

## INDICE DE CONTENIDO

<b>RESUMEN</b> .....	<b>12</b>
<b>SUMMARY</b> .....	<b>13</b>
<b>CAPITULO I: GENERALIDADES</b> .....	<b>15</b>
1.1 INTRODUCCIÓN.....	15
1.2 PLANTEAMIENTO DEL PROBLEMA.....	15
1.3 JUSTIFICACIÓN .....	16
1.4 OBJETIVOS .....	16
1.5 ALCANCE .....	17
1.6 FACTIBILIDAD .....	18
<b>CAPITULO II: MARCO TEÓRICO</b> .....	<b>20</b>
2.1 ARQUITECTURA WEB .....	20
2.2 HERRAMIENTA DE DESARROLLO ASP .NET .....	23
2.3 ANÁLISIS COMPARATIVO ENTRE LAS TECNOLOGÍAS MÁS RECONOCIDAS ACTUALMENTE: ASP .NET Y JSP ..	27
2.4 HTML .....	32
2.5 MICROSOFT SQL SERVER 2005 .....	33
<b>CAPITULO III: METODOLOGÍA</b> .....	<b>34</b>
3.1 INTRODUCCIÓN.....	34
3.2 METODOLOGÍAS PARA EL DESARROLLO DE SISTEMAS WEB. ....	35
3.3 COMPARATIVA ENTRE LAS METODOLOGÍAS MENCIONADAS:.....	43
3.4 UWE .....	48
<b>CAPITULO IV: FASE DE ANÁLISIS Y DISEÑO</b> .....	<b>63</b>
4.1 CONCEPTUALIZACIÓN DEL SISTEMA .....	63
4.2 ANÁLISIS DE REQUERIMIENTOS .....	64
4.3 FASE DE DISEÑO DEL PROYECTO GOLDCARD.....	72
4.4 DIAGRAMAS DE CASOS DE USO.....	75
4.5 DICCIONARIO DE CASOS DE USO .....	81
4.6 DISEÑO CONCEPTUAL .....	82
4.7 DISEÑO NAVEGACIONAL .....	84
4.8 DISEÑO DE PRESENTACIÓN .....	85
FIGURA 4.15 DIAGRAMA DE SECUENCIA PARA EL MODULO DE SEGURIDADES .....	85
<b>CAPITULO V: IMPLEMENTACIÓN</b> .....	<b>90</b>
5.1 ARQUITECTURA WEB DE LA APLICACIÓN GOLDCARD .....	90
5.2 ESTÁNDARES DEL PROYECTO GOLDCARD.....	91
5.3 ESTÁNDARES DE PROGRAMACIÓN .....	98
<b>CAPITULO VI: PRUEBAS</b> .....	<b>109</b>
<b>CAPITULO VII: CONCLUSIONES Y RECOMENDACIONES</b> .....	<b>117</b>
7.1 CONCLUSIONES.....	117
7.2 RECOMENDACIONES .....	119
<b>BIBLIOGRAFÍA</b> .....	<b>120</b>

## LISTADO DE TABLAS

Tabla 1.1. Factibilidad Económica del Proyecto GoldCard .....	18
Tabla 1.2. Valores Financiados por Solsupply.....	19
Tabla 2.1. Principales características de .NET y J2EE .....	29
Tabla 2.2. IDE's de Desarrollo.....	30
Tabla 2.3. Requerimientos J2EE.....	30
Tabla 2.4. Requerimientos .NET.....	30
Tabla 3.1 Requisitos contemplados en cada metodología propuesto.....	43
Tabla 3.2 Técnicas por metodología contempladas en cada Actividad.....	44
Tabla 3.3. Orientación de las Propuestas.....	46
Tabla 3.4. Tipos de Multiplicidad de una relación entre clases.....	56
Tabla 4.1. Tabla de Tipos de Tarjeta GoldCard.....	63
Tabla 4.2. Arquitectura de la Aplicación GoldCard.....	73
Tabla 4.3. Descripción y Actividades del Usuario Súper Administrador.....	73
Tabla 4.4. Descripción y actividades del Usuario Administrador.....	74
Tabla 4.5. Descripción y Actividades del Usuario Centro Autorizado.....	74
Tabla 5.1. Detalle de la Arquitectura de la Aplicación GoldCard.....	90
Tabla 5.2. Tipos de datos enteros en C Sharp.....	101
Tabla 5.3. Tipos de datos de punto flotante en C Sharp.....	102
Tabla 5.4. Tipos de datos caracteres en C Sharp.....	102
Tabla 5.5. Tipos de datos lógicos en C Sharp.....	103
Tabla 5.6. Operadores en C Sharp.....	104

## LISTADO DE FIGURAS

Figura 2.1. Diagrama lógico de capas.....	22
Figura 2.2. Modelo de Arquitectura de Aplicaciones .NET.....	24
Figura 3.1. Ejemplo de una clase.....	54
Figura 3.2. Características de una relación entre clases.....	55
Figura 3.3. Ejemplo de una relación de asociación.....	56
Figura 3.4. Ejemplo de una relación de herencia.....	57
Figura 3.5. Ejemplo de una relación de dependencia.....	58
Figura 3.6. Estereotipos para los elementos básicos de navegación.....	59
Figura 3.7. Ilustración de una clase menú con sus ítems y sus respectivos links.....	61
Figura 4.1. Proceso Logístico GoldCard (Proceso A).....	66
Figura 4.2. Proceso Logístico GoldCard (Proceso B).....	66
Figura 4.3. Proceso Logístico GoldCard (Proceso C).....	67
Figura 4.4. Proceso Logístico GoldCard (Proceso D).....	68
Figura 4.5. Proceso Logístico GoldCard (Proceso E).....	69
Figura 4.6. Proceso Logístico GoldCard (Proceso G).....	71
Figura 4.8. Diagrama de Caso de Uso General del Sistema.....	75
Figura 4.9. Diagrama de Caso de Uso para el usuario Súper Administrador del modulo de Seguridades.....	76
Figura 4.10. Diagrama de Caso de Uso para el usuario Administrador del modulo de Administración.....	77
Figura 4.11. Diagrama de Caso de Uso para el usuario Súper Administrador del modulo de Inscripción.....	78

Figura 4.12. Diagrama de Caso de Uso para el usuario Súper Administrador, Administrador y Taller del modulo de Orden de Trabajo.....	79
Figura 4.13. Diagrama de Caso de Uso para el usuario Súper Administrador, Administrador y Taller del modulo de deportes.....	80
Figura 4.14. Diagrama de Estructura de Navegación Usuario Taller .....	84
Figura 4.15. Diagrama de Secuencia para el modulo de Seguridades.....	85
Figura 4.16. Diagrama de Secuencia para el modulo de Inscripción.....	85
Figura 4.17. Diagrama de Secuencia para el modulo de Orden de Trabajo.....	86
Figura 4.18. Diagrama de Secuencia Reporte de Inscripción.....	86
Figura 4.19. Diagrama de Secuencia para Reporte de Orden de Trabajo.....	87
Figura 4.20. Diagrama de Secuencia para Reporte de Cliente/Vehículo.....	87
Figura 5.1. Login del Sistema GoldCard.....	94
Figura 5.2. Página Principal del sistema GoldCard.....	95
Figura 5.3. Cabecera de las Paginas del Sistema GoldCard.....	96
Figura 5.4. Pagina ejemplo de administración del Sistema GoldCard.....	96
Figura 5.5. Pagina Ejemplo de Reporte del Sistema GoldCard.....	97
Figura 5.6. Distribución de la Aplicación Sistema GoldCard.....	98
Figura 6.1. Prueba de la Pagina Login en Internet Explorer 8.....	109
Figura 6.2. Prueba de la Pagina Principal en Internet Explorer 8.....	109
Figura 6.3. Prueba de una Página de Administración en Internet Explorer 8 .....	110
Figura 6.4. Prueba de una Página de Reporte en Internet Explorer 8.....	110
Figura 6.5. Prueba de la Página Login en Mozilla Firefox.....	111
Figura 6.6. Prueba de una Pagina Administrativa en Mozilla Firefox.....	111
Figura 6.7. Prueba de una Pagina Reporte en Mozilla Firefox.....	112

Figura 6.8. Prueba de una Página Principal en Opera 9.6.....	112
Figura 6.9. Prueba de una Página Administrativa en Opera 9.6.....	113
Figura 6.10. Prueba de una Página Principal en Opera 9.6.....	113
Figura 6.11. Prueba de una Página Login en Google Chrome 2.0.....	114
Figura 6.12. Prueba de una Página Principal en Google Chrome 2.0.....	114
Figura 6.13. Prueba de una Página Administrativa en Safari 2.0.....	115
Figura 6.14. Prueba de una Página Login en Safari 2.0.....	115
Figura 6.15. Prueba de una Página Principal en Safari 2.0.....	116
Figura 6.16. Prueba de una Página Administrativa en Safari 2.0.....	116

## **LISTADO ANEXOS**

**Anexo 4.1** Proceso Logístico GoldCard

**Anexo 4.2** Proceso Logístico GoldCard (Proceso F)

**Anexo 4.3.** Proceso Actual de manipulación de la información

**Anexo 4.4.** Diccionario de Clases del Proyecto GoldCard.

**Anexo 4.5.** Diagrama de Clases de Dominio GoldCard

**Anexo 4.6** Diagrama de Estructura de Navegación Usuario Súper Administrador.

**Anexo 4.7** Diagrama de Estructura de Navegación Usuario Administrador.

**Anexo 4.8** Diagrama de Secuencia para el Módulo de Administración.

**Anexo 4.9** Manual de Implantación Sistema GoldCard.

**Anexo 4.8** Manual de Usuario Sistema GoldCard.

## NOMENCLATURA UTILIZADA

- I. **SISTEMA WEB:** Software diseñado para permitir interoperabilidad entre máquinas sobre una red. Permite comunicación entre diferentes máquinas, con diferentes plataformas.
- II. **GoldCard:** Tarjeta de descuento en servicios de mantenimiento vehicular.
- III. **UWE:** Herramienta para modelar aplicaciones, basado en el proceso unificado y UML pero adaptados a la Web.
- IV. **CENTRALIZADA:** Toda la información recolectada por la aplicación está ubicada en un mismo punto.
- V. **N-Tier:** Se refiere a los varios niveles de responsabilidad en el diseño de un sistema. La *N* en *N-Tier* puede ser cualquier número de 2 en adelante.
- VI. **.NET Framework:** Es la base sobre la que se asienta .NET. Es la pieza fundamental de esta tecnología que va a proporcionar herramientas y servicios necesarios para el desarrollo.
- VII. **JAVA:** Lenguaje de programación orientado a objetos. La sintaxis de Java es similar a la utilizada por lenguajes C y C++.
- VIII. **J2EE:** (Java 2 Enterprise Edition) define un estándar para el desarrollo de aplicaciones empresariales multicapa diseñado por Sun Microsystems.
- IX. **IU:** Los componentes de IU son aquellos que interactúan con el usuario (ventanas o páginas Web).

- X. **UML:** (Unified Modeling Language - Lenguaje Unificado de Modelado). UML es un lenguaje de modelado que permite construir, documentar, visualizar y especificar un sistema de software.
- XI. **XML:** son las siglas de Extensible Markup Language, una especificación/lenguaje de programación desarrollada especialmente para los documentos de la Web.
- XII. **Datasets:** En general se denomina así a cualquier conjunto de datos.
- XIII. **SQLHelper:** Es una clase que proporciona un conjunto de métodos estáticos que se pueden utilizar para ejecutar varios tipos de comandos diferentes contra una base de datos SQL Server.
- XIV. **CLR:** Entorno de Ejecución Común o Common Language Runtime es el motor en tiempo de ejecución del .NET Framework.
- XV. **API:** (Application Programming Interface - Interfaz de Programación de Aplicaciones). Una API representa un interfaz de comunicación entre componentes software.
- XVI. **ASP .NET:** ASP.NET de Microsoft es una tecnología de scripts que corren en el servidor y pueden ser utilizados para crear aplicaciones dinámicas e interactivas en la Web. Una página ASP.NET es una página de HTML que contiene scripts que son procesados por un servidor Web antes de ser enviados al navegador del usuario.
- XVII. **Web Forms:** Un formulario Web es un tipo de formulario que es presentado en un navegador y puede ser rellenado a través de una red como Internet. Generalmente cuando se ingresan los datos, se envían a un servidor Web para ser procesados.

- XVIII. Web Services:** es una forma estandarizada de integrar aplicaciones WEB mediante el uso de XML, SOAP, WSDL y UDDI sobre los protocolos de la Internet. Los Web Services permiten intercambiar datos sin necesidad de conocer los detalles de sus respectivos Sistemas de Información.
- XIX. DLL:** Componentes. Es un ejecutable de una clase que posee métodos los cuales pueden ser consumidos por otra aplicación.
- XX. JSP:** (Java Server Page) Página de Servidor Java. Es un tipo especial de páginas HTML, en las cuales se insertan programas muy pequeños denominados scripts que corren en Internet.
- XXI. CLI:** Command Line Interface o Línea de comandos
- XXII. SOAP (Simple Object Access Protocol):** SOAP es un protocolo para el intercambio de mensajes sobre redes de computadoras, generalmente usando HTTP.
- XXIII. WSDL (Web Services Definition Language):** WSDL es el lenguaje común basado en XML utilizado para la descripción de los servicios Web.
- XXIV. C#:** es un lenguaje orientado a objetos que interaccionan unos con otros a través de sus métodos.
- XXV. HTML:** Hyper Text Markup Language (Lenguaje de marcación de Hipertexto) es el lenguaje de marcas de texto utilizado normalmente en la WWW (World Wide Web).
- XXVI. DDL:** Los comandos DDL (definición de *comandos DDL*) o de *definición* de datos en SQL controlan la creación, modificación y eliminación de objetos.
- XXVII. UIs:** son modelos gráficos que representan la interacción entre el usuario y el sistema, sin considerar aspectos específicos de a interfaz.

- XXVIII. HDM (Hypermedia Design Model):** Es un modelo de diseño que propone una estructura jerarquizada que facilita la *navegación*.
- XXIX. Argo UWE:** Herramienta propia de la metodología de desarrollo de software UWE que permite elaborar todos los diagramas propuestos por esta metodología.
- XXX. UWA:** es un proyecto que ha nacido de la colaboración de varios grupos. Su fase de tratamiento de requisitos se basa en roles de usuario y en ir refinando requisitos en un proceso iterativo mediante el que se clasifiquen los objetivos según su carácter.
- XXXI. NDT-Tool:** es un conjunto de herramientas para aplicar la metodología NDT en entornos prácticos.
- XXXII. Herramientas CASE:** la automatización del desarrollo del software, contribuyendo a mejorar la calidad y la productividad en el desarrollo de sistemas de información.
- XXXIII. Reviews o Walk-throughs:** Está técnica consiste en modelar la definición de requisitos es decir la correcta y completa lectura y corrección de la documentación
- XXXIV. Query:** un query en base de datos es una búsqueda o pedido de datos almacenados en una base de datos.
- XXXV. Constraint:** sirve para definir una restricción que se podrá eliminar cuando queramos sin tener que borrar la columna.
- XXXVI. DALCS:** Capa de acceso a datos.
- XXXVII. Store Procedures:** Un procedimiento almacenado es una subrutina disponible para las aplicaciones que acceden a un sistema de base de datos relacional. Los procedimientos son almacenados en la base de datos.

**XXXVIII. Business Logic:** La lógica de negocio o Business logic, es un término no técnico, empleado en general para describir los algoritmos funcionales que manejan el intercambio de información entre una base de datos y una interfaz de usuario. Se diferencia de la validación de datos de entrada/salida y de la lógica de producto.

**XXXIX. Business Entity:** Entidad de negocio.

## RESUMEN

La empresa Solsupply requiere de una aplicación Web que permita una mejor administración de la información manejada en la venta de la tarjeta de descuentos en mantenimiento vehicular GoldCard, proyecto del cual participan tres talleres de la ciudad de Sangolquí.

El proyecto contempla siete capítulos en los que se han considerado parámetros que permitieron el correcto desarrollo de la aplicación Web.

Capítulo I. Se detallan aspectos que permiten tener una visión general del proyecto, como son: introducción, planteamiento del problema, justificación, objetivos, alcance y factibilidad técnica y económica.

Capítulo II. Se describe el marco teórico de nuestro proyecto destallando temas como la arquitectura Web utilizada para el desarrollo de la aplicación, las herramientas de desarrollo empleadas y las características y ventajas de las herramientas de apoyo.

Capítulo III. Aquí se mencionan aspectos metodológicos aplicados en el desarrollo, en los que consta, la metodología UWE con una descripción de sus fases y un análisis comparativo con las demás metodologías existentes.

Capítulo IV. Se describe el desarrollo de la aplicación Web como el análisis de requerimientos basados en el proceso logístico de la manipulación de datos y los diferentes diagramas utilizados en el diseño de la aplicación Web.

Capítulo V. Se detallan los estándares de programación de la aplicación Web, y estándares para pantallas de la aplicación.

Capítulo VI. Se muestra las pruebas de funcionamiento realizadas a la aplicación Web.

Al término del proyecto se presentan las conclusiones y recomendaciones, establecidas del análisis y desarrollo del Sistema de administración para la tarjeta de descuento en mantenimiento vehicular para la Empresa Solsupply S.A.

## **SUMMARY**

Solsupply Company requires a Web Application that allows a better administration of the information obtained from the sales of the GoldCard card for vehicle maintenance discounts; there are three authorized centers of Sangolquí City that are involved in this project.

This project provides seven chapters where had been considered parameters that allowed the correct development of the Web Application.

Chapter I. Aspects related to an overview of the project, such as: introduction, problem statement, justification, objectives, project scope and technical and economic feasibility.

Chapter II. Web Architecture description used for the development of the Web application, development tools used and support tools characteristics and advantages.

Chapter IV. Web Application Development such as the requirements analysis based in the logistic handling of data, and the different diagrams used for the Web Application Design.

Chapter V. Web Application development standards and windows standards.

Chapter VI. Web Application performance test.

Finally we find the conclusions and recommendations established from the analysis and development of the GoldCard Administration System.

# **CAPITULO I: Generalidades**

## **1.1 Introducción**

Hoy en día el Internet es una puerta abierta para las empresas que desean vender sus productos y servicios acaparando el mayor mercado posible, brindando a pequeñas, medianas y grandes empresas la posibilidad de incrementar sus ventajas competitivas.

## **1.2 Planteamiento del Problema**

SOLSUPPLY y sociedad ha visto la necesidad de la creación de una aplicación Web que le permita una mejor administración de la información dentro del proyecto GoldCard así como el uso de la misma para toma de decisiones.

El sistema GoldCard será versátil, orientado a la Web permitiendo un mejor manejo de información. Al ser una aplicación centralizada podrá proporcionar a la gerencia una herramienta de negocios, la que le permitirá a todos los talleres de servicios automotores contar con información oportuna y real de cada cliente.

Finalmente se implementará un sistema eficaz, eficiente, de fácil manejo y con una interfaz amigable para el usuario, brindando de esta manera un servicio mucho más rápido y confiable.

### **1.3 Justificación**

El desarrollo de un Sistema Web permite que la información este accesible donde y cuando sea requerida, es decir, la información puede ser ingresada, consultada y administrada en cualquier momento y desde cualquier lugar que tenga acceso a Internet.

Con la implantación de esta tecnología se podrá mejorar el servicio de atención al cliente integrando no solo información sino también servicios que van a beneficiar en el desempeño de las distintas actividades de trabajo y atención al cliente.

Dentro de las características principales del Portal destacan:

- Administración Segura y Ordenada de la información
- Administración de la tarjeta GoldCard

### **1.4 Objetivos**

#### **1.4.1 General**

- Analizar, diseñar, implementar e implantar una aplicación Web que permita la gestión y administración de la tarjeta de descuento GoldCard para mantenimiento vehicular.

#### **1.4.2 Específicos**

- Analizar los requerimientos funcionales y no funcionales del sistema previo a la implementación del mismo.

- Aplicar los procedimientos establecidos por la metodología UWE requeridos para el análisis, diseño y desarrollo de la aplicación Web.
- Implementar la aplicación utilizando los diagramas y modelos obtenidos al aplicar la metodología anteriormente mencionada.
- Implantar el sistema y capacitar al personal autorizado para su uso.

## 1.5 Alcance

El sistema GoldCard permite realizar las siguientes tareas:

- **Gestión de talleres autorizados**
  - Registro de talleres de servicios automotores.
  - El registro de los talleres de servicios automotores se realizará únicamente por parte del usuario administrador del sistema.
- **Gestión de tarjeta habientes**
  - Registro de clientes y sus automotores.
- **Gestión de la tarjeta de mantenimiento**
  - Registro de tarjetas con su respectivo cliente.
  - Administración de tarjetas.
  - Administración de la venta de tarjetas.
- **Reportes**
  - Impresión de reportes mensuales para toma de decisiones.

Es muy importante tomar en cuenta que la aplicación Web estará centralizada mediante la cual accederán los cuatro talleres de servicios automotores y una oficina principal vía Internet.

## 1.6 Factibilidad

### 1.6.1 Técnica

Los recursos técnicos necesarios para el desarrollo del sistema se encuentran disponibles en la empresa SOLSUPPLY de la misma manera todos los requerimientos legales del software a utilizar. En cuanto a la fase de análisis diseño y desarrollo de la aplicación se cuenta con todo el software y hardware necesario para el desarrollo de la misma. Además se posee la bibliografía adecuada para el desarrollo de la aplicación Web, contamos con la experiencia de los profesores y tenemos el total apoyo de SOLSUPPLY como empresa auspiciante.

### 1.6.2 Económica

Los valores que serán financiados por los estudiantes:

**Tabla 1.1:** Factibilidad Económica del Proyecto GoldCard

<b>Rubros</b>	<b>Valor USD</b>
Dedicación 1488 horas-hombre	0
Material bibliográfico	200
Material de oficina	200
Uso de Equipos / Internet	240
Derechos de grado	1400
Adicionales	200
<b>Total</b>	<b>2240</b>

Los valores que serán financiados por SOLSUPPLY y asociados son los siguientes:

**Tabla 1.2:** Valores Financiados por Solsupply

<b>Cantidad</b>	<b>Descripción</b>	<b>Valor aproximando (dólares)</b>
<b>Hardware</b>		
2	Desktop PC	0.00
1	Servidor	6.500,00
<b>Software</b>		
1	Windows Server 2003	700.00
1	Visual Studio.NET 2005 Professional Edition	799,00
1	Base de Datos SQL Server 2005	0.00
<b>Total</b>		<b>7999.00</b>

### 1.6.3 Operativa

Contamos con el auspicio de SOLSUPPLY, una empresa seria dentro del mercado lo cual garantiza la seriedad y compromiso para la realización y puesta en marcha del sistema GoldCard y la predisposición de los alumnos.

## **CAPITULO II: Marco Teórico**

### **2.1 Arquitectura Web**

#### **2.1.1 Definición**

La vista conceptual de la estructura de una aplicación se refiere a la arquitectura de la misma, esta difiere según como estén distribuidos sus códigos (código de presentación, código de procesamiento de datos y código de almacenamiento de datos).

La arquitectura de una aplicación Web se desarrolla bajo un esquema de tres niveles:

- La capa de presentación o primer nivel incluyen el Navegador y el Servidor Web que es el responsable de dar datos en un formato adecuado.
- El segundo nivel se refiere a un script o tipo de programa.
- El tercer nivel le proporciona al segundo nivel todos los datos indispensables para su ejecución

El funcionamiento de una aplicación Web bajo el esquema de esta arquitectura recogerá los datos del usuario (primer nivel) luego los enviará al servidor el cual va a ejecutar el script o programa referido(segundo y tercer nivel) y el resultado será formateado y presentado en el navegador al usuario final (primer nivel).(1)

---

(1) <http://www.geocities.com/trescapas/TresCapas.htm>, Consulta: marzo 2009, Actualización.: enero 2008

### **Nivel de Aplicación (CLIENTE)**

Este nivel aplicado en una arquitectura de tres capas trabaja con la semántica propia de aplicación, sin tener que preocuparse de cómo está implementado, ni de cómo es su estructura física.

### **Nivel de Dominio de Aplicación (SERVIDOR)**

Este nivel es el que se encarga de toda la estructura física y dominio de la aplicación. La ventaja más importante de esta arquitectura es que si se cambiara alguna regla en el servidor de aplicación este cambio se verá reflejado en todos los clientes, lo que no sucedía en una arquitectura de dos capas ya que si alguna regla es cambiada esta se tenía que cambiar en cada cliente.

### **Nivel de Repositorio (SERVIDOR DE DATOS)**

En este nivel o capa es en donde se almacenan los datos y toda la información.

#### **2.1.2 Diagramas n – Tier**

Hoy en día muchas empresas olvidan que se debe tener una arquitectura adecuada, lo cual provoca un grave problema en el desarrollo de aplicaciones debido a que el desarrollo se vuelve más complejo, así como el mantenimiento y los patrones de diseño.

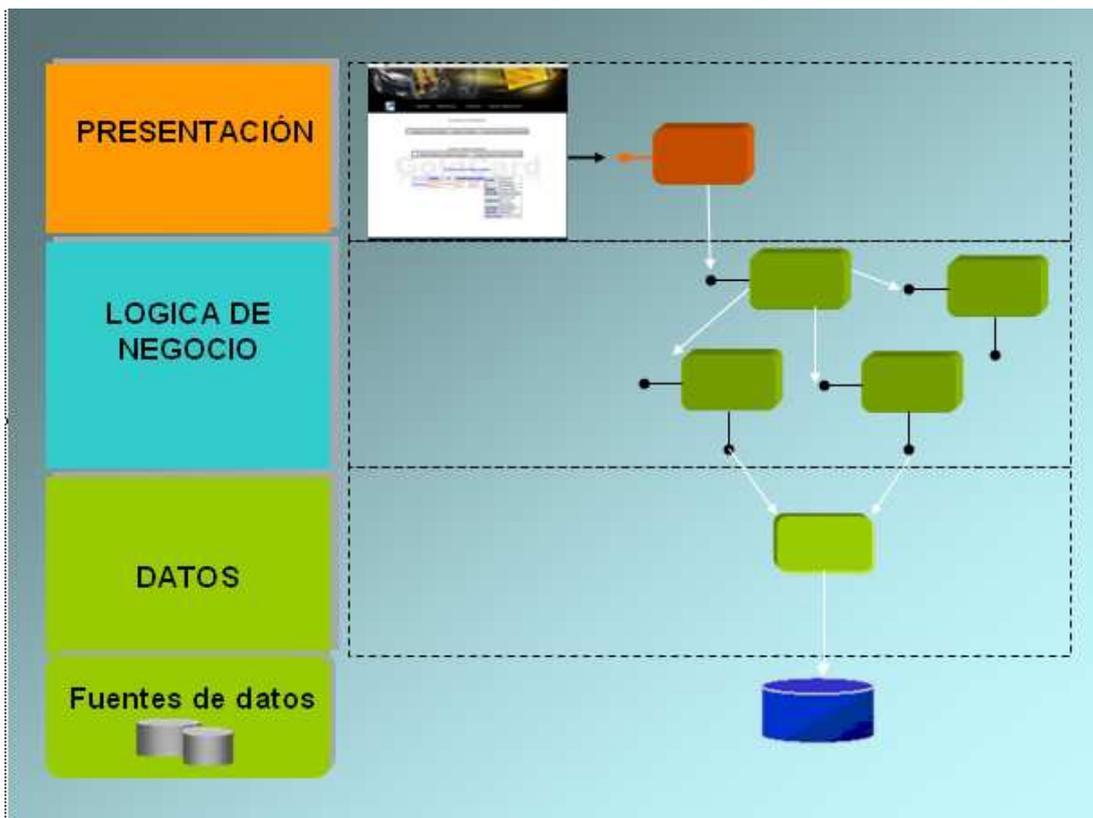
La arquitectura n-Tiers consiste en dividir las reglas de negocio de la interfaz de usuario y el acceso a datos. (2)

---

(2) Fuente: [http://www.webtaller.com/maletin/articulos/las\\_tecnologias\\_de\\_las\\_paginas\\_web.php](http://www.webtaller.com/maletin/articulos/las_tecnologias_de_las_paginas_web.php), Consulta marzo 2009, Actualización: marzo 2004

## Ventajas del modelo

- Desarrollo paralelo (en cada capa)
- Encapsulamiento (Aplicaciones más robustas)
- Mantenimiento más sencillo.
- Mayor flexibilidad
- La principal ventaja es la alta escalabilidad debido a que se puede manejar varias peticiones bajo el mismo rendimiento sencillamente incrementando más hardware. Es un crecimiento casi lineal y no se necesita incrementar o añadir más código para lograr conseguir dicha escalabilidad.



**Figura 2.1** Diagrama lógico de capas

- **SERVICIOS DE PRESENTACION**
  - Muestra el interfaz de usuario
  - Gestiona las acciones del usuario
- **SERVICIOS DE NEGOCIO**
  - Aplicar la lógica y reglas de negocio.
- **SERVICIOS DE DATOS**
  - Abstracción del almacén de datos.
  - Asegurar la consistencia, y aislamiento de datos.

## **2.2 Herramienta de Desarrollo ASP .NET**

### **2.2.1 Introducción**

“La Plataforma .NET es una tecnología creada por Microsoft para sus productos de programación en la línea de .NET Framework”. (3) Por lo cual Microsoft creó una herramienta no solo de programación sino también de diseño que cuenta con mayor número de funcionalidades ubicándola como una tecnología tan importante y de punta como JAVA y su arquitectura J2EE en el desarrollo de aplicaciones

Con esto Microsoft, agrego el apellido “.NET” (4) a la mayoría de sus software comerciales creados bajo este nuevo enfoque de desarrollo, productos como Visual Basic .NET

Se añaden nuevas herramientas no incluidas en versiones anteriores, además de cambios para los programadores quienes afrontan garantías del desarrollo de la

---

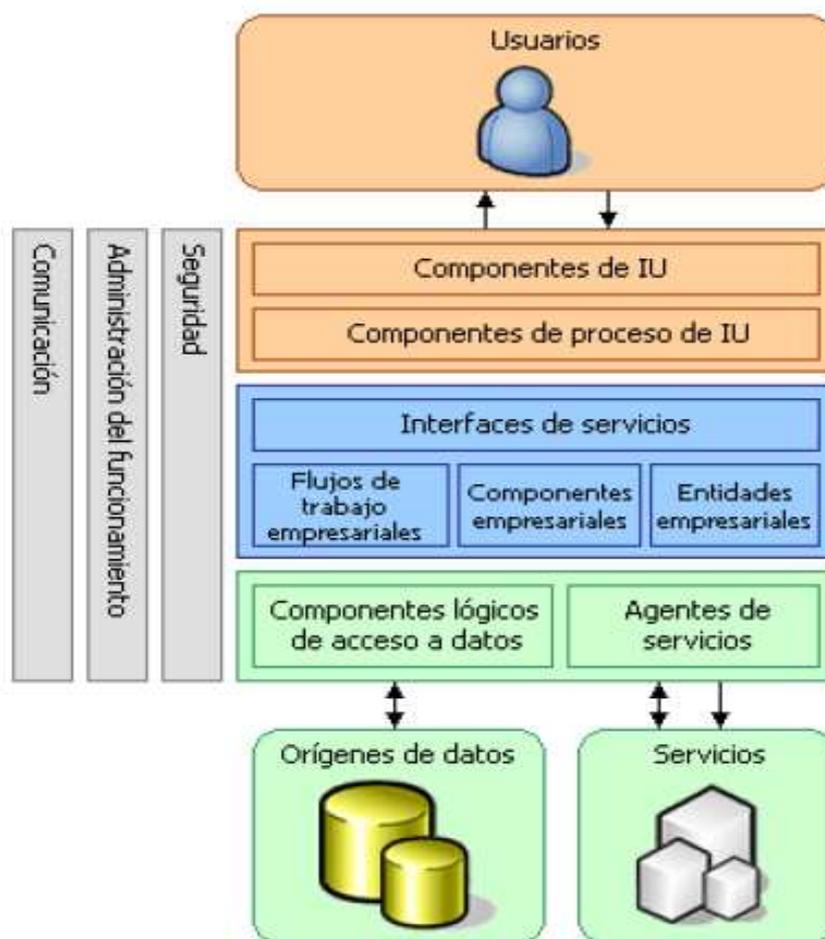
(3) Fuente: <http://biblioteca.uct.cl/tesis/miguel-garrido/tesis.pdf>, Consulta: abril 2009, Actualización: marzo 2004

(4) .NET: Tecnología creada por Microsoft para el desarrollo de aplicaciones.

nueva generación de aplicaciones en general, especialmente en el ámbito Web por ser el objetivo perseguido por todas las herramientas de desarrollo actuales.

### 2.2.2 Arquitectura .NET

Más allá de la programación en .NET se pueden incluir el análisis y el diseño dentro de esta tecnología. “De esta manera se integran todas las etapas del ciclo de desarrollo dentro de .NET, definiendo las especificaciones y modelado del problema para llegar al resultado final reflejado en el código obtenido.” (5)



**Figura 2.2:** Modelo de Arquitectura de Aplicaciones .NET. (5)

(5) Fuente: <http://biblioteca.uct.cl/tesis/miguel-garrido/tesis.pdf>, Consulta: abril 2009, Actualización: marzo 2004

Como se puede observar en la figura anterior, la arquitectura .NET presenta distintos niveles de aplicaciones y componentes permitiendo de esta forma ofrecer varios tipos de servicios apoyada en los distintos sistemas operativos Windows y sistemas compatibles. La arquitectura .NET se divide en tres capas descritas a continuación:

- **La capa de presentación:** Esta capa está formada por los Componentes de IU<sup>(6)</sup> y los componentes de proceso de IU. Los componentes de IU son aquellos que interactúan con el usuario (ventanas o páginas Web). “Los componentes de proceso de IU encapsulan la lógica de navegación y control de eventos de la interface.”<sup>(7)</sup>
- **La capa de negocios:** En esta capa están los componentes empresariales que contienen la lógica de negocios los cuales en general se refieren a las transacciones. Las llamadas “entidades empresariales” representan a los objetos que van a ser manejados por la aplicación (modelo de objetos, xml, datasets con tipo, estructuras de datos, para nuestro caso bussines entities), objetos que deberán ser identificados durante el modelamiento.
- **La capa de acceso a datos:** capa que contiene las clases que interactúan con las base de datos. Éstas surgen con la necesidad de mantener la cohesión o clases altamente especializadas que ayuden a reducir la dependencia entre las clases y capas. Aquí se encuentra la clase de métodos estáticos que permite

---

(6) IU: Interfaz de Usuario

(7) Fuente: <http://biblioteca.uct.cl/tesis/miguel-garrido/tesis.pdf>, Consulta: abril 2009, Actualización: marzo 2004

uniformizar las operaciones de acceso a datos a través de un único conjunto de métodos.

### **2.2.3 .NET Framework.**

Se puede definir a .NET Framework como “Una plataforma independiente del lenguaje para el desarrollo de Aplicaciones Web” (8), la cual tiene como objetivo el modelo de programación basado en aplicaciones y servicios, permitiendo a los desarrolladores centrarse en escribir la lógica específica a desarrollar con esta arquitectura compuesta por una serie de librerías y un entorno de ejecución común.

.NET Framework agrupa tres bloques principales que permiten el desarrollo de aplicaciones a través del uso de herramientas y servicios, estos tres bloques son:

- CLR: Entorno de Ejecución Común o Common Language Runtime
- La jerarquía de clases básicas de la plataforma o .NET Framework Base Classes
- Motor de generación de interfaz de usuario, que permite crear interfaces para la Web o para el tradicional entorno Windows.

### **2.2.4 La Arquitectura .NET Framework.**

Se pueden diferenciar con claridad 3 tipos de niveles en Framework .NET

En la base del entorno de ejecución se encuentra el CLR, que constituye el núcleo de .NET Framework, este se encarga de la gestión del código en cuanto a su carga,

---

(8) Fuente: <http://biblioteca.uct.cl/tesis/miguel-garrido/tesis.pdf>, Consulta: abril 2009, Actualización: marzo 2004

ejecución, manipulación de memoria, seguridad, etc., todas funciones propias en cuanto al Sistema Operativo y la aplicación.

En el nivel intermedio, se sitúa la jerarquía de clases básicas del entorno de ejecución, que constituyen un sólido API de servicios a disposición del programador, para realizar múltiples tareas como, gestión del sistema de ficheros, acceso a datos, etc.

Finalmente, en el nivel superior, se encuentran las clases que permiten el diseño del interfaz de usuario de las aplicaciones. Si se necesita desarrollar aplicaciones para Internet, se utiliza ASP.NET, la cual provee todo lo necesario para crear aplicaciones para la Red, (Web Forms, Web Services, etc.). Este modelo no solo se aplica para programación Web, sino también para aplicaciones de ambiente Windows.

Esta arquitectura presenta 2 nuevos ítems que son:

- Un modelo de programación sencillo y consistente orientado a objetos.
- Eliminación del problema de compatibilidad entre DLLs

## **2.3 Análisis comparativo entre las tecnologías más reconocidas actualmente: ASP .NET y JSP**

### **2.3.1 Introducción**

Actualmente se puede encontrar mucha información sobre estas dos tecnologías muy reconocidas en su ámbito. Existen variadas opiniones acerca de que tecnología es mejor y porque, sin embargo más allá de las opiniones tanto del usuario como de las empresas que auspician a estas dos tecnologías algunos estudios concluyen que

en un futuro ambas plataformas competirán por ganar mayor mercado pero sin embargo ninguna reemplazará a la otra, incluso se puede pensar que podrían llegar a fusionarse para juntas logras más y mayores objetivos.

### 2.3.2 Definiciones

- **Definición .NET**

La plataforma .NET se basa en estándares abiertos tales como CLI<sup>(9)</sup> (Command Line Interface), SOAP<sup>(10)</sup> (Simple Object Access Protocol) y WSDL<sup>(11)</sup> (Web Services Definition Language) los mismos que le permiten al programador disponer de un entorno único para trabajar en múltiples lenguajes.

Los lenguajes base del entorno .NET son C# , Visual Basic y Jscript.

El lenguaje C# que se caracteriza por ser simple, fuertemente tipado, completamente orientado a objetos y a componentes, y contener mecanismos que permiten que el código sea consistente y durable.

Una de las principales características de .NET es su orientación al desarrollo de servicios principalmente servicios Web.

“Un servicio Web es un software que se suministra a aplicaciones basadas en componentes y que resuelve una determinada necesidad.”<sup>(12)</sup>

---

(9) CLI: [http://kb.NETgear.com/app/answers/detail/a\\_id/955](http://kb.NETgear.com/app/answers/detail/a_id/955), Consulta: Mayo 2009, Actualización: enero 2003

(10) SOAP: <http://www.alegsa.com.ar/Dic/soap.php>, Consulta: Mayo 2009, Actualización: 2009

(11) WSDL: [http://www.w3schools.com/wSDL/wSDL\\_intro.asp](http://www.w3schools.com/wSDL/wSDL_intro.asp), Consulta: Mayo 2009, Actualización: 2009

(12) Fuente: <http://www.issi.uned.es/pea/programacion-c/downloadds/NET%20:y%20J2EE.pdf>, Consulta: abril 2009, Actualización: junio 2006.

- **Definición de J2EE**

La plataforma J2EE es el proceso de construcción de varias empresas en forma conjunta y no exclusiva de tal manera que no le pertenece a ninguna de las empresas participantes en su desarrollo.

El lenguaje base es Java, cuyas características principales son la portabilidad, robustez, estabilidad, independencia de la plataforma y una completa orientación a objetos. Al igual que C++, Java es fuertemente tipado.

Java se divide en dos entornos, un entorno de ejecución el cual permite la instalación de todo lo necesario para la carga de la máquina virtual Java y un SDK que tiene todo lo necesario para poder escribir programas en Java.

**Tabla 2.1:** Principales características de .NET y J2EE

Característica	.NET	Java J2EE
Tipo de Tecnología	Producto	Estándar
Empresas que lo ofrecen	Microsoft	Más de 30
Librería de Desarrollo	.NET Framework SDK	Java Core API
Interprete	CLR	JRE
Páginas Dinámicas	ASP .NET	Servletts, JSP
Componentes	.NET Framework SDK	JRE
Acceso a Base de Datos	ADO .NET	JDBC, SQL/J
Servicios WEB	SOAP, WSDL, UDDI	SOAP, WSDL, UDDI
Interfaces Gráficas	WIN Forms y WEB Forms	Java Swing
Herramientas de Programación	Visual Studio .NET	Depende del Fabricante
Transacciones Distribuidas	MS-DTC	JTC
Servicios de Directorios	ADSI	JNDI
Librería de Mensajes	MSMQ	JMS 1.0
Lenguajes Utilizados	C#, Visual Basic, C	JAVA
Lenguaje Intermedio	IL	Bytecodes

**Tabla 2.2:** IDE's de Desarrollo

Característica	NetBeans	Visual Studio .NET
Manejo de Licencia	Licencia Libre	Comercial
Componentes de Funcionamiento	JVM (Máquina Virtual)	Ninguno
Orientación al Desarrollo	Específico para aplicaciones Java	Tanto Web con aplicación Win32
Componentes de Funcionamiento Web	Tomcat	IIS
Duración de Instalación	Más menos 15 min.	Más menos 1Hr 40 min.
Tamaño de IDE	90 MB (más JDK)	1.751 MB

**Tabla 2.3 :** Requerimientos J2EE

Hardware	Funcionalidad Mínima	Funcionalidad Recomendada
Procesador	Pentium III 500 MHz	Cualquiera Superior
Memoria RAM	256 MB	512 MB
Espacio en Disco Duro	350 MB	350 MB

**Tabla 2.4:** Requerimientos .NET

Hardware	Funcionalidad Mínima	Funcionalidad Recomendada
Procesador	Pentium II 450 MHz	Pentium II 733 MHz
Memoria RAM	128 MB	256 MB
Espacio en Disco Duro	3 GB	3 GB

### 2.3.3 Ventajas de punto Net frente a J2EE

- Múltiples lenguajes de programación.
- C# es la evolución de lenguajes de programación orientada a objetos en el cual, se incorpora características y capacidades de Java o C++ que los programadores pudieran necesitar o sencillamente extrañar en el cambio, a la vez que simplifica algunos de los complejos mecanismos de otros lenguajes.
- El entorno .NET está especialmente orientado a la creación de servicios Web. La existencia de una arquitectura clara y sencilla de clases para crear y distribuir estos servicios, incluyendo formatos como SOAP (Simple Object Access Protocol), WSDL (Web Services Description Languages) y UDDI (Universal Description, Discovery, and Integration).<sup>(12)</sup>

### 2.3.4 Ventajas de J2EE Frente a .NET

- Varias empresas brindan las implementaciones de J2EE a diferencia de .NET en donde solamente una es a la que debemos adquirir.
- JAVA tiene un largo camino andado en relación al desarrollo de su arquitectura sobre diferentes plataformas.
- La tecnología JAVA es abierta y se base en gran medida en estándares de organizaciones de normalización.

---

(12) Fuente: <http://www.issi.uned.es/pea/programacion-c/downloads/NET%20y%20J2EE.pdf>, Consulta: abril 2009, Actualización: junio 2006.

## 2.4 HTML

### 2.4.1 Definición

El HTML, Hyper Text Markup Language (Lenguaje de marcación de Hipertexto) es el lenguaje de marcas de texto utilizado normalmente en la WWW (World Wide Web). El concepto de Hipertexto (Conocido también como link o ancla) permite conectar dos elementos entre si y el SGML (Lenguaje Estándar de Marcación General) el cual sirve para colocar etiquetas o marcas en un texto que indique como debe verse.

HTML no presenta ningún compilador, por lo tanto algún error de sintaxis que se presente éste no lo detectará y se visualizará en la forma como éste lo entienda.

Estos documentos pueden ser mostrados por los visores o "browser" de páginas Web en Internet, como Opera<sub>(13)</sub>, Mozilla Firefox<sub>(14)</sub>, Google Chrome<sub>(15)</sub>, Zafari<sub>(16)</sub> y Microsoft Internet Explorer.

También existe el HTML Dinámico (DHTML), que es una mejora de Microsoft de la versión 4.0 de HTML que le permite crear efectos especiales como, por ejemplo, texto que vuela desde la página palabra por palabra o efectos de transición al estilo de anuncio publicitario giratorio entre página y página.

---

(13) <http://es.wikipedia.org/wiki/%C3%93pera>, Consulta: junio 2009, Actualización: 2007

(14) <http://www.mozilla-europe.org/es/firefox/>, Consulta: junio 2009, Actualización: 2009

(15) <http://www.alegsa.com.ar/Dic/google%20chrome.php>, Consulta: junio 2009, Actualización: 2009

(16) <http://www.mundoextra.com/internet/el-explorador-de-apple-safari-ahora-en-windows>, Consulta: junio 2009, Actualización: 2008

## **2.5 Microsoft SQL Server 2005**

### **2.5.1 Definición**

Microsoft SQL Server 2005 es un Gestor de bases de datos relacionales de Microsoft el cual se encuentra en la capacidad de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.

### **2.5.2 Características de Microsoft SQL Server**

- Soporte de transacciones.
- Seguridad, estabilidad y escalabilidad.
- Soporta procedimientos almacenados.
- 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 clientes de la red sólo acceden a la información.
- Permite administrar información de otros servidores de datos

## CAPITULO III: Metodología

### 3.1 Introducción

Debido al continuo avance del Internet y las comunicaciones en los últimos tiempos se han venido desarrollando propuestas metodológicas que ofrezcan un marco referencial adecuado para el desarrollo de una aplicación Web. Son varias las propuestas de metodologías que ofrecen procesos y técnicas adecuadas para trabajar con este tipo de aplicativos. Por lo cual es necesario realizar un estudio comparativo que refleje las diferentes características de adaptación y procesos de realización de cada tipo de metodología propuesta para de esta forma trabajar con la más adecuada a nuestro proyecto.

“El desarrollo de aplicaciones Web agrupa una serie de características que lo hacen diferente del desarrollo de otros sistemas.”<sup>(17)</sup> Primeramente hay que tomar en cuenta que diferentes tipos de agentes forman parte del proceso de desarrollo de la aplicación, entre ellos: analistas, clientes, usuarios, diseñadores gráficos, expertos en multimedia, seguridad, entre otros. Por otro lado, “la existencia en estos sistemas de una importante estructura de navegación obliga a un desarrollo preciso de este aspecto que garantice que el usuario no se pierde en el espacio navegacional del sistema”.<sup>(18)</sup>

---

(17), (18) Fuente: <http://www.lsi.us.es/docs/informes/LSI-2002-4.pdf>(1), Consulta: marzo 2009, Actualización.: enero 2008

Los sistemas Web tratan con múltiples medios por lo cual es indispensable que ofrezcan una interfaz adecuada, esto obliga a que todas estas características propias de la Web deban ser tratadas de manera especial en el proceso de desarrollo. Por ello, la mayoría de las metodologías propuestas ofrecen diferentes clasificaciones de los requisitos a pesar de que la terminología utilizada no es siempre la misma.

### **3.2 Metodologías para el Desarrollo de Sistemas Web.**

#### **3.2.1 WSDM: Web Site Design Method**

WSDM (De Troyer & Leune, 1997) es una propuesta para el desarrollo de sitios Web, en la que el sistema se define en base a los grupos de usuarios. El desarrollo del proceso se divide en cuatro fases: modelo de usuario, diseño conceptual, diseño de la implementación e implementación. La fase que más repercusión tiene según nuestra opinión es la primera en la que se trata de detectar los perfiles de usuario. Para lo cual se deben realizar dos tareas:

- Clasificación de usuarios
- Descripción de los grupos de usuarios

#### **3.2.2 SOHDM: Scenario-based Object-Oriented Hypermedia Design Methodology**

Esta propuesta (Lee, Lee & Yoo, 1998) inicia con el proceso de capturar las necesidades del sistema. Para lo cual propone el uso de escenarios. La fase de definición de requisitos parte de la realización de un diagrama de contexto. En este

se identifican las entidades externas que se comunican con el sistema, así como los eventos que provocan esa comunicación. La lista de eventos es una tabla que indica en qué eventos puede participar cada entidad. Por cada evento diferente, SOHDM propone elaborar un escenario. Éstos son representados gráficamente mediante los denominados SACs2 (Scenario Activity Chart). Cada escenario describe el proceso de interacción entre el usuario y el sistema cuando se produce un evento determinado, especificando el flujo de actividades, los objetos involucrados y las transacciones realizadas. SOHDM propone un proceso para conseguir a partir de estos escenarios el modelo conceptual del sistema, que es representado mediante un diagrama de clases. El proceso de SOHDM continúa reagrupando estas clases para conseguir un modelo de clases navegacionales del sistema.

### **3.2.3 RNA: Relationship-Navigational Analysis**

RNA (Bieber, Galnares & Lu, 1998) plantea una secuencia de pasos para el desarrollo de aplicaciones Web, centrándose fundamentalmente en el flujo de trabajo de análisis.

El proceso de trabajo que presenta RNA se basa en la realización de las siguientes fases:

- Fase 1- Análisis del entorno
- Fase 2- Elementos de interés
- Fase 3- Análisis del conocimiento
- Fase 4- Análisis de la navegación
- Fase 5- Implementación del análisis

La propuesta de RNA es quizás una de las que más ha resaltado la necesidad de trabajar con la especificación de requisitos, incluyendo tareas como el análisis del entorno y de los elementos de interés. Además, resulta interesante pues plantea la necesidad de analizar los requisitos conceptuales de manera independiente a los navegacionales.

#### **3.2.4 HFPM: Hypermedia Flexible Process Modeling**

La propuesta de HFPM (Olsina, 1998) describe un proceso detallado que cubre todo el ciclo de vida de un proyecto software. HFPM propone un total de trece fases para las cuales se especifican a su vez una serie de tareas. Para este estudio es principalmente relevante la primera fase, denominada de modelado de requisitos, cuyas tareas son las siguientes:

- Descripción breve del problema.
- Descripción de los requisitos funcionales mediante casos de uso.
- Realizar un modelo de datos para esos casos de uso, proponiendo el uso de un modelo de clases.
- Modelar la interfaz de usuario. Para ello, propone el uso de sketches y prototipos que permitan presentar los datos al usuario.
- Modelar los requisitos no funcionales. En éstos incluyen la navegación, la seguridad, etc.

### **3.2.5 OOHDM: Object Oriented Hypermedia Design Model**

OOHDM es una propuesta metodológica ampliamente aceptada para el desarrollo de aplicaciones de la Web (Schwabe & Rossi, 1998). Esta propuesta parte de los casos de uso. Igualmente, resalta la necesidad de empezar el diseño del sistema, especialmente en los entornos Web, teniendo un claro y amplio conocimiento de las necesidades de interacción, o lo que es lo mismo de la forma en la que el usuario va a comunicarse con el sistema.

Partiendo de estas dos premisas, OOHDM propone que la comunicación con el usuario se realice utilizando los casos de uso y a partir de ellos los analistas elaboran los UIs. Estos UIs son modelos gráficos que representan la interacción entre el usuario y el sistema, sin considerar aspectos específicos de la interfaz. OOHDM centra el desarrollo de un sistema de información Web entorno al modelo conceptual de clases. Este diagrama debe surgir de los requisitos que se definan del sistema, pero los casos de uso resultan demasiado ambiguos para ello. Así, propone refinar el proceso de desarrollo descrito en UML, de forma que de los casos de uso se generen los UIs que concreten más la definición de los requisitos para, a partir de ellos, obtener el diagrama conceptual.

### **3.2.6 UWE: UML-Based Web Engineering**

UML-Based Web Engineering (UWE) es una propuesta metodológica basada en el Proceso Unificado (Jacobson, Booch & Rumbaugh, 1999) y UML para el desarrollo de aplicaciones Web (Hennicker & Koch, 2000, Koch, 2001). UWE cubre todo el ciclo

de vida de este tipo de aplicaciones, centrando además su atención en aplicaciones personalizadas. Esta metodología distingue entre la tarea de capturar requisitos, definir y validar los requisitos. El resultado final de la captura de requisitos en UWE es un modelo de casos de uso acompañado de documentación que describe los usuarios del sistema, las reglas de adaptación, los casos de uso y la interfaz.

UWE clasifica los requisitos en dos grandes grupos: funcionales y no funcionales.

Los requisitos funcionales tratados por UWE son:

- Requisitos relacionados con el contenido
- Requisitos relacionados con la estructura
- Requisitos relacionados con la presentación
- Requisitos relacionados con la adaptación
- Requisitos relacionados con los usuarios

Además, UWE propone como técnicas apropiadas para la captura de los requisitos de un sistema Web las entrevistas, los cuestionarios y los checklists y los casos de uso, los escenarios y el glosario para la definición de requisitos. Para la validación propone walk-throughs, auditorías y prototipos.

### **3.2.7 W2000**

W2000 (Baresi, Garzotto & Paolini, 2001) supone una propuesta que amplía la notación de UML con conceptos para modelar elementos de multimedia heredados

de la propuesta HDM (Hypermedia Design Model) (Garzotto, Schwabe & Paolini, 1993). El proceso de desarrollo de W2000 se divide en tres etapas: análisis de requisitos, diseño de hipermedia y diseño funcional. El primero de ellos es el que resulta interesante para este trabajo.

La especificación de requisitos en W2000 se divide en dos subactividades: análisis de requisitos funcionales y análisis de requisitos navegacionales. La especificación de requisitos comienza haciendo un estudio de los diferentes roles de usuario que van a interactuar con el sistema. Cada actor potencialmente distinto tendrá su modelo de requisitos de navegación y de requisitos funcionales.

El modelo de requisitos funcionales es representado como un modelo de casos de uso tal y como se propone en UML. En él se representa la funcionalidad principal asociada a cada rol y las interacciones que se producen entre el sistema y cada rol. El segundo modelo consiste en otro diagrama de casos de uso pero que no representa funcionalidad sino posibilidades de navegación de cada actor. La representación gráfica es realizada con una extensión de UML.

### **3.2.8 UWA: Ubiquitous Web Applications**

UWA ha nacido de la colaboración entre diferentes grupos de trabajo, por lo que resulta realmente una agrupación de propuestas y técnicas. En concreto, la propuesta de W2000 se encuentra incluida en UWA. Sin embargo, W2000 ha sido incluida en UWA sólo en la fase de diseño hipermedia, siendo ambas propuestas

diferentes en la fase de definición de requisitos. Por esta razón han sido incluidos en este trabajo de forma separada.

El proceso de captura de requisitos en UWA (Uwa Requirements Elicitation, 2001) comienza definiendo los diferentes roles de usuario que pueden interactuar con el sistema, los objetivos globales de éste y la relación entre ellos. El proceso continúa haciendo un refinamiento de esos objetivos globales, concretándolos en subobjetivos. Estos subobjetivos son estudiados y refinados para detectar conflictos entre ellos. De esta forma, se concretizan aún más dividiéndolos en requisitos. Los requisitos son clasificados en varios tipos: de contenido, de estructura de contenido, de acceso, de navegación, de presentación, de operaciones de usuario y de operaciones del sistema.

De esta forma, los requisitos se van refinando hasta que solo pertenezcan a uno de estos grupos. Y finalmente los requerimientos son asignados a artefactos de diseño o a reglas de customización.

Para definir los objetivos, UWA propone una notación propia, basada en una plantilla.

La definición de los actores y la relación con los objetivos se hace usando un diagrama basado en casos de uso. Por último, para definir y refinar los subobjetivos y los 16 requisitos, utilizan una notación gráfica propia que denominan *grafo de refinamiento de objetivos*, el refinamiento de este grafo permite ir representando la relación entre los requisitos y hacer un seguimiento para validar la consecución de

los objetivos del sistema. Una vez que los requisitos son detectados, hacen uso de XML para definirlos de una manera formal.

### **3.2.9 NDT - Navigational Development Techniques**

NDT (Navigational Development Techniques) (Escalona, Torres & Mejías, 2002) es una técnica para especificar, analizar y diseñar el aspecto de la navegación en aplicaciones Web. Para este trabajo, solo es relevante la propuesta que ofrece para la definición y captura de requisitos. El flujo de especificación de requisitos de NDT comienza con la fase de captura de requisitos y estudio del entorno. Para ello, plantea el uso de técnicas como las entrevistas. Tras esta fase, se propone la definición de los objetivos del sistema. En base a estos objetivos, el proceso continúa definiendo los requisitos que el sistema debe cumplir para cubrir los objetivos marcados. NDT clasifica los requisitos en:

- Requisitos de almacenamiento de información
- Requisitos de actores
- Requisitos funcionales
- Requisitos de interacción, representados mediante:
  - Frases, que recogen cómo se va a recuperar la información del sistema utilizando un lenguaje especial denominado BNL (Bounded Natural Language) (Brisaboa, Penabad, Places & Rodríguez, 2001).

- Prototipos de visualización, que representan la navegación del sistema, la visualización de los datos y la interacción con el usuario.

### 3.3 Comparativa entre las metodologías mencionadas:

#### 3.3.1 Requisitos Contemplados

La primera comparativa a realizarse será determinar qué tipos de requisitos contempla cada metodología propuesta. En la tabla que se muestra a continuación se presentan los diferentes requisitos y se indica cuáles de ellos son tratados en cada metodología.

**Tabla 3.1** Requisitos contemplados en cada metodología propuesta

METODOLOGIA	Req. de Datos	Req. de Usuario	Req. Navegacionales	Req. Personalización	Req. Transaccionales	Req. No Funcionales
WSDM	✓			✓		✓
SOHDM	✓	✓			✓	
RNA	✓	✓	✓		✓	
HFPM	✓	✓	✓			✓
OOHDM	✓	✓	✓			
UWE	✓	✓	✓	✓		✓
W2000			✓	✓	✓	
UWA	✓	✓	✓	✓	✓	
NDT	✓	✓	✓	✓	✓	✓
DDDP	✓	✓	✓	✓	✓	✓

En la tabla comparativa puede verse que las propuestas metodológicas más actuales resaltan la utilización de requisitos de personalización y navegación, así como los transaccionales de forma independiente.

### 3.3.2 Técnicas y Actividades Contempladas.

Para realizar esta comparativa, se presentan en la tabla 3.2 las técnicas para capturar, definir y validar requisitos. Para cada propuesta se indica qué técnicas son descritas o enumeradas de forma explícita en su proceso de desarrollo. En algunos puntos de la tabla se han indicado las técnicas concretas que proponen, si las mismas no son estándares o son variantes de técnicas conocidas.

**Tabla 3.2** Técnicas por metodología contempladas en cada actividad.

		WSDM	SOHDM	RNA	HFPM	COHDM	UWE	W2000	UWA	NDT	DDP
Captura	Entrevistas	✓		✓			✓			✓	✓
	JAD									✓	
	Brainstorming									✓	
	Concept Mapping		Roll-Activ.								
	Casos de Uso					✓					
	Cuestionarios/Checklist						✓				
	Prototipos										✓
Otras Técnicas		DFD									
Definición	Lenguaje Natural	✓		✓	✓		✓			✓	
	Glosarios						✓			✓	
	Patrones/plantillas							✓	✓	✓	
	Escenarios		SAC				✓				
	Casos de Uso				✓	✓	✓	✓	✓	✓	
	Lenguaje Formal								XML		
	Sketches de interfaz				✓						
Prototipos										✓	
Otras Técnicas		Lista evento				UIDs		Grafo refinamiento req.	Frase s en BNL		
Validación	Reviews/Walk-throughs						✓			✓	
	Auditorias						✓				
	Matriz trazabilidad									✓	
	Prototipos					✓	✓				✓
	Otras Técnicas								Grafo refinamiento req.		

Se pueden sacar varias conclusiones de esta tabla comparativa. Se puede observar que en la fase de captura de requisitos la técnica más utilizada es la de las

entrevistas. Por otro lado, se ve que hay varias metodologías en las que no se recomienda ninguna técnica en concreto para la realización de la captura.

### 3.3.3 Grado de detalle en las descripciones

La base del siguiente estudio comparativo tiene su base en la forma en que las metodologías propuestas definen el tratamiento de requisitos.

Las propuestas se evaluarán de la siguiente manera:

- Si son **orientadas al proceso**: Es decir “si describen un proceso claro y concreto a seguir para realizar la definición, captura y validación de requisitos.”<sup>(19)</sup>
- Si son **orientadas a la técnica**: Es decir “si describen técnicas concretas a aplicar durante el proceso.”<sup>(20)</sup>
- Si son **orientadas al resultado**: Es decir “si da una descripción concreta del producto que se debe obtener tras el proceso de desarrollo.”<sup>(21)</sup>

El peso dado a cada técnica corresponde con la siguiente evaluación:

#### ***Orientación al proceso:***

- (+) describe con claridad los pasos a seguir.
- (o) describe el proceso sin detallarlo.
- (-) no describe ningún proceso.

---

(19), (20), (21) <http://www.lsi.us.es/docs/informes/LSI-2002-4.pdf>, Consulta: junio 2009, Actualización: 2008

- **Orientación a la técnica:**
  - (+) describe con claridad las técnicas y la forma de aplicarlas.
  - (o) enumera las técnicas a aplicar.
  - (-) no propone ninguna técnica concreta o referencia a técnicas generales.
  
- **Orientación al producto:**
  - (+) describe con claridad la estructura del producto a obtener.
  - (o) describe el contenido del producto sin entrar en detalle de su estructura.
  - (-) no comenta nada sobre el producto resultante.

**Tabla 3.3:** Orientación de las Propuestas

	Orientación al proceso	Orientación a la técnica	Orientación al resultado
WSDM	O	-	-
SOHDM	-	+	-
RNA	+	-	-
HFPM	+	O	+
OOHDM	O	+	-
UWE	+	O	O
W2000	O	O	-
UWA	+	O	-
NDT	O	+	+
DDDP	+	O	-

Analizando este cuadro comparativo podemos notar tres características importantes, la primera es que ninguna propuesta describe totalmente el proceso, las técnicas y el formato del producto a conseguir, sin embargo se puede notar una gran tendencia al proceso.

Finalmente del estudio comparativo realizado se ha podido observar que las propuestas expuestas sobre todo en la fase de definición de requisitos tienden por un lado a hacer todo de manera grafica como UWE por ejemplo y por otro lado existen tendencias de con representación mas textual como es el caso de SOHDM.

Sin embargo aun cuando no se ha demostrado en las tablas comparativas anteriormente presentadas es importante mencionar la falta de herramientas CASE que brindan apoyo a aplicaciones dirigidas para la Web en el tratamiento de requisitos.

Existen muy pocas opciones de soporte para la realización de requisitos, existiendo únicamente algunas aproximaciones como ArgoUWE para UWE y UWA o NDT-Tool para NDT.

Por lo tanto basados en este análisis y tomando en cuenta que una herramienta CASE es un gran apoyo en el desarrollo no solo de la definición de requerimientos sino en la elaboración de los distintos diagramas que se deben elaborar, hemos escogido para el desarrollo de nuestro proyecto a la Metodología denominada UWE

---

(22):<http://uwe.pst.ifi.lmu.de/teachingCourseMaterial.html>(1), Consulta: 2009, Actualización: 2008

apoyados por la herramienta case desarrollada para esta metodología denominada ArgoUWE.

## 3.4 UWE

### 3.4.1 Características

- UWE es una metodología orientada a objetos, iterativa e incremental basada en UML. <sup>(22)</sup>
- Se basa también en el proceso de desarrollo de software unificado (Unified Software Development Process, UP).
- Proporciona un diseño sistemático y uno de generación semiautomática en las aplicaciones Web a través de un framework de publicación XML (UWEXML). <sup>(23)</sup>
- UWE define su propio perfil UML en el cual se definen todos los elementos necesarios para modelar los diferentes aspectos de una aplicación Web que son: la presentación, la navegación, entre otros.
- En esta metodología se proponen dos tipos de diagramas para el modelado de la navegación que son: el modelo de espacio, en el cual se definen todos los caminos navegacionales, es decir todas las aquellas asociaciones de navegación directa entre todos los distintos objetos de la aplicación más bien conocidos como clases de navegación, y el segundo modelo de estructura de navegación el cual define la estructuras de acceso que son utilizadas en la navegación, es decir todo aquello referente a menús, índices y demás

---

(23): [http://uwe.pst.ifi.lmu.de/teachingCourseMaterial.html\(1\)](http://uwe.pst.ifi.lmu.de/teachingCourseMaterial.html(1)), Consulta: abril 2009, Actualización.:

## **3.4.2 Actividades del Modelado**

### **3.4.2.1 Análisis de requerimientos**

Dentro de la metodología UWE el primer paso es el análisis de requerimientos, en donde se determinan los requerimientos funcionales de todo el sistema. <sup>(24)</sup>

La Metodología UWE separa en tres fases el análisis de requerimientos que son: Fase de Captura, Fase de definición y Fase de validación. <sup>(25)</sup>

Como resultado final se obtiene un modelo de casos de uso y la documentación que describe a los usuarios, los casos de uso, la interfaz y las reglas de adaptación.

Esta metodología clasifica los requisitos en dos grupos que son: requisitos funcionales y requisitos no funcionales.

Los requisitos funcionales que UWE trata son:

- Requisitos de relación con el contenido
- Requisitos de relación con la estructura
- Requisitos de relación con la presentación
- Requisitos de relación con la adaptación
- Requisitos de relación con los usuarios

---

(24),(25) [http://uwe.pst.ifi.lmu.de/teachingCourseMaterial.html\(1\)](http://uwe.pst.ifi.lmu.de/teachingCourseMaterial.html(1)) <http://www.geocities.com> Consulta: abril 2009.

- **Fase de Captura:**

La Fase de Captura de requisitos es aquella en la que las personas que conforman el equipo de desarrollo de una aplicación extraen toda la información necesaria para cubrir las necesidades del sistema a desarrollarse de diferentes tipos de fuentes de información disponibles. Durante la captura de requerimientos de un sistema siempre el desarrollador se ha encontrado con una tarea compleja, en relación a esto se han ido creando varias técnicas que permitan realizar este proceso de una forma mucho más sencilla, eficiente, y precisa. Para la fase de captura de requisitos de un sistema Web la metodología UWE propone entrevistas, cuestionarios, checklist y casos de uso

- **Entrevistas:** Con esta técnica se puede extraer información de una manera muy sencilla y sobre todo natural, sin embargo es importante que el entrevistador sea alguien de mucha experiencia para así poder elegir bien a los entrevistados y para obtener de los mismos la mayor cantidad y calidad de información que sea posible dentro de un período de tiempo limitado.
- **Casos de Uso:** Son una herramienta ampliamente utilizada para descubrir y registrar los requisitos de un sistema, sobre todo aquellos requisitos funcionales que existirán en el mismo. Por medio de los casos se describirá por completo el sistema teniendo así un mejor entendimiento de que es lo que este va a realizar en cada una de sus distintas funciones sin revelar la estructura interna de dichas funciones.

Los casos de uso permiten indicar el contorno y alcance de un sistema expresados como actores y requisitos funcionales como casos de uso respectivamente. Los actores son todos aquellos elementos externos que interactúan con el sistema por ejemplo: usuarios, clientes, otro sistema, etc.

Los casos de uso muestran la secuencia de interacciones producidas entre los actores y el sistema para realizar una determinada actividad o función.

- **Cuestionarios y Checklists:** Esta técnica consiste en la redacción de un documento que consiste en preguntas concretas y cortas, o preguntas cerradas (checklist) para obtener información de manera independiente.

#### - Fase de Definición de Requisitos

Con respecto a la Fase de definición de requisitos UWE propone los escenarios, el glosario y los casos de uso

- **Glosario y ontologías:** Consiste en la creación de un glosario de términos en el que están las definiciones más críticas e importantes del sistema.
- **Escenarios:** Esta técnica se basa en la descripción de todas las características del sistema a desarrollar por medio de una secuencia de pasos.

- **Casos de uso:** Esta técnica es la más utilizada en la fase de definición de requisitos sin embargo, puede resultar muy ambiguo por lo que para eliminar esa ambigüedad se utilizan plantillas o diccionarios de datos.

Para el desarrollo del Proyecto GoldCard se utilizaran los Casos de Uso para las Fases de Captura y Definición de Requisitos respectivamente.

#### - Fase de Validación de Requisitos

Finalmente para la fase de validación de requisitos UWE propone los walk-throughs, prototipos y las auditorias:

- **Reviews o Walk-throughs:** Está técnica consiste en modelar la definición de requisitos es decir la correcta y completa lectura y corrección de la documentación.
- **Auditorias:** Esta técnica de validación consiste en revisar la documentación mediante un checklist ya definida a comienzos del proceso.
- **Prototipos:** Esta técnica se basa en la obtención de la definición de requisitos prototipos permitiendo a los usuarios tener una idea de la estructura de la interfaz del sistema con el usuario sin tener la total funcionalidad del mismo.

En el Proyecto GoldCard esta fase se definirá mediante Prototipos.

### 3.4.2.2 Diseño Conceptual

Una vez realizado todo el análisis de requerimientos se realiza un modelo conceptual. El modelo conceptual está basado en el análisis de requerimientos, el cual, incluye los objetos involucrados en la interacción entre el usuario y la aplicación, este modelo incluye todas las actividades que el usuario va a realizar dentro de la aplicación Web.

Para la construcción de este modelo se utilizará un diagrama de clases, el cual nos ayuda a representar de forma gráfica un modelo conceptual que muestre una colección de todos los elementos estáticos dentro de un dominio.

- **Diagrama de clases:** Este tipo de diagrama es muy importante en el análisis y diseño de un sistema, aquí se describe la estructura del sistema mostrando sus clases, sus respectivos atributos y sus relaciones. Un diagrama de clases es desarrollado a través de la información obtenida de los Casos de Uso, Diagramas de Secuencia y Colaboración

Un diagrama de clases se compone de los siguientes elementos:

- **Clase:** Una clase es aquella que describe un conjunto de objetos con sus propiedades (atributos, métodos y visibilidad.)
- **Relaciones:** Una herencia es un enlace entre los diferentes elementos del diagrama. (Herencia, Composición, Agregación y Asociación.)

### Ejemplo:

Una Clase Auto tiene como Atributos: placa, año y modelo. Y como operaciones: IngresarAuto, EliminarAuto, ModificarAuto.



**Figura 3.1** Ejemplo de una clase

### Atributos:

Los atributos son aquellos que identifican las características propias de una Clase, un atributo se define según su visibilidad la misma que indica el nivel de acceso que tienen las demás clases a los datos y operaciones definidas, y podrán ser del tipo:

 **Public (+):** Visible para todos los clientes de la clase, es decir que el atributo será visible tanto dentro como fuera de la clase. (26)

 **Private (-):** Visible solo para la propia clase, es decir que el atributo sólo será accesible desde dentro de la clase o que sólo sus métodos lo pueden acceder.(27)

 **Protected (#):** Visible únicamente para subclases, es decir que el atributo no puede ser accesible desde fuera de la clase, pero si por métodos de la clase además de las subclases que se deriven.(28)

---

(26),(27),(28): [http://uwe.pst.ifi.lmu.de/teachingCourseMaterial.html\(1\)](http://uwe.pst.ifi.lmu.de/teachingCourseMaterial.html(1)), Consulta: julio 2009,

Actualización.: abril 2008

## Operaciones:

Las operaciones de una clase son las que describen el comportamiento de una clase, también se definen según su visibilidad como:

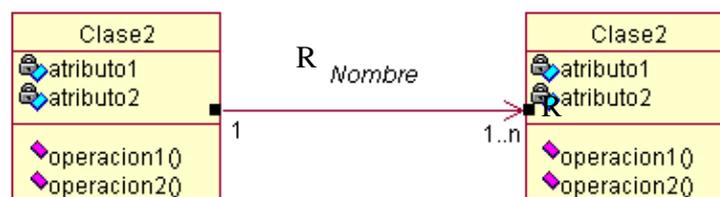
 **Public (+)**: Quiere decir que la operación será visible tanto dentro como fuera de la clase. (29)

 **Private (-)**: La operación será accesible únicamente desde dentro de la clase, es decir que únicamente aquellas operaciones de la clase pueden accederla.(30)

 **Protected (#)**: La operación únicamente podrá ser accesible por métodos de la clase además de métodos de las subclases que se deriven. (31)

## Relaciones entre Clases:

Las relaciones entre clases nos permiten indicar la interrelación que existe entre dos o varias clases, estas están compuestas por las siguientes características:



**Figura 3.2** Características de una relación entre clases

- **Navegabilidad:** Indica el sentido de la navegación entre las clases, se representa mediante la cabeza de una flecha

(29),(30),(31): [http://uwe.pst.ifi.lmu.de/teachingCourseMaterial.html\(1\)](http://uwe.pst.ifi.lmu.de/teachingCourseMaterial.html(1)), Consulta: 2009, Actualización.:

- **Rol:** Indica el papel que desempeña una clase en la asociación
- **Multiplicidad:** Es un rango o colección de números que se colocan en el extremo adecuado respectivamente.

**Tabla 3.4** Tipos de Multiplicidad de una relación entre clases

Nombre	Representación
Varios	*
Uno	1
Uno a muchos	1...*
Cero	0
Cero a muchos	0
Número fijo	m (denota el numero)

### Tipos de Relaciones:

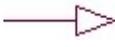
- **Asociación:** 

Representa una relación genérica entre dos clases, es decir nos permite asociar objetos que colaboran entre sí, no es una relación fuerte ya que el tiempo de vida de un objeto no depende del otro.

### Ejemplo

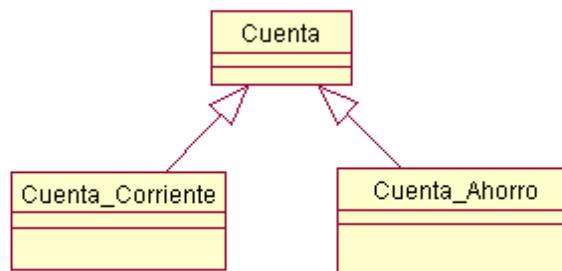


**Figura 3.3** Ejemplo de una relación de asociación.

- **Herencia (Especialización/Generalización):** 

Una herencia es una relación entre una clase Padre o Superclase y una clase más específica de la misma es decir una clase hijo o subclase.

**Ejemplo**



**Figura 3.4** Ejemplo de una relación de herencia

- **Agregación:** 

Existen diferentes tipos de relaciones de asociación que nos permiten mejorar el modelamiento de nuestro diagrama de clases permitiéndonos explicarlo de forma más explícita, una de ellas es la agregación la misma que nos permite incluir un objeto en una clase, la relación de agregación se puede derivar en dos tipos :

- o **Por Valor (Composición):** este es un tipo de relación en la que el tiempo de vida del objeto incluido se ve condicionado por el tiempo de vida del que lo incluye. Su representación gráfica se denota por un rombo relleno

○ **Por Referencia (Agregación):** Es una relación en donde el tiempo de vida del objeto incluido es independiente del que lo incluye. Su representación gráfica se denota por un rombo sin relleno.

- **Dependencia o Instanciación (uso):** - - - - -

Esta relación se utiliza cuando la clase instanciada es dependiente de otro objeto o clase. Una relación de dependencia denota que si se realiza un cambio en la especificación de un elemento de una clase este puede afectar a algún otro elemento de otra clase que lo utiliza pero no necesariamente a la inversa. Su representación gráfica es una línea discontinua dirigida hacia el elemento del que depende.

**Ejemplo:**



**Figura 3.5** Ejemplo de una relación de dependencia.

UWE construye un modelo conceptual el cual no procura hacer **demasiado** énfasis en los aspectos de navegación y aspectos de iteración de la aplicación Web ya que estos pasos son llevados cabo por el modelo navegacional y el modelo de presentación del diseño respectivamente.

### 3.4.2.3 Diseño Navegacional:

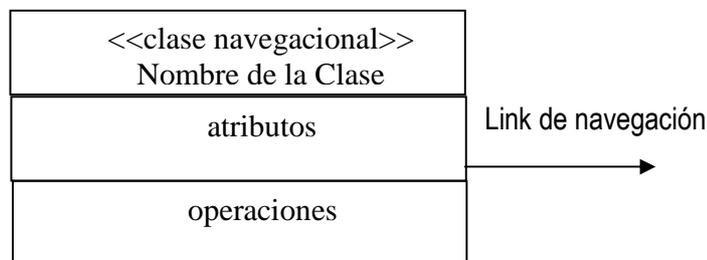
El diseño navegacional en UWE tiene como metas diseñar las trayectorias de navegación, representar nodos y links de la estructura hipertexto y evitar la desorientación y la sobrecarga cognoscitiva arrojando como resultado el modelo estructural de navegación el mismo que comprende: diagrama de clases estereotipado y elementos de específicos de modelado

#### - Elementos básicos de navegación

- **Clase de navegación:** Especifica los nodos que van a ser visitados por el usuario a través del browser. Una clase navegación tendrá el mismo nombre que la clase dominio. Para su representación se usa el estereotipo UML <<Navigation Class>>
- **Link de navegación:** especifica que el objeto navegacional destino es accedido por navegación desde el objeto navegacional fuente.

#### - Estereotipos para los elementos básicos de navegación

- Clase navegacional
- Link de navegación



**Figura 3.6** Estereotipos para los elementos básicos de navegación

## - Elementos adicionales de navegación

<<index>>



Un índice permite el acceso directo a instancias de una clase de navegación. Este es modelado por un objeto compuesto el cual contiene un número arbitrario de ítems índice. Cada ítem índice es a su vez un objeto el cual tiene un nombre que identifica la instancia y posee un link a una instancia de una clase de navegación. (32)

<<guidedTour>>



Un tour guía proporciona acceso secuencial a instancias de una clase navegación. Para las clases, las cuales contienen objetos tour guía usamos el estereotipo <<guidedTour>> y su correspondiente icono, las guías tour pueden ser controlados por un usuario o por el sistema. (33)

<<query>>



Un Query es modelado por una clase la cual tiene un query string como un atributo. Para las clases query utilizamos el estereotipo <<query>> cualquier clase query es la fuente de dos asociaciones directas relacionadas por el constraint (XOR).<sup>(34)</sup> De esta forma el resultado es un query con varios objetos modelado para llevar primero un índice soportando la selección de una instancia particular de una clase navegación.

---

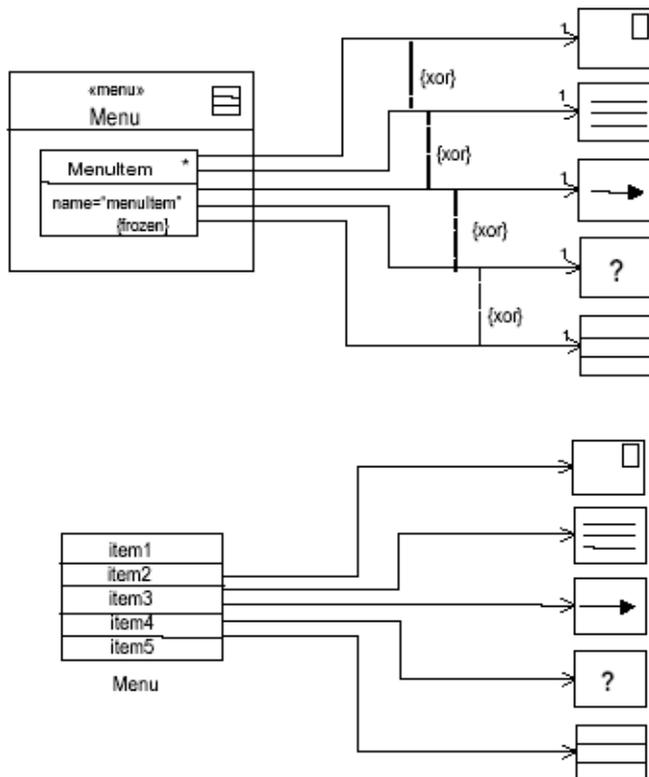
(32),(33),(34):[http://uwe.pst.ifi.lmu.de/teachingCourseMaterial.html\(1\)](http://uwe.pst.ifi.lmu.de/teachingCourseMaterial.html(1)),Consulta: 2009, Actualización.:

El resultado del query puede ser utilizado alternativamente como entrada para un tour guía

<<menú>>



Un menú es un índice de para un set de elementos, tales como tour guía, un query, una instancia de una clase navegación u otro menú.(35)Este es modelado por un objeto compuesto que contiene un número fijo de ítems de menú. Cada ítem menú tiene un nombre constante y posee un link a una instancia de una clase de navegación o a un elemento de acceso.



**Figura 3.7** Ilustración de una clase menú con sus ítems y sus respectivos links.

(35): [http://uwe.pst.ifi.lmu.de/teachingCourseMaterial.html\(1\)](http://uwe.pst.ifi.lmu.de/teachingCourseMaterial.html(1)), Consulta: 2009, Actualización.: enero

#### 3.4.2.4 Diseño de Presentación

El diseño de presentación tiene como objetivo la representación de la estructura y comportamiento de la interfaz de usuario.

Un diagrama de secuencia de UML demuestra la interacción de elementos dispuesta en orden temporal. Presenta los objetos que participan en la interacción y la secuencia de los mensajes enviados entre ellos. UWE propone el uso de los diagramas de secuencia para representar los aspectos dinámicos de la navegación, es decir, las secuencias describen la realización de los casos de uso. <sup>(36)</sup> De esta manera, los diagramas de secuencia proveen una representación funcional centrada en el tiempo del modelo de navegación

Un diagrama de estados de UML denota una secuencia de los estados que un objeto puede adquirir durante su vida, junto con acciones responsivas, disparando eventos y las condiciones asociadas para indicar transiciones. UWE da otro sentido a los diagramas de estados del UML puro ya que los utiliza para visualizar escenarios de navegación. <sup>(37)</sup> Estos diagramas permiten detallar la parte dinámica del modelo de navegación, especificando los eventos que disparan las situaciones, definen condiciones y explícitamente incluyen las acciones que son realizadas. Junto con el modelo de interacción temporal, los escenarios Web proveen la representación funcional dinámica del modelo de navegación.

---

(36), (37): [http://uwe.pst.ifi.lmu.de/teachingCourseMaterial.html\(1\)](http://uwe.pst.ifi.lmu.de/teachingCourseMaterial.html(1)), Consulta: 2009, Actualización.:

## CAPITULO IV: Fase de Análisis y Diseño

### 4.1 Conceptualización del Sistema

Este sistema es desarrollado con el fin de administrar el proceso logístico, operativo y administrativo del producto tarjeta de descuento GOLDCARD desarrollado como un sistema para tarjeta habientes.

GOLDCARD es una tarjeta que cubrirá un descuento del 50% en el cambio de aceite de motor durante todo un año, con cupo de 7 cambios para la clase familiar y 16 cambios para la clase comercial además de un 10% de descuento en lo referente mano de obra. (38)

Tenemos 4 niveles de inversión con DESCUENTO DEL 50% + 10% cobertura de 365 días que se detallan a continuación:

**Tabla 4.1** Tabla de Tipos de Tarjeta GoldCard

Tipo de Tarjeta	Precio	#de cambios (anuales)	Aplicable a
FAMILIAR JUNIOR	\$34.90	7	Automóviles de cilindraje de 1800 CC
SUPER FAMILIAR	\$44.90	7	Camionetas, Vans, Jeep
COMERCIAL JUNIOR	\$74.90	16	Automóviles de Servicio Público
SUPER COMERCIAL	\$89.90	16	Camionetas, Cooperativas

(38): Documentación, proceso lógico de administración de la tarjeta GoldCard de la empresa Solsupply

## 4.2 Análisis de Requerimientos

### 4.2.1 Objetivos

- Obtener un sistema versátil, orientado hacia la Web para el proceso de negocios de la empresa.
  - Implantar un sistema eficaz, eficiente, de fácil manejo, para los usuarios finales, y sus administradores.
  - Proporcionar a la gerencia una herramienta de negocios,
  - Proveer posibles errores de ingreso de información; con un sistema de auditoría y validación de campos, permitiendo que la información ingresada sea confiable.
  - Conocer en forma precisa las fuentes de desarrollo para implementar las siguientes fases, con el fin de seguir perfeccionando el sistema de desarrollo.
- 
- **Proceso Operativo:**

El proceso operativo es el siguiente:

Se llena un formulario que debe ser ingresado a una base de datos de posibles clientes.

Del proceso de ingreso se extraerá datos para realizar un tele mercadeo, que permita a la fuerza de ventas, realizar un seguimiento hasta la consecución de la venta.

Luego de realizado el proceso de tele mercadeo y ejecutado la venta, se procede a la facturación y emisión de la tarjeta GOLD CARD con los datos básicos del cliente, más una identificación única e intransferible, que permita identificar al tarjeta habiente o cliente.

- **Proceso logístico**

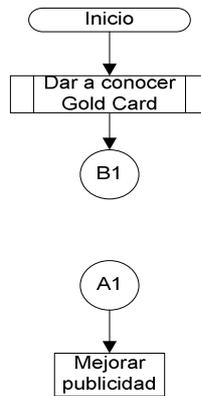
El manejo logístico de la tarjeta se describe en el Anexo 4.1. Es importante mencionar que los centros se conectaran con nuestro servidor para solicitar una autorización de servicio, si es el caso de que la tarjeta sea convenio para pago de empresa, tipo crédito o convenio.

Los productos descritos en el proceso son estrictamente canalizados por la escala de clientes por el mecanismo padre e hijos, es decir cada tarjeta habiente tendrá dos usuarios referidos para hacer merecedor al descuento.

A continuación se describirá por completo el proceso GoldCard:

**- Proceso A**

**Encargado:** Personal de Marketing



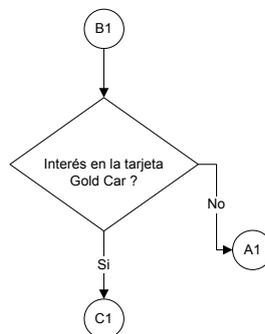
**Figura 4.1** Proceso Logístico GoldCard (Proceso A)

El personal de Marketing dará a conocer la Tarjeta GoldCard a través de distintas estrategias al cliente para la adquisición y uso de la misma.

En caso de que no funcionara alguna de las estrategias establecidas se deberá mejorar el Marketing.

**- Proceso B**

**Encargado:** Cliente

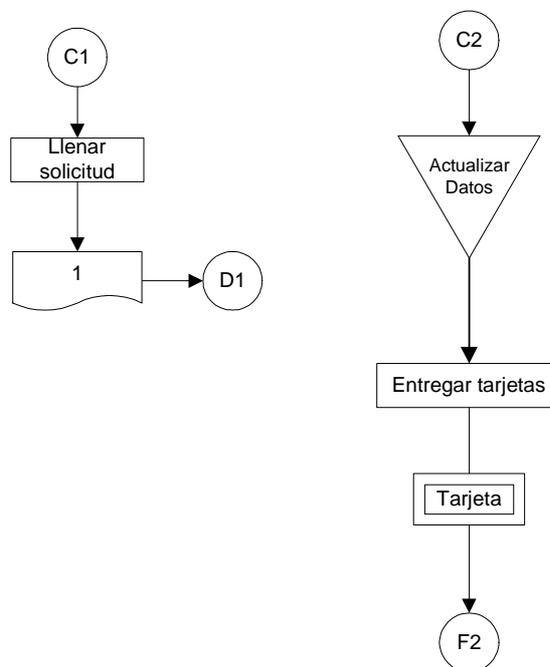


**Figura 4.2** Proceso Logístico GoldCard (Proceso B)

Una vez realizado el proceso de Marketing el cliente está en disposición de adquirir o no la Tarjeta GoldCard, si el cliente decide adquirir la tarjeta deberá pasar al siguiente proceso (C1), caso contrario se ejecutará el proceso referente a la mejora de publicidad (A1).

### - Proceso C

**Encargado:** Personal de atención al cliente



**Figura 4.3** Proceso Logístico GoldCard (Proceso C)

Si el cliente está interesado en adquirir la tarjeta deberá llenar una solicitud



(39) con sus datos personales. Datos que serán ingresados al sistema en el siguiente proceso (D1).

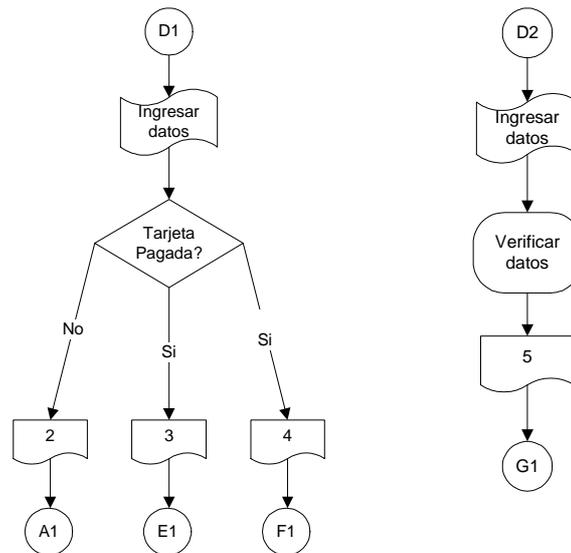
---

(39): Solicitud de Inscripción

El proceso C2 permite la actualización de datos en el sistema una vez que la tarjeta ha sido cancelada en su totalidad e impresa para su posterior entrega. El monto total cancelado por la tarjeta permite la ejecución del siguiente proceso (F2).

**- Proceso D**

**Encargado:** Personal de Sistemas



**Figura 4.4** Proceso Logístico GoldCard (Proceso D)

Los datos del cliente son ingresados al sistema en donde se constata el pago total de la Tarjeta GoldCard. Si el cliente realizó su pago se emitirán dos documentos

 (40) y  (41) los cuales permitirán el desarrollo de los procesos (E1) y

(F1) respectivamente. Caso contrario se emitirá un documento  (42) el cual permitirá el desarrollo del proceso (A1).

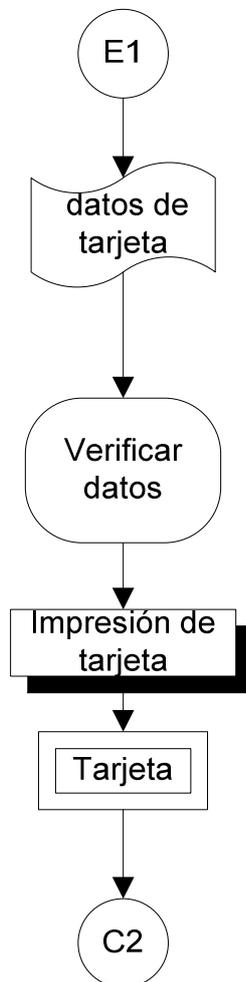
---

(40) Datos que van impresos en la tarjeta  
 (41) Lista de clientes y vehículos  
 (42) Documento

El proceso D2 permite el ingreso y verificación de datos luego del cobro de una factura teniendo como resultado un reporte  (43) el mismo que permitirá la ejecución del siguiente proceso (G1).

### - Proceso E

**Encargado: Imprenta de Tarjetas**



**Figura 4.5** Proceso Logístico GoldCard (Proceso E)

---

(43): Historial del vehículo

Se envían los datos que han de ser impresos en la tarjeta, se verifican y se procede a la impresión de la tarjeta lo cual permite la ejecución del siguiente proceso (C2).

#### **- Proceso F**

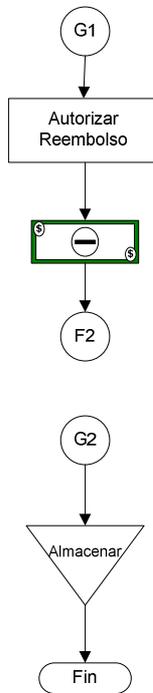
**Encargado:** Centros Automotores. El diagrama para Centros Automotores o (Proceso F) se puede visualizar en el Anexo 4.2

Una vez adquirida la tarjeta, el cliente podrá hacer uso de ella en cualquiera de los centros autorizados. Se verificará el número de tarjeta del cliente previo a cualquier atención o servicio.

Si el cliente desea realizar un cambio de aceite se le facturará con el 50% de descuento, caso contrario si el cliente desea realizar cualquier otro servicio se le facturará con un descuento del 10% en mano de obra. Los datos serán almacenados para poder obtener un historial de vehículos lo cual permitirá a cada uno de los centros proceder a la facturación referente al reembolso correspondiente al cambio de aceite.

## - Proceso E

**Encargado:** Gerencia Solsupply



**Figura 4.6** Proceso Logístico GoldCard (Proceso G)

Una vez facturado el historial de clientes que realizaron cambios de aceite, dichas facturas serán enviadas a la gerencia de Solsupply para autorizar el reembolso correspondiente a cada uno de los centros automotrices a la vez que se procederá a almacenar los mismos.

- **Proceso Administrativo:**

El proceso administrativo dentro de la compañía está orientado hacia el manejo logístico, operativo, y la comunicación mediante un servidor WEB 24 horas en línea, que permita los usuarios en un futuro, hacer su compra on-line.

El proyecto conlleva un involucramiento en el proceso de manejo de un servidor WEB más un servidor local de administración del sistema, separando los centros autorizados y la matriz, emisora de la tarjeta GOLD CARD.

#### **4.2.3 Proceso Actual de manipulación de la información**

Actualmente el proceso de manipulación de la información se lo realiza de forma manual en cada uno de los centros cuyo proceso se muestra en el Anexo 4.3.

### **4.3 Fase de Diseño del Proyecto GoldCard**

#### **4.3.1 Arquitectura Web de la Aplicación GoldCard**

A continuación muestra los Lenguajes y Herramientas aplicadas al Sistema GoldCard según el nivel de aplicación n-capas

- **SERVICIOS DE PRESENTACION**
  - MICROSOFT VISUAL STUDIO 2005/ASP. NET
- **SERVICIOS DE NEGOCIO**
  - WEB SERVICES
- **SERVICIOS DE DATOS**
  - SQL 2005

En el cuadro a continuación se describe el lenguaje de aplicación utilizado para el Sistema GOLD CAR en cada nivel dentro de la arquitectura:

**Tabla 4.2:** Arquitectura de la Aplicación GoldCard

Nivel		Lenguaje
Cliente	Aplicación Cliente	ASP.NET HTML JAVASCRIPT
Servidor	Servidor Web	WINDOWS 2003 SERVER
	Aplicación Servidor	ASP .NET
	Aplicación Web Service	ASP .
Servidor de Datos	Base de Datos	SQL Server 2005

### 4.3.2 Especificación de Actores

**Tabla 4.3:** Descripción y Actividades del Usuario Súper Administrador

Actor	Descripción	Actividades
<b>Usuario Súper Administrador</b>	<b>Este usuario tiene acceso a todos los módulos del Sistema, es quien a través del modulo de Seguridades podrá administrar usuarios y sus accesos.</b>	Ingresa a la aplicación Web Accede al menú principal a través de su clave de usuario administrador <i>Accede al link de Seguridades</i> Registra, actualiza o elimina usuarios Configura niveles de perfiles Configura cambios de contraseña <i>Accede al link de Administración</i> Registra, actualiza o elimina clientes Registra, actualiza o elimina vehículos Registra, actualiza o elimina centros automotrices Registra, actualiza o elimina datos del personal de cada uno de los centros automotrices Registra, actualiza o elimina tipos de productos Registra, actualiza o elimina productos Registra, actualiza o elimina tipos e accesorios Registra, actualiza o elimina accesorios Registra, actualiza o elimina formas de pago Registra, actualiza o elimina tipos de tarjetas GoldCard Registra, actualiza o elimina inscripciones de tarjetas GoldCard <i>Accede al link de Orden de Trabajo</i> Registra, modifica, elimina y busca Ordenes de Trabajo <i>Accede al link Reportes</i> Elabora e imprime reportes

**Tabla 4.4:** Descripción y actividades del Usuario Administrador

Actor	Descripción	Actividades
<b>Usuario Administrador</b>	<b>Este usuario tiene acceso a todos los módulos excepto al modulo de seguridades.</b>	Ingresa a la aplicación Web Accede al menú principal a través de su clave de usuario administrador <i>Accede al link de Administración</i> Registra, actualiza o elimina clientes Registra, actualiza o elimina vehículos Registra, actualiza o elimina centro automotrices Registra, actualiza o elimina datos del personal de cada uno de los centros automotrices Registra, actualiza o elimina productos Registra, actualiza o elimina accesorios Registra, actualiza o elimina formas de pago Registra, actualiza o elimina tipos de productos Registra, actualiza o elimina tipos de accesorios Registra, actualiza o elimina accesorios Registra, actualiza o elimina tipos de tarjetas GoldCard Registra, actualiza o elimina inscripciones de tarjetas GoldCard <i>Accede al link Orden de Trabajo</i> Registra, modifica y elimina Ordenes de Trabajo <i>Accede al Link de Reportes</i> Elabora e imprime reportes

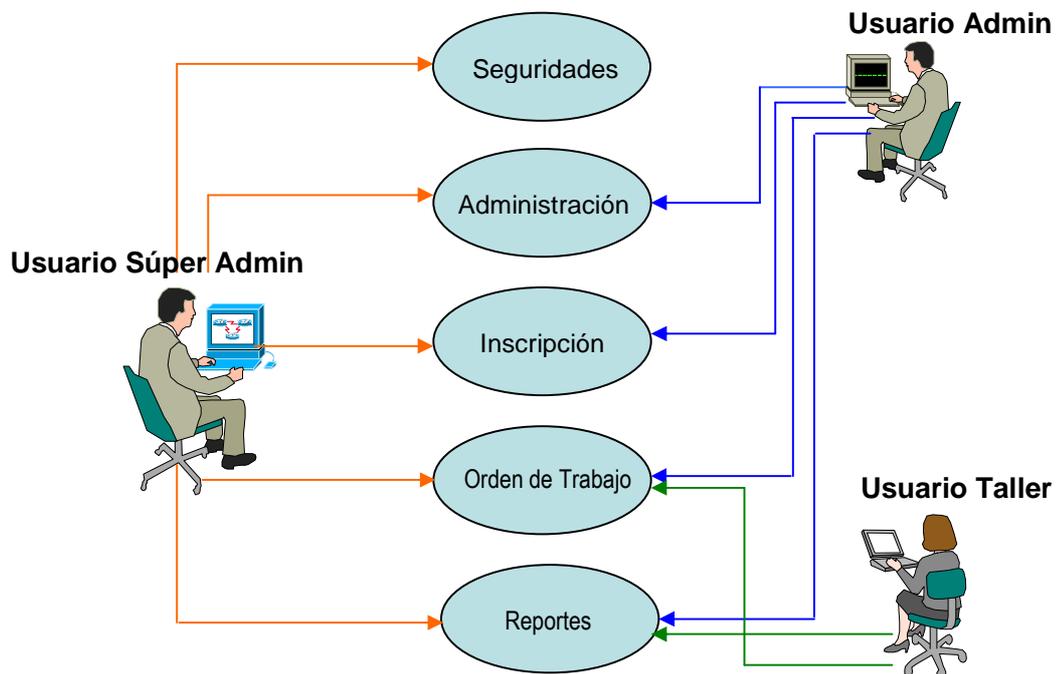
**Tabla 4.5:** Descripción y Actividades del Usuario Centro Autorizado

Actor	Descripción	Actividades
<b>Usuario Centro Autorizado</b>	<b>Este usuario tiene acceso únicamente al modulo de Orden de Trabajo.</b>	<ul style="list-style-type: none"> <li>- Ingresa a la aplicación Web</li> <li>- Accede al menú principal a través de su clave de usuario administrador</li> <li>- <i>Accede al link de Orden de Trabajo</i></li> <li>- Registra, modifica y elimina Ordenes de Trabajo</li> <li>- <i>Accede al link de Reportes</i></li> <li>- Elabora e imprime reportes</li> </ul>

## 4.4 Diagramas de Casos de Uso

### 4.4.1 Diagrama de Caso de Uso General del Sistema

La Figura 4.8 a continuación describe la funcionalidad General del Sistema Web GoldCard con sus distintos actores o usuarios y los módulos con los que interactuarán los mismos



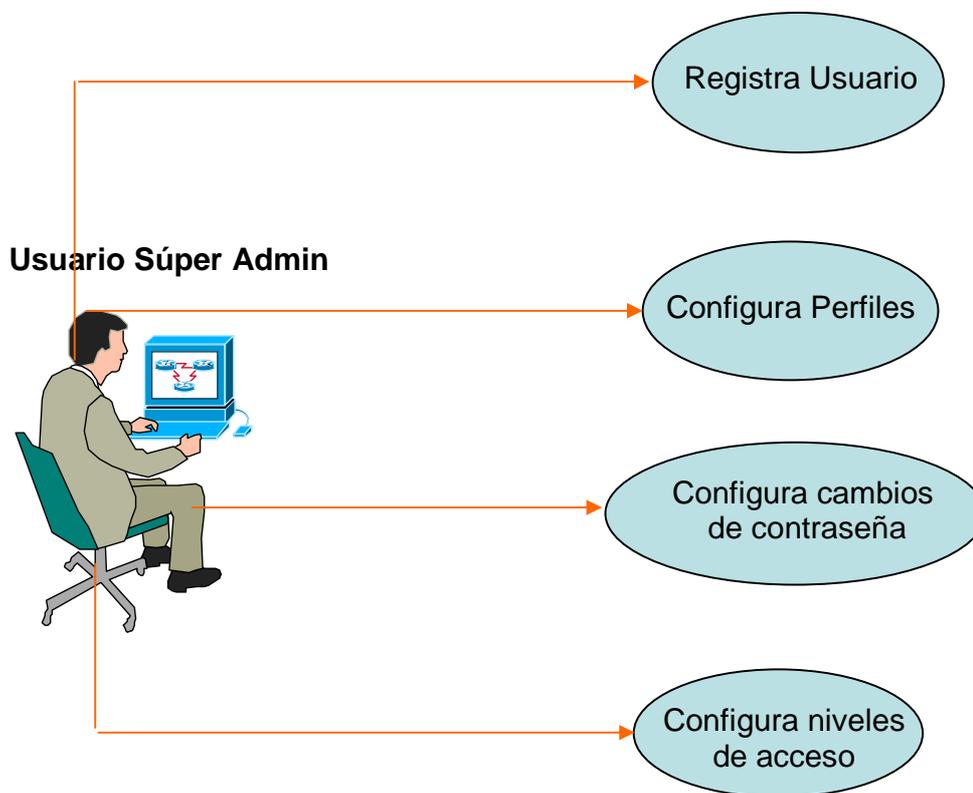
**Figura 4.8:** Diagrama de Caso de Uso General del Sistema

### 4.4.2 Diagrama de Caso de Uso por Actividades

- **Seguridades (Súper Administrador)**

Diagrama de casos de uso por actividades, que describe la funcionalidad del módulo Seguridades (Configuración y accesos del sistema), por parte del usuario

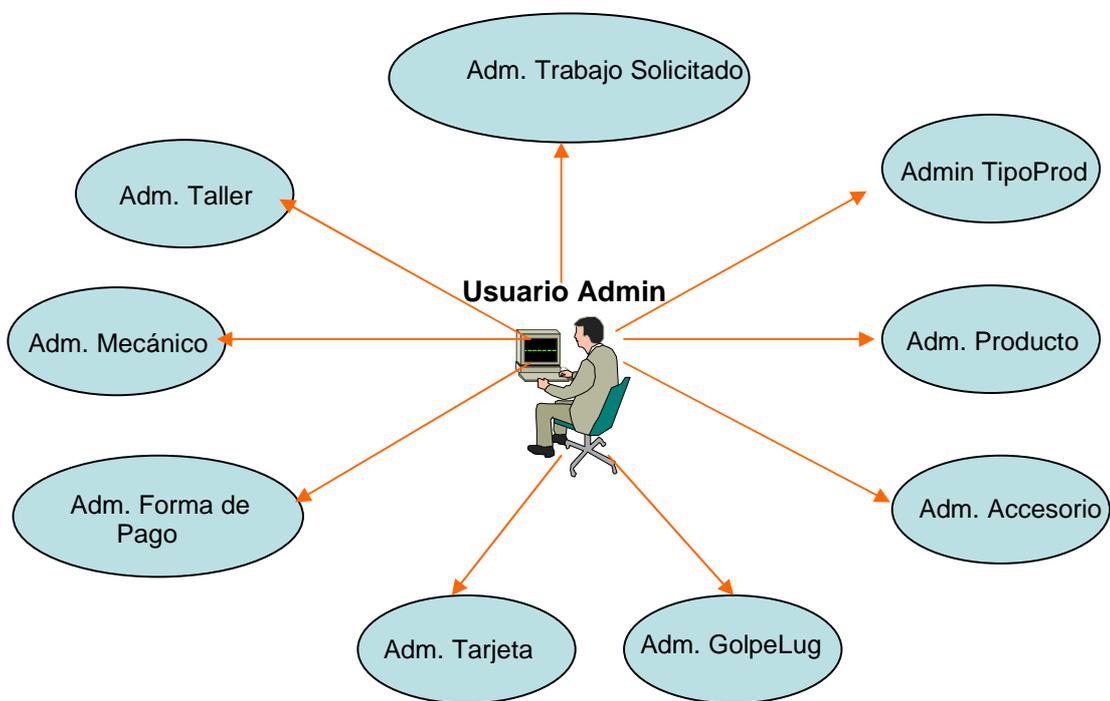
Administrador como se puede observar en la Figura 4.9 que se muestra a continuación



**Figura 4.9:** Diagrama de Caso de Uso para el usuario Súper Administrador módulo de Seguridades.

- **Administración**

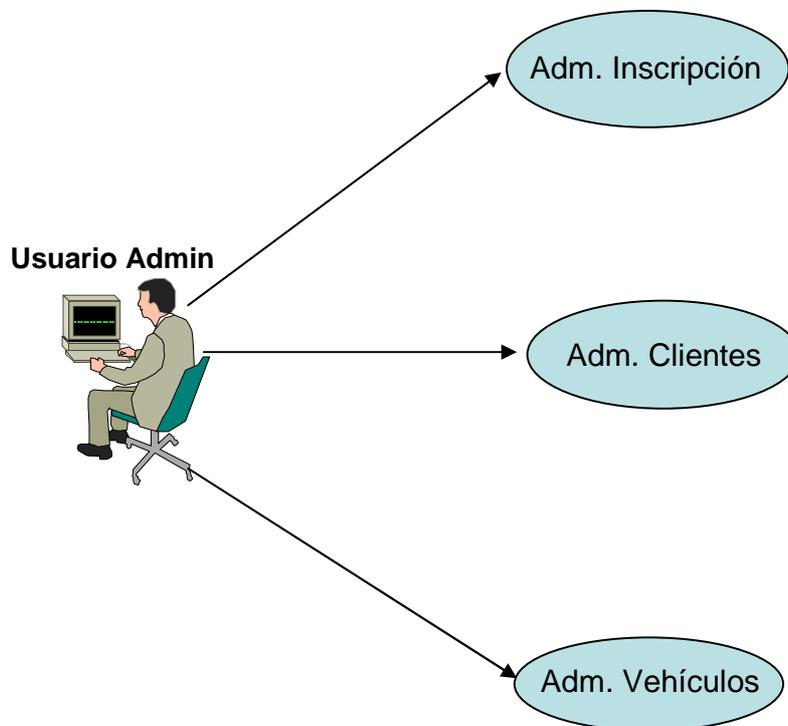
En la Figura 4.10 se visualiza el Diagrama de casos de uso por actividades, que describe la funcionalidad del módulo de Administración del sistema por parte del usuario Administrador.



**Figura 4.10:** Diagrama de Caso de Uso para el usuario Administrador módulo de Administración

- **Inscripción**

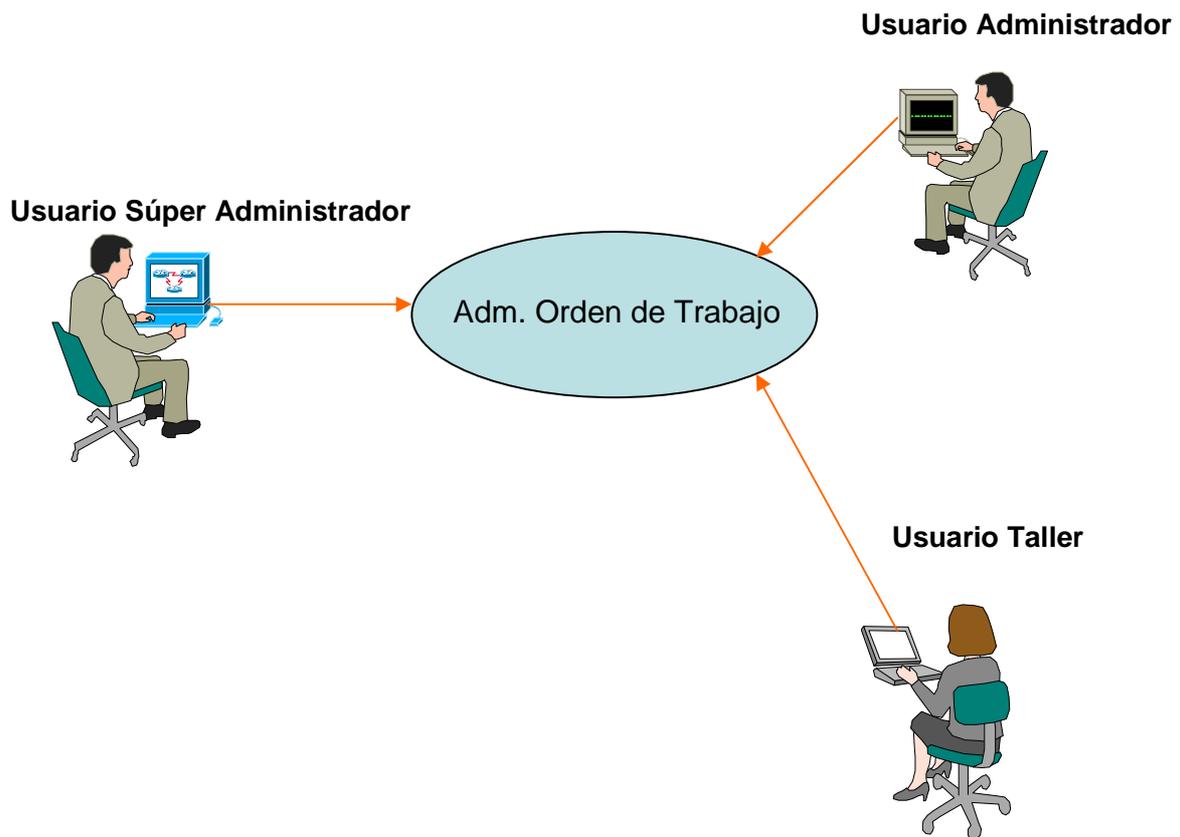
Diagrama de casos de uso por actividades, que describe en la Figura 4.11 la funcionalidad del módulo de Inscripción del sistema, por parte del usuario Súper Administrador.



**Figura 4.11:** Diagrama de Caso de Uso para el usuario Súper Administrador módulo de Inscripción

- **Orden de Trabajo**

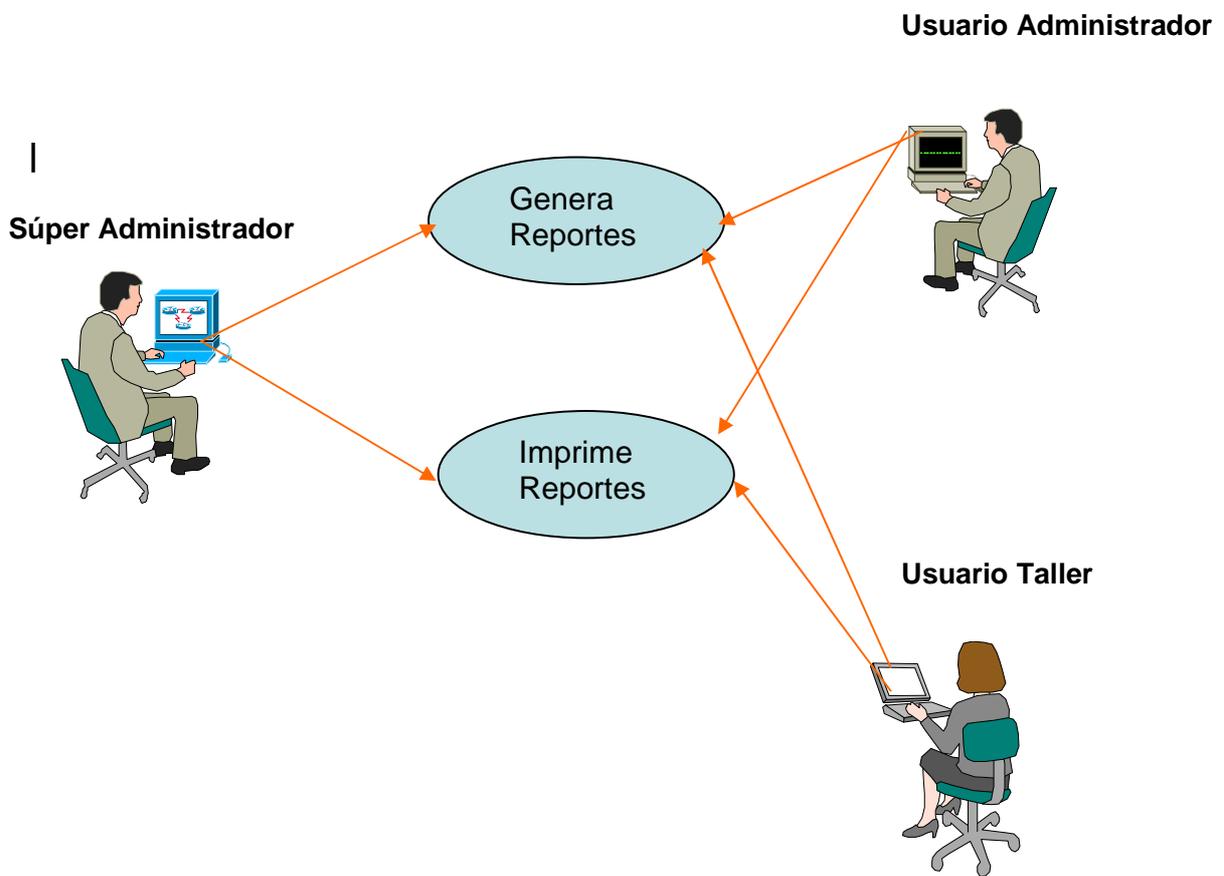
Diagrama de casos de uso por actividades, que describe en la Figura 4.12 la funcionalidad del módulo de Orden de Trabajo, por parte de los Usuarios Súper Administrador, Administrador y Taller



**Figura 4.12:** Diagrama de Caso de Uso para Orden de Trabajo.

- **Consultas y Reportes**

Diagrama de casos de uso por actividades, que describe la funcionalidad del módulo de Consulta de Reportes, por parte de los usuarios Súper Administrador, Administrador y Taller en la Figura 4.13.



**Figura 4.13:** Diagrama de Caso de Uso para los usuario Súper Administrador, Administrador y Taller del modulo de Reportes.

#### **4.5 Diccionario de Casos de Uso**

**Seguridades:** Este proceso lo realiza únicamente el Usuario Súper Administrador ingresa al sistema a través de su perfil de usuario y clave personal, accede al link de Seguridades donde tiene acceso a las opciones de registro, actualización, eliminación y configuración de perfiles, accesos y claves de usuario.

**Administración:** En este proceso el Usuario Administrador ingresa a través de su perfil y clave personal, accede a los diferentes link Administración y a los diferentes links de configuración de tablas básicas, donde se le permite las actividades de ingreso, actualización, eliminación y consulta.

**Inscripción:** Este proceso lo realiza el Usuario Administrador el cual ingresa a través de su perfil de usuario y clave personal, el usuario recibe los pedidos o inscripciones del cliente, verifica e ingresa los datos del mismo en estado pendiente hasta que el total de la tarjeta GoldCard (ingresada en la correspondiente inscripción) sea cancelada en su totalidad y emitida su factura.

**Orden de Trabajo:** Este proceso lo realiza el Usuario Taller el cual ingresa a través de su perfil de usuario y clave personal, el usuario accede al link de Orden de Trabajo ingresa una orden de trabajo de un cliente GoldCard al sistema en estado pendiente hasta la que la misma sea facturada.

**Consultas y Reportes:** Este proceso lo realiza tanto el Usuario. Administrador como el Usuario Súper Administrador, y Terminal en diferentes niveles de consulta todos

los usuarios accederán a la aplicación Web a través de su clave personal asignada, aquí se emitirán reportes.

En el Anexo 4.4 se puede observar la Tabla detallada para el Diccionario de clases de la Aplicación.

## **4.6 Diseño Conceptual**

### **4.6.1 Análisis del Diagrama de Clases de Dominio**

El Anexo 4.5 detalla a un nivel de abstracción el Diagrama de Clases de Dominio de la Aplicación Web GoldCard. Se puede observar en el Diagrama de Clases de Dominio de la Aplicación Web GoldCard las clases principales del diagrama (inscripción, Cliente, Vehículo, Tarjeta y Orden de Trabajo) que se explican a continuación

La Clase Inscripción (GCE\_Inscripcion) es de las más importantes del Diagrama esta contiene información de casi todas las demás clases tal como: Datos del Cliente (Clase GCE\_Cliente) con la cual existe una relación de composición, Datos del Vehículo (Clases GCE\_Vehiculo) la cual tiene una relación de composición con la Clase Cliente, Datos del tipo de tarjeta que adquirió el Cliente (Clase GCE\_Tarjeta) que tiene una asociación con la Clase Cliente, la forma de pago con la que se adquirió la tarjeta (Clase GCE\_FormaPago), e información propia de la Clase inscripción como el nombre del ejecutivo de ventas que realizo la venta de la tarjeta, Fecha de Emisión de la Inscripción, abono, total a pagar y datos para toma de decisiones o mercadeo a través de una pequeña encuesta agregada al formulario de

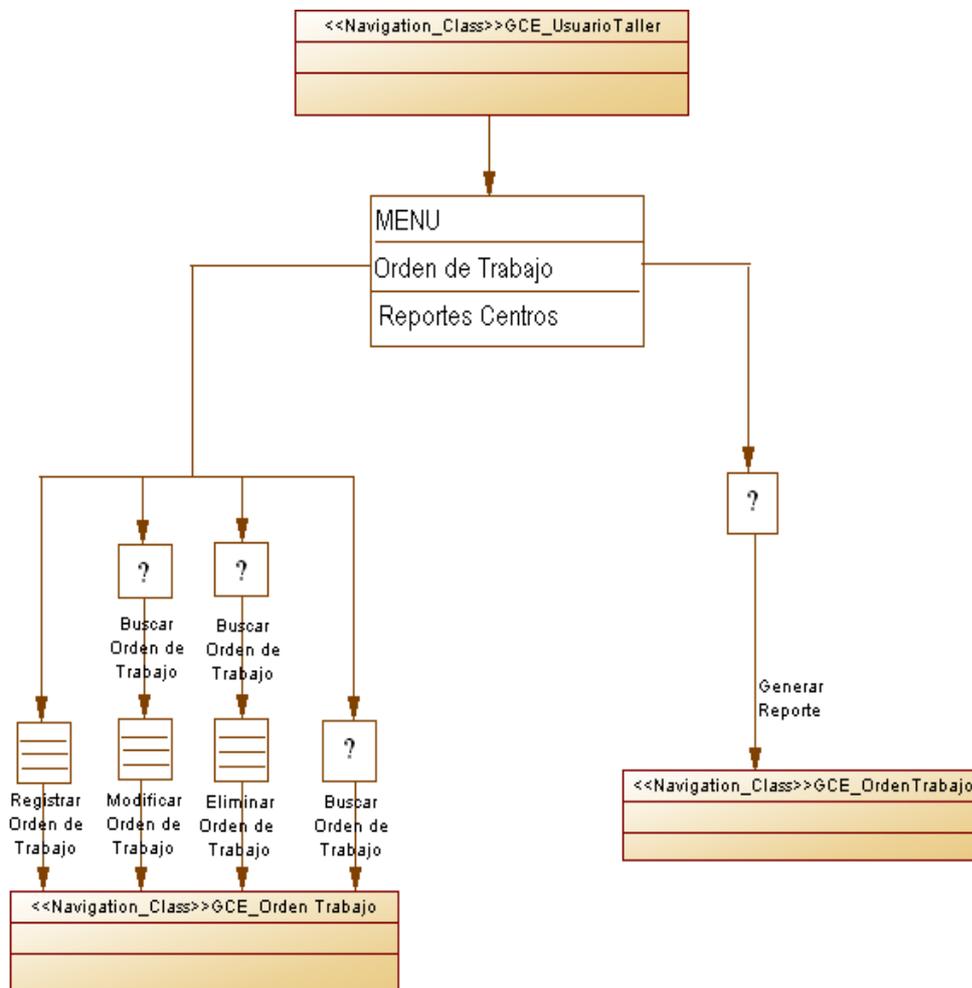
Inscripción en el que se pide información acerca de la vía por la que el cliente se entero de GoldCard dándole cuatro opciones que son: volanteo, carteles, amigos y otros.

La Clase Orden de Trabajo, que como se puede observar en el Anexo 4.5 mantiene una relación de dependencia con la Clase Cliente y la Clase Vehículo, es otra de las clases importantes del Diagrama en ella se guardan todos los datos de los trabajos realizados a un Vehículo (Clase Vehículo) tales como: la fecha de recepción y entrega del vehículo, número de factura (referencial) con la que se facturó el trabajo realizado en el taller y estado de la factura.

## 4.7 Diseño Navegacional

El diseño Navegacional describe la estructura de navegación del Sistema a través de los denominados diagramas de navegación descritos por la metodología UWE. Los Anexos 4.6 y 4.7 detallan la estructura de navegación para los usuarios Súper Administrador y administrador.

En la figura a continuación se detalla la estructura de navegación para uno de los usuarios.



**Fig 4.14:** Diagrama de Estructura de Navegación Usuario Taller (Centro Autorizado).

## 4.8 Diseño de Presentación

La fase de diseño de presentación del proyecto GoldCard está definida por los diagramas de secuencia presentados a continuación (ver diagrama de secuencia para el módulo de Administración en el anexo 4.8)

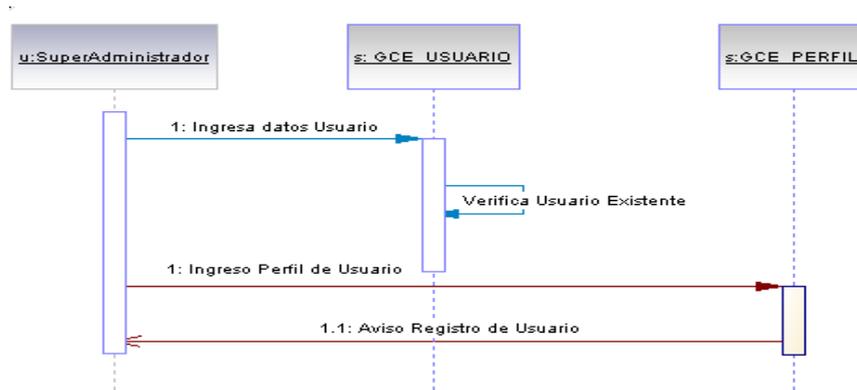


Figura 4.15 Diagrama de Secuencia para el módulo de Seguridades

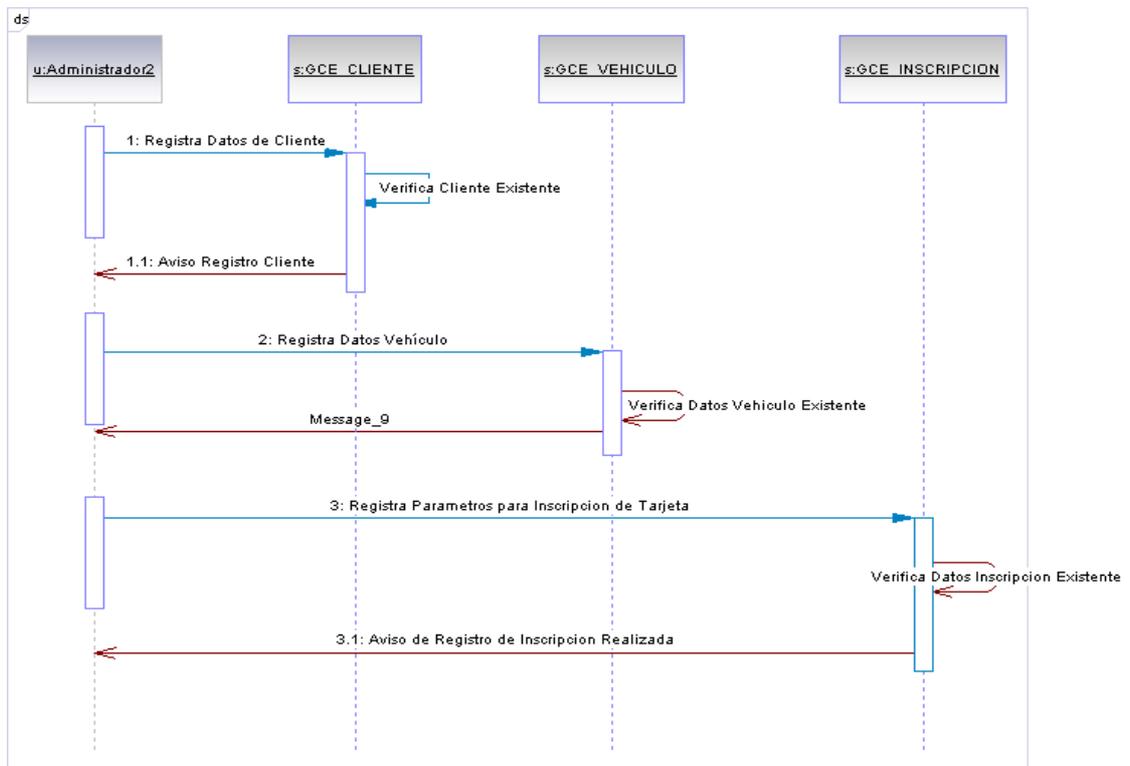
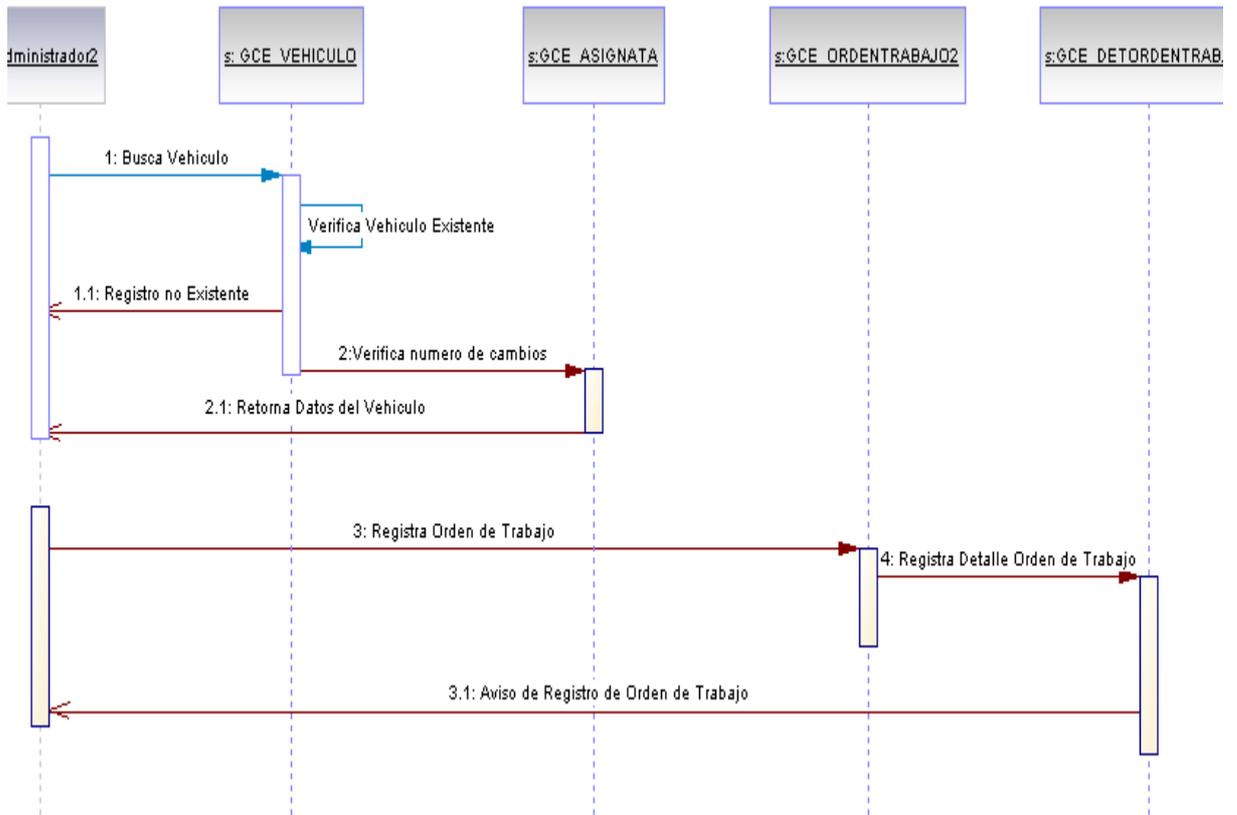
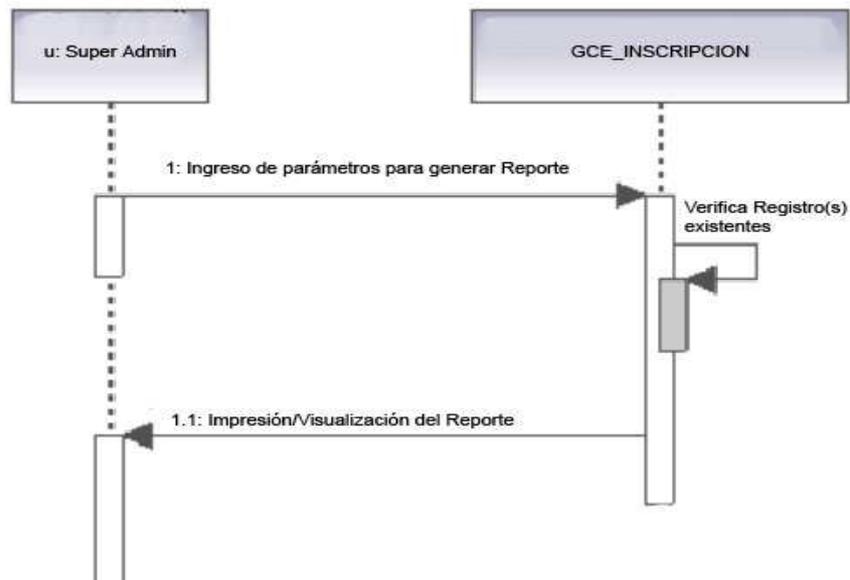


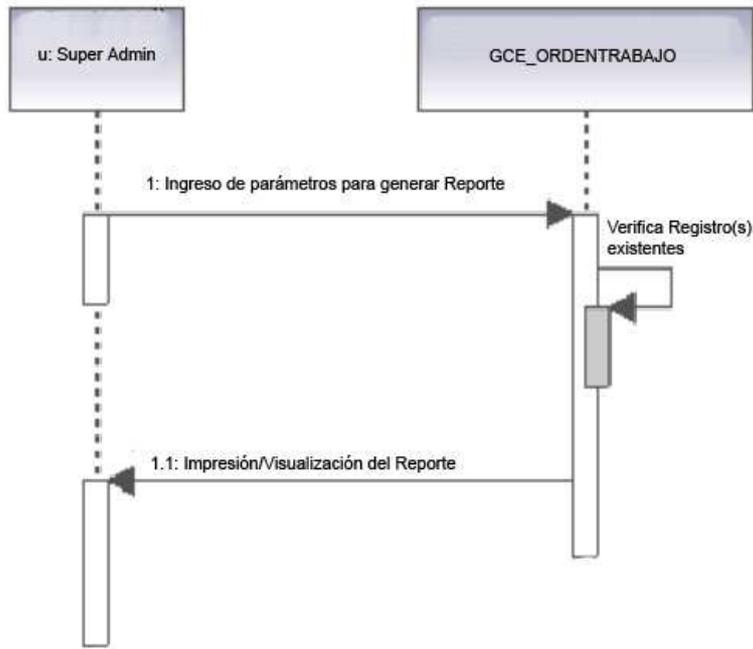
Figura 4.16 Diagrama de Secuencia para el módulo de Inscripción



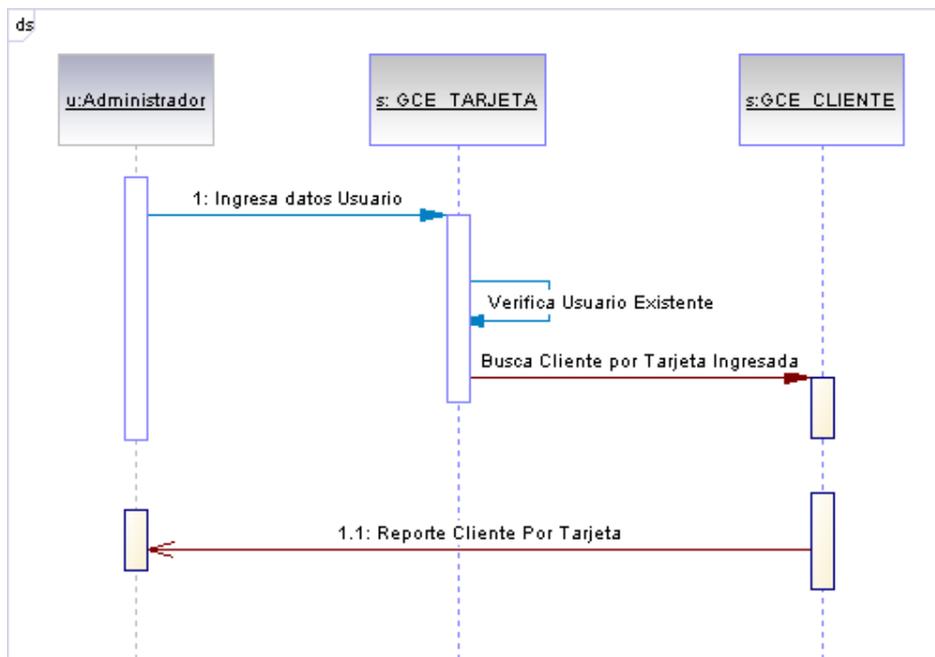
**Figura 4.17** Diagrama de Secuencia para el módulo de Orden de Trabajo



**Figura 4.18** Diagrama de Secuencia Reporte de Inscripción



**Figura 4.19** Diagrama de Secuencia para Reporte de Orden de Trabajo



**Figura 4.20** Diagrama de Secuencia para Reporte de Fechas de Caducidad

## CAPITULO V: Implementación

### 5.1 Arquitectura Web de la Aplicación GoldCard

A continuación muestra los Lenguajes y Herramientas aplicadas al Sistema GoldCard según el nivel de aplicación n-capas

- **SERVICIOS DE PRESENTACION**
  - MICROSOFT VISUAL STUDIO 2005/ASP .NET
- **SERVICIOS DE NEGOCIO**
  - WEB SERVICES
- **SERVICIOS DE DATOS**
  - SQL 2005

En el cuadro a continuación se describe el lenguaje de aplicación utilizado para el Sistema GOLD CAR en cada nivel dentro de la arquitectura:

**Tabla 5.1:** Detalle de la Arquitectura de la Aplicación GoldCard

Nivel		Lenguaje
Cliente	Aplicación Cliente	ASP.NET HTML JAVASCRIPT
Servidor	Servidor Web	WINDOWS SERVER 2003
	Aplicación Servidor	ASP .NET
Servidor de Datos	Base de Datos	SQL Server 2005
	Aplicación Web Service	XML

## 5.2 Estándares del Proyecto GoldCard

### 5.2.1 Estándares para el Modelo Entidad Relación

#### a. Tablas

- Nombre de la tabla: máximo veinte caracteres en total.
- Debe ir en singular
- La designación del nombre deberá guiar al programador
- Se deberá tener en cuenta el siguiente estándar:
- Los tres primeros caracteres son constantes y representan el nombre proyecto: **GCE** (GOLD CAR ENTITY)
- Después de los tres caracteres representativos deberá ir el nombre que describe a la tabla separado con un guión(\_).

#### Ejemplo:

GCE\_CLIENTE (Tabla para almacenar clientes)

#### b. Campos

- Nombre del campo: máximo doce caracteres en total
- Los primeros tres caracteres deben corresponder al nombre de la tabla a la que pertenezcan seguidos de guión bajo (\_)
- La descripción del campo deberá ser lo más clara y concreta posible.

#### Ejemplo:

CLI\_CODIGO (Campo para almacenar el código de la tabla forma de cliente).

#### c. Claves Primarias

- Nombre: máximo diez caracteres.

- Se debe respetar el orden jerárquico de creación (para que los índices se generen correctamente).
- La clave primaria no debe estar compuesta por muchas columnas
- Los primeros tres caracteres deben ser los tres primeros caracteres que describen la tabla, seguido de un sub guión para separar los siguientes caracteres que siempre serán CODIGO.

**Ejemplo:**

CLI\_CODIGO (Constraint de clave primaria para el campo que almacena el código de cliente).

**d. Relaciones**

- Nombre de la relación: máximo 20 caracteres
- Los tres primeros caracteres deberán ser GCE seguidos de un guión bajo
- Los siguientes 10 caracteres se usarán para el nombre o abreviatura de la tabla padre , seguido de guión bajo
- Los siguientes 6 caracteres se usarán para el nombre o abreviatura de la tabla hija

**Ejemplo:**

GCE\_INSCRIP\_CLI (Relaciona las tablas Inscripción con Cliente)

**5.2.2 Estándares para pantallas**

**a.- Tamaño de las pantallas**

- 1024 x 768 píxeles

## **b.- Disposición de la información**

- Vínculos
- Botones

## **c.- Menús**

- Solo texto

## **d.- Botones**

- Sencillo (texto).

## **e.- Campos De Ingreso**

- Campos de ingreso manual
  - Fondo : blanco
- Campos de ingreso mediante tablas (maestro detalle)
  - Fondo : blanco
- Campos de ingreso por defecto
  - Fondo : blanco
- Campos sumatoria/calculados
  - Color letra: Negro
  - Fondo: blanco
- **Subtítulos frames**
  - 12 ptos
  - Color : Negro

#### **f.- Formatos De Presentación De Datos**

- Números: “999,999.00”
- Fechas largas : “dd-MM-yyyy”

#### **g.- Links**

- Tipo : Arial
- Tamaño : 12 ptos
- Color: Azul

#### **h. - Login del Sistema**



**Figura 5.1:** Login del Sistema GoldCard

#### **i.- Descripción Del Contenido De La Pantalla Principal**

- Cabecera de la Página en la parte superior
- A la izquierda del sitio se encuentran enlaces del sitio a la Escuela Politécnica del Ejército y a la empresa promotora y dueña del proyecto GoldCard, Solsupply.

- En la parte central está la información general acerca del Proyecto GoldCard y un link de los beneficios de esta tarjeta.
- A la derecha del sitio se encuentra el logotipo del proyecto y un link de noticias GoldCard.



**Figura 5.2:** Página Principal del sistema GoldCard

### **i.- Cabecera de las Páginas Web**

- Imagen (arte) de GoldCard
- Menú (en la parte inferior)



Figura 5.3: Cabecera de las Páginas del Sistema GoldCard

#### k.- Pantallas De Mantenimiento

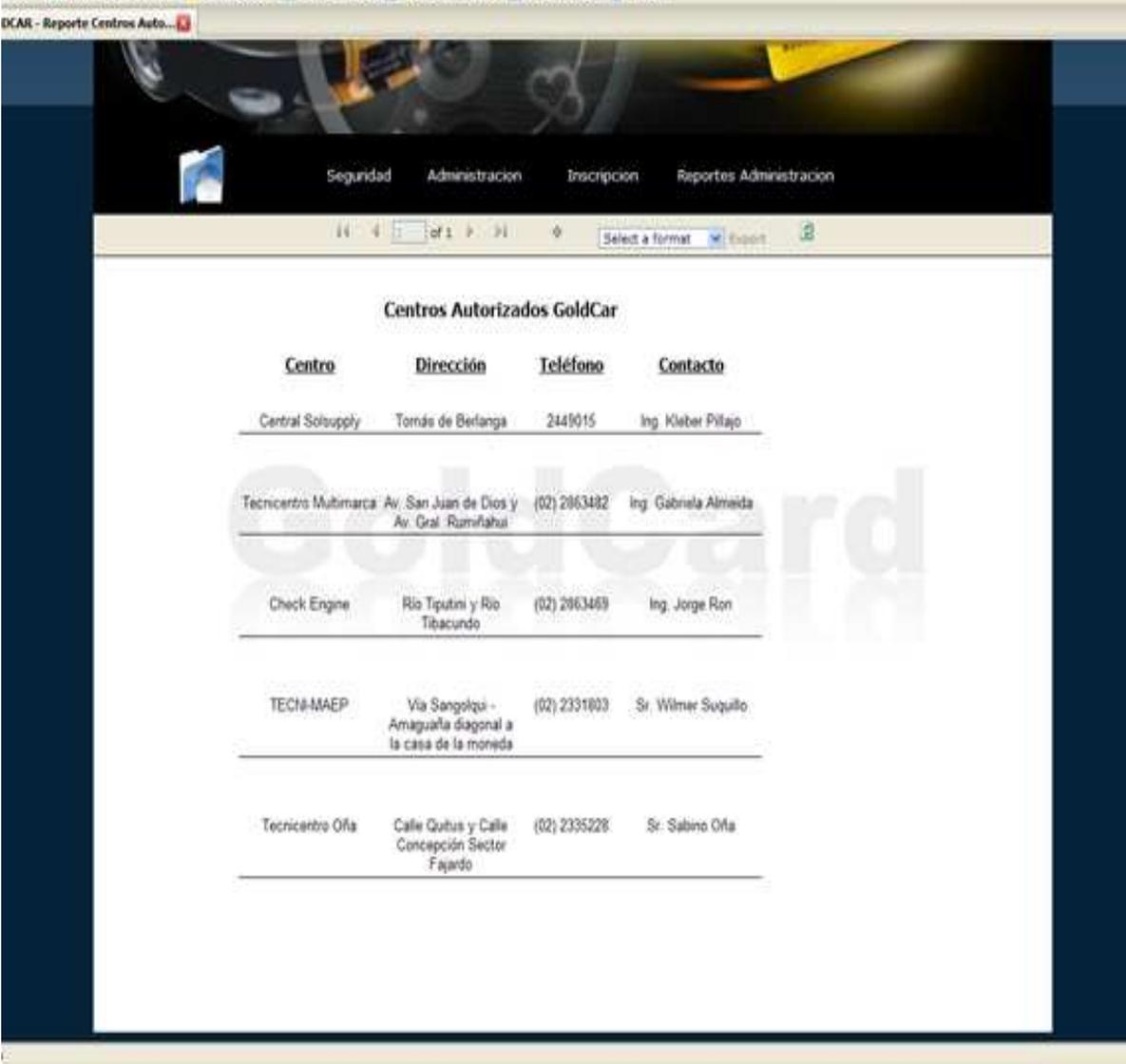
- Cabecera
- Listado de Opciones de navegación
- Maestro Detalle de la Información.
- Vínculos de administración



Figura 5.4: Página ejemplo de administración del Sistema GoldCard

## I.- Reportes

- Cabecera
- Filtros de búsqueda
- Reporte



DCAR - Reporte Centros Auto...

Seguridad Administración Inscripción Reportes Administración

14 of 1

Select a format Export

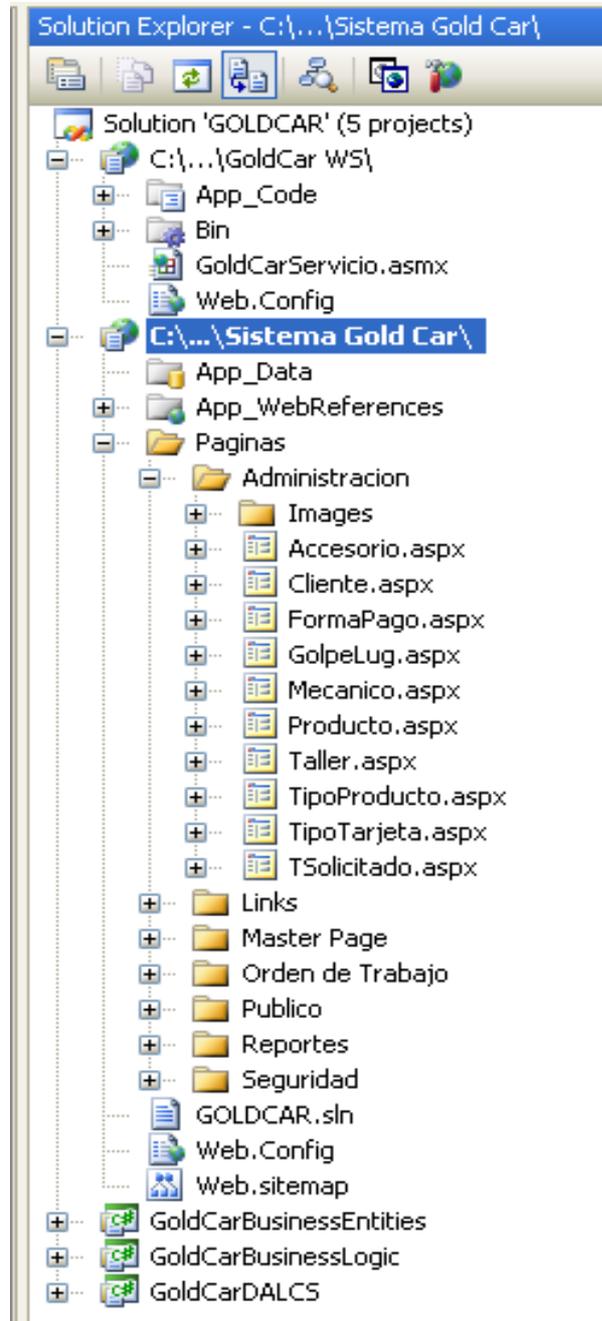
### Centros Autorizados GoldCar

Centro	Dirección	Teléfono	Contacto
Central Solupply	Tomás de Berlanga	2145015	Ing. Kleber Pillajo
Tecnicentro Multimarca	Av. San Juan de Dios y Av. Gral. Ramírez	(02) 2963482	Ing. Gabriela Almeida
Check Engine	Río Tputini y Río Tibacundo	(02) 2963469	Ing. Jorge Ron
TECNI-MAEP	Vía Sangolquí - Amagualta diagonal a la casa de la moneda	(02) 2331803	Sr. Wilmer Suquillo
Tecnicentro Oña	Calle Quito y Calle Concepción Sector Fajardo	(02) 2335228	Sr. Sabino Oña

Figura 5.4: Página Ejemplo de Reporte del Sistema GoldCard

### 5.3 Estándares de Programación

GoldCard está programado enteramente en Microsoft Visual C# .NET. En la figura a continuación se muestra la distribución de la aplicación.



**Figura 5.5:** Distribución de la Aplicación Sistema GoldCard

### 5.3.1 Nomenclatura para DALCS y Store Procedures

La nomenclatura para DALCS y Store Procedures es la siguiente:

- Nombre del método seguido de las tres primeras letras del nombre de la tabla.

Ejm: InsertACE

EliminaACE

- Los objetos declarados tanto en las DALCS se establecen así:  
Empieza con el termino obj. seguido de las tres letras mayúsculas pertenecientes al nombre de la tabla.

Ejm: objACC

### 5.3.2 Nomenclatura para Business Logic y WebService

La nomenclatura para los métodos de Bussiness Logic y los Web Services es la siguiente:

- El nombre del método seguido del nombre de la tabla cuya primera letra deberá ir en mayúscula

Ejm: InsertarAccesorio

EliminarAccesorio

BuscarAccesorio

ActualizarAccesorio

RetornaAccesos

De ser el caso en el que deba agregarse más descripción al nombre del método este deberá igualmente distinguirse con su primera letra en mayúscula.

**Ejm:** RetornaDescripAccesos

- Los objetos declarados tanto en la Capa Lógica o Business Logic y Web Service se establecen así:

Empieza con el termino *ob* seguido de las tres letras mayúsculas pertenecientes al nombre de la tabla.

**Ejm:** *obACC*

En cualquier otro método creado en la Business Logic se puede colocar cualquier nombre.

### **5.3.3 Nomenclatura para definición de variables**

- La definición de variables en la capa de entidades o Bussiness Entibies es la siguiente:
- Inicia con un guión bajo seguido de las tres primeras letras del nombre de la tabla seguido una vez más de un guión bajo y finalmente el nombre completo de la variable.

Ejm: `_ace_codigo`

`_ace_nombre`

- Para el resto de declaraciones la definición será el nombre completo de la variable
- Tipos de Datos

Cada variable en la aplicación debe tener un tipo de dato específico. El tipo de dato especifica, la clase de datos que una variable especifica puede contener, tal como un String, Integer, Date. La mayoría de los tipos básicos de C# provienen

de los tipos del C++, y son los siguientes que se muestran en el cuadro a continuación:

**Tabla 5.2:** Tipos de datos enteros en C Sharp

Tipo de datos de enteros			
Tipo	Ancho en bits	Rango	Significado
Byte	8	De 0 a 255	Entero sin signo de 8 bits
sbyte	8	De -128 a 127	Entero con signo de 8 bits
short	16	De -32.768 a 32.767	Entero corto
ushort	16	De 0 a 65.535	Entero corto sin signo
Int	32	De -2.147.483.648 a 2.147.483.647	Entero medio
UInt	32	De 0 a 4.294.967.295	Entero medio sin signo
long	64	De -9.223.372.036.854.775.808 a 9.223.372.036.854.775.807	Entero largo
ulong	64	De 0 a 18.446.744.073.709.551.615	Entero largo sin signo

**Tabla 5.3** Tipos de datos de punto flotante en C Sharp

Tipo de datos de punto flotante			
Tipo	Ancho en bits	Rango	Significado
<b>float</b>	32	De 1,5E-45 a 3,4E+38	Punto flotante corto
<b>double</b>	64	De 5E-324 a 1,7E+308	Punto flotante largo
<b>decimal</b>	128	De 1E-28 a 7,9E+28	Punto flotante monetario

**Tabla 5.4** Tipos de datos caracteres en C Sharp

Tipo de datos de caracteres			
Tipo	Ancho en bits	Rango	Significado
<b>char</b>	16	De 0 a 65,535 (código Unicode)	Caracter

**Tabla 5.5** Tipos de datos lógicos en C Sharp

Tipo de datos lógicos			
Tipo	Ancho en bits	Rango	Significado
<b>bool</b>	1	true or false, no se usa 1 ó 0 ya que no hay conversión definida	true or false

No existe una conversión definida entre **bool** y los valores enteros (1 no se convierte a verdadero ni 0 se convierte a falso).

**Conversión de tipos:** C# soporta dos tipos de conversiones: Implícita y explícita

- Implícita: son conversiones directas, que se realizan siempre que los tipos puedan hacerlo.

```
int iVal = 34;  
long lVal = iValue;
```

- Explícita: con conversiones en la que debemos forzar la conversión

```
long lVal = 123456;  
int iVal = (int)lValue;
```

**Operadores:** C# tiene cuatro clases generales de operadores: aritméticos, a nivel de bit, relacionales y lógicos

**Tabla 5.6** Operadores en C Sharp

Categoría	Operador	Asociatividad
Primarios	(x) x.y f(x) a[x] x++ x-- new typeof size of checked unchecked	izquierda
Unario o monarios	+ - ! ~ ++x --x (T)x	izquierda
Multiplicativos	* / %	izquierda
Aditivos	+ -	izquierda
Desplazamiento	<< >>	izquierda
Comparación	< > <= >= is	izquierda
Igualdad	== !=	derecha
AND lógico	&	izquierda
XOR lógico	^	izquierda
OR lógico		izquierda
AND Condicional	&&	izquierda
OR condicional		izquierda
Condicional	?:	derecha
Asignación	= *= /= %= += -= <<= >>= &= ^=  =	derecha

### Instrucciones de control

- La instrucción **if-else** es básicamente igual que en C, C++ y Java.
- La diferencia de la instrucción **switch** con la versión de C, C++ y Java es que todo cuerpo perteneciente a un case debe de toparse con un break o

un goto antes de toparse con otro case, a menos que dicho cuerpo esté vacío.

- La instrucción **for** es básicamente igual que en C, C++ y Java.
- La instrucción **while** es básicamente igual que en C, C++ y Java.
- La instrucción **do-while** es básicamente igual que en C, C++ y Java.
- La instrucción **foreach** realiza un ciclo a través de los elementos de una colección (grupo de objetos). El formato de esta instrucción es: **foreach**(tipo variable in colección) instrucción;. En este ciclo se recorre la colección y la variable recibe un respectivo elemento de dicha colección en cada iteración.
- Al igual que en C y C++, la instrucción break permite forzar la salida de un ciclo omitiendo el código restante en el cuerpo del ciclo.
- Al igual que en C y C++, la instrucción continue permite forzar la repetición temprana de un ciclo omitiendo el código restante en el cuerpo del ciclo.
- La instrucción return es básicamente igual que en C, C++. Se utiliza para devolver un valor y salir de un método.
- La instrucción goto se sigue utilizando en C# a pesar de toda la polémica que esto conlleva.

## **Métodos**

- Todo método debe de ser parte de una clase, no existen métodos globales.
- De forma predeterminada, los parámetros se pasan por valor (se copia dicho valor).
- El modificador ref fuerza a pasar los parámetros por referencia en vez de pasarlos por valor.

- El modificador `out` es similar al modificador `ref` con una excepción: sólo se puede utilizar para pasar un valor fuera de un método. El método debe de asignar un valor al parámetro antes de que el método finalice.
- El modificador `params` sirve para definir un número variable de argumentos los cuales se implementan como una matriz. Ejemplo: `public int maxVal(params int[] nums){...}`, esta función se podría llamar así: `maxVal(23,3,a,-12);`.
- Un método debe tener como máximo un único parámetro `params` y éste debe de ser el último.
- Un método puede devolver cualquier tipo de datos, incluyendo tipos de clase.
- Ya que en C# las matrices se implementan como objetos, un método también puede devolver una matriz (algo que se diferencia de C++ en que las matrices no son válidas como tipos de valores devueltos).
- C# implementa sobrecarga de métodos, dos o más métodos pueden tener el mismo nombre siempre y cuando se diferencien por sus parámetros.
- El método `Main` es un método especial al cual se refiere el punto de partida del programa. Tiene la siguiente sintaxis: `public static int Main(string[] args){...}`.

## **Clases y objetos**

Varios puntos a tener en cuenta en C# con respecto a clases y objetos son los siguientes:

- Una variable de objeto de cierta clase no almacena los valores del objeto sino su referencia (al igual que Java).

- El operador de asignación no copia los valores de un objeto, sino su referencia a él (al igual que Java).
- Un constructor tiene el mismo nombre que su clase y es sintácticamente similar a un método.
- Un constructor no devuelve ningún valor.
- Al igual que los métodos, los constructores también pueden ser sobrecargados.
- Si no se especifica un constructor en una clase, se usa uno por defecto que consiste en asignar a todas las variables el valor de 0, null o false según corresponda.
- Para crear un nuevo objeto se utiliza la siguiente sintaxis: `variable = new nombre_clase();`.
- Un destructor se declara como un constructor, aunque va precedido por un signo de tilde ~.
- Se emplea una desasignación de memoria de objetos no referenciados (recolección de basura), y cuando esto ocurre se ejecuta el destructor de dicha clase.
- El destructor de una clase no se llama cuando un objeto sale del ámbito.
- Todos los destructores se llamarán antes de que finalice un programa.
- La palabra clave `this` es un apuntador al mismo objeto en el cual se usa.
- La palabra clave `static` hace que un miembro pertenezca a una clase en vez de pertenecer a objetos de dicha clase. Se puede tener acceso a dicho miembro antes de que se cree cualquier objeto de su clase y sin referencias a un objeto.
- Un método `static` no tiene una referencia `this`.

- Un método `static` puede llamar sólo a otros métodos `static`.
- Un método `static` sólo debe tener acceso directamente a datos `static`.
- Un constructor `static` se usa para inicializar atributos que se aplican a una clase en lugar de aplicarse a una instancia.
- `C#` permite la sobrecarga de operadores con la palabra clave `operator`

## CAPITULO VI: Pruebas

Se realizo un testeo del sistema en los distintos exploradores existentes, se pudo observar que en todos se ejecuto y se visualizó el resultado deseado.

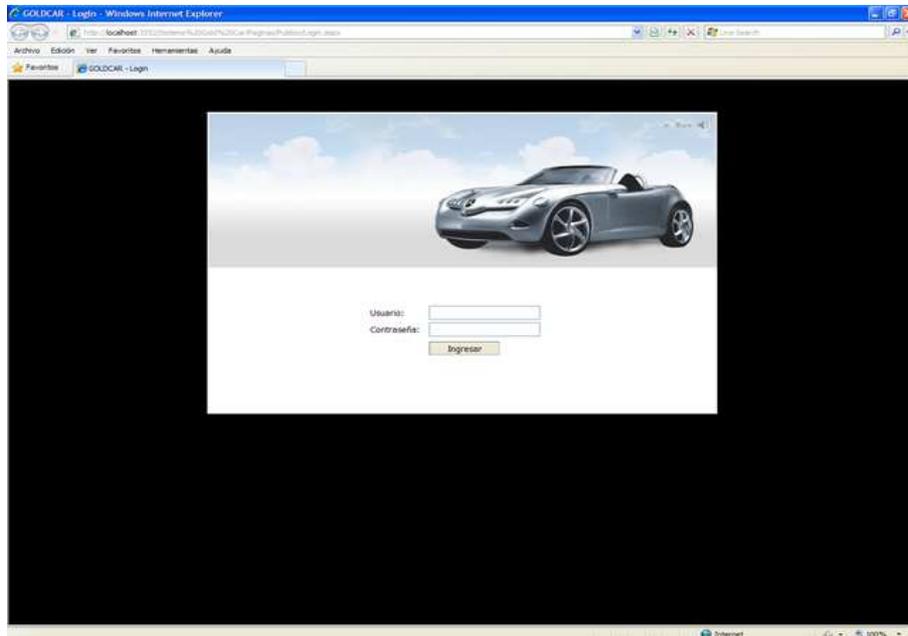


Figura 6.1: Prueba de la Página Login en Internet Explorer 8



Figura 6.2: Prueba de la Página Principal en Internet Explorer 8



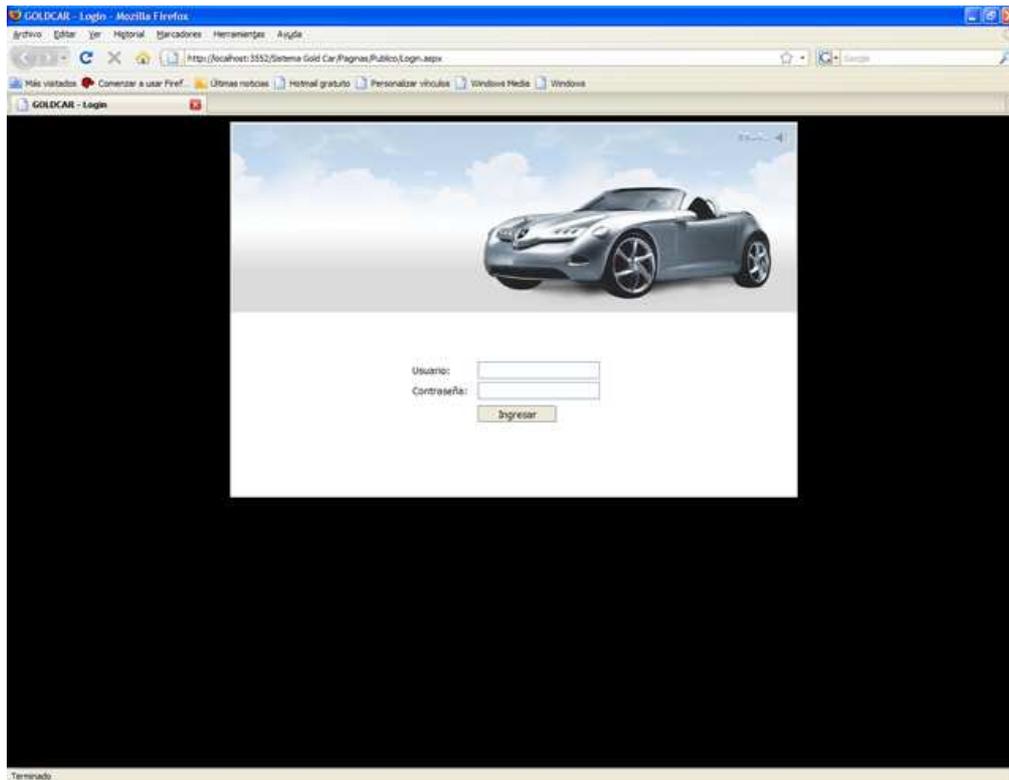


Figura 6.5: Prueba de la Página Login en Mozilla Firefox

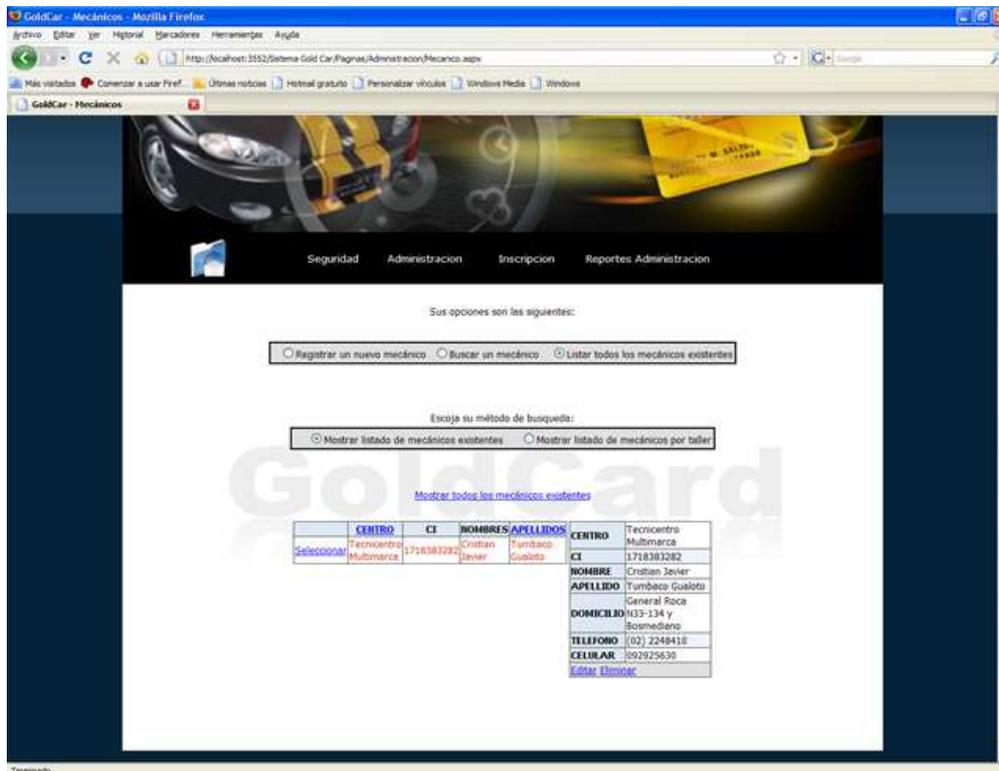


Figura 6.6: Prueba de una Página Administrativa en Mozilla Firefox

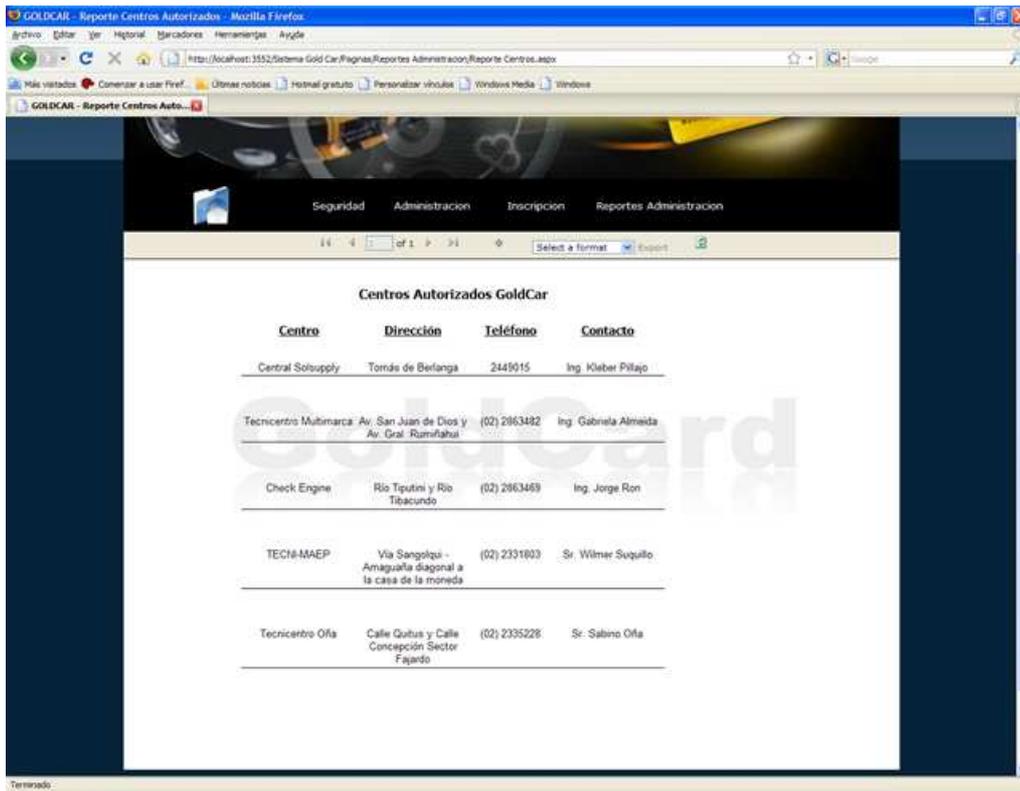


Figura 6.7: Prueba de una Página Reporte en Mozilla Firefox

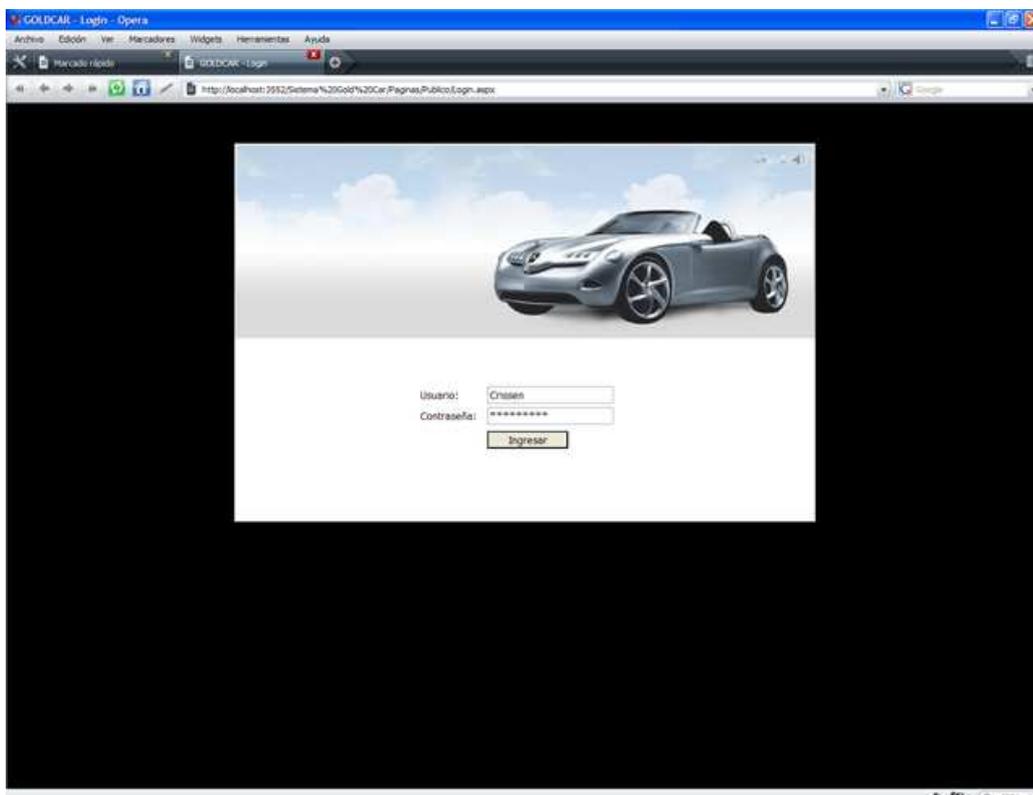


Figura 6.8: Prueba de Página Login en Opera 9.6

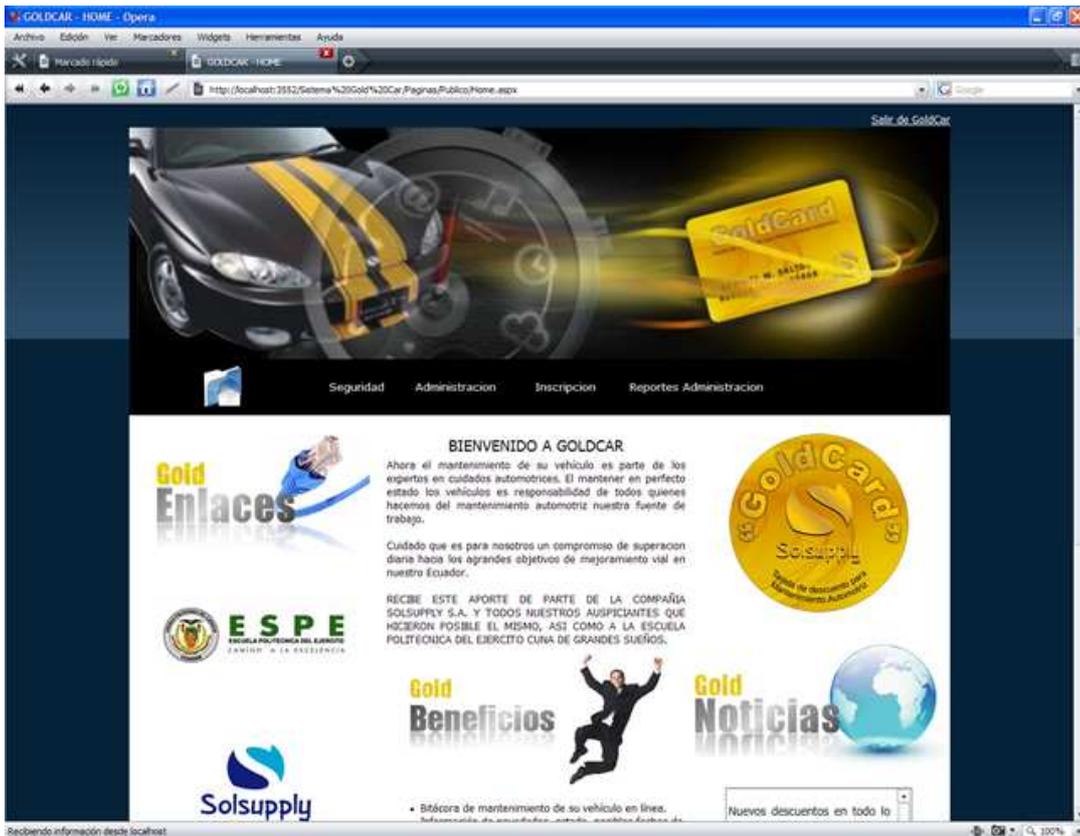
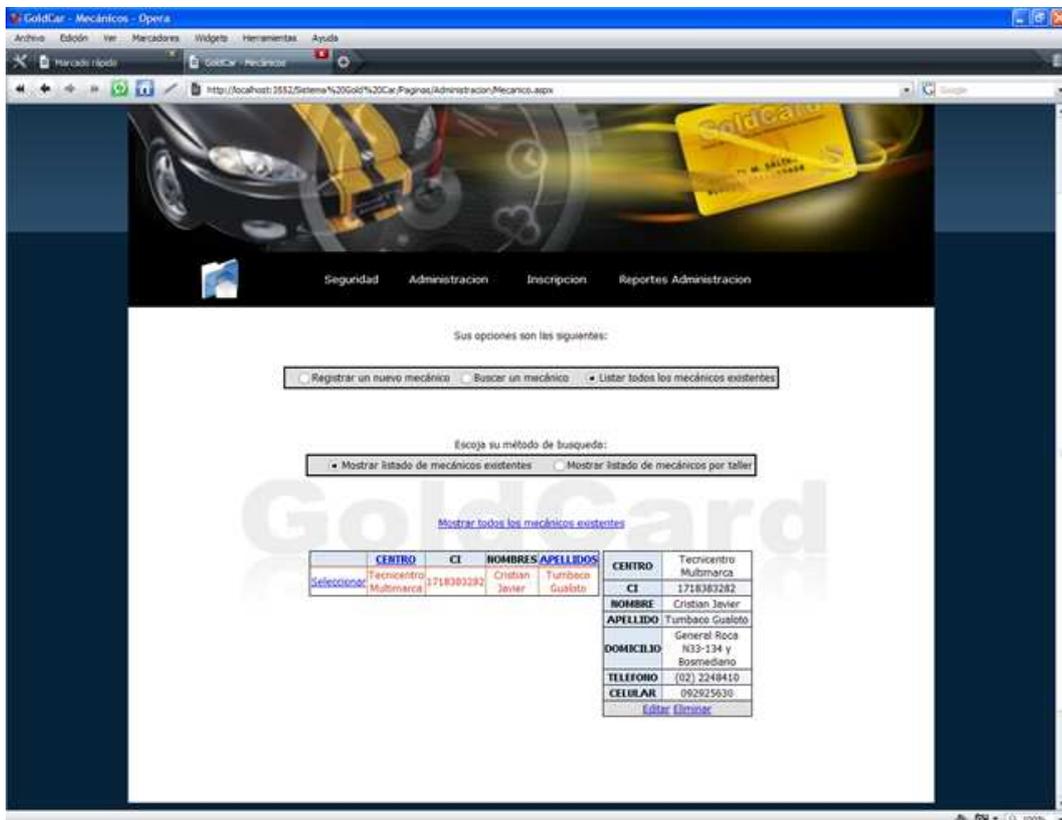


Figura 6.9: Prueba de Página Principal en Opera 9.6



6.10: Prueba de una Página Administrativa en Opera 9.6

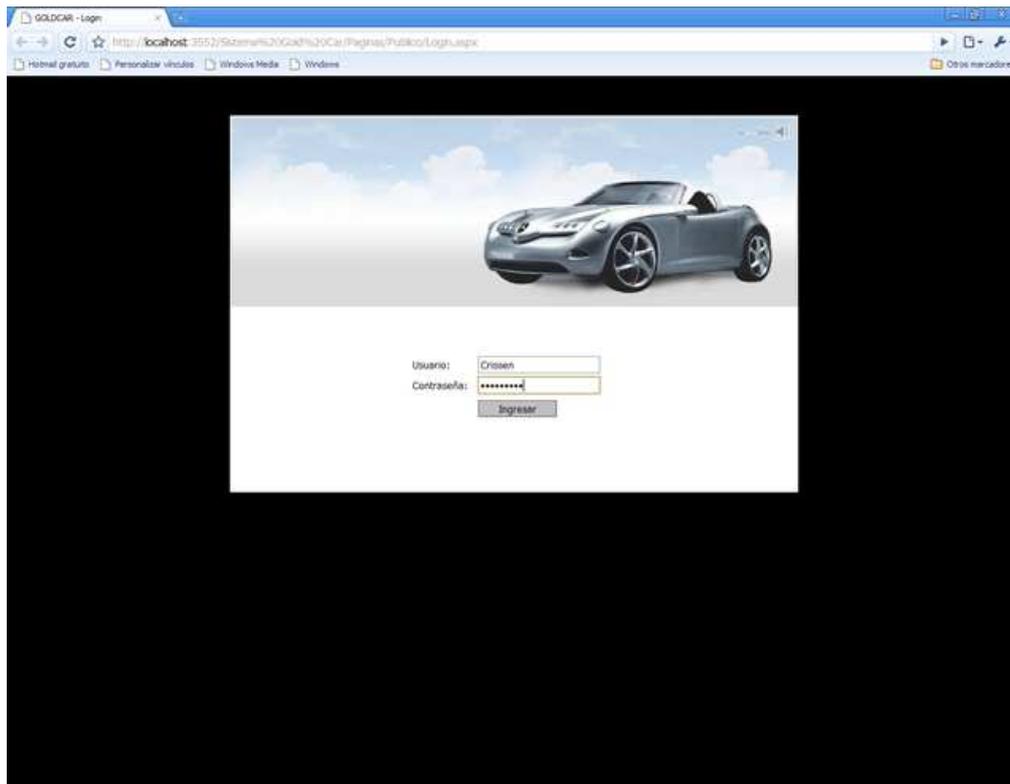


Figura: 6.11: Prueba de Página Login en Google Chrome 2.0

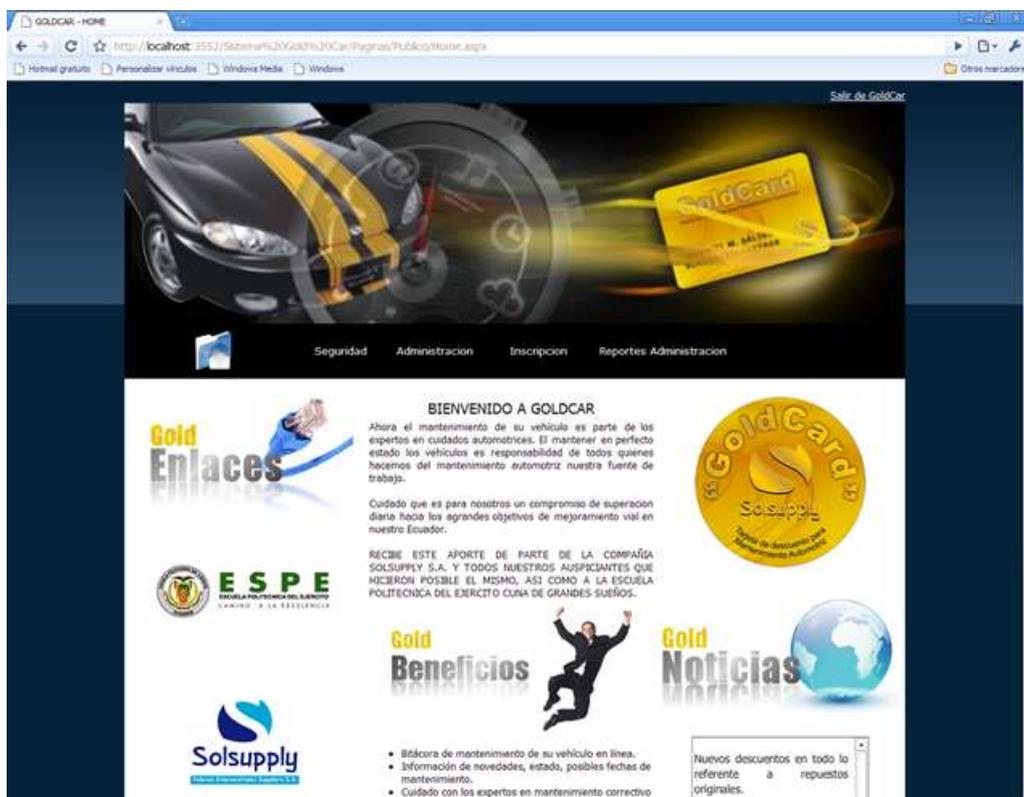


Figura: 6.12: Prueba de Página Principal en Google Chrome 2.0

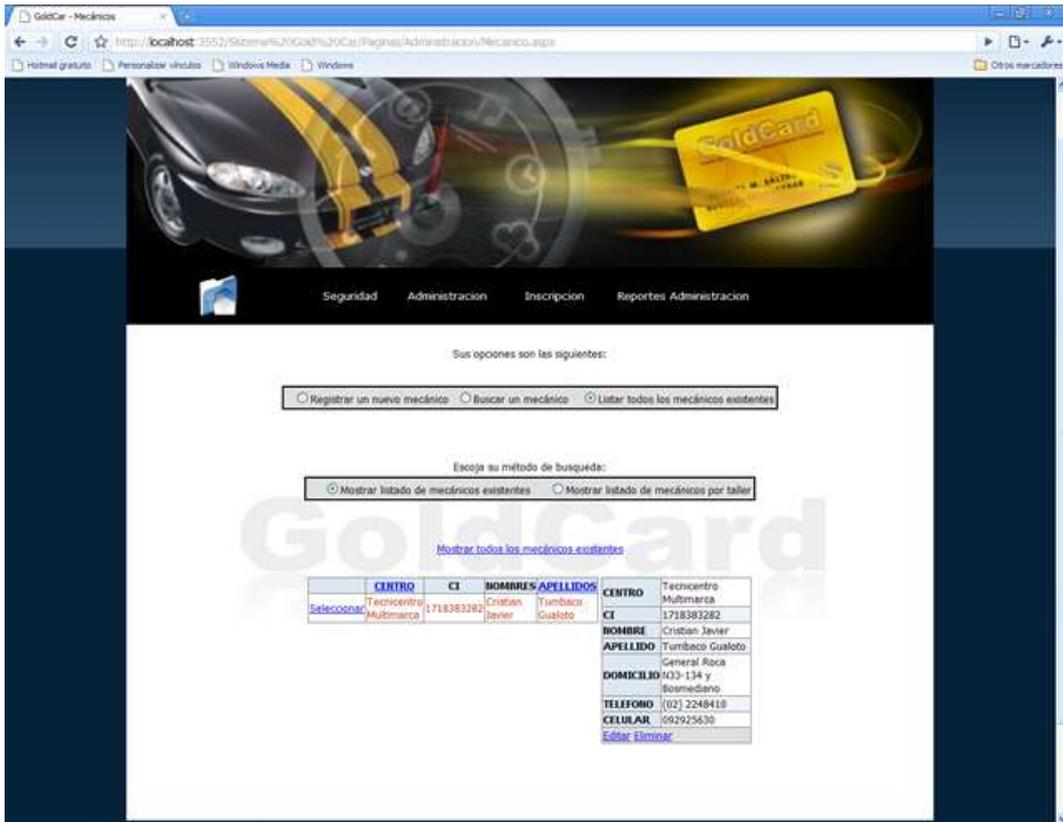


Figura: 6.13: Prueba de Página una Administrativa en Google Chrome 2.0

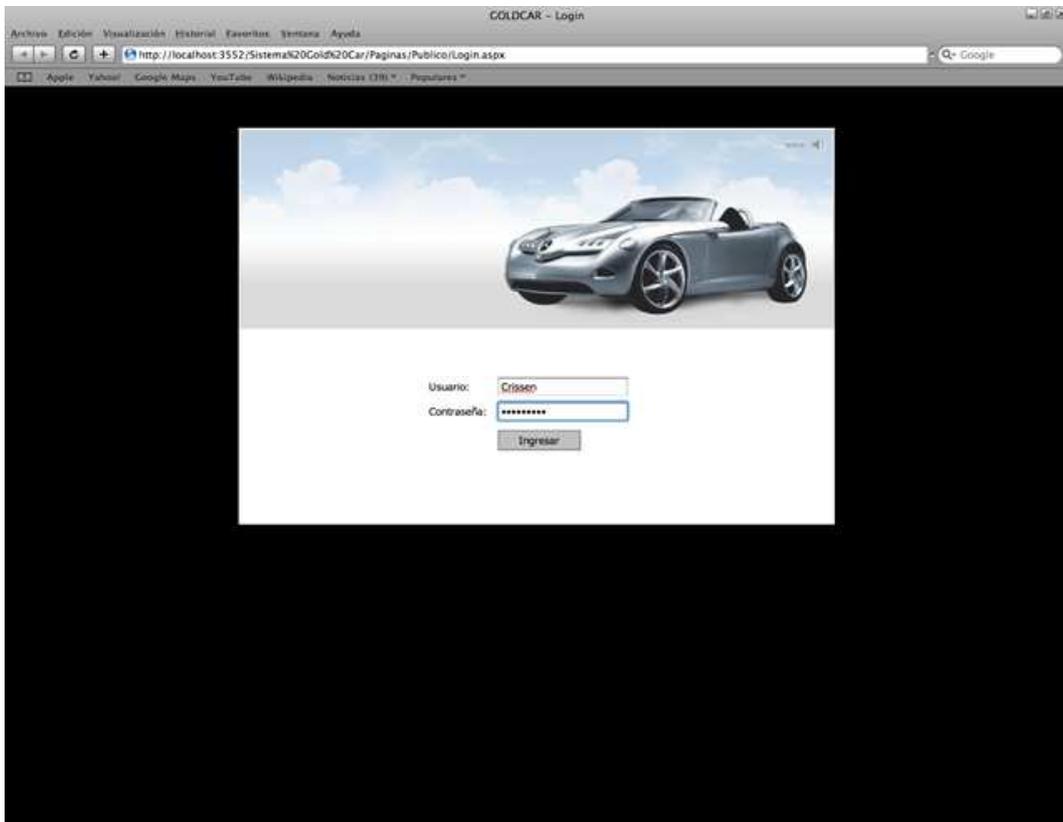


Figura: 6.14: Prueba de Página Login en Safari 2.0



Figura: 6.15: Prueba de Página Principal en Safari 2.0

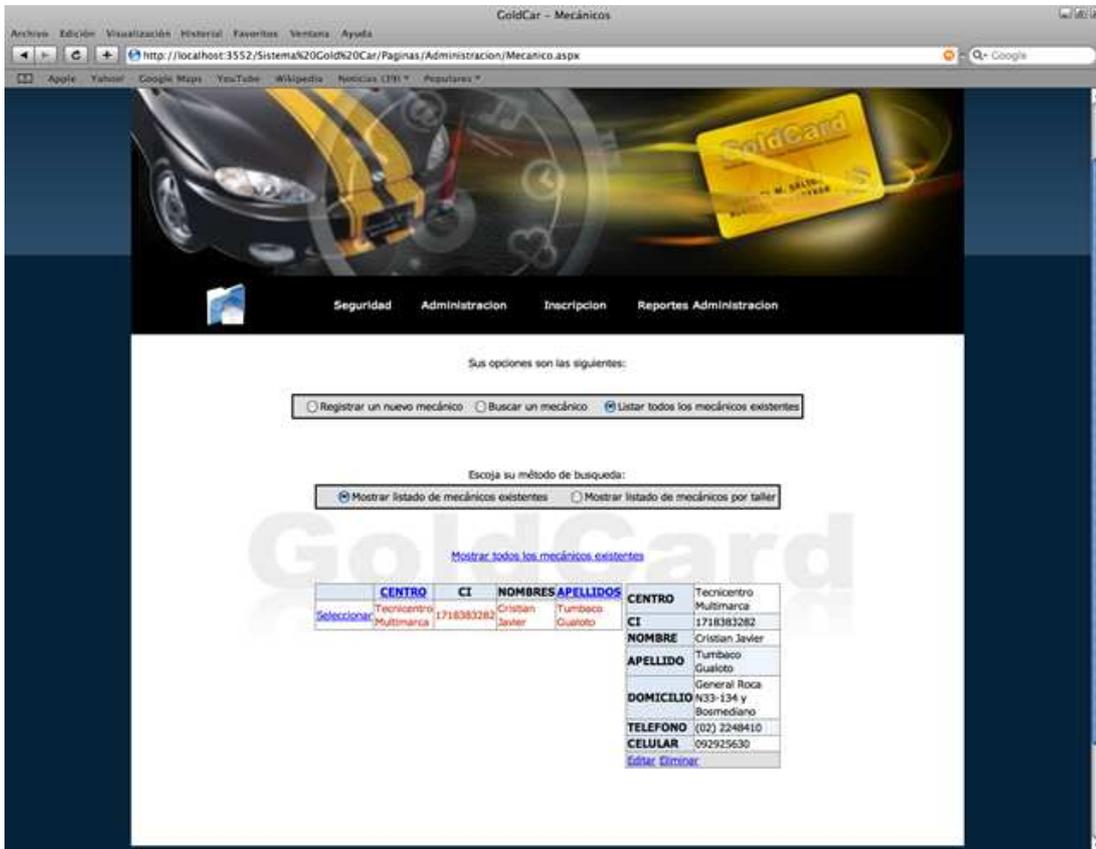


Figura: 6.16: Prueba de una Página Administrativa en Safari 2.0

## **CAPITULO VII: Conclusiones y Recomendaciones**

### **7.1 Conclusiones**

- El análisis de requerimientos funcionales y no funcionales de un sistema es muy importante ya que conforma la columna vertebral del mismo. La metodología que se utilizó para el desarrollo del sistema Web GoldCard tiene como su primera fase el análisis de requerimientos fundamentado en tres niveles en el cual de todas las técnicas expuestas se utilizaron las de Entrevistas y Casos de Uso. Los cuales resultaron ser una muy buena técnica puesto que evitó que en el proceso de desarrollo del sistema se tuvieran que hacer cambios relevantes y que por lo tanto retardaran el tiempo de desarrollo e implementación del mismo. Para la Validación de Requisitos, que es el tercer nivel de la fase de análisis de requerimientos, se utilizó la técnica de Prototipos, que no es una técnica muy compleja pero que sin embargo resulta muy interesante puesto que durante el proceso de desarrollo del sistema se van creando prototipos los cuales son revisados por el usuario final y al momento de la implantación el usuario se encuentra familiarizado en un gran porcentaje con el sistema debido a que el software se fue adaptando a las necesidades del cliente que aparecían según se probaban los prototipos.
- En el caso de GoldCard se aplicó la metodología de UWE debido a que es una metodología orientada a objetos y totalmente para la Web. A diferencia de otras metodologías, a más de la utilización de los ya conocidos casos de uso, UWE muestra únicamente los diagramas necesarios para el desarrollo, incluyendo diagramas Web, como los de

navegación que resultan muy útiles para comprender de mejor manera el software en el que se está trabajando.

- Todos los diagramas obtenidos en el proceso de análisis de requerimientos y de diseño permitieron un desarrollo óptimo y claro del software. Se realizaron cambios durante el proceso que no significaron un problema mayor en el acoplamiento tanto en los diferentes diagramas realizados como en la programación del mismo.
- Una vez finalizado el desarrollo del proyecto y antes de su implantación se pudo observar que más allá de un buen análisis de requerimientos para evitar posibles errores durante el proceso de implementación e implantación es indispensable realizar pruebas sobre el aplicativo ya que al realizar este proceso se descubren las falencias o errores del mismo y estos son corregidos a tiempo, evitando así que aparezcan errores que generen problemas más complicados los cuales significarían pérdidas inestimables para la empresa de aparecer el momento de estar manejando información real.
- La capacitación al personal acerca del software desarrollado fue muy importante durante el proceso de implantación, de esta forma el usuario final se adentró más en la funcionalidad del aplicativo pudiendo trabajar con el mismo de una forma mucho más amigable, fueron necesarias 20 horas de capacitación para que se pudiera dejar claro la total funcionalidad del sistema según el tipo de usuario.

## 7.2 Recomendaciones

- Utilizar buenas prácticas de programación como las que nos brinda Enterprise Library con la finalidad de construir una aplicación más mantenible y sobre todo escalable
- Sería de mucha ayuda para el estudiante, si se lo capacitara un poco más en lo que se refiere a costo de un software, este tema se lo trata superficialmente sin embargo en el entorno laboral nos pudimos dar cuenta que siempre surge esta duda o pregunta por parte de los interesados en el software. Preguntas que hacen alusión al precio de venta del software, dudas acerca de si se puede vender licencias o accesos sobre todo en software de este tipo orientado a la Web. Por lo cual sugerimos se den charlas o cursos acerca de este tema.
- Durante la realización de este proyecto pudimos notar que para el buen desarrollo e implementación de un software es necesario realizar un estudio de las diferentes metodologías que existen, trabajar con la que mejor se adapte al aplicativo y la que más beneficios ofrezca. Esto permitirá reducir notablemente tiempos y dinero en lo que al desarrollo del aplicativo se refiere.

## Bibliografía

[1] *Aplicación Web.*

Disponible en: <http://www.portalprogramas.com/ayuda/c19/programas-online>

[2] Relaciones entre clases, Diagramas de Clases UML.

Disponible en: <http://elvex.ugr.es/decsai/java/pdf/3C-Relaciones.pdf>

[3] *Desarrollo de Aplicaciones Web*

Disponible en: <http://empresas.hispavista.com/soluciones/aplicaciones/>

[4] Koch, N., Kraus, A., Cachero, C., Melia, S., "Modeling Web Business Processes with OO-H and UWE," 3rd International Workshop on Web Oriented Software Technology, Oviedo, Spain. July 2003.

[5] Manual de procedimiento Proyecto GoldCard de la empresa Solsupply.

[6] Manual de procedimiento Proyecto GoldCard de la empresa Solsupply.

[7] *Proyecto de fin de Carrera*

Universidad Pontificia Comillas, Escuela Técnica Superior de Ingeniería

(ICAI), Proyecto de fin de Carrera, Gestión y Control de Talleres

Intercomunicados mediante una Aplicación Web Boutique de Modificación

de Vehículos, Autor:Raúl San Segundo Pariente Director: D. Feliz Alfonso

Sanz, Madrid Junio 2007.

[8] *Diseño de Aplicaciones Tres Capas.*

Disponible en Web: <

<http://www.geocities.com/trescapas/TresCapas.htm>>.

[9] WEBTALLER, *Las Tecnologías de las Páginas Web* [en línea]

Disponible en Web:

<[http://www.webtaller.com/maletin/articulos/las\\_tecnologias\\_de\\_las\\_Paginas\\_web.php](http://www.webtaller.com/maletin/articulos/las_tecnologias_de_las_Paginas_web.php)>.

[10] MTBASE, SYBASE DE COLOMBIA. *Nuevas Características del EASERVER.*

Disponible en Web:

<<http://www.mtbase.com/contenido/documento?id=4,00029>>

[11] WEB URBANO. *WEB SERVICES.* [Marzo 2006]

Disponible en Web: < <http://web.urbano.com.mx/soporte/?secc=1>>

[12] *Visión Global INTRANET.* [Abril 2005]

Disponible en Web:

<<http://www.monografias.com/trabajos12/intrants/intrants.shtml>>

[13] WNTMAG. *INTRANET Resources.* [Enero 2002]

Disponible en Web:

<[http://www.wntmag.com/atrasados/1996/02\\_oct96/intranet.html](http://www.wntmag.com/atrasados/1996/02_oct96/intranet.html)>

[14] *El Desarrollo De Sistemas De Información Empleando El Lenguaje De Modelado Unificado UML.*

Disponible en Web:

<<http://www.monografias.com/trabajos16/lenguaje-modelado-unificado/lenguaje-modelado-unificado.shtml#PROCESO>>

[15] Booch, Jacobson, Rumbaugh, *El Proceso Unificado de Desarrollo de Software*.\_(1999). Editorial Addison Wesley.

[15] TDLP Popkin Software And Systems. *Modelado de Sistemas con UML*.

Disponible en Web:

<<http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/c87.html>>

[16] MSDN, Canchala Armando. *Un Ejemplo Sencillo Sobre Modelado de un*

Disponible en Web:

<[http://www.microsoft.com/spanish/msdn/comunidad/mtj.NET/voices/MTJ\\_2295.asp#M14#M14](http://www.microsoft.com/spanish/msdn/comunidad/mtj.NET/voices/MTJ_2295.asp#M14#M14)>

[17] UWE, *Material para curso de UWE Proyecto* [Mayo 2008]

Disponible en Web: <http://uwe.pst.ifi.lmu.de/teachingCourseMaterial.html>