

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

**SISTEMA DE CONTROL DE VENTAS Y STOCK
PARA MINIMARKET “FUTURO”**

**MELISA KARINA ASTUDILLO FAVI
SEBASTIÁN ANDRÉS ESPINOZA ARANDA**

INFORME FINAL DE PROYECTO
PARA OPTAR AL TÍTULO PROFESIONAL DE
INGENIERO DE EJECUCIÓN EN INFORMÁTICA

JULIO 2015

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

**SISTEMA DE CONTROL DE VENTAS Y STOCK
PARA MINIMARKET “FUTURO”**

**MELISA KARINA ASTUDILLO FAVI
SEBASTIÁN ANDRÉS ESPINOZA ARANDA**

Profesor Guía: **Iván Mercado Bermúdez.**

Carrera: **Ingeniería de Ejecución en Informática**

JULIO 2015

Dedicatoria

A Mamá y Papá, por siempre creer en mí y guiar mi
Camino, todos mis triunfos son gracias a ustedes.

A ti, por darme ánimo y motivarme cada vez

Que sentía que mi mundo se venía abajo.

Gracias por ser mis pilares y

Ganas de seguir adelante.

Con Amor.

Melisa Karina Astudillo Favi.

Dedicatoria

A mi familia por apoyarme en este trayecto,

Por no dejar que me diera por vencido

Sin ellos jamás hubiera llegado

Hasta donde estoy ahora.

Gracias por ser

Mi familia.

Los amo.

Sebastián Andrés Espinoza Aranda.

Índice

Resumen	iii
Abstract	iii
Lista de Figuras	iv
Lista de Tablas	v
1 Introducción	1
2 Definición del Problema	2
2.1 Problema y Solución	2
2.2 Flujos de Trabajo	3
2.2.1 Flujo de Trabajo Actual.....	3
2.2.2 Flujo de Trabajo Con Sistema	4
3 Objetivos del Sistema	6
3.1 Objetivo General	6
3.2 Objetivos Específicos	6
4 Especificación de Requerimientos	7
4.1 Requerimientos Funcionales	7
4.2 Requerimientos no Funcionales	7
5 Análisis de Riesgo	8
5.1 Plan de Mitigación	9
5.2 Plan de Contingencia	10
6 Desarrollo	11
6.1 Paradigma de Desarrollo	11
6.2 Metodología de Desarrollo	12
6.3 Arquitectura de la Solución	12
6.4 Herramientas y Tecnologías	13
6.4.1 Microsoft Visual Studio	14
6.4.2 Visual Basic .NET	14
6.4.3 Microsoft SQL Server	14
6.4.4 Microsoft Visio.....	14
7 Diseño de la Solución	15
7.1 Casos de Uso	15
7.1.1 Caso de Uso General	15
7.2 Diagramas de Secuencia	17
7.2.1 Realizar Venta	17
7.2.2 Registrar Compra.....	18
7.2.3 Crear Producto.....	19
7.3 Diagrama de Clases	20
7.4 Modelo Entidad-Relación	21

7.5	Modelo Relacional	22
7.6	Interfaces Software Futuro.....	22
8	Diseño de Pruebas.....	26
9	Conclusión	28
10	Bibliografía.....	29
Anexos	A-1
A:	Casos de Uso.....	A-1

Resumen

Debido a la reciente adquisición del minimarket “Futuro”, por parte del padre de uno de los integrantes del grupo de trabajo, es que se requiere de un sistema para poder tener un registro de los movimientos realizados dentro del local, ya sea, compra y/o venta de productos. Lo anterior, es necesario para una buena administración, y generar un positivo desarrollo y crecimiento del negocio, de lo contrario se crea un descontrol, lo que puede provocar pérdidas importantes.

Para cumplir con lo solicitado por el cliente, y llevar a cabo así el objetivo principal, la problemática fue abarcada desde su diseño. Por lo que, en el siguiente documento se dan a conocer el dilema existente, los requerimientos y funcionalidades que debe poseer el sistema y la solución planteada.

Palabras Clave: Compra, venta, producto, sistema, desarrollo.

Abstract

Due to the recently acquisition of Minimarket “Futuro”, by the father of one of the members of the working group, is required a system to have a register of the movements made in the store, be it, the buy and/or sell of products. The previous, is necessary to a good administration and generate a positive development of the bussiness, on the opposite it creates a disarray which can cause important losses.

To achieve the request of the client and accomplish the main objective, the problem was cover from its design. Therefore, in the following document is given to know the existing problem, the requeriments and functionalities the system must have and the final solution.

Key words: buy, sale, product, system, development.

Lista de Figuras

Figura 2.1 Control de Ventas y Stock, Actual	4
Figura 2.2 Control de Ventas y Stock, Con Sistema	5
Figura 6.1 Arquitectura Cliente-Servidor de tres capas	13
Figura 7.1 Caso de Uso General	15
Figura 7.2 Diagrama de Secuencia Realizar Venta	17
Figura 7.3 Diagrama de Secuencia Registrar Compra	18
Figura 7.4 Diagrama de Secuencia Crear Producto.....	19
Figura 7.5 Diagrama de Clases	20
Figura 7.6 Modelo Entidad-Relación	21
Figura 7.7 Modelo Relacional	22
Figura 7.8 Interfaz Inicio de sesión	23
Figura 7.9 Interfaz Menu Principal.....	23
Figura 7.10 Interfaz Operación Venta	23
Figura 7.11 Interfaz Registrar Compra.....	24
Figura 7.12 Interfaz Mantenimiento de Productos	24
Figura 7.13 Interfaz Tipos de Informes	25

Lista de Tablas

Tabla 5.1 Análisis de Riesgo	8
Tabla 5.2 Plan de Mitigación.....	9
Tabla 5.3 Plan de Contingencia.....	10
Tabla 7.1 Especificación Caso de Uso Realizar Venta	15
Tabla 7.2 Especificación Caso de Uso Registrar Compra.....	16

1 Introducción

El dueño del minimarket “Futuro”, al ser nuevo en el rubro del comercio, desconoce la gran importancia de implementar un control sistematizado del flujo de información de inventario o de las actividades relacionadas con actualizaciones o ajustes. Esto, usualmente genera como consecuencia, la tenencia inexacta de datos contables y financieros, pérdidas considerables en el ejercicio y una dinámica poco eficiente en lo relacionado a despacho y cadenas de suministros.

En todos los almacenes del tipo minimarket, es necesario llevar un eficiente control de inventario, de lo contrario se genera un notorio descontrol en la organización, lo cual puede ser aprovechado para robos sistemáticos del tipo “merma”, lo que genera pérdidas considerables y a su vez causa un impacto negativo en el desarrollo y crecimiento de la empresa.

Si un micro-empresario no sabe con precisión lo que posee en almacenamiento, no puede saber qué es lo que debe ordenar. Si el balance de inventario es no perezoso, los elementos que hay en el punto de almacenamiento pueden quedar en residuos. Puesto que, en la mayoría de los casos el espacio de almacenamiento es un activo valioso, el uso de espacio de almacenamiento debe ser utilizado de manera eficiente.

En el siguiente informe, se pretende presentar al lector, el diseño de un sistema de información, basado en el control de ventas e inventario de un minimarket específico.

2 Definición del Problema

2.1 Problema y Solución

El minimarket “Futuro”, se encuentra ubicado en Avenida Gran Bretaña N°560, Playa Ancha, Valparaíso. Entre los productos que se venden, en este negocio, están: golosinas, abarrotes, congelados, bebidas gaseosas, helados, comida para mascotas, pan, etc.

El dueño del local, Don Gonzalo Astudillo Delgado, es quien atiende el negocio, el cual se encuentra abierto de lunes a sábado de 10:00 AM a 20.30 PM y los días domingos de 11.00 AM a 15:00 PM. El minimarket no posee empleados externos, el dueño a veces es ayudado por miembros de su familia, pero por distintas situaciones esto no es constante.

Esta persona además de preocuparse de la venta de productos, debe estar siempre pendiente de qué es lo que necesita comprar y esto no lo hace por stock exactamente sino por intuición, anotando en un cuaderno los pedidos que necesita hacer a cada proveedor (los cuales no son pocos), cada semana. Como si todo esto fuera poco, también va registrando, en un cuaderno, el monto de las ventas diarias (cada venta asociada a la boleta entregada al cliente), calculando así al final del día, sus ganancias. Al hacer esto, tiene conocimiento del dinero que ingresa, pero no así de los productos que vendió.

Por todo lo anterior, se cree que es necesario contar con un sistema de información o software que ordene y controle adecuadamente sus productos, es decir, automatizar las operaciones o registros con la finalidad de tener información real y actualizada, en el momento oportuno.

Como objetivo final, se pretende realizar un sistema de información específico, basado en el control de inventario del minimarket “Futuro”. Para cumplir con esto, se desea comenzar abarcando la problemática desde su diseño, para lo cual se utilizaran herramientas, tales como: uml, diagramas de actividad, diagramas de secuencia, etc.

Curiosamente, la buena administración de inventario puede poner fin a situaciones como las previamente descritas, entre otras, así como asegurar un apropiado nivel de productividad y eficiencia. Se espera que gracias a la implantación y aplicación efectiva de métodos, que se pretenden desarrollar para cumplir con los requerimientos de la administración de ventas y stock, el dueño del minimarket “Futuro”, con un nivel de esfuerzo humano relativamente menor, pueda mantener un seguimiento adecuado de las pérdidas y ganancias, que se reclamen a lo largo del tiempo.

2.2 Flujos de Trabajo

2.2.1 Flujo de Trabajo Actual

El dueño del minimarket “Futuro”, se preocupa de registrar en un cuaderno todos los movimientos diarios de su local, tales como, las ganancias e inversiones. Como es de saber, esta técnica es muy engorrosa y poco segura, ya que dicho cuaderno puede perderse o sufrir cualquier tipo de alteración, y con él se estropearían los datos que contiene pertenecientes al local.

El dueño también debería tener un claro conocimiento de los productos que se tienen en inventario, ya que de esta manera puede evitar “mermas”, las cuales pueden generar pérdidas y afectar negativamente el crecimiento del local. Pero, es muy complicado estar contando los productos, diariamente o cada ciertos momentos, pues es una pérdida de tiempo, por lo que esta actividad no se realiza en el local y por lo tanto, no se puede saber con certeza si se vendieron todos los productos almacenados en inventario, o se ha sido víctima de algún tipo de robo, ya sea de parte de los clientes o de alguno de los vendedores.

Por otro lado, el propietario debe estar siempre alerta, para poder recordar que productos son los que le hacen falta y necesita comprar. Las compras se realizan semanalmente, y el local se abastece de alrededor 15 proveedores distintos, por lo que hay veces en que el dueño olvida algunos productos o compra otros que quizás no eran tan necesarios.

En la figura 2.1, se muestra la forma en que, actualmente, se administran los movimientos de dinero y de los productos que se encuentran en inventario, en el minimarket en cuestión.

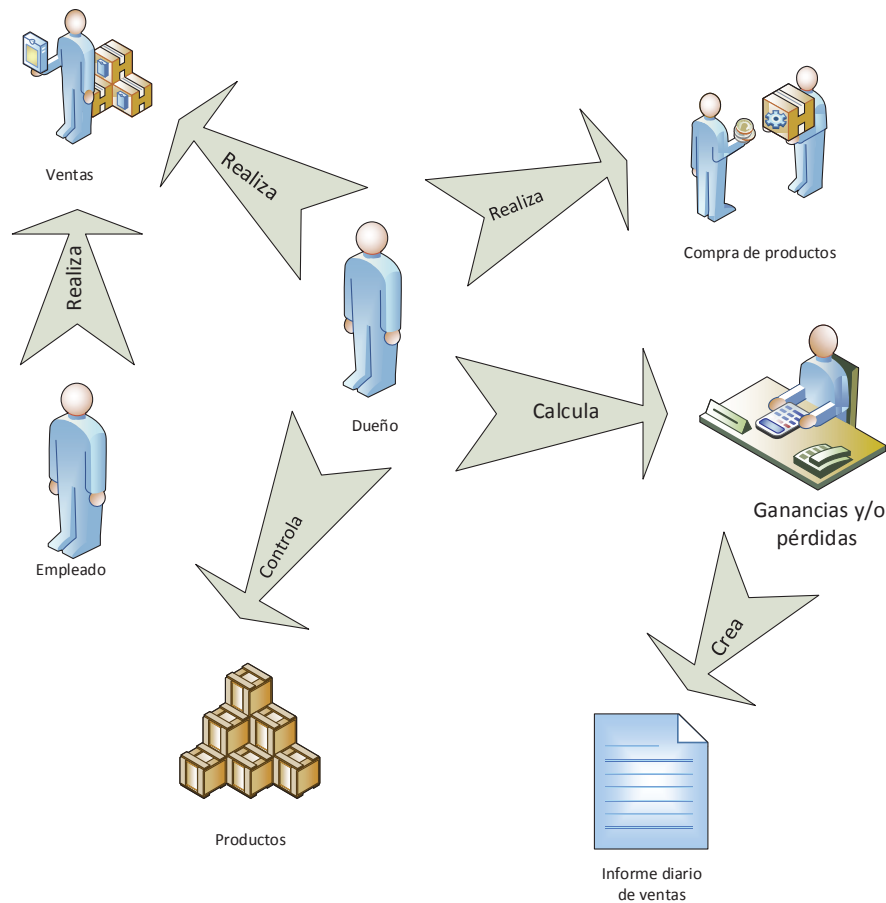


Figura 2.1 Control de Ventas y Stock, Actual.

2.2.2 Flujo de Trabajo Con Sistema

Como se puede apreciar en la Figura 2.1, el dueño del local, debe efectuar bastantes tareas, lo que puede traer como consecuencia una mala administración del negocio y con ello un lento crecimiento en el rubro. El sistema a desarrollar, permitirá al usuario realizar las ventas de una forma más cómoda, ya que controlará por sí solo el stock de los productos que se vayan registrando en cada venta, además podrá registrar las compras realizadas a sus proveedores. Gracias a esto el sistema podrá entregar al usuario información respecto a los stocks de los productos facilitando la decisión de que productos comprar.

También podrá generar documentos con los datos de las ventas, compras (devoluciones dentro del informe de compras) y consumo interno tanto diarias como mensuales (consumo interno será solo mensual).

En la figura 2.2, se presenta la forma en que se administrarán todos los movimientos de dinero e inventario realizados en el local, con el sistema a desarrollar.

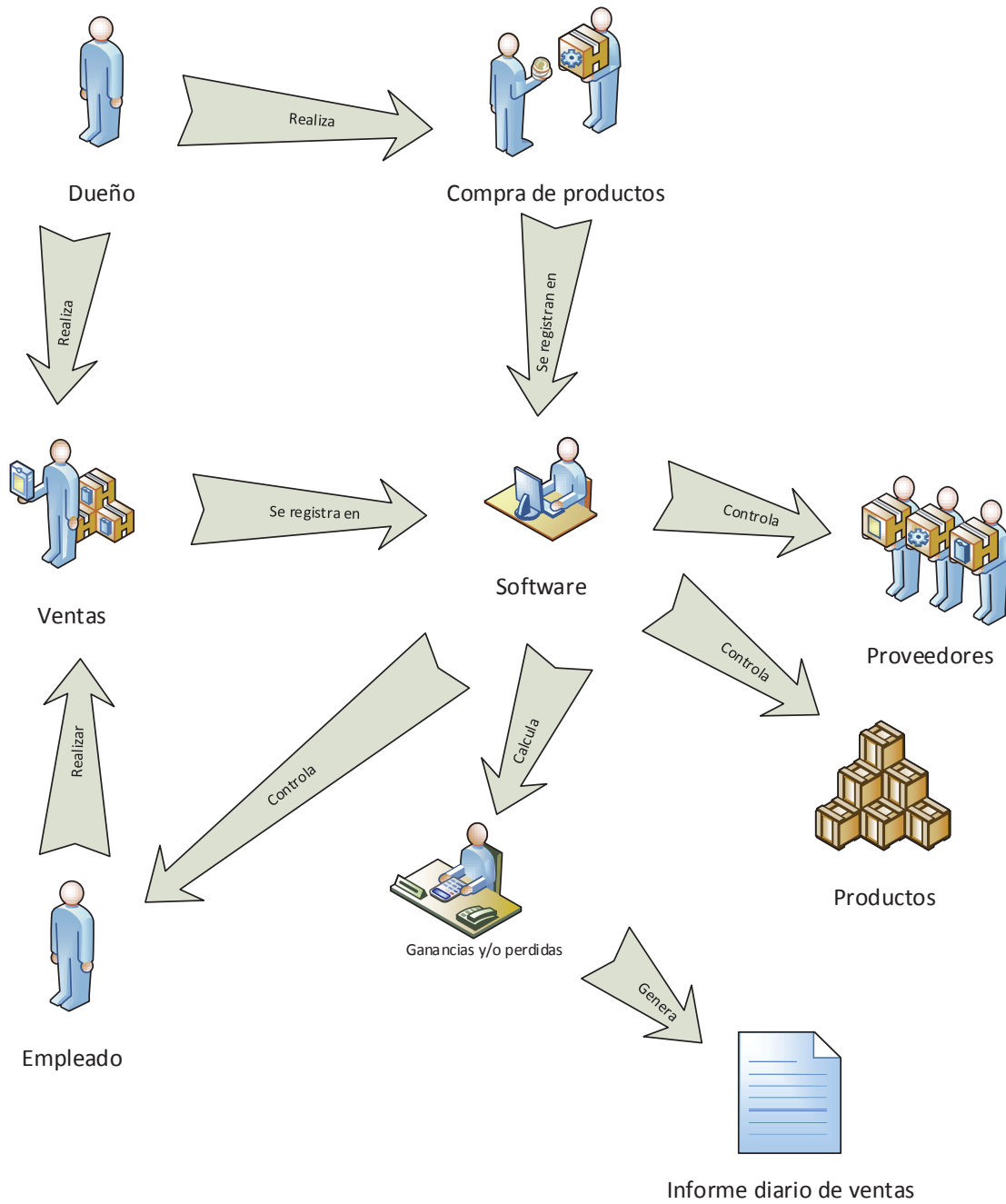


Figura 2.2 Control de Ventas y Stock, Con Sistema.

3 Objetivos del Sistema

3.1 Objetivo General

Se pretende desarrollar un sistema para el control de ventas y stock del Minimarket “Futuro”.

3.2 Objetivos Específicos

- Obtener conocimiento sobre minimarket “Futuro”, respecto a su organización, tipos de productos, empleados y proveedores asociados.
- Implementar un módulo de ventas con ayuda de la información obtenida, para facilitar el manejo del minimarket al dueño.
- Verificar junto al cliente que el sistema abarque y cumpla con los requerimientos especificados.

4 Especificación de Requerimientos

Dentro de los requerimientos se pueden identificar dos tipos. Están aquellos que son generales y los que son específicos, para poder tenerlos claros a continuación, se detallará cada uno de ellos.

4.1 Requerimientos Funcionales

El sistema tendrá que contar con las siguientes características:

- Deberá manejar productos con diferentes unidades de medida (Kilo, Litro y unidad) y con distinto impuesto.
- Deberá controlar el stock de productos (mínimo y un máximo para cada producto).
- Permitirá el ingreso de facturas o boletas de compra.
- Calculará las utilidades y el vuelto.
- Tendrá que controlar otras transacciones (mermas, devoluciones y/o consumo interno).
- Deberá ser accesible a través de un usuario y contraseña.
- Registrará la venta de cada vendedor.
- Permitirá la creación, modificación y/o eliminación de productos, familia de productos, operadores (vendedores) y proveedores.
- Generará informe de detección de compra según el stock mínimo definido para cada producto.
- Deberá chequear la cantidad de productos que hay en stock.
- Permitirá la creación de informes diarios y mensuales relacionados con las ventas y compras.

4.2 Requerimientos no Funcionales

- Interfaz del sistema debe ser simple, intuitiva, fácil de usar y aprender.
- Permitir el ingreso de productos al sistema, a través de un lector de código de barras.
- Estabilidad del sistema.
- El software tiene que ser escalable, para así poder adaptarse y mejorar a través del tiempo.
- Las plataformas a utilizar serán Windows 7 y Windows 8.
- El lenguaje de programación a utilizar será Visual Basic .NET.

5 Análisis de Riesgo

Durante el proceso de desarrollo de un software es normal que se puedan presentar inconvenientes al momento de llevar a cabo las actividades correspondientes. Como medida para reducir la aparición simultánea de éstos peligros, es que el grupo de trabajo elabora un análisis de posibles riesgos que pudiesen presentarse durante el diseño de la solución propuesta.

Riesgo	Probabilidad de Ocurrencia	Tipo de Impacto	Plan de Mitigación Asociado
Miembro del equipo abandona el proyecto	5%	Medio	1
Inconsistencia al momento de integrar la compatibilidad de módulos desarrollados por distintos programadores	50%	Alto	2
Mala organización en el almacenamiento de datos	10%	Alto	3
Pobre coordinación entre programadores ante nuevas versiones de software	50%	Medio	4
Debilidades relacionadas con desconocimiento de temas legales acerca del manejo de la información	20%	Catastrófico	5
Sistema es rechazado por cliente	50%	Catastrófico	6
Mala distribución de recursos.	20%	Medio	7
Falla elemental en el servidor	10%	Alto	8
Incumplimiento de plazos establecidos para desarrollo de proyecto	50%	Medio	9
Un miembro renuncie a la empresa.	5%	Medio	9

Un miembro presenta una licencia médica	20%	Medio	9
Retraso a causa de fenómenos naturales	5%	Alto	10

Tabla 5.1 Análisis de Riesgos

5.1 Plan de Mitigación

Consisten en medidas que el equipo de trabajo adoptará, para evitar la aparición de riesgos como los anteriormente identificados.

Identificador	Medidas para minimizar o mitigar los riesgos	Plan de Contingencia Asociado
1	Al inicio de cada fase del proyecto, se realizará un chequeo del estado de cada miembro del equipo de trabajo.	3
2	Se efectuarán pruebas de compatibilidad por cada avance, implementado a éstos en el sistema general, para arreglar errores e incompatibilidades.	3
3	El analista de sistema realizará un chequeo semanal de cómo se están almacenando los datos.	6
4	Se asignarán integrantes del equipo que mantienen buena comunicación y coordinación para una función dada, sin interferir con la productividad del proceso.	3
5	Capacitación previa al comienzo del proyecto de los miembros del equipo de trabajo.	1

Tabla 5.2 Plan de Mitigación

5.2 Plan de Contingencia

Consisten en una serie de definiciones acerca de qué acciones realizará la micro-empresa cuando ya se haya producido uno de los riesgos descritos en la tabla de análisis anterior

Identificador	Medida de Contingencia
1	Requerir asesoría legal externa
2	Coordinar una reunión extraordinaria con el cliente para conocer el motivo del rechazo y los posibles cambios que se le puedan hacer al proyecto.
3	Se establecerán nuevas asignaciones tareas entre los integrantes del equipo que quedan y los potenciales remplazantes.
4	Recuperar información desde alguno de los respaldos realizados.
5	Disminuir levemente el flujo de recursos utilizados por los miembros del equipo de trabajo.
6	Reestructuración del almacenamiento de datos por parte del desarrollador de sistema.

Tabla 5.3 Plan de Contingencia

6 Desarrollo

A continuación se detallarán todos los aspectos tomados en cuenta tanto antes de comenzar a desarrollar el software como durante el período de avance de éste.

6.1 Paradigma de Desarrollo

El método de desarrollo que será utilizado en el sistema, es el Modelo Iterativo Enfocado al Desarrollo Incremental. La razón por la cual se ha optado trabajar con este método es que, de la unión del ciclo de vida iterativo y el incremental al final de cada iteración da como resultado requisitos terminados y con ello una interpretación mucho más estable del sistema, con más cualidades, minimizando el número de errores que se producen en el desarrollo y aumentando la calidad, agregando además nuevas funcionalidades respecto a las versiones anteriores.

Algunos beneficios que éste modelo puede brindar con respecto al cliente, es que éste puede dar comienzo al proyecto con requisitos de alto nivel, de manera que se vayan mejorando en siguientes iteraciones. Sólo es necesario conocer con más detalle los requisitos de las primeras iteraciones, los que más valor aportan. El cliente puede obtener resultados importantes desde las primeras iteraciones y como máximo puede perder los recursos dedicados a solo una iteración, no los de todo el proyecto.

Para el equipo de trabajo, este modelo, permite mitigar desde el inicio los posibles riesgos del proyecto. Al tener conocimiento de estos riesgos, es posible iniciar su mitigación de manera anticipada. Al finalizar cada iteración, el equipo decidirá cómo enriquecer el trabajo en proceso, con respecto a la experiencia obtenida. De esta manera, es posible planificar los cambios necesarios para aumentar la productividad y calidad desde las primeras iteraciones.

A continuación, se dan a conocer los incrementos e iteraciones realizadas a lo largo del desarrollo del sistema.

- Primer Incremento: Se crearon las tablas que se utilizaran en la Base de Datos, y a su vez las funciones mantenedoras, las cuales permiten guardar y cargar datos del software a la Base de Datos, en conjunto con sus respectivas interfaces.
- Primera Iteración: Se corrigieron errores en relación con las tablas de venta y compra, debido a un problema de redundancia. Además se mejoró la mantención de productos respecto a los productos sin código de barra.
- Segundo Incremento: Se desarrollaron las funciones principales del software, las cuales vienen siendo la de venta y compra, junto a sus respectivas interfaces.

- Segunda Iteración: Se realizaron cambios en las interfaces de compra y venta respecto a su tamaño y a su vez se mejoró el manejo del stock al permitir cifras negativas.
- Tercer Incremento: Se crearon las funciones de respaldo y documentación del software, este último se refiere a los informes de venta, compra y consumo interno, además implementó un gráfico que permite al usuario comparar el mes actual con el anterior (ya sea venta, compra o consumo interno), todo esto en conjuntos a sus respectivas interfaces.
- Tercera Iteración: Se mejoraron las funciones de documentación añadiendo stock mínimo y stock actual a la lista de informes que se pueden generar, además se corrigió un error respecto a los gráficos debido a que se repetían algunos productos y se mejora la comparación de meses cambiándolo por un histórico (en vez del mes anterior).

6.2 Metodología de Desarrollo

La metodología a utilizar será Orientada a Objetos, ya que además de ser cómoda para el grupo de trabajo, también brinda beneficios, de los que se pueden destacar, una mayor flexibilidad a la hora de querer realizar algún tipo de modificación o mantenimiento al software, el código puede ser reutilizado y con esto se incrementa la productividad, se puede crear un sistema más sencillo para los usuarios (los datos innecesarios están ocultos) y facilita el trabajo en equipo.

6.3 Arquitectura de la Solución

Se cree que la arquitectura física que se adecúa de mejor manera al sistema a desarrollar, es la Arquitectura Cliente-Servidor de tres capas, debido a que permite separar de mejor manera, la forma en la que se mostrará el sistema. También como ventaja de ésta arquitectura, está la corrección de errores o cambios, ya que si falla o se desea cambiar alguna parte del sistema, sólo hay que dirigirse a la capa en la cual pertenece el cambio o fallo y se modifica ese nivel sin entrar a ver el código completo.

Esta arquitectura se divide en tres capas, las cuales son: Presentación, Procesamiento o Negocio y Administración de Datos o Base de Datos. Cada capa solamente tiene relación con la siguiente. A continuación, se explicitarán las diferentes capas y como están organizadas:

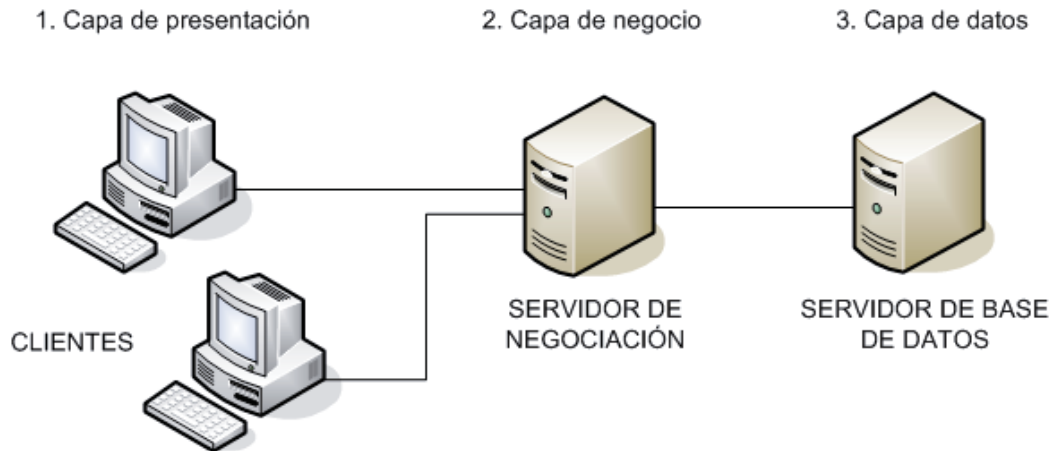


Figura 6.1 Arquitectura Cliente-Servidor de tres Capas.

- Capa de Presentación: Es la que se encarga de que el sistema interactúe con el usuario y viceversa, muestra el sistema al usuario, le presenta la información y obtiene la información del usuario en un mínimo de proceso. También es conocida como interfaz gráfica y debe ser amigable, fácil de entender y de utilizar para el usuario. Ésta sólo trabaja con la capa de Procesamiento.
- Capa de Negocio: Será la capa que contiene los procesos a realizar con la información recibida desde la capa de presentación, se reciben las peticiones que el usuario ha realizado, se procesa la información y se envían las respuestas después de realizar el proceso. Esta capa se comunica con la capa de presentación, para recibir las peticiones y otorgarle las respuestas, también interactúa con la capa de acceso de datos, para almacenar o recuperar la información que se encuentra en la base de datos.
- Capa de Datos: En esta capa es donde residen los datos del sistema y de los usuarios. Está compuesta por uno o más gestores de bases de datos, los cuales se encargan del almacenamiento de datos. Su principal función es la de almacenar y retornar datos a la capa de negocio.

6.4 Herramientas y Tecnologías

Para el desarrollo del sistema que se construirá, es necesario el uso de distintas herramientas, para así permitir y facilitar la creación de distintas funcionalidades del software que se realizaran a lo largo del proyecto.

6.4.1 Microsoft Visual Studio

Microsoft Visual Studio es un entorno de desarrollo integrado, para sistemas operativos Windows. Soporta múltiples lenguajes de programación y proporciona todo lo necesario a los desarrolladores para poder crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET.

6.4.2 Visual Basic .NET

Visual Basic .NET (VB.NET) es un lenguaje de programación orientado a objetos que se puede considerar una evolución de Visual Basic implementada sobre el framework .NET. Su introducción resultó muy controvertida, ya que debido a cambios significativos en el lenguaje VB.NET no es compatible hacia atrás con Visual Basic, pero el manejo de las instrucciones es similar a versiones anteriores de Visual Basic, facilitando así el desarrollo de aplicaciones más avanzadas con herramientas modernas.

Actualmente es un lenguaje de programación bastante popular entre los desarrolladores de software, ya que brinda un completo número de características para hacer que el desarrollo de aplicaciones sea realmente rápido y simple.

6.4.3 Microsoft SQL Server

Microsoft SQL Server, es un sistema para la gestión de bases de datos creado por Microsoft, basado en el modelo relacional. El lenguaje de desarrollo utilizado es Transact-SQL, una implementación del estándar ANSI del lenguaje SQL, el cual es utilizado para manipular y recuperar datos, crear tablas y definir relaciones entre ellas, lo que lo hace capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.

6.4.4 Microsoft Visio

Microsoft Visio es un software de dibujo vectorial para Microsoft Windows. Las herramientas que lo componen permiten realizar diagramas de oficinas, bases de datos, flujo de programas, UML, y más, que permiten iniciar al usuario en los lenguajes de programación. Como resultado de esto los desarrolladores de software pueden entenderse mejor con sus clientes.

7 Diseño de la Solución

7.1 Casos de Uso

Un caso de uso es una secuencia de transacciones, desarrollada por un sistema en respuesta a un evento que inicia un actor sobre el propio sistema. Este tipo de diagramas, sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas.

7.1.1 Caso de Uso General

En el caso de uso general se muestran las acciones que pueden llevar a cabo los usuarios registrados en el sistema. Cabe destacar, que el administrador posee más funcionalidades que un usuario común.

El resto de los casos de uso y sus respectivas especificaciones, pueden ser consultados en Anexo A.

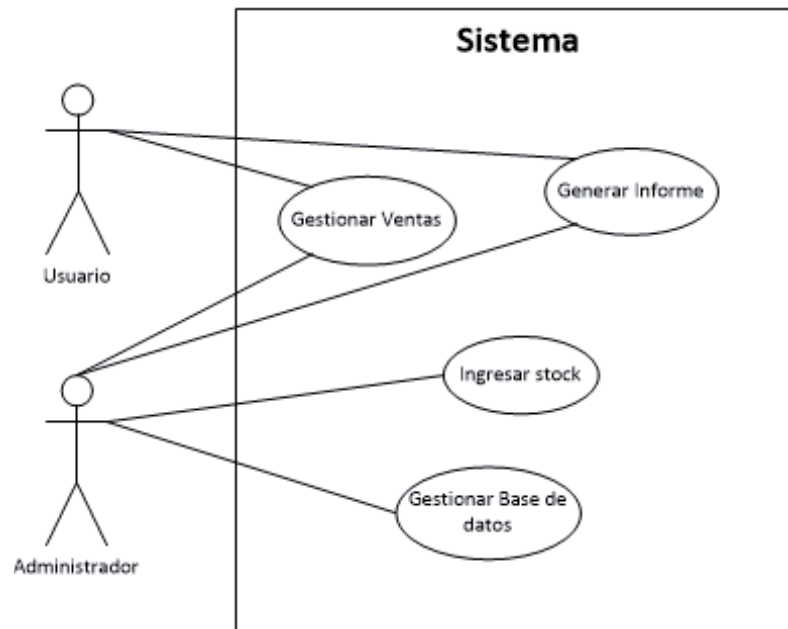


Figura 7.1 Caso de Uso General

Caso de Uso	Realizar Venta
Actor Principal	Administrador Usuario
Participantes e Intereses	Administrador: Desea realizar una venta. Usuario: Desea Realizar una venta.
Precondiciones	1. El usuario debe estar logueado para realizar la venta.
Postcondiciones	Al finalizar la venta se tendrá un listado de los productos y sus respectivos precios más el precio total.
Escenario principal	1. El Usuario ingresa al sistema 2. El Usuario selecciona la opción realizar venta. 3. El Usuario ingresa los productos que se venderán a través de un lector de código de barra.
Extensiones	1.1 El Usuario ingresa incorrectamente su contraseña. 1.1.1 El sistema le indica al usuario que su ingreso no ha sido exitoso. 1.1.2. El sistema manda un mensaje pidiendo que lo intente de nuevo. 3.1 El Usuario ingresa incorrectamente su contraseña. 3.1.1 El sistema le indica al usuario que su ingreso no ha sido exitoso. 3.1.2. El sistema manda un mensaje pidiendo que lo intente de nuevo. 3.2 El usuario realiza correctamente la venta. 3.2.1 Se muestra por pantalla los productos y sus precios más el total.
Requisitos Especiales	No hay requisitos especiales.
Frecuencia de Ocurrencia	Alta

Tabla 7.1 Especificación Caso de Uso Realizar Venta.

Caso de Uso	Registrar compra
Actor Principal	Administrador
Participantes e Intereses	Administrador: Desea registrar compras.
Precondiciones	1. El usuario debe tener privilegios de administrador. 2. El usuario debe haber ingresado en la opción de ingresar stock.
Postcondiciones	Por pantalla se indica que el ingreso de stock fue exitoso.
Escenario principal	1. El Usuario ingresa los códigos de los productos y la cantidad de cada uno. 2. El sistema verifica si los productos existen con la función buscar productos. 3. El sistema calcula los costos totales de cada producto y muestra el registro por pantalla. 4. El usuario confirma el registro. 5. El sistema guarda el registro en la base de datos.
Extensiones	1.1 El usuario ingresa letras en el espacio de código de producto. 1.1.1 El sistema le pide que ingrese un valor numérico por pantalla.

Requisitos Especiales	El paso 2 del escenario principal debe indicar que exista el producto.
Frecuencia de Ocurrencia	Alta

Tabla 7.2 Especificación Caso de Uso Registrar Compra.

7.1 Diagramas de Secuencia

7.1.1 Realizar Venta

Para llevar a cabo la venta de algún producto, el usuario (vendedor) deberá escanear su código de barra, éste se agrega a un listado, se busca el producto, se confirma su existencia, se genera un listado de todos los productos a vender y se muestran los datos al usuario. Luego éste confirma la venta, incluyendo con cuanto dinero le cancelan. El sistema calcula el total de la venta y entrega la información a través de la interfaz. Se actualiza el stock de los productos recién vendidos, se confirma la operación realizada y finalmente se entrega al usuario, el vuelto que deberá entregar al cliente, junto con esto se genera un informe de la venta recién realizada.

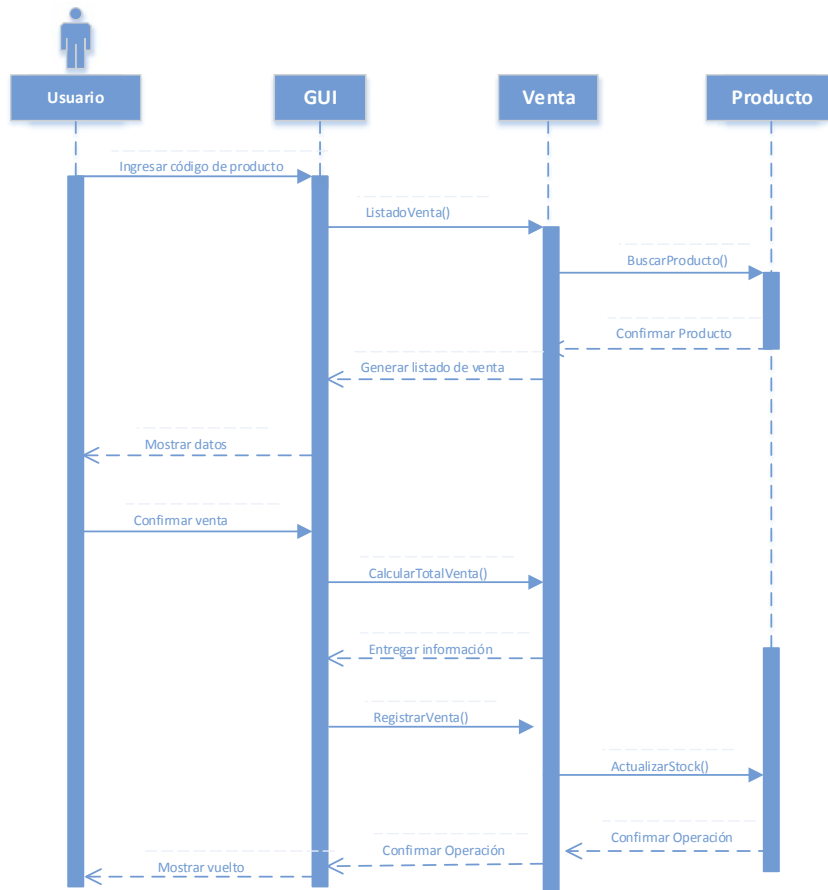


Figura 7.2 Diagrama de Secuencia Realizar Venta.

7.1.2 Registrar Compra

El administrador del minimarket “Futuro”, es el único que puede registrar una compra. Cabe destacar que al registrar una compra, se deberá realizar sólo por un producto a la vez, no se genera un listado, a diferencia de la venta, si no que se registra la compra de un producto y se procede con el registro de otro. Para lo anterior, se tiene que escanear el código de barra del producto e ingresar manualmente la cantidad a comprar de éste. Se registra la compra, la que calcula automáticamente el total de ésta. Se busca que el producto exista y se corrobora, luego se busca el proveedor y se confirma su existencia. Se actualiza el stock del producto comprado, se comprueba la operación. Se entrega la información a través de la interfaz y a su vez ésta se la muestra al administrador, quien visualiza el total de la compra.

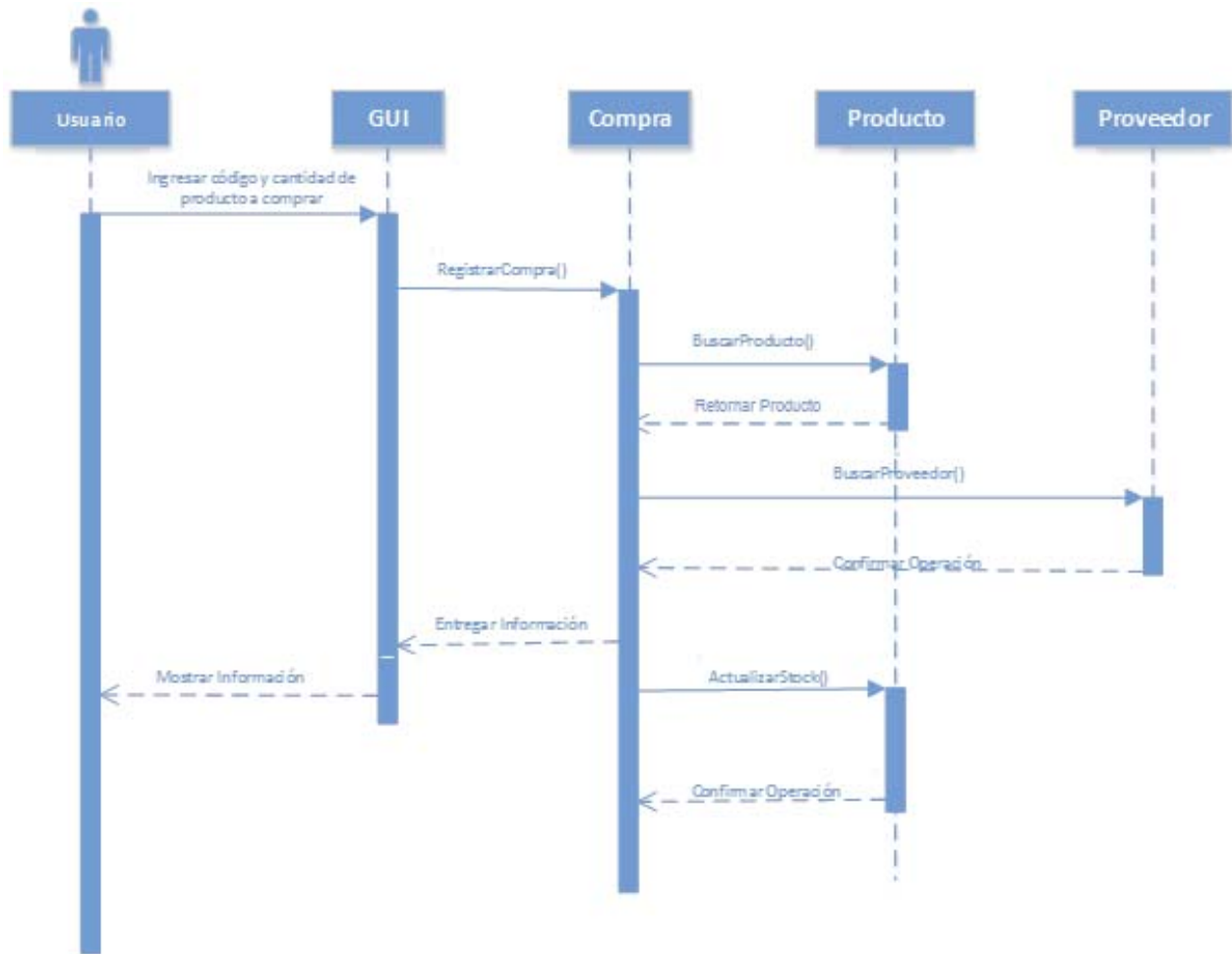


Figura 7.3 Diagrama de Secuencia Registrar Compra.

7.1.3 Crear Producto

Para crear un producto el administrador, deberá ingresar el código de barra del producto, éste se busca para confirmar que no existe. Luego, se ingresan los datos del producto: una breve descripción, costo bruto, márgen, precio de venta y su medida. Se reciben estos datos y se selecciona de una lista el proveedor al que está ligado el producto, se confirma la operación, se selecciona familia a la que pertenece y a su vez se comprueba la operación, posteriormente se selecciona el impuesto que posee y al igual que las anteriores se ratifica la operación realizada. Finalmente se guarda el producto creado y se muestra un mensaje de éxito al administrador.

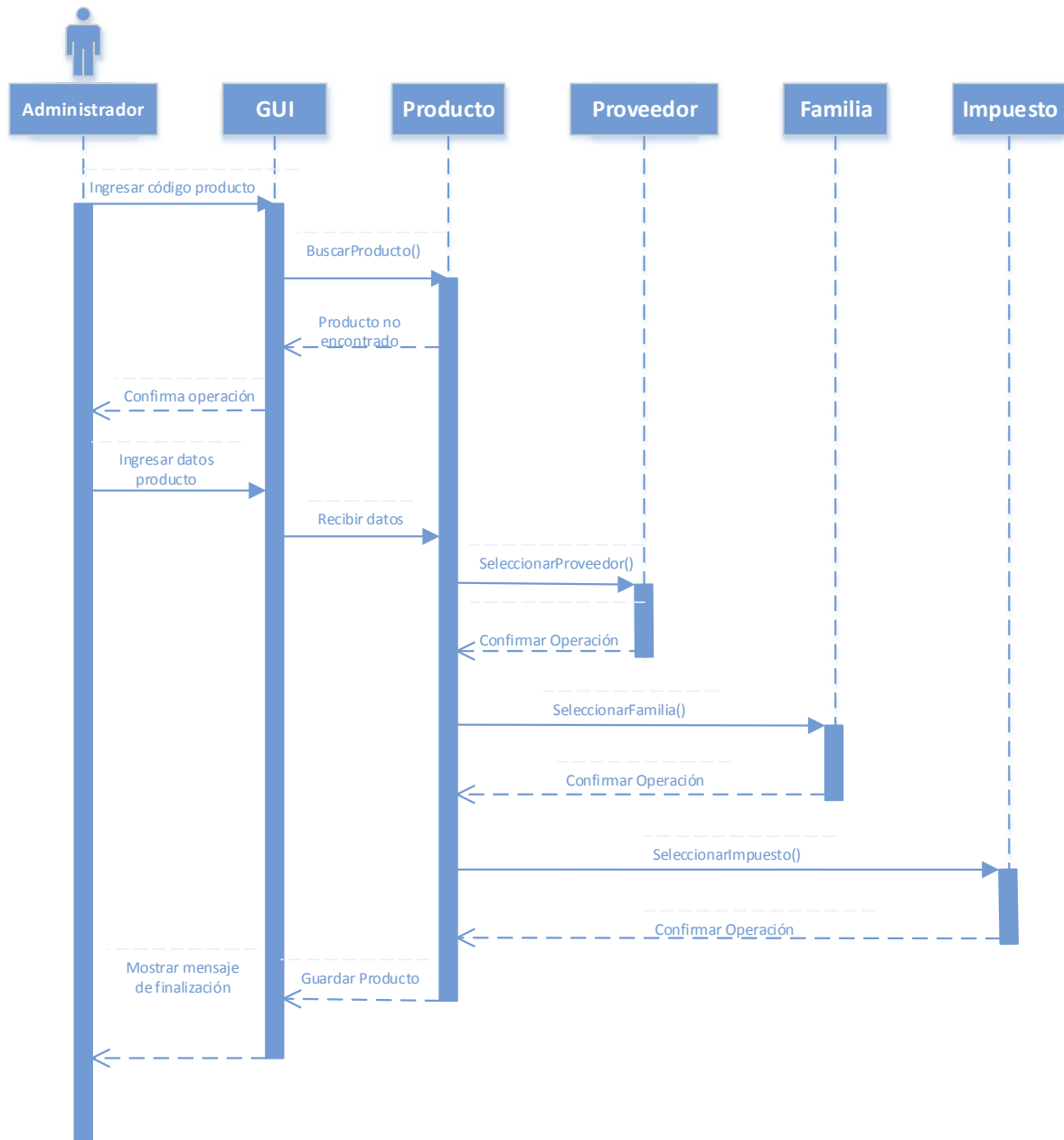


Figura 7.4 Diagrama de Secuencia Crear Producto.

7.2 Diagrama de Clases

Un diagrama de clases, permite mostrar las clases del sistema y sus interrelaciones. Éstos sirven tanto para mostrar lo que el sistema puede hacer (análisis), como para mostrar como puede ser construido (diseño). A continuación, se visualiza el diagrama de clases que representa el sistema a desarrollar, para el minimarket “Futuro”.

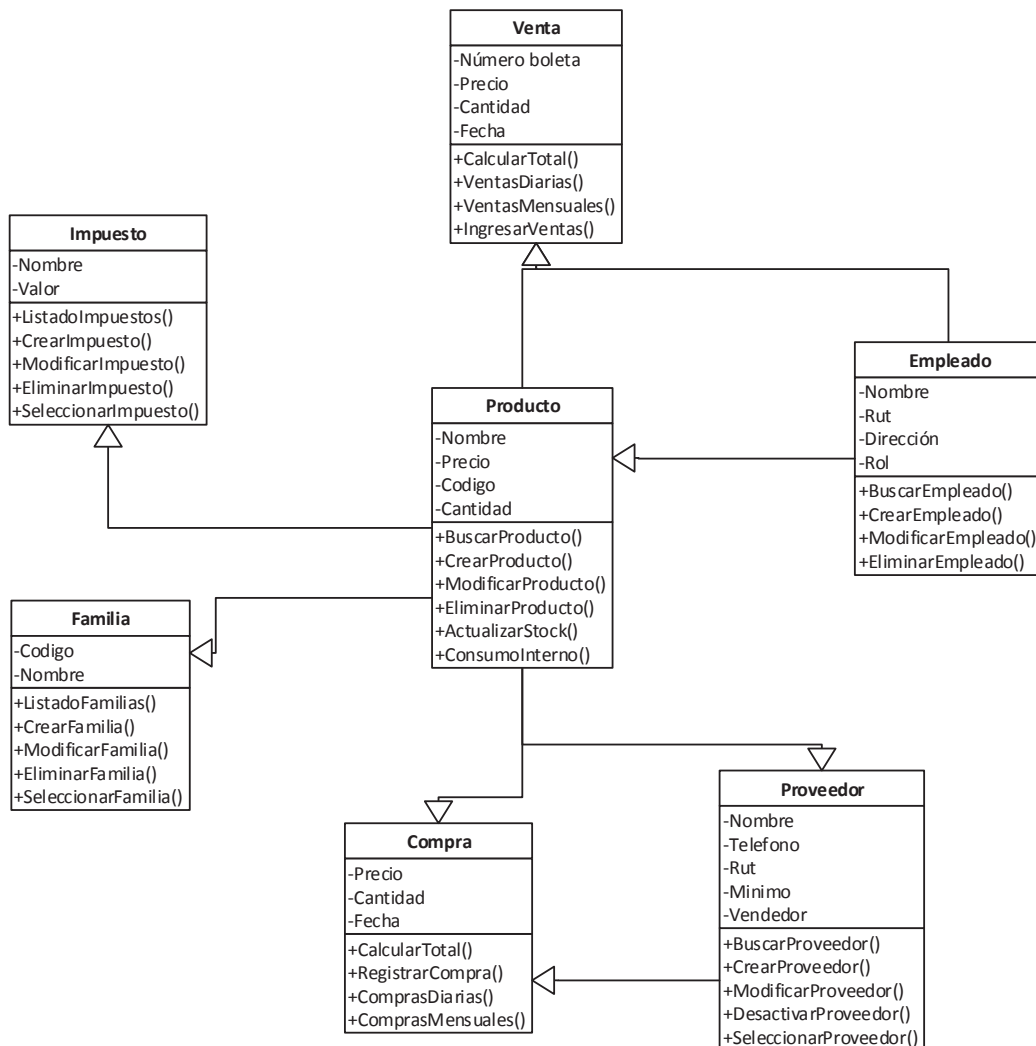


Figura 7.5 Diagrama de Clases.

7.3 Modelo Entidad Relación

Un diagrama o modelo entidad-relación es una herramienta para el modelado de datos, que permite representar las entidades relevantes de un sistema de información, así como sus interrelaciones y propiedades. A continuación, se mostrará el modelo entidad-relación que refleja el sistema del minimarket “Futuro”.

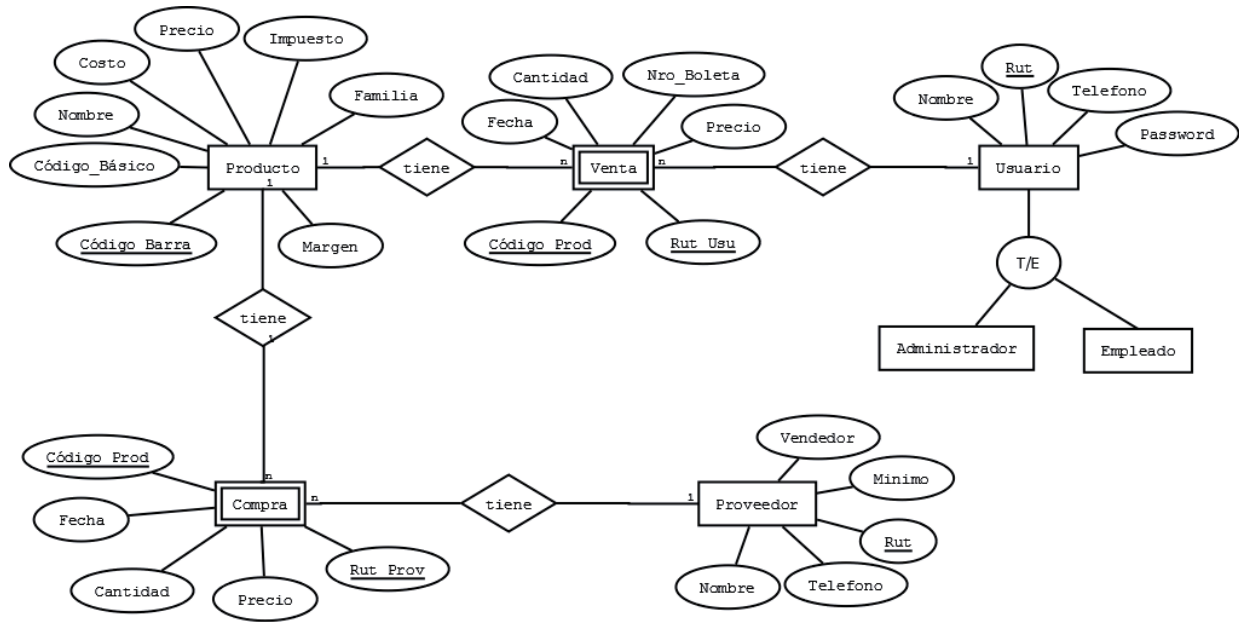


Figura 7.6 Modelo Entidad Relación.

7.4 Modelo Relacional

En este tipo de modelo, todos los datos son almacenados en relaciones y como cada relación es un conjunto de datos, carece de orden en el que éstos se almacenen. Esto tiene la considerable ventaja de que es más fácil de comprender y de utilizar por un usuario no experto. La información puede ser recuperada o almacenada por medio de consultas que ofrecen una amplia flexibilidad y poder para administrar la información. El modelo Relacional a continuación Refleja las relaciones entre los datos del negocio “Futuro”.

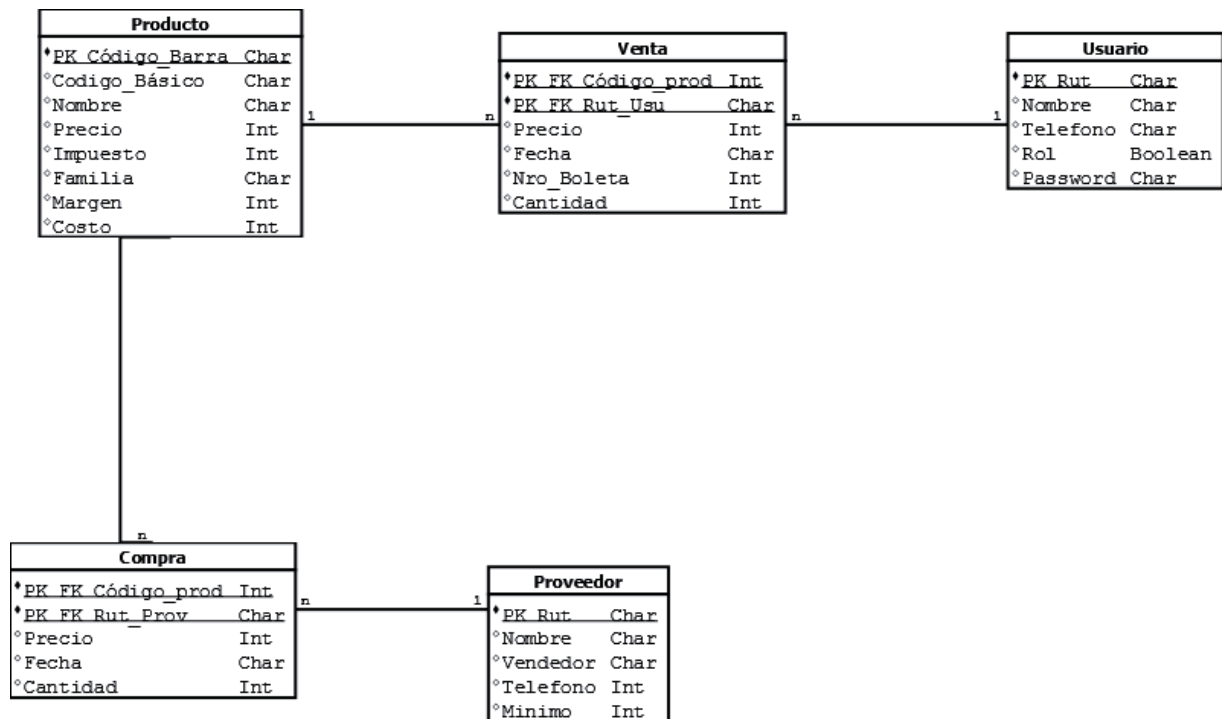


Figura 7.7 Modelo Relacional.

7.5 Interfaces Software Futuro

A continuación se muestra como luce el sistema, el cual fue diseñado con interfaces simples e intuitivas.



Figura 7.8 Interfaz Inicio de sesión.



Figura 7.9 Interfaz Menú Principal.

Futuro Software

Boleta N°:

Operación Tipo Venta

Codigo Producto:

Vendedor: Melisa Astudillo **Fecha:** 08/09/2015

Cod.	Producto	Precio	Cantidad	Sub total

Operación tipo:

Total:

Figura 7.10 Interfaz Operación Venta.

Futuro Software

Registrar Compra

Proveedor:

Código Producto:

Costo Bruto:

Precio Venta: **Margen (%):**

Factura N°:

Cod.	Producto	Costo	Cantidad	Desccto	Subtotal

Total:

Figura 7.11 Interfaz Registrar Compra.

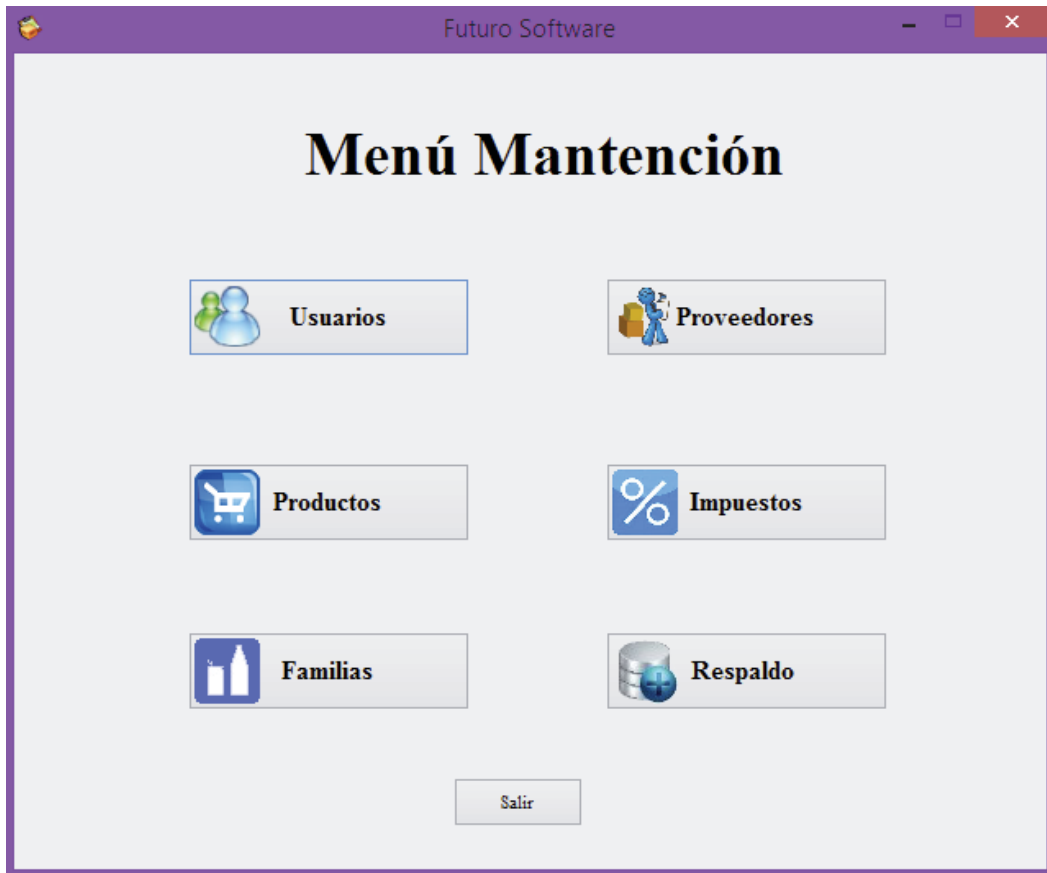


Figura 7.12 Interfaz Mantención de Productos.



Figura 7.13 Interfaz Tipos de Informes.

8 Diseño de Pruebas

En esta etapa se diseñan los distintos escenarios de prueba para cada una de las funciones con el fin de encontrar algún error dentro de ellas.

- El enfoque a utilizar será el de la caja negra debido a que lo importante aquí son los datos que utilizamos y los resultados que estos nos dan, sin importar el proceso por el que pasan.
- La técnica utilizada es la de causa-efecto usando valores límites, permitiendo detectar errores rápidamente.
- Las pruebas fueron llevadas a cabo por los desarrolladores, realizando operaciones en el sistema, como las que veremos a continuación.

1. Función Agregar Producto: Añade un producto a la base de datos. Recibe como parámetro el código de barra, código básico, nombre, familia, impuesto, proveedor, costo, margen y precio.

- ❖ Ingreso de código básico, código de barra valido.
 - ✓ 32432423
 - *ABCDE , abc123, Nulo
- ❖ Ingreso de margen.
 - ✓0, 100
 - *-1 , 101, sadas, Nulo
- ❖ Ingreso de precio, costo bruto.
 - ✓32432423
 - *ABCDE , abc123, Nulo
- ❖ Ingreso de Descripción.
 - ✓asdasda123
 - *Nulo

2. Función Agregar Usuario: Añade un usuario nuevo a la base de datos. Recibe como parámetros el Rut, rol, nombre, teléfono, password.

- ❖ Ingreso de Rut.
 - ✓ 18.964.834-k
 - *18964834k , 18964834-k, Nulo
- ❖ Ingreso de rol.
 - ✓0, 1
 - *-1, 101, sadas, Nulo
- ❖ Ingreso de nombre.
 - ✓Sebastián
 - *abc123, 23213, Nulo
- ❖ Ingreso de Telefono.

- ✓87654321
- * ABC213, abc, Nulo

- ❖ Ingreso de password.
 - ✓abc1234
 - * Nulo

3. Función RealizarVenta: Deja registro de la venta en la Base de datos y disminuye las cantidades de los productos vendidos. Recibe como parámetros el Código del producto, y el tipo de operación.

- ❖ Ingreso de Código.
 - ✓ 012324024246423274
 - * 189\$#6\\483 , Nulo
- ❖ Ingreso de Tipo de operación.
 - ✓Venta
 - * Nulo

4. Función GenerarInformeDiarioConsumoInterno: Genera un documento Excel con los datos de consumo interno. Recibe como parámetros la fecha.

- ❖ Ingreso de Fecha.
 - ✓ 17 de Julio de 2015
 - * Nulo.

9 Conclusión

Para el desarrollo del sistema “Futuro Software” se han utilizado distintas herramientas y tecnologías destacando el uso de Visual Basic .NET como lenguaje de programación, el cual fue de mucha ayuda, ya que permite crear aplicaciones de manera eficaz. Se ha comenzado desarrollando los mantenedores, ya que de esta forma se tienen las funciones básicas más importantes y da paso a crear las funciones principales, que en este caso son las de Venta y Compra. Además, las interfaces diseñadas, son bastante simples e intuitivas, para favorecer su comprensión y manipulación.

La realización de diagramas para visualizar la interacción entre los distintos componentes del sistema, ha resultado una herramienta muy útil y práctica para poder comenzar con la etapa de desarrollo. Los casos de uso y diagramas de secuencia, ayudaron a la visualización del sistema, para hacer más simple su diseño. Los modelos entidad-relación y relacional, ayudaron para la creación de la base de datos, aunque estos necesitaron algunas modificaciones, ya que poseían atributos que no eran necesarios y faltaban otros, los cuales no habían sido considerados previamente.

Para poder avanzar de manera adecuada y llevar a cabo las tareas desarrolladas, se especificaron todas las actividades y sus respectivas fechas de realización, en una Carta Gantt, lo que ayudó al grupo de trabajo a organizarse mejor. Cabe destacar, que el trabajo en conjunto, ayudó a salir del individualismo y aprender a compartir ideas, para asistirse en la solución de obstáculos en el desarrollo del sistema.

Dos semanas antes de la finalización del informe, se ha comenzado a implementar el software en el minimarket “Futuro”. Por el momento, se trata de un período de prueba, el que se estima durará aproximadamente un mes. Hasta ahora todo ha funcionado bien, con los productos que poseen código de barras, los que no, se están registrando manualmente, pero se está trabajando en la creación de códigos para poder escanearlos, e implementarlos pasado el período de prueba. Lo más importante y satisfactorio para los desarrolladores es que el sistema ha sido de mucha ayuda para el cliente, quien está feliz con el trabajo logrado.

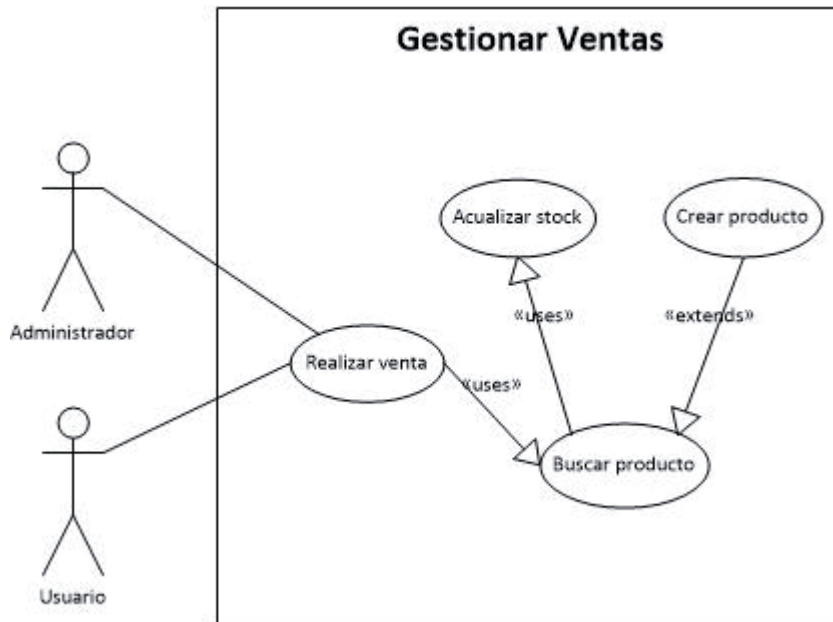
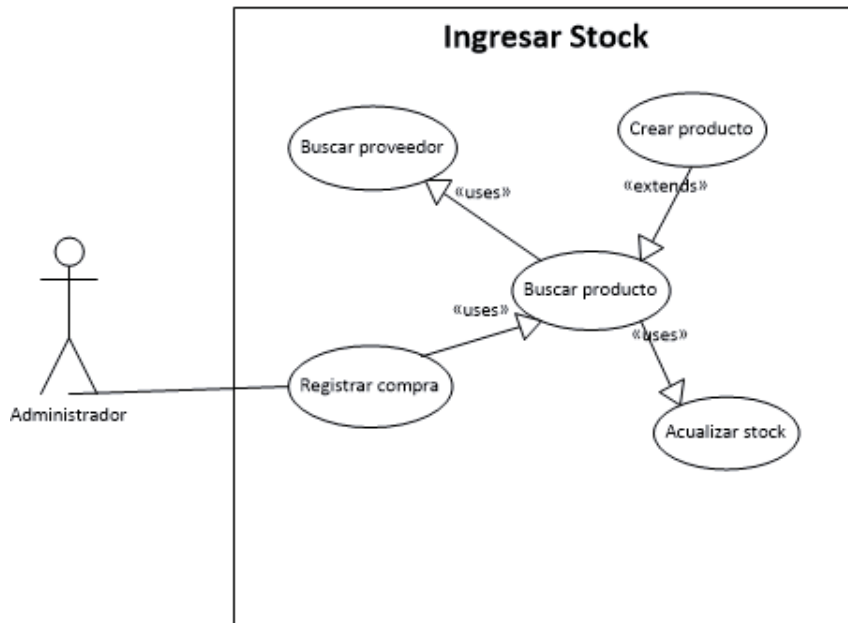
Luego del periodo dedicado al desarrollo e implementación del sistema en cuestión, el grupo de trabajo se siente satisfecho de poder haber cumplido con cada uno de los requerimientos funcionales exigidos por el cliente. Ya que el sistema tiene la particularidad de poder ir mejorando sus características, se pretende, en un futuro, agregar más funcionalidades para que llegue a ser un software genérico y poder comercializarlo.

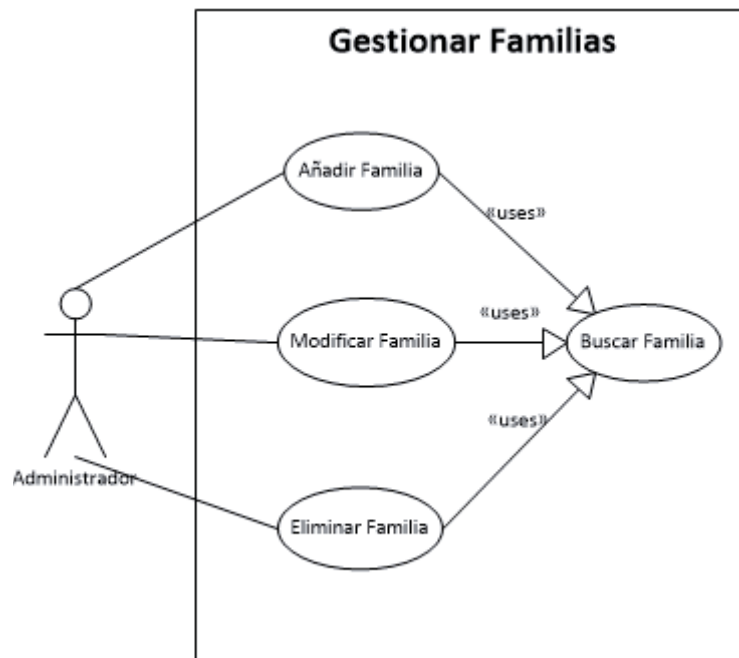
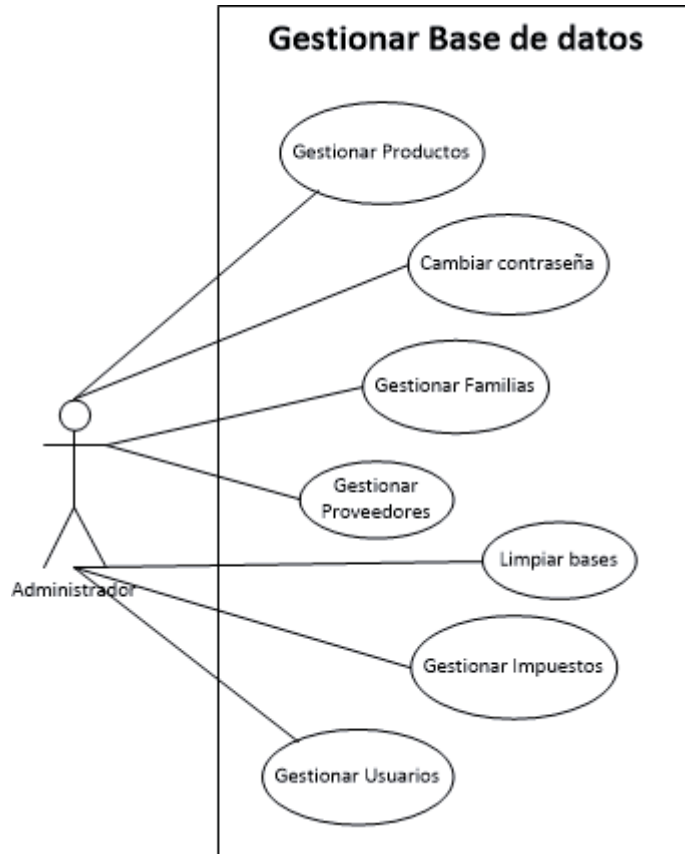
10 Referencias

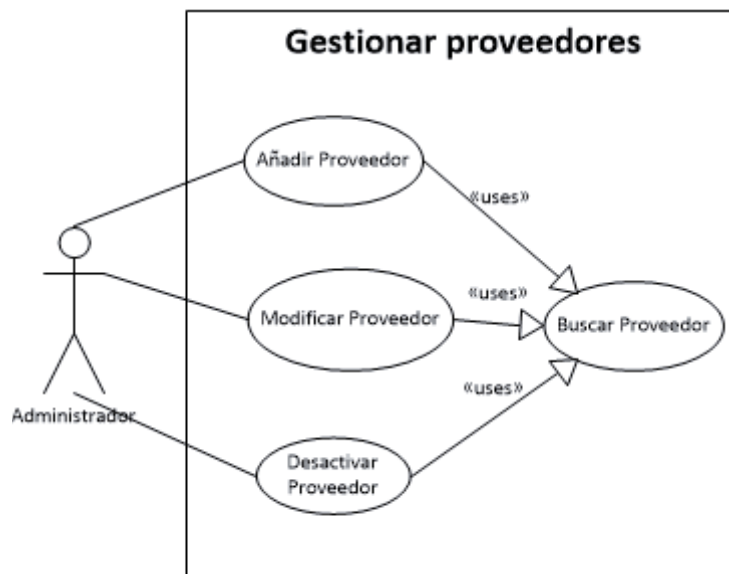
- [1] Ingeniería del Software, Royer Presman, VI edición.
- [2] Aprendiendo UML en 24 horas, Joseph Schmuller, Editorial Prentis Hall.
- [3] UML y Patrones, Craig Larman, Editorial Prentice Hall.

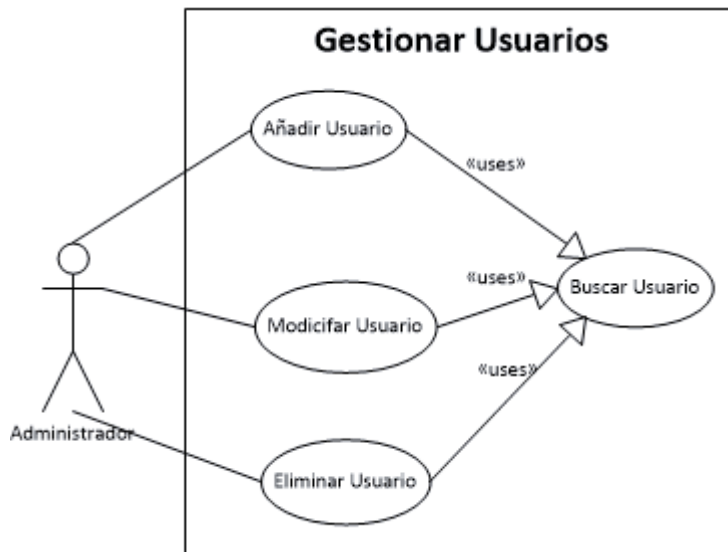
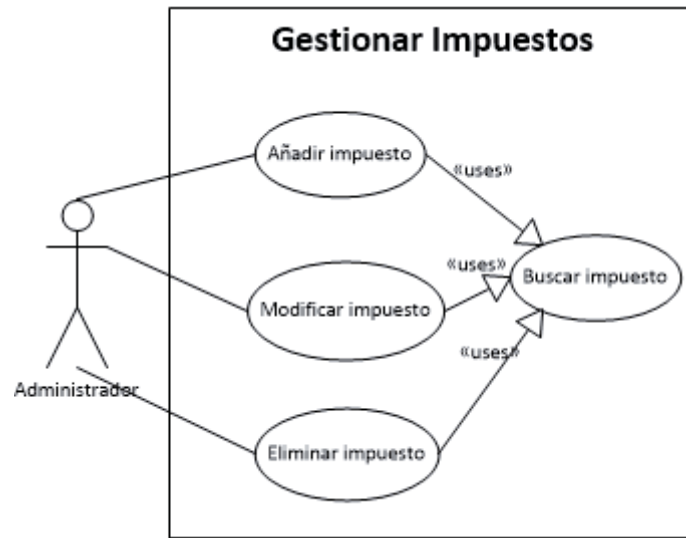
Anexos:

A: Casos de Uso









Caso de Uso	Buscar producto
Actor Principal	Administrador Usuario
Participantes e Intereses	Administrador: Desea realizar una venta, crear un producto, modificar un producto o eliminar un producto. Usuario: Desea realizar una venta.
Precondiciones	1. El usuario debe haber seleccionado la opción realizar venta, ingresar stock, crear producto, modificar producto o eliminar producto. 2. El usuario debe leer el código de barra del producto o ingresar el código del producto.
Postcondiciones	El sistema indicara si existe o no el producto.
Escenario principal	1. El sistema ofrece la opción de crear el producto 2. El sistema llama a la función buscar proveedor en el caso de estar en un ingreso de stock.
Extensiones	1.1 El usuario acepta la opción. 1.1.1 El sistema abre re direcciona a la interfaz de crear producto.
Requisitos Especiales	No hay requisitos especiales.
Frecuencia de Ocurrencia	Alta

Caso de Uso	Actualizar stock
Actor Principal	Administrador Usuario
Participantes e Intereses	Administrador: Desea realizar una venta, crear un producto, modificar un producto o eliminar un producto. Usuario: Desea realizar una venta.
Precondiciones	1. El usuario debe haber realizado correctamente una venta o un ingreso de stock.
Postcondiciones	El sistema configurara el stock disminuyendo o aumentándolo dependiendo de si es un ingreso de stock o una venta.
Escenario principal	No hay escenario principal
Extensiones	No hay Extensiones
Requisitos Especiales	No hay requisitos especiales.
Frecuencia de Ocurrencia	Alta

Caso de Uso	Buscar proveedor
Actor Principal	Administrador
Participantes e Intereses	Administrador: desea gestionar un proveedor, un producto o quiere ingresar stock.
Precondiciones	1. El usuario debió crear un producto, modificarlo o registrar una compra
Postcondiciones	El sistema le detecta si existe o no el proveedor en la base de datos.
Escenario principal	1. El sistema busca al proveedor en la base de datos.
Extensiones	1.1 El sistema no encuentra al proveedor en la base de datos. 1.1.1 El sistema le ofrece al usuario añadir al proveedor en la base de datos.
Requisitos Especiales	No hay requisitos especiales.
Frecuencia de Ocurrencia	Media

Caso de Uso	Gestionar productos.
Actor Principal	Administrador
Participantes e Intereses	Administrador: desea Crear, modificar o eliminar un producto.
Precondiciones	1. El usuario debe tener privilegios de administrador. 2. El usuario debe haber seleccionado la opción de Gestionar base de datos.
Postcondiciones	El sistema lo envía a la interfaz de gestión de productos.
Escenario principal	1. El usuario selecciona la opción Gestionar productos. 2. El sistema abre la interfaz de gestión de productos. 3. El sistema le ofrece crear, modificar o eliminar producto al usuario.
Extensiones	No hay Extensiones.
Requisitos Especiales	No hay requisitos especiales.
Frecuencia de Ocurrencia	Media

Caso de Uso	Gestionar Proveedores.
Actor Principal	Administrador
Participantes e Intereses	Administrador: desea Crear, modificar o desactivar un proveedor.
Precondiciones	1. El usuario debe tener privilegios de administrador. 2. El usuario debe haber seleccionado la opción de Gestionar base de datos.
Postcondiciones	El sistema lo envía a la interfaz de gestión de Proveedores.

Escenario principal	1. El usuario selecciona la opción Gestionar proveedores. 2. El sistema abre la interfaz de gestión de proveedores. 3. El sistema le ofrece crear, modificar o desactivar proveedor.
Extensiones	No hay Extensiones.
Requisitos Especiales	No hay requisitos especiales.
Frecuencia de Ocurrencia	Media

Caso de Uso	Gestionar Familias.
Actor Principal	Administrador
Participantes e Intereses	Administrador: desea Crear, modificar o eliminar una familia.
Precondiciones	1. El usuario debe tener privilegios de administrador. 2. El usuario debe haber seleccionado la opción de Gestionar base de datos.
Postcondiciones	El sistema lo envía a la interfaz de gestión de familias.
Escenario principal	1. El usuario selecciona la opción Gestionar familias. 2. El sistema abre la interfaz de gestión de familias. 3. El sistema le ofrece crear, modificar o eliminar familia.
Extensiones	No hay extensiones.
Requisitos Especiales	No hay requisitos especiales.
Frecuencia de Ocurrencia	Media

Caso de Uso	Gestionar Impuestos.
Actor Principal	Administrador
Participantes e Intereses	Administrador: desea Crear, modificar o eliminar un impuesto.
Precondiciones	1. El usuario debe tener privilegios de administrador. 2. El usuario debe haber seleccionado la opción de Gestionar base de datos.
Postcondiciones	El sistema lo envía a la interfaz de gestión de impuestos.
Escenario principal	1. El usuario selecciona la opción Gestionar impuestos. 2. El sistema abre la interfaz de gestión de impuestos. 3. El sistema le ofrece crear, modificar o eliminar impuesto.
Extensiones	No hay extensiones.
Requisitos Especiales	No hay requisitos especiales.
Frecuencia de Ocurrencia	Media

Caso de Uso	Gestionar Usuarios.
Actor Principal	Administrador
Participantes e Intereses	Administrador: desea Crear, modificar o eliminar un impuesto.
Precondiciones	1. El usuario debe tener privilegios de administrador. 2. El usuario debe haber seleccionado la opción de Gestionar base de datos.
Postcondiciones	El sistema lo envía a la interfaz de gestión de usuarios.
Escenario principal	1. El usuario selecciona la opción Gestionar usuarios. 2. El sistema abre la interfaz de gestión de usuarios. 3. El sistema le ofrece crear, modificar o eliminar usuarios.
Extensiones	No hay extensiones.
Requisitos Especiales	No hay requisitos especiales.
Frecuencia de Ocurrencia	Media

Caso de Uso	Modificar producto.
Actor Principal	Administrador
Participantes e Intereses	Administrador: desea modificar un producto.
Precondiciones	1. El usuario debe tener privilegios de administrador. 2. El usuario debe haber seleccionado la opción modificar producto.
Postcondiciones	Por pantalla se indica que la modificación fue exitosa.
Escenario principal	1. El usuario ingresa el código del producto. 2. El sistema verifica el producto. 3. El usuario selecciona la familia, impuesto y proveedor del producto. 4. El usuario ingresa precio de venta, margen de ganancia, precio de compra y stock. 5. El usuario finaliza la modificación presionando el botón de guardar.
Extensiones	1.1 El usuario ingresa letras o caracteres especiales. 1.1.1 El sistema le pide al usuario intentarlo de nuevo y que use solo caracteres numéricos. 2.1 El Sistema llama a la función buscar producto para ver que no exista. 3.1 El sistema ofrece crear una familia, impuesto o proveedor nuevo. 4.1 El usuario ingresa letras o caracteres especiales. 4.1.1 El sistema le pide al usuario intentarlo de nuevo y que use solo caracteres numéricos. 4.2 El usuario ingresa margen diferente al precio de venta. 4.2.1 El sistema reajusta el precio de venta según el margen ingresado.
Requisitos Especiales	El paso 2 del escenario principal debe indicar que exista el producto.

Frecuencia de Ocurrencia	Alta
Caso de Uso	Eliminar producto.
Actor Principal	Administrador
Participantes e Intereses	Administrador: desea Eliminar un producto.
Precondiciones	1. El usuario debe tener privilegios de administrador. 2. El usuario debe haber seleccionado la opción Eliminar producto.
Postcondiciones	Por pantalla se indica que la creación fue exitosa.
Escenario principal	1. El usuario ingresa el código del producto. 2. El sistema verifica el producto. 3. El usuario selecciona la opción eliminar 4. el sistema elimina el producto.
Extensiones	2.1 El Sistema llama a la función buscar producto para ver si existe.
Requisitos Especiales	El paso 2 del escenario principal debe indicar que exista el producto.
Frecuencia de Ocurrencia	Media

Caso de Uso	Crear Proveedor.
Actor Principal	Administrador
Participantes e Intereses	Administrador: desea Crear un proveedor.
Precondiciones	1. El usuario debe tener privilegios de administrador. 2. El usuario debe haber seleccionado la opción crear proveedor.
Postcondiciones	Por pantalla se indica que la creación fue exitosa.
Escenario principal	1. El usuario ingresa el nombre del proveedor. 2. El sistema verifica el proveedor. 3. El usuario presiona la opción guardar. 4. El sistema guarda al nuevo proveedor en la base de datos.
Extensiones	2.1 El sistema llama a la función buscar proveedor.
Requisitos Especiales	El paso 2 del escenario principal debe indicar que no exista el proveedor.
Frecuencia de Ocurrencia	Alta

Caso de Uso	Modificar Proveedor.
Actor Principal	Administrador
Participantes e Intereses	Administrador: desea Modificar un proveedor.

Precondiciones	1. El usuario debe tener privilegios de administrador. 2. El usuario debe haber seleccionado la opción Modificar proveedor.
Postcondiciones	Por pantalla se indica que la modificación fue exitosa.
Escenario principal	1. El usuario ingresa el nombre del proveedor. 2. El sistema verifica el proveedor. 3. El usuario ingresa nuevos datos de proveedor. 4. El usuario presiona guardar. 5. El sistema modifica el proveedor de todos los productos de ese proveedor. 6. El sistema guarda al nuevo proveedor.
Extensiones	2.1 El sistema llama a la función buscar proveedor. 3.1 El usuario ingresa números en el nombre de proveedor. 3.1.1 El sistema le pide al usuario intentarlo de nuevo y que use solo letras.
Requisitos Especiales	El paso 2 del escenario principal debe indicar que exista el proveedor.
Frecuencia de Ocurrencia	Baja

Caso de Uso	Desactivar Proveedor.
Actor Principal	Administrador
Participantes e Intereses	Administrador: desea desactivar un proveedor.
Precondiciones	1. El usuario debe tener privilegios de administrador. 2. El usuario debe haber seleccionado la opción desactivar proveedor.
Postcondiciones	Por pantalla se indica que la desactivación fue exitosa.
Escenario principal	1. El usuario ingresa el nombre del proveedor. 2. El sistema verifica el proveedor. 3. El usuario selecciona la opción desactivar. 4. El sistema cambia el estado del proveedor a desactivado. 5. El sistema muestra por pantalla que se ha desactivado correctamente.
Extensiones	1.1 El usuario ingresa números en el nombre de proveedor. 1.1.1 El sistema le pide al usuario intentarlo de nuevo y que use solo letras. 2.1 El sistema llama a la función Buscar proveedor.
Requisitos Especiales	El paso 2 del escenario principal debe indicar que exista el proveedor.
Frecuencia de Ocurrencia	Baja

Caso de Uso	Crear Familia.
Actor Principal	Administrador
Participantes e Intereses	Administrador: desea crear una familia.
Precondiciones	1. El usuario debe tener privilegios de administrador. 2. El usuario debe haber seleccionado la opción desactivar proveedor.
Postcondiciones	Por pantalla se indica que la creación fue exitosa.
Escenario principal	1. El usuario ingresa el nombre de la familia. 2. El sistema verifica la familia. 3. El usuario selecciona la opción guardar. 4. El sistema guarda la nueva familia en la base de datos. 5. El sistema muestra por pantalla que se ha creado correctamente.
Extensiones	1.1 El usuario ingresa números en el nombre de proveedor. 1.1.1 El sistema le pide al usuario intentarlo de nuevo y que use solo letras. 2.1 El sistema llama a la función Buscar Familia.
Requisitos Especiales	El paso 2 del escenario principal debe indicar que no exista la familia.
Frecuencia de Ocurrencia	Baja

Caso de Uso	Crear Producto
Actor Principal	Administrador Usuario
Participantes e Intereses	Administrador: Desea crear un producto, realizar una venta o ingresar stock. Usuario: Desea realizar una venta.
Precondiciones	1. El usuario debe tener privilegios de administrador. 2. El usuario debe haber ingresado en la opción de gestionar productos, realizar venta o ingresar stock, e ingresar un producto desconocido para la base de datos.
Postcondiciones	Por pantalla se indica que el producto fue creado exitosamente.

Escenario principal	<ol style="list-style-type: none"> 1. El Usuario ingresa código de barra del producto. 2. El sistema verifica el producto. 3. El Usuario ingresa y selecciona los datos del producto que desea crear. 4. El usuario selecciona la opción guardar. 5. El sistema guarda el producto en la base de datos.
Extensiones	<ol style="list-style-type: none"> 1.1 El usuario ingresa letras en el código del producto. <ol style="list-style-type: none"> 1.1.1 El sistema le pide que lo reintente y que ingrese solo caracteres numéricos. 2.1 El sistema llama a la función Buscar Producto. 3.1 El usuario ingresa letras en los precios, margen o stock. <ol style="list-style-type: none"> 3.1.1 El sistema le pide que lo reintente y que ingrese solo caracteres numéricos. 3.2 El usuario quiere crear una nueva familia, impuesto o proveedor. <ol style="list-style-type: none"> 3.2.1 El sistema lo envía a la interfaz correspondiente para la creación de una familia, impuesto o proveedor nuevo.
Requisitos Especiales	<p>En el caso de ser una venta el usuario debe verificar que tiene permisos de administrador.</p> <p>El paso 2 del escenario principal debe indicar que no exista el producto.</p>
Frecuencia de Ocurrencia	Alta

Caso de Uso	Modificar Familia.
Actor Principal	Administrador
Participantes e Intereses	Administrador: desea Modificar una familia.
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe tener privilegios de administrador. 2. El usuario debe haber seleccionado la opción modificar familia.
Postcondiciones	Por pantalla se indica que la modificación fue exitosa.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario ingresa el nombre de la familia. 2. El sistema verifica la familia. 3. El usuario ingresa nuevos datos de familia. 4. El usuario selecciona la opción guardar. 5. El sistema modifica el nombre de la familia en todos los productos asociados. 6. El sistema guarda los cambios de la familia en la base de datos. 5. El sistema muestra por pantalla que se ha creado correctamente.
Extensiones	<ol style="list-style-type: none"> 1.1 El usuario ingresa números en el nombre de familia. <ol style="list-style-type: none"> 1.1.1 El sistema le pide al usuario intentarlo de nuevo y que use solo letras. 2.1 El sistema llama a la función Buscar Familia. 3.1 El usuario ingresa números en el nombre de familia. <ol style="list-style-type: none"> 3.1.1 El sistema le pide al usuario intentarlo de nuevo y que use solo letras.

Requisitos Especiales	El paso 2 del escenario principal debe indicar que exista la familia.
Frecuencia de Ocurrencia	Baja

Caso de Uso	Eliminar Familia.
Actor Principal	Administrador
Participantes e Intereses	Administrador: desea Eliminar una familia.
Precondiciones	1. El usuario debe tener privilegios de administrador. 2. El usuario debe haber seleccionado la opción eliminar familia.
Postcondiciones	Por pantalla se indica que la eliminación fue exitosa.
Escenario principal	1. El usuario ingresa el nombre de la familia. 2. El sistema verifica la familia. 3. El usuario confirma la eliminación. 4. El sistema elimina la familia de la base de datos. 5. El sistema muestra por pantalla que se ha eliminado correctamente.
Extensiones	1.1 El usuario ingresa números en el nombre de familia. 1.1.1 El sistema le pide al usuario intentarlo de nuevo y que use solo letras. 2.1 El sistema llama a la función Buscar Familia.
Requisitos Especiales	El paso 2 del escenario principal debe indicar que exista la familia. La familia no debe tener productos asociados.
Frecuencia de Ocurrencia	Baja

Caso de Uso	Crear Impuesto.
Actor Principal	Administrador
Participantes e Intereses	Administrador: desea Crear un impuesto.
Precondiciones	1. El usuario debe tener privilegios de administrador. 2. El usuario debe haber seleccionado la opción Crear impuesto.
Postcondiciones	Por pantalla se indica que la creación fue exitosa.

Escenario principal	<ol style="list-style-type: none"> 1. El usuario ingresa el nombre del impuesto. 2. El sistema verifica el nombre del impuesto ingresado. 3. El usuario ingresa el valor del impuesto. 4. El usuario selecciona la opción guardar. 4. El sistema guarda el nuevo impuesto en la base de datos. 5. El sistema muestra por pantalla que se ha creado exitosamente el impuesto.
Extensiones	<ol style="list-style-type: none"> 1.1 El usuario ingresa números en el nombre de familia. <ol style="list-style-type: none"> 1.1.1 El sistema le pide al usuario intentarlo de nuevo y que use solo letras. 2.1 El sistema llama a la función Buscar Impuesto. 3.1 El usuario ingresa letras en el valor del impuesto. <ol style="list-style-type: none"> 3.1.1 El sistema le pide al usuario intentarlo de nuevo y que ingrese un valor numérico.
Requisitos Especiales	El paso 2 del escenario principal debe indicar que no exista el impuesto.
Frecuencia de Ocurrencia	Baja

Caso de Uso	Modificar Impuesto.
Actor Principal	Administrador
Participantes e Intereses	Administrador: desea Modificar un impuesto.
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe tener privilegios de administrador. 2. El usuario debe haber seleccionado la opción Crear impuesto.
Postcondiciones	Por pantalla se indica que la creación fue exitosa.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario ingresa el nombre del impuesto. 2. El sistema verifica el nombre del impuesto ingresado. 3. El usuario ingresa nuevos datos para el impuesto. 4. El usuario selecciona la opción guardar. 4. El sistema guarda los cambios en la base de datos. 5. El sistema muestra por pantalla que se ha modificado exitosamente el impuesto.
Extensiones	<ol style="list-style-type: none"> 1.1 El usuario ingresa números en el nombre de familia. <ol style="list-style-type: none"> 1.1.1 El sistema le pide al usuario intentarlo de nuevo y que use solo letras. 2.1 El sistema llama a la función Buscar Impuesto. 3.1 El usuario ingresa letras en el valor del impuesto. <ol style="list-style-type: none"> 3.1.1 El sistema le pide al usuario intentarlo de nuevo y que ingrese un valor numérico.
Requisitos Especiales	El paso 2 del escenario principal debe indicar que exista el impuesto.

Frecuencia de Ocurrencia	Baja
--------------------------	------

Caso de Uso	Eliminar Impuesto.
Actor Principal	Administrador
Participantes e Intereses	Administrador: desea Eliminar un impuesto.
Precondiciones	1. El usuario debe tener privilegios de administrador. 2. El usuario debe haber seleccionado la opción Eliminar impuesto.
Postcondiciones	Por pantalla se indica que la eliminación fue exitosa.
Escenario principal	1. El usuario ingresa el nombre del impuesto. 2. El sistema verifica el nombre del impuesto ingresado. 3. El usuario confirma la eliminación. 4. El sistema guarda los cambios en la base de datos. 5. El sistema muestra por pantalla que se ha eliminado exitosamente el impuesto.
Extensiones	1.1 El usuario ingresa números en el nombre de familia. 1.1.1 El sistema le pide al usuario intentarlo de nuevo y que use solo letras. 2.1 El sistema llama a la función Buscar Impuesto. 3.1 El usuario ingresa letras en el valor del impuesto. 3.1.1 El sistema le pide al usuario intentarlo de nuevo y que ingrese un valor numérico.
Requisitos Especiales	El paso 2 del escenario principal debe indicar que exista el impuesto. El impuesto no debe estar asociado a ningún producto
Frecuencia de Ocurrencia	Baja

Caso de Uso	Crear Usuario.
Actor Principal	Administrador
Participantes e Intereses	Administrador: desea Crear un usuario.
Precondiciones	1. El usuario debe tener privilegios de administrador. 2. El usuario debe haber seleccionado la opción Crear usuario.
Postcondiciones	Por pantalla se indica que la creación fue exitosa.

Escenario principal	<ol style="list-style-type: none"> 1. El Administrador ingresa el Rut del usuario. 2. El sistema verifica el Rut del usuario ingresado. 3. El Administrador ingresa datos del usuario. 4. El Administrador selecciona la opción guardar. 5. El sistema guarda al nuevo usuario en la base de datos. 6. El sistema muestra por pantalla que se ha creado exitosamente el usuario.
Extensiones	<ol style="list-style-type: none"> 1.1 El <u>Administrador</u> ingresa letras en el Rut del usuario. <ol style="list-style-type: none"> 1.1.1 El sistema le pide al usuario intentarlo de nuevo y que use solo números. 2.1 El sistema llama a la función Buscar usuario. 3.1 El Administrador ingresa caracteres no válidos. <ol style="list-style-type: none"> 3.1.1 El sistema le pide al usuario intentarlo de nuevo y que ingrese del tipo que se indica.
Requisitos Especiales	El paso 2 del escenario principal debe indicar que no exista el usuario.
Frecuencia de Ocurrencia	Baja

Caso de Uso	Modificar Usuario.
Actor Principal	Administrador
Participantes e Intereses	Administrador: desea Modificar un usuario.
Precondiciones	<ol style="list-style-type: none"> 1. El usuario debe tener privilegios de administrador. 2. El usuario debe haber seleccionado la opción Modificar usuario.
Postcondiciones	Por pantalla se indica que la modificación fue exitosa.
Escenario principal	<ol style="list-style-type: none"> 1. El Administrador ingresa el Rut del usuario. 2. El sistema verifica el Rut del usuario ingresado. 3. El Administrador cambia datos del usuario. 4. El usuario selecciona la opción guardar. 5. El sistema guarda al nuevo usuario en la base de datos. 6. El sistema muestra por pantalla que se ha creado exitosamente el impuesto.
Extensiones	<ol style="list-style-type: none"> 1.1 El Administrador ingresa letras en el Rut del usuario. <ol style="list-style-type: none"> 1.1.1 El sistema le pide al usuario intentarlo de nuevo y que use solo números. 2.1 El sistema llama a la función Buscar usuario. 3.1 El Administrador ingresa caracteres no válidos. <ol style="list-style-type: none"> 3.1.1 El sistema le pide al usuario intentarlo de nuevo y que ingrese del tipo que se indica.
Requisitos Especiales	El paso 2 del escenario principal debe indicar que exista el usuario.
Frecuencia de Ocurrencia	Baja

Caso de Uso	Eliminar Usuario.
Actor Principal	Administrador
Participantes e Intereses	Administrador: desea Eliminar un usuario.
Precondiciones	1. El administrador debe haber seleccionado la opción Eliminar usuario.
Postcondiciones	Por pantalla se indica que la eliminación fue exitosa.
Escenario principal	<ol style="list-style-type: none"> 1. El Administrador selecciona el nombre del usuario, de una lista. 2. El Administrador selecciona la opción eliminar. 3. El sistema elimina al usuario de la base de datos. 4. El sistema muestra por pantalla que se ha eliminado exitosamente al usuario.
Extensiones	No hay Extensiones
Requisitos Especiales	No hay requisitos especiales
Frecuencia de Ocurrencia	Baja