilitar la legibilidad de código y conseguir que los nuevos tipos de datos que se definan a través de las estructuras estén al mismo nivel que los elementos predefinidos en el lenguaje. C# permite redefinir el significado d la mayoría de los operadores cuando se aplique a diferentes tipos de objetos.

Las redefiniciones de operadores se hacen de manera inteligente, de modo que a partir de una única definición de los operadores ++ y - el compilador puede deducir automáticamente como ejecutarlos de manera prefija y postfija. Definiendo operadores simples como la suma, el compilador deduce como aplicar la versión de asignación compuesta (+=). Además para asegurar la consistencia, el compilador exige que los operadores son opuestos (por ejemplo el operador igualdad == y su opuesto!=) siempre se redefinan por parejas.

- Extensión de Modificadores: C# ofrece a través de los atributos la posibilidad de añadir a los metadatos del módulo resultante de la compilación de cualquier fuente de información adicional a la generada por el compilador que luego podrá ser consultada en tiempo de ejecución a través de la biblioteca de reflexión de .NET.



- Eficiente: En C# todo el código incluye numerosas restricciones para garantizar su seguridad, no permitiendo el uso de punteros. Esta característica puede resultar de utilidad en situaciones en las que se necesite gran velocidad de procesamiento.
- Compatible: Para facilitar la migración de programadores de C++ o Java a C# , no solo se mantiene una sintaxis muy similar a la de los dos anteriores lenguajes, sino que el CLR también ofrece la posibilidad de acceder a código nativo escrito como funciones sueltas no orientadas a objetos tales como las DLL de la API de Win32.

MICROSOFT VISUAL STUDIO 2010.

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE) de Microsoft. Se utiliza para desarrollar consola y de interfaz gráfica de usuario de las aplicaciones junto con Windows Form aplicaciones, sitios Web, aplicaciones Web y servicios Web, Tanto en código nativo junto con código administrado para todas las plataformas soportadas por Microsoft Windows, Windows Mobile, Windows CE, .NET Marco, .NET Compact Framework y Microsoft Silverlight.

Visual Studio incluye un editor de código de soporte IntelliSense y refactorización de código. El integrado depurador funciona como un depurador a nivel de fuente y un depurador a nivel de máquina. Otras herramientas integradas incluyen un diseñador de formularios para la construcción de GUI de aplicaciones, diseñador web, clase de diseño y esquema de la base de diseño. Se acepta plug-ins que mejoran la funcionalidad en casi todos los niveles, incluyendo la adición de soporte para control de código fuente de sistemas (como Subversión y Visual SourceSafe) y la adición de nuevos conjuntos de herramientas como editores y diseñadores visuales para lenguajes específicos de dominio o conjuntos de herramientas para otros aspectos de la ciclo de vida de desarrollo de software (como el Team Foundation Server cliente: Team Explorer).

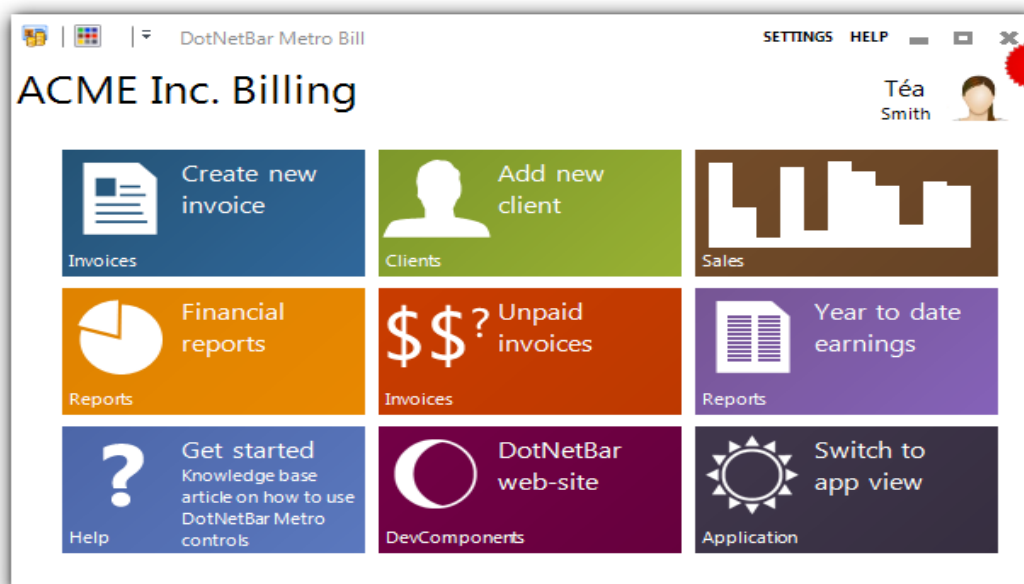


Visual Studio es compatible con diferentes lenguajes de programación por medio de los servicios de idiomas. Lo que permite el editor de código y un depurador para apoyar (en diversos grados) casi cualquier lenguaje de programación, siempre un servicio específico del lenguaje existente. Construido en idiomas incluyen C / C++ (a través de Visual C + +), VB.NET (a través de Visual Basic. NET), C # (mediante Visual C #), y F # (a partir de Visual Studio 2010). Soporte para otros idiomas como el M, Python, y Rubí, entre otros están disponibles a través de servicios de idiomas instalados por separado. También es compatible con XML / XSLT, HTML / XHTML, Java Script y CSS. Individuales específicos del idioma versiones de Visual Studio también existen que proporcionan servicios lingüísticos más limitados para el usuario: Microsoft Visual Basic, Visual J #, Visual C # y Visual C ++.

DEVCOMPONENTS DOTNETBAR

DotNetBar es una herramienta para Windows Forms, con 77 componentes para la creación de profesionales interfaces de usuario con facilidad, utilizando Visual Studio 2005 hasta el 2012. Por más de 10 años DotNetBar está ayudando a los desarrolladores a crear elegantes interfaces inigualables.

DotNetBar es el primero en el mundo en introducir todas las funciones de Office 2013, Office 2010, Windows 7 y Office 2007, controles de cinta de estilo, en primer lugar para ofrecer plena compatibilidad con Windows XP Theme.



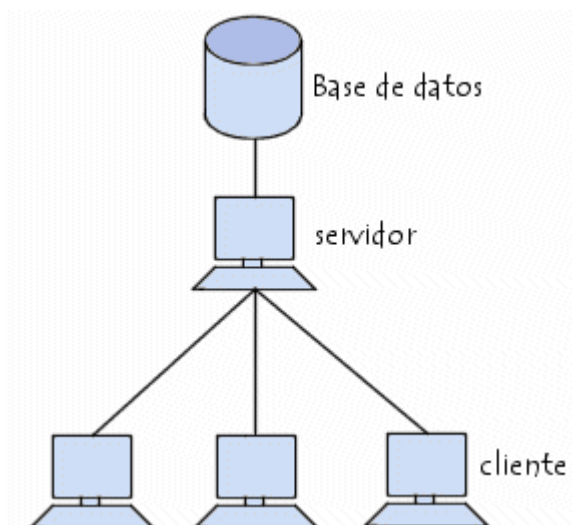
CRYSTAL REPORTS VISUAL STUDIO 2010.

Crystal Reports es una aplicación de inteligencia empresarial utilizada para diseñar y generar informes desde una amplia gama de fuentes de datos (bases de datos).

Varias aplicaciones, como Microsoft Visual Studio, incluyen una versión OEM de *Crystal Reports* OEM (en español "fabricante de equipamiento original") es una empresa que fabrica productos que luego son comprados por otra empresa y vendidos bajo la marca de la empresa compradora como una herramienta de propósito general para informes/reportes. Crystal Reports se convirtió en el escritor de informes estándar cuando Microsoft lo liberó con Visual Basic.

Los usuarios al instalar Crystal Reports en un equipo y utilizarlo para seleccionar filas y columnas específicas en una tabla de datos compatibles, pueden organizar los datos en el informe en el formato que necesiten. Una vez que el diseño está completo, el informe se puede guardar como un archivo con extensión rpt, pdf, xls, docx. Se puede acceder nuevamente al informe reabriendo el mismo y poder refrescar los datos. Si la fuente de base de datos se ha actualizado, el informe se refrescará reflejando estas actualizaciones.

BASE DE DATOS.



Una base de datos es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite. Una base de datos es un sistema de archivos electrónico.

Las bases de datos tradicionales se organizan por campos, registros y archivos. Un campo es una pieza única de información; un registro es un sistema completo de campos; y un archivo es una colección de registros. Por ejemplo, una guía de teléfono es análoga a un archivo. Contiene una lista de registros, cada uno de los cuales consiste en tres campos: nombre, dirección, y número de teléfono.

Tipos de Base de Datos:

Las bases de datos pueden clasificarse de varias maneras, de acuerdo al contexto que se esté manejando, la utilidad de las mismas o las necesidades que satisfagan.

Según la variabilidad de los datos almacenados.



Base de datos estáticas:

Son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones, tomar decisiones y realizar análisis de datos para inteligencia empresarial.

Base de datos dinámicas:

Éstas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización, borrado y adición de datos, además de las operaciones fundamentales de consulta. Un ejemplo de esto puede ser la base de datos utilizada en un sistema de información de un supermercado, una farmacia, un videoclub o una empresa.

Modelo de Base de Datos:

Actualmente, los SGBD's constituyen el núcleo fundamental del soporte lógico a los sistemas de información. Desde la aparición de los primeros SGBD's comerciales en la década de los 60 hasta la actualidad, se han sucedido tres generaciones distintas de SGBD's basadas en tres modelos de datos diferentes.

Los tres modelos en los que se ha basado el desarrollo de las bases de datos son el jerárquico, en red y relacional. El modelo jerárquico domino el mercado de los SGBD's hasta mediados de los 80. Durante este mismo periodo, surgió el modelo en red con el que se pretendía sustituir a los SGBD's jerárquicos, lo que no se consiguió. A principios de los 80 el modelo jerárquico comenzó a ser sustituido por una nueva generación de SGBD's basados en el modelo relacional que en la actualidad, dominan totalmente el mercado.

Los SGBD's jerárquicos fueron los primeros en aparecer. Una base de datos jerárquica se puede visualizar como una estructura en árbol. Las bases de datos jerárquicas son



bastantes rígidas. Una vez diseñada la base de datos, es complejo cambiarla y además, es necesario un conocimiento amplio de la forma en la que se han almacenado los datos para poder recuperarlos en forma efectiva. Por ello, a pesar de haber dominado el mercado en sus comienzos, los SGBD's han ido decayendo y actualmente no se encuentran en el mercado.

Los SGBD's en red fueron una evolución del mercado jerárquico. En una base de datos en red, cada uno de los registros están enlazados entre sí, pero no necesariamente siguiendo una estructura en árbol. El modelo en red elimina parte de las rigideces del modelo jerárquico, pero aumenta la complejidad para modificar la estructura de la base de datos. Por ello, a pesar de su buen rendimiento, el número de instalaciones con SGBD's en red siempre ha sido pequeño y hoy en día tampoco se encuentran en el mercado. Sin embargo, aún quedan instalaciones basadas en estos dos modelos de datos, que responden con gran eficiencia y plena satisfacción de sus usuarios.

Los SGBD's relacionales. En una base de datos relacional, se representan los datos como un conjunto de tablas bidimensionales compuestas de filas y columnas. Cada fila representa una relación entre un conjunto de valores y está identificada por una clave única. Los SGBD's relacionales son muy flexibles y de fácil manejo, lo que los ha convertido en el modelo dominante en la actualidad. Un factor decisivo en la implantación de los SGBD's relacionales, ha sido el lenguaje SQL (StructuredQueryLanguage) para la interrogación y el manejo de datos del modelo relacional.

A pesar de que los avances en el campo de los SGBD's han sido muy grandes y de que los SGBD's actuales proporcionan un rendimiento elevado, aun están lejos de satisfacer todas las demandas de las aplicaciones más avanzadas. Siguen existiendo hoy en día numerosas áreas científicas e industriales, en las que los SGBD's relacionales están escasamente introducidos. Muchos de estos campos exigen la manipulación de nuevos tipos de



información o suponen unos volúmenes de datos de difícil manejo por los sistemas actuales.

Características:

- Una base de datos relacional se compone de varias tablas o relaciones.
- No pueden existir dos tablas con el mismo nombre ni registro.
- Cada tabla es a su vez un conjunto de registros (filas y columnas).
- La relación entre una tabla padre y un hijo se lleva a cabo por medio de las claves primarias y ajenas (o foráneas).
- Las claves primarias son la clave principal de un registro dentro de una tabla y éstas deben cumplir con la integridad de datos.
- Las claves ajenas se colocan en la tabla hija, contienen el mismo valor que la clave primaria del registro padre; por medio de éstas se hacen las relaciones.

MYSQL SERVER

MySQL es un SGBD (sistema de gestor bases de datos) relacional, multihilo y multiusuario con más de seis millones de instalaciones.

Una base de datos es una colección estructurada de tablas que contienen datos. Para agregar, acceder y procesar datos guardados en un computador, usted necesita un administrador como MySQL Server, dado que los computadores son muy buenos manejando grandes cantidades de información, los administradores de bases de datos juegan un papel central en computación, como aplicaciones independientes o como parte de otras aplicaciones.

También es un sistema de administración relacional de bases de datos.

MySQL es software de código fuente abierto (open source), significa que es posible modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar.



Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades. MySQL usa el GPL (GNU General Public License) para definir qué puede hacer y que no puede hacer con el software en diferentes situaciones.

Características:

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas web con contenido dinámico, justamente por su simplicidad.

Poco a poco los elementos de los que carecía MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar:

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferente velocidad de operación, soporte físico, capacidad, distribución geográfica, transacciones.
- Transacciones y claves foráneas.
- Conectividad segura.
- Búsqueda e indexación de campos de texto.

MYSQL CONNECTOR/ NET 6.6.4.

MySQL Connector/ Net 6.6.4, una nueva versión del conector disponible para tecnologías.NET para MySQL ha sido puesto en libertad. Pretende introducir a los usuarios a las nuevas características de la versión.



La versión 6.6 de MySQL Connector/ Net trae las siguientes novedades:

- Depuración de procedimientos almacenados.
- Entity Framework 4.3 Apoyo primer código.
- Autenticación conectable (ahora terceros puede conectar mecanismos de autenticaciones nuevas en el controlador.)

MYSQLDUMP.

MYSQLDUMP es una aplicación del sistema gestor de Base de Datos MYSQL, sirve para hacer copias de seguridad. Este comando dispone de una amplia variedad de opciones que nos permitirá realizar la copia de la forma más conveniente para el propósito de la misma.

Para poder restaurar la copia de seguridad, bastara con ejecutar todas las sentencias SQL que se encuentran dentro del fichero, bien desde la línea de comandos MYSQL o desde la pantalla de creación de sentencias SQL de cualquier entorno gráfico.

Las limitaciones de la restauración dependerán de las opciones que han especificado a la hora de hacer la copia de seguridad, por ejemplo, si se incluye la opción `--add-drop-table` al hacer la copia de seguridad, se podrán restaurar tablas que existen actualmente en el servidor (borrándolas primero). Por lo que es necesario estudiar los procedimientos que se utilizaran tanto en la copia como en la restauración, para que todo salga correcto.

Ejemplos del uso de Mysqldump:

Para realizar la copia de seguridad de la base de datos mibase al fichero `copia_seguridad.sql`

```
mysqldump --opt mibase> copia_seguridad.sql
```



Otro ejemplo más complejo de comando mysqldump para hacer backup de una base de datos es el siguiente:

```
Mysqldump -opt -password=miclave -user=miuser mibasededatos> archive.sql
```

En este último caso estamos indicando un nombre de usuario y una clave para acceder a la base de datos sobre la que se está haciendo el backup: mibasededatos. Las sentencias SQL para reconstruir esa base de datos se volcara en el fichero archivo.sql.

Restaurar la Base de Datos.

Si deseamos recuperar la información de un fichero para restaurar una copia de seguridad de la base de datos lo haremos con el comando MySQL. Utilizaremos una sintaxis como esta:

```
MySQL mibase < archivo.sql
```

En este ejemplo se restauraría la base de datos mibase con el backup almacenado en el fichero archivo.sql.

Otro ejemplo más complejo de comando para restaurar una base de datos es el siguiente:

```
Mysql -password=miclave -user=miuser mibase < archive.sql
```

En el mismo ejemplo que el anterior, pero indicando un nombre de usuario y una clave con las que se podrá acceder a la base de datos mibase.



LIBRERÍAS DE ENLACE DINÁMICO:

Una librería de enlace dinámico, en adelante DLL (Dynamic Link Library), es un archivo que contiene funciones y recursos (Mapas de bits, definiciones de fuentes, etc.) que podrán ser llamados desde cualquier aplicación Windows.

De hecho, se puede considerar que Windows está construido sobre una gran cantidad de DLL's. La mayoría de librerías de enlace dinámico se guardan en archivos que tienen extensión DLL, pero también pueden ser guardados en archivos con extensiones EXE (ejecutable), DRV (controlador de dispositivo) y (fuente de Windows).

La diferencia entre las librerías de enlace dinámico con extensión DLL y el resto, es que las primeras se cargan porque el programa que las de utilizar lo pide a Windows y las demás, en cambio, se cargan porque están referenciadas en archivos de inicialización de Windows. Estas referencias pueden ser creadas por el propio Windows o por el programa de instalación de alguna aplicación.

Ventajas e Inconvenientes del uso de DLL:

Ventajas:

- Una función definida dentro de una DLL está disponible para cualquier aplicación Windows.
- Se reduce el tamaño de las aplicaciones que utilizan DLL por la reutilización de su código.
- Mejora en el tiempo de compilación y carga de la aplicación (debido al menor tamaño del código).
- Ahorro de espacio en disco.
- Las DLL's son independientes de la aplicación.



Inconvenientes:

- Tienen que estar presente en la carpeta del sistema antes de ser utilizadas.
- El tiempo de acceso a la DLL por parte de la aplicación que la usa es más lento.

Estructura de una DLL de 32 bits:

Una DLL se puede dividir, básicamente, en tres partes:

- Archivo de cabecera: Contendrá todas las funciones y definiciones (de variables, funciones, etc.) que use la DLL.
- Punto de entrada y salida a la DLL: Es la función principal de la DLL y es la que se encarga de cargar la DLL (cuando se vaya a usar) y descargarla de la memoria (cuando se deje de usar). Dicha función se llama DllEntryPoint.
- Funciones que contiene la DLL: Son las funciones que contiene la DLL y fueron declaradas por el programador de la misma.

MYSQLBACKUP.

Fue creado el miércoles 20 de Junio del 2012, v1.2 es revisado y mejorado basada en la versión v1.1. Este lanzamiento no afecta el desarrollo v2.0 (Agosto 2012).

Es una herramienta para realizar el respaldo y restauración de base de datos MYSQL, se puede utilizar en lenguaje de programación C#/ VB.NET/ASP.NET y soporta Unicode/UTF8.

Estructura de Clases de MySqlBackup.

Constructor

- MySQLBackup ()
- MySQLBackup (cadena)
- MySQLBackup (MySqlConnection)
- MySQLBackup (MySqlCommand)



Propiedades Públicas:

- **Conexión** (MySqlConnection) - Obtiene o establece el MySqlConnection que utiliza esta instancia.
- **DatabaseInfo** (Base de datos) - Obtiene la información alguna acerca de la base de datos conectada.
 - Propiedades, campos, métodos
 - **CreateDatabaseSql** (string) - Obtiene la instrucción SQL CREATE DATABASE de esta base de datos.
 - **DatabaseName** (string) - Obtiene el nombre de la base de datos.
 - **Tablas** (Diccionario <string, <Tabla>) - Obtiene toda la información de las tablas. Clave = Nombre de la tabla; = Valor de tabla.
 - **ServerVersion** (string) - Obtiene el número de la versión del servidor MySQL y el nombre.
 - **ServerVersionNo** (string) - Obtiene el número de la versión del servidor MySQL.
 - **ServerMajorVersion** (doble) - Obtiene el número de la versión del servidor MySQL Mayor.
 - **DefaultDatabaseCharSet** (string) - Obtiene el conjunto de caracteres por defecto de la base de datos actual.
 - **Nombres de tablas** (String []) - Obtiene el nombre de todos mesa.
 - **StoredProcedure** (<string, string> Dictionary) - Obtiene o establece procedimientos almacenados de la base de datos actual.
 - **StoredFunction** (<string, string> Dictionary) - Obtiene o establece las funciones guardadas de la base de datos actual.
 - **StoredTrigger** (<string, string> Dictionary) - Obtiene o establece activadores almacenados de base de datos actual;
 - **StoredEvents** (<string, string> Dictionary) - Obtiene o establece los eventos almacenados de la base de datos actual.



- **StoredView** (<string, string> Dictionary) - Obtiene o establece Vistas almacenados de la base de datos actual.
- **GetTotalRows** (Long) - Obtiene el total de filas de todas las tablas de la base de datos actual.
- **(Event) CalculateTotalRowsProgressChanged** - Ocurren cuando las filas cálculo total de 1 mesa ha completado.
- **(Event) CalculateTotalRowsCompleted** - Ocurren cuando calculation filas total de toda la tabla se ha completado.
- **ExportInfo (ExportInformations)** - Obtiene o establece la información que define el comportamiento del proceso de exportación. Lea más en la clase de ExportInformations abajo.
 - **Propiedades, métodos y campos**
 - **Nombre Archivo** - Obtiene o establece la ruta completa y el nombre de archivo (archivo de volcado) que se guardó. El resultado del proceso de exportación.
 - **TableCustomSql** - Obtiene o establece las tablas que serán exportados y el SQL para seleccionar las filas de las tablas. Si el SQL se deja en blanco, el SQL reemplazará por "SELECT * FROM tablename". Clave = nombretabla, Valor = SQL
 - **TablesToBeExported** - Obtiene o establece las tablas que se exportarán
 - **RecordDumpTime** - Obtiene o establece un valor que indica si el tiempo de descarga se debe registrar en el archivo de volcado.
 - **AsynchronousMode** - Obtiene o establece un valor que indica si el proceso de exportación se debe ejecutar en modo asíncrono.
 - **CalculateTotalRowsFromDatabase** - Obtiene o establece un valor que indica si el total de filas se debe calcular antes de proceso de exportación se inicia.



- **AutoCloseConnection** - Obtiene o establece un valor que indica si el MySqlConnection y usados MySqlCommand debe cerrar y eliminar después de terminado el proceso de exportación.
- **EnableEncryption** - Obtiene o establece un valor que indica si el archivo de volcado de exportación debe estar encriptado.
- **EncryptionKey** - Obtiene o establece la clave o contraseña que se utiliza para cifrar el archivo de volcado de exportación.
- **SaltSize** - Obtiene la longitud de la sal utilizada en el cifrado.
- **AddCreateDatabase** - Obtiene o establece un valor que indica el estado de cuenta de SQL "CREATE DATABASE" se debe añadir en el archivo de volcado.
- **ExportTableStructure** - o establece un valor que indica la estructura de la tabla (CREATE TABLE) debe ser exportado.
- **ResetAutoIncrement** - Obtiene o establece un valor que indica el valor de incremento automático de cada tabla se debe restablecer a 1.
- **ExportRows** - Obtiene o establece un valor que indica las filas deben ser exportados.
- **MaxSqlLength** - Obtiene o establece la longitud máxima permitida para múltiples sentencias SQL INSERT para unirse. El valor predeterminado es 1 MB.
- **ExportStoredProcedures** - Obtiene o establece un valor que indica si los procedimientos almacenados deben ser exportados.
- **ExportFunctions** - Obtiene o establece un valor que indica si las funciones almacenados deben ser exportados.
- **ExportTriggers** - Obtiene o establece un valor que indica si los disparadores almacenados deben ser exportados.
- **ExportViews** - Obtiene o establece un valor que indica si los dictámenes almacenados deben ser exportados.



- **ExportEvents** - Obtiene o establece un valor que indica si los eventos almacenados deben ser exportados.
- **CompleteArg** - ExportCompleteArg (Los argumentos del evento de ExportCompleted).
- **ZipOutputFile** - Obtiene o establece un valor que indica si el archivo exportado SQL Dump debe ser comprimido como archivo Zip.
- **ImportInfo (ImportInformations)** - Obtiene o establece la información que define el comportamiento del proceso de importación.
 - **Propiedades, métodos y campos**
 - **Nombre Archivo** - Obtiene o establece la ruta completa y el nombre de archivo (archivo de volcado) que se importarán.
 - **AsynchronousMode** - Obtiene o establece un valor que indica si el proceso de importación se debe ejecutar en modo asíncrono.
 - **AutoCloseConnection** - Obtiene o establece un valor que indica si el MySqlConnection y usados MySqlCommand debe cerrar y eliminar después de terminado el proceso de importación.
 - **EnableEncryption** - Obtiene o establece un valor que indica si el archivo de volcado importado es encriptado.
 - **EncryptionKey** - Obtiene o establece la clave o contraseña que se utiliza para descifrar el archivo de volcado de exportación.
 - **SaltSize** - Obtiene la longitud de la sal utilizada en el cifrado.
 - **SetTargetDatabase** (string, string) - Defina el nombre de la base de datos de destino que será importada y el conjunto de caracteres por defecto. Si la base de datos si no existe, se crea.
 - **SetTargetDatabase** (cadena) - Defina el nombre de la base de datos de destino que se importarán a. Si la base de datos si no existe, se crea. El juego de caracteres por defecto de la corriente de conexión del servidor MySQL será utilizado como juego de caracteres predeterminado para la base de datos de destino nuevo.



- **SetTargetDatabase** (cadena, Enum) - el nombre de la base de datos de destino que será importada y el conjunto de caracteres por defecto. Si la base de datos si no existe, se crea.
- **CreateTargetDatabaseSql** - Obtiene el CREATE DATABASE SQL de base de datos actual.
- **TargetDatabase** - Obtiene el nombre de la base de datos de destino.
- **IgnoreSqlError** - Obtiene o establece un valor que indica si SQL occus errores en el proceso de importación debe ser ignorado. Si es verdad, todos los errores (excepciones) se recogerán en ImportCompleteArg.Errors.
- **CompleteArg** - ImportCompleteArg (argumento Evento de ImportCompleted)

Métodos Públicos:

- **Exportar ()** - Ejecuta la exportación (copia de seguridad) del proceso.
- **Exportación (ExportInformations)** - Ejecutar la exportación (copia de seguridad) del proceso. Establece los ExportInformations utilizados por este proceso de exportación.
- **CancelExport ()** - Cancelar el proceso actual de ejecución de las exportaciones.
- **Importe ()** - Ejecuta el proceso de importación.
- **Importación (ImportInformations)** - Ejecutar la importación (restauración) del proceso. Establece los ImportInformations acerca de este proceso de importación.
- **CancelImport ()** - Cancelar el proceso actual de ejecución de la importación.
- **DeleteAllRows (bool)** - Eliminar todas las filas de todas las tablas. Establece un valor que indica si el incremento automático debe restablecer a 1.
- **DeleteAllRows (bool, string [])** - Eliminar todas las filas de todas las tablas. Establece un valor que indica si el incremento automático debe restablecer a 1. Excluir estas tablas se borre filas.



- **EncryptSqlDumpFile (string, string)** - Cifrar un archivo de volcado de SQL y guardarlo como archivo nuevo.
 - **DecryptSqlDumpFile (string, string)** - Descifrar un archivo de volcado de SQL y guardarlo como archivo nuevo.
 - **ExportBlobAsFile (cadena, cadena, cadena, cadena, cadena)** - Exportación de datos BLOB tipo y guardar como archivo.
 - **Dispose ()** - Desbloquear todos los recursos utilizados por esta instancia.
 - **Dispose (bool)** - Desbloquear todos los recursos utilizados por esta instancia. Determinar si MySqlConnection y MySqlCommand utiliza esta instancia debe disponer.
- '''

Eventos:

- **ExportProgressChanged** - Ocurren cuando una fila de datos se exportan o el cálculo del total de filas de una tabla completa.
 - Argumentos del evento:
 - **CurrentTableName** - String
 - **TotalRowsInCurrentTable** - Long
 - **TotalRowsInAllTables** - Long
 - **CurrentRowInCurrentTable** - Long
 - **CurrentRowInAllTable** - Long
 - **TotalTables** - Int.
 - **CurrentTableIndex** - Int.
 - **PercentageCompleted** - Int.
 - **PercentageGetTotalRowsCompleted** - Int.
- **ExportCompleted** - Ocurren cuando process exportación ha finalizado.
 - Argumentos del evento:
 - **TimeStart** - Date Time



- **TimeEnd** - Date Time
- **TimeUsed** - TimeSpan
- **Error** - Excepción (error que se producen durante el proceso)
- **CompletedType** - Enum (Completado, Cancelado, Error)
- **ImportProgressChanged** - Ocurren cuando una línea en el archivo de volcado se importa.
 - Argumentos del evento:
 - **CurrentByte** - Long
 - **TotalBytes** - Long
 - **Error** - Excepción (La información de error que encuentro.)
 - **CurrentLineNo** - Long
 - **ErrorSql** - String (La sentencia SQL que crea el error)
 - **PercentageCompleted**
- **ImportCompleted** - Ocurren cuando el proceso de importación ha terminado.
 - Argumentos del evento:
 - **TimeStart** - Date Time
 - **TimeEnd** - Date Time
 - **TimeUsed** - TimeSpan
 - **CurrentLineNo** - Long
 - **HasErrors** - Bool
 - **LastError** - Excepción (El último error (excepción) se producen en el proceso de importación.)
 - **CompletedType** - Enum (Completado, Cancelado, Error)
 - **Errores** - Diccionario <Long, Exception> (. La colección de errores Las excepciones serán recogidos si Ignorar errores de SQL se ha establecido en true en Información de Importación.)



HELP N-DOC

Help NDoc es un generador de documentación para Infraestructura de lenguaje común que utiliza documentación conectable para generar varios formatos de salida diferentes, incluyendo el estilo MSDN de la Ayuda HTML (.Chm), VisualStudio.NET (HTML Help 2), y páginas web de MSDN en línea. NDoc también viene con una interfaz gráfica de usuario para facilitar el proceso de generación.

NDoc actualmente es un generador de documentación libre, su licencia es la GNU General Public License.

Funcionamiento:

NDoc utiliza dos fuentes para generar la documentación. La primera es un conjunto de archivos que se producen al compilar el código fuente. La otra es un pre-generador de documentación en archivos XML, por lo general producidos por analizar el código fuente de los comentarios especiales (compiladores de C# de .NET y Mono apoyan esto usando el comando "/doc" como argumento de la línea de comandos).

LA CRIPTOGRAFÍA COMO ELEMENTO DE LA SEGURIDAD INFORMÁTICA.

El surgimiento de redes de comunicación, en particular de Internet, ha abierto nuevas posibilidades para el intercambio de información. Al mismo tiempo, son cada vez mayores las amenazas a la seguridad de la información que se transmite. Es necesario entonces, crear diferentes mecanismos, dirigidos a garantizar la confidencialidad y autenticidad de los documentos electrónicos, todo ello es parte de una nueva tecnología denominada Criptografía. Se aborda el tema de la seguridad informática, en específico las diversas variantes criptográficas: simétrica y asimétrica. Se explican algunos esquemas de manejo de llaves privadas y públicas: RSA y PGP. Finalmente, se tratan algunas de las limitaciones de las soluciones que ofrece esta nueva disciplina.



En Internet, las principales amenazas para la protección de la información provienen de:

- Anexos a mensajes enviados por correo electrónico infectados con virus.
- El intercambio de códigos de virus.
- Firewalls o cortafuegos mal configurados.
- Ataques a la disponibilidad de los recursos de información existentes en la red (bancos de datos o software disponibles para descargar por los usuarios).
- La alteración de las páginas web.
- El "repudio" y las estafas asociadas al comercio electrónico.
- Las vulnerabilidades de los sistemas operativos y la desactualización de los "parches" concernientes a su seguridad.
- La rotura de contraseñas.
- La suplantación de identidades.
- El acceso a páginas pornográficas, terroristas, etc.
- El robo y la destrucción de información.
- Pérdida de tiempo durante el acceso a sitios ajenos a la razón social de la entidad.
- El hecho de que herramientas de hacking y cracking se ofrezcan como freeware.

CRIPTOGRAFIA: Es la ciencia y arte de escribir mensajes en forma cifrada o en código.

Es parte de un campo de estudios que trata las comunicaciones secretas, usadas, entre otras finalidades, para:

- Autentificar la identidad de usuarios.
- Autentificar y proteger el sigilo de comunicaciones personales y de transacciones comerciales y bancarias.
- Proteger la integridad de transferencias electrónicas de fondos.

El objetivo principal de la criptografía es mandar un mensaje de "manera "oculta", llamado "cifrado" o "encriptado" y que solo el receptor con una llave o clave secreta pueda "descifrarlo" y leer su contenido.



SHA-1.

Hash es una función o método para generar claves o llaves que representen de manera casi unívoca a un documento o archivo.

Es una función criptográfica de hash diseñado por la Agencia de Seguridad Nacional y publicado por el NIST (National Institute of Standards Technology) de los Estados Unidos en 1995.

SHA significa Secure Hash Algorithm (Algoritmo Hash de Seguridad). Es uno de los algoritmos más utilizados y se emplea en varias aplicaciones y protocolos de seguridad, el tamaño de sus mensajes es de 160 bits.

Es un método muy bueno de criptografía que convierte una cadena de texto en otra de cuarenta caracteres sin importar la longitud de la cadena original y cifrándola de manera que se hace más difícil poder obtenerla, ya que SHA 1 no tiene método de reversa para obtener la clave original a partir de una ya cifrada.

El Hash de la cadena de longitud cero es:

=Da39a3ee 5e6b4b0d 3255bfef 95601890 afd80709

CONCURRENCIA Y TRANSACCIONES.

A menudo, desde el punto de vista del usuario, se considera a un conjunto de varias operaciones sobre una base de datos como una única operación. Por ejemplo, una transferencia de fondos desde una cuenta corriente de ahorros es una operación simple desde el punto de vista del cliente; sin embargo, en el sistema de base de datos, está compuesta internamente por varias operaciones. Evidentemente es esencial que tengan lugar todas las operaciones o que, en caso de fallo, ninguna de ellas produzca. Sería inaceptable efectuar el cargo de la transferencia en la cuenta corriente y que no se abonase en la cuenta de ahorros.



Transacción: Es una colección de operaciones que forman una única unidad lógica de trabajo. Un sistema de base de datos debe asegurar que la ejecución de las transacciones se realice adecuadamente a pesar de la existencia de fallos o se ejecute una transacción completa o no se ejecute en lo absoluto. Además debe gestionar la ejecución concurrente de las transacciones evitando introducir inconsistencias. Además se puede decir que una transacción es una unidad de la ejecución de un programa que accede y posiblemente actualiza varios elementos de datos, esta se inicia por la ejecución de un programa de usuario escrito en un lenguaje de manipulación de datos de alto nivel o un lenguaje de programación (por ejemplo SQL, COBOL, C, C++ o Java) y está delimitado por instrucciones o llamadas a función de la forma inicio transacción y fin transacción.

Para asegurar la integridad de los datos se necesita que el sistema de base de datos mantenga las siguientes propiedades de las transacciones:

Atomicidad: O todas las operaciones de la transacción se realizan adecuadamente en la base de datos o ninguna de ellas.

Consistencia: La ejecución aislada de la transacción(es decir, sin otra transacción que se ejecute concurrentemente) conserva la consistencia de la base de datos.

Aislamiento: Aunque se ejecuten varias transacciones concurrentemente, el sistema garantiza que por cada par de transacciones T_i y T_j se cumple que para los efectos de T_i o bien T_j ha terminado su ejecución antes que comience T_i o bien que T_j ha comenzado su ejecución después que T_i termine. De este modo, cada transacción ignora al resto de las transacciones que se ejecutan concurrentemente en el sistema.

Durabilidad: Tras la finalización con éxito de una transacción, los cambios realizados en la base de datos permanecen, incluso si hay fallos en el sistema.

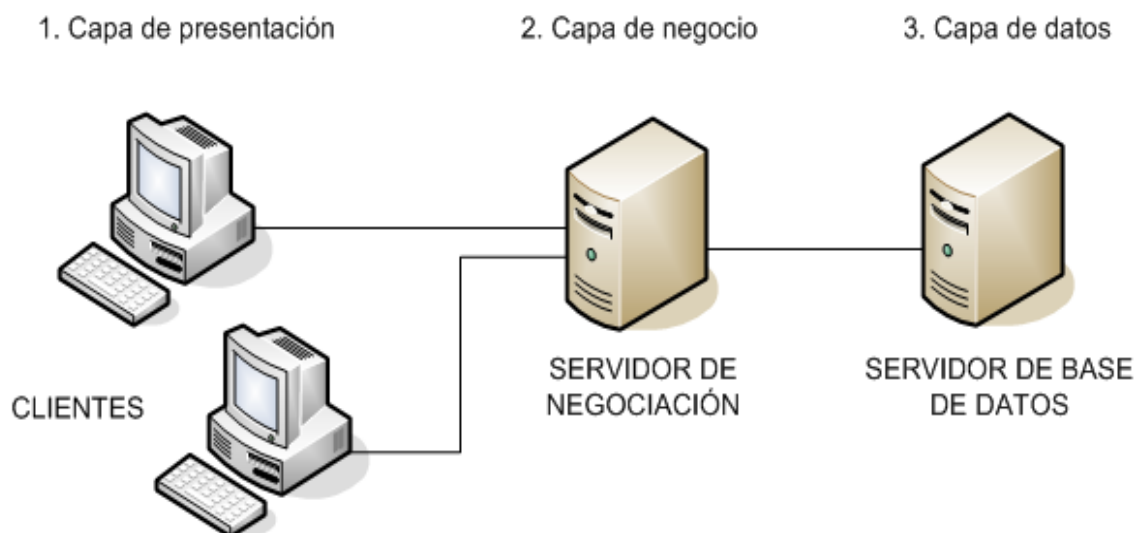


HERENCIA Y POLIMORFISMO.

La Herencia: Es el mecanismo fundamental para implementar la reutilización y extensibilidad del software. A través de ella los diseñadores pueden construir nuevas clases partiendo de una jerarquía de clases ya existente (comprobadas y verificadas) evitando con ello el rediseño, la remodificación y verificación de la parte ya implementada. La herencia facilita la creación de objetos a partir de otros ya existentes, obteniendo características (métodos y atributos) similares a los ya existentes.

El polimorfismo: Es otro de los pilares fundamentales de la programación orientada a objetos. Es la capacidad de almacenar objetos de un determinado tipo en variables de tipo antecesores del primero a costa, claro está, de solo poderse acceder a través de dicha variable a los miembros comunes a ambos tipos. Sin embargo las versiones de los métodos virtuales a las que se llamaría a través de esas variables no serían las definidas como miembros del tipo de dichas variables, sino las definidas en el verdadero tipo de los objetos que almacenan.

MODELO EN TRES CAPAS:



La programación por capas es un estilo de programación en el que el objetivo primordial es la separación de la lógica de negocios de la lógica del diseño; un ejemplo básico de esto consiste en separar la capa de datos de la capa de presentación al usuario.

- La ventaja primordial de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y en caso de que sobrevenga algún cambio, solo se ataca al nivel requerido sin tener que revisar entre código mezclado. Un buen ejemplo de este método de programación sería el modelo de interconexión de sistemas abiertos.
- Además, permite distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, de forma que basta con conocer la API que existe entre niveles.
- En el diseño de sistemas informáticos actual suele ser las arquitecturas multinivel o Programación por capas. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables (que pueden ampliarse con facilidad en caso de que las necesidades aumenten).
- El diseño más utilizado actualmente es el diseño en tres niveles o en tres capas.



Capas y Niveles:

1. Capa de presentación: Es la que ve el usuario (también se le denomina "capa de usuario"), presenta el nivel al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz grafica y debe tener la característica de ser amigable, entendible y fácil de usar para el usuario.
 2. Capa de Negocio: Es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio e incluso de lógica del negocio porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la de presentación, para recibir las solicitudes y presentar los resultados y con la capa de datos, para solicitar el gestor de base de datos para almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.
 3. Capa de Datos: Es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de ases de datos que realizan todo el almacenamiento o recuperación de información desde la capa de negocio.
- Todas estas capas pueden residir en un único ordenador, si bien lo más usual es que haya una multitud de ordenadores en donde reside la capa de presentación (son los clientes de la arquitectura Cliente / Servidor). Las capas de negocio y de datos pueden residir en el mismo ordenador y si el crecimiento de las necesidades lo aconseja que pueden separar en dos o más ordenadores. Así, si el tamaño o complejidad de la base de datos aumenta se puede separar en varios ordenadores los cuales recibirán las peticiones del ordenador en que resida la capa de negocio.
 - Si por el contrario fuese la complejidad en la capa de negocio lo que obligase a la separación, esta capa de negocio podría residir en uno o más ordenadores que realizarían solicitudes a una única base de datos. En sistemas muy complejos se llega a tener una serie de ordenadores sobre los cuales corre la capa de negocio y otra serie de ordenadores sobre los cuales corre la base de datos.



Aclaraciones:

- En una arquitectura de tres niveles, los términos “capas” y “niveles” no significan lo mismo ni son similares.
- El término “capa” hace referencia a la forma como una solución es segmentada desde el punto de vista lógico: Presentación/ Lógica de Negocio/ Datos.
- En cambio el término “nivel” corresponde a la forma en que las capas lógicas se encuentran distribuidas de forma física. Por ejemplo:

Una solución de tres capas (presentación, lógica del negocio, datos) que residen en un solo ordenador (presentación+ lógica+ datos). Se dice que la arquitectura de la solución es de tres capas y un nivel.

Una solución de tres capas (presentación, lógica del negocio, datos) que residen en dos ordenadores (presentación+ lógica+ datos). Se dice que la arquitectura de la solución es de tres capas y dos niveles.

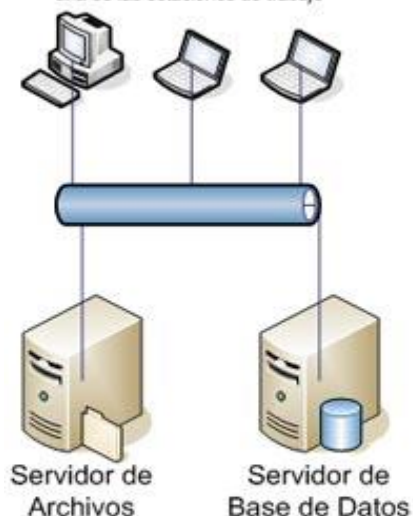
Una solución de tres capas (presentación, lógica del negocio, datos) que residen en tres ordenadores (presentación+ lógica+ datos). Se dice que la arquitectura de la solución es de tres capas y tres niveles.



RED CLIENTE/ SERVIDOR

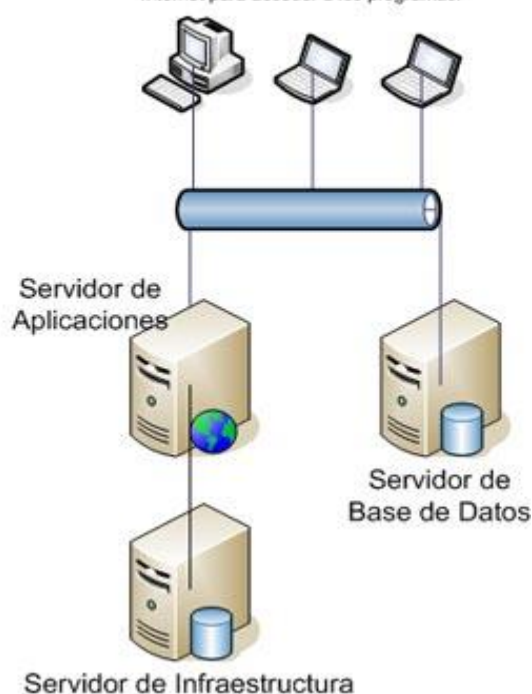
Arquitectura Cliente/Servidor

Los programas se ejecutan en las estaciones de trabajo. Es necesario instalar un ambiente de ejecución en cada una de las estaciones de trabajo



Arquitectura n-capas

Los programas se ejecutan en el servidor de aplicaciones. Las estaciones de trabajo sólo necesitan un navegador de Internet para acceder a los programas.



La red Cliente/Servidor es aquella red de comunicaciones en la que todos los clientes solicitan un servicio a un servidor, en el que se centralizan los diversos recursos y aplicaciones con que se cuenta; y que los pone a disposición de los clientes cada vez que estos son solicitados. Esto significa que todas las gestiones que se realizan se concentran en el servidor, de manera que en él se disponen los requerimientos provenientes de los clientes que tienen prioridad, los archivos que son de uso público y los que son de uso restringido, los archivos que son de sólo lectura y los que, por el contrario, pueden ser modificados, etc. Este tipo de red puede utilizarse conjuntamente en caso de que se esté utilizando en una red mixta.

Una estación de trabajo es una computadora cliente la cual se utiliza para ejecutar aplicaciones.



Características

Cliente:

- Es quien inicia solicitudes o peticiones, tienen por tanto un papel activo en la comunicación (dispositivo **maestro o amo**).
- Espera y recibe las respuestas del servidor.
- Por lo general, puede conectarse a varios servidores a la vez.
- Normalmente interactúa directamente con los usuarios finales mediante una interfaz gráfica de usuario.
- Al contratar un servicio de redes, se tiene que tener en la velocidad de conexión que le otorga al cliente y el tipo de cable que utiliza, por ejemplo: cable de cobre ronda entre 1 mts y 99 mts.

Servidor:

- Al iniciarse esperan a que lleguen las solicitudes de los clientes, desempeñan entonces un papel pasivo en la comunicación (dispositivo *esclavo*).
- Tras la recepción de una solicitud, la procesan y luego envían la respuesta al cliente.
- Por lo general, aceptan conexiones desde un gran número de clientes (en ciertos casos el número máximo de peticiones puede estar limitado).
- No es frecuente que interactúen directamente con los usuarios finales.



ESPECIFICACIÓN DE REQUISITO SOFTWARE.

Análisis de Requisitos Software.

1. Introducción.

1.1. Propósito.

Definir el conjunto de especificaciones de requisitos software que debe cumplir la aplicación Automatización de los procesos de ventas, compras, facturación e inventario de la farmacia ***"LA ASUNCION" del municipio de Telica ,León***, este proyecto se basa en la automatización de las entradas y salidas de los productos, y también el control de cuentas por cobrar y cuentas por pagar. Este documento se dirige 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 nombre con el que se conocerá a esta aplicación será: Sistema de automatización de los procesos de ventas, compras, facturación e inventario, para la farmacia LA ASUNCION, del municipio de Telica, León.

Esta aplicación realizara las siguientes funciones:

- 1) Registrar Nuevo Cliente.
- 2) Registrar Nueva Compra.
- 3) Registrar Nuevo Producto.
- 4) Registrar Nuevo Proveedor.
- 5) Registrar Nuevo Usuario.
- 6) Registrar Nueva Venta (crédito, contado).
- 7) Editar Cliente.
- 8) Editar Usuario.



- 9) Editar Producto.
- 10) Editar Proveedor.
- 11) Buscar Clientes.
- 12) Buscar Compra.
- 13) Buscar Producto.
- 14) Buscar Ventas.
- 15) Buscar Proveedores.
- 16) Buscar Usuario.
- 17) Reporte de Productos controlados.
- 18) Reporte de Ventas al Día.
- 19) Reporte de Inventario.
- 20) Reporte de Ventas al Mes.
- 21) Reporte de Productos Vencidos.
- 22) Reporte de Cuentas por Cobrar.
- 23) Reporte de Cuentas por Pagar.
- 24) Visualizar Facturas.
- 25) Visualizar Calculadora.
- 26) Respalidar Base de Datos.
- 27) Restaurar Base de Datos.
- 28) Configuración del servidor.
- 29) Visualizar Calendario.
- 30) Visualizar Manual de Usuario.
- 31) Cambiar Contraseña.

La aplicación no se encargara de registrar los pagos a los trabajadores.



1.3. Definiciones, acrónimos y abreviaturas.

- **Administrador:** Persona que tendrá todos los privilegios en la aplicación.
- **Proveedor:** Puede ser una persona o una empresa que abastece a otras empresas con existencias (artículos), los cuales serán transformados para venderlos posteriormente o directamente se compran para su venta.
- **Cliente:** Empresa o persona que realiza la compra de los productos
- **Nombre Comercial:** Nombre empleado para distinguir un principio activo que no está amparado por una marca de fábrica. . Es usado comúnmente por diversos fabricantes y reconocido por la autoridad competente para denominar producto farmacéutico que contienen el mismo principio activo.
- **Nombre Genérico:** Nombre empleado para distinguir un principio activo que no está amparado por una marca de fábrica. Es usado comúnmente por diverso fabricante y reconocido por la autoridad competente para denominar producto farmacéutico que contienen el mismo principio activo. El nombre genérico generalmente con la denominación común internacional.
- **Concentración:** Es el contenido de principio activo en masa (peso) o volumen, expresado en unidades del sistema internacional de unidades (SI) o en unidades Internacionales (UI), en función de la forma farmacéutica.
- **Forma farmacéutica:** Es la forma física que se le da a un medicamento para facilitar la administración del producto al paciente.
- **Reporte:** Informe que se emite o presenta con base en la realización de una actividad o tarea.
- **Cuentas por cobrar:** Cobro a clientes los cuales se encuentran pendientes de pago de artículos a la empresa, después del plazo establecido por el dueño de la empresa.
- **Cuentas por pagar:** Facturas pendientes por pagar a proveedores.
- **Inventario:** Cantidad de productos existente en bodega.



1.4. Visión General.

Primeramente se realizara una descripción general del artículo que se desea desarrollar para pasar posteriormente a estudiar cada uno de los requisitos específicos individualmente.

2. Descripción general.

2.1. Relaciones del producto.

La aplicación trabajara con una base de datos FARMACIA_DB, que existe en la computadora donde se ejecutara la aplicación.

El equipo en que se implantara el producto final es:

- Procesador AMD E-240 1.5 GHz
- 2 GB de RAM.
- HDD 250 GB.
- Impresora HP Deskjet F2280.

2.2. Funciones del producto.

La aplicación debe contener todas las tareas que realiza el personal manualmente, agregando un módulo de seguridad.

- Cuando la aplicación inicie, solicitara un usuario y contraseña, para validar el acceso.
- Si el administrador desea brindar el acceso a otra persona, se encargara de crear un usuario en el sistema.
- Si el usuario tiene los privilegios correspondientes podrá cambiar su contraseña, sino lo hará el administrador.
- Si hay un nuevo cliente, al cual se le brindara crédito, se podrá registrar a este cliente.
- Cuando se haga una compra al proveedor, se deberá registrar.
- Si la empresa adquiere un nuevo producto, el usuario lo podrá ingresar a la base de datos.



- Si hay un nuevo proveedor se deberá registrar.
- El administrador podrá realizar el ingreso de un nuevo usuario en el sistema.
- Cuando un cliente haga una compra, se deberá registrar la venta.
- El administrador podrá editar la información de un cliente cuando sea necesario.
- El administrador podrá editar la información de un usuario cuando considere adecuado.
- El administrador podrá cambiar los datos de un producto cuando considere conveniente.
- El usuario podrá editar la venta facturada al cliente al efectuarse un error.
- El administrador podrá editar la información de un proveedor.
- Se podrá hacer la búsqueda de datos de un determinado cliente.
- Se podrá hacer la búsqueda de una compra específica.
- Se podrá hacer la búsqueda de información de un determinado producto.
- El administrador podrá hacer la búsqueda de una determinada venta.
- El sistema podrá buscar la compra realizada al proveedor.
- El administrador podrá buscar la información definida de un proveedor.
- El sistema tendrá la facultad de buscar un usuario.
- El sistema deberá generar un informe de la lista de los productos de tipo droga que se han vendido mensualmente.
- Cuando el usuario desee conocer el ingreso que ha tenido la empresa, el sistema generara un reporte con las ventas realizadas diariamente y al mes.
- Cuando el administrador desee conocer la cantidad de existencias de los productos, se generara un informe.
- Cuando el administrador desee saber que productos están pronto a vencer, generara un informe con la lista de productos vencidos.



- Cuando el administrador desee saber que clientes le deben a la empresa, el sistema generara un informe detallado con la información correspondiente.
- Cuando el administrador desee saber cuáles son las cuentas y a quienes tiene que pagar, el sistema deberá generar un informe.
- Se podrá realizar una copia de seguridad de toda la información de la base de datos, así como restaurarla.
- Se podrá acceder al manual de usuario que brindara una ayuda a la manipulación correcta y de las opciones que presenta el sistema.

2.3. Características del usuario.

Los usuarios finales de la aplicación serán personas cuya experiencia informática es media, se hará y entregara un manual de usuario, sobre cómo manejar el software.

2.4. Restricciones generales.

El lenguaje de programación a utilizar será C# 2010 .NET, con una conexión a una base de datos MySQL. Se deberá seguir el patrón de programación MVC (Modelo Vista Controlador). El idioma de la aplicación será en español.

2.5. Suposiciones y dependencias.

El administrador de la farmacia adquirirá otro equipo para que los clientes puedan buscar un determinado producto.



3. Requisitos Específicos.

3.1. Requisitos funcionales.

3.1.1. Registrar Nuevo Cliente.

3.1.1.1. Especificación.

3.1.1.1.1. Introducción.

Se solicitaran los datos del nuevo cliente y guardaran en la base de datos.

3.1.1.1.2. Entradas.

Por pantalla: datos del cliente.

- Cedula.
- Nombres.
- Apellidos.
- Dirección.
- Teléfono.
- Email.

3.1.1.1.3. Proceso.

Por medio del Menú Registrar, el Administrador podrá ingresar un nuevo cliente.

- Cedula: campo obligatorio.
- Nombres: campo obligatorio.
- Apellido: campo obligatorio.
- Dirección: campo obligatorio.
- Teléfono: campo obligatorio.
- Email: campo obligatorio.

3.1.1.1.4. Salidas.

Se guardara la información en la base de datos.



3.1.2. Registrar Nueva Compra.

3.1.2.1. Especificación.

3.1.2.1.1. Introducción.

Este proceso se encargará de capturar la compra a un proveedor, ya sea al crédito o al contado.

3.1.2.1.2. Entradas.

Por pantalla datos de compra:

- Código compra.
- Nombre de proveedor.
- Tipo de Pago.
- Fecha de compra.
- Monto.
- Descuento Adquirido.
- Total: El sistema realizará esa operación.

3.1.2.1.3. Proceso.

Por medio del Menú Registrar, el Administrador tendrá la opción de ingresar una nueva Compra a un proveedor, el sistema le solicitará:

- Código de compra.
- Nombre de proveedor.
- Tipo de pago.
- Fecha de compra.
- Monto.
- Descuento Adquirido.
- Total: El sistema realizará el cálculo del total.

3.1.2.1.4. Salidas.

Se guardará la información en la base de datos.



3.1.3. Registrar Nuevo Producto.

3.1.3.1. Especificación.

3.1.3.1.1. Introducción.

Cuando la empresa adquiriera un nuevo producto, el Administrador lo podrá registrar.

3.1.3.1.2. Entradas.

Por pantalla: datos del producto.

- Nombre Comercial.
- Nombre Genérico.
- Presentación.
- Concentración.
- Nombre Proveedor.
- Nombre Laboratorio.
- Tipo Producto.
- Cantidad.
- Precio de costo.
- Precio de venta.
- Fecha Ingreso.
- Fecha Vence.

3.1.3.1.3. Proceso.

Por medio del Menú Registrar, el Administrador podrá ingresar los datos de un nuevo producto, aparecerá un formulario Solicitando:

- Nombre Comercial.
- Nombre Genérico.
- Presentación.
- Concentración.
- Nombre Proveedor.
- Nombre Laboratorio.
- Tipo de producto.



- Cantidad.
- Precio de costo.
- Precio de venta.
- Fecha ingreso: la fecha en que el producto es adquirido por la empresa.
- Fecha Vencimiento.

3.1.3.1.4. Salidas.

Se guardara la información en la base de datos.

3.1.4. Registrar Nuevo Proveedor.

3.1.4.1. Especificación.

3.1.4.1.1. Introducción.

Este proceso se encargara de registrar un nuevo proveedor.

3.1.4.1.2. Entradas.

Por pantalla: datos del proveedor:

- Nombre proveedor.
- Dirección.
- Teléfono.
- Correo Electrónico.
- Fax.
- Siglas.

3.1.4.1.3. Proceso.

Por medio del Menú Registrar el Administrador tendrá la opción de ingresar un nuevo proveedor, el sistema le solicitara toda la información anteriormente mencionada.

3.1.4.1.4. Salidas.

Toda la información se guardará en la base de datos.



3.1.5. Registrar Nuevo Usuario.

3.1.5.1. Especificación.

3.1.5.1.1. Introducción.

Se solicitarán los datos del nuevo usuario y se guardarán en la base de datos.

3.1.5.1.2. Entradas.

Por pantalla: datos del usuario:

- Cedula.
- Nombres.
- Apellidos.
- Username.
- Contraseña.
- Dirección.
- Teléfono.
- Email.
- Perfil de usuario.
- ID usuario: el sistema lo brinda automáticamente.
- Estado.

3.1.5.1.3. Proceso.

Por medio del Menú Registrar, el Administrador podrá ingresar un nuevo usuario.

- Cedula: campo obligatorio.
- Nombres: campo obligatorio.
- Apellidos: campo obligatorio.
- Username: nombre de usuario que se brindara para acceder al sistema.
- Contraseña: conjunto de caracteres asignados al usuario para validar su acceso.
- Dirección: campo obligatorio.



- Teléfono: campo obligatorio.
- Email: campo obligatorio.

Por parte del sistema:

- ID usuario.
- Perfil Usuario, Estado se tendrán que especificar al momento de registrar los datos.

3.1.5.1.4. Salidas.

Toda la información se guardara en la base de datos.

3.1.6. Cambiar contraseña de Usuario.

3.1.6.1. Especificación.

3.1.6.1.1. Introducción.

El Administrador podrá cambiar la contraseña de acceso del usuario, cuando él considere necesario.

3.1.6.1.2. Entradas.

Por pantalla: datos de usuario.

- Contraseña actual.
- Contraseña Nueva.
- Confirmar Contraseña.

3.1.6.1.3. Proceso.

El usuario tiene que acceder al sistema, habrá una opción en el menú usuario, que le permitirá cambiar su contraseña, solicitándole:

- Contraseña actual: clave de acceso actual, que le brindo el administrador.
- Contraseña Nueva: la nueva clave de acceso que desea el usuario.



- Confirmar contraseña: confirmar la nueva contraseña, para evitar inconvenientes.

3.1.6.1.4. Salidas.

Se guardará la información en la base de datos.

3.1.7. Registrar Venta (crédito, contado).

3.1.7.1. Especificación.

3.1.7.1.1. Introducción.

Este proceso se encargara de capturar una venta a un cliente.

3.1.7.1.2. Entradas.

Si el tipo de pago es de contado, por pantalla:

Selección del producto y la cantidad de producto que desea comprar.

- Por parte del sistema:

Se realizaran las operaciones necesarias y se mostrará un resultado.

Si el tipo de pago es al crédito, por pantalla:

Solamente ingresar el número de cedula.

Por parte del sistema:

- Se realizaran las operaciones necesarias y se mostrará un resultado.

3.1.7.1.3. Proceso.

El proceso podrá registrar la venta a un cliente dependiendo del tipo de pago que realice el cliente. Si es al contado el sistema solamente solicitara el producto que desee comprar.

Si es al crédito, el cliente tiene que estar registrado en el sistema y solamente ingresar el número de cedula, el sistema mostrara toda la información del cliente. Si no aparece el cliente, el usuario deberá registrarlo.



3.1.7.1.4. Salidas.

Toda la información se guardara en la base de datos.

3.1.8. Editar Datos del Cliente.

3.1.8.1. Especificación.

3.1.8.1.1. Introducción.

Este proceso le permitirá al Administrador cambiar algún dato de un determinado cliente.

3.1.8.1.2. Entradas.

Por pantalla: datos del cliente:

- Cedula.
- Nombres.
- Apellidos.
- Dirección domicilio.
- Numero de contacto.
- Email Actual.
- Estado

3.1.8.1.3. Proceso.

Por medio del Menú Editar, el Administrador tendrá la opción de cambiar algún dato de un determinado cliente, ingresando la cedula en el sistema aparecerán todos los datos del cliente registrado.

3.1.8.1.4. Salidas.

Se guardará toda la información en la base de datos. Aparecerá un mensaje indicando si la operación fue exitosa.



3.1.9. Editar Datos de Usuario.

3.1.9.1. Especificación.

3.1.9.1.1. Introducción.

Este proceso le permitirá al Administrador cambiar algún dato de un determinado usuario.

3.1.9.1.2. Entradas.

Por pantalla: datos de usuario:

- Cedula.
- Nombres.
- Apellidos.
- Nombre de usuario.
- Dirección de domicilio.
- Password nuevo.
- Numero de contacto.
- Email actual.
- Perfil.
- Estado.

3.1.9.1.3. Proceso.

Por medio del Menú Editar, el Administrador tendrá la opción de cambiar algún dato de un determinado usuario, ingresando la cedula en el sistema aparecerán todos los datos del usuario registrado.

3.1.9.1.4. Salidas.

Se guardará toda la información en la base de datos. Aparecerá un mensaje si la operación fue exitosa.



3.1.10. Editar Datos de Producto.

3.1.10.1. Especificación.

3.1.10.1.1. Introducción.

Este proceso se encargara de cambiar los datos de un determinado producto.

3.1.10.1.2. Entradas.

Por pantalla: una caja de texto en blanco que nos permitirá hacer la búsqueda y selección del producto.

3.1.10.1.3. Proceso.

Por medio del Menú Editar, el Administrador podrá cambiar algún dato de un determinado producto. Aparecerá en la pantalla una caja de texto en blanco y escogerá el producto que desea editar, en donde el Administrador deberá ingresar el nombre comercial o genérico, el sistema mostrara las coincidencias registradas en la base de datos.

3.1.10.1.4. Salidas.

Se mostrara en la pantalla: información del producto.

- Nombre comercial.
- Nombre genérico.
- Presentación.
- Concentración.
- Laboratorio.
- Ingreso.
- Vencimiento.
- Tipo.
- Existencia.
- Precio costo.
- Precio venta.



3.1.11. Editar Datos del Proveedor.

3.1.11.1. Especificación.

3.1.11.1.1. Introducción.

Este proceso se encargara de cambiar algún dato de un determinado proveedor.

3.1.11.1.2. Entradas.

Por pantalla: datos del proveedor.

- Nombre proveedor.
- Dirección.
- Teléfono.
- Email.
- Fax.
- Siglas.

3.1.11.1.3. Proceso.

Por medio del Menú Editar el Administrador podrá Editar una información de un determinado proveedor cuando este lo crea conveniente.

Aparecerá en la pantalla una caja de texto en blanco, donde deberá ingresar el nombre del proveedor, el sistema mostrara una lista de proveedores existentes y el usuario elegirá el proveedor que desee editar.

3.1.11.1.4. Salidas.

Se guardara la información en la base de datos.



3.1.12. Buscar Datos de Cliente.

3.1.12.1. Especificación.

3.1.12.1.1. Introducción.

Este proceso se encargara de buscar a un cliente existente en el sistema.

3.1.12.1.2. Entradas.

Por pantalla: una caja de texto en blanco.

3.1.12.1.3. Proceso.

Cuando el Administrador desee buscar un cliente, lo podrá realizar por medio del Menú Buscar, aparecerá en la pantalla una caja de texto en blanco, en donde se hará la búsqueda por coincidencia en la base de datos, se mostrara.

3.1.12.1.4. Salidas.

Se mostrara en la pantalla: información de un cliente.

- Cedula.
- Nombres.
- Apellidos.
- Dirección.
- Teléfono.
- Correo electrónico.
- Estado.



3.1.13. Buscar Compras.

3.1.13.1. Especificación.

3.1.13.1.1. Introducción.

Este proceso se encargara de realizar una búsqueda de una determinada compra.

3.1.13.1.2. Entradas.

Por pantalla: una caja de texto en blanco.

3.1.13.1.3. Proceso.

Por medio del menú Buscar, el usuario tendrá la opción de buscar una compra, por el nombre del proveedor.

3.1.13.1.4. Salidas.

Se mostrara por pantalla toda la información de la compra.

- Código compra.
- Fecha.
- Proveedor.
- Tipo pago.
- Estado.
- Monto.
- Descuento.
- Total compra.

3.1.14. Buscar Producto.

3.1.14.1. Especificación.

3.1.14.1.1. Introducción.

Este proceso le permitirá al Administrador/Usuario hacer una búsqueda de un producto.

3.1.14.1.2. Entradas.

Por pantalla:

- Caja de texto en blanco.



3.1.14.1.3. Proceso.

Por medio del menú Buscar, el usuario tendrá la opción de buscar un producto solo ingresando su nombre.

3.1.14.1.4. Salidas.

Por pantalla: Datos del producto.

- Código producto.
- Nombre comercial.
- Nombre genérico.
- Presentación.
- Concentración.
- Nombre proveedor.
- Nombre laboratorio.
- Fecha ingreso.
- Fecha vencimiento.
- Tipo.
- Cantidad.
- Precio costo.
- Precio venta.

3.1.15. Buscar Ventas.

3.1.15.1. Especificación.

3.1.15.1.1. Introducción.

Este proceso se encargara de buscar una venta realizada.

3.1.15.1.2. Entradas.

Por pantalla:

- Caja de texto en blanco.



3.1.15.1.3. Proceso.

Por medio del Menú Buscar, el Administrador tendrá la opción de buscar una venta, aparecerá una caja de texto en blanco, en donde el usuario podrá ingresar el nombre del producto vendido. El sistema se encargara de mostrar las coincidencias de lo escrito en la base de datos.

3.1.15.1.4. Salidas.

Se mostrara por pantalla toda la información de la venta:

- Código factura.
- Cedula cliente.
- Nombre genérico.
- Cantidad producto.
- Precio unitario.
- Cedula usuario.

3.1.16. Buscar Proveedores.

3.1.16.1. Especificación.

3.1.16.1.1. Introducción.

Este proceso efectuara la búsqueda de un proveedor existente.

3.1.16.1.2. Entradas.

Por pantalla:

- Caja de texto en blanco.

3.1.16.1.3. Proceso.

Cuando el Administrador desee buscar un proveedor, lo podrá realizar por medio del Menú Buscar, aparecerá en la pantalla una caja de texto en blanco, en donde se hará la búsqueda por coincidencias en la base de datos, se mostrara.

3.1.16.1.4. Salidas.

Se mostrara en la pantalla: información del proveedor.



- Nombre proveedor.
- Dirección.
- Teléfono.
- E-mail.
- Fax.
- Siglas.

3.1.17. Buscar Usuario.

3.1.17.1. Especificación.

3.1.17.1.1. Introducción.

Este proceso realizar la búsqueda de un usuario existente en el sistema.

3.1.17.1.2. Entradas.

Por pantalla:

- Caja de texto en blanco.

3.1.17.1.3. Proceso.

Cuando el Administrador desee buscar un usuario, lo podrá realizar por medio del Menú Buscar, en donde aparecerá en la pantalla una caja de texto en blanco, donde se hará la búsqueda por coincidencias en la base de datos, se mostrara.

3.1.17.1.4. Salidas.

Se mostrara en la pantalla: información del usuario.

- Cedula usuario.
- Nombres.
- Apellidos.
- Username.
- Dirección.
- Teléfono.
- Correo electrónico.



- Id usuario.
- Perfil.
- Estado.

3.1.18. Reporte de Productos Controlados.

3.1.18.1. Especificación.

3.1.18.1.1. Introducción.

Este proceso se encargara de generar un informe con los productos vendidos de tipo droga mensualmente.

3.1.18.1.2. Entradas.

Por pantalla: aparecerá una ventana en donde tendremos que ingresar el mes que deseamos generar el reporte.

3.1.18.1.3. Proceso.

Por medio del Menú Generar el Administrador tendrá la opción de generar un reporte de los productos tipo droga vendidos en el mes.

3.1.18.1.4. Salidas.

Un documento impreso.

3.1.19. Reporte de Ventas al Día.

3.1.19.1. Especificación.

3.1.19.1.1. Introducción.

Este proceso se encargara de generar un informe con las ventas efectuadas en el día.

3.1.19.1.2. Entradas.

Por pantalla:

- Aparecerá un calendario en el cual seleccionamos la fecha de la venta.

3.1.19.1.3. Proceso.



Por medio del Menú Generar, el Administrador tendrá la opción de generar un informe detallado con las ventas realizadas en el día.

3.1.19.1.4. Salidas.

Un documento impreso en el cual aparecerá los siguientes datos:

- Factura.
- Fecha.
- Precio total.
- Cliente.
- Nombre producto.
- Cantidad.
- Precio.
- Total vendido.

3.1.20. Reporte de Inventario.

3.1.20.1. Especificación.

3.1.20.1.1. Introducción.

Este proceso se encargara de generar un informe con la cantidad existente de todos los productos.

3.1.20.1.2. Entradas.

Aparecerá el reporte automáticamente al dar click en Inventario.

3.1.20.1.3. Proceso.

Por medio del Menú Generar, el Administrador tendrá la opción de generar el reporte de la cantidad existente de todos los productos.

3.1.20.1.4. Salidas.

Un documento impreso en donde aparecerá la siguiente información:

- Tipo.
- Código producto.
- Nombre comercial.



- Nombre genérico.
- Presentación.
- Proveedor.
- Ingreso.
- Vencimiento.
- Existencia.
- Precio costo.
- Precio venta.

3.1.21. Reporte de Ventas al Mes.

3.1.21.1. Especificación.

3.1.21.1.1. Introducción.

Este proceso se encargara de crear un informe con las ventas realizadas en el mes.

3.1.21.1.2. Entradas.

Por pantalla:

- Aparecerá un calendario en donde escogeremos la fecha de inicio y la fecha de fin para saber las ventas realizadas en un determinado mes.

3.1.21.1.3. Proceso.

Por medio del Menú Generar, el Administrador tendrá la opción de generar un informe detallado con las ventas realizadas al mes.

3.1.21.1.4. Salidas.

Un documento impreso con la siguiente información:

- Factura.
- Fecha.
- Cedula cliente.
- Nombre producto.
- Cantidad.



- Precio.
- Total factura.

3.1.22. Reporte de Productos Vencidos.

3.1.22.1. Especificación.

3.1.22.1.1. Introducción.

Este proceso se encargara de generar un informe con una lista de los productos que están pronto a vencer.

3.1.22.1.2. Entradas.

Por pantalla:

- Aparecerá directamente el reporte.

3.1.22.1.3. Proceso.

Por medio del menú Generar, el Administrador podrá crear un informe detallado con la información de los productos que están pronto a vencer.

3.1.22.1.4. Salidas.

Un documento impreso con toda la información:

- Tipo.
- Código producto.
- Nombre comercial.
- Nombre genérico.
- Presentación.
- Proveedor.
- Ingreso.
- Vencimiento.
- Existencia.
- Precio costo.
- Precio venta.



3.1.23. Reporte de Cuentas por Cobrar.

3.1.23.1. Especificación.

3.1.23.1.1. Introducción.

Este proceso se encargara de generar un informe con las cuentas por cobrar a los clientes al crédito.

3.1.23.1.2. Entradas.

Por pantalla:

- Aparecerá el reporte directamente al dar click en cuentas por cobrar.

3.1.23.1.3. Proceso.

Por medio del Menú Generar, el Administrador podrá crear un informe detallado de las cuentas por cobrar a un determinado cliente.

3.1.23.1.4. Salidas.

Un documento impreso con la siguiente información:

- Cedula.
- Nombres.
- Apellidos.
- Dirección.
- Monto.
- Total a cobrar.

3.1.24. Reporte de Cuentas por Pagar.

3.1.24.1. Especificación.

3.1.24.1.1. Introducción.

Este proceso se encargara de generar un informe con una lista de los cuentas por pagar a proveedores.

3.1.24.1.2. Entradas.

Por pantalla:



- Aparecerá directamente el reporte al dar click en cuantas por pagar.

3.1.24.1.3. Proceso.

Por medio del Menú Generar reporte de cuentas por pagar, el Administrador podrá crear un informe detallado con la información de las cuentas que hay que pagar a un determinado proveedor.

3.1.24.1.4. Salidas.

Un documento impreso con la siguiente información:

- N° de compra.
- Fecha.
- Proveedor.
- Monto.
- Descuento.
- Total.

3.1.25. Visualizar Facturas.

3.1.25.1. Especificación.

3.1.25.1.1. Introducción.

Este proceso se encargara de generar un informe en donde se visualizaran las facturas comprendidas en una determinada fecha.

3.1.25.1.2. Entradas.

Por pantalla:

- Fecha inicial.
- Fecha fin.

3.1.25.1.3. Proceso.

Por medio del menú Generar, el Administrador podrá tener acceso a la visualización de las facturas en un determinado periodo de tiempo.



3.1.25.1.4. Salidas.

Un documento impreso con toda la información solicitada.

3.1.26. Visualizar Calculadora.

3.1.26.1. Especificación.

3.1.26.1.1. Introducción.

Este proceso se encargara de mostrar por pantalla una pequeña calculadora con las funciones básicas.

3.1.26.1.2. Entradas.

Por pantalla:

- Datos para calcular.

3.1.26.1.3. Proceso.

Se encargara de hacer las funciones que el usuario necesite realizar ya sea suma, resta, multiplicación, división, porcentaje.

3.1.26.1.4. Salidas.

Se mostrara por pantalla el cálculo que el usuario solicito.

3.1.27. Respaldo base de datos.

3.1.27.1. Especificación.

3.1.27.1.1. Introducción.

Por medio de Menú Generar, el Administrador tendrá la opción de elegir hacer una copia de seguridad de la base de datos.

3.1.27.1.2. Entradas.

Por pantalla:

- Nombre del archivo.
- Ruta de almacenaje.

3.1.27.1.3. Proceso.

Aparecerá una ventana solicitando la siguiente información:



- Nombre del archivo: Puede ser cualquier nombre que le podemos asignar.
- Ruta de almacenaje: Es donde se almacenara la copia de la base de datos.

3.1.27.1.4. Salidas.

Se mostrara un mensaje indicando si la operación fue satisfactoria de lo contrario aparecerá un error.

3.1.28. Restaurar Base de Datos.

3.1.28.1. Especificación.

3.1.28.1.1. Introducción.

Por medio del Menú Generar, el Administrador tendrá la opción de hacer una restauración de la base de datos.

3.1.28.1.2. Entradas.

Por pantalla:

- Ruta y nombre del archivo.

3.1.28.1.3. Proceso.

Se mostrara la opción de restaurar la base de datos, solicitando:

- Ruta y nombre del archivo: Primero se indica la ruta en donde se almacenara la base de datos y luego el nombre que desee.

3.1.28.1.4. Salidas.

Aparecerá un mensaje indicando si la operación fue satisfactoria de lo contrario un mensaje que ha ocurrido un error.



3.1.29. Configuración.

3.1.29.1. Especificación.

3.1.29.1.1. Introducción.

Este proceso se encargara de realizar una configuración al servidor para poder tener acceder a la aplicación.

3.1.29.1.2. Entradas.

Por pantalla:

- Dirección IP o Nombre del servidor.
- Contraseña de acceso.

3.1.29.1.3. Proceso.

Al dar click en configuración el formulario pide la dirección IP o el nombre de la máquina que es el servidor, también solicita la contraseña para acceder, si alguno de los datos es introducido de manera incorrecta la aplicación no se va a conectar con la base de datos es decir con el servidor.

3.1.29.1.4. Salidas.

Se guardara la nueva configuración.

3.1.30. Visualizar Calendario.

3.1.30.1. Especificación.

3.1.30.1.1. Introducción.

Este proceso se encargara de mostrar un pequeño calendario.

3.1.30.1.2. Entradas.

Ninguna.

3.1.30.1.3. Proceso.

Por medio del Menú Herramientas, el usuario podrá mostrar el calendario.

3.1.30.1.4. Salidas.

Ninguna.



3.1.31. Visualizar Manual de Usuario.

3.1.31.1. Especificación.

3.1.31.1.1. Introducción.

Este proceso se encargara de mostrar el manual de usuario en donde aparecerán los detalles de cómo funciona la aplicación.

3.1.31.1.2. Entradas.

Ninguna.

3.1.31.1.3. Proceso.

En la parte superior derecha donde dice Ayuda el usuario podrá acceder al manual de usuario.

3.1.31.1.4. Salidas.

Se podrá visualizar la información de cómo funciona la aplicación.



3.2. Requisitos de funcionamiento.

Requisitos dinámicos: Es importante que el tiempo de respuesta no aumente exponencialmente con el número de usuarios.

3.3. Restricciones de diseño.

El formato de pantalla e informes de la aplicación deberá contener información acerca del nombre de la empresa, el nombre del usuario que realiza el trabajo, la fecha y la hora del trabajo.

3.4. Atributos.

3.4.1. Seguridad.

Solamente tendrán acceso al sistema las personas que posean un usuario y contraseña válidos.

3.5. Otros requisitos.

3.5.1. Base de datos.

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

3.5.2. Operaciones.

Todas las operaciones sobre la base de datos se realizaran según lo mencionado en el subapartado.



CASOS DE USO EXTENDIDO.

1. Identificación de Actores.

Actores

Usuario: Es quien está encargado de la venta de medicamentos en la farmacia, requiere servicios de la aplicación.

Administrador: Perfiles: Define derechos y restricciones de acceso a determinadas funciones y operaciones del sistema de acuerdo con el perfil.

Privilegios: Asigna a un empleado determinada clasificación en función con la labor que tenga asignada, para así administrar los permisos como registrar, editar, buscar, eliminar, generar, respaldar la base de dato, requiere servicio de la aplicación.

Personal: Registra a todos los empleados existentes en la farmacia estableciendo el perfil y la contraseña.

Glosario de casos de uso del sistema.

Caso de uso	Registrar al cliente.
Actores	Administrador
Propósito	Almacenar los datos del cliente.
Precondición	Los campos determinados sean llenados correctamente.
Post condición	Reconocer al cliente
Flujo básico de eventos	
Actor	Respuesta del sistema
1. El administrador modifica o llena el formulario con los datos requeridos del cliente nuevo.	2. Verificara que la información sea la correcta. 3. Registra el formulario en la base de dato.
Flujo alterno	
1. Si el administrador no ingresa correctamente los datos en el formulario el sistema mandara una advertencia e indicara el error, ingresar de nuevo los datos.	



Caso de uso	Registrar las compras.
Actores	Administrador.
Propósito	Llevar registro de las compras efectuadas.
Precondición	Los campos determinados sean llenados correctamente.
Post condición	Almacenar en la base de datos.
Flujo básico de eventos	
Actor	Respuesta del sistema
1. El administrador ingresa al formulario los datos requeridos en la aplicación.	2. Verifica que la información este correcta. 3. Registra la compra realizada en la base de datos.
Flujo alterno	
1. Si el administrador no ingresa correctamente la información requerida el sistema volverá a pedir la información.	

Caso de uso	Registrar producto.
Actores	Administrador.
Propósito	Tener control de los medicamentos existentes.
Precondición	Los campos determinados sean llenados correctamente.
Post condición	Almacenar en la base de datos.
Flujo básico de eventos	
Actor	Respuesta del sistema
1. El administrador ingresa al formulario los datos requeridos en la aplicación.	2. Filtra la información a la base de datos y se le asigna un código.
Flujo alterno	
1. Si el administrador no ingresa correctamente los datos en el formulario el sistema mandara una advertencia e indicara el error, ingresar de nuevo los datos.	



Caso de uso	Registrar al proveedor.
Actores	Administrador.
Propósito	Identificar que producto le corresponde a cada proveedor.
Precondición	Los campos estén ingresados correctamente.
Post condición	Reconocer al proveedor.
Flujo básico de eventos	
Actor	Respuesta del sistema
1. El administrador ingresa al formulario los datos requeridos en la aplicación.	2. Verifica que la información este correcta. 3. Almacena la información del nuevo proveedor en la base de dato.
Flujo alterno	
1. Si los cambios no se efectúan correctamente mandara una advertencia e indicara el error y no se guardara correctamente la información.	

Caso de uso	Registrar al usuario.
Actores	Administrador
Propósito	Almacenar los datos del usuario.
Precondición	Los campos determinados sean llenados correctamente.
Post condición	Reconocer al usuario.
Flujo básico de eventos	
Actor	Respuesta del sistema
1. El administrador tiene dos áreas. Primero: modifica o llena el formulario con los datos requeridos del usuario nuevo.	3. Verificara que la información sea la correcta.
2. Segundo: otorgará los permisos requeridos dependiendo del usuario en la aplicación.	4. Registra el formulario en la base de dato y le asigna un código si es nuevo.
Flujo alterno	
1. Si el administrador no ingresa correctamente los datos en el formulario el sistema mandara una advertencia e indicara el error, ingresar de nuevo los datos.	



Caso de uso	Registrar las ventas.
Actores	Administrador/usuario.
Propósito	Llevar registro de las ventas efectuadas.
Precondición	Los campos determinados sean llenados correctamente.
Post condición	Almacenar en la base de datos.
Flujo básico de eventos	
Actor	Respuesta del sistema
1. El administrador/usuario ingresa al formulario los datos requeridos en la aplicación.	2. según el nombre del medicamento el sistema visualizara el costo. 3. se podrá editar la venta en caso de un error del usuario. 4. Registra la venta realizada en el reporte de ventas al día y reporte de ventas al mes automáticamente en la base de datos.
Flujo alterno	
1. Si el administrador / usuario no ingresa correctamente la información requerida el sistema volverá a pedir la información.	

Caso de uso	Editar los datos de cliente.
Actores	Administrador.
Propósito	Modificar los datos del cliente.
Precondición	Estar registrado en base de datos.
Post condición	Actualizar los datos del cliente.
Flujo básico de eventos	
Actor	Respuesta del sistema
1. El administrador modifica los datos necesarios del cliente.	2. El sistema requiere la cedula del cliente para mostrar su información. 3. El sistema muestra el cliente ya existente al cual se modificara y guarda los nuevos cambios.
Flujo alterno	
1. Si el administrador no ingresa correctamente la información requerida el sistema volverá a pedir la información.	



Caso de uso	Editar los datos de usuario.
Actores	Administrador.
Propósito	Modificar los datos del usuario.
Precondición	Estar registrado en base de datos.
Post condición	Actualizar los datos del usuario.
Flujo básico de eventos	
Actor	Respuesta del sistema
1. El administrador modifica los datos necesarios del usuario.	2. El sistema requiere la cedula del usuario para mostrar su información. 3. El sistema mostrara al usuario ya existente al cual se modificara y guarda los nuevos cambios.
Flujo alterno	
1. Si el administrador no ingresa correctamente los datos en el formulario el sistema mandara una advertencia e indicara el error, ingresar de nuevo los datos.	

Caso de uso	Editar los datos de producto.
Actores	Administrador.
Propósito	Modificar los datos del producto.
Precondición	Estar registrado en base de datos.
Post condición	Actualizar los datos del producto.
Flujo básico de eventos	
Actor	Respuesta del sistema
1. El administrador modifica los datos necesarios del producto.	2. El sistema mostrara el producto seleccionado al cual se modificara y guarda los nuevos cambios.
Flujo alterno	
1. Si el administrador no ingresa correctamente la información requerida el sistema volverá a pedir la información.	



Caso de uso		Editar los datos de proveedor.
Actores		Administrador.
Propósito		Modificar los datos del proveedor.
Precondición		Estar registrado en base de datos.
Post condición		Actualizar los datos del proveedor.
Flujo básico de eventos		
Actor		Respuesta del sistema
1. El administrador modifica los datos necesarios del proveedor.		2. El sistema mostrara el proveedor seleccionado ya existente al cual se modificara y guarda los nuevos cambios.
Flujo alternativo		
1. Si el administrador no ingresa correctamente la información requerida el sistema volverá a pedir la información.		

Caso de uso		Buscar cliente.
Actores		Administrador.
Propósito		Buscar la información de un determinado cliente.
Precondición		Ingresar los datos del cliente.
Post condición		Detalle del cliente.
Flujo básico de eventos		
Actor		Respuesta del sistema
1. El administrador busca los datos necesarios del cliente existente.		2. El sistema mostrara los datos del cliente registrado.
Flujo alternativo		
1. Si el administrador no ingresa correctamente los datos en el formulario el sistema mandara una advertencia e indicara el error y no se podrá realizar la búsqueda.		



Caso de uso	Buscar la compra.
Actores	Administrador.
Propósito	Visualizar la compra realizada.
Precondición	Ingresar los datos de la compra.
Post condición	Detalle de la compra.
Flujo básico de eventos	
Actor	Respuesta del sistema
1. El administrador busca los datos necesarios de la compra realizada.	2. El sistema mostrara los datos de la compra ya efectuada.
Flujo alterno	
1. Si el administrador no ingresa correctamente los datos en el formulario el sistema mandara una advertencia e indicara el error y no se podrá realizar la búsqueda.	

Caso de uso	Buscar los productos.
Actores	Administrador/usuario.
Propósito	Proporciona el servicio de búsqueda para verificar la existencia en el inventario.
Precondición	Ingresar el código / nombre del medicamento.
Post condición	Detalle del producto.
Flujo básico de eventos	
Actor	Respuesta del sistema
1. El administrador busca los datos necesarios del producto existente.	2. El sistema mostrara los datos del producto registrado.
Flujo alterno	
1. Si el administrador/usuario no ingresa correctamente los datos en el formulario el sistema mandara una advertencia e indicara el error y no se podrá realizar la búsqueda.	



Caso de uso	Buscar la venta.
Actores	Administrador/usuario.
Propósito	Visualizar la realización de la venta elaborada.
Precondición	Ingresar los datos de la venta.
Post condición	Detalle de la venta.
Flujo básico de eventos	
Actor	Respuesta del sistema
1. El administrador/usuario buscara los datos necesarios de la venta realizada.	2. El sistema visualizara la venta.
Flujo alternativo	
1. Si el administrador no ingresa correctamente los datos en el formulario el sistema mandara una advertencia e indicara el error y no se podrá realizar la búsqueda.	

Caso de uso	Buscar el proveedor.
Actores	Administrador.
Propósito	Proporciona el servicio de búsqueda para verificar la existencia del proveedor.
Precondición	Ingresar el dato del proveedor.
Post condición	Detalle del proveedor
Flujo básico de eventos	
Actor	Respuesta del sistema
1. El administrador busca los datos necesarios del proveedor existente.	2. El sistema mostrara los datos del proveedor registrado.
Flujo alternativo	
1. Si el administrador/usuario no ingresa correctamente los datos en el formulario el sistema mandara una advertencia e indicara el error y no se podrá realizar la búsqueda.	



Caso de uso	Buscar el usuario.
Actores	Administrador.
Propósito	Buscar la información de un determinado usuario.
Precondición	Ingresar los datos del usuario.
Post condición	Detalle del usuario.
Flujo básico de eventos	
Actor	Respuesta del sistema
1. El administrador busca los datos necesarios del cliente existente.	2. El sistema mostrara los datos del usuario registrada.
Flujo alterno	
1. Si el administrador/usuario no ingresa correctamente los datos en el formulario el sistema mandara una advertencia e indicara el error y no se podrá realizar la búsqueda.	

Caso de uso	Reporte de drogas.	
Actores	Administrador/usuario.	
Propósito	Revisar / actualizar que drogas sean	
Precondición	Seleccionar el día correspondiente.	
Post condición	Mandar a mostrar el reporte.	
Flujo básico de eventos		
Actor	Respuesta del sistema	
1. El administrador /usuario selecciona la fecha del mes que desea el reporte.	3. muestra el informe productos controlados del mes seleccionado.	
2. El Administrador tiene la tarea de exportar, imprimir reporte, actualizar, copiar.		
Flujo alternativo		
1.	Si el administrador /usuario no selecciona el mes no le mostrara ningún resultado.	



Caso de uso	Reporte ventas al día.
Actores	Administrador/usuario.
Propósito	Tener control de las ventas al día.
Precondición	Registrar las ventas.
Post condición	Las ventas tienen que estar registrada.
Flujo básico de eventos	
Actor	Respuesta del sistema
1. El administrador/usuario selecciona la fecha. 2. El Administrador tiene la tarea de exportar, imprimir reporte, actualizar, copiar.	3. Muestra el reporte de las ventas del día.
Flujo alterno	
1. Si el administrador /Usuario no selecciona la fecha de la venta el sistema no le mostrara nada.	

Caso de uso	Reporte de inventario.
Actores	Administrador.
Propósito	Revisar/ Actualizar el registro de todos los productos en existencia.
Precondición	Tener registro de todos los productos en existencia.
Post condición	Estar registrado en la base de datos.
Flujo básico de eventos	
Actor	Respuesta del sistema
1. El administrador selecciona Inventario. 2. El Administrador tiene la tarea de exportar, imprimir reporte, actualizar, copiar.	3. El sistema muestra el reporte de inventario.
Flujo alterno	
1. Si el administrador no selecciona el inventario no le mostrara ningún resultado.	



Caso de uso	Reporte ventas al mes.
Actores	Administrador.
Propósito	Tener control de todas las ventas realizadas en el mes.
Precondición	Las ventas estén registradas.
Post condición	
Flujo básico de eventos	
Actor	Respuesta del sistema
1. El administrador selecciona la fecha inicio y la fecha final que desee el informe.	3. Muestra el reporte de las ventas del mes.
2. El Administrador tiene la tarea de exportar, imprimir reporte, actualizar, copiar.	
Flujo alterno	
1. Si el administrador no selecciona el intervalo de fecha de la venta el sistema no le mostrara nada.	

Caso de uso	Reporte productos vencidos.
Actores	Administrador.
Propósito	Descartar los productos pronto a vencer de la farmacia.
Precondición	Productos estén registrados.
Post condición	Actualizar las base de datos.
Flujo básico de eventos	
Actor	Respuesta del sistema
1. El administrador selecciona productos vencidos.	3. Muestra el reporte de productos vencidos.
2. El Administrador tiene la tarea de exportar, imprimir reporte, actualizar, copiar.	
Flujo alternativo	
1. Si el administrador no selecciona productos vencidos el sistema no le mostrara nada.	



Caso de uso	Reporte cuentas por cobrar.
Actores	Administrador.
Propósito	Poseer la información de las cuentas por cobrar.
Precondición	Las cuentas estén registradas.
Post condición	Adquirir todas las cuentas que tienen que ser cobradas.
Flujo básico de eventos	
Actor	Respuesta del sistema
1. El administrador selecciona cuentas por cobrar.	3. Muestra el reporte de las cuentas por cobrar.
2. El Administrador tiene la tarea de exportar, imprimir reporte, actualizar, copiar.	
Flujo alternativo	
1. Si el administrador no selecciona cuenta por cobrar el sistema no le mostrara nada.	

Caso de uso	Reporte cuentas por pagar.
Actores	Administrador.
Propósito	Obtener la información de las cuentas por pagar.
Precondición	Las cuentas estén registradas.
Post condición	Adquirir todas las cuentas que tienen que ser pagadas.
Flujo básico de eventos	
Actor	Respuesta del sistema
1. El administrador selecciona cuenta por pagar.	3. Muestra el reporte de las cuentas por pagar.
2. El Administrador tiene la tarea de exportar, imprimir reporte, actualizar, copiar.	
Flujo alternativo	
1. Si el administrador no selecciona cuenta por pagar el sistema no le mostrara nada.	



Caso de uso	Visualizar Factura
Actores	Administrador/Usuario.
Propósito	Obtener datos de la factura ya existente.
Precondición	Restablecer una factura ya existente.
Post condición	Obtener los resultados de la factura a visualizar.
Flujo básico de eventos	
Actor	Respuesta del sistema
1. El administrador/Usuario seleccionaran la fecha inicio, fecha fin y el número de la venta realizada.	2. El sistema visualizara la factura seleccionada.
Flujo alterno	
1. Si el administrador/Usuario no ingresa correctamente los datos en el formulario el sistema mandara una advertencia y no se podrá realizar la visualización de la factura.	

Caso de uso	Respaldar la base de datos.
Actores	Administrador.
Propósito	Tener respaldo de la base de datos.
Precondición	Asignar un lugar de almacenamiento.
Post condición	Obtener respaldo de toda la base de datos.
Flujo básico de eventos	
Actor	Respuesta del sistema
1. El administrador introduce el nombre del archivo y selecciona la ruta de almacenaje.	2. El sistema manda un mensaje de confirmación (Archivo Respaldado creado).
Flujo alterno	
1. Si el administrador no ingresa correctamente los datos en el formulario el sistema mandara una advertencia y no se podrá realizar el respaldo.	

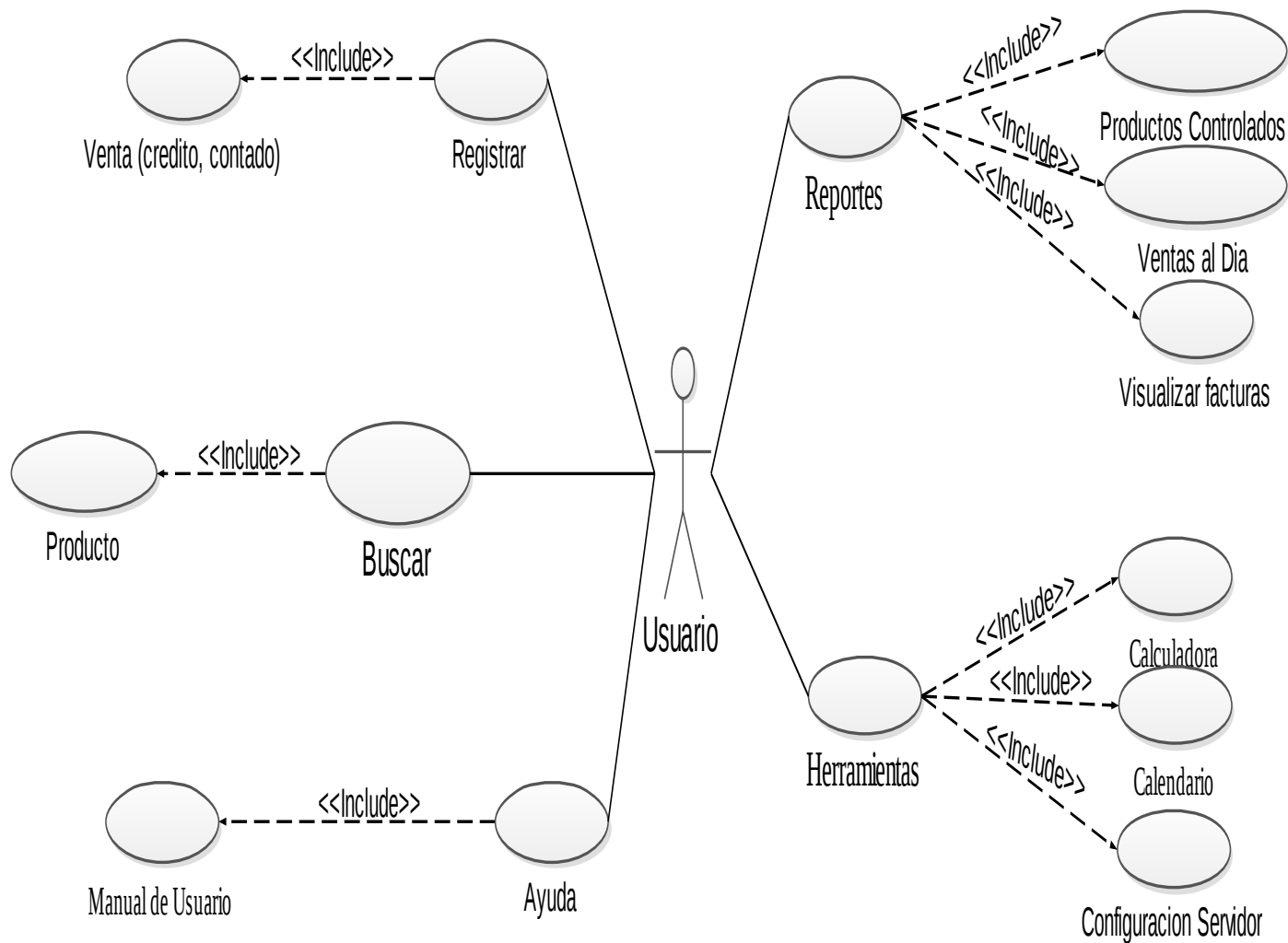


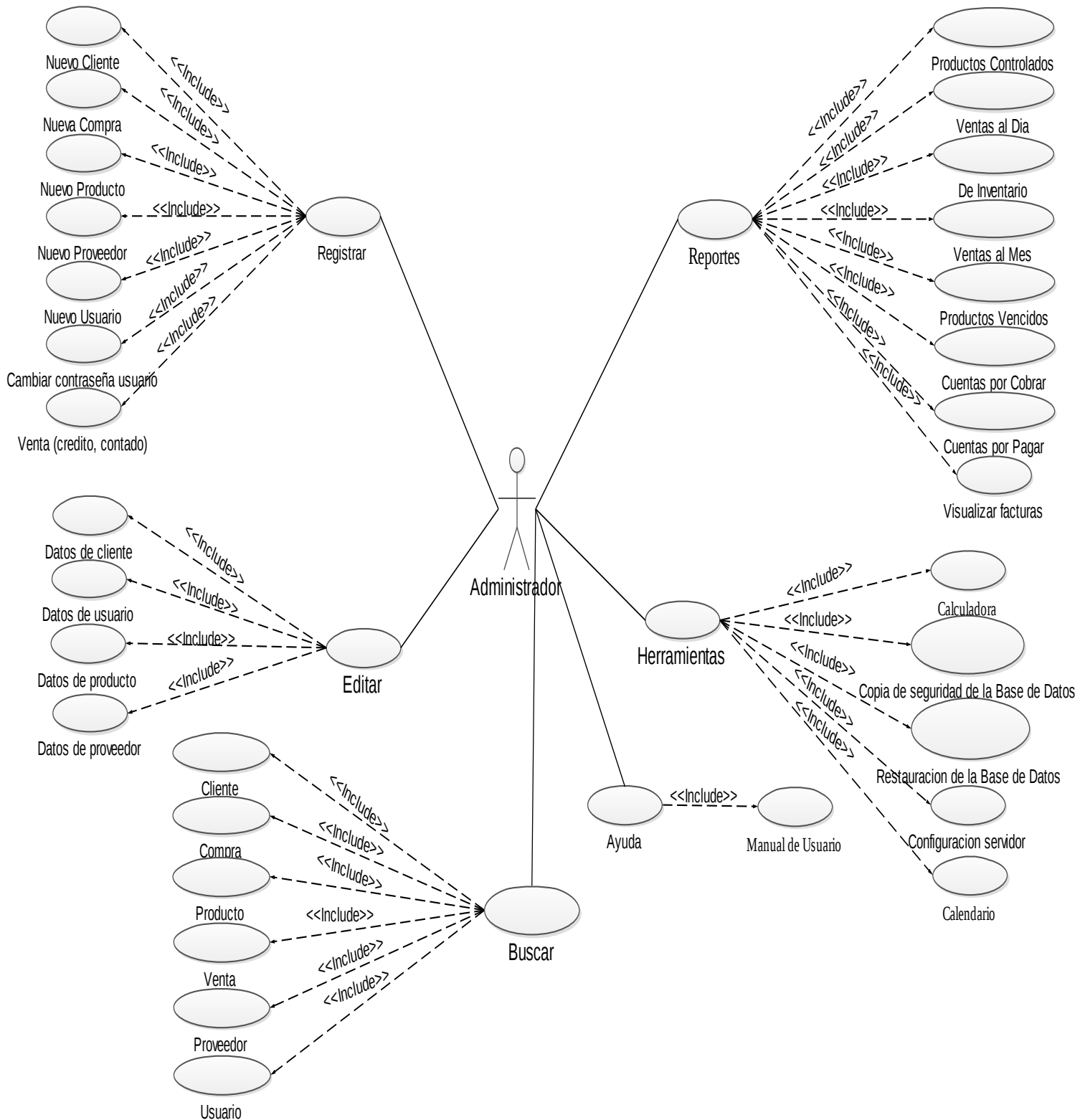
Caso de uso	Restaurar la base de datos.	
Actores	Administrador.	
Propósito	Buscar ubicación para restaurar la base de datos.	
Precondición	Tener la base de datos respaldada	
Post condición	Obtener la información de la base de datos.	
Flujo básico de eventos		
Actor	Respuesta del sistema	
1. El administrador busca la ubicación de donde desea hacer la restauración de la BBDD.	2. El sistema manda un mensaje confirmación (BBDD Restaurada).	
Flujo alternativo		
1. Si el administrador no ingresa correctamente los datos en el formulario el sistema mandara una advertencia y no se podrá realizar el respaldo.		

Caso de uso	Configuración servidor
Actores	Administrador/Usuario
Propósito	Configurar el servidor de la red.
Precondición	Acceder a la aplicación.
Post condición	Utilizar la aplicación.
Flujo básico de eventos	
Actor	Respuesta del sistema
1. El administrador/Usuario seleccionara configuración servidor e ingresa la dirección IP o el nombre del servidor, y la contraseña de acceso.	2. El sistema se conectara al servidor. 3. Se tendrá acceso a la aplicación.
Flujo alternativo	
1. Si el administrador/ usuario al introducir de manera incorrecta los datos a la aplicación no se va conectar con el servidor.	
2. Si el administrador/ usuario no puede ingresar a la aplicación se va a poder presionar f12 y aparecerá el formulario de configuración.	



DIAGRAMA DE CASOS DE USO.

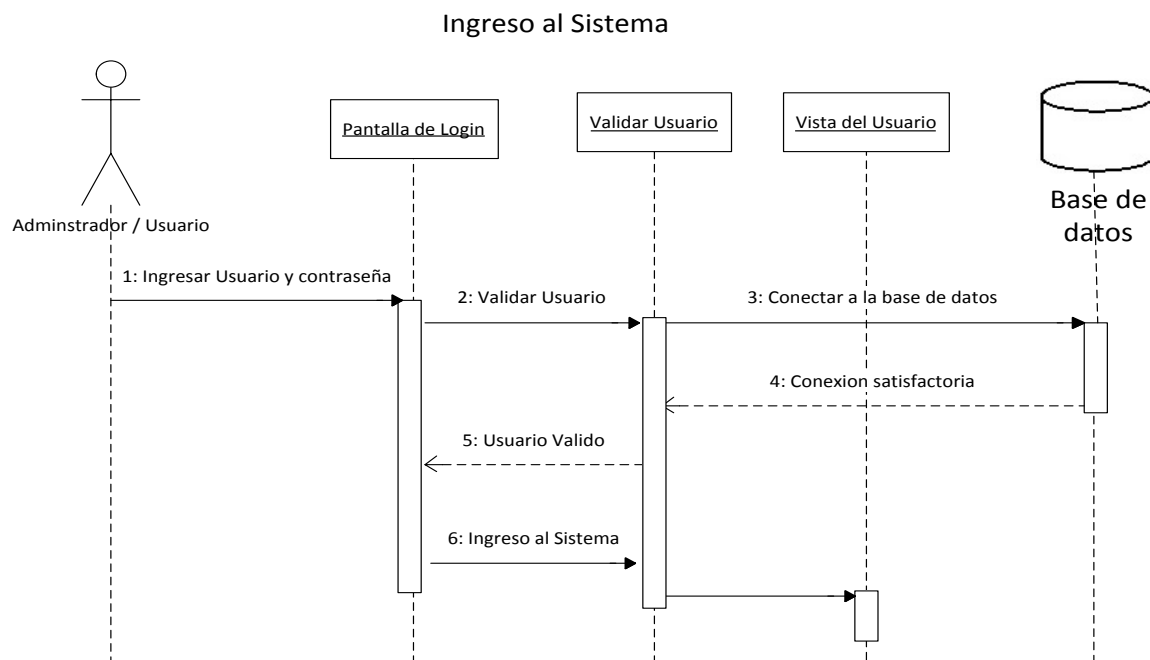




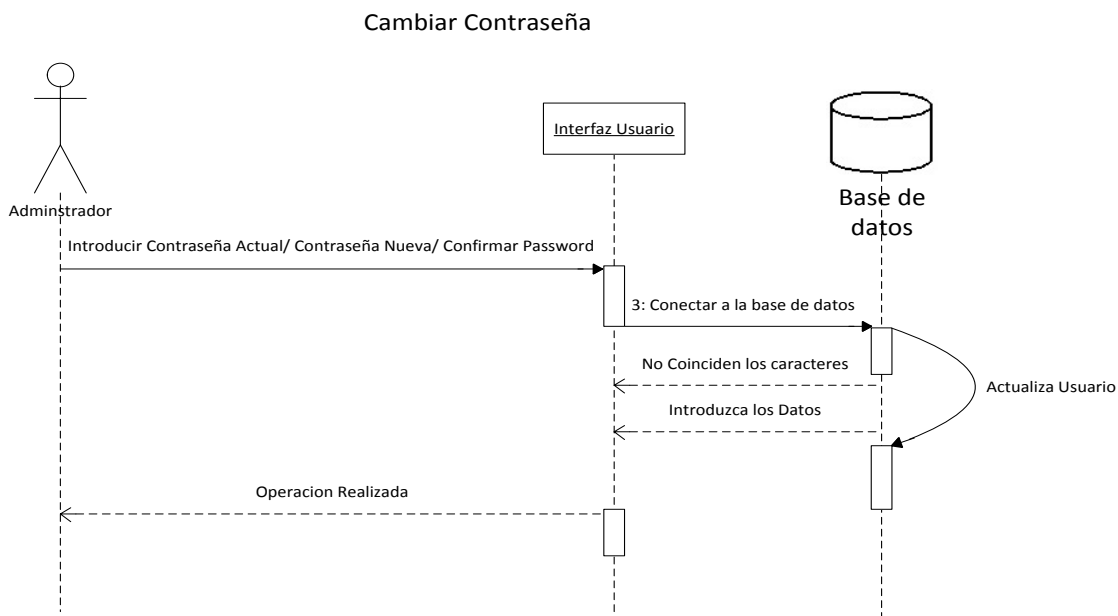


DIAGRAMAS DE SECUENCIA.

1.



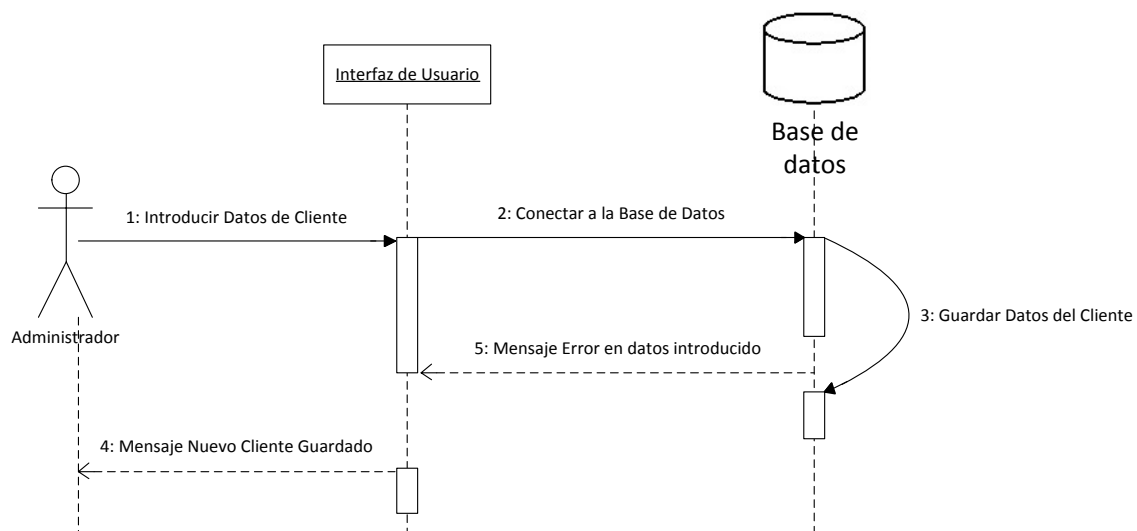
2.





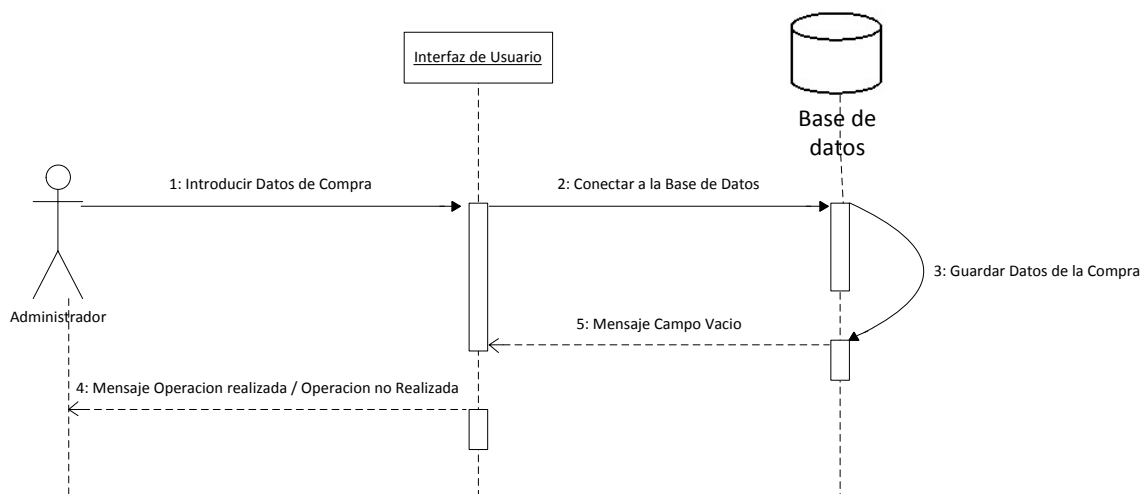
3.

Registrar Nuevo Cliente



4.

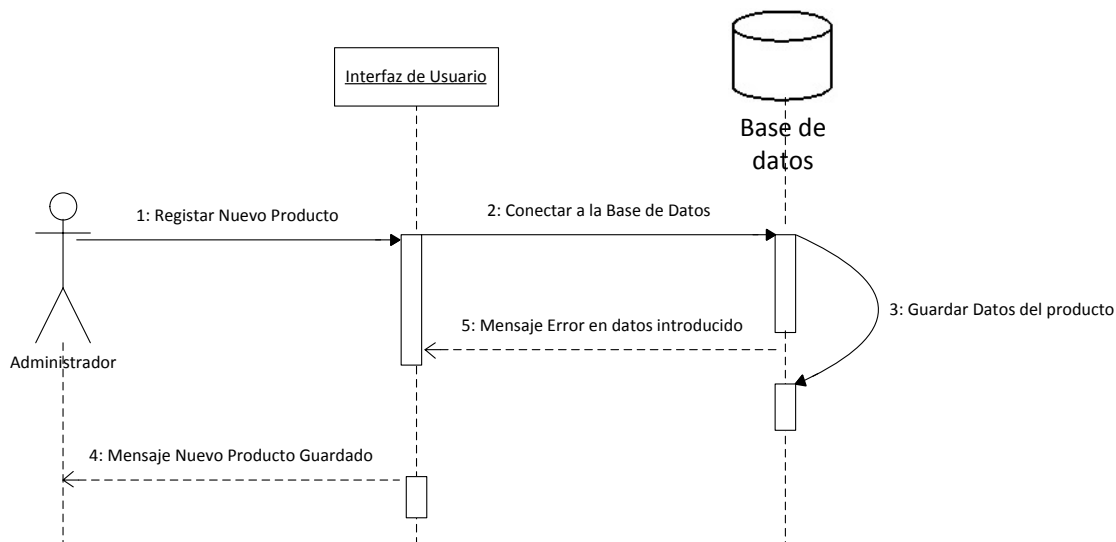
Registrar Nueva Compra





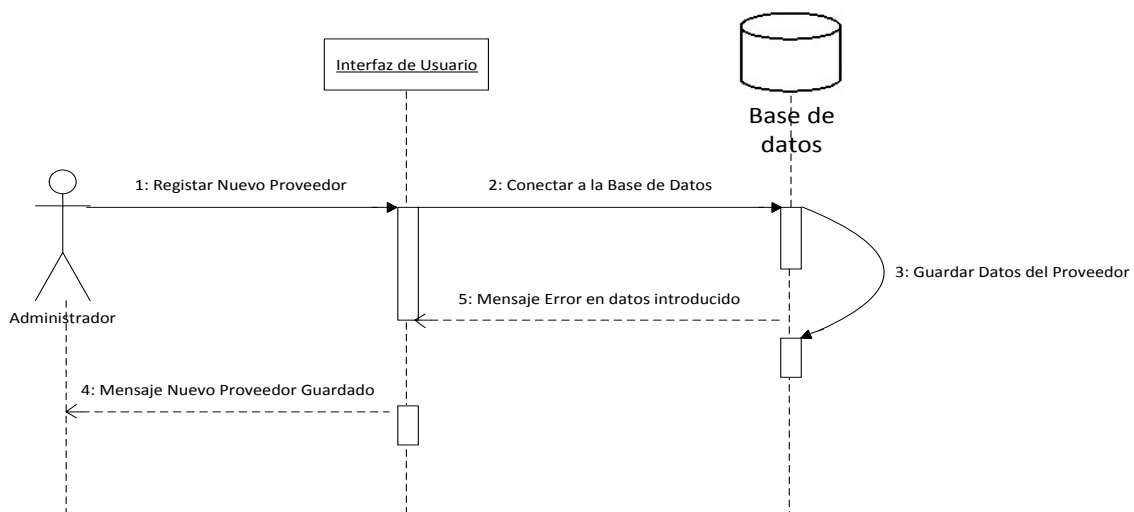
5.

Registrar Nuevo Producto



6.

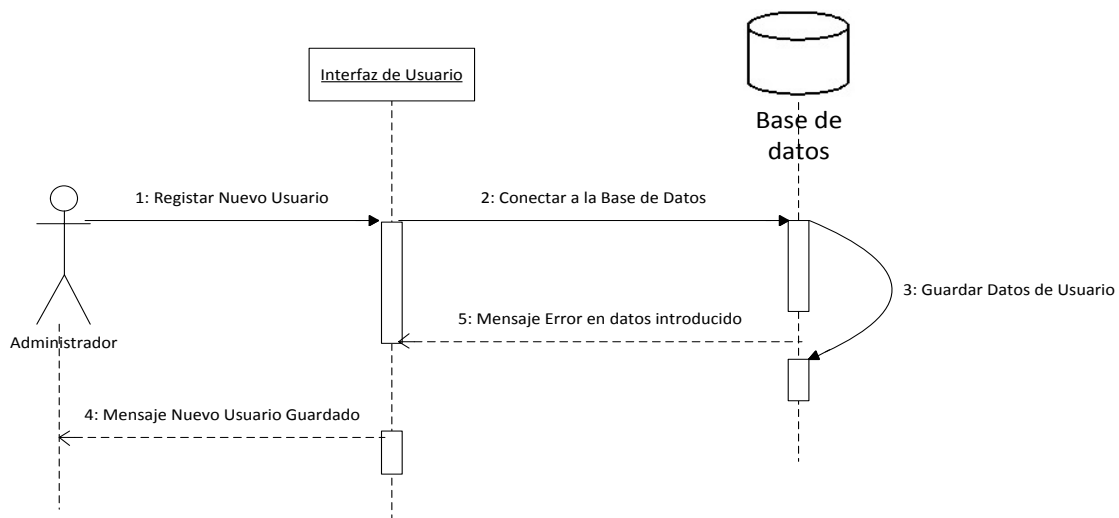
Registrar Nuevo Proveedor





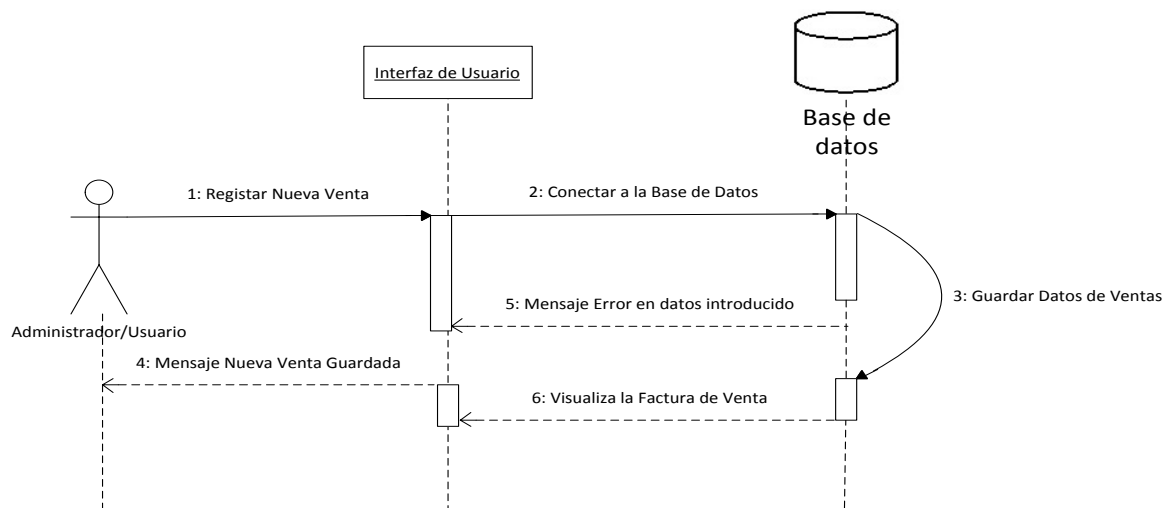
7.

Registrar Nuevo Usuario



8.

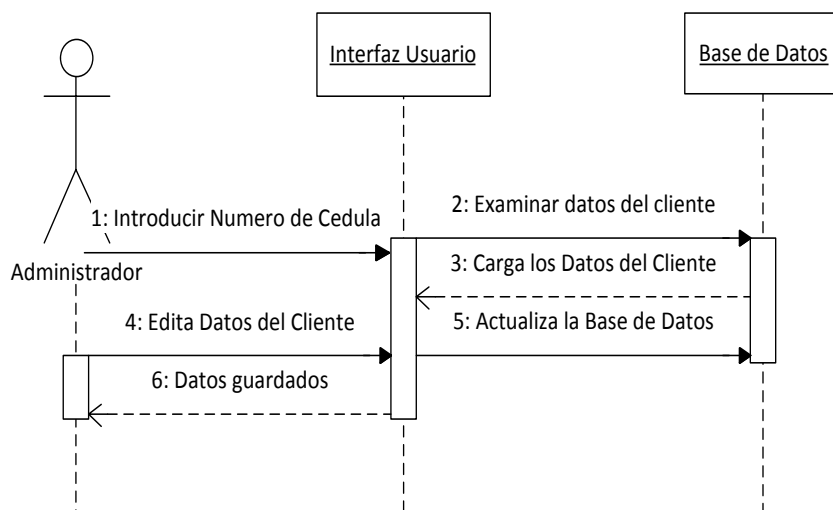
Registrar Nueva Venta





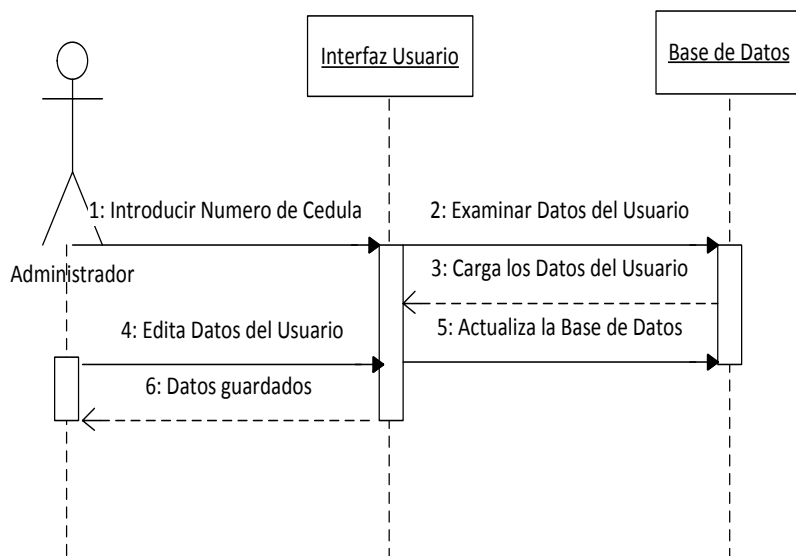
9.

Editar Cliente



10.

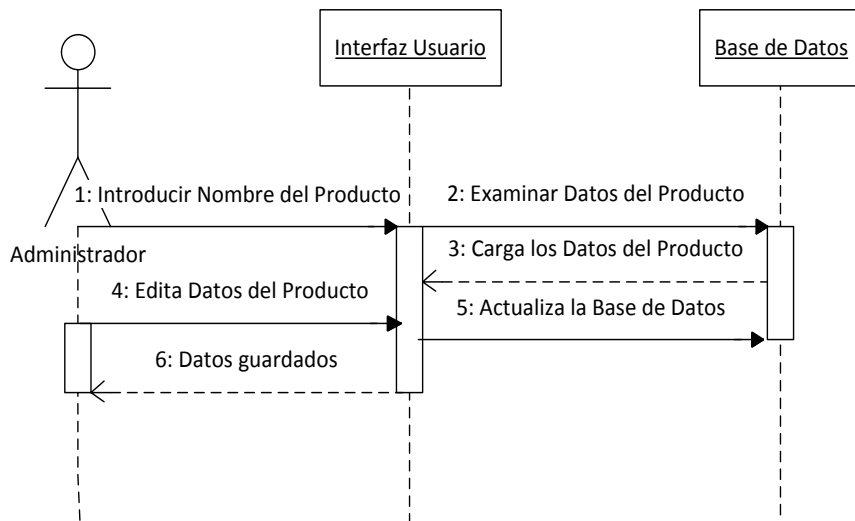
Editar Usuario





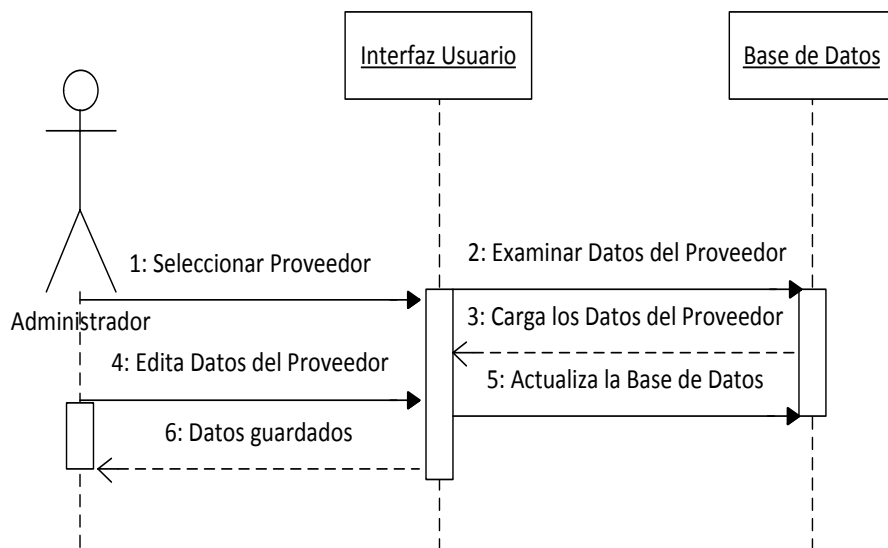
11.

Editar Producto



12.

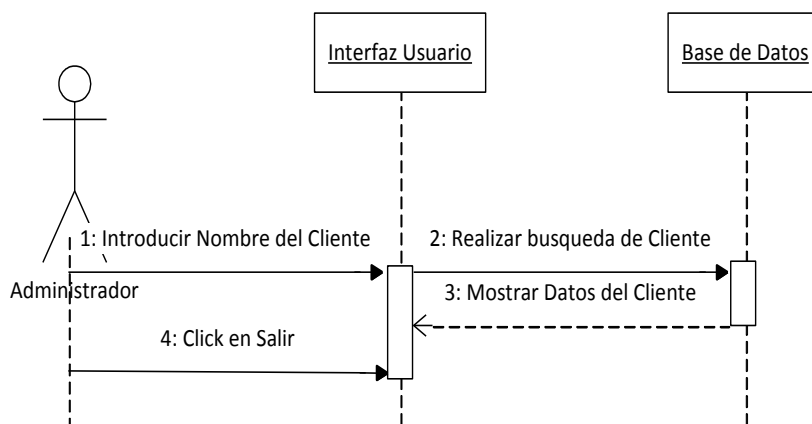
Editar Proveedor





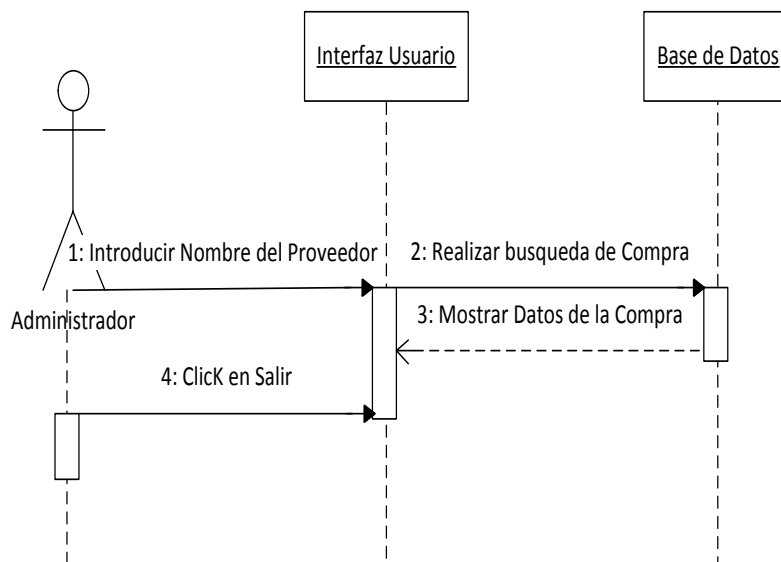
13.

Buscar Cliente



14.

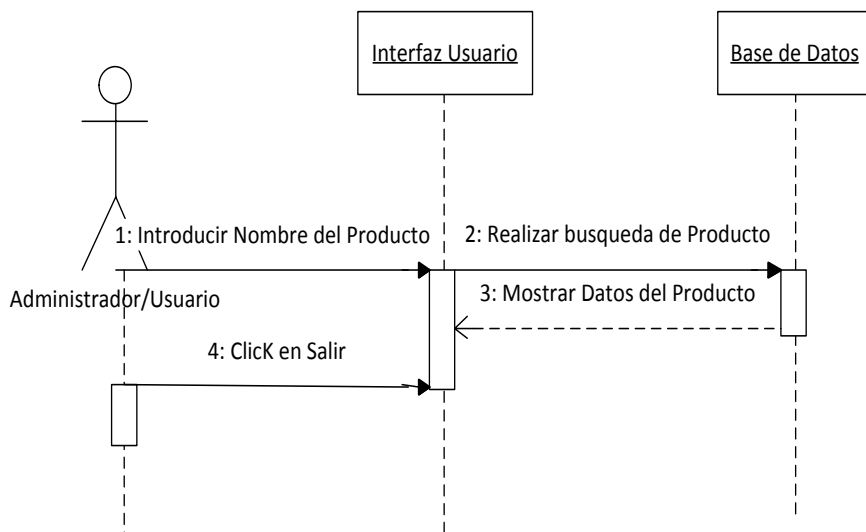
Buscar Compra





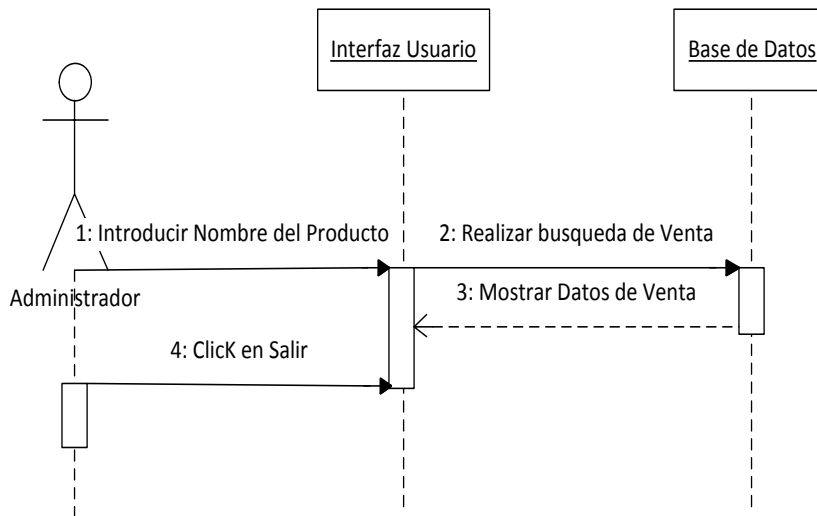
15.

Buscar Producto



16.

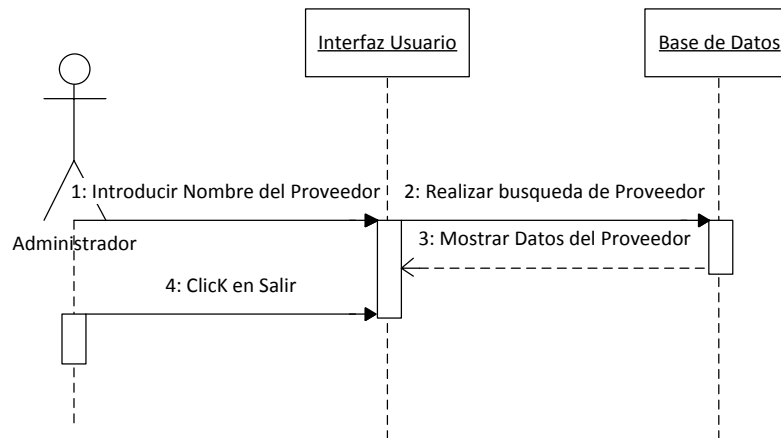
Buscar Ventas





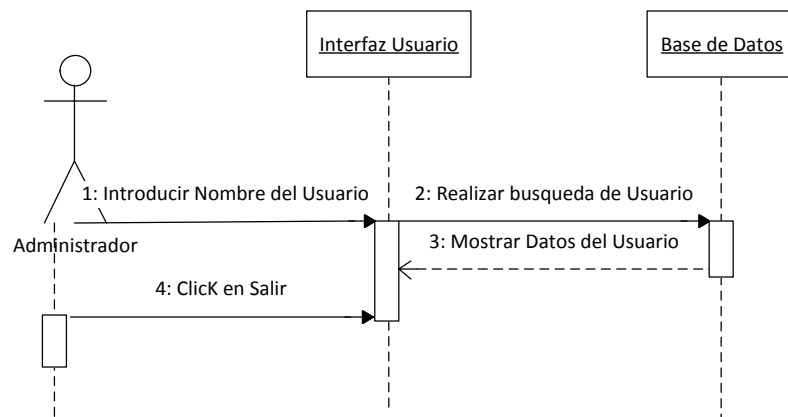
17.

Buscar Proveedor



18.

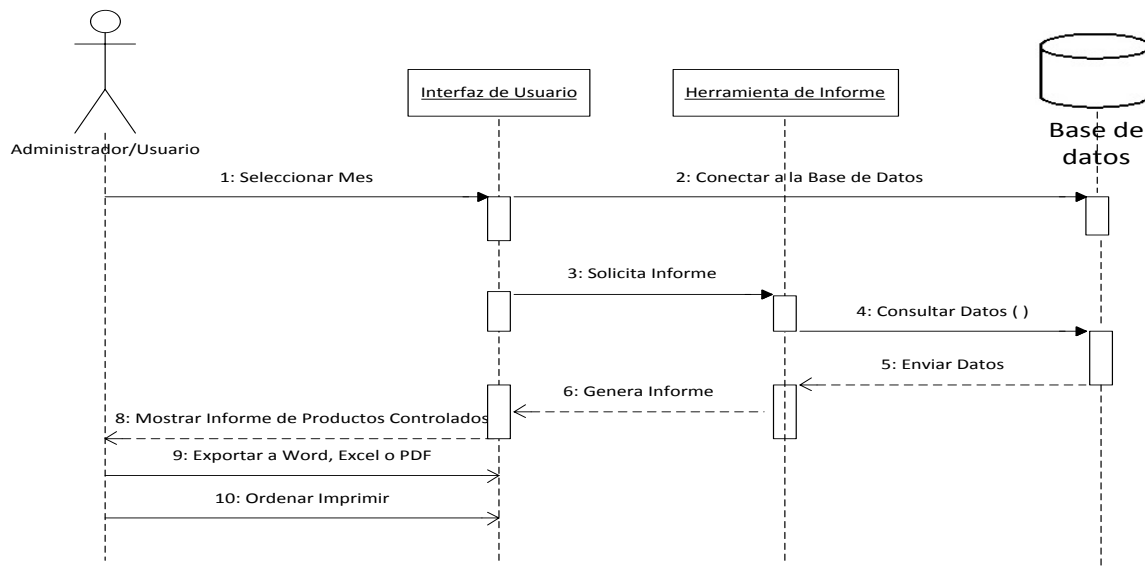
Buscar Usuario





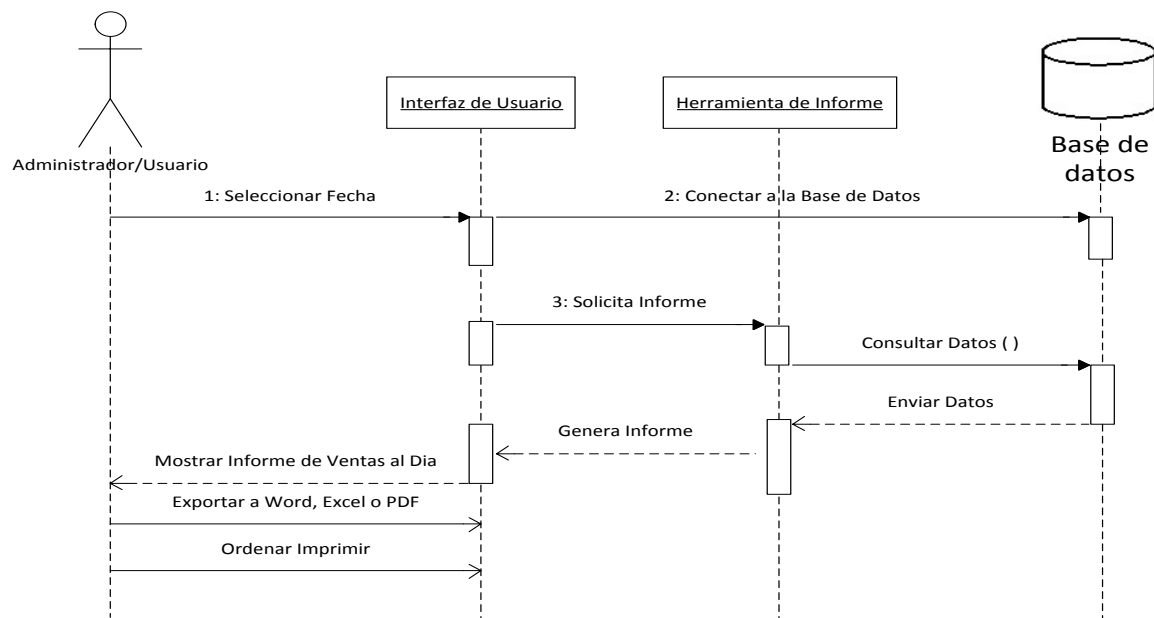
19.

Reporte de Productos Controlados



20.

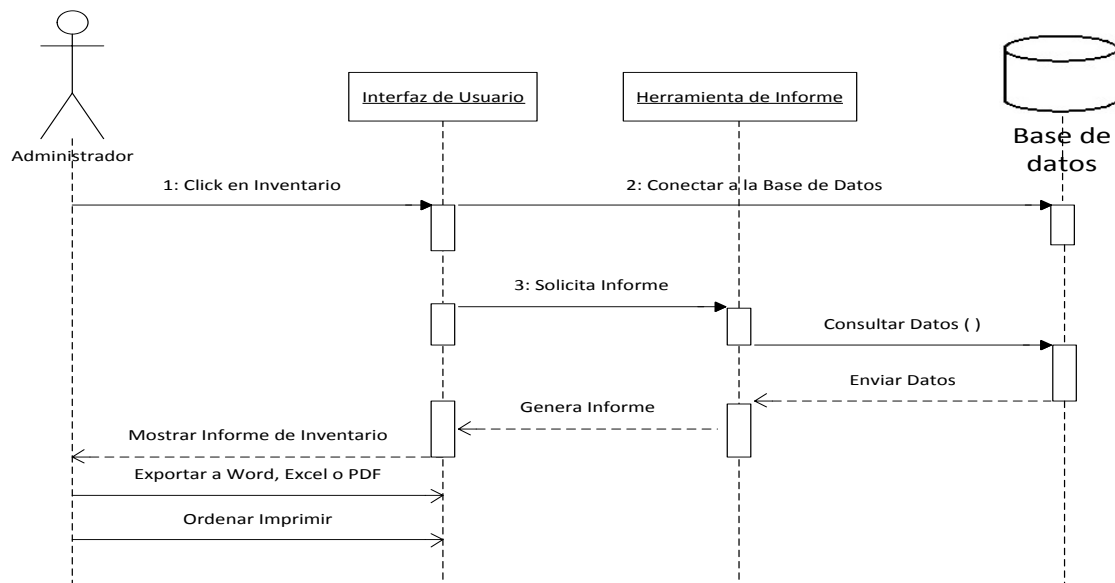
Reporte de Ventas al Día





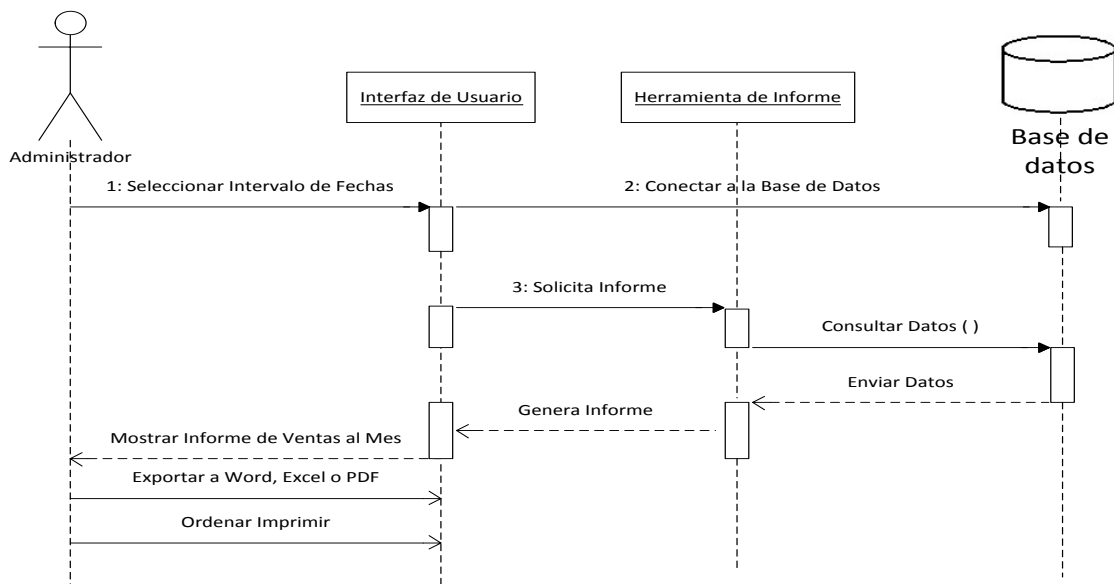
21.

Reporte de Inventario



22.

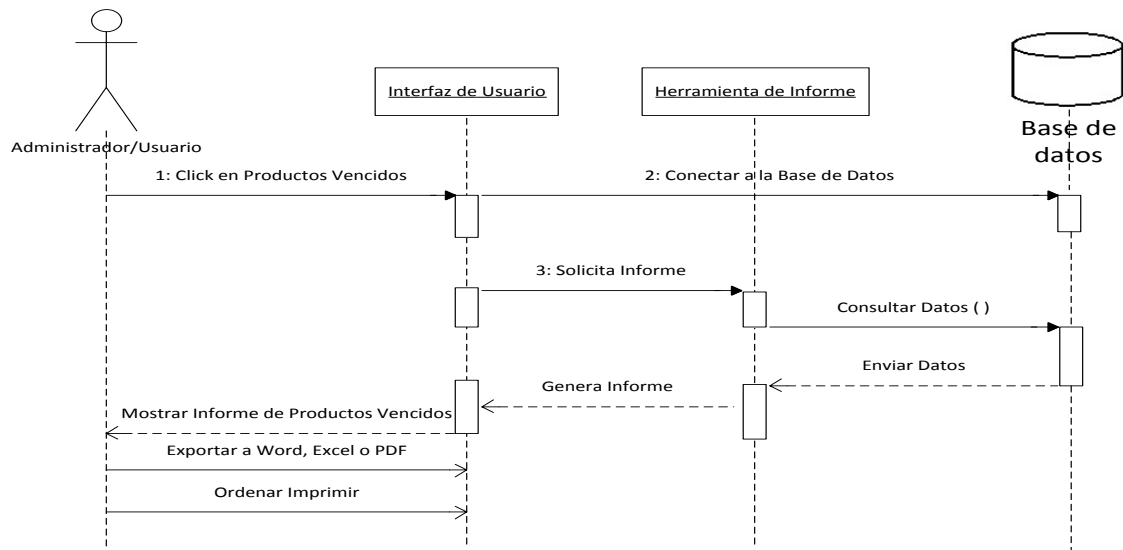
Reporte de Ventas al Mes





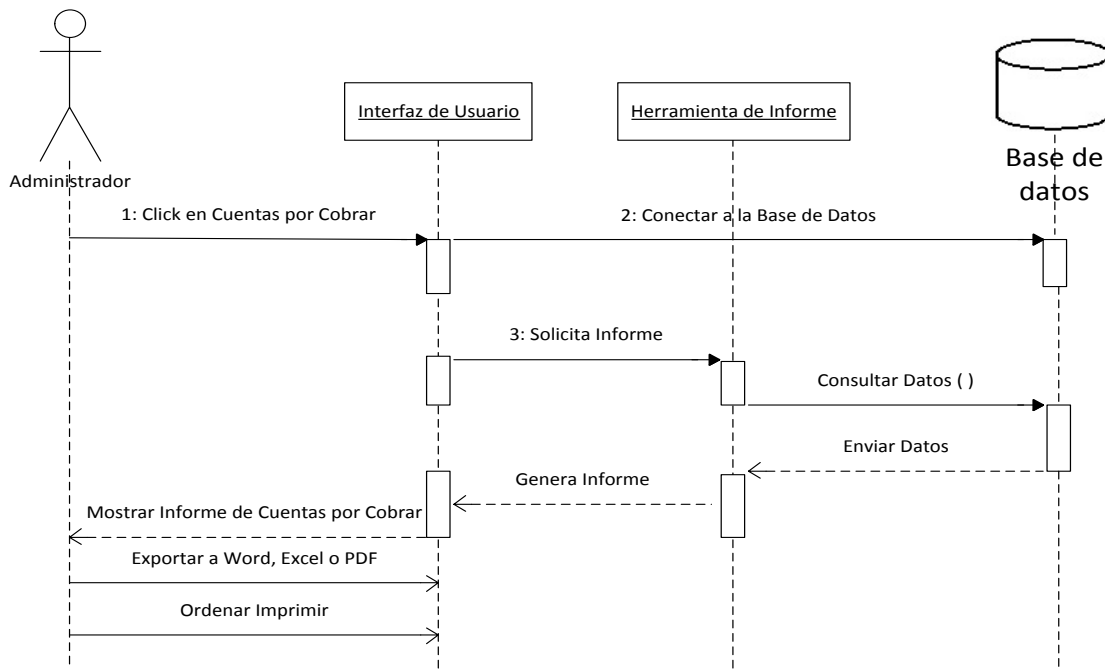
23.

Reporte de Productos Vencidos



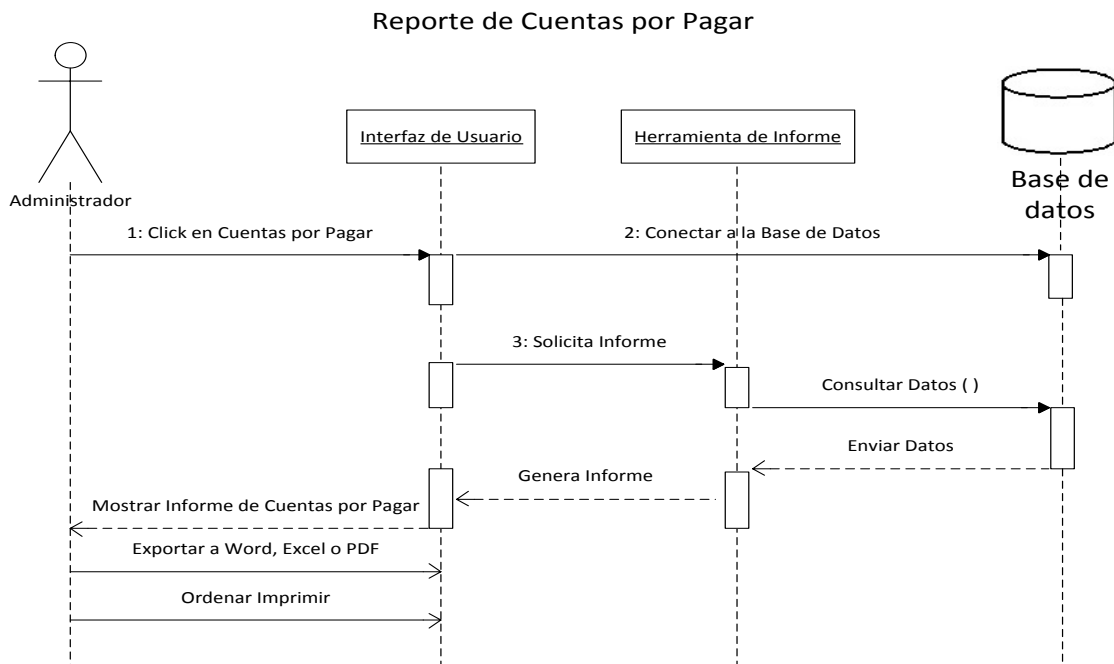
24.

Reporte de Cuentas por Cobrar

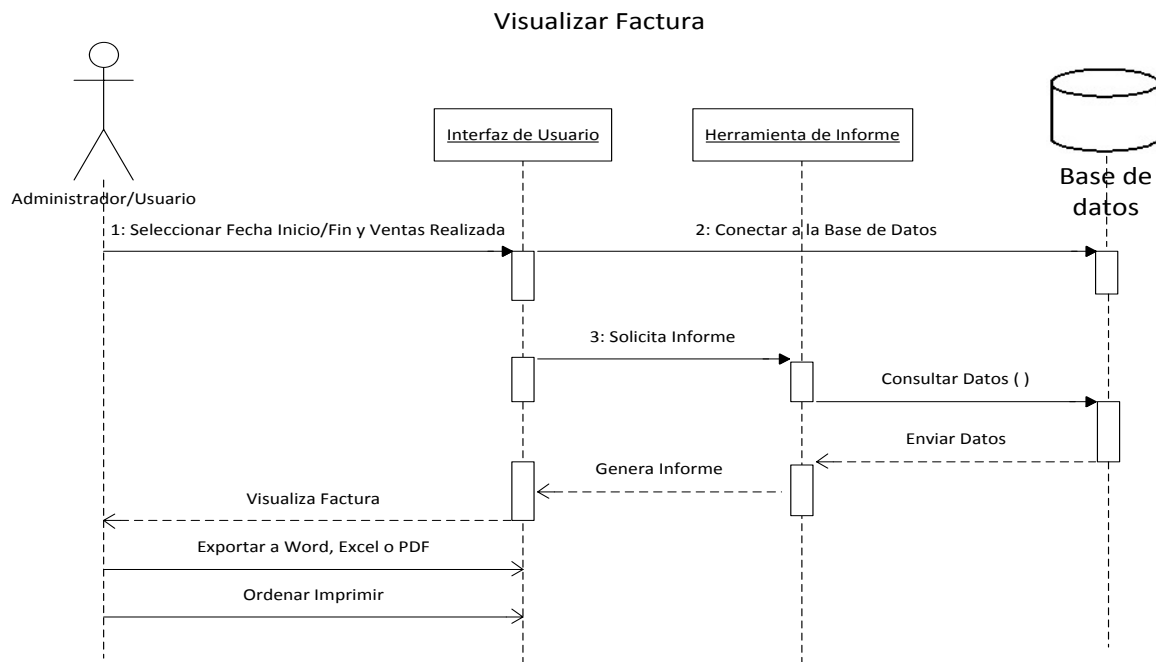




25.



26.





METODOLOGIA.

El método de desarrollo de trabajo que se utilizó para la elaboración de la aplicación fue el modelo en cascada, ya que resulta conveniente en la elaboración de las actividades de la aplicación.

El modelo en cascada descompone el proceso de desarrollo en diferentes fases, constituyendo la salida de cada una de ellas la entrada requerida por la siguiente. En este modelo se supone que todos los requisitos son conocidos y comprendidos perfectamente al iniciar el desarrollo del software.

El método de la cascada es considerado como el enfoque clásico para el ciclo de vida del desarrollo de sistemas, se puede decir que es un método puro que implica un desarrollo rígido y lineal. Un ejemplo de la metodología en cascada es:

1. Análisis de requisitos
2. Diseño del sistema
3. Diseño del programa
4. Codificación
5. Pruebas
6. Implantación
7. Mantenimiento

- **Análisis de requisitos**

En esta fase se analizan las necesidades de los usuarios finales del software para determinar qué objetivos debe cubrir. De esta fase surge una memoria llamada



SRD (documento de especificación de requisitos), que contiene la especificación completa de lo que debe hacer el sistema sin entrar en detalles internos.

Es importante señalar que en esta etapa se debe consensuar todo lo que se requiere del sistema y será aquello lo que seguirá en las siguientes etapas, no pudiéndose requerir nuevos resultados a mitad del proceso de elaboración del software.

- **Diseño del Sistema**

Descompone y organiza el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo. Como resultado surge el SDD (Documento de Diseño del Software), que contiene la descripción de la estructura relacional global del sistema y la especificación de lo que debe hacer cada una de sus partes, así como la manera en que se combinan unas con otras.

Es conveniente distinguir entre diseño de alto nivel o arquitectónico y diseño detallado. El primero de ellos tiene como objetivo definir la estructura de la solución (una vez que la fase de análisis ha descrito el problema) identificando grandes módulos (conjuntos de funciones que van a estar asociadas) y sus relaciones. Con ello se define la arquitectura de la solución elegida. El segundo define los algoritmos empleados y la organización del código para comenzar la implementación.

- **Diseño del Programa**

Es la fase en donde se realizan los algoritmos necesarios para el cumplimiento de los requerimientos del usuario así como también los análisis necesarios para saber que herramientas usar en la etapa de Codificación.

- **Codificación**



Es la fase en donde se implementa el código fuente, haciendo uso de prototipos así como de pruebas y ensayos para corregir errores.

Dependiendo del lenguaje de programación y su versión se crean las bibliotecas y componentes reutilizables dentro del mismo proyecto para hacer que la programación sea un proceso mucho más rápido.

- **Pruebas**

Los elementos, ya programados, se ensamblan para componer el sistema y se comprueba que funciona correctamente y que cumple con los requisitos, antes de ser entregado al usuario final.

- **Verificación**

Es la fase en donde el usuario final ejecuta el sistema, para ello el o los programadores ya realizaron exhaustivas pruebas para comprobar que el sistema no falle.

- **Mantenimiento**

Una de las etapas más críticas, ya que se destina un 75% de los recursos, es el mantenimiento del Software ya que al utilizarlo como usuario final puede ser que no cumpla con todas nuestras expectativas.

Se inicia con la especificación de requerimientos del cliente, continua con la planificación, el modelado, la construcción y el despliegue para finalizar en el enfoque del software. El modelo está dirigido por documentos y no proporciona resultados tangibles de software hasta el final del ciclo de vida de algunas herramientas.

El diseño en cascada es una secuencia definida de los acontecimientos y los resultados finales para proporcionar una estructura para cualquier proyecto que siga el contenido



específico y detallado. Puede ser apropiado para proyectos de software que son estables especialmente cuando sus requisitos no cambian.

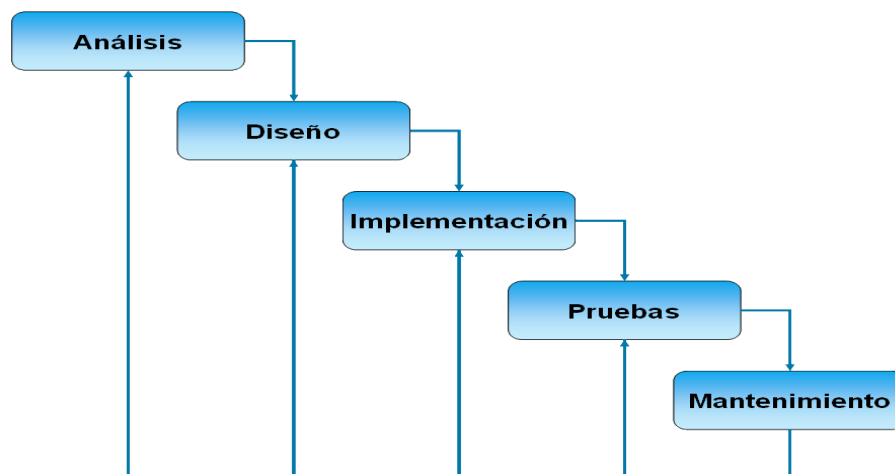
Este modelo requiere también que los implementadores sigan el bien hecho, el diseño completo de precisión, asegurando así la integración de los ingresos del sistema sin problemas.

Es caracterizado por ordenar de manera rigurosa las etapas del ciclo de vida de software, dado que el comienzo de cada etapa debe esperar a la finalización de la inmediata anterior. Cuando la revisión determina que el proyecto no está listo para pasar a la siguiente etapa, permanece en la etapa actual hasta que esté preparado. Y debido a que el proceso está planeado es más fácil determinar costos y los plazos.

Para decidir implantar la metodología en cascada se necesita hacer un análisis de la situación, por ejemplo: si el cliente quiere intervenir en el proceso una vez iniciado, este método no sería el indicado, sino un método iterativo.

Para proceder al diseño primero hay que determinar la especificación de requisitos los cuales no pueden ser modificados tras el cierre de sesión. Una modificación o cambio mediante la ejecución de alguna de las fases, implicaría reiniciar desde el principio todo el ciclo completo, esto implica mayor inversión de tiempo y desarrollo. Asegurarse en el inicio de que las necesidades y el diseño son los correctos nos ahorrara tiempo y esfuerzo

El modelo en cascada proporciona un enfoque estructurado, progresa linealmente a través de sus fases por lo que resulta fácil de entender.



Modelo en Cascada.



RECURSOS.

1. Recursos Hardware

Computadora 1:

Procesador Intel i3 2.3 GHz

Memoria 3GB de RAM.

HDD 250 GB.

Computadora 2:

Procesador Intel Pentium(R) Dual-Core 2.3 GHz

Memoria 3GB de RAM.

HDD 320 GB.

2. Recursos Software

MySQL Server 5.1.62

Visual Studio Ultimate 2010 (C#).

Help N- Docs

Dev. Component 10.4

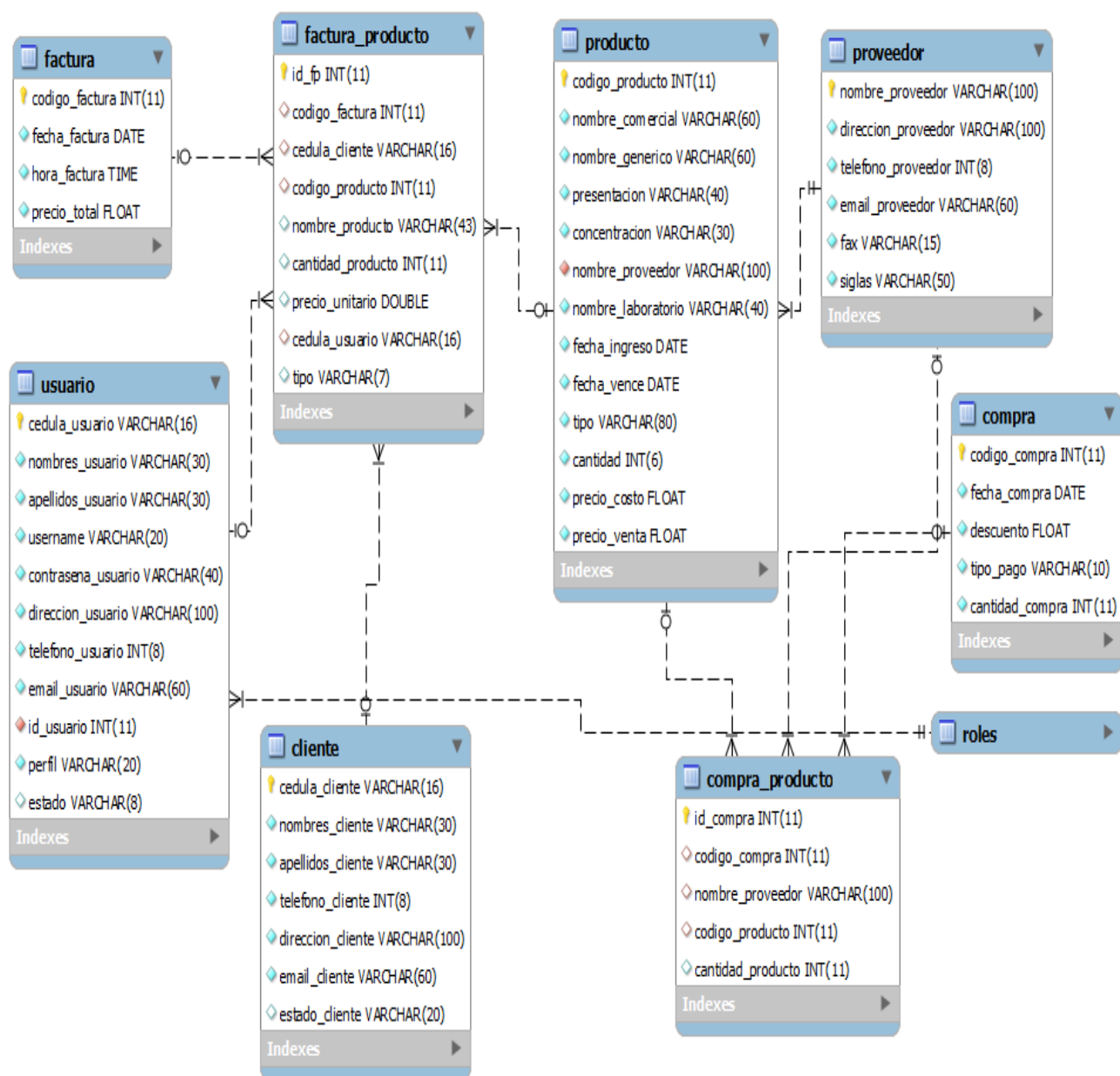
Window 7 Ultimate Service Pack 1.

Crystal Reports.

MySQL Connector / Net



DIAGRAMA ENTIDAD-RELACION.





DICCIONARIO DE DATOS.

Tabla 1. Clientes.

	Nombre	Tipo	Primaria	No Nulo
1	 cedula_cliente	varchar(16)	☒	☒
2	 nombres_cliente	varchar(30)	☐	☒
3	 apellidos_cliente	varchar(30)	☐	☒
4	 telefono_cliente	int(8)	☐	☒
5	 direccion_cliente	varchar(100)	☐	☒
6	 email_cliente	varchar(60)	☐	☒
7	 estado_cliente	varchar(20)	☐	☐

Nombre del campo	Tipo de dato	Longitud	Valores permitidos
Cedula_cliente	Varchar	16	Not Null
Nombres_cliente	Varchar	30	Not Null
Apellidos_cliente	Varchar	30	Not Null
Teléfono_cliente	Int	8	Not Null
Dirección_cliente	Varchar	100	Not Null
Email_cliente	Varchar	60	Not Null
Estado_cliente	Varchar	20	



Tabla 2. Compra.

Nombre	Tipo	Primaria	No Nulo
1 > código_compra	int	☒	☒
2 fecha_compra	date	☐	☒
3 nombre_proveedor	varchar(100)	☐	☐
4 tipo_pago	varchar(10)	☐	☒
5 estado	varchar(9)	☐	☐
6 monto	double	☐	☐
7 descuento	double	☐	☒
8 total_compra	double	☐	☒

Nombre del Campo	Tipo de dato	Longitud	Valores Permitidos
Código_compra	Int		Not Null
Fecha_compra	Date		Not Null
Nombre_proveedor	Varchar	100	
Tipo_pago	Varchar	10	Not Null
Estado	Varchar	9	
Monto	Double		
Descuento	Double		
Total_compra	Double		Not Null



Tabla 3. Compra_producto.

	Nombre	Tipo	Primaria	No Nulo
1	 id_compra	int	☒	☒
2	 codigo_compra	int	☐	☐
3	 nombre_proveedor	varchar(100)	☐	☐
4	 codigo_producto	int	☐	☐
5	 cantidad_producto	int	☐	☐

Nombres del campo	Tipo de dato	Longitud	Valores permitidos
Id_compra	Int		Not Null
Código_compra	Int		
Nombre_proveedor	Varchar	100	
Código_producto	Int		
Cantidad_producto	Int		



Tabla 4. Factura.

	Nombre	Tipo	Primaria	No Nulo
1	 codigo_factura	int	☒	☒
2	 fecha_factura	date	☐	☒
3	 hora_factura	time	☐	☒
4	 precio_total	float	☐	☒

Nombre del campo	Tipo de dato	Longitud	Valores permitidos
Código_ factura	Int		Not Null
Fecha_ factura	Date		Not Null
Hora_ factura	Time		Not Null
Precio_ total	Float		Not Null



Tabla 5. Factura_producto.

	Nombre	Tipo	Primaria	No Nulo
1	id_fp	int	☒	☒
2	codigo_factura	int	☐	☐
3	cedula_cliente	varchar(16)	☐	☐
4	codigo_producto	int	☐	☐
5	nombre_producto	varchar(43)	☐	☐
6	cantidad_producto	int	☐	☐
7	precio_unitario	double	☐	☐
8	cedula_usuario	varchar(16)	☐	☐
9	tipo	varchar(7)	☐	☐

Nombre del campo	Tipo de dato	Longitud	Valores permitidos
Id_fp	Int		Not Null
Código_factura	Int		
Cedula_cliente	Varchar	16	
Código_producto	Int		
Nombre_producto	Varchar	43	
Cantidad_producto	Int		
Precio_unitario	Double		
Cedula_usuario	Varchar	16	
Tipo	Varchar	7	



Tabla 6. Producto.

	Nombre	Tipo	Primaria	No Nulo
1	codigo_producto	int	☒	☒
2	nombre_comercial	varchar(60)	☐	☒
3	nombre_generico	varchar(60)	☐	☒
4	presentacion	varchar(40)	☐	☒
5	concentracion	varchar(30)	☐	☒
6	nombre_proveedor	varchar(100)	☐	☒
7	nombre_laboratorio	varchar(40)	☐	☒
8	fecha_ingreso	date	☐	☒
9	fecha_vence	date	☐	☒
10	tipo	varchar(80)	☐	☒
11	cantidad	int(6)	☐	☒
12	precio_costo	float	☐	☒
13	precio_venta	float	☐	☒

Descripción del campo	Tipo de dato	Longitud	Valores permitidos
Código_producto	Int		Not Null
Nombre_comercial	Varchar	60	Not Null
Nombre_genérico	Varchar	60	Not Null
Presentación	Varchar	40	Not Null
Concentración	Varchar	30	Not Null
Nombre_proveedor	Varchar	100	Not Null
Nombre_laboratorio	Varchar	40	Not Null
Fecha_ingreso	Date		Not Null
Fecha_vence	Date		Not Null
Tipo	Varchar	80	Not Null
Cantidad	Int	6	Not Null
Precio_costo	Float		Not Null
Precio_venta	Float		Not Null



Tabla 7. Proveedor.

	Nombre	Tipo	Primaria	No Nulo
1	nombre_proveedor	varchar(100)	☒	☒
2	direccion_proveedor	varchar(100)	☐	☒
3	telefono_proveedor	int(8)	☐	☒
4	email_proveedor	varchar(60)	☐	☒
5	fax	varchar(15)	☐	☒
6	siglas	varchar(50)	☐	☒

Nombre del campo	Tipo de dato	Longitud	Valores permitidos
Nombre_ proveedor	Varchar	100	Not Null
Dirección_ proveedor	Varchar	100	Not Null
Teléfono_ proveedor	Int	8	Not Null
Email_ proveedor	Varchar	60	Not Null
Fax	Varchar	15	Not Null
Siglas	Varchar	50	Not Null



Tabla 8. Roles.

Campos				
	Nombre	Tipo	Primaria	No Nulo
1	id_usuario	int	☒	☒
2	nuevo_cliente	int(3)	☐	☐
3	nuevo_usuario	int(3)	☐	☐
4	contrasena_usuario	int(3)	☐	☐
5	nuevo_producto	int(3)	☐	☐
6	nuevo_proveedor	int(3)	☐	☐
7	nueva_compra	int(3)	☐	☐
8	nueva_venta	int(3)	☐	☐
9	editar_cliente	int(3)	☐	☐
10	editar_usuario	int(3)	☐	☐
11	editar_producto	int(3)	☐	☐
12	editar_proveedor	int(3)	☐	☐
13	buscar_producto	int(3)	☐	☐
14	buscar_cliente	int(3)	☐	☐
15	buscar_proveedor	int(3)	☐	☐
16	buscar_venta	int(3)	☐	☐
17	buscar_compra	int(3)	☐	☐
18	buscar_usuario	int(3)	☐	☐
19	reporte_dogras	int(3)	☐	☐
20	reporte_inventario	int(3)	☐	☐
21	reporte_venta	int(3)	☐	☐
22	reporte_ymes	int(3)	☐	☐
23	reporte_cxc	int(3)	☐	☐
24	reporte_cxp	int(3)	☐	☐
25	reporte_prodvenidos	int(3)	☐	☐
26	respaldar_bbdd	int(3)	☐	☐
27	restaurar_bbdd	int(3)	☐	☐



Nombres del campo	Tipo de dato	Longitud	Valores permitidos
Id_ usuario	Int		Not Null
Nuevo_ cliente	Int	3	
Nuevo_ usuario	Int	3	
Contraseña_ usuario	Int	3	
Nuevo_ producto	Int	3	
Nuevo_ proveedor	Int	3	
Nueva_ compra	Int	3	
Nueva_ venta	Int	3	
Editar_ cliente	Int	3	
Editar_ usuario	Int	3	
Editar_ producto	Int	3	
Editar_ proveedor	Int	3	
Buscar_ producto	Int	3	
Buscar_ cliente	Int	3	
Buscar_ proveedor	Int	3	
Buscar_ venta	Int	3	
Buscar_ compra	Int	3	
Buscar_ usuario	Int	3	
Reporte_ drogas	Int	3	
Reporte_ inventario	Int	3	
Reporte_ vdía	Int	3	
Reporte_ vmes	Int	3	
Reporte_ cxc	Int	3	
Reporte_ cpx	Int	3	
Reporte_ prodvencidos	Int	3	
Respaldar_ bbdd	Int	3	
Restaurar_ bbdd	Int	3	



Tabla 9. Usuario.

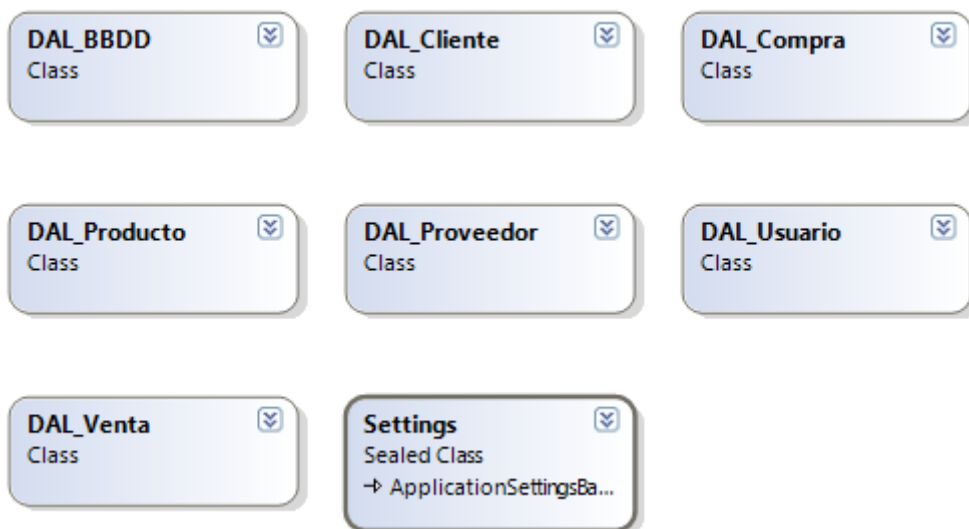
	Nombre	Tipo	Primaria	No Nulo
1	cedula_usuario	varchar(16)	☒	☒
2	nombres_usuario	varchar(30)	☐	☒
3	apellidos_usuario	varchar(30)	☐	☒
4	username	varchar(20)	☐	☒
5	contrasena_usuario	varchar(40)	☐	☒
6	direccion_usuario	varchar(100)	☐	☒
7	telefono_usuario	int(8)	☐	☒
8	email_usuario	varchar(60)	☐	☒
9	id_usuario	int	☐	☒
10	perfil	varchar(20)	☐	☒
11	estado	varchar(8)	☐	☐

Nombre del campo	Tipo de dato	Longitud	Valores permitidos
Cedula_usuario	Varchar	16	Not Null
Nombres_usuario	Varchar	30	Not Null
Apellidos_usuario	Varchar	30	Not Null
Username	Varchar	20	Not Null
Contraseña_usuario	Varchar	40	Not Null
Dirección_usuario	Varchar	100	Not Null
Teléfono_usuario	Int	8	Not Null
Email_usuario	Varchar	60	Not Null
Id_usuario	Int		Not Null
Perfil	Varchar	20	Not Null
Estado	Varchar	8	



DIAGRAMAS DE CLASES.

CAPA DE DATOS.



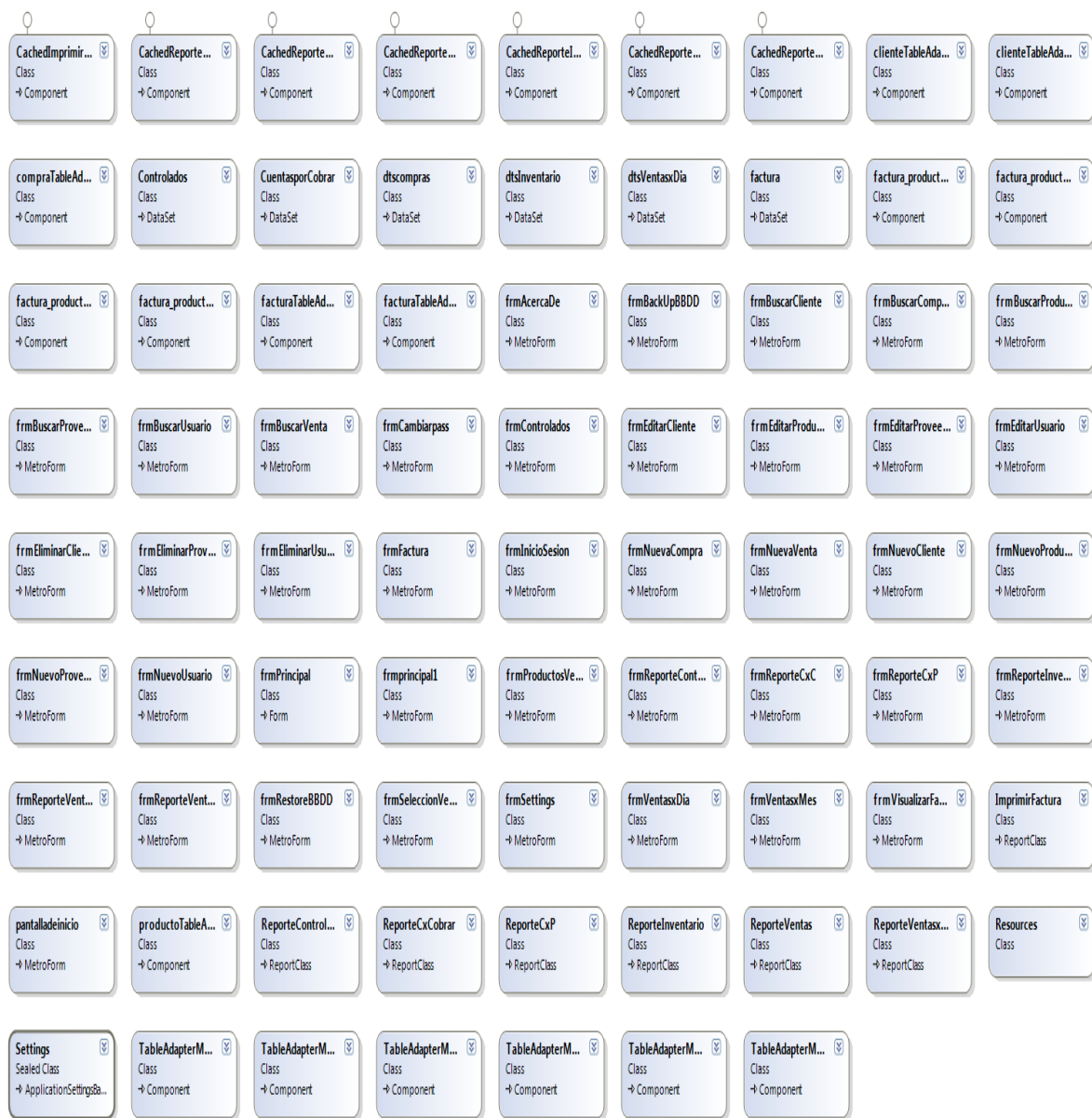


CAPA DE NEGOCIOS.



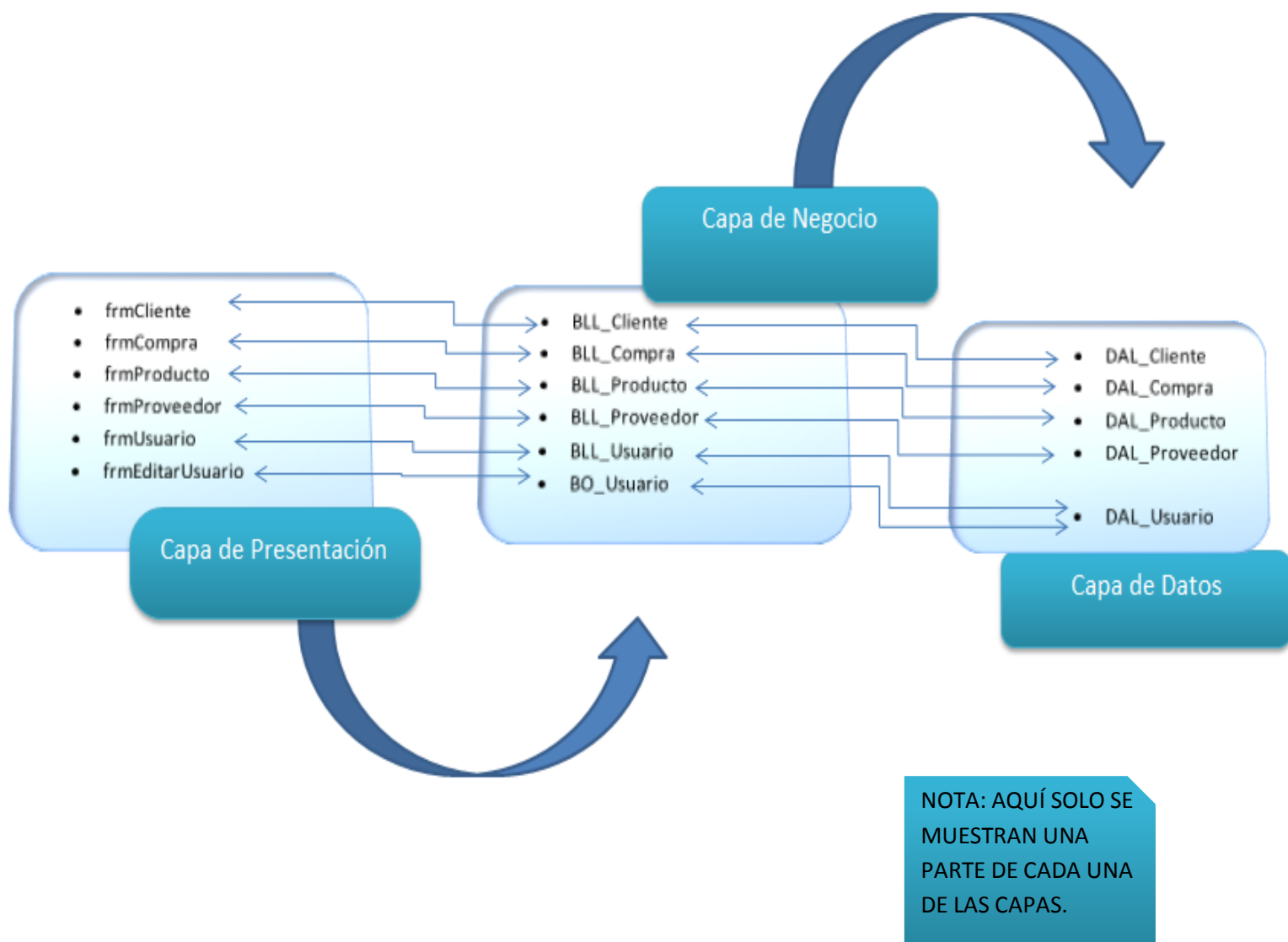


CAPA DE PRESENTACIÓN.





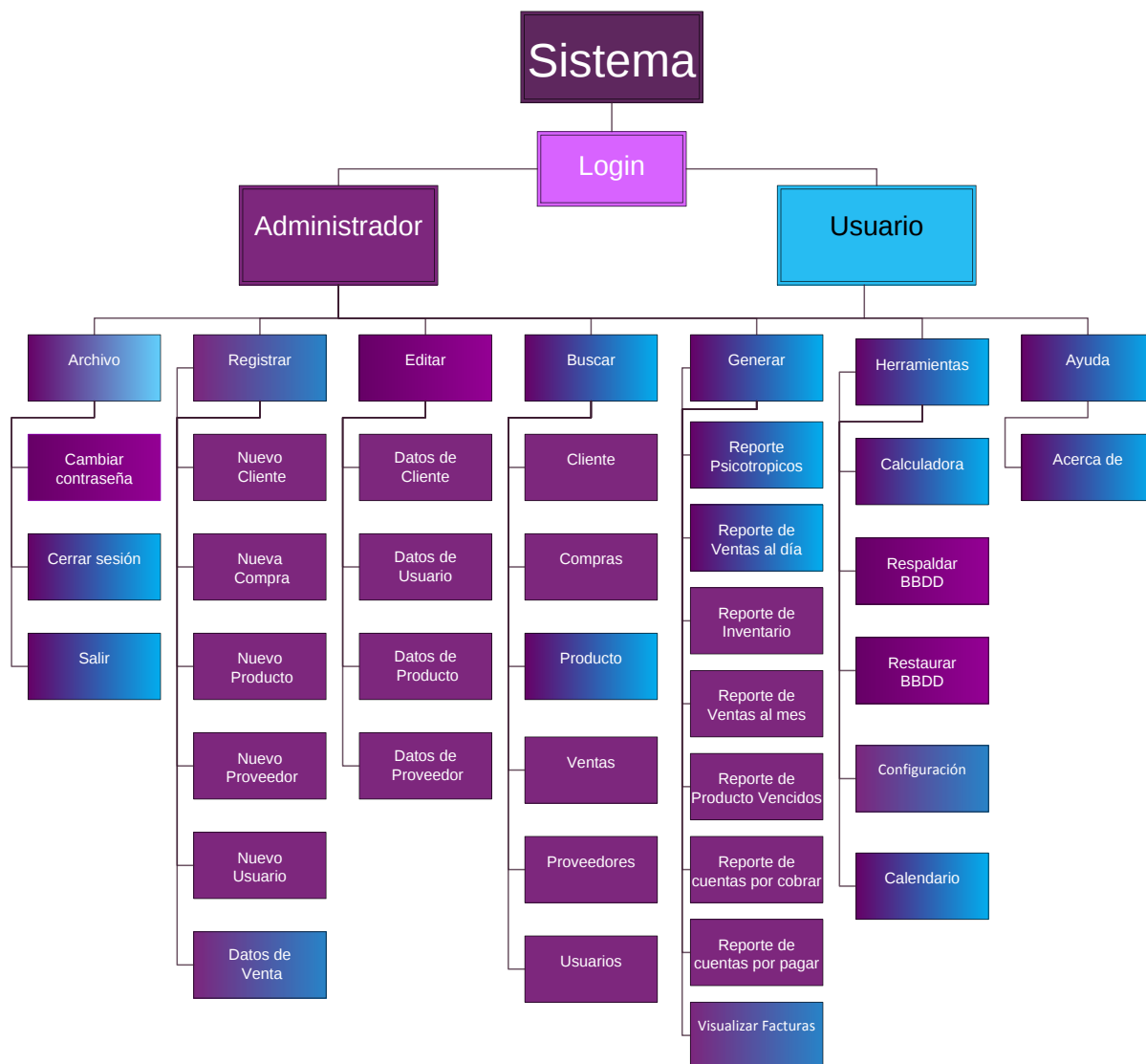
Gráfica de Acceso entre Capas.





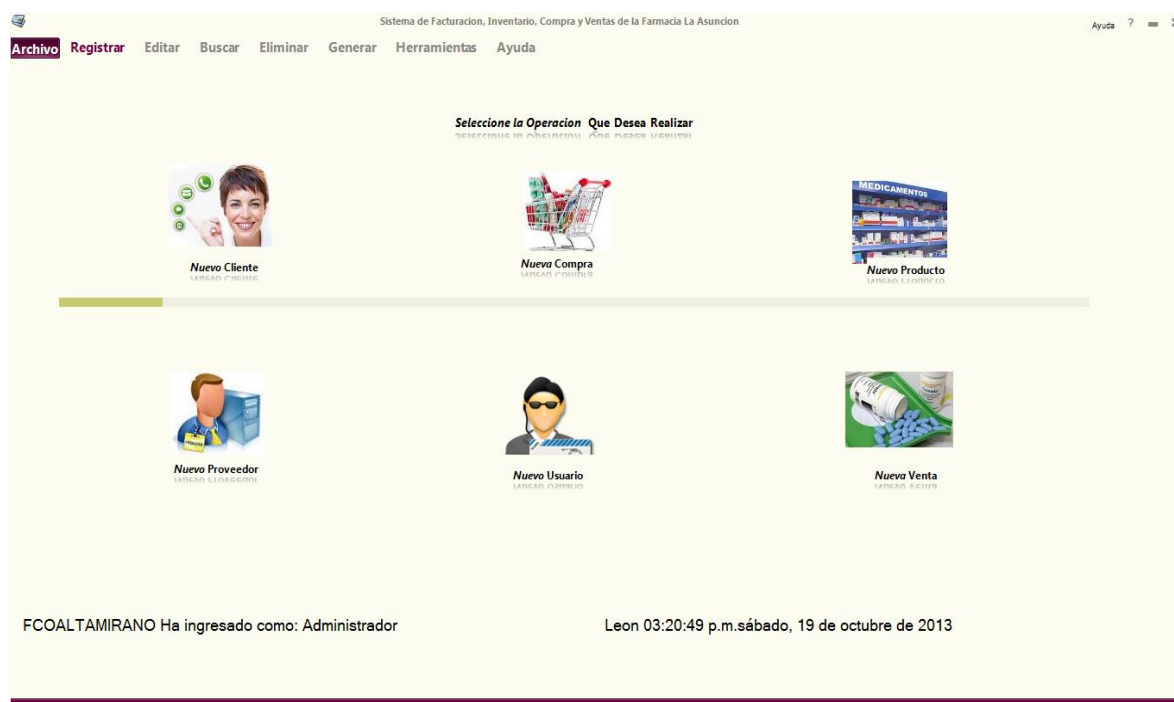
DISEÑO ARQUITECTONICO.

Diseño Arquitectónico de Sistema de Automatización de los Procesos de Ventas, Compras, Facturación e Inventario, de la Farmacia La ASUNCION.





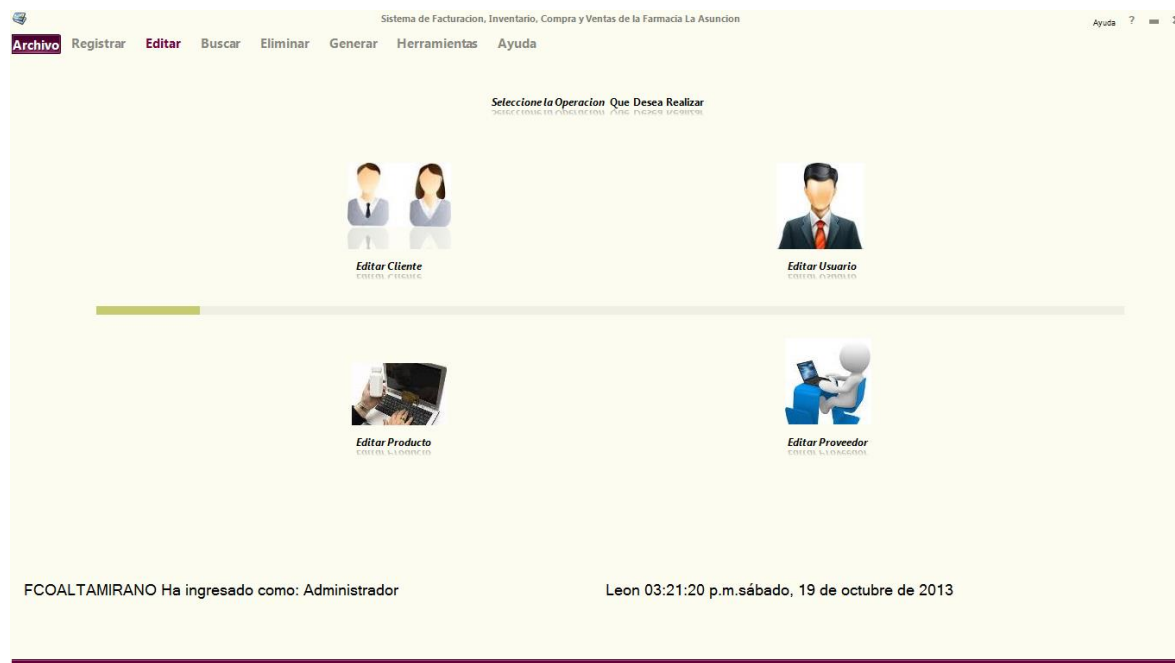
INTERFACES.



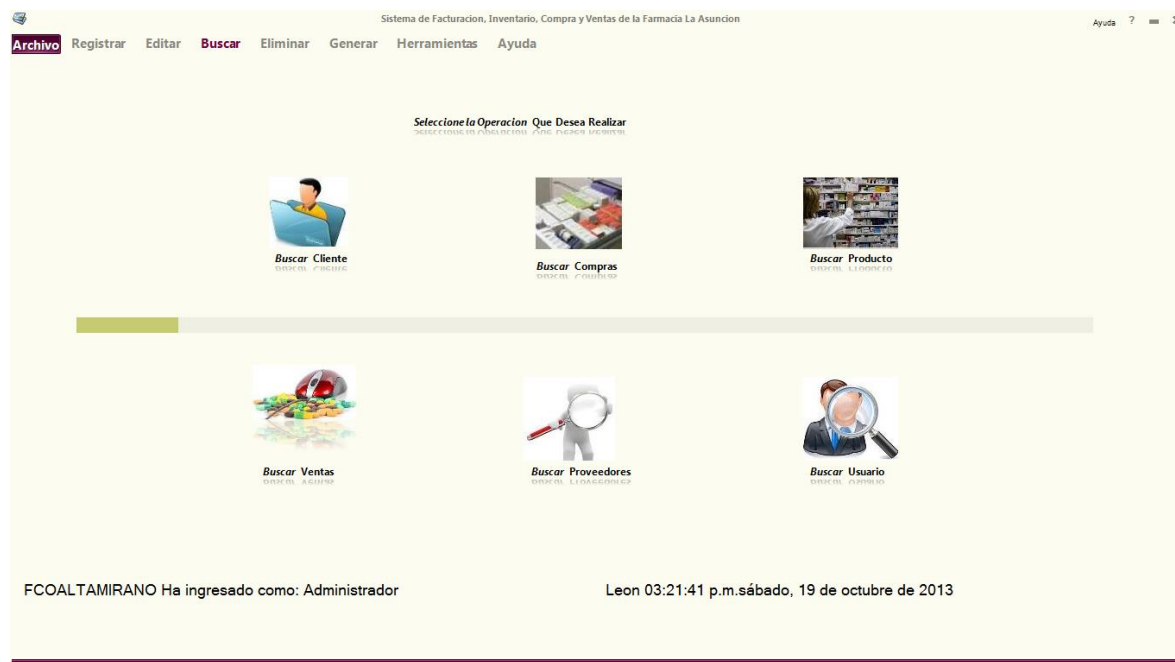
FORMULARIO PRINCIPAL DE LA APLICACIÓN.



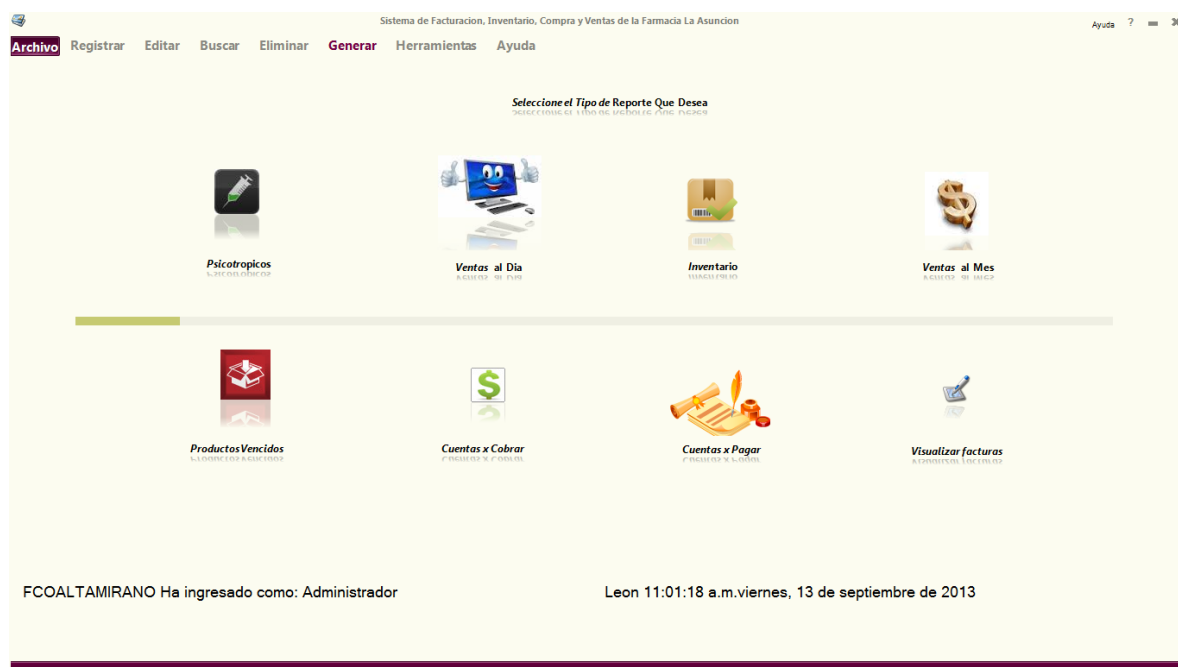
EDITAR:



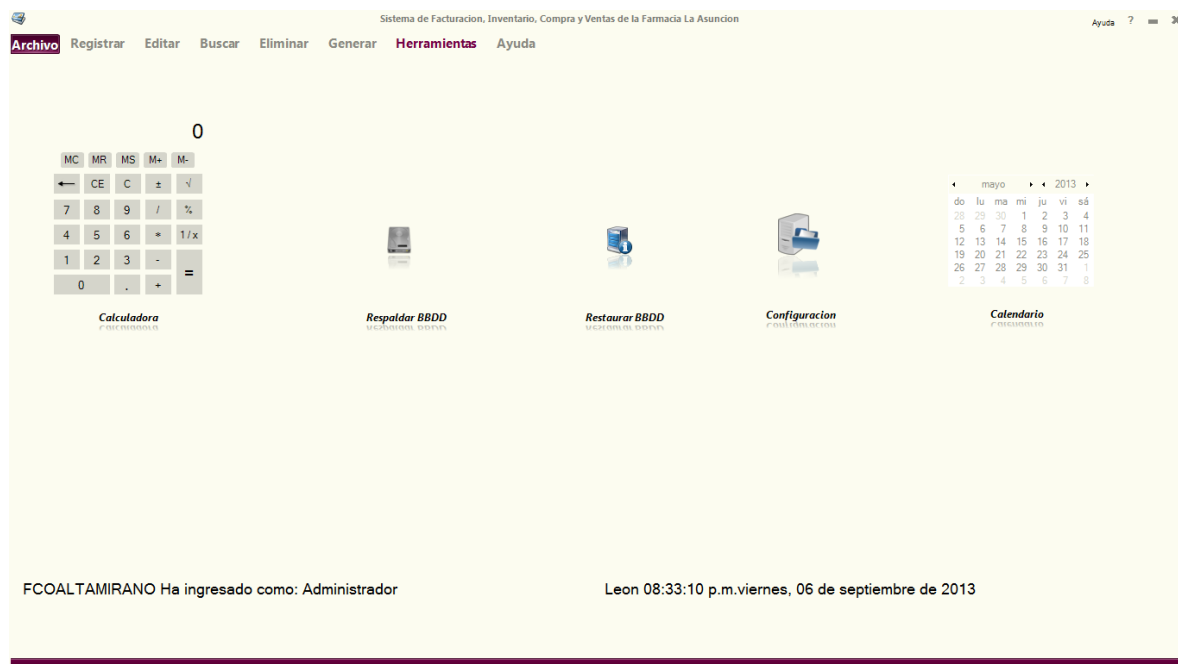
BUSCAR:



REPORTES:



HERRAMIENTAS:





CONCLUSIONES.

Al haber culminado con nuestro trabajo monográfico, hemos logrado cumplir con cada uno de los objetivos planteados, dando como resultado el desarrollo final de la aplicación Automatización de los procesos de Ventas, Compras, Facturación e Inventario de la Farmacia La Asunción del municipio de Telica-León.

Para el desarrollo de esta aplicación usamos nuestros conocimientos adquiridos durante los años de estudio, sobre programación, base de datos, ingeniería del software, análisis y diseño de sistemas, con el fin de proporcionar al usuario final un sistema eficiente e útil en su labor diaria.

La aplicación lleva un control de la información referente a los procesos que se realizan en dicha farmacia, mostrándola de manera organizada y actualizada. Dentro de las funciones que presenta el software cabe destacar el acceso restringido a usuarios no autorizados al inicio de la aplicación para evitar la alteración de la información que en dicha se maneja. Así como también la realización de copia y restauración de la base de datos y los distintos reportes que en ella se generan.

Esta aplicación será de gran ayuda para los usuarios que lo implementaran ya que les facilita su trabajo en el control y manejo de la información, generando reportes actualizados en el momento que los usuarios deseen. Con la elaboración de este software podemos decir que Visual Studio .NET 2010(Visual C#.NET) es una plataforma de mucho potencial al incluir muchas herramientas que hacen al programador hacer su trabajo más fácil y esto le permite crear aplicaciones de gran calidad.

La aplicación fue implementada en una red cliente- servidor para mejor aprovechamiento de la misma.



RECOMENDACIONES.

- Agregar al sistema un módulo contable y el módulo devoluciones.
- Utilizar procedimientos almacenados.
- Verificar la capacidad del ordenador donde será instalado para su buen funcionamiento y así no tener problemas a la hora de utilizarlo.



BIBLIOGRAFIA.

- Suehring Steve - MySQL Bible.
- Salvador Pozo Coronado, MySQL con Clase Gestión de Base de Datos – Curso de MySQL por, Mayo 2005. www.conclase.net
- Silberschatz Abraham, Korth Henry, Sudarshan.S .Fundamentos de Bases de Datos, Cuarta Edición. 2002.
- Microsoft Help Viewer 1.1 – Catalogo VS_2010_ES-ES.
- Ferguson Jeff, Patterson Brian, Beres Jason. La Biblia de C#. ANAYA Multimedia 2003.
- Charte Ojeda Francisco. Programación Visual C#.Net. ANAYA Multimedia.

Sitios Web

- <http://www.mysql.com/>
- <http://www.devcomponents.com/>
- <http://mysqlbackupnet.codeplex.com/>
- <http://es.wikipedia.org/wiki/Criptografia>



ANEXOS.



Instalaciones Farmacia La Asunción.



Productos Farmacia La Asunción.