



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

---

**SISTEMA DE CONTROL Y ADMINISTRACIÓN DE  
PRODUCTOS Y SERVICIOS PARA LA EMPRESA CARPAS  
VIÑA EVENTOS**

**ISAÍAS RICARDO BARAHONA CÁRCAMO  
GONZALO ELÍAS SARIEGO GALLARDO**

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

**Diciembre 2008**



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

---

**SISTEMA DE CONTROL Y ADMINISTRACIÓN DE  
PRODUCTOS Y SERVICIOS PARA LA EMPRESA CARPAS  
VIÑA EVENTOS**

**ISAÍAS RICARDO BARAHONA CÁRCAMO  
GONZALO ELÍAS SARIEGO GALLARDO**

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

**Diciembre 2008**

*... A mis padres Nancy y Ricardo. Ellos  
hicieron posible este logro tan importante en  
mi vida.  
A Vanessa, por ser un pilar fundamental que  
me incentivó y apoyó en todo momento. A  
toda mi familia, Muchas Gracias.*

***Isaías Ricardo Barahona Cárcamo.***

*...A mis padres Luis y Lilian, por su esfuerzo  
y cariño.  
A mi esposa Claudia, por su constante  
apoyo, amor y comprensión.*

***Gonzalo Elías Sariago Gallardo.***

## **RESUMEN**

El presente documento corresponde al trabajo final para optar al título de Ingeniero de ejecución en Informática, llamado “Sistema de Control y Administración de Productos y Servicios” para la empresa Carpas Viña, el cual tiene como objetivo dar solución a los problemas de manejo de stock y ventas dentro de la empresa.

La realización de este sistema surge por la necesidad de la empresa de tener un mayor control de los recursos existentes en ella, es por ello que este proyecto estará orientado principalmente hacia ese tema.

El manejo de la información en cuanto a ventas y stock se realiza de manera escrita lo cual genera problemas de seguridad y dificultades a la hora de entregar informes. Este sistema de información pretende solucionar estos problemas además de permitir agilizar los tiempos de respuesta para sus clientes, lo que se traduce en un mejor servicio de atención hacia estos.

**PALABRAS CLAVE:** Sistema de Información, manejo de stock, ventas, presupuesto, evento.

## **ABSTRACT**

The following document corresponds to the final Project with the core principle of obtaining the degree of “Ingeniero de Ejecución en Informática”, called “Sistema de Control y Administración de Productos y Servicios”, which aims to provide solutions to problems of management of products and sales within the company.

The realization of this system arises from the need for the company to have more control of existing resources into it, which is why this project will be geared mainly towards this issue.

The handling of information in terms of sales and stock is done in written form which generates security problems and difficulties in delivering reports. This information system aims to solve these problems in addition to allowing expedite response times for customers, which translates into better service to them.

**KEY WORDS:** Information System, stock management, sales, budget, event.

# Índice

<b>GLOSARIO .....</b>	<b>V</b>
<b>ABREVIATURAS O SIGLAS .....</b>	<b>VII</b>
<b>ÍNDICE DE ILUSTRACIONES .....</b>	<b>VIII</b>
<b>ÍNDICE DE TABLAS.....</b>	<b>IX</b>
<b>INTRODUCCIÓN .....</b>	<b>XI</b>
<b>1 CAPÍTULO 1: ESTUDIO DE LA SITUACIÓN ACTUAL.....</b>	<b>1</b>
<b>1.1 DESCRIPCIÓN DE LA ORGANIZACIÓN ANALIZADA .....</b>	<b>1</b>
1.1.1 DESCRIPCIÓN GENERAL .....	1
1.1.2 CLIENTES Y USUARIOS .....	2
1.1.3 FORMA ACTUAL DE TRABAJO.....	3
1.1.4 REALIDAD TECNOLÓGICA DE LA EMPRESA.....	3
1.1.5 SITUACIONES QUE GENERAN PROBLEMÁTICA .....	4
<b>1.2 OBJETIVOS.....</b>	<b>5</b>
1.2.1 OBJETIVO GENERAL .....	5
1.2.2 OBJETIVOS ESPECÍFICOS.....	5
<b>2 CAPÍTULO 2: SOLUCIÓN PROPUESTA.....</b>	<b>6</b>
<b>2.1 FORMA FUTURA DE TRABAJO.....</b>	<b>6</b>
<b>2.2 SOLUCIÓN TECNOLÓGICA PROPUESTA.....</b>	<b>7</b>
<b>2.3 ARQUITECTURA PLANTEADA COMO SOLUCIÓN .....</b>	<b>7</b>
2.3.1 ARQUITECTURA LÓGICA .....	7
2.3.2 ARQUITECTURA FÍSICA .....	8
2.3.3 EXPLICACIÓN DE LAS CAPAS Y COMPONENTE DE LA ARQUITECTURA.....	8
2.3.4 MÓDULOS DEL SISTEMA.....	9
<b>2.4 HERRAMIENTAS A UTILIZAR .....</b>	<b>10</b>
2.4.1 HERRAMIENTA DE MODELADO: UML .....	10
2.4.2 BASE DE DATOS: SQL SERVER .....	11
2.4.3 MICROSOFT VISIO .....	12
2.4.4 MICROSOFT PROJECT .....	13
<b>2.5 PARADIGMAS DE DESARROLLO DE SOFTWARE .....</b>	<b>13</b>
2.5.1 MODELO EN CASCADA .....	13
2.5.2 DESARROLLO EVOLUTIVO.....	14
2.5.3 DESARROLLO EN ESPIRAL .....	17

2.5.4	PROCESO UNIFICADO DE DESARROLLO (UP)	18
2.5.5	SELECCIÓN DEL PARADIGMA A UTILIZAR	19
<b>2.6</b>	<b>METODOLOGÍAS DE DESARROLLO DE SOFTWARE</b>	<b>22</b>
2.6.1	METODOLOGÍA ESTRUCTURADA	22
2.6.2	METODOLOGÍA ORIENTADA A OBJETOS	22
2.6.3	SELECCIÓN DE METODOLOGÍA A UTILIZAR	23
<b>2.7</b>	<b>ESTUDIO DE FACTIBILIDAD</b>	<b>23</b>
2.7.1	FACTIBILIDAD TÉCNICA	24
2.7.2	FACTIBILIDAD OPERACIONAL	26
2.7.3	FACTIBILIDAD ECONÓMICA	26
2.7.4	FACTIBILIDAD LEGAL	27
2.7.5	CONCLUSIÓN DE ESTUDIO DE FACTIBILIDAD	27
<b>2.8</b>	<b>ANÁLISIS DE RIESGO</b>	<b>28</b>
2.8.1	IDENTIFICACIÓN DEL RIESGO	29
2.8.2	GESTIÓN DEL RIESGO	30
<b>3</b>	<b>CAPÍTULO 3: DESARROLLO FASE INICIO</b>	<b>31</b>
<b>3.1</b>	<b>PRIMERA ITERACIÓN</b>	<b>31</b>
3.1.1	CLASIFICACIÓN DE REQUERIMIENTOS	31
3.1.1.1	Requerimientos Funcionales	31
3.1.1.2	Requerimientos No Funcionales	32
3.1.2	ANÁLISIS	32
3.1.2.1	Actores	32
3.1.2.2	Diagrama General de Casos de Uso	33
<b>4</b>	<b>CAPÍTULO 4: DESARROLLO FASE ELABORACIÓN</b>	<b>34</b>
<b>4.1</b>	<b>PRIMERA ITERACIÓN: SESIÓN</b>	<b>34</b>
4.1.1	ANÁLISIS	34
4.1.1.1	Diagramas de Casos de Uso	34
4.1.2	DISEÑO	34
4.1.2.1	Especificación de Casos de Uso	34
4.1.2.2	Diagramas de Secuencia	35
<b>4.2</b>	<b>SEGUNDA ITERACIÓN: CONFIGURACIÓN DEL SISTEMA</b>	<b>35</b>
4.2.1	ANÁLISIS	35
4.2.1.1	Diagramas de Casos de Uso	35
4.2.2	DISEÑO	36
4.2.2.1	Especificación de Casos de Uso	36
4.2.2.2	Diagramas de Secuencia	38
<b>4.3</b>	<b>TERCERA ITERACIÓN: PRESUPUESTOS Y EVENTOS</b>	<b>40</b>
4.3.1	ANÁLISIS	40
4.3.1.1	Diagramas de Casos de Uso	40
4.3.2	DISEÑO	41
4.3.2.1	Especificación de Casos de Uso	41

4.3.2.2 Diagramas de Secuencia .....	46
<b>4.4 CUARTA ITERACIÓN: VENTAS .....</b>	<b>50</b>
4.4.1 ANÁLISIS .....	50
4.4.1.1 Diagramas de Casos de Uso .....	50
4.4.2 DISEÑO .....	51
4.4.2.1 Especificación de Casos de Uso.....	51
4.4.2.2 Diagramas de Secuencia .....	54
<b>5 CAPÍTULO 5: DESARROLLO FASE CONSTRUCCIÓN .....</b>	<b>57</b>
<b>5.1 DIAGRAMA DE CLASES .....</b>	<b>57</b>
<b>5.2 MODELO RELACIONAL.....</b>	<b>58</b>
<b>5.3 PRIMERA ITERACIÓN: SESIÓN .....</b>	<b>60</b>
5.3.1 IMPLEMENTACIÓN.....	60
5.3.1.1 Interfaces .....	60
5.3.2 PRUEBAS.....	62
5.3.3 RESULTADOS DE PRUEBAS .....	64
<b>5.4 SEGUNDA ITERACIÓN: CONFIGURACIÓN DEL SISTEMA .....</b>	<b>65</b>
5.4.1 IMPLEMENTACIÓN.....	65
5.4.1.1 Interfaces .....	65
5.4.2 PRUEBAS.....	75
5.4.2.1 Resultados de Pruebas.....	77
<b>5.5 TERCERA ITERACIÓN: PRESUPUESTOS Y EVENTOS.....</b>	<b>78</b>
5.5.1 IMPLEMENTACIÓN.....	78
5.5.1.1 Interfaces .....	78
5.5.2 PRUEBAS.....	81
5.5.2.1 Resultados de Pruebas.....	83
<b>5.6 CUARTA ITERACIÓN: VENTAS .....</b>	<b>84</b>
5.6.1 IMPLEMENTACIÓN.....	84
5.6.1.1 Interfaces .....	84
5.6.2 PRUEBAS.....	86
5.6.2.1 Resultados de Pruebas.....	87
<b>6 CAPÍTULO 6: DESARROLLO FASE TRANSICIÓN .....</b>	<b>88</b>
<b>6.1 PRIMERA ITERACIÓN.....</b>	<b>88</b>
6.1.1 USABILIDAD DEL SISTEMA .....	88
6.1.1.1 Métodos de Análisis de Usabilidad.....	88
6.1.2 PRUEBAS DE ACEPTACIÓN.....	91
<b>CONCLUSIONES .....</b>	<b>92</b>
<b>BIBLIOGRAFÍA .....</b>	<b>94</b>

# Glosario

- Administrador: Actor y Usuario Principal del Sistema.
- Análisis de requerimientos: Proceso de comprensión del problema en cuestión, y las necesidades del Cliente.
- Escenario: Secuencia específica de acciones e interacciones entre los actores y el sistema en desarrollo.
- Evento: Negocio o Servicio entregado por la Empresa Carpas Viña a sus clientes.
- Excel: Software Computacional que permite la gestión de información mediante planillas y hojas de cálculo.
- Factura: Recibo donde se detallan los productos vendidos o servicios prestados con sus respectivos precios, el cual es presentado al cliente como justificante del pago realizado.
- Gerente: Actor y usuario del Sistema, encargado de la elaboración y aprobación de Presupuestos.
- Gestión: Conjunto de acciones llevadas a cabo para solucionar un asunto, o administrar una entidad.
- Guía de despacho: Formulario en el cual se detalla los recursos a utilizar en cada evento.
- Loguear: Identificar a un usuario válido en el sistema.
- Presupuesto: Cálculo anticipado de los ingresos y costos de un negocio, o servicio particular.
- Proyecto: Conjunto de acciones y su planificación que son objeto de este documento.
- Recursos: Medios de los que se dispone para satisfacer una necesidad, como ejemplo de recursos disponibles en Carpas Viña tenemos; mesas, sillas, cenefas, carpetas, manteles, lazos, pistas de baile, etc.
- Sistema: Software o componente computacional.
- Tipos de usuario (Interacción base de datos):



- Usuario Final Ocasional: En nuestro sistema es el gerente de Carpas Viña quien requiere información de manera esporádica, sus necesidades de información varía constantemente ya que requiere generar informes de ventas en distintos periodos de tiempo.
- Usuario Final Simple: En nuestro sistema es el administrador de Carpas Viña, el cual está facultado para realizar consultas y actualizaciones constantes a la base de datos para manejar de manera eficiente los recursos de la empresa, y prestar un servicio eficiente a sus diversos clientes.

# Abreviaturas o Siglas

- SQL : Structured Query Language (Lenguaje de Consulta Estructurado).
- UML : Unified Modeling Language (Lenguaje Unificado de Modelado).
- SGBD : Sistema de Gestión de Base de Datos.
- DDL : Data Definition Language (Lenguaje de Definición de Datos).
- DML : Data Manipulation Language (Lenguaje de Manipulación de Datos).
- C : Lenguaje de Programación Estructurado.
- BD : Base de Datos.
- UP : Unified Process (Proceso Unificado).
- Mhz : Megahertz (Megahercio).
- Mb : Megabyte.
- Gb : Gigabyte.
- Ghz : Gigahertz (Gigahercio).
- UF : Unidad de Fomento.
- HW : Hardware.
- SW : Software.
- PUCV : Pontificia Universidad Católica de Valparaíso.
- UI : User Interface (Interfaz de Usuario).
- ISO : International Organization for Standardization (Organización Internacional de Estandarización).

# Índice de Ilustraciones

Ilustración 2.1 Arquitectura Lógica .....	8
Ilustración 2.2 Arquitectura Física .....	8
Ilustración 2.3 Módulos del Sistema .....	9
Ilustración 2.4 Desarrollo Evolutivo .....	16
Ilustración 6.1 Diagrama General de Casos de Uso .....	33
Ilustración 7.1 Caso de Uso: Loguearse .....	34
Ilustración 7.2 Diagrama de Secuencia: Caso de Uso Loguearse .....	35
Ilustración 7.3 Caso de Uso: Administrar Usuarios .....	35
Ilustración 7.4: Diagrama de Secuencia para Caso de Uso Ingresar Usuario .....	38
Ilustración 7.5: Diagrama de Secuencia para Caso de Uso Modificar Usuario .....	39
Ilustración 7.6: Diagrama de Secuencia para caso de Uso Eliminar Usuario .....	39
Ilustración 7.7: Caso de Uso Administrar Presupuestos .....	40
Ilustración 7.8: Caso de Uso Administrar Eventos .....	40
Ilustración 7.9: Diagrama de Secuencia para Caso de Uso Generar Presupuesto .....	46
Ilustración 7.10: Diagrama de Secuencia para Caso de Uso Modificar Presupuesto .....	47
Ilustración 7.11: Diagrama de Secuencia para Caso de Uso Confirmar Presupuesto .....	48
Ilustración 7.12: Diagrama de Secuencia para Caso de Uso Imprimir Factura .....	49
Ilustración 7.13: Caso de Uso Administrar Ventas .....	50
Ilustración 7.14: Diagrama de Secuencia para Caso de Uso Consultar Ventas .....	54
Ilustración 7.15: Diagrama de Secuencia para Caso de Uso Generar Reporte Ventas .....	55
Ilustración 7.16: Diagrama de Secuencia para Caso de Uso Comparar Período .....	56
Ilustración 8.1: Diagrama de Clases .....	57
Ilustración 8.2: Modelo Relacional Parte 1 .....	58
Ilustración 8.3: Modelo Relacional Parte 2 .....	59
Ilustración 8.4: Interfaz Inicio de Sesión .....	61
Ilustración 8.5: Interfaz Menú Principal .....	61
Ilustración 8.6: Interfaz Administrar Productos .....	65
Ilustración 8.7: Interfaz Ingresar Productos .....	66
Ilustración 8.8: Interfaz Modificar Producto .....	66
Ilustración 8.9: Interfaz Ingresar Cliente .....	67
Ilustración 8.10: Interfaz Modificar Cliente (Búsqueda) .....	67
Ilustración 8.11: Interfaz Modificar Cliente .....	68
Ilustración 8.12: Interfaz Administrar Usuarios .....	69
Ilustración 8.13: Interfaz Ingresar Usuarios .....	69
Ilustración 8.14: Interfaz Modificar Usuario (Búsqueda) .....	70
Ilustración 8.15: Interfaz Modificar Usuario .....	70
Ilustración 8.16: Interfaz Eliminar Usuario (Búsqueda) .....	71
Ilustración 8.17: Interfaz Eliminar Usuario .....	71
Ilustración 8.18: Interfaz Administrar Proveedores .....	72
Ilustración 8.19: Interfaz Ingresar Proveedor .....	72
Ilustración 8.20: Interfaz Modificar Proveedor (Búsqueda) .....	73
Ilustración 8.21: Interfaz Modificar Proveedor .....	73

Ilustración 8.22: Interfaz Eliminar Proveedor (Búsqueda) .....	74
Ilustración 8.23: Interfaz Eliminar Proveedor .....	74
Ilustración 8.24: Interfaz Administrar Presupuestos .....	78
Ilustración 8.25: Interfaz Presupuesto .....	78
Ilustración 8.26: Interfaz Modificar Presupuesto (Búsqueda) .....	79
Ilustración 8.27: Interfaz Modificar Presupuesto .....	79
Ilustración 8.28: Interfaz Administrar Eventos (Búsqueda) .....	80
Ilustración 8.29: Administrar Eventos .....	80
Ilustración 8.30: Administrar Ventas .....	84
Ilustración 8.31: Interfaz Consultar Ventas .....	84
Ilustración 8.32: Interfaz Comparar Períodos de Ventas .....	85
Ilustración 8.33: Interfaz Consultar Artículo Más Vendido .....	85

## Índice de Tablas

Tabla 2-1 Virtudes de Paradigmas .....	21
Tabla 2-2 Identificación del Riesgo .....	29
Tabla 2-3 Clasificación de Riesgos .....	29
Tabla 2-4 Gestión del Riesgo .....	30
Tabla 3-1 Actores del Sistema .....	32
Tabla 4-1 Caso de Uso: Loguearse .....	34
Tabla 4-2 Caso de Uso: Ingresar Usuario .....	36
Tabla 4-3 Caso de Uso Modificar Usuario .....	37
Tabla 4-4 Caso de Uso: Eliminar Usuario .....	38
Tabla 4-5 Caso de Uso Generar Presupuesto .....	41
Tabla 4-6: Caso de Uso Modificar Presupuesto .....	42
Tabla 4-7: Caso de Uso Confirmar Presupuesto .....	43
Tabla 4-8 Caso de Uso Modificar Evento .....	44
Tabla 4-9: Caso de Uso Imprimir Factura .....	45
Tabla 4-10: Caso de Uso Consultar Ventas .....	51
Tabla 4-11: Caso de Uso Generar Reporte Ventas .....	52
Tabla 4-12: Caso de Uso Comparar Períodos .....	52
Tabla 4-13: Caso de Uso Consultar Más Vendido .....	53
Tabla 5-1 Caso de Prueba Login Usuario 1 .....	63
Tabla 5-2: Caso de Prueba Login Usuario 2 .....	63
Tabla 5-3: Caso de Prueba Login Usuario 3 .....	63
Tabla 5-4: Caso de Prueba Recuperar Contraseña 1 .....	63
Tabla 5-5: Caso de Prueba Recuperar Contraseña 2 .....	64
Tabla 5-6: Resultados de Pruebas Iteración Sesión .....	64
Tabla 5-7: Caso de Prueba Ingresar USuario 1 .....	75
Tabla 5-8: Caso de Prueba Ingresar Usuario 2 .....	75
Tabla 5-9: Caso de Prueba Ingresae Usuario 3 .....	75
Tabla 5-10: Caso de Prueba Modificar Usuario 1 .....	76
Tabla 5-11: Caso de Prueba Modificar Usuario 2 .....	76

Tabla 5-12: Caso de Prueba Eliminar Usuario 1 .....	76
Tabla 5-13: Caso de Prueba Eliminar Usuario 2 .....	76
Tabla 5-14: Resultado de Pruebas Iteración Configuración del Sistema .....	77
Tabla 5-15: Caso de Prueba Ingresar Presupuesto 1 .....	81
Tabla 5-16: Caso de Prueba Ingresar Presupuesto 2 .....	81
Tabla 5-17: Caso de Prueba Ingresar Presupuesto 3 .....	82
Tabla 5-18: Caso de Prueba Modificar Presupuesto 1 .....	82
Tabla 5-19: Caso de Prueba Modificar Presupuesto 2 .....	82
Tabla 5-20: Resultados Pruebas Iteración Presupuestos y Eventos .....	83
Tabla 5-21: Caso de Prueba Consultar Ventas 1 .....	86
Tabla 5-22: Caso de Prueba Consultar Ventas 2 .....	86
Tabla 5-23: Caso de Prueba Consultar Más Vendido 1 .....	86
Tabla 5-24: Caso de Prueba Consultar Más Vendido 2 .....	87
Tabla 5-25: Resultados Pruebas Iteración Ventas .....	87

# Introducción

En la actualidad, para toda organización que desee crecer y expandir sus límites, es de vital importancia mantener niveles de excelencia en todo ámbito; de forma que se pueda cumplir con los estándares de competitividad que la impulsen en su mercado objetivo. La información es uno de los principales recursos para la mayoría de las organizaciones ya que cimienta un pilar de soporte para todos los demás recursos existentes dentro de la propia organización. Por esto, es muy recomendado contar con las mejores herramientas de gestión y manejo para dicha información relevante. Los sistemas de información, a su vez, han sido de gran ayuda para que pequeñas y medianas empresas (e incluso las más desarrolladas) mejoren considerablemente la gestión de su información permitiéndoles de este modo, obtener un progreso y mejoramiento en todo nivel.

En el presente informe se da a conocer el análisis, diseño e implementación de los requerimientos del proyecto en investigación el cual se realizará para la Empresa Carpas Viña. El documento muestra de manera detallada el análisis realizado y las características que el sistema tendrá. Además, se identifican los diversos escenarios y los diferentes tipos de usuario que tendrán interacción con el sistema, junto a con las distintas funciones que conlleva cada interacción.

La realización de este Sistema surge por la necesidad de la empresa de tener un mayor control de los recursos existentes en ella, es por ello que el enfoque está orientado principalmente a este tema.

En el capítulo 1 de este informe se encuentra una descripción general de la organización analizada y la forma en la que trabaja en la actualidad, también se detallan los problemas y dificultades que acarrea esta forma de trabajo. Por otra parte, en este primer capítulo se especifican también los objetivos y la arquitectura planteada como solución, además de la forma en cómo se espera que la organización trabaje en el futuro. En el capítulo 2 se define la planificación del proyecto, es decir, se muestran en detalle las herramientas, el paradigma y la metodología de desarrollo escogidos. También en este capítulo se describe el análisis de riesgo realizado como también el estudio de factibilidad. En el capítulo 3 se detalla el desarrollo del proyecto con respecto al análisis y diseño. Se definen los requerimientos funcionales y no funcionales, los actores y sus funciones (casos de uso), se presentan los distintos modelos del sistema, se muestran las interfaces y se describe el plan de pruebas. Por último, en el capítulo 4 se exponen las conclusiones, anexos y la bibliografía de apoyo utilizada.

# Capítulo 1: Estudio de la Situación Actual

## 1.1 Descripción de la Organización Analizada

### 1.1.1 Descripción General

La empresa objeto de estudio en este proyecto es Carpas Viña. Esta empresa se dedica principalmente al rubro de arriendos para la realización de eventos. Carpas Viña otorga un servicio integrado para el cliente, debido a que entrega tanto el servicio de instalación de las carpas, como también el arriendo de alfombras, escenarios, pistas de baile, mesas, sillas, mantelerías, entre otras, hasta la calefacción del lugar. Para la realización de estas actividades la empresa cuenta con un diverso y capacitado equipo de trabajo, el cual está encabezado por su dueño el señor Jorge Lobos Olivares; el responsable del área administrativa es el señor Marcos Lobos Lavín. Debido a que esta empresa tiene el carácter de familiar todos los departamentos comúnmente conocidos como Finanzas, Marketing, Operaciones, Recursos Humanos van en responsabilidad directa de estas dos personas mencionadas anteriormente. Es importante señalar que la empresa es apoyada por un grupo de personas que se desempeña en las áreas de costura, lavandería, electricidad y por supuesto a los actores en terreno. Para el traslado de materiales se cuenta con un moderno conjunto de vehículos especialmente diseñados para este tipo de tareas. Cabe destacar que esta empresa brinda un servicio personalizado y único. Es por eso que cuenta con la mayor preferencia a nivel regional.

Actualmente en la empresa se trabaja con Excel para almacenar la información relevante como presupuestos, control de facturas y gestión de clientes.

Para el manejo de la salida y entrada de los materiales la empresa trabaja con sistemas escritos. El problema está en que cuando un cliente solicita un servicio, el vendedor necesita saber la disponibilidad de sus recursos en forma rápida y eficiente cosa que con los sistemas escritos actuales que se manejan no se logra. Además se puede agregar que como no existe un control eficaz en la carga y descarga de los materiales, se produce una considerable pérdida de material, para lo cual se necesita un compromiso real de parte de los administradores para que el sistema que se cree pueda ayudar de una forma efectiva y concreta a la organización.

### 1.1.2 Clientes y Usuarios

La cartera de clientes de la empresa abarca distintas áreas de la sociedad, tanto instituciones públicas como privadas, como también clientes particulares.

Algunos de sus principales clientes son:

- Pontificia Universidad Católica de Valparaíso
- Universidad Técnica Federico Santa María
- Universidad Andrés Bello
- Universidad de Valparaíso
- Universidad de Viña del Mar
- Universidad del Mar
- Universidad Marítima de Chile
- Universidad de Los Lagos
- DuocUC
- Inacap
- Colegio Mckay
- Colegio Alemán
- Colegio José Cortes Brown
- Valparaíso Sporting Club
- I. Municipalidad de Viña del Mar
- I. Municipalidad de Valparaíso
- I. Municipalidad de Concón
- I. Municipalidad de La Calera
- I. Municipalidad de Puchuncaví
- Hotelera Las Encinas
- Hotelera Santa Ana
- Club de Yates Higuierillas
- Empresa Portuaria de Valparaíso
- TPS Valparaíso
- Valparaíso Terminal de Pasajeros (VTP)
- Embonor
- El Mercurio de Valparaíso
- Esva
- Aguas del Valle
- Clínica Reñaca
- CSAV
- SAAM
- VTR
- Empresas Carozzi
- Caja de Compensación 18 de Septiembre
- Caja de Compensación Los Andes
- París
- Falabella
- Hospital de Quilpue
- Escuela Naval
- Presidencia de la República de Chile
- Armada de Chile
- Comandancia en Jefe de la Armada
- ENAP
- ENJOY

Los usuarios de este sistema serán principalmente el Gerente (dueño de la empresa) quien utilizará el sistema con el fin de obtener información de las ventas realizadas y el administrador quien será la persona que tendrá una mayor interacción con el sistema; gestionando los clientes, productos, proveedores y ventas, además de manejar el control de bodega.



### **1.1.3 Forma Actual de Trabajo**

Hoy en día en la empresa se trabaja de la siguiente manera: la mayoría de los clientes de la empresa se contacta con ella a través de la vía telefónica. Cuando el encargado recibe un llamado de un cliente solicitando un presupuesto para la realización de un evento, toma sus datos personales y la solicitud del cliente apuntándolo de forma manual en un cuaderno que está dedicado especialmente para este uso. Luego, se procede a realizar un presupuesto para el cliente en una planilla Excel diseñada especialmente para este fin, este presupuesto es enviado al cliente y luego es impreso y archivado.

A continuación se espera la aprobación por parte del cliente, una vez aceptado el presupuesto por parte del cliente se procede a anotar el nombre del cliente, el lugar del evento, la fecha, los materiales a utilizar en forma general y el monto total del evento, esto se apunta en un cuaderno distinto al anterior dedicado exclusivamente para los eventos que se realizarán.

En este último cuaderno los eventos se van tarjando de acuerdo al estado en que se encuentren, estos estados son: Instalado (que se tarja con una línea de color negro), retirado (que se tarja con una línea de color azul) y cancelado (que se tarja con una línea roja).

Llegado el día del evento se genera una guía de despacho con el detalle de los materiales que salen de bodega hacia el evento, luego estos materiales se cargan y se despachan.

Por lo general un evento puede durar más de un día, hay veces que retorna solo parte del material despachado para la realización de un evento, quedando el resto en el lugar del evento para seguir siendo utilizado, en este caso se procede a anotar en una hoja los materiales que han retornado y adjuntarla al cuaderno donde se encuentra el detalle del evento, hasta que se retira por completo los materiales del lugar del evento y las cantidades especificadas en la hoja se suman a las que retornan finalmente para poder realizar las comparaciones del material despachado con el retornado.

### **1.1.4 Realidad Tecnológica de la Empresa**

Actualmente en la empresa no se cuenta con ningún tipo de sistema computacional que permita manejar los datos de sus productos, ventas y clientes. La administración de estos datos es realizada en forma manual utilizando cuadernos y archivadores específicos para cada tarea, por lo cual las respuestas a inquietudes son dificultosas y lentas y hay poca seguridad con respecto a estos datos. En la empresa cuentan con un par de computadores de escritorio que poseen acceso a Internet los cuales sólo son utilizados para confeccionar los presupuestos solicitados por los clientes y para su posterior envío, y además se lleva un registro de las ventas que se distribuyen por meses con el fin de poder obtener estadísticas de las ventas. Tanto la confección de presupuestos y el registro de las ventas son realizados en Microsoft Excel. En cuanto a las ventas sólo es registrado el nombre del cliente y el monto total.

### 1.1.5 Situaciones que Generan Problemática

En esta organización se pudo detectar las siguientes falencias o problemas:

- No se logra un adecuado control de los recursos con los que se cuenta en bodega. Este problema acarrea la situación de que ante una eventual cotización no se sabe con precisión los productos que pueden ser ofrecidos a sus clientes.
- No se logra un adecuado control de los productos que salen y entran en bodega. Como todo se trabaja con papeleo es fácil de que se produzcan pérdidas de los productos.
- Administración manual de los datos. Luego de confeccionar un presupuesto a un cliente este es impreso para luego ser archivado según la fecha del evento y las distintas gestiones que se deben realizar con estos se realizan en forma manual.
- Búsqueda manual de los datos. En cuanto a la organización de los eventos, cuando un presupuesto es aceptado, se ingresa en un cuaderno destinado para los eventos que se tienen que realizar, este cuaderno está dividido por semanas y en la fecha correspondiente se escribe el evento a realizarse y el número de presupuesto asociado a este. Cuando llega el día del evento se busca por medio del número de presupuesto los artículos que deben llevarse para la realización del evento.
- Lentitud en la obtención de reportes. Debido al manejo en forma manual de los datos se torna dificultosa y lenta la obtención de reportes asociados a las ventas.
- Poca seguridad de los datos. Debido al manejo de forma manual de los datos existe poca seguridad de los datos.

## **1.2 OBJETIVOS**

### **1.2.1 Objetivo General**

- Desarrollar un sistema computacional que permita optimizar los procesos de ventas y de gestión de productos y clientes, logrando una integración general con el resto de los procesos de negocio involucrados.

### **1.2.2 Objetivos Específicos**

- Entregar respuesta rápida y eficiente a la disponibilidad de recursos involucrados en el proceso de ventas
- Mejorar el control de los materiales que se manejan en bodega el proceso que conlleva la realización de eventos.
- Facilitar la toma de decisiones de la empresa a través del sistema que brindará un mayor conocimiento de las ventas que se realicen en determinados períodos.
- Ampliar el mercado objetivo de la empresa a través de un nuevo canal de distribución como será la creación de una página Web.

# Capítulo 2: Solución Propuesta

## 2.1 Forma Futura de Trabajo

En el futuro al utilizar el sistema que se está diseñando, la empresa trabajará de la siguiente manera:

Cuando un cliente solicite un presupuesto a la empresa lo podrá hacer mediante la página Web de Carpas Viña, de forma telefónica o en forma personal en la empresa. El administrador generará el respectivo presupuesto para luego enviárselo al cliente, este presupuesto podrá ser agilizado mediante el sistema de precarga que será implementado, para la precarga de los datos el administrador ingresará el Rut del cliente en el sistema y este arrojará al formulario de cotización de evento con los datos del último evento realizado por el cliente, en caso de ser el primer evento realizado por el cliente el sistema arrojará el respectivo formulario de cotización para ser llenado con los datos del cliente y del evento en cuestión, este presupuesto quedará almacenado en la base de datos en espera de la confirmación del cliente, una vez confirmado el presupuesto por el cliente, se procede a reservar los productos existentes en bodega necesarios para la realización del evento para la fecha indicada también se guardan todos los datos de importancia como lo son el nombre del cliente, Rut del cliente, fecha y hora de instalación, lugar del evento, etc. Además tendrá la opción de ingresar los datos de facturación para el evento.

El administrador tendrá la opción de consultar los eventos por su fecha, con el número de presupuesto, con el Rut del cliente, así como también mediante el nombre del cliente con lo cual el sistema le devolverá el o los eventos a realizar y los implementos necesarios para su realización, el administrador tendrá la opción de imprimir una guía de despacho con los materiales necesarios para la realización del evento, con la cual se procederá a sacar los productos de bodega para dirigirse al lugar del evento.

Una vez que el evento ha terminado y los implementos vuelven a la bodega se procederá a comparar la guía de despacho respectiva del evento con la carga del camión, luego el administrador procederá a actualizar la base de datos ingresando el número de guía de despacho en el sistema con lo cual este le arrojará la guía de despacho su respectivo detalle para que luego el administrador proceda a actualizar la base de datos con los artículos que han retornado.

El gerente tendrá la capacidad de solicitar reportes de las ventas percibidas por su empresa en un determinado tiempo, para esto el sistema pedirá al gerente ingresar un rango de fechas por las cuales se efectuará una búsqueda en la base de datos la cual arrojará todas las ventas percibidas durante el período, además el sistema le permitirá realizar consultas comparativas de las ventas percibidas en dos períodos de tiempo distintos y también podrá realizar consultas de cuál fue el artículo más vendido en un cierto período de tiempo. Esto es lo que pretende brindar ayuda al gerente en la toma de decisiones.

## 2.2 Solución Tecnológica Propuesta

Para dar solución a estos problemas se planteó lo siguiente:

- Para dar solución al problema de conocer los recursos con lo que se cuenta en bodega se creará una base de datos la cual almacenará la cantidad de recursos con los cuales se cuenta, será tarea del Administrador actualizar la base de datos a medida que los recursos salen de bodega o retornan luego de ser utilizados. O sea, para que el sistema cumpla con sus objetivos se necesita un compromiso real de parte de los usuarios ya que de no existir este compromiso el sistema tendrá poca o ninguna utilidad.
- El sistema podrá entregar reportes de las ventas realizadas por la empresa en un determinado período de tiempo, además de consultas comparativas entre dos períodos, y consultas tales como cuál fue el artículo más vendido dentro de un cierto período, lo que pretende apoyar la toma de decisiones de la organización. Esto será posible al almacenar cada venta percibida por la empresa con su respectivo detalle.
- Agilizar la búsqueda de los presupuestos para así ser capaces de dar respuestas rápidas y satisfactorias a los clientes según lo necesiten, esto se logrará ya que se podrá ejecutar la búsqueda de presupuestos mediante cuatro formas: con el número de presupuesto, con el Rut del cliente (ordenando los eventos según fecha), así como también mediante el nombre del cliente y por último mediante la fecha del evento.
- Proteger la información mediante la creación de un login específico para cada usuario con lo cual se restringirá el acceso a la información protegiéndose así la integridad de los datos.
- También se agilizará el proceso de confección de presupuestos ya que se implementará un sistema de precarga de los datos de un presupuesto, esto se logrará cargando los datos del último evento realizado por el cliente.

## 2.3 Arquitectura Planteada Como Solución

### 2.3.1 Arquitectura Lógica

Considerando la naturaleza académica y los alcances acotados del Proyecto, como también las características de la organización involucrada, se optó por implementar una arquitectura que permitiera obtener resultados óptimos en el menor tiempo posible con los recursos disponibles y sin la necesidad de incurrir en procesos superfluos que conllevarían a un mayor costo en la implementación de la solución.

Es por esto que se escogió la Arquitectura de dos Capas como estrategia de solución. Esta arquitectura se divide en: El llamado front-end (la interfaz del usuario, llamadas a SQL, aplicación de escritorio, etc.) y el llamado Back-end (servidor de Bases de datos SQL, Sistema operativo multitareas, etc.).



Ilustración 2.1 Arquitectura Lógica

### 2.3.2 Arquitectura Física

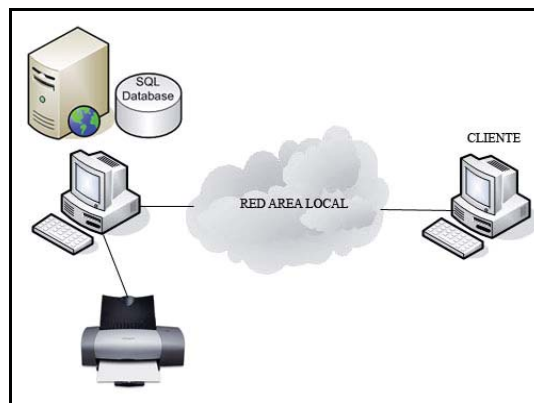


Ilustración 2.2 Arquitectura Física

Forma de trabajo:

- El administrador envía solicitudes al sistema por medio de las interfaces de la aplicación.
- El sistema procesa la solicitud yendo a la base de datos para obtener la información requerida.
- Por último se envía el resultado de la solicitud para ser mostrado en la aplicación.

### 2.3.3 Explicación de las Capas y Componentes de la Arquitectura

En la primera, es decir, en la capa de Presentación se dispone de una aplicación que permitirá gestionar las distintas funcionalidades del sistema, la administración de ventanas y formatos, etc.

También en esta capa se encuentra la parte lógica que se encarga de la unión entre Interfaz y Base de datos, se reciben datos ya sea provenientes del usuario como de la Base de Datos y efectúa los cálculos correspondientes según los requerimientos del usuario.

La segunda capa corresponde a la de Base de datos que es donde se almacenan los diversos datos relevantes para la empresa los cuales son requeridos por el o los clientes, así como también por la empresa.

### 2.3.4 Módulos del Sistema

En el sistema existirán 6 módulos:

- Módulo Administración de Productos.
- Módulo Informe de ventas.
- Módulo Administración de Evento.
- Módulo Administración de Presupuesto.
- Módulo Administración de Clientes.
- Módulo Administración de Usuarios.
- Módulo Administración de Proveedores

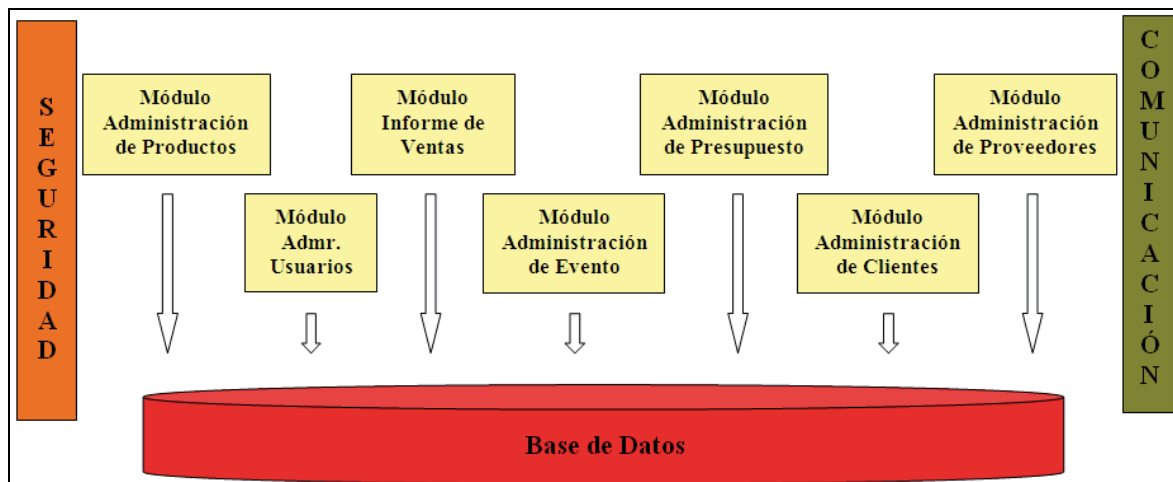


Ilustración 2.3 Módulos del Sistema

El módulo administración de evento es el que posee los requerimientos principales del usuario. Estos son a grandes rasgos la mejora del control y manejo de stock de bodega relacionado con un evento y la optimización operacional de la gestión de los eventos. Estos eventos poseen 3 estados: el estado “Presupuesto”, en el cual se guarda la cotización que recibió el administrador, la cual puede ser confirmada por el cliente. Otro estado o segundo estado en que se puede encontrar un evento es “Aceptado” en el cual se paga el 50% del presupuesto. Finalmente se pasa a un tercer estado “Realizado” cuando se envían los productos o recursos el día del evento para la realización de éste, el material es enviado según lo estipulado en la guía de despacho correspondiente a cada evento.

En el módulo de Informe de Ventas se obtiene información sobre las ventas percibidas en un cierto periodo de tiempo, además de realizar consultas comparativas de las ventas en un cierto período de tiempo dentro de un año en comparación del mismo periodo de tiempo en años anteriores y también permite realizar consultas de cuál fue el artículo más vendido en un cierto periodo de tiempo, todo lo anterior podrá prestar gran ayuda para la toma de decisiones a nivel

gerencial. Este módulo trabaja sobre los datos asociados a eventos en estado “Realizado” y/o “Aceptado”, los cuales se encuentran almacenados en el módulo de Administración de evento.

El módulo Administración de Presupuesto es aquel encargado de gestionar los presupuestos, ya sea para su creación o modificación, será tarea del administrador gestionar estos presupuestos.

El módulo Administración de Clientes será el encargado de gestionar los datos correspondientes a los clientes de la empresa, ya sea para ingresar datos de clientes nuevos, modificar datos ya existentes o eliminar datos relativos a un cliente.

El módulo Administración de Productos es el encargado de gestionar el ingreso, actualización y eliminación de los productos con los que cuenta la empresa, ya sea para el reemplazo o la compra de nuevos insumos.

El módulo Administración de Usuarios, es aquel que se encarga del ingreso, actualización y eliminación de usuarios del sistema.

Por último, el módulo Administración de Proveedores es el que permite realizar el ingreso, modificación o eliminación de los Proveedores que posee la empresa.

## **2.4 Herramientas a Utilizar**

### **2.4.1 Herramienta de Modelado: UML**

UML es un lenguaje para especificar, construir, visualizar y documentar los artefactos de un sistema de software orientado a objetos.

UML se quiere convertir en un lenguaje estándar con el que sea posible modelar todos los componentes del proceso de desarrollo de aplicaciones. Sin embargo, hay que tener en cuenta un aspecto importante del modelo: no pretende definir un modelo estándar de desarrollo, sino únicamente un lenguaje de modelado.

En UML los procesos de desarrollo son diferentes según los distintos dominios de trabajo; no puede ser el mismo el proceso para crear una aplicación en tiempo real, que el proceso de desarrollo de una aplicación orientada a gestión, por poner un ejemplo.

Las diferencias son muy marcadas y afectan a todas las fases del proceso. El método del UML recomienda utilizar los procesos que otras metodologías tienen definidos.

UML es una técnica de modelado de objetos y como tal supone una abstracción de un sistema para llegar a construirlo en términos concretos. El modelado no es más que la construcción de un modelo a partir de una especificación.

Un modelo es una abstracción de algo, que se elabora para comprender ese algo antes de construirlo. El modelo omite detalles que no resultan esenciales para la comprensión del original y por lo tanto facilita dicha comprensión.



## Artefactos para el Desarrollo de Proyectos

Un artefacto es una información que es utilizada o producida mediante un proceso de desarrollo de software. Pueden ser artefactos un modelo, una descripción o un software. Los artefactos de UML se especifican en forma de diagramas, éstos, junto con la documentación sobre el sistema constituyen los artefactos principales que el modelador puede observar.

Se necesita más de un punto de vista para llegar a representar un sistema. UML utiliza diagramas gráficos para obtener estos distintos puntos de vista de un sistema:

Algunos diagramas de UML son:

- Diagrama de Estructura Estática.
- Diagrama de Casos de Uso.
- Diagrama de Secuencia.
- Diagrama de Colaboración.
- Diagrama de Estados.

### 2.4.2 Base de Datos: SQL server

SQL Server es un sistema de gestión de bases de datos relacionales (SGBD) basado en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.

Entre sus características figuran:

- Soporte de transacciones.
- Gran estabilidad.
- Gran seguridad.
- Escalabilidad.
- Soporta procedimientos almacenados.

Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.

Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o que clientes de la red sólo accedan a la información. Además permite administrar información de otros servidores de datos.

Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle o Sybase.

## SQL como Lenguaje

Primero que todo debe quedar claro que SQL no es un lenguaje de programación como C o Pascal, ya que no posee sentencias de selección o iteración (IF ... THEN, FOR, WHILE, etc.).

Por lo tanto, cuando se desean utilizar los datos de la BD para realizar algún proceso, se debe programar en otro lenguaje (lenguaje anfitrión), desde el cual se hacen llamadas a sentencias SQL (SQL embebido) cuando se desea interactuar con la BD.

Además, como lenguaje es bastante natural y fácil de leer, y se aplica de forma natural al problema de implementar BD Relacionales, pero como consecuencia de esto es poco estructurado y utiliza muchas palabras superfluas (que podrían omitirse quitando claridad a las sentencias).

SQL permite abstraer al usuario de la estructura lógica de la BD mediante las vistas, aunque éstas imponen bastantes restricciones a los procesos de insertar y modificar datos en ellas. Además, habiendo sido estandarizado ya hace mucho tiempo, las BD Relacionales en SQL son muy portables, ya que existen muchos SGBD que lo soportan.

Al ser SQL un lenguaje declarativo, es muy fácil verificar los programas, ya que simplemente se declara lo que se desea como resultado de la ejecución.

### SQL y los SGBD

SQL es mucho más que un lenguaje de consulta de BD, ya que permite:

- Definir la estructura de los datos;
- Recuperar y manipular datos;
- Administrar y controlar el acceso a los datos;
- Compartir datos de forma concurrente;
- Asegurar su integridad.

Lo anterior se hace basándose siempre en el Modelo Relacional de datos. En consecuencia, SQL permite hacer efectivos los objetivos de los SGBD, permitiendo a los usuarios utilizar todo el potencial de los SGBD.

En conclusión, y tomando en consideración las distintas características y funcionalidades que presta esta herramienta que además se adecúan a las necesidades y naturaleza del proyecto, se optó por la utilización del Motor de SQL Server para la implementación de la capa de Base de Datos del Sistema.

### 2.4.3 Microsoft Visio

Microsoft Visio es un programa para la creación de diagramas que puede ayudarle a crear diagramas empresariales y técnicos en los que se documenten y organicen ideas, procesos y sistemas complejos. Los diagramas creados en Visio le permiten clara, concisa y eficazmente

visualizar y comunicar información, de unas formas no posibles utilizando exclusivamente texto y números.

Microsoft Visio es un paquete que emplea la creación y diseño de los diagramas, al cien por ciento de su capacidad. Es decir, que Microsoft Visio proporciona una amplia creatividad en el diseño de los diagramas, al igual que muchos tipos de diagramas que se pueden formular, desde un simple diagrama de flujo hasta un diagrama de lluvia de ideas con su esquema correspondiente, organigramas y mapas de sitios Web.

Por el mismo lado, Microsoft Visio posee todos los diagramas que utiliza UML para sus representaciones gráficas.

Además, Microsoft Visio es de fácil uso y ofrece mucha flexibilidad con respecto al manejo de los diagramas con los que se está trabajando.

#### **2.4.4 Microsoft Project**

Se utilizará esta herramienta para programar y organizar los recursos y tareas a fin de manejar el proyecto paso a paso, controlar su progreso y reportar información del mismo.

## **2.5 Paradigmas de Desarrollo de Software**

Para determinar cuál de todos los paradigmas que existen se acomoda de mejor manera al desarrollo de este sistema, se realizó un análisis detallado de cada uno de ellos señalando sus ventajas y desventajas.

### **2.5.1 Modelo en Cascada**

Este modelo es conocido también como Ciclo de vida clásico del software. El enfoque que este modelo le da al desarrollo es realizar las actividades en orden consecutivo, teniendo que finalizar una actividad antes de comenzar con la siguiente. Las actividades o etapas son *análisis y definición de requerimientos*, donde se definen las metas del sistema basado en la comunicación con los clientes, con esto se especifica lo que el sistema debe hacer; *diseño*, donde se dividen los requerimientos obtenidos en la primera etapa y se diseña o establece la arquitectura del sistema; *implementación y pruebas*, en esta etapa se lleva a cabo los programas que resultaron de la etapa de diseño. Estos programas deben probarse con el fin de que cada uno haga lo que debe hacer y lo realice correctamente. *Integración y prueba* es donde se integran todos los programas realizados en la etapa anterior para luego probarse como un conjunto asegurándose el correcto funcionamiento del sistema y del cumplimiento de todos los requerimientos. La última etapa

corresponde a la *Operación y mantenimiento*, en esta fase se instala el programa y comienza su utilización. El mantenimiento se refiere a corregir posibles errores que puedan surgir durante el uso del sistema y que no fueron descubiertos en las etapas anteriores.

Es probable que en la última etapa se encuentren errores o nuevas funcionalidades del sistema que no fueron cubiertas, por lo que es necesario realizar nuevamente todas las actividades. El problema de esto es que las iteraciones son costosas dado que cada etapa es documentada y aprobada nuevamente además de volver a realizar el trabajo, por lo que los costos en tiempo y dinero son altos.

### **Ventajas**

1. Si los requerimientos son bien especificados y entendidos en su totalidad, el sistema tiene muy buenos resultados.
2. Es fácil planificar las distintas etapas del desarrollo dado que están definidas.
3. Este modelo es la base para los otros paradigmas.

### **Desventajas**

1. En la realidad es muy difícil seguir este modelo, dado que es difícil que los proyectos sigan una secuencia estricta. Es más común encontrarse con proyectos en que las etapas se solapan.
2. Sólo se ve resultados al final del proyecto.
3. Si se hizo algo que no era lo que el cliente deseaba, esto se descubre una vez que el sistema está terminado o próximo a su finalización.
4. Difícil manejar los cambios de requerimientos por parte del cliente. Como los requerimientos son tomados en la etapa inicial del proyecto, es difícil tomar más requerimientos o modificar los existentes, durante las etapas posteriores.
5. Si se descubre un error de manera tardía, su solución es muy costosa.

## **2.5.2 Desarrollo Evolutivo**

La idea de este paradigma es desarrollar una implementación inicial y mostrársela al cliente (o usuario) con el fin de que éste pueda hacer sus objeciones y comentarios para ir refinando esta implementación inicial, a lo largo del tiempo. Las actividades se llevan a cabo en conjunto, es decir, la especificación, desarrollo e implementación se realizan juntas y entre ellas se retroalimentan de manera rápida.

Dentro del desarrollo evolutivo podemos encontrar dos modalidades de trabajo. Una es trabajar junto al cliente en el desarrollo del sistema y haciendo evolucionar este mismo con nuevos aportes dados por el cliente hasta llegar a un resultado final, esta modalidad es conocida como desarrollo exploratorio. La otra manera es tomar los requerimientos dados realizar una definición mejorada de los mismos y desarrollar un sistema basado en estos, experimentando en

aquellos requerimientos que no son del todo entendibles. Por este motivo, una vez desarrollado esto, se muestra al cliente el cual decide si lo que se realizó es acorde a lo que él deseaba, en caso contrario se debe desarrollar nuevamente esa parte, para ver si se logró entender los requerimientos hasta llegar a una solución óptima. Esta segunda modalidad se conoce como prototipos desechables.

Las etapas del desarrollo evolutivo, sea cual sea la modalidad que se esté usando, son las siguientes:

**Especificación inicial:** Se toman los requerimientos del cliente para poder realizar el trabajo.

**Desarrollo del producto:** Se desarrolla un producto basado en las especificaciones dadas por el cliente al inicio (o durante esta etapa de desarrollo).

**Implementación, uso y evaluación:** Se entrega el producto al cliente, para que este pueda utilizarlo y determinar si el sistema cumple con las especificaciones y satisface las necesidades que se tienen. A partir de este punto, pueden surgir dos etapas o hitos: la re-especificación en caso que el cliente quiera hacer modificaciones al producto entregado por no ser lo que este quería o por cualquier otro motivo que implique un nuevo desarrollo; y las versiones del sistema, a medida que se va iterando, van surgiendo las distintas versiones, hasta llegar a la última versión del software, la que cumple con todos los requerimientos del cliente.

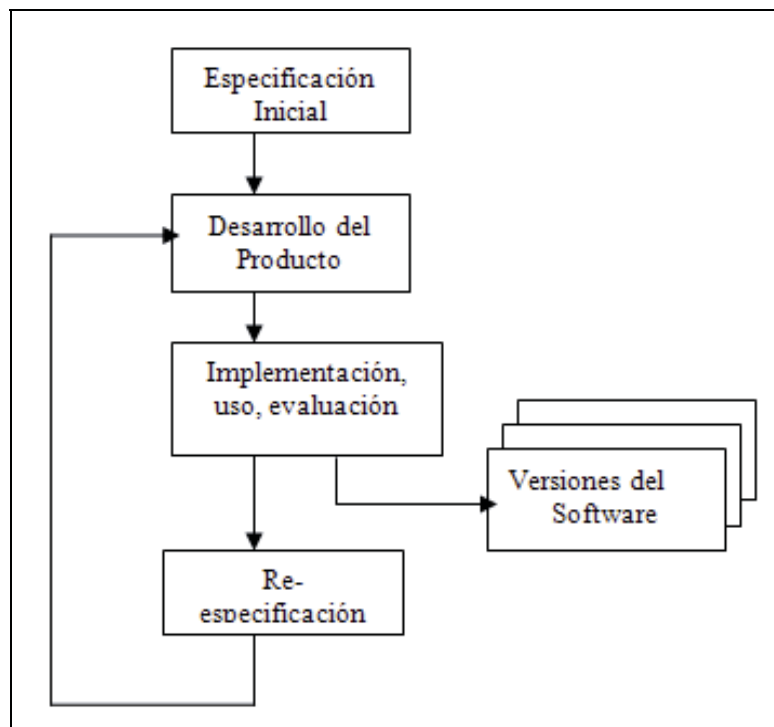


Ilustración 2.4 Desarrollo Evolutivo

**Ventajas:**

1. Muy efectivo para proyectos pequeños.
2. Funciona bien si el cliente con el desarrollador llegan a un buen entendimiento inicial en cuanto a la modalidad de trabajo, dejando en claro que se trabajará con prototipos y que las distintas entregas no serán una versión final, sino que serán iteraciones para refinar lo que será la última versión del producto. De no quedar esto claro, este punto se convierte en una desventaja para este paradigma.
3. Asegura el cumplimiento total (o en una gran cantidad) de los requerimientos, dado que el cliente puede ir viendo lo que se hace, realizar acotaciones o nuevos requerimientos, con el fin de que el sistema entregado, sea lo más cercano a lo que el desea, sin tener cosas de más o de menos.

**Desventajas:**

1. No es posible ver el proceso. Es difícil documentar todo si se trabaja de manera rápida, por lo que se desarrolla e implementa sin una documentación que pueda reflejar las distintas versiones del sistema.
2. El cliente puede no entender que lo que se le entrega es un prototipo, pensar que es el producto final y desmoralizarse por tener un sistema que no cumple con los requerimientos exigidos.

3. Una desventaja desde el punto de vista académico, es que no se puede trabajar con este modelo, debido a la rigidez con la que está estructurado el ramo de proyecto 1. Las fechas de entrega de los informes, más la cantidad de informes, hace que este paradigma sea difícil (sino imposible) de utilizar para los propósitos del ramo. Si se considera que existen instancias de entrega de informe, sería difícil estar trabajando en una iteración más el informe de este, ya que dicho informe no podría ser tan detallado o no mostraría lo que realmente se está haciendo. Por otra parte, la realización de informes, restaría tiempo a la creación de los prototipos, la parte más importante de este enfoque.

### 2.5.3 Desarrollo en Espiral

Es un paradigma evolutivo de desarrollo de software que conjuga la naturaleza evolutiva de construcción de prototipos con los aspectos controlados y sistemáticos de ciclo de vida clásico, el cual proporciona el potencial para el desarrollo rápido de versiones incrementales de software [1] , [2] . Cada ciclo se divide en 4 etapas:

Definición de objetivos, en donde se definen los objetivos específicos, las restricciones y se desarrolla un plan de administración.

La segunda etapa es la evaluación y reducción de riesgos, donde se detallan los riesgos que pueden surgir en el proyecto y el plan de contingencia para identificar, estimar y ver las formas de gestionar dichos riesgos, buscando alternativas de solución en caso de que se presenten.

La tercera etapa, es la de desarrollo y validación, donde se elige un modelo para el desarrollo del sistema.

Por último, la cuarta etapa, la de planeación, es aquella en que el proyecto se evalúa y se decide si se debe continuar con su ejecución, tomando las consideraciones para seguir con el siguiente ciclo del espiral.

#### Ventajas

1. A diferencia de los otros paradigmas, es que el riesgo se considera de manera explícita para cada una de las etapas, conduciendo con esto a una mejor planificación y disminuye los riesgos.
2. Útil para proyectos pequeños.

## Desventajas

1. Es necesario controlar muy bien el riesgo. En caso de no ser así, riesgos pequeños pueden transformarse en problemas mayores a medida que avanzan las distintas etapas del proyecto.
2. Puede considerarse más complejo, dado que no existen fases fijas. Este modelo puede contener otros modelos. En el ámbito académico esto puede ser una gran dificultad dado que la capacidad de trabajar con más de un modelo en un proyecto esta dada por la experiencia y conocimientos avanzados que se tengan.

### 2.5.4 Proceso Unificado de Desarrollo (UP)

Además de ser un paradigma de desarrollo de software basado en la metodología de orientación a objetos, UP es un marco de trabajo genérico capaz de especializarse en una gran variedad de sistemas, diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos. UP está compuesto por un conjunto de actividades necesarias para pasar de los requisitos dados por el cliente a un producto de software [3] , [4] .

El Proceso Unificado guía a los equipos de proyecto en cómo administrar el desarrollo iterativo de un modo controlado mientras se balancean los requerimientos del negocio, el tiempo al mercado y los riesgos del proyecto. El proceso describe los diversos pasos involucrados en la captura de los requerimientos y en el establecimiento de una guía arquitectónica lo más pronto, para diseñar y probar el sistema hecho de acuerdo a los requerimientos y a la arquitectura. El proceso describe qué entregables producir, cómo desarrollarlos y también provee patrones.

El Proceso Unificado es soportado por herramientas que automatizan entre otras cosas, el modelado visual, la administración de cambios y las pruebas.

El ciclo del Proceso Unificado consta de cuatro etapas: Inicio, donde se planifica el proyecto, Elaboración, donde se establece un plan y una arquitectura acorde a las necesidades del sistema; la etapa de Construcción donde es desarrollado el sistema y la etapa de Transición, en la cual el sistema es proporcionado a los usuarios finales.

## Ventajas

1. Puede ser iterativo o lineal, dependiendo del tamaño del sistema.
2. Permite trabajar con un enfoque dirigido a la utilización del sistema (casos de uso).
3. Permite identificar y gestionar riesgos.
4. Se centra en la arquitectura del sistema.



### 2.5.5 Selección del Paradigma a Utilizar

Con la idea de hacer la mejor elección de un paradigma de desarrollo para el sistema, es que se tomó el sistema y se ideó sobre los distintos paradigmas nombrados anteriormente. La idea de esto, era ver el comportamiento del sistema ante las diversas características, ventajas y desventajas de cada paradigma. De esta manera se pudo ver qué características facilitan el desarrollo del sistema en cuestión y cuales lo complican o simplemente no son útiles para el desarrollo por lo que hacen que el paradigma sea inutilizable para este caso.

A continuación se nombran, a modo de ejemplo, las características de los paradigmas, pero, a diferencia de la descripción anterior, ahora se nombrarán las características que sirven y aquellas que no sirven para que el proyecto sea desarrollado bajo este paradigma.

#### **Modelo En Cascada**

Este paradigma muestra una serie de características que no hacen posible la realización óptima del proyecto. Puede que ofrezca virtudes, pero son tantas las falencias que presenta que se decidió no realizar el sistema basado en este paradigma. Estas falencias son las siguientes

- Solo muestra los resultados al final del proyecto, lo cual es un punto en contra dado que el cliente quiere estar en amplio contacto con el desarrollo del sistema, por lo que quiere ir viendo avances y resultados en la medida que pasa el tiempo, por lo que, tener resultados sólo al final del proyecto, no permite participar de la mejor manera posible.
- El cliente, a pesar de haber realizado requerimientos, dejó abierta la posibilidad de seguir haciendo más requerimientos en la medida que pase el tiempo y el vaya reconociendo más falencias del sistema actual que puedan ser solucionadas. Esto no es manejable por este modelo, dado que los requerimientos son hechos al inicio del proyecto y se trabaja con estos hasta el final.

Por estos motivos se decidió no realizar el proyecto siguiendo este modelo

#### **Desarrollo Evolutivo**

Este modelo nos muestra ciertas características que podrían ser útiles para el desarrollo del sistema. Estas principalmente están dadas por la capacidad que da el desarrollo evolutivo de incluir en gran porcentaje la participación del cliente durante todas las fases del proceso. Por otro lado permite incluir requerimientos a lo largo del proyecto.

Los inconvenientes que se produjeron al simular el sistema bajo este paradigma son los siguientes:

- Bajo este modelo no se realiza tanta documentación, sino que el trabajo se realiza concentrado más en la creación de los prototipos. Ante la necesidad académica de entregar informes en fechas determinadas, se hace difícil trabajar basado en este paradigma, debido a que el tiempo usado para la creación de prototipos, es usado en gran parte en el desarrollo de informes y documentaciones del proyecto.
- Al conocer los gustos y exigencias del cliente, es fácil darse cuenta que ante un prototipo, éste piense que es el sistema final y en vez de aportar con los cambios que se deben hacer, puede pensar que el sistema no cumple con lo que el esperaba y desilusionarse.

### **Desarrollo en Espiral**

El desarrollo en espiral brinda características muy útiles respecto al tratamiento del riesgo por lo que apoya en gran parte al desarrollo de un sistema seguro.

En realidad, este modelo presta una gran utilidad para el desarrollo del proyecto, pero presenta el inconveniente de que el modelo no presenta fases fijas, es probable pasar por una misma etapa reiteradas veces con el fin de conseguir que esta quede bien definida para pasar a la siguiente.

Cabe destacar que la experiencia de los desarrolladores bajo este paradigma es baja, y el no conocer herramientas para el uso de este modelo puede traer pérdida de tiempo, no cumplir con los objetivos realizados y no poder realizar el sistema de la mejor manera posible.

Otra desventaja es la poca comunicación con el cliente. Si bien puede existir comunicación, esta no se da en la cantidad, ni con la fluidez esperada. Como último punto se puede ver que este modelo no asegura resultados rápidos, lo cual es una limitación importante para el desarrollo del sistema bajo este modelo, dadas las exigencias académicas y del cliente.

### **Proceso Unificado de Desarrollo**

Este paradigma presenta una serie de características que permiten una realización adecuada del sistema, dado que las características de este modelo fueron sin dudas las que de mejor manera se comportaron ante todas las circunstancias requeridas por el desarrollo del sistema para poder entregar una buena solución al cliente. Dentro de estas características se puede encontrar herramientas de fácil uso, o que son conocidas por los desarrolladores, asegurando al máximo una buena producción por parte de estos. Por otra parte, se tiene una buena documentación como material de apoyo así como también la que se entrega, ya sea para fines del sistema como académicos. Si bien es cierto, se trata de un paradigma que no tiene una fácil planificación, posee características buenas que provienen del modelo evolutivo, las que mezcladas con la documentación, la experiencia de los desarrolladores y una estructura de desarrollo bien diseñada permiten realizar el sistema de la mejor manera posible.

Entre las características que mejor se adaptan al desarrollo del sistema, se encuentran:

- Comunicación con el cliente.
- Tiempo reducido en la obtención de resultados visibles.
- Enfocado al desarrollo.
- Permite requerimientos dentro de las distintas etapas de desarrollo.
- Iteraciones.
- Entrega de documentos de avance del proyecto.

Debido a todas estas características, se decidió realizar el proyecto bajo el paradigma de Proceso Unificado de Desarrollo.

Con la idea de hacer más entendible la elección del paradigma, se creó una tabla que muestra una serie de características y los distintos paradigmas, marcando aquellas características que posee el modelo. Sólo se pusieron características ventajosas, para poder ver mejor que virtudes ofrecían ciertos paradigmas en comparación a los demás.

**Tabla 2-1 Virtudes de Paradigmas**

	Modelo en Cascada	Desarrollo Evolutivo	Desarrollo en Espiral	Proceso Unificado de Desarrollo
Corto tiempo para obtener resultados visibles.		✓	✓	✓
Facilidad de planificación.	✓	✓	✓	
Requerimientos en distintas etapas.		✓	✓	✓
Entrega de documentación.	✓			✓
Orientado al desarrollo.		✓	✓	✓
Manejo de riesgos.	✓		✓	✓
Documentación disponible acerca del paradigma.	✓	✓		✓
Adaptable a exigencias académicas.	✓			✓
Adaptable al proyecto.				✓
Experiencia de los desarrolladores.	✓			✓
Buena comunicación con el cliente.		✓		✓
Iteraciones.		✓	✓	✓
Disponibilidad de herramientas.	✓	✓	✓	✓
Cliente relacionado con el desarrollo del sistema.		✓		✓

Tomando en cuenta la naturaleza del proyecto a realizar y tras analizar las distintas metodologías disponibles en la actualidad se ha decidido usar UP (Unified Process). Debido a que se basa en muchos años de experiencia en el uso de la tecnología de objetos en el desarrollo de software.

## **2.6 Metodologías de Desarrollo de Software**

### **2.6.1 Metodología Estructurada**

En este tipo de Metodología se utiliza una notación que muestra el flujo y el contenido de la información. El sistema se divide funcionalmente y dependiendo de los distintos comportamientos que este muestre, se establece lo que se debe construir. Existen dos enfoques de esta Metodología: Enfoque estructurado clásico y Enfoque estructural esencial.

Enfoque estructurado clásico: Consiste en describir el sistema actual, para luego redefinir aquello que hace, con el fin de conformar una nueva versión para el proceso. Se termina con la decisión acerca de cuál será el grado de automatización que se le incorporará al sistema actual a modo de satisfacer las necesidades del cliente.

Enfoque estructural esencial: El modelo esencial del sistema, es un modelo de lo que el sistema debe hacer para satisfacer los requerimientos del usuario, sin importar de que manera esto se implementará. La idea es centrarse en lo que se desea hacer con la idea de que todo es realizable para cumplir las necesidades del cliente.

### **2.6.2 Metodología Orientada a Objetos**

Esta metodología se basa en clases y objetos los que pueden ser cualquier cosa real o abstracta. Los objetos existen para prestar servicios ya que al pasarle ciertos parámetros, los objetos realizan una acción determinada. Esta operación ejecuta el método apropiado, generando una respuesta.

En este enfoque se modela en términos de tipos de objetos y de lo que a estos le ocurre o lo que estos realizan. Un tipo de objeto puede tener varios subtipos. Además como una clase implementa el tipo de objeto, una subclase puede heredar propiedades de su clase padre; puede heredar su estructura de datos y los métodos, o algunos de sus métodos.

Existen dos tipos de modelos para este enfoque:

Análisis de la estructura de objetos: Este análisis guía a la representación de los tipos de objetos que le son propios al sistema y las relaciones que mantienen mediante diagramas de relación entre estos.

Análisis de Comportamiento de Objetos: Permite ver que es lo que le ocurre a los objetos con el paso del tiempo. Los estados en los cuales puede estar un objeto y las transiciones de estado en que puede estar, se muestran mediante diagramas de transición de estados, los que permiten ver el comportamiento de un objeto en distintas ocasiones.

### **2.6.3 Selección De Metodología a Utilizar**

Luego de evaluar las distintas metodologías de desarrollo, se decidió utilizar la metodología de Orientación a Objetos, debido a que presenta una serie de características que se adaptan de buena manera al proyecto desarrollado en comparación a las otras metodologías.

Si bien las otras metodologías mencionadas, poseen también, características útiles para el sistema, al realizar una comparación de las distintas metodologías, se pudo observar que eran las características del enfoque orientado a objeto las que mejor se adaptaban al desarrollo del sistema en cuestión.

Al evaluar las características de claridad, diagramas, entendimiento por parte del cliente y del desarrollador, reusabilidad, adaptación de esta metodología al paradigma escogido, rapidez del desarrollo, comunicación con el cliente y tipo de modelado, se pudo obtener como resultado la elección de la metodología de orientación a objetos por sobresalir en sus características ante las demás metodologías.

Por último, esta metodología es la que de mejor manera se acomoda al paradigma elegido (Proceso Unificado de Desarrollo) dado que el uso de UML que permite dicho paradigma, está orientado a la metodología de orientación a objetos.

## **2.7 Estudio de Factibilidad**

Siempre a la hora de llevar a cabo un proyecto de cualquier índole es de vital importancia realizar un estudio y evaluación de su factibilidad, es decir, conocer la disponibilidad de los recursos necesarios para lograr los objetivos establecidos, como también comprender las limitaciones potenciales inherentes a la ejecución del proyecto.

La información obtenida mediante este estudio de factibilidad es indispensable debido a que indicará si las limitaciones o condiciones evaluadas permitirán o no la realización del proyecto en su totalidad, ya sea con los recursos establecidos originalmente o si se deben seguir cursos alternativos para lograr que este sea viable.

Para realizar un estudio de factibilidad óptimo se deben revisar cuatro aspectos básicos:

- Factibilidad Técnica.
- Factibilidad Operacional
- Factibilidad Económica.
- Factibilidad Legal.

### **2.7.1 Factibilidad Técnica**

El análisis de factibilidad técnica evalúa si las tecnologías: el equipo y software están disponibles (o, en el caso del software, si puede desarrollarse) y si tienen las capacidades técnicas requeridas por cada alternativa del diseño que se esté considerando. Por otra parte, los estudios de factibilidad técnica también consideran si se cuenta con la experiencia técnica requerida para diseñar, implementar, operar y mantener el sistema propuesto.

Se debe considerar lo siguiente:

- Recursos de Software.
- Recursos de Hardware.
- Recursos Humanos.

#### **Recursos de Software**

Esto se refiere al software con el que se debe contar para la realización del proyecto. Los recursos de software que están contemplados a ocupar en este proyecto son los siguientes:

1. Para el análisis de requerimientos y diseño del sistema:
  - Microsoft Office 2003.
  - Microsoft Visio 2003.
  - Microsoft Project 2003.
2. Para la implementación del sistema:
  - Microsoft Visual Studio
  - Lenguaje de Programación Visual Basic .NET
  - Windows 2000 Professional
3. Para la implantación y pruebas del sistema:
  - Microsoft SQL Server.

De estos recursos mencionados, la empresa deberá adquirir las licencias que no posea. Sin embargo, dado que este proyecto es de carácter académico, el producto de software se realizará en la Escuela de Ingeniería en Informática de la Pontificia Universidad Católica de Valparaíso, la que cuenta con los recursos suficientes para la realización del proyecto.

### **Recursos de Hardware**

Estos se refieren a ver si la empresa cuenta con el hardware necesario para la realización del proyecto.

Los recursos de hardware que están contemplados a ocupar en este proyecto son los siguientes:

1. Para el análisis, diseño y construcción del producto de software:

- Procesador: Pentium III – 750 Mhz o Similar.
- Ram: 128 Mb.
- Espacio libre en Disco Duro: 2 Gb.
- Monitor 15”soporte resolución 800x600, Teclado y Mouse.

2. Para la implantación y prueba del producto:

- Procesador: Pentium IV – 2.4 Ghz o Similar.
- Ram: 512 Mb.
- Espacio libre en Disco Duro: 40 Gb.

3. Para la operación con el sistema:

- Procesador : Pentium II – 400 Mhz. ó Similar
- Ram : 64 MB
- Tarjeta de Red : Tarjeta de Red 10/100 Mb
- Monitor 15”soporte resolución 800x600, Teclado y Mouse.

Debido a que es un proyecto académico, en la parte de análisis, diseño y construcción se utilizará el hardware facilitado por la escuela.

En la empresa los computadores de escritorio que poseen superan los requisitos para la operación con el sistema por lo que no hay problemas en cuanto a Hardware se refiere.

## Recursos Humanos

En esta sección se evalúa la preparación del equipo desarrollador. Si bien es cierto que el equipo desarrollador no posee una gran experiencia en la realización de proyectos, el conocimiento adquirido durante la carrera y un trabajo de investigación permanente, ayudará a terminar el proyecto en forma eficiente.

En conclusión, el proyecto es técnicamente factible.

### 2.7.2 Factibilidad Operacional

Este estudio pretende determinar la probabilidad de que el nuevo sistema se opere tal como fue ideado, siguiendo los parámetros establecidos y los cursos de acción definidos, logrando que se cumpla realmente con los objetivos y pueda solucionar los problemas por los que ha sido diseñado. Para garantizar que lo anterior se cumpla, se deberían tomar en cuenta los siguientes aspectos importantes:

Primero, el nuevo sistema no debe ser demasiado complejo para los usuarios u operadores, ya que esto podría causar que el sistema no se utilice de manera óptima y se produzcan posibles fallas o errores. Segundo, el sistema no debe motivar un sentimiento de renuencia o aversión en los usuarios, es fundamental que la resistencia al cambio sea mínima. Tercero, el sistema nuevo no debería introducir cambios radicales de forma apresurada para promover una mejor adaptación. Una última consideración es la probabilidad de la obsolescencia prematura de las tecnologías utilizadas dejando, por implicancia, todo el sistema obsoleto.

Para que el sistema sea aceptado de buena manera por los usuarios se crearán interfaces amigables con tecnología de punta y se mantendrá la estructura de la manera en que se trabaja hoy en día en la empresa, por lo que desde el punto de vista operacional el proyecto es viable.

### 2.7.3 Factibilidad Económica

Los estudios de factibilidad económica incluyen análisis de costos y beneficios esperados inicialmente y que van asociados al proyecto en su conjunto.

Entre los puntos importantes a tomar en cuenta en este tipo de factibilidad estarían:

- El costo de llevar a cabo la investigación y desarrollo completo del sistema.
- El costo del hardware y software para la aplicación.
- Los beneficios en la forma de reducción de costos o de menos errores costosos.
- El costo si nada sucede (si el proyecto no se lleva a cabo).

En este caso, por tratarse de un proyecto de tipo académico, la Universidad brinda todas las herramientas necesarias para llevar a cabo el desarrollo de este sistema.



Sin perjuicio de lo anterior y para efectos de este estudio, se procedió a estimar los costos monetarios asociados a cada uno de los conceptos involucrados en la realización del proyecto.

En primer lugar, se debe considerar el costo asociado al total de horas hombre requeridas para el análisis y desarrollo completo del Sistema. Este monto se estimó en 204 UF, correspondiente al desempeño realizado por dos analistas desarrolladores en el transcurso de dos meses de trabajo efectivo.

Con respecto a las licencias de Softwares necesarias, la versión Microsoft SQL Server 2005 Express Edition, utilizada como almacén de datos, es gratuita, por lo que es económicamente factible de utilizar.

Tomando en cuenta todos los puntos anteriores, el valor total aproximado del Proyecto es de 204 UF que corresponden sólo al valor del trabajo realizado.

Se informaron estos costos a la administración de la empresa Carpas Viña y fueron aceptados considerando el beneficio que traería a la empresa, por lo que este proyecto desde el punto de vista económico es viable.

#### **2.7.4 Factibilidad Legal**

Con el estudio de factibilidad legal se tiene como principal objetivo verificar si el proyecto que se va a realizar no caerá en ninguna infracción, violación o responsabilidad legal de acuerdo a las leyes y decretos establecidos, ya sea por parte del estado o de la organización en sí.

Para este proyecto se debe considerar lo siguiente:

- Compra de licencias para el desarrollo del sistema, ya que se utilizarán las herramientas de desarrollo con que cuenta la universidad.
- Copia o adaptación de proyectos informáticos anteriores, ya que el desarrollo será para una empresa en particular, por lo tanto será un software a medida.
- Por último se puede señalar que el software no daña, ni altera, algún sistema de información.

Por lo citado anteriormente el proyecto es legalmente factible.

#### **2.7.5 Conclusión de Estudio de Factibilidad**

Puesto que se comprobó que el proyecto es factible en todos sus ámbitos, se puede concluir que este sistema puede ser desarrollado sin ningún problema.

## **2.8 Análisis de Riesgo**

Dentro de lo que es el desarrollo de un sistema, se debe tomar en cuenta ciertos factores que pueden incidir en lo que será el desarrollo y puesta en marcha de este. Por ende, se debe hacer un análisis a todas las posibles falencias que se podrían presentar mientras el sistema se desarrolla y luego se ejecuta.

Existen tres tipos de riesgos, que son:

Los riesgos del proyecto, estos son los riesgos que amenazan al plan del proyecto. Es decir, si los riesgos del proyecto se hacen realidad, es probable que la planificación temporal del proyecto se retrase y que los costos aumenten.

Los riesgos técnicos, estos son los que amenazan la calidad y la planificación temporal del software que hay que producir. Si un riesgo técnico se convierte en realidad, la implementación puede llegar a ser difícil o imposible.

Los riesgos del negocio, estos amenazan la viabilidad del software a construir. Los riesgos del negocio a menudo ponen en peligro el proyecto o el producto.

Tomando en cuenta los tipos de riesgos y su descripción, se llevará a cabo un análisis de riesgos que puedan influir en el desarrollo y la puesta en marcha del sistema, ya que existen ciertos tipos de riesgos que si no se toman en cuenta pueden llevar a que todo el desarrollo sea en vano.

### 2.8.1 Identificación del Riesgo

**Tabla 0-1 Identificación del Riesgo**

Riesgo	Tipo			Impacto			Probabilidad			Marco de tiempo		
	Proyecto	técnico	Negocio	Marginal	Critico	Catastrófico	Muy Probable	Probable	Improbable	Corto Plazo	Medio Plazo	Largo Plazo
Recursos limitados	X				X			X			X	
Planificación Temporal	X				X			X		X		
Poca experiencia del equipo de trabajo	X				X			X			X	
Subestimación del tamaño del sistema.	X				X				X	X		
Mala acotación de los límites del sistema	X				X			X			X	
Errores de codificación.		X			X				X		X	
Interfaces complejas para el usuario del sistema.		X			X				X		X	
Sistema de difícil mantenimiento.		X				X			X			X
Datos incorrectos entregados por el usuario del sistema.			X			X		X			X	
El sistema no cumpla las expectativas del usuario.			X	X					X			X
El usuario no utilice el sistema y siga trabajando con los métodos antiguos.			X		X				X		X	
Poca comunicación entre el usuario del sistema y los desarrolladores de este.			X			X			X	X		

Para que la tabla anterior pueda entenderse mejor, se puede observar la tabla siguiente que muestra valores asociados con respecto al análisis realizado anteriormente.

**Tabla 0-2 Clasificación de Riesgos**

Atributo	Valor	Descripción
Impacto	Catastrófico	Perdida del sistema. Costo >50%
	Critico	Recuperación de la capacidad Operativa. Costo >10% (<50%)
	Marginal	Costo <10%
Probabilidad	Muy Probable	> 70%
	Probable	De 30% a 70%
	Improbable	< 30%
Marco de tiempo	Corto plazo	30 días
	Medio plazo	1 a 4 meses
	Largo plazo	Más de 4 meses

## 2.8.2 Gestión del Riesgo

**Tabla 0-3 Gestión del Riesgo**

<b>Riesgo</b>	<b>Estrategia</b>	<b>Categoría</b>
Recursos limitados	Se utilizará HW y SW perteneciente al grupo de trabajo, como también se hará con el que posee la Escuela de Informática de la PUCV.	Disminución
Planificación Temporal	Se realizarán reuniones con una alta tasa de frecuencia, además se pondrá un gran énfasis en cumplir con las fechas estipuladas en la carta Gantt.	Anulación
Poca experiencia del equipo de trabajo	Investigando los temas sobre los que se tiene poco dominio y complementando conocimiento.	Anulación
Subestimación del tamaño del sistema.	Se deben tener claros los requerimientos tanto funcionales como no funcionales del sistema.	Anulación
Mala acotación de los límites del sistema	Teniendo claros los requerimientos y efectuando reuniones periódicas con el cliente y usuario.	Anulación
Errores de codificación.	Probando cada módulo por separado antes de seguir avanzando.	Anulación
Interfaces complejas para el usuario del sistema.	Mediante pequeños prototipos de interfaz mostrados a los usuarios, y éstos puedan elegir el más adecuado	Disminución
Sistema de difícil manutención.	Se desarrollará de forma limpia y clara, además de documentar los hechos importantes.	Disminución
Datos incorrectos entregados por el usuario del sistema.	En ciertas ocasiones se deberá hacer una evaluación en terreno del uso del sistema.	Anulación
El sistema no cumpla las expectativas del usuario.	Mediante reuniones periódicas dejando bien claro el funcionamiento del sistema.	Anulación
El usuario no utilice el sistema y siga trabajando con los métodos antiguos.	Se incitará al usuario a utilizar el sistema, enseñándole las virtudes que éste posee.	Anulación
Poca comunicación entre el usuario del sistema y los desarrolladores de éste.	Mediante reuniones periódicas y pactadas con anterioridad.	Disminución.

# Capítulo 3: Desarrollo Fase Inicio

## 3.1 Primera Iteración

### 3.1.1 Clasificación de Requerimientos

#### 3.1.1.1 Requerimientos Funcionales

1. Mantenición de presupuestos relacionados a un evento. (Creación, modificación y eliminación).
2. Búsqueda de registros de Eventos (Por realizar, realizados).
3. Mantenición de registros asociados a los Clientes de la Empresa. (creación, modificación y eliminación).
4. Búsqueda de registros de Clientes.
5. Mantenición de registros asociados a los Proveedores de la Empresa (Creación, modificación y eliminación).
6. Búsqueda de registros de Proveedores.
7. Generación de reportes estadísticos de las ventas de un cierto período.
8. Generación de reportes estadísticos del o los artículos más solicitados en cierto período.
9. Generación de reportes estadísticos entre las ventas emitidas en un cierto período en comparación a las realizadas en un período distinto.
10. Generación de guías de despacho.
11. Impresión de facturas.
12. Registrar las fechas de las cotizaciones por parte de los clientes.
13. Registrar las fechas de las confirmaciones de realización de eventos por parte de los clientes.
14. Mantenición de registros asociados a los Productos de la Empresa (Creación, modificación y eliminación).
15. Disponer información acerca de la disponibilidad de los productos.
16. Búsqueda de registros de Productos.
17. Mantenición de registros asociados a los Usuarios del Sistema (Creación, modificación y eliminación).
18. Entrega de alternativas de tipo similar en caso de no contar con la disponibilidad de ciertos productos.
19. Validación de existencia de topes de fecha entre los eventos.

### 3.1.1.2 Requerimientos No Funcionales

1. Evitar errores en datos de entrada (sustituir recurso manual)
2. El sistema debe dar seguridad a los datos almacenados en el sentido de guardarlos y permitir su respaldo con el fin de asegurar la integridad de estos.
3. Tiempos de respuesta al usuario de no más de 2 segundos.
4. Brindar seguridad en el acceso a la información mediante la implementación de cuentas de usuarios y contraseñas.
5. Integración de los distintos módulos del Sistema
6. El uso del Sistema ha de ser fácil de entender, independientemente de la experiencia, conocimiento o nivel de concentración del usuario.
7. El Sistema debe contar con material de apoyo para el usuario (Manual de Usuario).
8. El Sistema debe permitir la recuperación de errores.
9. En caso de errores o equivocaciones el sistema debe mostrar mensajes instructivos y no amenazantes.

## 3.1.2 Análisis

### 3.1.2.1 Actores

**Tabla 3-1 Actores del Sistema**

Actores	Descripción
Gerente	Persona que interactúa con el sistema con el fin de obtener reportes elaborados en cuanto a las ventas, lo cual le ayudara en la toma de decisiones.
Administrador	Sujeto que interactúa de lleno con el sistema, ingresa los datos de los clientes al sistema. Consulta la disponibilidad de material para la realización del evento y maneja la entrada y salida de estos. Además, es quien realiza las ventas de la empresa y devuelve las solicitudes de los clientes que llegaran a través de la página Web.

### 3.1.2.2 Diagrama General de Casos de Uso.

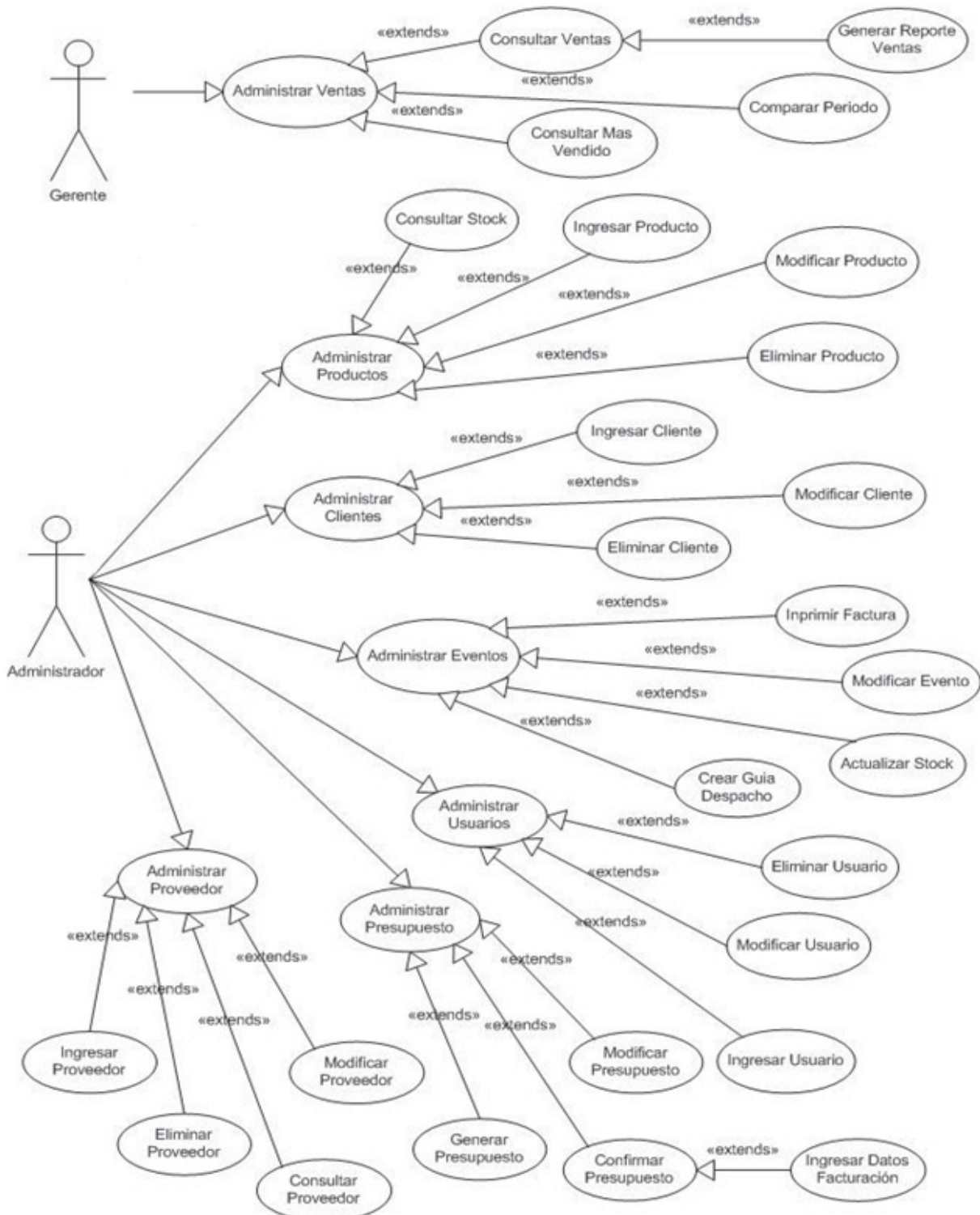


Ilustración 3.1 Diagrama General de Casos de Uso

# Capítulo 4: Desarrollo Fase Elaboración

## 4.1 Primera Iteración: Sesión

### 4.1.1 Análisis

#### 4.1.1.1 Diagramas de Casos de Uso

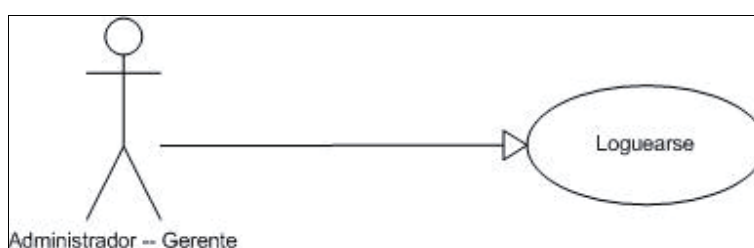


Ilustración 4.1 Caso de Uso: Loguearse

### 4.1.2 Diseño

#### 4.1.2.1 Especificación de Casos de Uso

Tabla 4-1 Caso de Uso: Loguearse

Caso de Uso:	Loguearse.
Objetivo:	Permitir seguridad y entregar las funciones a los distintos usuarios del sistema.
Actor Principal:	Administrador y Gerente.
Actor Secundario:	No hay.
Precondiciones:	No hay.
Poscondiciones:	El usuario ingresa al sistema y se muestran las interfaces dependiendo de su sesión.
Escenario Principal (flujo básico):	<ol style="list-style-type: none"><li>1. El sistema muestra los campos "Rut de Usuario" y "Contraseña".</li><li>2. El usuario ingresa su Rut de usuario y contraseña.</li><li>3. El sistema confirma Rut de usuario y contraseña.</li><li>4. El sistema muestra interfaces dependiendo del usuario.</li></ol>
Extensiones (flujo alternativo):	<p>3.a Rut de Usuario o contraseña incorrecta:</p> <ol style="list-style-type: none"><li>1. El sistema muestra mensaje "Rut de Usuario incorrecto" o "contraseña incorrecta" se vuelve al paso 1.</li></ol>



### 4.1.2.2 Diagramas de Secuencia

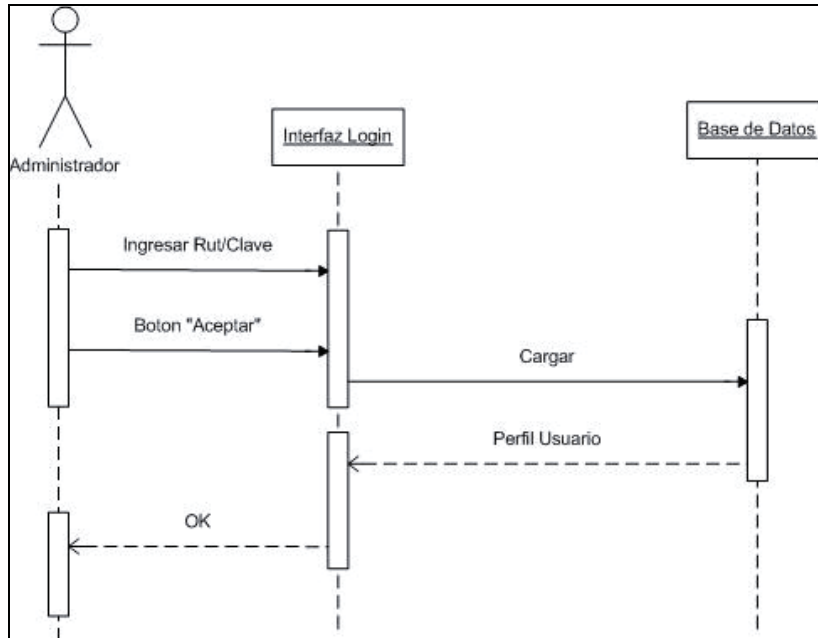


Ilustración 4.2 Diagrama de Secuencia: Caso de Uso Loguearse

## 4.2 Segunda Iteración: Configuración del Sistema

### 4.2.1 Análisis

#### 4.2.1.1 Diagramas de Casos de Uso

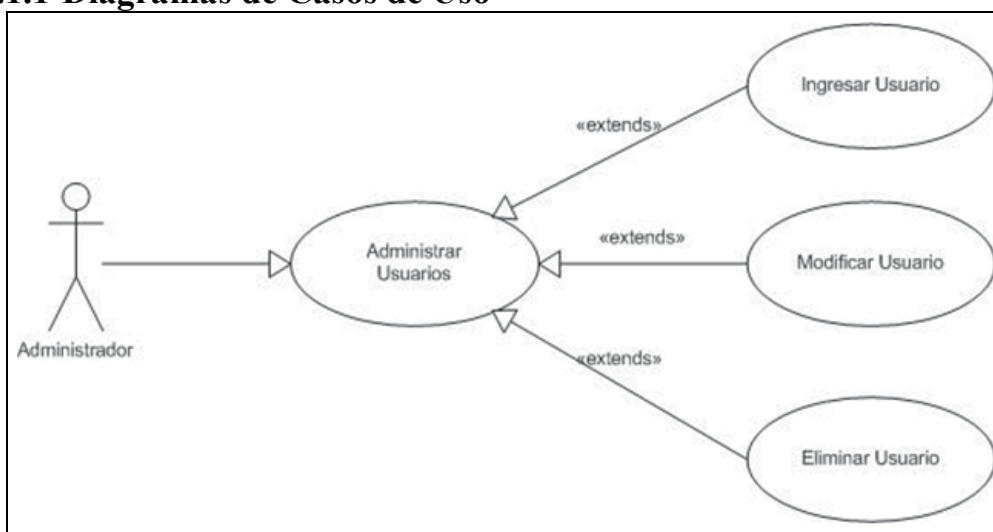


Ilustración 4.3 Caso de Uso: Administrar Usuarios

## 4.2.2 Diseño

### 4.2.2.1 Especificación de Casos de Uso

**Tabla 4-2 Caso de Uso: Ingresar Usuario**

<b>Caso de Uso:</b>	<b>Ingresar Usuario</b>
Objetivo:	Ingresar de manera efectiva un nuevo Usuario en el sistema.
Actor Principal:	Administrador.
Actor Secundario:	No hay.
Precondiciones:	El Administrador se ha logueado en el sistema.
Poscondiciones:	El nuevo Usuario ha sido ingresado en el sistema.
Escenario Principal (flujo básico):	<ol style="list-style-type: none"> <li>1. Administrador selecciona la opción “Ingresar Usuario”.</li> <li>2. Sistema despliega formulario para ser llenado con los datos del Usuario y solicita el perfil del nuevo usuario.</li> <li>3. Administrador ingresa datos necesarios para el registro de un nuevo usuario en el sistema, además de su perfil correspondiente.</li> <li>4. Sistema muestra datos del Usuario ingresado y pide confirmación.</li> <li>5. Administrador confirma el ingreso.</li> </ol>
Extensiones (flujo alternativo):	<p>5.a Administrador no confirma el ingreso.</p> <ol style="list-style-type: none"> <li>1. Se cancela la operación.</li> </ol>

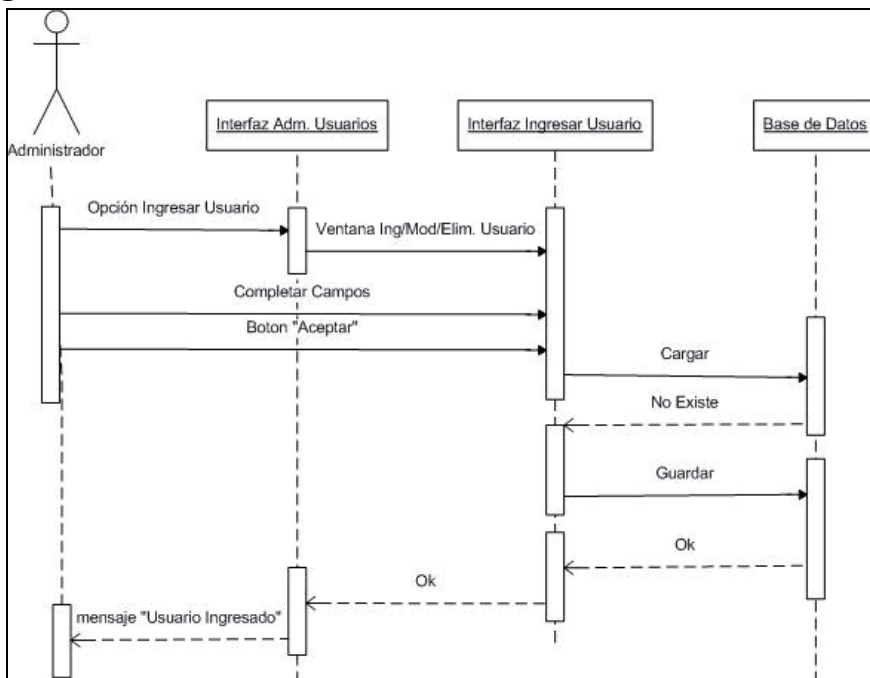
Tabla 4-3 Caso de Uso Modificar Usuario

Caso de Uso:	Modificar Usuario
Objetivo:	Modificar de manera efectiva los datos de un Usuario almacenado en el sistema.
Actor Principal:	Administrador.
Actor Secundario:	No hay.
Precondiciones:	El Administrador se ha logueado en el sistema, el Usuario debe estar registrado en el sistema.
Poscondiciones:	Los nuevos datos del Usuario son almacenados por el sistema y éste se actualiza.
Escenario Principal (flujo básico):	<ol style="list-style-type: none"> <li>1. Administrador selecciona la opción “Modificar Usuario”.</li> <li>2. Sistema pide ingreso de Rut de Usuario.</li> <li>3. Administrador ingresa el Rut del Usuario a modificar.</li> <li>4. Sistema valida la existencia del Rut ingresado.</li> <li>5. Sistema muestra datos del Usuario asociados al Rut ingresado, los cuales se pueden modificar.</li> <li>6. Administrador actualiza los datos del Usuario que desea modificar y confirma la operación.</li> <li>7. Sistema guarda los cambios realizados y muestra mensaje “Datos Actualizados”.</li> </ol>
Extensiones (flujo alternativo):	<ol style="list-style-type: none"> <li>4.a Rut de Usuario ingresado no existe.               <ol style="list-style-type: none"> <li>1. Sistema muestra mensaje “Rut no registrado”.</li> <li>2. Se vuelve al paso 2.</li> </ol> </li> <li>4.b Rut de Usuario ingresado no es válido.               <ol style="list-style-type: none"> <li>1. Sistema muestra mensaje “Rut no válido”.</li> <li>2. Se vuelve al paso 2.</li> </ol> </li> <li>6.a Administrador no confirma la actualización.               <ol style="list-style-type: none"> <li>1. Se cancela la operación.</li> </ol> </li> </ol>

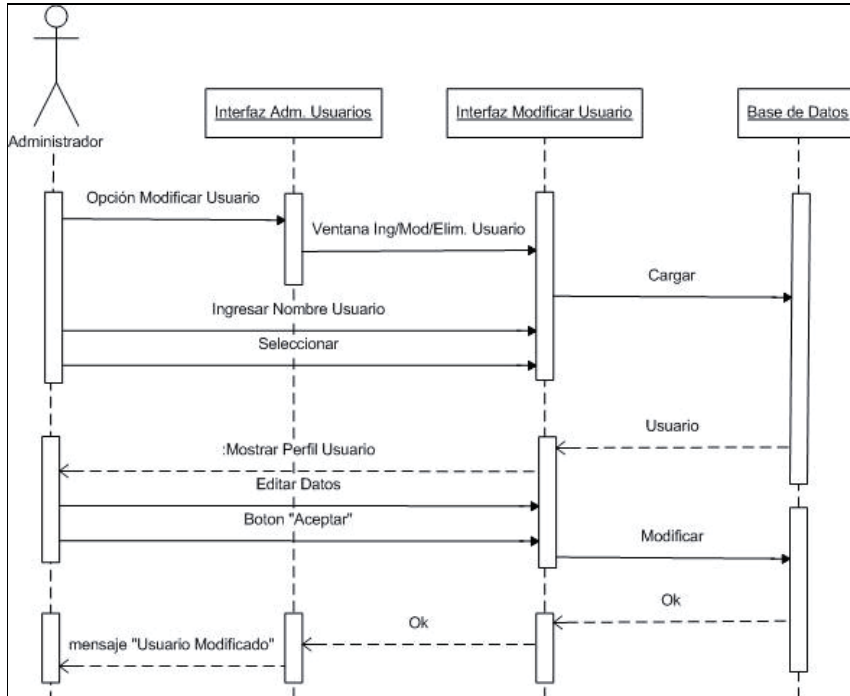
**Tabla 4-4 Caso de Uso: Eliminar Usuario**

<b>Caso de Uso:</b>	<b>Eliminar Usuario</b>
Objetivo:	Eliminar de manera efectiva a un Usuario del sistema.
Actor Principal:	Administrador.
Actor Secundario:	No hay.
Precondiciones:	El Administrador se ha logueado en el sistema, el Usuario debe estar registrado en el sistema.
Poscondiciones:	El Usuario ha sido eliminado del sistema.
Escenario Principal (flujo básico):	<ol style="list-style-type: none"> <li>Administrador selecciona la opción “Eliminar Usuario”.</li> <li>Sistema pide ingreso de Rut de usuario.</li> <li>Administrador ingresa el Rut del Usuario a eliminar.</li> <li>Sistema valida la existencia del Rut ingresado.</li> <li>Sistema muestra datos del Usuario asociados al Rut ingresado y pide confirmación de eliminación.</li> <li>Administrador confirma la operación.</li> <li>Sistema realiza la eliminación y muestra mensaje “Usuario Eliminado”.</li> </ol>
Extensiones (flujo alternativo):	<ol style="list-style-type: none"> <li>4.a Rut de Usuario ingresado no existe.                             <ol style="list-style-type: none"> <li>Sistema muestra mensaje “Rut no registrado”.</li> <li>Se vuelve al paso 2.</li> </ol> </li> <li>4.b Rut de Usuario ingresado no es válido.                             <ol style="list-style-type: none"> <li>Sistema muestra mensaje “Rut no válido”.</li> <li>Se vuelve al paso 2.</li> </ol> </li> <li>6.a Administrador no confirma la eliminación.                             <ol style="list-style-type: none"> <li>Se cancela la operación.</li> </ol> </li> </ol>

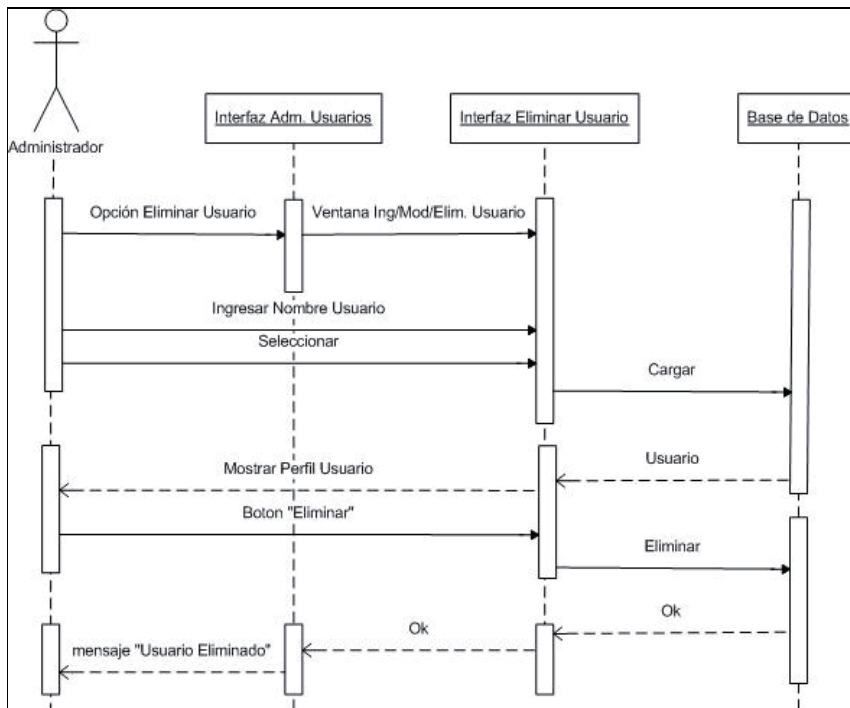
**4.2.2.2 Diagramas de Secuencia**



**Ilustración 4.4: Diagrama de Secuencia para Caso de Uso Ingresar Usuario**



**Ilustración 4.5: Diagrama de Secuencia para Caso de Uso Modificar Usuario**



**Ilustración 4.6: Diagrama de Secuencia para caso de Uso Eliminar Usuario**

### 4.3 Tercera Iteración: Presupuestos y Eventos

#### 4.3.1 Análisis

##### 4.3.1.1 Diagramas de Casos de Uso

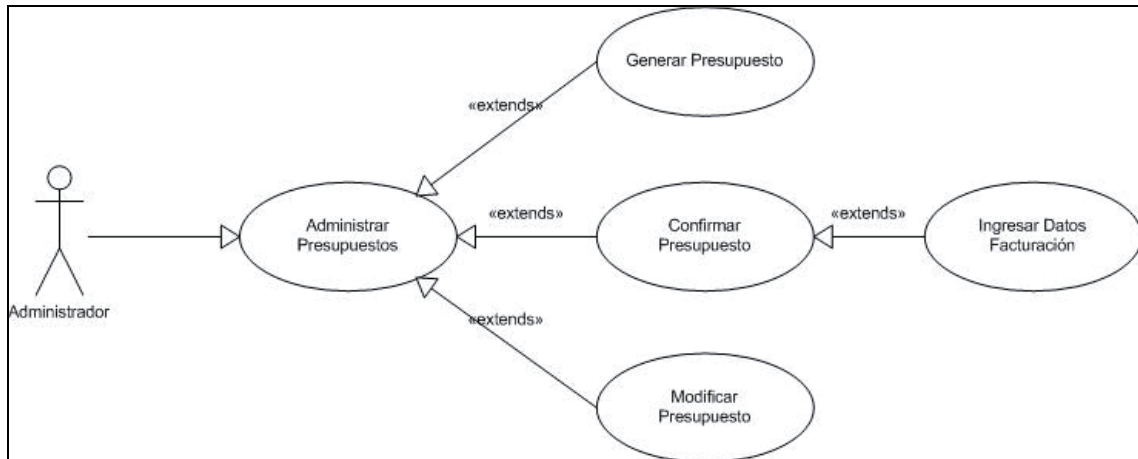


Ilustración 4.7: Caso de Uso Administrar Presupuestos

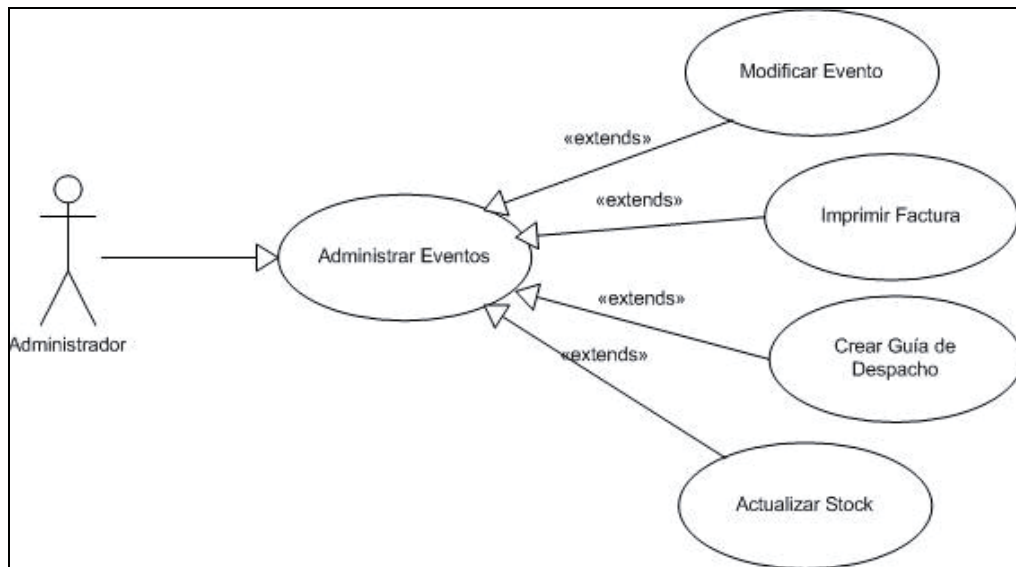


Ilustración 4.8: Caso de Uso Administrar Eventos

### 4.3.2 Diseño

#### 4.3.2.1 Especificación de Casos de Uso

**Tabla 4-5 Caso de Uso Generar Presupuesto**

<b>Caso de Uso:</b>	<b>Generar Presupuesto</b>
Objetivo:	Generar un presupuesto para la solicitud de un cliente.
Actor Principal:	Administrador.
Actor Secundario:	No hay.
Precondiciones:	Debe existir alguna solicitud de un cliente.
Poscondiciones:	Se envía el presupuesto al cliente y es almacenado en la base de datos.
Escenario Principal (flujo básico):	<ol style="list-style-type: none"> <li>1. Caso de uso comienza cuando el Administrador selecciona la opción Generar Presupuesto.</li> <li>2. El sistema muestra una lista de los productos y servicios prestados por la empresa.</li> <li>3. Administrador selecciona los productos que forman parte de la solicitud escogida e ingresa cantidad y precio de cada producto.</li> <li>4. El Sistema pide la confirmación de los productos seleccionados.</li> <li>5. El Administrador confirma la selección.</li> <li>6. El sistema pide ingresar los datos relativos al cliente y fecha del evento.</li> <li>7. El Administrador ingresa los datos solicitados.</li> <li>8. El sistema pide confirmación de datos ingresados.</li> <li>9. El Administrador confirma ingreso.</li> <li>10. Sistema genera y almacena el evento en la base de datos en el estado de presupuesto.</li> </ol>
Extensiones (flujo alternativo):	<ol style="list-style-type: none"> <li>4.a El Administrador no realiza confirmación.                             <ol style="list-style-type: none"> <li>1. El sistema vuelve al paso 2.</li> </ol> </li> <li>9.a El Administrador no realiza confirmación.                             <ol style="list-style-type: none"> <li>1. El sistema vuelve al paso 6.</li> </ol> </li> </ol>

**Tabla 4-6: Caso de Uso Modificar Presupuesto**

<b>Caso de Uso:</b>	<b>Modificar Presupuesto</b>
Objetivo:	Modificar de manera efectiva los datos de un Presupuesto almacenado en el sistema.
Actor Principal:	Administrador.
Actor Secundario:	No hay.
Precondiciones:	El Administrador se ha logueado en el sistema, el presupuesto debe estar registrado en el sistema.
Poscondiciones:	Los nuevos datos del presupuesto son almacenados por el sistema y este se actualiza.
Escenario Principal (flujo básico):	<ol style="list-style-type: none"> <li>1. Administrador selecciona la opción “Modificar Presupuesto”.</li> <li>2. Sistema pide ingreso de patrón de búsqueda de presupuesto.</li> <li>3. Administrador ingresa el patrón del presupuesto a modificar.</li> <li>4. Sistema valida la existencia del presupuesto ingresado.</li> <li>5. Sistema muestra datos del presupuesto asociado al patrón de búsqueda ingresado y estos se pueden modificar.</li> <li>6. Administrador actualiza los datos del presupuesto que desea modificar y confirma la operación.</li> <li>7. Sistema guarda los cambios realizados y muestra mensaje “Datos Actualizados”.</li> </ol>
Extensiones (flujo alternativo):	<ol style="list-style-type: none"> <li>4.a El presupuesto no existe según patrón de búsqueda.                             <ol style="list-style-type: none"> <li>1. El sistema vuelve al paso 2.</li> </ol> </li> <li>6.a Administrador no confirma la actualización.                             <ol style="list-style-type: none"> <li>1. Se cancela la operación.</li> </ol> </li> </ol>



**Tabla 4-7: Caso de Uso Confirmar Presupuesto**

<b>Caso de Uso:</b>	<b>Confirmar Presupuesto</b>
Objetivo:	Dar aceptación a un presupuesto almacenado.
Actor Principal:	Administrador.
Actor Secundario:	No hay.
Precondiciones:	Debe estar el presupuesto almacenado.
Poscondiciones:	El presupuesto cambia a estado aceptado .
Escenario Principal (flujo básico):	<ol style="list-style-type: none"> <li>1. Caso de uso comienza cuando el Administrador selecciona la opción Confirmar Presupuesto.</li> <li>2. El sistema pide ingreso de patrones de búsqueda.</li> <li>3. Administrador ingresa patrón.</li> <li>4. El sistema valida la existencia del presupuesto.</li> <li>5. El sistema muestra datos del presupuesto y espera confirmación de presupuesto.</li> <li>6. El Administrador confirma presupuesto.</li> <li>7. El sistema cambia estado de presupuesto a “aceptado” y muestra mensaje “Presupuesto Confirmado”.</li> </ol>
Extensiones (flujo alternativo):	<ol style="list-style-type: none"> <li>4.a El presupuesto no existe según patrón de búsqueda.                             <ol style="list-style-type: none"> <li>1. El sistema vuelve al paso 2.</li> </ol> </li> <li>6.a No hay disponibilidad de los recursos                             <ol style="list-style-type: none"> <li>1. El sistema despliega un mensaje indicando los recursos que no se encontraran disponibles para la fecha indicada en la que se realizará el evento.</li> <li>2. Se vuelve al punto 2.</li> </ol> </li> </ol>

**Tabla 4-8 Caso de Uso Modificar Evento**

<b>Caso de Uso:</b>	<b>Modificar Evento.</b>
Objetivo:	Modificación del detalle de un evento a realizarse.
Actor Principal:	Administrador.
Actor Secundario:	No hay.
Precondiciones:	El Administrador se debe haber logueado en el sistema. Se debe haber almacenado una venta de un servicio en una fecha determinada.
Poscondiciones:	Modificación y almacenamiento del detalle de un evento a realizarse.
Escenario Principal (flujo básico):	<ol style="list-style-type: none"> <li>1. El caso de uso comienza después de que el Administrador realice la consulta de un evento específico y decide la modificación del detalle del evento.</li> <li>2. El sistema muestra una lista con los materiales necesarios para la ejecución del evento, la cual es modificable.</li> <li>3. El Administrador realiza los cambios necesarios.</li> <li>4. Sistema pide confirmar la modificación.</li> <li>5. Administrador acepta la modificación.</li> <li>6. El sistema consulta la disponibilidad de los recursos y guarda los cambios.</li> <li>7. Finaliza el caso de uso.</li> </ol>
Extensiones (flujo alternativo):	<ol style="list-style-type: none"> <li>5.a Administrador no confirma operación.                             <ol style="list-style-type: none"> <li>1. Sistema cancela la operación.</li> </ol> </li> <li>6.a No hay disponibilidad de los recursos                             <ol style="list-style-type: none"> <li>1. El sistema despliega un mensaje indicando los recursos que no se encontrarán disponibles para la fecha indicada en la que se realizará el evento.</li> <li>2. Se vuelve al punto 2.</li> </ol> </li> </ol>

**Tabla 4-9: Caso de Uso Imprimir Factura**

<b>Caso de Uso:</b>	<b>Imprimir Factura</b>
Objetivo:	Impresión de una Factura con el detalle de una venta.
Actor Principal:	Administrador
Actor Secundario:	No hay
Precondiciones:	Se debe haber finalizado una venta. El administrador se a logueado en el sistema. Se deben haber ingresado los datos de facturación.
Poscondiciones:	Se debe realizar la impresión de una factura.
Escenario Principal (flujo básico):	<ol style="list-style-type: none"> <li>1. El caso de uso comienza cuando el Administrador decide realizar la impresión de una factura y tras buscar el evento selecciona la opción Imprimir Factura.</li> <li>2. El sistema devuelve el detalle del evento junto con los datos de facturación respectivos.</li> <li>3. Sistema solicita confirmación de impresión de factura.</li> <li>4. El administrador confirma la operación.</li> <li>5. El sistema realiza la impresión de una factura.</li> </ol>
Extensiones (flujo alternativo):	<ol style="list-style-type: none"> <li>1.a No se han ingresado los datos de facturación.                             <ol style="list-style-type: none"> <li>1. El sistema despliega mensaje “Los Datos de Facturación no han sido Ingresados”.</li> <li>2. Se cancela la operación.</li> </ol> </li> <li>3.a El administrador no confirma operación.                             <ol style="list-style-type: none"> <li>1. El sistema cancela operación.</li> </ol> </li> </ol>

### 4.3.2.2 Diagramas de Secuencia

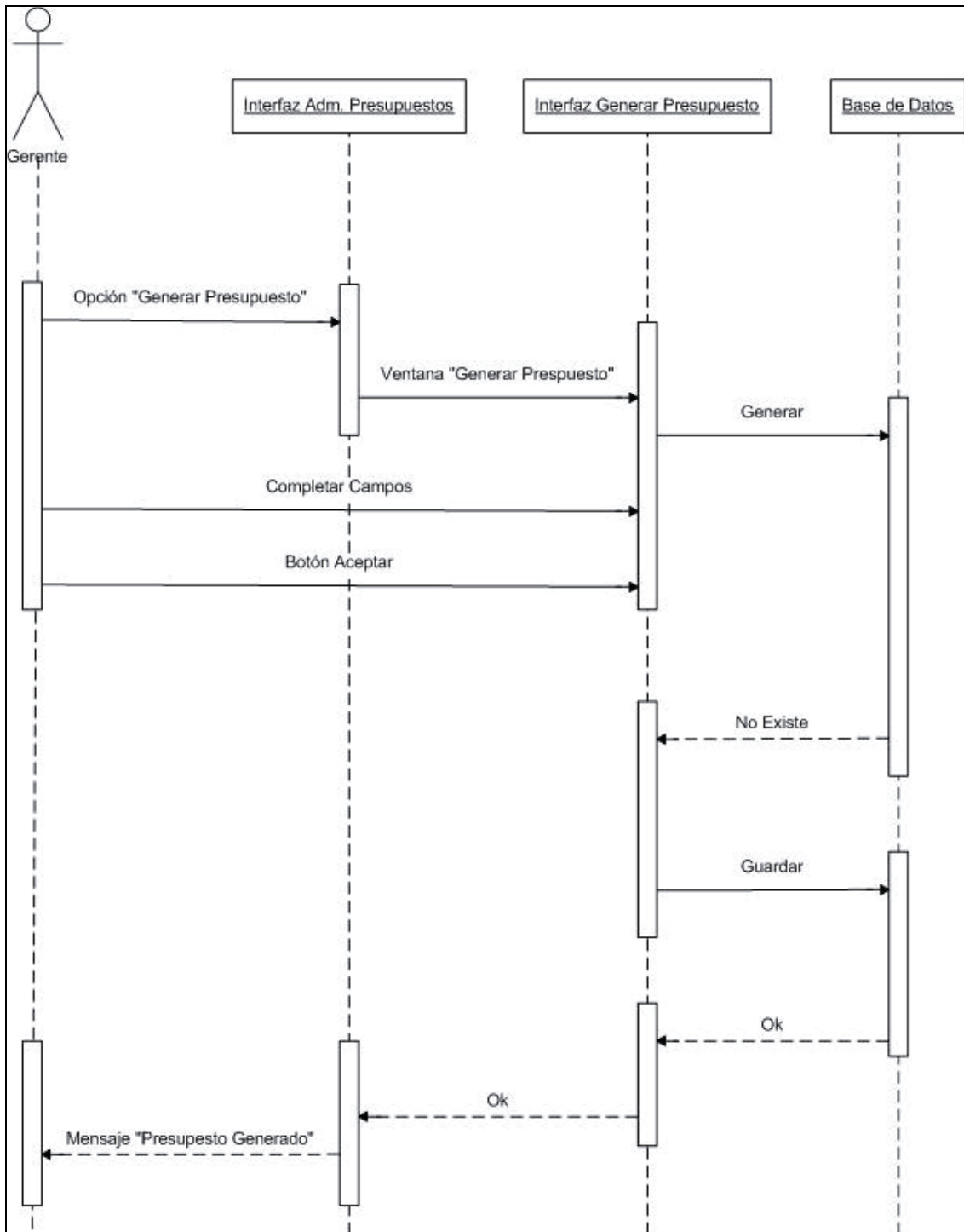


Ilustración 4.9: Diagrama de Secuencia para Caso de Uso Generar Presupuesto

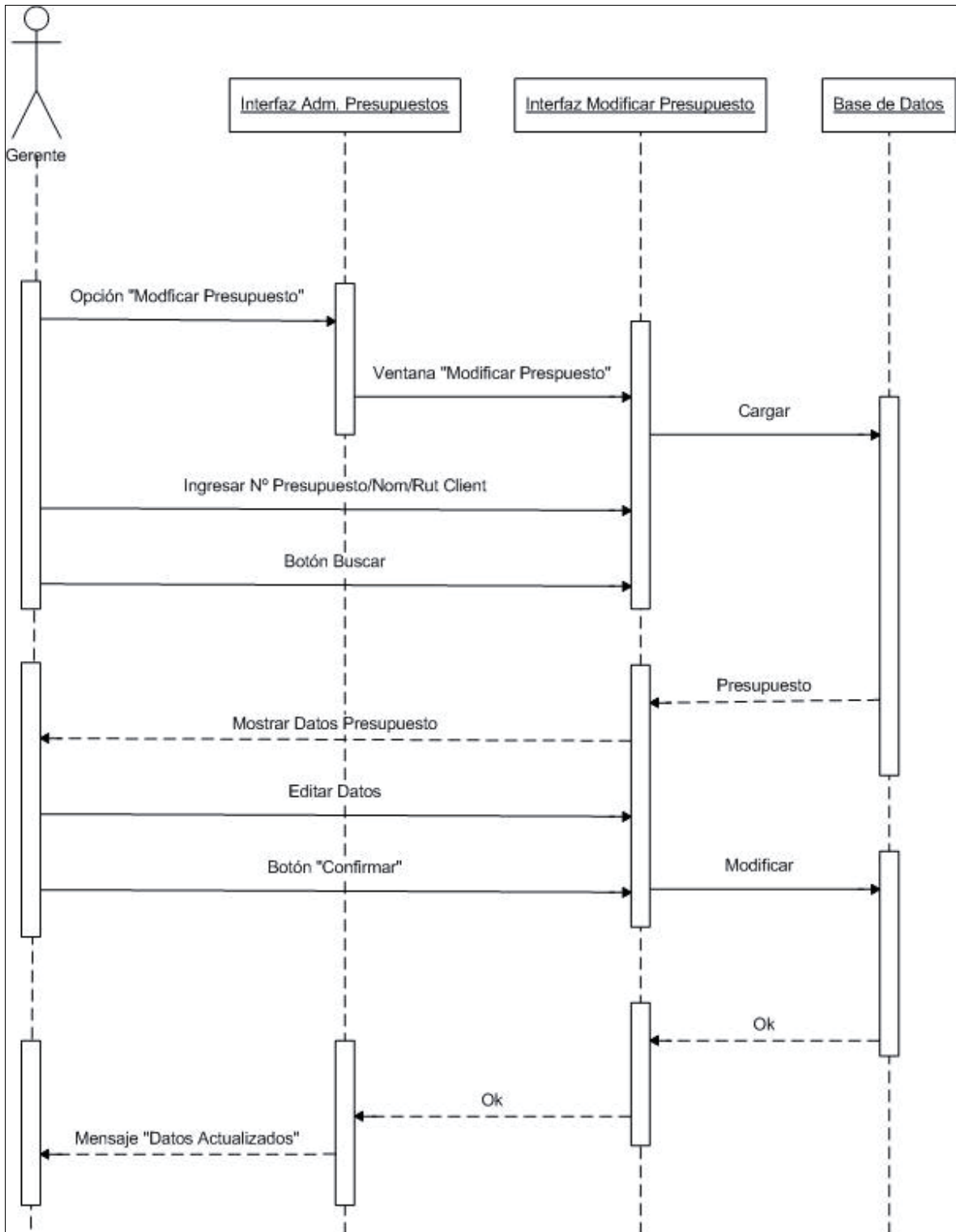


Ilustración 4.10: Diagrama de Secuencia para Caso de Uso Modificar Presupuesto

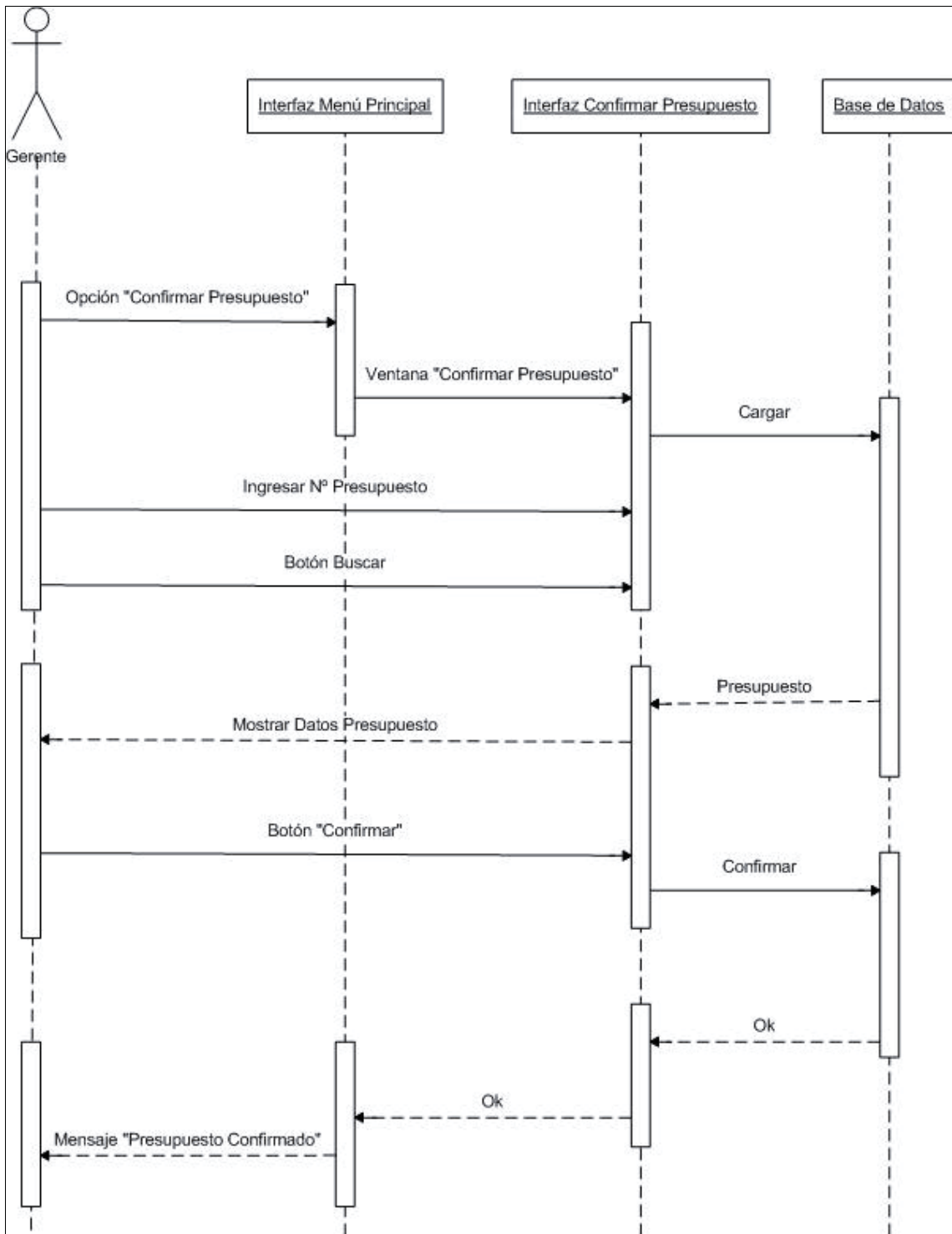


Ilustración 4.11: Diagrama de Secuencia para Caso de Uso Confirmar Presupuesto

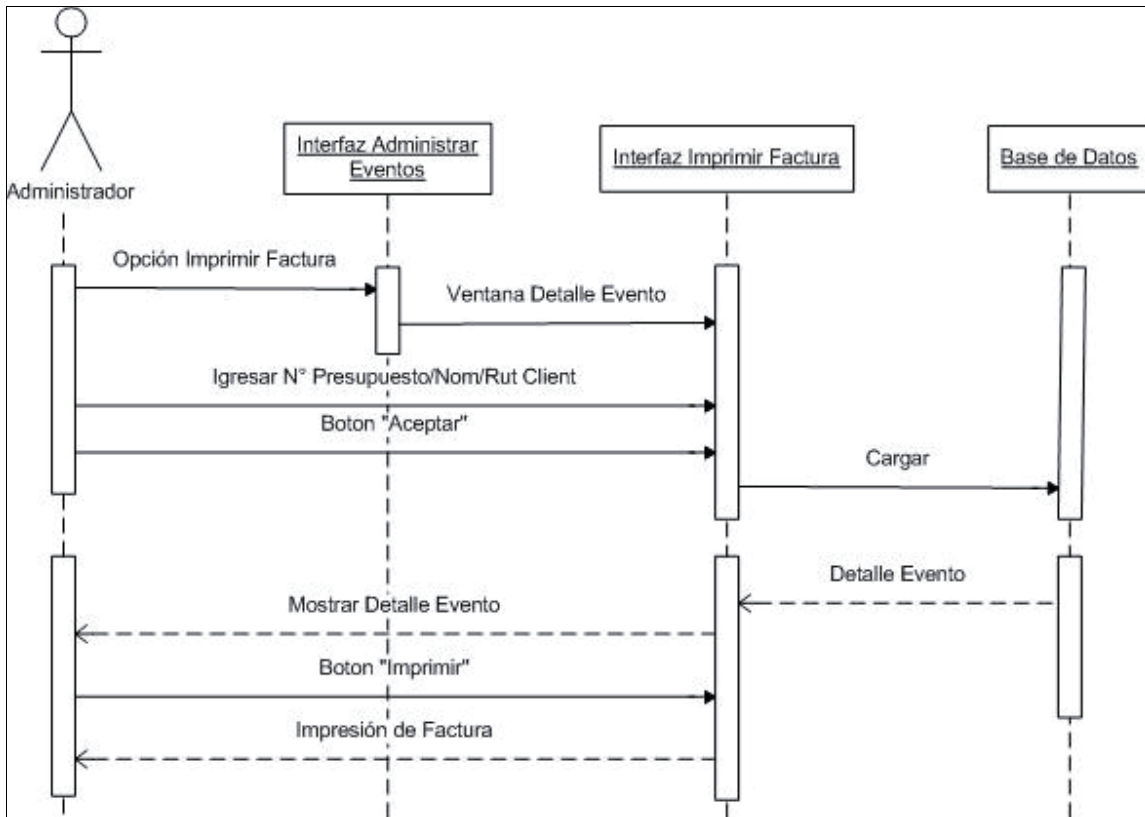


Ilustración 4.12: Diagrama de Secuencia para Caso de Uso Imprimir Factura

## 4.4 Cuarta Iteración: Ventas

### 4.4.1 Análisis

#### 4.4.1.1 Diagramas de Casos de Uso

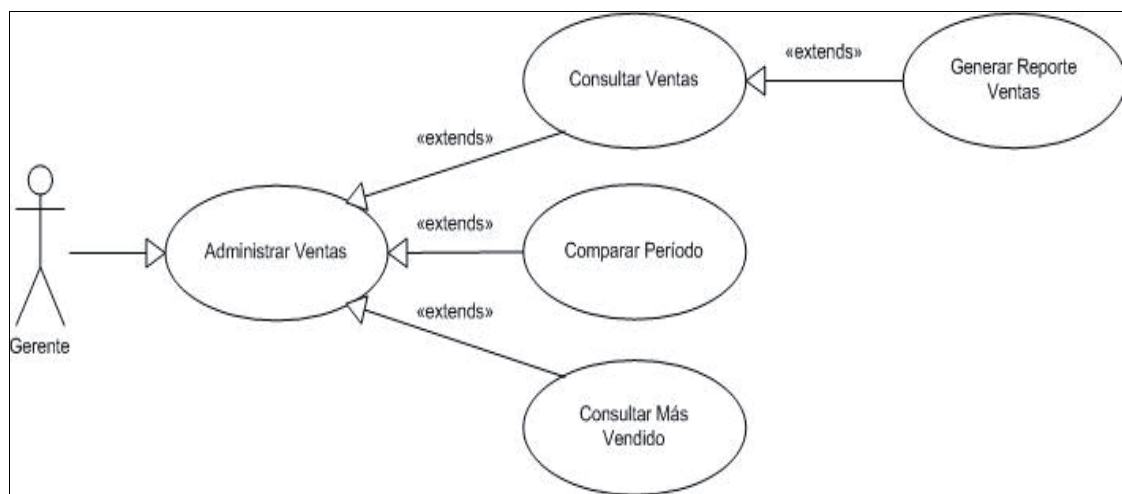


Ilustración 4.13: Caso de Uso Administrar Ventas



## 4.4.2 Diseño

### 4.4.2.1 Especificación de Casos de Uso

**Tabla 4-10: Caso de Uso Consultar Ventas**

Caso de Uso:	Consultar Ventas
Objetivo:	Generar un reporte de las ventas realizadas en un determinado periodo de tiempo.
Actor Principal:	Gerente.
Actor Secundario:	No hay.
Precondiciones:	El Gerente debe estar logueado en el sistema.
Poscondiciones:	Se obtiene reporte de ventas en el período de tiempo señalado por el usuario, el cual puede además ser impreso.
Escenario Principal (flujo básico):	<ol style="list-style-type: none"> <li>1. El caso de uso comienza después que el Gerente se a logueado en el sistema.</li> <li>2. El sistema solicita el ingreso de las fechas entre las cuales se quiere obtener un resumen de las ventas.</li> <li>3. El gerente ingresa el período de tiempo en el cual desea consultar las ventas.</li> <li>4. El sistema procesa la solicitud y despliega un informe de ventas. Se repiten pasos del 3 al 4 hasta que el usuario indique término.</li> <li>5. El Gerente finaliza las consultas.</li> </ol>
Extensiones (flujo alternativo):	<p>3.a El Gerente ingresa período en el cual no se han registrado ventas</p> <ol style="list-style-type: none"> <li>1. El sistema muestra mensaje “No se han percibido ventas durante este periodo” y se vuelve al paso 3.</li> </ol> <p>3.b El Gerente ingresa periodo de tiempo no válido.</p> <ol style="list-style-type: none"> <li>1. El sistema muestra mensaje “Debe ingresar el periodo en el formato fecha inicio – fecha término” y vuelve al paso 3.</li> </ol>

**Tabla 4-11: Caso de Uso Generar Reporte Ventas**

<b>Caso de Uso:</b>	<b>Generar Reporte Ventas.</b>
Objetivo:	Imprime el reporte de las ventas que se han percibido durante un periodo específico de interés para el gerente.
Actor Principal:	Gerente.
Actor Secundario:	No hay.
Precondiciones:	Gerente se a logueado en el sistema y ha decidido generar una impresión del reporte de las ventas percibidas en un cierto periodo de tiempo.
Poscondiciones:	Se obtiene un reporte impreso de ventas que han sido arrojadas por la consulta del Gerente.
Escenario Principal (flujo básico):	<ol style="list-style-type: none"> <li>1. Caso de uso comienza cuando el gerente desea una copia impresa de un reporte de ventas.</li> <li>2. Gerente solicita al sistema la impresión del reporte.</li> <li>3. Sistema solicita confirmar la operación.</li> <li>4. Gerente confirma operación.</li> <li>5. El sistema lleva a cabo la impresión del reporte.</li> <li>6. Gerente obtiene copia impresa del reporte de ventas, finaliza el caso de uso.</li> </ol>
Extensiones (flujo alternativo):	<ol style="list-style-type: none"> <li>3.a Gerente no confirma operación.                             <ol style="list-style-type: none"> <li>1. Sistema cancela la operación, finaliza caso de uso.</li> </ol> </li> </ol>

**Tabla 4-12: Caso de Uso Comparar Períodos**

<b>Caso de Uso:</b>	<b>Comparar Períodos</b>
Objetivo:	Obtener una comparación de las ventas en un mismo periodo de tiempo pero de distintos años.
Actor Principal:	Gerente.
Actor Secundario:	No hay.
Precondiciones:	El Gerente se ha logueado en el sistema.
Poscondiciones:	El Gerente obtiene informe de las ventas percibidas en los respectivos periodos y además muestra el diferencial entre ambos.
Escenario Principal (flujo básico):	<ol style="list-style-type: none"> <li>1. El sistema solicita el ingreso de las fechas entre las cuales se quiere obtener el reporte.</li> <li>2. El gerente ingresa los periodos de tiempo para los cuales desea obtener el reporte.</li> <li>1. El sistema procesa la solicitud y despliega un informe con las ventas percibidas en los dos periodos además de mostrar el respectivo diferencial entre ambos. Se repiten pasos del 1 al 3 hasta que el usuario indique término.</li> <li>3. 4. El Gerente finaliza las consultas.</li> </ol>
Extensiones (flujo alternativo):	<ol style="list-style-type: none"> <li>2.a El Gerente ingresa periodo de tiempo no válido.                             <ol style="list-style-type: none"> <li>1. El sistema muestra mensaje “Debe ingresar el periodo en el formato fecha inicio – fecha término” y vuelve al paso 3.</li> </ol> </li> </ol>

**Tabla 4-13: Caso de Uso Consultar Más Vendido**

Caso de Uso:	Consultar Más Vendido
Objetivo:	Consultar el artículo más solicitado en un cierto periodo de tiempo.
Actor Principal:	Gerente.
Actor Secundario:	No hay.
Precondiciones:	El Gerente se ha logueado en el sistema.
Poscondiciones:	El Gerente obtiene informe de los artículos más vendidos según su categoría en un cierto periodo de tiempo.
Escenario Principal (flujo básico):	<ol style="list-style-type: none"> <li>2. El sistema solicita el ingreso de las fechas entre las cuales se quiere obtener el reporte</li> <li>3. El gerente ingresa el periodo de tiempo en el cual desea obtener el reporte y la categoría.</li> <li>4. El sistema procesa la solicitud y despliega un informe con los artículos más vendidos según su . Se repiten pasos del 1 al 3 hasta que el usuario indique término.</li> <li>4. El Gerente finaliza las consultas.</li> </ol>
Extensiones (flujo alternativo):	<p>2.a El Gerente ingresa periodo de tiempo no válido.</p> <ol style="list-style-type: none"> <li>1. El sistema muestra mensaje “Debe ingresar el periodo en el formato fecha inicio – fecha término” y vuelve al paso 3.</li> </ol>

### 4.4.2.2 Diagramas de Secuencia

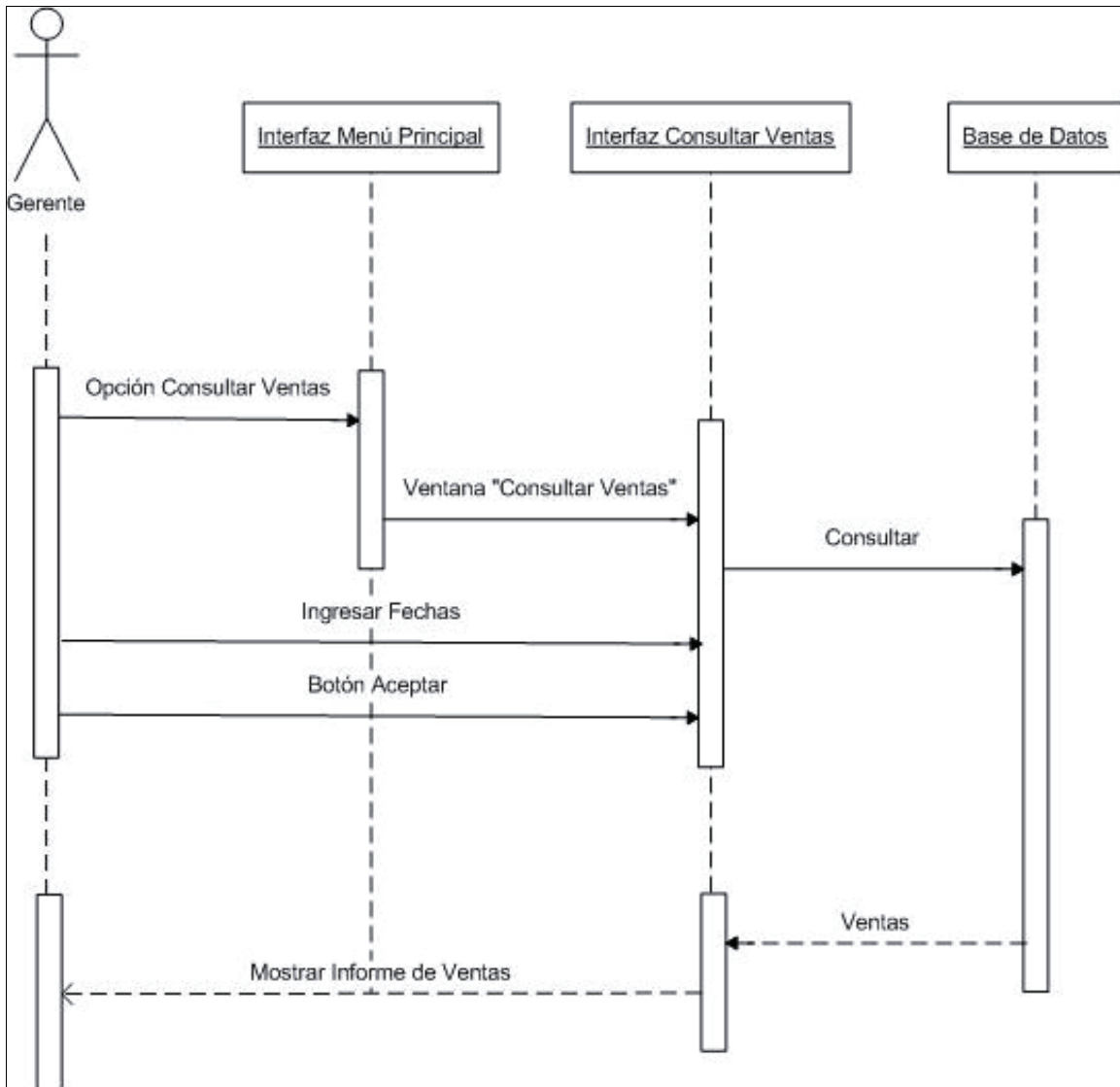


Ilustración 4.14: Diagrama de Secuencia para Caso de Uso Consultar Ventas

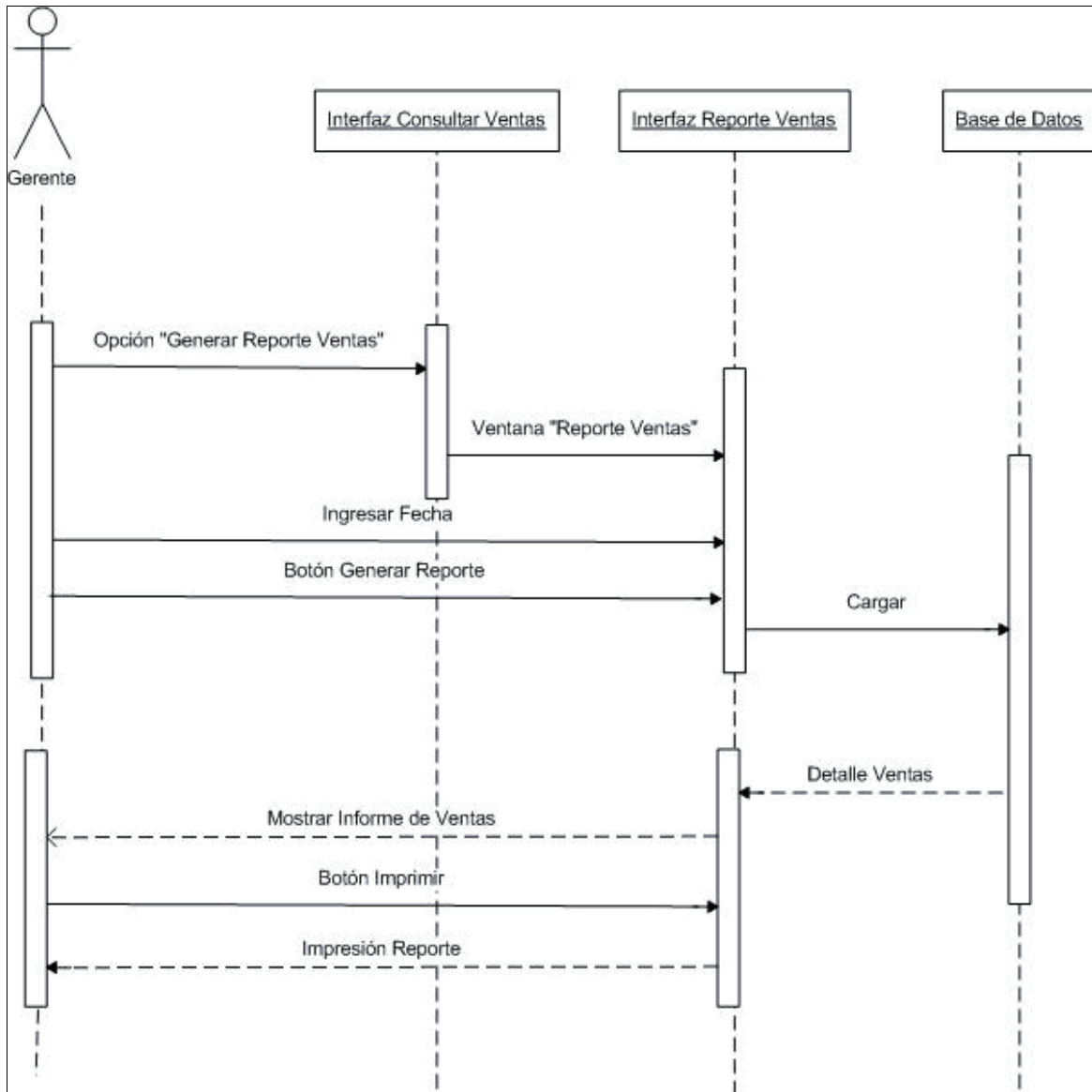
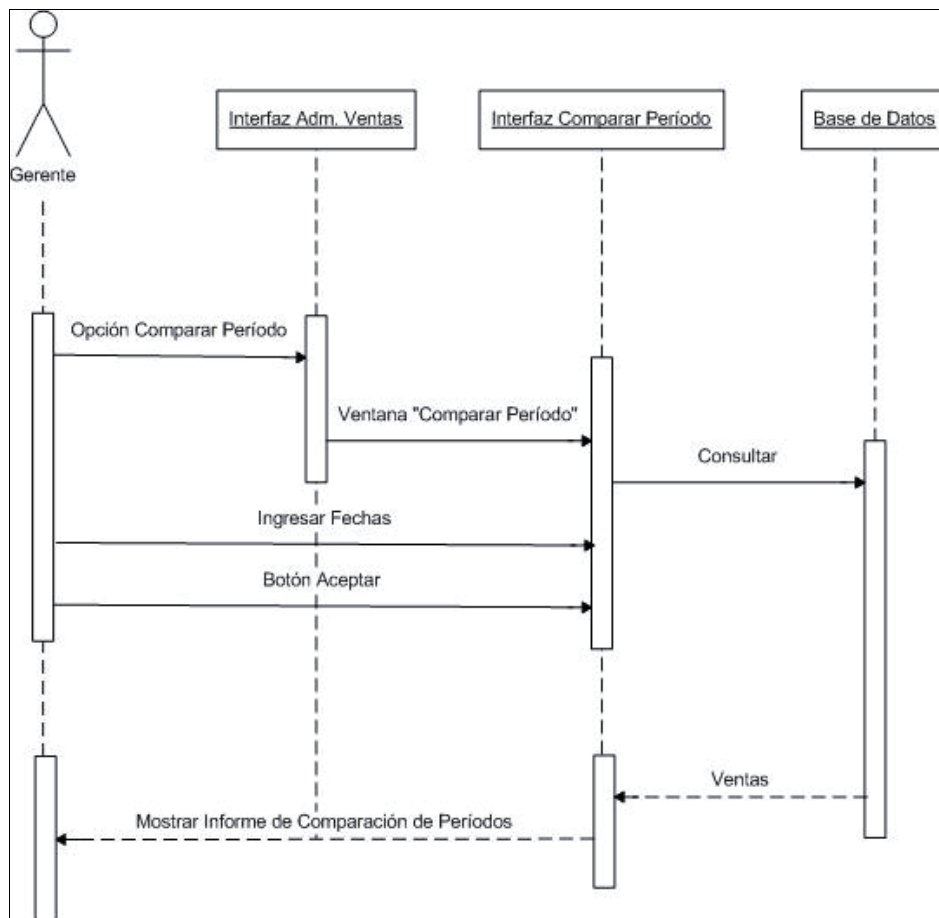


Ilustración 4.15: Diagrama de Secuencia para Caso de Uso Generar Reporte Ventas



**Ilustración 4.16: Diagrama de Secuencia para Caso de Uso Comparar Período**

# Capítulo 5: Desarrollo Fase Construcción

## 5.1 Diagrama de Clases

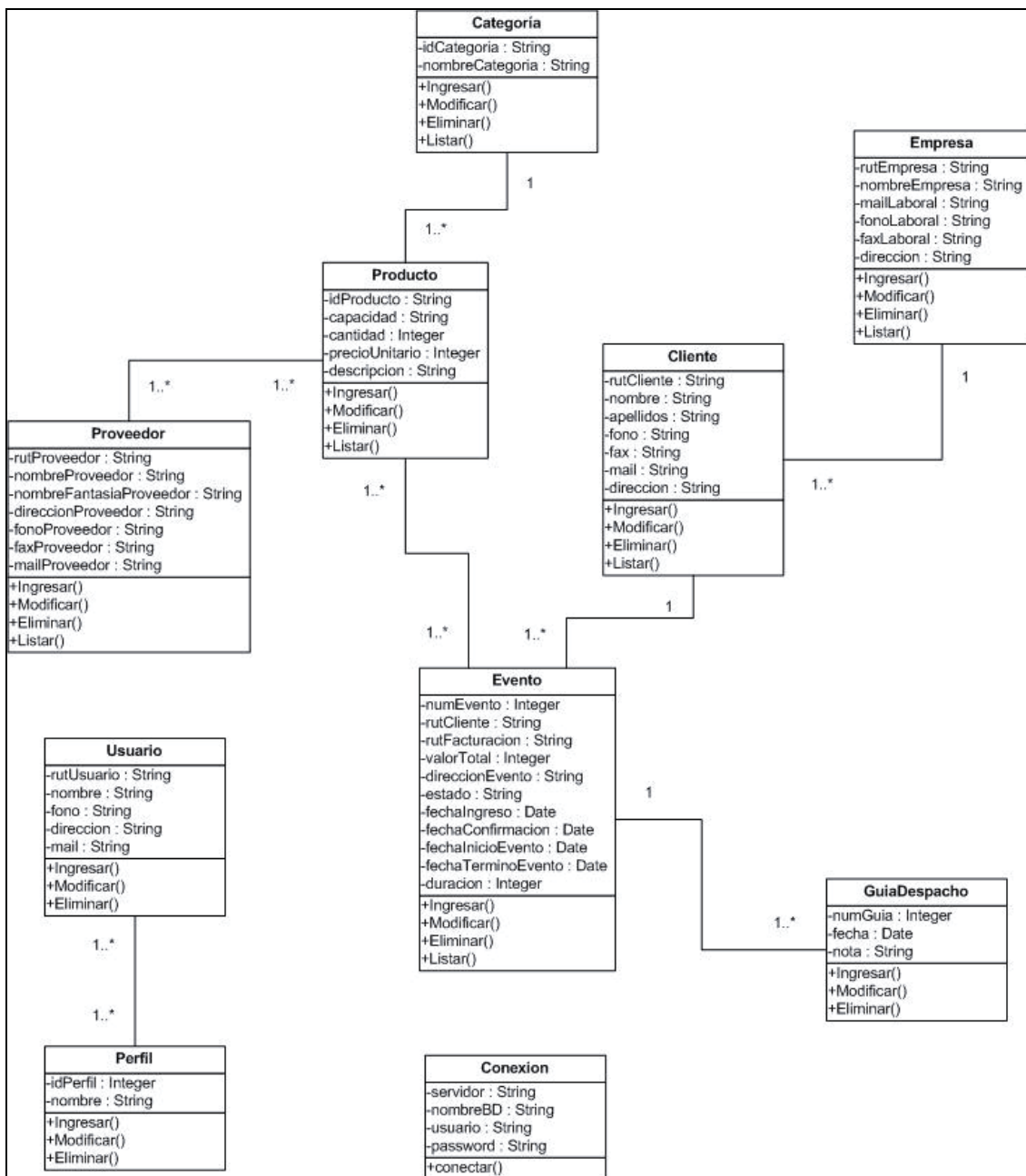


Ilustración 5.1: Diagrama de Clases

## 5.2 Modelo Relacional

Este modelo considera las reglas básicas del modelado relacional. Está compuesto por las tablas que representan los datos y las relaciones entre ellos. Cada una de las tablas posee un nombre exclusivo y una lista de atributos que la definen.

El modelo relacional se caracteriza por ser de fácil entendimiento y utilización por parte del usuario. La información puede ser recuperada o almacenada por medio de consultas que ofrecen una amplia flexibilidad y poder para administrar la información. [5]

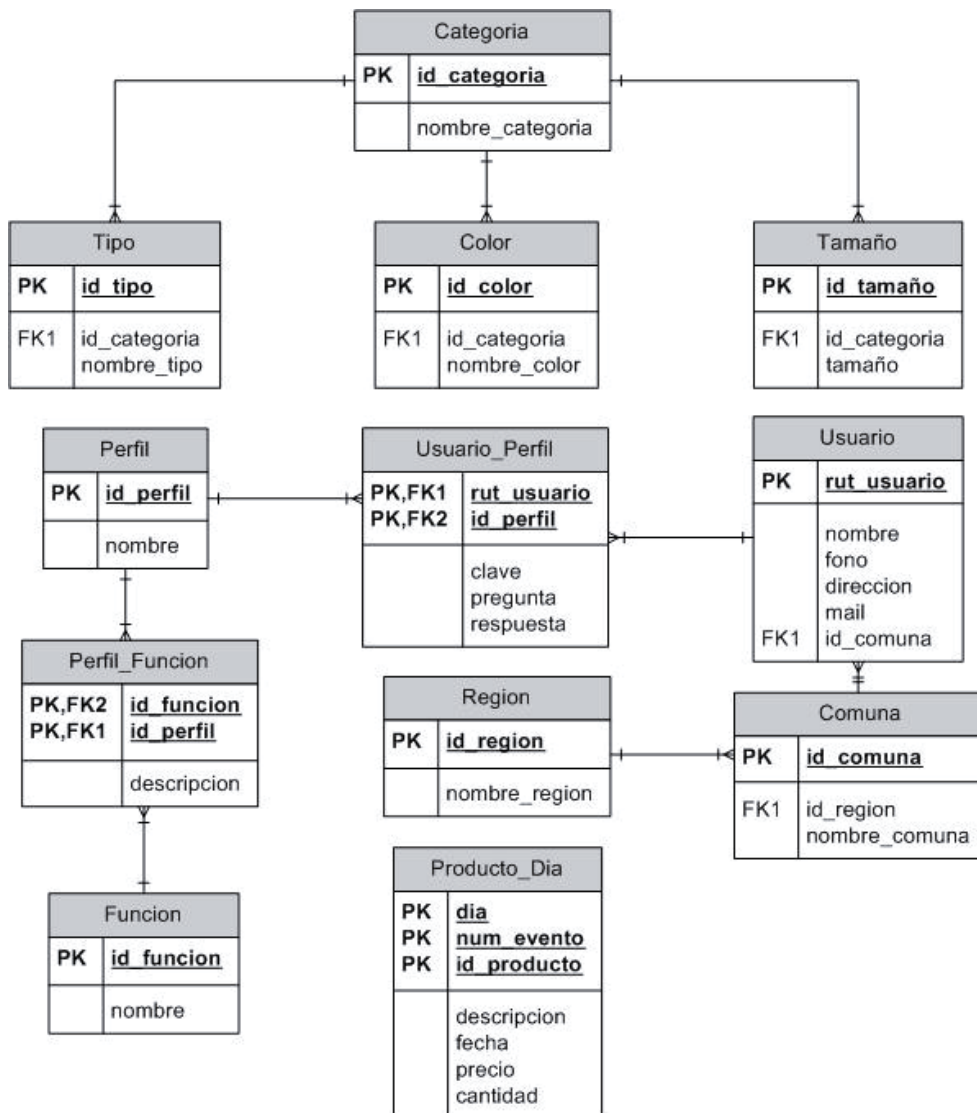


Ilustración 5.2: Modelo Relacional Parte 1



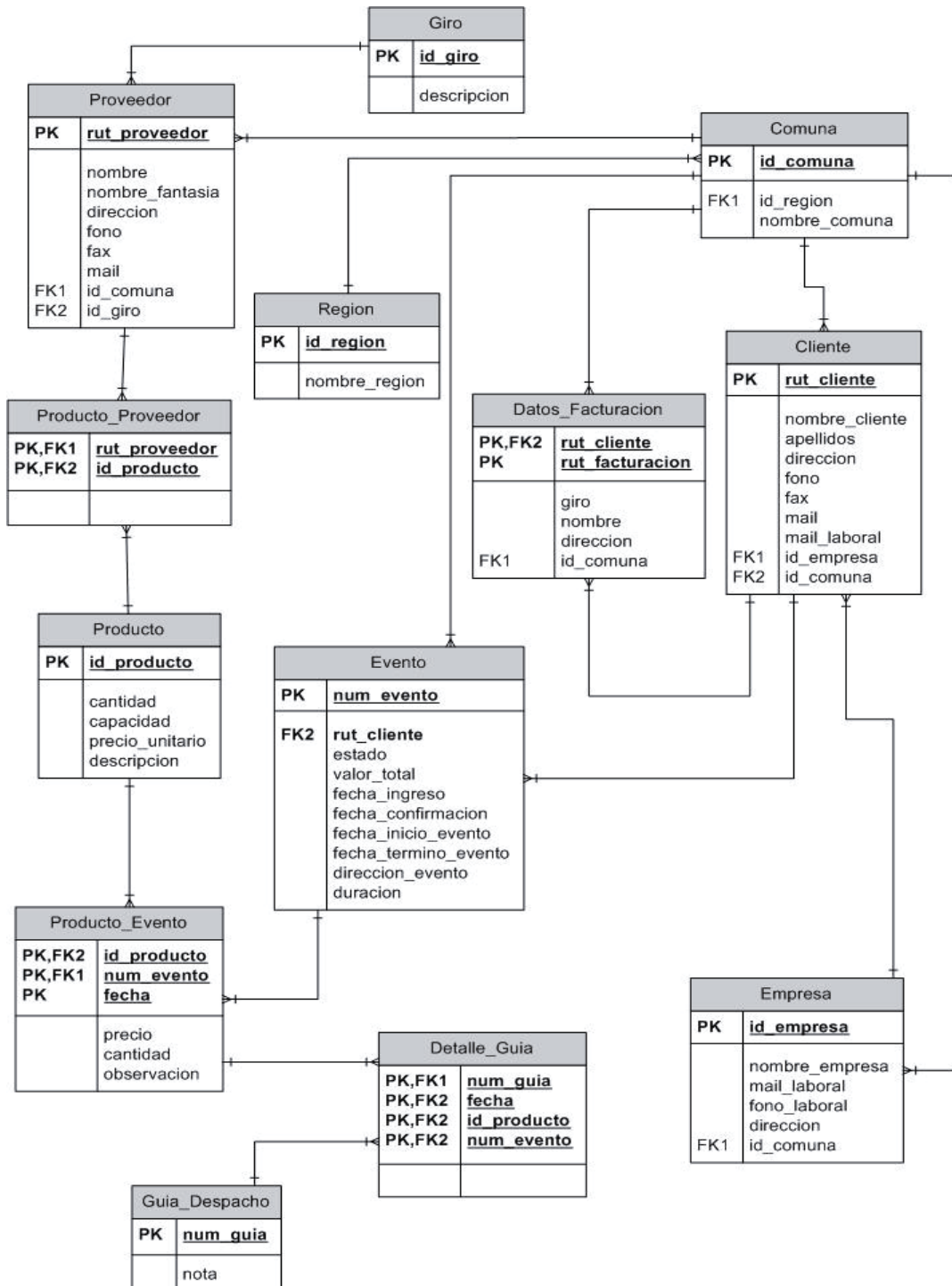


Ilustración 5.3: Modelo Relacional Parte 2

## 5.3 Primera Iteración: Sesión

### 5.3.1 Implementación

#### 5.3.1.1 Interfaces

El diseño de la Interfaz de Usuario (UI), es un proceso iterativo en donde los usuarios interactúan con los diseñadores y prototipos de interfaz para decidir así, las características, organización, apariencia y funcionamiento de la interfaz de usuario del sistema.

Existen 3 actividades esenciales en este proceso:

- **Análisis del usuario:** en este proceso se desarrolla una comprensión de las tareas que el usuario realiza, su entorno de trabajo, como interactúa con el resto de las personas en su trabajo etc.
- **Prototipado del Sistema:** el diseño y desarrollo de la interfaz, en un proceso iterativo. Aunque los usuarios pueden hablar de las facilidades que necesitan en una interfaz, es muy difícil para ellos ser muy específicos hasta que ven algo tangible, por lo tanto se deben generar prototipos del sistema y exponerlos a los usuarios para que así estos puedan guiar el desarrollo de la interfaz.
- **Evaluación de la interfaz:** es importante poseer una tarea de evaluación mas formalizada donde se recopile información sobre las experiencias reales de los usuarios con la interfaz.

En cuanto al prototipado de la UI, idealmente, cuando se está construyendo el prototipo de la interfaz de usuario, se debe adoptar un proceso de prototipado en dos etapas:

- Al principio se deben desarrollar prototipos en papel y mostrárselos a los usuarios finales, esta técnica es un enfoque poco costoso y sorprendentemente efectivo para el desarrollo de prototipos.
- Luego de la tarea anterior, se perfecciona el diseño y se desarrollan prototipos automatizados cada vez más sofisticados, y se ponen a disposición de los usuarios para realizar pruebas.

Visual Basic .Net, incorpora un potente entorno de desarrollo, accede a una gran variedad de objetos reutilizables y a un sistema de desarrollo de interfaces de usuario que permite crear interfaces de una forma rápida y con una gran potencialidad.

Entre los puntos a considerar a la hora de diseñar interfaces de usuario podemos mencionar: la familiaridad del usuario, la mínima sorpresa, la recuperación de errores, la guía del usuario y la diversidad de usuarios ayudan a guiar el diseño de interfaces.

El color se debe utilizar con moderación y de forma uniforme en las interfaces de usuario.

Los objetivos de la evaluación de las interfaces de usuario son obtener una retroalimentación de cómo se puede mejorar el diseño de la UI y evaluar si una interfaz cumple sus requerimientos de usabilidad.

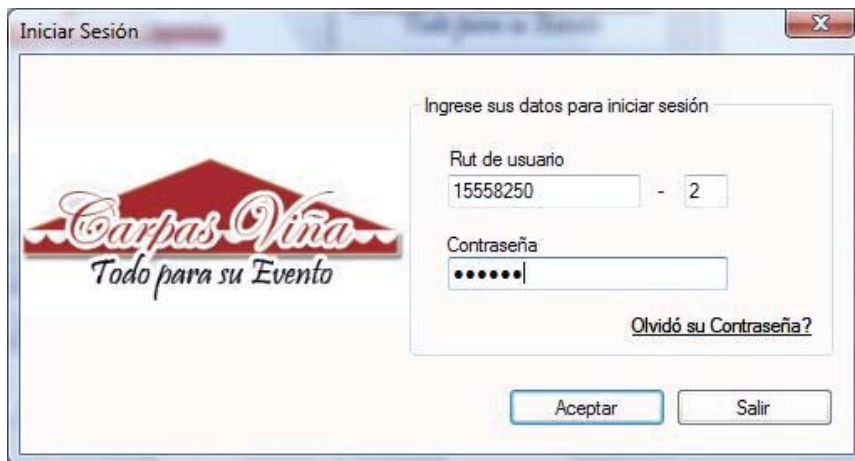


Ilustración 5.4: Interfaz Inicio de Sesión



Ilustración 5.5: Interfaz Menú Principal

### 5.3.2 Pruebas

El proceso de pruebas se realiza con el fin de verificar las funcionalidades del Sistema, asegurando de que se está construyendo el software correcto y que lógicamente cumpla a los requerimientos planteados en las primeras etapas del proceso.

Es importante destacar que las pruebas deben ser de carácter “destrutivo”, ya que el éxito de una prueba dependerá de si ésta encuentra errores que no hayan sido detectados hasta ese momento.

Una vez ejecutadas las pruebas, se evalúan los resultados obtenidos v/s los esperados, en caso de que exista algún error, comienza el proceso de depuración, lo ideal es que este proceso se realice en la etapa de desarrollo del software y no en etapas posteriores como, por ejemplo, la mantención, en donde los costos son mucho más elevados.

Existen dos métodos para probar un producto de software:

- **Prueba de Caja Blanca.**

- Este método de diseño de casos de pruebas se basa en un examen exhaustivo de los detalles procedimentales, en otras palabras, mediante los casos de pruebas asegurar que por lo menos se ejecutaron una vez todos los caminos independientes de cada módulo, todas las decisiones lógicas, bucles, etc. Esto sería lo ideal, pero en la realidad es imposible, aún para pequeños programas, ya que el número de caminos lógicos existente puede ser muy grande, por lo que una solución sería elegir aquellos caminos lógicos más importantes.

- **Prueba de Caja Negra.**

- Se centran en los requisitos funcionales del software, para lo cual se utiliza un conjunto de condiciones de entrada. Este método de prueba intenta encontrar errores en funciones incorrectas o ausentes, errores de interfaz, errores en estructuras de datos o en accesos a bases de datos externas, errores de rendimiento, errores de inicialización y de terminación, etc.

Para realizar el proceso de pruebas se ha decidido utilizar el enfoque de Caja Negra, orientándose en identificar los posibles errores del sistema a nivel funcional.

A continuación se observan los distintos casos de pruebas realizados como parte de éste proceso, correspondientes a la Iteración actual (Sesión):

Tabla 5-1 Caso de Prueba Login Usuario 1

<b>Caso de Prueba Login Usuario 1</b>	
Especificaciones	Ingresar rut de usuario y contraseña válidos (usuario existente)
Entrada	Rut de usuario, Contraseña
Salida	Interfaz Menú Principal
Condiciones	Ninguna
Datos de Prueba	Rut "15558250-2", Contraseña "123456"

Tabla 5-2: Caso de Prueba Login Usuario 2

<b>Caso de Prueba Login Usuario 2</b>	
Especificaciones	Ingresar rut de usuario y contraseña inválidos (usuario no existente)
Entrada	Rut de usuario, Contraseña
Salida	Mensaje "Datos de Usuario Incorrecto" o "Usuario no registrado"
Condiciones	Ninguna
Datos de Prueba	Rut "15071109-6", Contraseña "123456"

Tabla 5-3: Caso de Prueba Login Usuario 3

<b>Caso de Prueba Login Usuario 3</b>	
Especificaciones	Ingresar con campos en blanco
Entrada	Ninguna
Salida	Mensaje "Debe ingresar todos los campos vacíos"
Condiciones	Ninguna
Datos de Prueba	Rut " ", Contraseña " "

Tabla 5-4: Caso de Prueba Recuperar Contraseña 1

<b>Caso de Prueba Recuperar Contraseña 1</b>	
Especificaciones	Ingresar rut de usuario registrado
Entrada	Rut de usuario
Salida	Mensaje "Su contraseña ha sido enviada a su correo electrónico"
Condiciones	Ninguna
Datos de Prueba	Rut "15558250-2"

**Tabla 5-5: Caso de Prueba Recuperar Contraseña 2**

<b>Caso de Prueba Recuperar Contraseña 2</b>	
Especificaciones	Ingresar rut de usuario no registrado
Entrada	Rut de usuario
Salida	Mensaje “Usuario no registrado”
Condiciones	Ninguna
Datos de Prueba	Rut “6680402-K”

### 5.3.3 Resultados de Pruebas

Los resultados obtenidos al llevar a cabo el proceso de pruebas se clasificaron de la siguiente forma:

- **Aprobada:** La prueba determina que el caso experimentado cumple con lo esperado.
- **Aprobada con Observación:** La prueba determina que el caso experimentado cumple con lo esperado, pero con algún tipo de observación a ser tomada en cuenta.
- **Fallida:** La prueba determina que el caso experimentado NO cumple con lo esperado.

**Tabla 5-6: Resultados de Pruebas Iteración Sesión**

<b>Caso de Prueba</b>	<b>Resultados</b>		
	<b>Aprobada</b>	<b>Aprobada con Observación</b>	<b>Fallida</b>
Login Usuario 1	X		
Login Usuario 2		X	
Login Usuario 3	X		
Recuperar Contraseña 1	X		
Recuperar Contraseña 2	X		

#### **Resumen:**

Pruebas Aprobadas: 4 de 5 (80%)

Pruebas Aprobadas con Observación: 1 de 5 (20%)

Pruebas Fallidas: 0 de 5 (0%)

## 5.4 Segunda Iteración: Configuración del Sistema

### 5.4.1 Implementación

#### 5.4.1.1 Interfaces



Ilustración 5.6: Interfaz Administrar Productos

Ilustración 5.7: Interfaz Ingresar Productos

Ilustración 5.8: Interfaz Modificar Producto



Ilustración 5.9: Interfaz Ingresar Cliente

Nombre	Apellidos	Rut
Gonzalo	Sariego G.	155582502
Graciela	Lavín	60858810

Ilustración 5.10: Interfaz Modificar Cliente (Búsqueda)

**Modificar Cliente**

*Carpas Viña*  
Todo para su Evento

Ingrese los datos a modificar:

**Datos del Cliente**

Rut Cliente: 15558250 - 2 Fono: (\*) (032) 3172236

Nombre: (\*) Gonzalo Fax:

Apellidos: (\*) Sariego G. e-mail: gonzalosl@gmail.com

Dirección: Las Camelias 29, Achupallas

Región: V - Valparaíso

Comuna: Viña del Mar

**Datos de La Empresa**

Nombre: GSG Ltda. e-mail laboral: gonzalo@gsg.cl

Dirección: ALgun lado Fono: 312345

Región: V - Valparaíso

Comuna: Viña del Mar

Nota: Los datos señalados con un asterisco (\*) son obligatorios.

Confirmar Limpiar

Volver Salir

**Ilustración 5.11: Interfaz Modificar Cliente**



Ilustración 5.12: Interfaz Administrar Usuarios



Ilustración 5.13: Interfaz Ingresar Usuarios

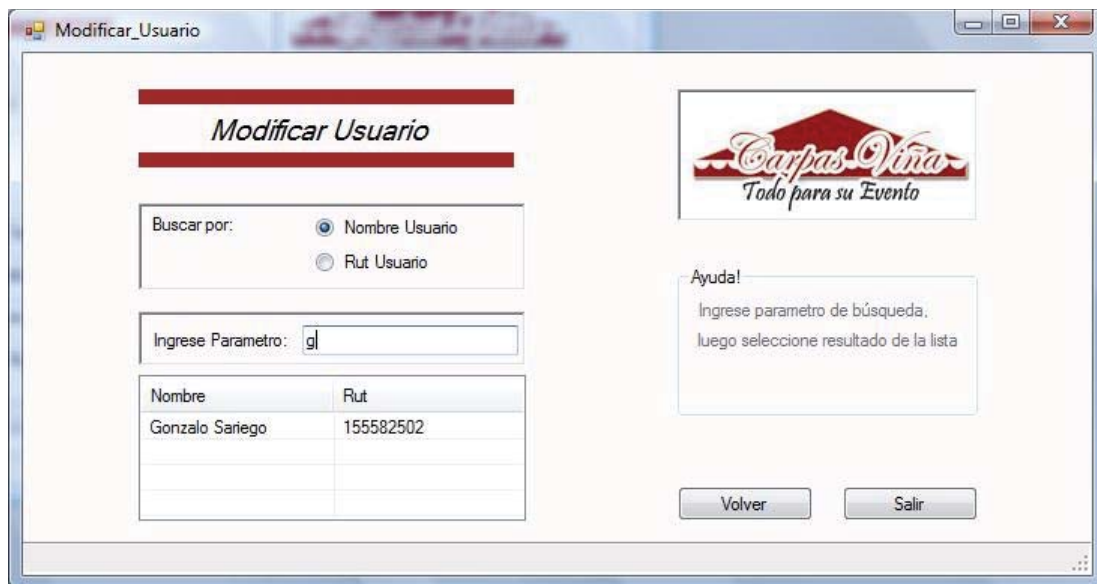


Ilustración 5.14: Interfaz Modificar Usuario (Búsqueda)



Ilustración 5.15: Interfaz Modificar Usuario

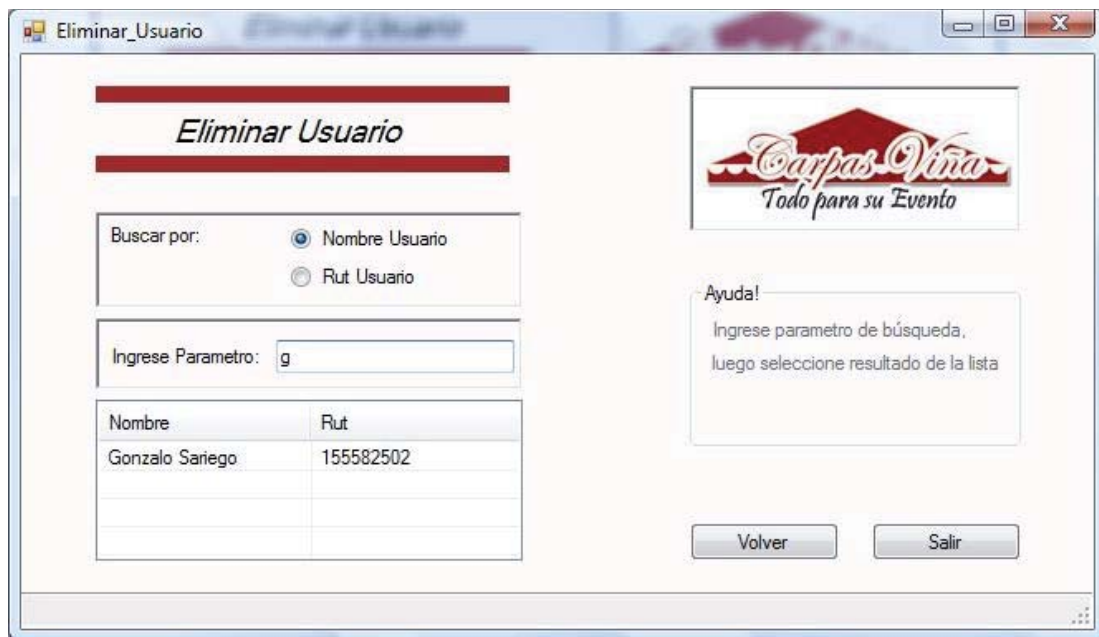


Ilustración 5.16: Interfaz Eliminar Usuario (Búsqueda)

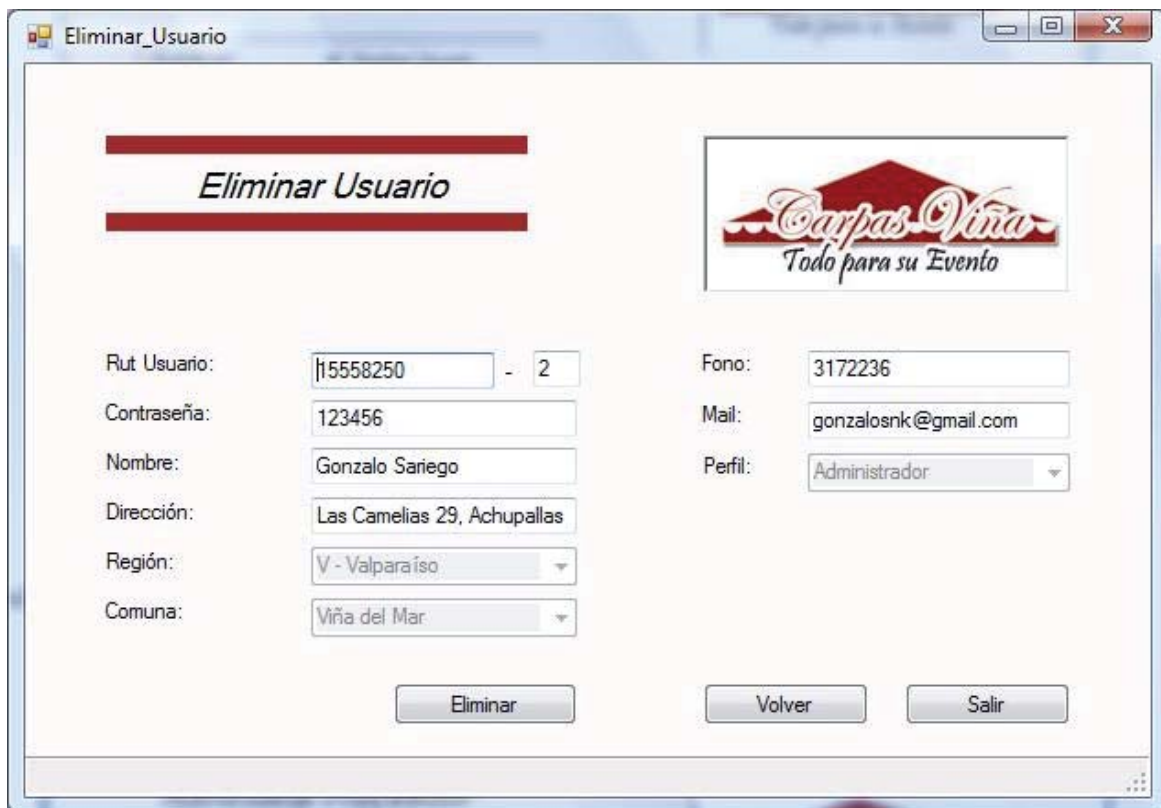


Ilustración 5.17: Interfaz Eliminar Usuario





Ilustración 5.18: Interfaz Administrar Proveedores

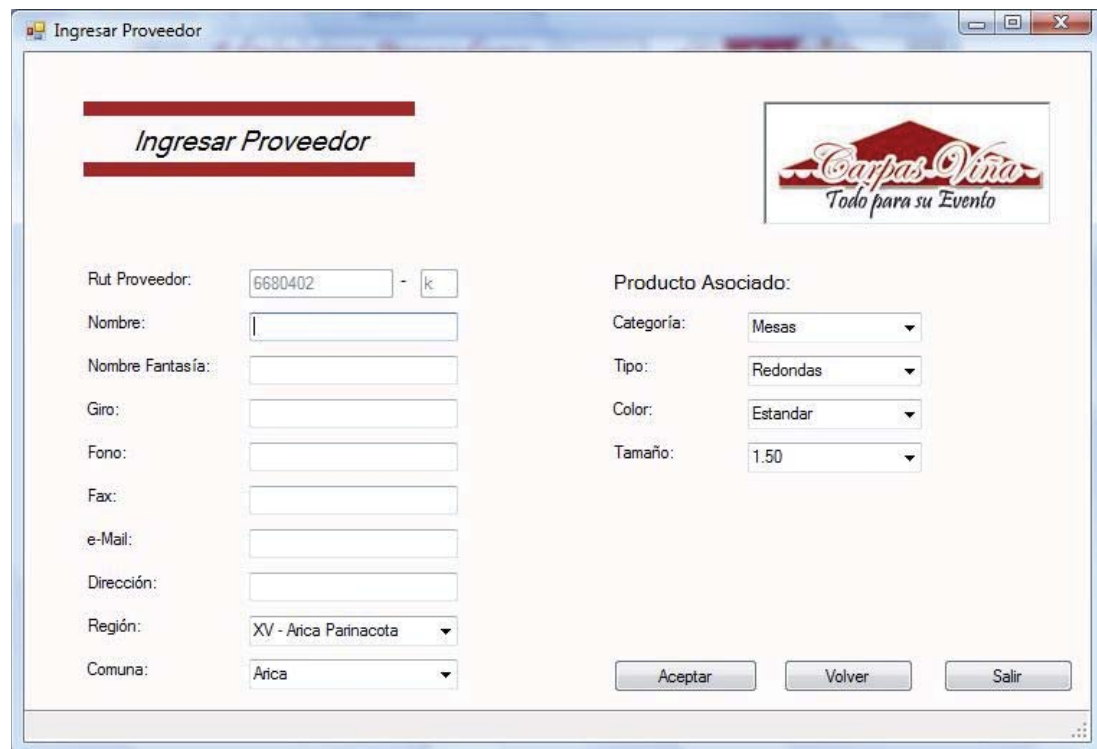
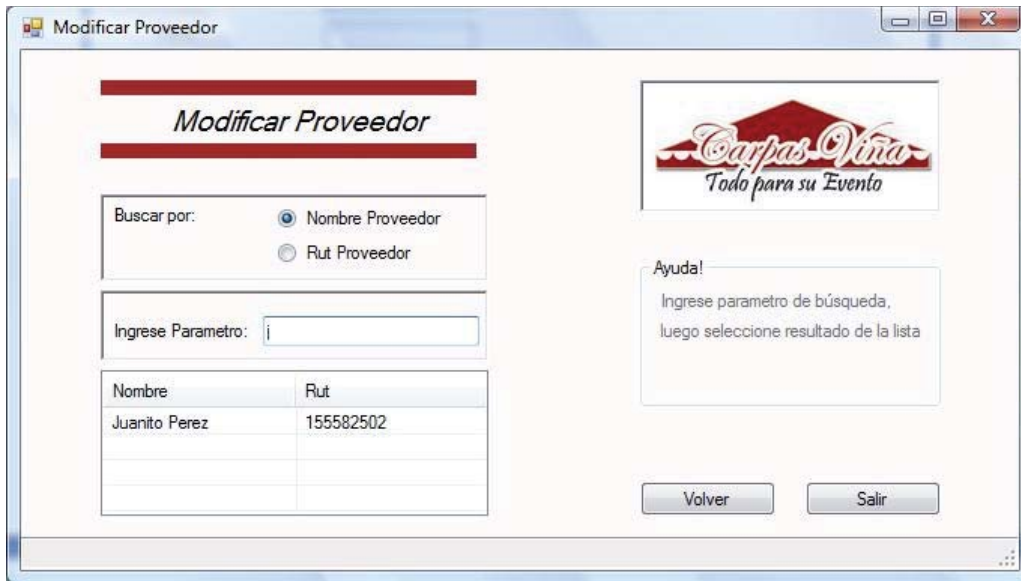
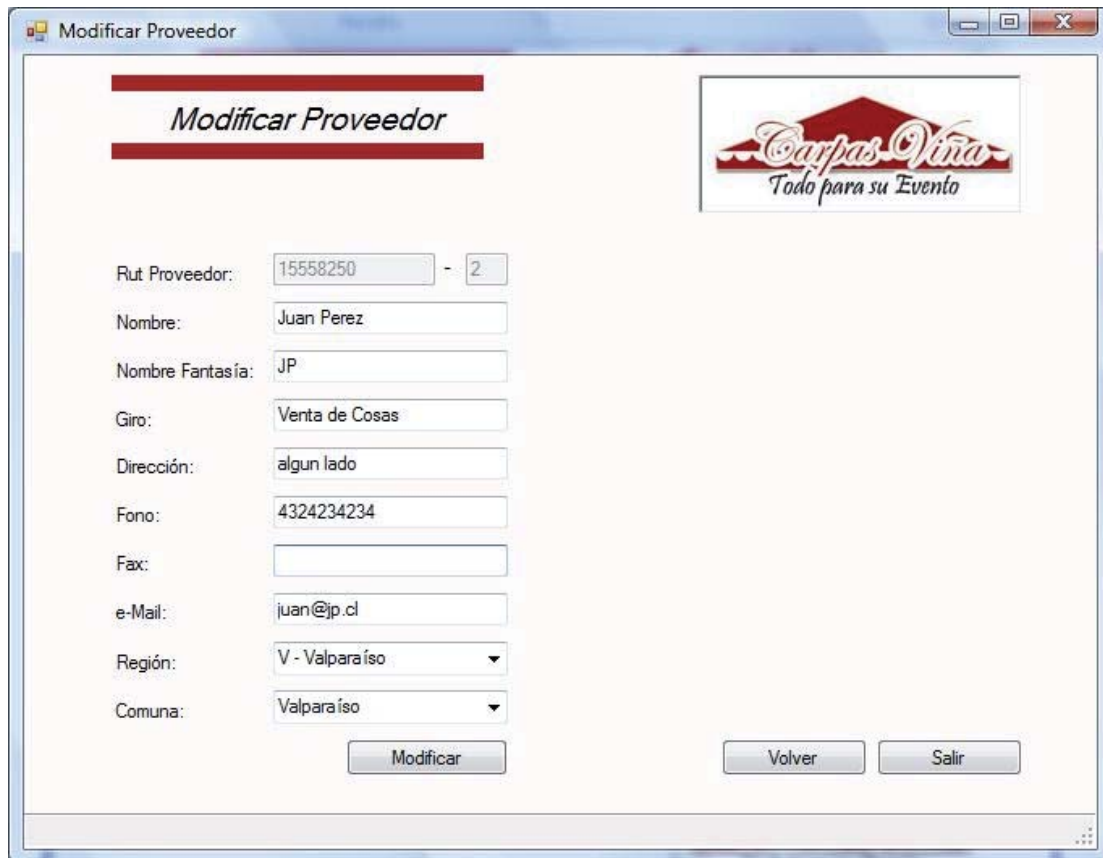


Ilustración 5.19: Interfaz Ingresar Proveedor



**Ilustración 5.20: Interfaz Modificar Proveedor (Búsqueda)**



**Ilustración 5.21: Interfaz Modificar Proveedor**

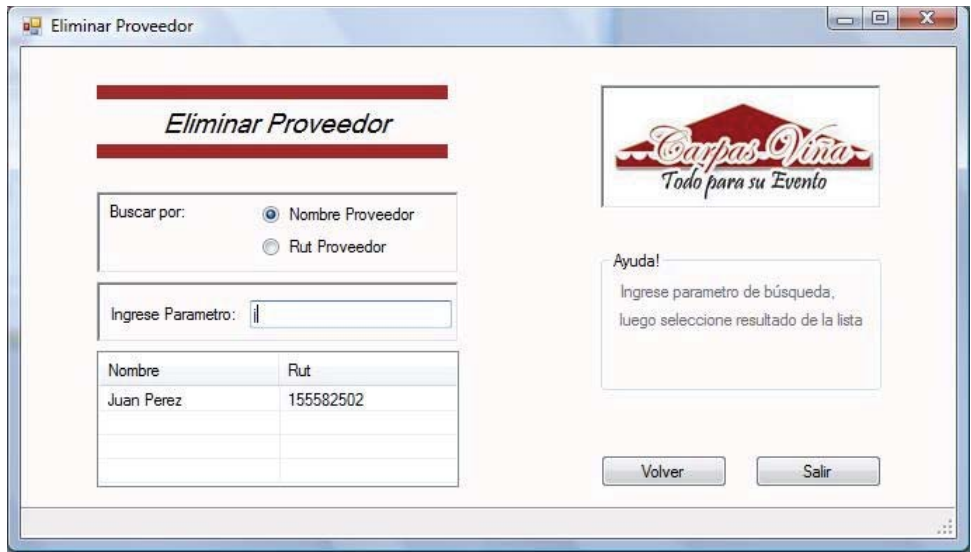


Ilustración 5.22: Interfaz Eliminar Proveedor (Búsqueda)

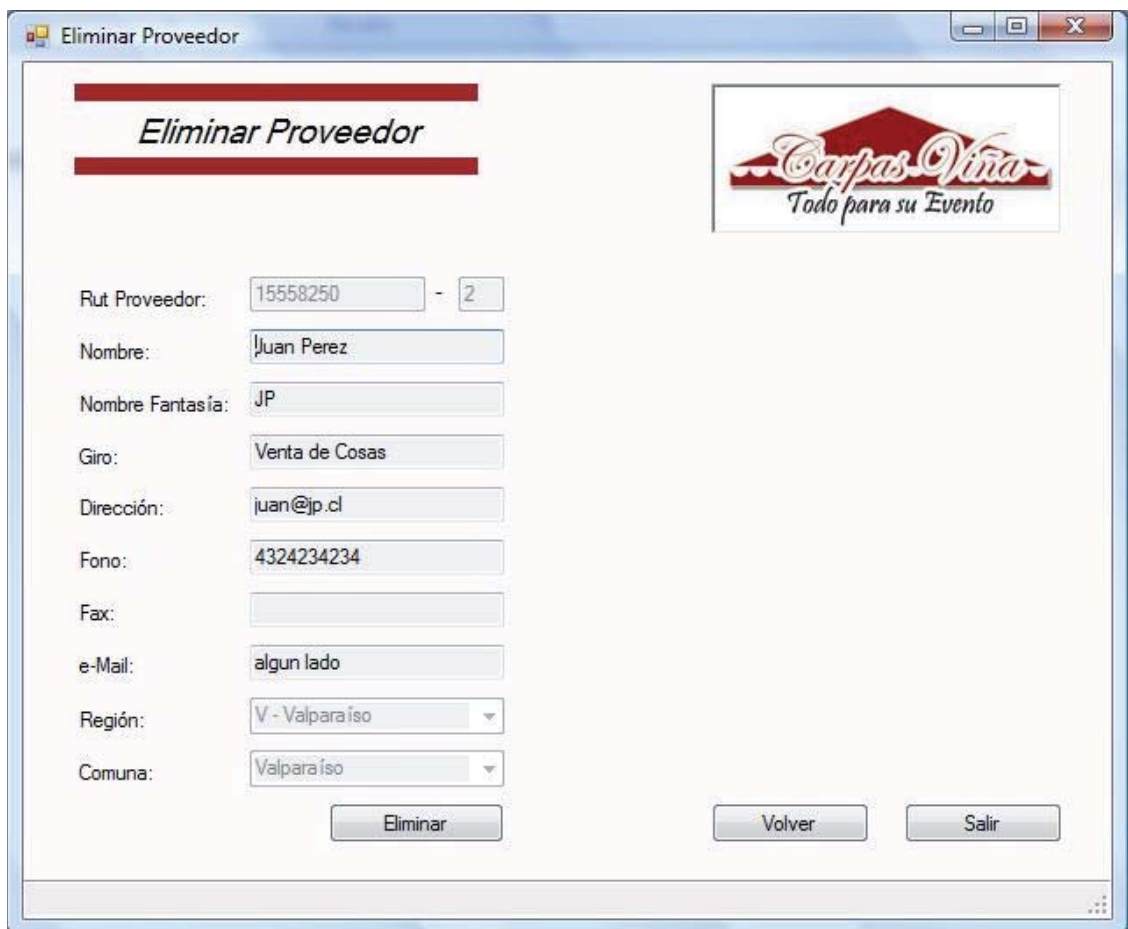


Ilustración 5.23: Interfaz Eliminar Proveedor



## 5.4.2 Pruebas

A continuación se observan los distintos casos de pruebas realizados como parte de éste proceso, correspondientes a la Iteración actual (Configuración del Sistema):

**Tabla 5-7: Caso de Prueba Ingresar Usuario 1**

<b>Caso de Prueba Ingresar Usuario 1</b>	
Especificaciones	Ingresar Usuario no existente ingresando los datos solicitados.
Entrada	Rut usuario, nombre, contraseña, mail.
Salida	Mostrar mensaje “Usuario Ingresado”
Condiciones	Ninguna
Datos de Prueba	Rut “6680402-K”, Nombre “Jorge Lobos”, Contraseña “123456”, mail “info@carpasvina.cl”

**Tabla 5-8: Caso de Prueba Ingresar Usuario 2**

<b>Caso de Prueba Ingresar Usuario 2</b>	
Especificaciones	Ingresar usuario dejando el campo Rut en blanco
Entrada	Nombre usuario, contraseña, mail
Salida	Mensaje “El campo Rut usuario no puede ser nulo”
Condiciones	Ninguna
Datos de Prueba	Rut “”, Nombre “Jorge Lobos”, Contraseña “123456”, mail “info@carpasvina.cl”

**Tabla 5-9: Caso de Prueba Ingresar Usuario 3**

<b>Caso de Prueba Ingresar Usuario 3</b>	
Especificaciones	Ingresar con todo los campos en blanco
Entrada	Ninguna
Salida	Mensaje “Debe ingresar todos los campos obligatorios”
Condiciones	Ninguna
Datos de Prueba	Rut “ ”, Nombre “ ”, Contraseña “ ”, mail “ ”

Tabla 5-10: Caso de Prueba Modificar Usuario 1

<b>Caso de Prueba Modificar Usuario 1</b>	
Especificaciones	Usuario existente.
Entrada	Rut usuario
Salida	Mostrar mensaje “Usuario Modificado”
Condiciones	Ninguna
Datos de Prueba	Rut “15558250-2”

Tabla 5-11: Caso de Prueba Modificar Usuario 2

<b>Caso de Prueba Modificar Usuario 2</b>	
Especificaciones	Usuario no existente.
Entrada	Rut usuario
Salida	Mostrar mensaje “Usuario no registrado”
Condiciones	Ninguna
Datos de Prueba	Rut “15071109-6”

Tabla 5-12: Caso de Prueba Eliminar Usuario 1

<b>Caso de Prueba Eliminar Usuario 1</b>	
Especificaciones	Eliminar usuario existente.
Entrada	Rut usuario
Salida	Mostrar mensaje “Usuario Eliminado”
Condiciones	Ninguna
Datos de Prueba	Rut “15558250-2”

Tabla 5-13: Caso de Prueba Eliminar Usuario 2

<b>Caso de Prueba Eliminar Usuario 2</b>	
Especificaciones	Usuario No existente.
Entrada	Rut usuario
Salida	Mostrar mensaje “Usuario no registrado”
Condiciones	Ninguna
Datos de Prueba	Rut “15071109-6”

### 5.4.2.1 Resultados de Pruebas

Los resultados obtenidos al llevar a cabo el proceso de pruebas se clasificaron de la siguiente forma:

- **Aprobada:** La prueba determina que el caso experimentado cumple con lo esperado.
- **Aprobada con Observación:** La prueba determina que el caso experimentado cumple con lo esperado, pero con algún tipo de observación a ser tomada en cuenta.

**Fallida:** La prueba determina que el caso experimentado NO cumple con lo esperado.

Tabla 5-14: Resultado de Pruebas Iteración Configuración del Sistema

Caso de Prueba	Resultados		
	Aprobada	Aprobada con Observación	Fallida
Ingresar Usuario 1	X		
Ingresar Usuario 2		X	
Ingresar Usuario 3	X		
Modificar Usuario 1			X
Modificar Usuario 2	X		
Eliminar Usuario 1			X
Eliminar Usuario 2	X		

Resumen:

Pruebas Aprobadas: 4 de 7 (57,14%)

Pruebas Aprobadas con Observación: 1 de 7 (14,29%)

Pruebas Fallidas: 2 de 5 (28,57%)

## 5.5 Tercera Iteración: Presupuestos y Eventos

### 5.5.1 Implementación

#### 5.5.1.1 Interfaces



Ilustración 5.24: Interfaz Administrar Presupuestos

id_producto	descripcion	fecha	precio	cantidad
1-1-1-1	Mesa redonda para 10 personas	17-09-2008	1500	100
1-2-1-2	Mesa cuadrada para 12 personas	17-09-2008	2000	50
1-1-1-1	Mesa redonda para 10 personas	18-09-2008	1500	100
1-2-1-2	Mesa cuadrada para 12 personas	18-09-2008	2000	50

Ilustración 5.25: Interfaz Presupuesto

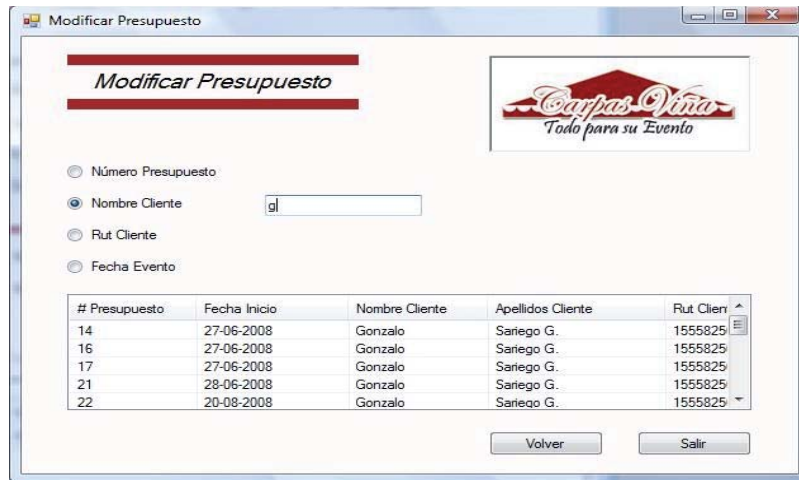


Ilustración 5.26: Interfaz Modificar Presupuesto (Búsqueda)

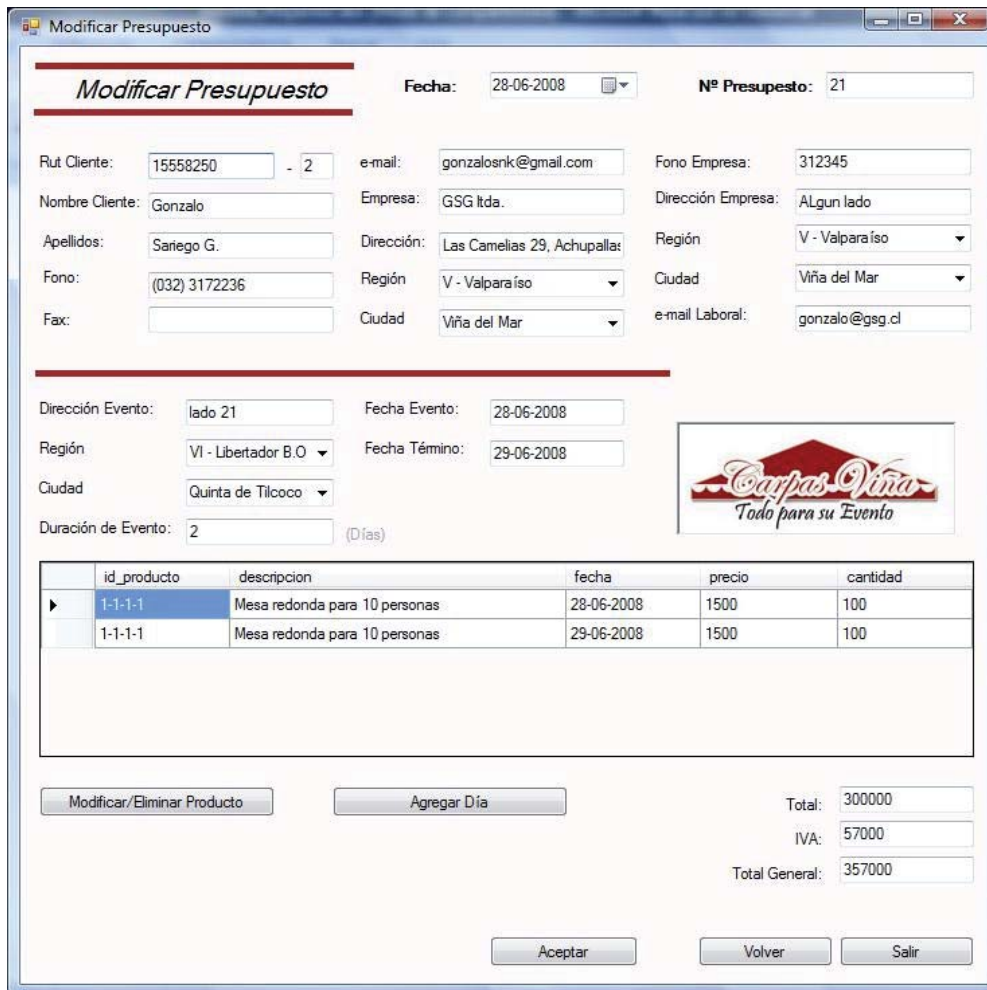


Ilustración 5.27: Interfaz Modificar Presupuesto

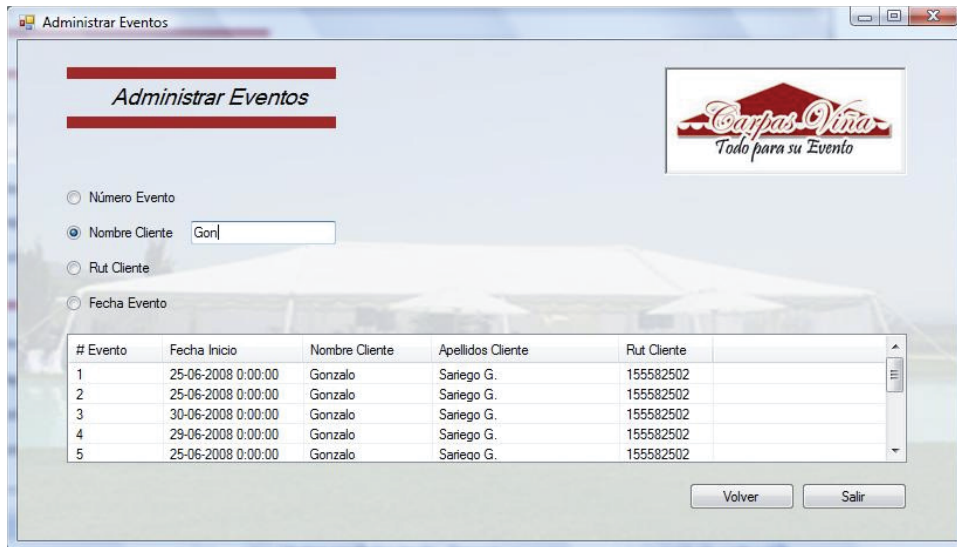


Ilustración 5.28: Interfaz Administrar Eventos (Búsqueda)

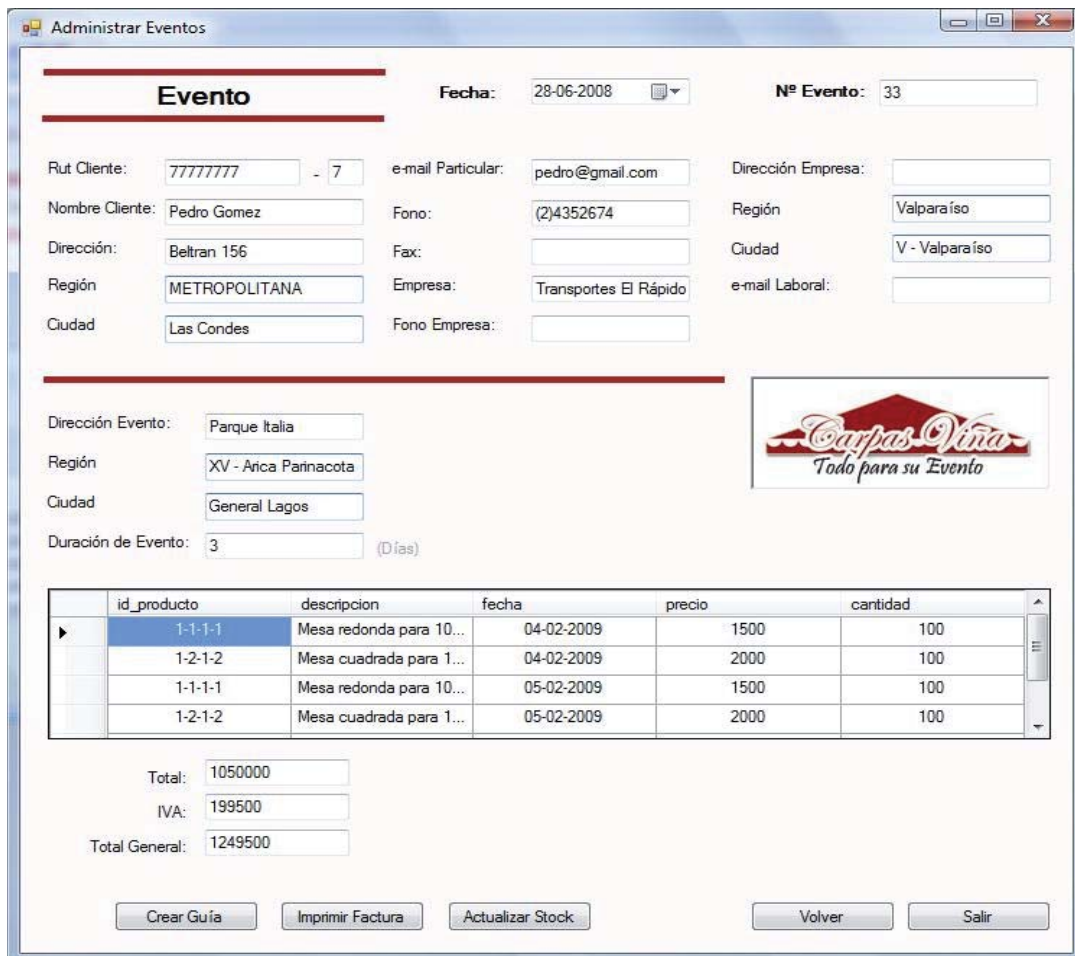


Ilustración 5.29: Administrar Eventos

### 5.5.2 Pruebas

A continuación se observan los distintos casos de pruebas realizados como parte de éste proceso, correspondientes a la Iteración actual (Presupuestos y Eventos):

**Tabla 5-15: Caso de Prueba Ingresar Presupuesto 1**

<b>Caso de Prueba Ingresar Presupuesto 1</b>	
Especificaciones	Ingresar Presupuesto ingresando todos los datos solicitados.
Entrada	Rut cliente, nombre, , mail, fono, dirección evento, fecha evento, duración
Salida	Mostrar mensaje “Presupuesto Almacenado Exitosamente”
Condiciones	Ninguna
Datos de Prueba	Rut “6680402-K”, Nombre “Jorge Lobos”, mail “ <a href="mailto:info@carpasvina.cl">info@carpasvina.cl</a> ”, fono “2868663”, fecha “15/04/2011”, duración “1”

**Tabla 5-16: Caso de Prueba Ingresar Presupuesto 2**

<b>Caso de Prueba Ingresar Presupuesto 2</b>	
Especificaciones	Ingresar Presupuesto dejando el campo fecha evento en blanco
Entrada	Rut cliente, nombre, mail, fono, dirección evento, duración
Salida	Mensaje “Debe ingresar campos obligatorios”
Condiciones	Ninguna
Datos de Prueba	Rut “6680402-K”, Nombre “Jorge Lobos”, mail “ <a href="mailto:info@carpasvina.cl">info@carpasvina.cl</a> ”, fono “2868663”, fecha “ ”, duración “1”

**Tabla 5-17: Caso de Prueba Ingresar Presupuesto 3**

<b>Caso de Prueba Ingresar Presupuesto 3</b>	
Especificaciones	Ingresar Presupuesto dejando el campo mail en blanco
Entrada	Rut cliente, nombre, fono, dirección evento, fecha evento, duración
Salida	Mensaje “Debe ingresar todos los campos obligatorios”
Condiciones	Ninguna
Datos de Prueba	Rut “6680402-K”, Nombre “Jorge Lobos”, mail “ ”, fono “2868663”, fecha “15/04/2011”, duración “1”

**Tabla 5-18: Caso de Prueba Modificar Presupuesto 1**

<b>Caso de Prueba Modificar Presupuesto 1</b>	
Especificaciones	Presupuesto existente.
Entrada	Rut Cliente
Salida	Mostrar mensaje “Presupuesto Modificado”
Condiciones	Ninguna
Datos de Prueba	Rut “15558250-2”

**Tabla 5-19: Caso de Prueba Modificar Presupuesto 2**

<b>Caso de Prueba Modificar Presupuesto 2</b>	
Especificaciones	Presupuesto no existente.
Entrada	Rut cliente
Salida	Ninguna
Condiciones	Ninguna
Datos de Prueba	Rut “15071109-6”



### 5.5.2.1 Resultados de Pruebas

Los resultados obtenidos al llevar a cabo el proceso de pruebas se clasificaron de la siguiente forma:

- **Aprobada:** La prueba determina que el caso experimentado cumple con lo esperado.
- **Aprobada con Observación:** La prueba determina que el caso experimentado cumple con lo esperado, pero con algún tipo de observación a ser tomada en cuenta.
- **Fallida:** La prueba determina que el caso experimentado NO cumple con lo esperado.

**Tabla 5-20: Resultados Pruebas Iteración Presupuestos y Eventos**

Caso de Prueba	Resultados		
	Aprobada	Aprobada con Observación	Fallida
Ingresar Presupuesto 1		X	
Ingresar Presupuesto 2	X		
Ingresar Presupuesto 3	X		
Modificar Presupuesto 1		X	
Modificar Presupuesto 2	X		

Resumen:

Pruebas Aprobadas: 3 de 5 (60%)

Pruebas Aprobadas con Observación: 2 de 5 (40%)

Pruebas Fallidas: 0 de 5 (0%)

## 5.6 Cuarta Iteración: Ventas

### 5.6.1 Implementación

#### 5.6.1.1 Interfaces



Ilustración 5.30: Administrar Ventas

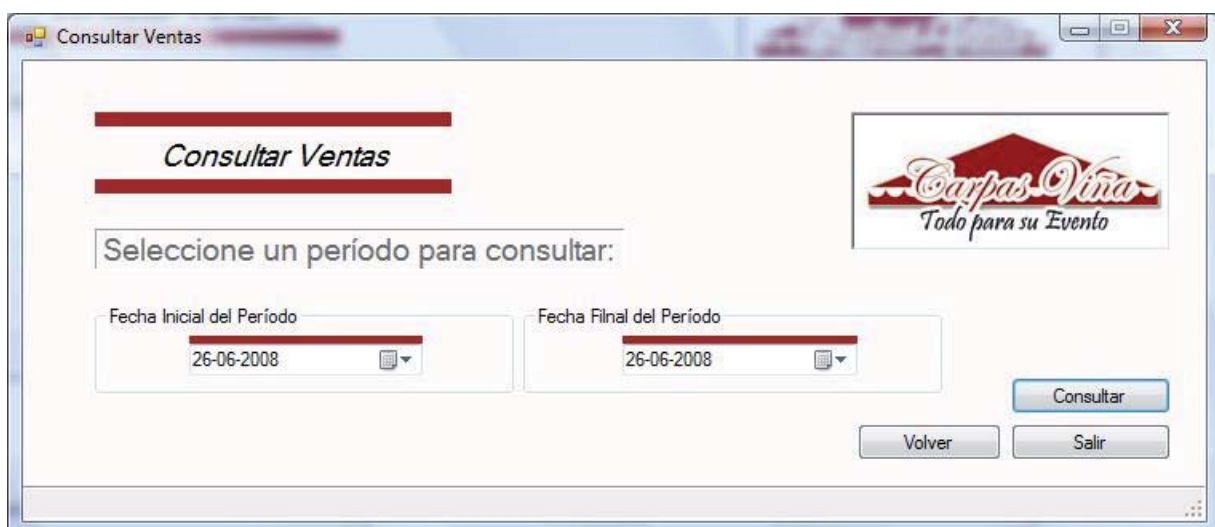


Ilustración 5.31: Interfaz Consultar Ventas

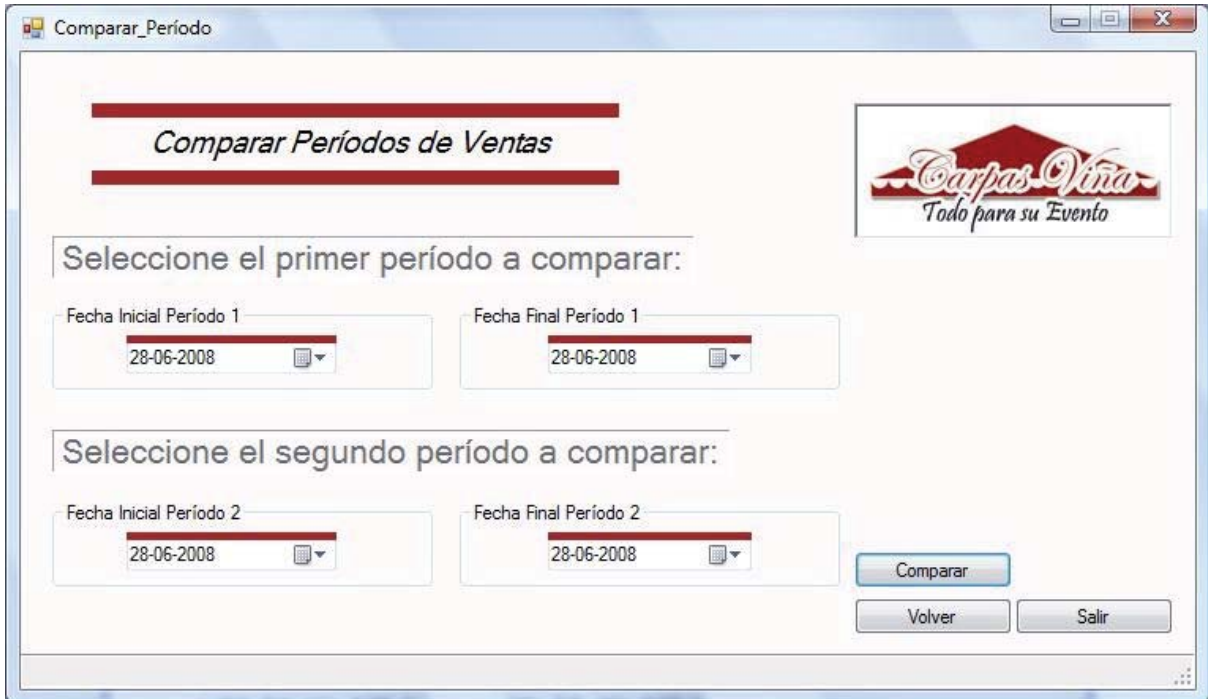


Ilustración 5.32: Interfaz Comparar Períodos de Ventas

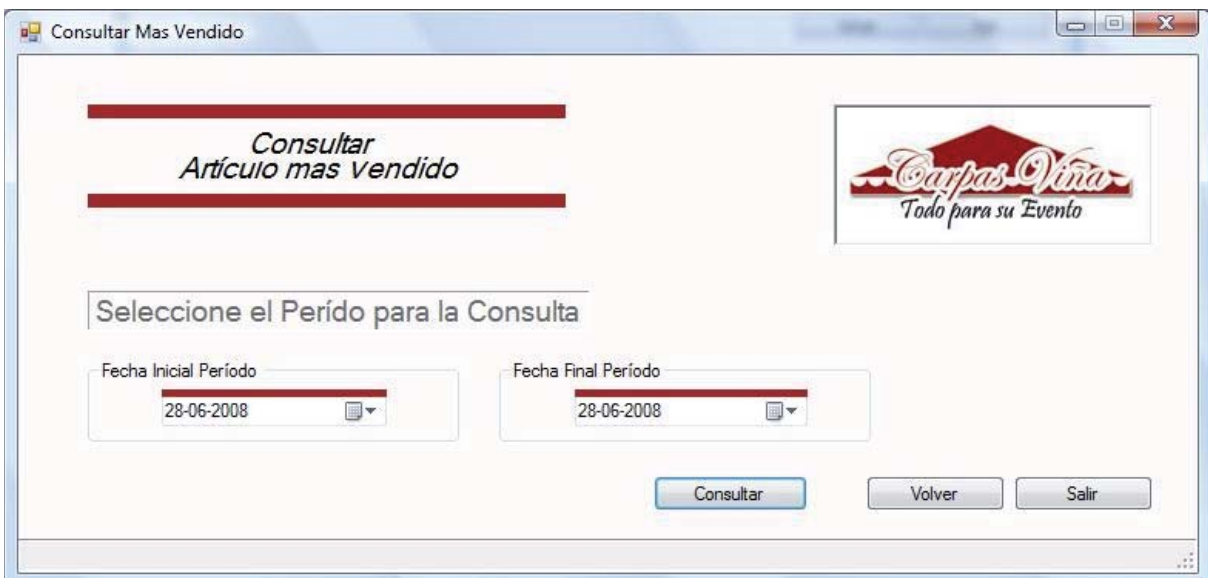


Ilustración 5.33: Interfaz Consultar Artículo Más Vendido

## 5.6.2 Pruebas

A continuación se observan los distintos casos de pruebas realizados como parte de éste proceso, correspondientes a la Iteración actual (Ventas):

**Tabla 5-21: Caso de Prueba Consultar Ventas 1**

<b>Caso de Prueba Consultar Ventas 1</b>	
Especificaciones	Seleccionar campos Fecha Inicial y Fecha Final, Siendo Fecha Inicial Menor a Fecha Final.
Entrada	Fecha Inicial, Fecha Final
Salida	Despliegue de Ventas del Período
Condiciones	Ninguna
Datos de Prueba	Fecha Inicial “01/01/2008”, Fecha Final “31/03/2008”

**Tabla 5-22: Caso de Prueba Consultar Ventas 2**

<b>Caso de Prueba Consultar Ventas 2</b>	
Especificaciones	Seleccionar campos Fecha Inicial y Fecha Final, Siendo Fecha Inicial Mayor a Fecha Final.
Entrada	Fecha Inicial, Fecha Final
Salida	Mostrar mensaje “La Fecha Inicial debe ser menor que la Fecha Final”
Condiciones	Ninguna
Datos de Prueba	Fecha Inicial “31/03/2008”, Fecha Final “01/01/2008”

**Tabla 5-23: Caso de Prueba Consultar Más Vendido 1**

<b>Caso de Prueba Consultar Más Vendido 1</b>	
Especificaciones	Seleccionar campos Fecha Inicial y Fecha Final, Siendo Fecha Inicial Mayor a Fecha Final.
Entrada	Fecha Inicial, Fecha Final
Salida	Despliegue del(los) artículo(s) más vendido(s) en el período seleccionado.
Condiciones	Ninguna
Datos de Prueba	Fecha Inicial “01/01/2008”, Fecha Final “31/03/2008”

**Tabla 5-24: Caso de Prueba Consultar Más Vendido 2**

<b>Caso de Prueba Consultar Más Vendido 2</b>	
Especificaciones	Seleccionar campos Fecha Inicial y Fecha Final, Siendo Fecha Inicial Mayor a Fecha Final.
Entrada	Fecha Inicial, Fecha Final
Salida	Mostrar mensaje “La Fecha Inicial debe ser menor que la Fecha Final”
Condiciones	Ninguna
Datos de Prueba	Fecha Inicial “31/03/2008”, Fecha Final “01/01/2008”

### 5.6.2.1 Resultados de Pruebas

Los resultados obtenidos al llevar a cabo el proceso de pruebas se clasificaron de la siguiente forma:

- **Aprobada:** La prueba determina que el caso experimentado cumple con lo esperado.
- **Aprobada con Observación:** La prueba determina que el caso experimentado cumple con lo esperado, pero con algún tipo de observación a ser tomada en cuenta.
- **Fallida:** La prueba determina que el caso experimentado NO cumple con lo esperado.

**Tabla 5-25: Resultados Pruebas Iteración Ventas**

<b>Caso de Prueba</b>	<b>Resultados</b>		
	<b>Aprobada</b>	<b>Aprobada con Observación</b>	<b>Fallida</b>
Consultar Ventas 1	X		
Consultar Ventas 2	X		
Consultar más Vendido 1	X		
Consultar más Vendido 2	X		

Resumen:

Pruebas Aprobadas: 4 de 4 (100%)

Pruebas Aprobadas con Observación: 0 de 4 (0%)

Pruebas Fallidas: 0 de 4 (0%)

# Capítulo 6: Desarrollo Fase Transición

## 6.1 Primera Iteración

### 6.1.1 Usabilidad del Sistema

La Organización Internacional para la Estandarización (ISO) ofrece dos definiciones de usabilidad:

#### **ISO/IEC 9126:**

*"La usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso"*

Esta definición hace énfasis en los atributos internos y externos del producto, los cuales contribuyen a su funcionalidad y eficiencia. La usabilidad depende no sólo del producto sino también del usuario. Por ello un producto no es en ningún caso intrínsecamente usable, sólo tendrá la capacidad de ser usado en un contexto particular y por usuarios particulares. La usabilidad no puede ser valorada estudiando un producto de manera aislada [6] .

#### **ISO/IEC 9241:**

*"Usabilidad es la eficiencia y satisfacción con la que un producto permite alcanzar objetivos específicos a usuarios específicos en un contexto de uso específico"*

Es una definición centrada en el concepto de calidad en el uso, es decir, se refiere a cómo el usuario realiza tareas específicas en escenarios específicos con efectividad [7] .

#### **6.1.1.1 Métodos de Análisis de Usabilidad**

##### **Métodos heurísticos**

- Técnicas heurísticas, desarrolladas por expertos, que analizan los interfaces de los módulos, evalúan su arquitectura y determinan sus puntos fuertes y débiles desde la perspectiva del usuario.

##### **Métodos Subjetivos**

- Obtienen información de los usuarios sobre prototipos operativos del prototipo en desarrollo (observación directa, cuestionarios, entrevistas, grupos de control,...)

## Métodos Empíricos

- Obtención de datos objetivos acerca de cómo los usuarios utilizan el sistema.
- Exactitud: Número de errores provocados durante un determinado lapso de tiempo.
- Velocidad: Celeridad en la interacción con el sistema.
- Exactitud y velocidad son magnitudes inversamente proporcionales.

Para probar la Usabilidad de este sistema se tomó en cuenta el método de análisis subjetivo debido a que su realización es más sencilla y directa con lo que se pretende obtener resultados más eficaces.

En lo que respecta a las medidas subjetivas, éstas se llevaron a cabo mediante la realización de cuestionarios dirigidos a los usuarios y observaciones de cómo éstos utilizaron y trabajaron en el sistema.

## Cuestionario Dirigido a los Usuarios

- **¿Ha encontrado el sistema innecesariamente complejo?**

R: La verdad es que el sistema se centra en los requerimientos que se establecieron en un comienzo y no desvía su enfoque hacia temas irrelevantes o que no vayan al caso, y por esta misma razón creo que está lejos de ser complejo y menos de forma innecesaria.

- **¿Le gustaría utilizar frecuentemente el sistema?**

R: Claro que sí, ese es el objetivo, ya que utilizando el sistema creo que tendremos una mejora en todo ámbito, ya sea en la atención hacia el cliente, el manejo de entrada y salida de los productos y también en lo que respecta a las finanzas.

- **¿Cree que necesitará la ayuda de personal técnico para utilizar el sistema?**

R: Creo que hay algunos aspectos en los que quizás necesite algún tipo de ayuda o guía para utilizar de manera correcta el sistema, sin embargo también creo que sólo pasa por un tema de costumbre

- **Con respecto a este tema, para que el usuario no experimentara problemas a la hora de utilizar el sistema, se incluyeron diversos “tool-tips” y mensajes de ayuda en cada una de las interfaces del sistema, además se creó un manual de usuario lo suficientemente sencillo de seguir con lo que se espera que la experiencia por parte del usuario sea la mejor posible.**

- **¿Se ha sentido seguro utilizando el sistema?**

R: Bueno creo que sí, aunque en algunas ocasiones uno tiene miedo de “meter la pata” pero el software cuenta con las opciones para modificar los datos que se ingresan en el caso de errores.

- **¿El sistema entrega “buenos” mensajes de error?**

R: Creo que proporciona buenos mensajes de errores porque me indican claramente de que se trata y que debo hacer para corregirlos.

- **¿Realizaría alguna modificación al sistema?**

R: De momento no, pero quizás en un futuro cercano, cuando comprendamos en su totalidad todas las prestaciones de este producto, es probable que nos pongamos ambiciosos y queramos agregarle funcionalidades extra.

- Como observación a esta respuesta se puede mencionar el hecho de que existió una planificación constante a lo largo del proyecto con el fin de diseñar un producto que pueda ser mejorado a través del tiempo.

- **¿Las imágenes usadas en el sistema son consecuentes con lo que se está mostrando?**

R: Creo que las imágenes que fueron seleccionadas para el software son claras y consecuentes con los distintos tópicos del sistema. No son “chillonas” ni de mal gusto.

- **¿En general obtiene de manera rápida lo que busca?**

R: Eso es lo mejor, obtengo lo que busco de forma inmediata, ni se compara a cuando tenía que buscar los datos manualmente, realmente eso nos tomaba demasiado tiempo.

- **¿Considera que el software cumplió con los objetivos establecidos?**

R: La verdad es que creo que el sistema cumplió con todos los objetivos e incluso superó nuestras expectativas por lo que pienso que nos será de mucha ayuda y utilidad.



## **6.1.2 Pruebas de Aceptación**

El objetivo de las Pruebas de Aceptación es comprobar que el Sistema, además de cumplir con lo requerido a nivel funcional, se desempeñe de forma eficiente y sea amigable al usuario, de tal modo que éste entregue su aprobación y el Sistema pueda ser entregado.

En reunión con el usuario se realizan las pruebas pertinentes al Sistema, obteniendo los resultados esperados a nivel de funcionalidad, rendimiento y usabilidad. Lo anterior garantiza que el Sistema pueda ser entregado sin problemas al Usuario.

# Conclusiones

Durante la elaboración de este proyecto ha aumentado considerablemente la comprensión por parte de los desarrolladores acerca de la gran importancia que conlleva el proceso de análisis y planificación a la hora de emprender proyectos de cualquier índole y envergadura. Junto con lo anterior también se pudieron dimensionar los beneficios que se obtienen mediante la mantención del orden y el control del grupo de trabajo. Es por esto que resulta imprescindible estudiar con claridad cada uno de los aspectos que influirán en la completa ejecución del proyecto, como también es preciso saber evaluar de forma correcta las distintas posibilidades o circunstancias por las cuales puede pasar el proyecto y sus características.

A la hora de elegir el ámbito del proyecto a realizar, se tomaron en cuenta muchas alternativas que variaban en cuanto a demanda de tiempo y recursos, como también en dificultad. Una vez que se analizaron todas esas opciones, junto con las condiciones y características de lo que significa realizar un proyecto de carácter académico, fue que finalmente se decidió realizar un sistema de información a la medida, esto debido a que este tipo de proyectos proveen un modo de trabajo ordenado y consistente el cual ha sido de gran ayuda para obtener mayor experiencia y comprender de mejor manera la forma en que se debe llevar a cabo un proyecto de software en general.

Debido a las características de Carpas Viña, organización con la que se trabajó durante este proyecto y que está orientada al rubro de los arriendos de materiales para eventos en general, el proceso de diseño de los modelos relacional y de clases implicó un trabajo exhaustivo de análisis de las entidades que forman parte de la Compañía, siendo los productos, los elementos que requerían de un estudio y modelado más especial debido a su naturaleza altamente variable. Luego de varias iteraciones con modificaciones y pruebas, se logró dar con un modelo relacional que satisfizo todas las necesidades funcionales que debía tener el Sistema, además de brindar una clara visión conceptual de todas las relaciones y atributos que poseían los elementos dentro de la empresa.

A medida que se fue progresando y avanzando con el trabajo se entendió la dificultad que implica la creación de un producto de software eficiente que satisfaga todos los requerimientos establecidos y que cumpla con estándares de nivel empresarial. Además es muy importante que cualquier tipo de sistema orientado a una solución en particular, sea lo suficientemente beneficioso y amigable para todos los usuarios finales, ya que con un sistema que aunque cumpla con los requerimientos, pero que sin embargo no es bienvenido por los usuarios y provoca una continua resistencia al cambio, existirá una alta probabilidad de no sea utilizado en el futuro. Por esto, es fundamental asegurar en todo momento que la usabilidad y la innovación del sistema sea aprobada de forma anticipada por los usuarios que van a interactuar con el sistema.

Gracias al activo involucramiento por parte de los desarrolladores a la información y las distintas reglas del negocio por las que se rige Carpas Viña, se logró trabajar a la par desde el punto de vista del cliente y en particular de los usuarios potenciales del Sistema. Por otra parte se realizaron numerosas visitas y reuniones con el cliente, que resultaron en una completa

capacitación y ambientación a la forma de trabajo específica de la empresa. Asimismo, se fueron presentando constantemente los avances y prototipos del sistema a los usuarios finales, lo que permitió conocer de manera oportuna los distintos comentarios y sugerencias que ellos creían pertinentes, como también se les orientó y aconsejó en algunos otros aspectos.

Esta buena comunicación entre los miembros del proyecto y también con el cliente, fue de vital importancia para entender de mejor modo los detalles y características tanto del funcionamiento de la Empresa, los requerimientos reales del cliente y los usuarios del sistema, como también las funcionalidades y prestaciones que debía tener el sistema a implementar. De esta manera, el sistema cumplió con las expectativas y se transformó en una solución óptima para las necesidades de la Empresa.

Ahora, se puede concluir que, aprendiendo de los errores y recogiendo los aciertos, se ha adquirido una gran experiencia que ha permitido llevar a cabo el desarrollo del proyecto de una manera eficaz y que cumplió a cabalidad con los objetivos establecidos lo que sin duda brindará también, una preparación realista a los desarrolladores para lo que les espera como futuros ingenieros

# Bibliografía

- [1] Pressman R.: *Ingeniería de Software, un enfoque práctico*, Mc Gra-Hill, 2002.
- [2] Sommerville I.: *Ingeniería de Software*, Pearson Educación SA, 2002.
- [3] Booch G., Rumbaugh J., Jacobson I., *El Proceso Unificado de Desarrollo de Software*, Pearson Educación SA, 1999.
- [4] \*\*\*, Documento sobre UP,  
[http://lml.ls.fi.upm.es/mdp/si/pres\\_rup.pdf](http://lml.ls.fi.upm.es/mdp/si/pres_rup.pdf)
- [5] Elmasri, Navathe, *Fundamentos de Sistemas de Bases de Datos*, Addison-Wesley, 2007.
- [6] Bevan, 1994.
- [7] \*\*\*, Definición de Usabilidad,  
<http://es.wikipedia.org/wiki/Usabilidad>