

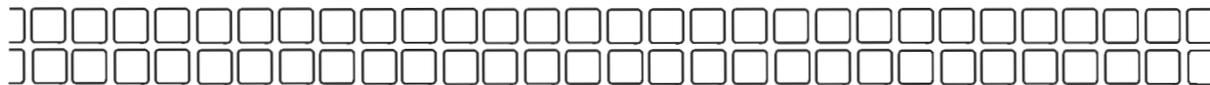
005.12

Día

S

TD-0167-06

TD-0167-06



Instituto Superior Politécnico "José Antonio Echeverría"  
Facultad de Ingeniería Industrial  
Centro de Estudios de Ingeniería y Sistemas

SISTEMA DE ASIGNACIÓN DE TRANSPORTE  
SAT



Trabajo de diploma para optar por  
el título de Ingeniero Informático

Autores: Hitochi Díaz Ramirez  
Adalberto Suárez de Armas  
Tutor: Ing. Yunier Saborit Ramírez  
Colaborador: José Fidalgo Hidalgo



Ciudad de La Habana, Cuba

[ Junio, 2006 ]



Ante la creciente necesidad de crear un sistema que gestione las solicitudes de transporte en el área de Infraestructura Productiva (IP), surge el proyecto: *Sistema de Asignación de Transporte*, siendo este uno de los pasos de la informatización en la Universidad de Ciencias Informáticas (UCI).

En aras de continuar con los objetivos trazados en el proceso de informatización, el proyecto surge con la necesidad de hacer una distribución más racionada de la cantidad de solicitudes de viajes a realizar. Es una aplicación de administración que facilita la planificación actualizada de viajes del área IP.

Mediante el desarrollo del sistema se podrá disponer de una herramienta automatizada, con actualidad en la tecnología utilizada, que brinda considerables beneficios a la entidad a la cual va dirigida: se podrán agilizar las tareas que se desarrollan así como también proporcionar un ahorro considerable de tiempo a las personas involucradas en las mismas, debido a que anteriormente la mayoría debían ser hechas de forma manual, implicando gran pérdida de tiempo y errores a la hora de procesar la información; se podrá también facilitar el procesamiento que debe hacerse sobre la información, así como también lograr mayor seguridad y una utilización óptima de la misma.

Este documento resume las etapas del análisis y diseño de estudio del nuevo sistema que brinda una solución a los problemas que presenta actualmente la infraestructura productiva, y se dejan algunas recomendaciones para el desarrollo futuro.

# Índice

<b>Introducción</b>	<b>1</b>
<b>Capítulo 1. Fundamentación Teórica</b>	<b>5</b>
<b>1.1 Introducción.</b>	<b>6</b>
<b>1.2 Objeto de estudio</b>	<b>6</b>
1.2.1 Objetivos estratégicos de la organización.	6
1.2.2 Flujo actual de los procesos.	7
1.2.3 Análisis crítico de la ejecución de los procesos.	8
<b>1.3 Procesos objeto de automatización</b>	<b>8</b>
<b>1.4 Sistemas automatizados existentes vinculados al campo de acción.</b>	<b>9</b>
<b>1.5 Fundamentación de los objetivos</b>	<b>9</b>
<b>1.6 Tendencias y tecnologías actuales.</b>	<b>10</b>
1.6.1 Internet	10
1.6.2 Soluciones Web	11
1.6.3 Comparación entre Web dinámicas y estáticas.	12
1.6.4 Servidor.	14
1.6.5 Sistemas de Bases de Datos.	18
1.6.6 Arquitectura Cliente/Servidor.	23
1.6.7 Tecnologías de Desarrollo.	25
1.6.8 Tecnología utilizada en el desarrollo de la aplicación.	31
1.6.9 Fundamentación del lenguaje, gestor de BD y servidor Web a utilizar.	38
1.6.10 Fundamentación de la metodología utilizada.	41
<b>1.7 Conclusiones</b>	<b>45</b>
<b>Capítulo 2 . Modelo del Negocio</b>	<b>46</b>
<b>2.1 Introducción</b>	<b>46</b>
<b>2.2 Reglas del negocio a considerar</b>	<b>46</b>
<b>2.3 Descripción de los procesos del negocio propuesto</b>	<b>47</b>
<b>2.4 Diagrama de casos de uso del negocio</b>	<b>47</b>
<b>2.5 Trabajadores del negocio</b>	<b>48</b>
<b>2.6 Casos de uso del negocio</b>	<b>48</b>
2.6.1 Caso de uso Controlar Estado de Autos	48
2.6.2 Caso de uso Controlar Estado de Viajes	50
2.6.3 Caso de uso Controlar Solicitudes	52
2.6.4 Caso de uso Controlar Reportes	53
<b>2.7 Modelo de objetos</b>	<b>55</b>
<b>2.8 Conclusiones</b>	<b>55</b>
<b>Capítulo 3 .Requisitos</b>	<b>56</b>
<b>3.1 Introducción</b>	<b>56</b>
<b>3.2 Actores del sistema a automatizar</b>	<b>56</b>

<b>3.3</b>	<b>Paquetes y sus relaciones</b>	<b>56</b>
<b>3.4</b>	<b>Requerimientos funcionales.</b>	<b>57</b>
<b>3.5</b>	<b>Definición de los requerimientos no funcionales.</b>	<b>59</b>
<b>3.6</b>	<b>Descripción de los casos de uso</b>	<b>66</b>
3.6.1	Descripción de los casos de uso del Paquete Control Usuarios	66
3.6.2	Descripción de los casos de uso del Paquete Control Viajes	63
3.6.3	Descripción de los casos de uso del Paquete Control Solicitud	70
3.6.4	Descripción de los casos de uso del Paquete Control Autos	72
3.6.5	Descripción de los casos de uso del Paquete Control Autos	74
<b>3.7</b>	<b>Conclusiones</b>	<b>76</b>
<b>Capítulo 4 . Descripción de la solución propuesta.</b>		<b>77</b>
<b>4.1</b>	<b>Introducción</b>	<b>77</b>
<b>4.2</b>	<b>Diagrama de clases del diseño Web</b>	<b>77</b>
4.2.1	Paquete Control Usuario	78
4.2.2	Paquete Control Viajes	79
4.2.3	Paquete Control Solicitud	80
4.2.4	Paquete Control Auto	81
4.2.5	Paquete Control Navegación	82
<b>4.3</b>	<b>Principios de diseño</b>	<b>82</b>
4.3.1	Interfaz de usuario	82
4.3.2	Formato de salida de los reportes	83
4.3.3	Ayuda	84
<b>4.4</b>	<b>Tratamiento de errores</b>	<b>84</b>
4.4.1	Modelo lógico de datos	85
4.4.2	Modelo físico de datos	86
<b>4.5</b>	<b>Diagrama de despliegue</b>	<b>86</b>
<b>4.6</b>	<b>Conclusiones</b>	<b>88</b>
<b>Capítulo 5 Estudio de factibilidad</b>		<b>89</b>
<b>5.1</b>	<b>Introducción</b>	<b>89</b>
<b>5.2</b>	<b>Planificación basada en casos de uso.</b>	<b>89</b>
5.2.1	Calcular los Puntos de casos de uso desajustados (UUCP).	90
5.2.2	Calcular los puntos de casos de uso ajustados (UCP).	92
5.2.3	Esfuerzo (E).	96
5.2.4	Esfuerzo Total del Proyecto (ET).	96
<b>5.3</b>	<b>Beneficios Tangibles e Intangibles</b>	<b>97</b>
<b>5.4</b>	<b>Análisis de Costos y Beneficios.</b>	<b>99</b>
<b>5.5</b>	<b>Conclusiones</b>	<b>99</b>
<b>Conclusiones</b>		<b>109</b>
<b>Recomendaciones</b>		<b>101</b>
<b>Referencias Bibliográficas</b>		<b>102</b>
<b>Glosario de Términos</b>		<b>105</b>

La utilización ordenada y masiva de las tecnologías de la información y las comunicaciones en todas las áreas de la universidad, en su esfuerzo por lograr cada vez más eficacia y eficiencia en todos los procesos y por consiguiente mayor generación de riqueza y aumento en la calidad de los servicios. La informatización se basa en los principios y valores fundamentales de nuestro sistema social, con una marcada tendencia a defender las conquistas que hemos logrado.

Cuba está consciente de que la Universidad de Ciencias Informáticas para ser más eficaz, eficiente y competitiva debe aplicar la informatización en todas sus áreas y procesos, y convencida de que para los países subdesarrollados resulta imprescindible el logro de este propósito, ya que su fundamental objetivo es lograr la supervivencia de sus pueblos.

En este sentido, la UCI ha identificado desde muy temprano la conveniencia y necesidad de dominar e introducir en la práctica social las Tecnologías de la Información y las Comunicaciones y lograr una cultura digital como una de las características imprescindibles del hombre nuevo, lo que facilitaría a nuestra sociedad acercarse más hacia el objetivo de un desarrollo sostenible.

El área de infraestructura productiva (IP), por su estado funcional requiere de la necesidad y obligación de realizar numerosos viajes fuera de la ciudad universitaria para dar cumplimiento a las estrategias trazadas para obtener resultados satisfactorios en el desarrollo de sus proyectos. Debido a la gran cantidad de solicitudes de viajes, por varios motivos en relación con la actividad de IP, en muchas ocasiones resulta insuficiente el transporte.

La causa de insuficiencia de cobertura de los viajes esta dada por el trabajo manual que actualmente se desarrolla, al no poder controlar los estados y situaciones de los autos de la universidad por su incesante movimiento en los horarios de entrada y salida en actividad cotidiana, por tanto no es posible conocer el total de autos a disponer en determinada circunstancia. Otro aspecto seria que parte de su jornada laboral esta dispuesto a cubrir funciones de transportación en IP, esto ultimo acompañado del desconocimiento de la planificación del directivo del auto, que permite planificar de antemano la respuesta a una determinada solicitud. La falta de información sobre la cantidad de pasajeros que soporta el auto, seria en ocasiones otro descontento en la planificación a realizar para un determinado itinerario que contempla un número de solicitudes, que podría sobrepasar la cuantía de carga. Durante el proceso de respuesta a las solicitudes planteadas resulta común asignar puntos de destino y recogida a diferentes vehículos, que al ser cercanos pueden ser realizados por un mismo auto.

Tomando en consideración lo anteriormente expuesto nace el *Sistema de Asignación de Transporte*, propuesta encargada de automatizar las tareas que actualmente se desarrollan de forma manual, permite mejorar las condiciones de trabajo de los trabajadores, ofreciendo seguridad y confiabilidad en el manejo de esta actividad.

SAT, automatiza todas las tareas que se hacen de forma manual, permite eliminar los fallos en el procesamiento de la información, ofrece seguridad y confianza en el procesamiento de datos mediante una interfaz Web, sencilla, amigable y de exquisito diseño, ofrece mantener actualizado a los directivos responsables de las diferentes áreas de la UCI del estado de sus autos, además de reducir costos en material de oficina y esfuerzo humano. La ganancia que ofrece la aplicación SAT será de aumentar el control del estado de los vehículos y dar cobertura a las solicitudes de viajes.

El Sistema de Asignación de Transporte ideado para la infraestructura productiva parte en el momento en que los directivos de cada área planifican el horario de uso de los autos según agenda de trabajo y de este modo contar con el plan de disponibilidad para cubrir una petición de viaje.

La aplicación que se propone permitirá mantener la información actualizada a través de una vía más dinámica y fácil, y generará la asignación autos-solicitudes de forma automática.

Con la puesta en funcionamiento de la nueva aplicación se logrará mejor integración y comunicación entre los trabajadores, considerable ahorro en gastos de papel (uno de los pocos argumentos medibles), mayor contribución a la gestión del conocimiento en la organización, mayores facilidades y reducción de costos en concepto de transportación.

Otros beneficios que traerá consigo este proyecto son: el acceso fácil a la información existente de los autos para la transportación, la organización y actualización del flujo de trabajo, el incremento en la eficacia de los trabajadores que se traduce en mayor productividad, la creación de un ambiente de trabajo agradable y la sintonización con las tendencias tecnológicas actuales.

El *objeto de estudio* del trabajo investigativo se basa en controlar la asignación de transporte para los viajes de la infraestructura productiva

El *campo de acción* esta enmarcado en la transportación del área de la infraestructura productiva.

La *hipótesis* esta dada en la posibilidad de contar con una aplicación Web sostenida en la plataforma .NET y que utilice un gestor de base de datos, permitirá controlar el estado de los vehículos de la UCI y dar solución a las peticiones de transporte del área IP.

Sobre la base de darle solución a la problemática planteada el *objetivo general* propuesto es:

- Diseñar e implementar una aplicación Web que permita garantizar la transportación de IP con el control del estado de los autos de la universidad.

El cual está acompañado de los siguientes *objetivos específicos*:

- Generar una interfaz Web adecuada mediante el control de acceso de los usuarios al sistema.
- Definir sección editable que muestre la asignación de autos.
- Definir sección editable que muestre solicitudes de viajes
- Crear servicios informativos que brinden datos del estado de un vehiculo en la Web.
- Gestionar los espacios informativos de los autos de acuerdo a las áreas organizacionales.
- Enviar avisos (e-mail, beeper, llamada telefónica) a directivos de la transportación y uso del vehículo.

Para cumplimentar los objetivos se desarrollarán las siguientes tareas:

- Estudio de los artefactos existentes para el análisis y diseño del sistema que propone RUP (Rational Unified Process) apoyándose en UML (Unified Modelling Language) como estándar notacional.
- Estudio detallado de las herramientas que existen para desarrollar aplicaciones sobre plataforma Web.
- Estudio acerca de la esencia y funcionamiento de una Web
- Estudio sobre la seguridad en la Web.
- Recopilación de los diferentes formatos de presentación de la información deseados por los usuarios.
- Diseñar e implementar la base de datos que se requiera para el manejo de la información a publicar en la Web.

El presente documento esta formado por una Introducción, cinco capítulos, las Conclusiones que incluyen todo lo relacionado con el trabajo investigativo realizado, así como el análisis, el diseño e implementación de la herramienta que se propone, además de las Recomendaciones y una lista de Referencias bibliográficas empleadas en su desarrollo.

La Introducción sirve de presentación del trabajo y orienta a sus lectores sobre los aspectos fundamentales de este.

En el *Capítulo 1*, "Fundamentación teórica", se abordan los principales conceptos asociados al dominio del problema, y el objeto de estudio. Se realiza un análisis comparativo de la solución actual al problema con otras soluciones existentes. Se describen las tendencias y tecnologías actuales sobre las que se apoya la propuesta de solución, fundamentalmente las tecnologías de Internet por estar concebida como una aplicación Web y los sistemas gestores de bases de datos para almacenar y operar la información. También se refieren elementos básicos de la metodología utilizada para el desarrollo del proyecto: RUP.

En el *Capítulo 2*, "Modelo del negocio", se especifican las reglas de negocio, se describen los procesos de negocio propuestos con todos los artefactos que plantea RUP.

En el *Capítulo 3*, "Requisitos", se enuncian los requerimientos funcionales y no funcionales que debe cumplir el sistema.

En el *Capítulo 4*, "Descripción de la solución propuesta", se desarrolla un conjunto de elementos necesarios para la construcción del sistema, entre ellos, diagrama de clases, diseño de la base de datos, estándares de codificación, y otros relacionados con los flujos de trabajo de implementación y prueba según RUP.

En el *Capítulo 5*, "Estudio de factibilidad", se realizan los cálculos para estimar el costo del proyecto, se exponen los beneficios tangibles e intangibles que representan la propuesta para la UCI, y finalmente análisis de costos y beneficios.

### **1.1 Introducción.**

El Sistema de Asignación de Transportes (SAT), es un sistema automatizado que gestiona la asignación de vehículos para el área de Infraestructura Productiva (IP).

En el presente capítulo se brinda una explicación general del objeto de estudio, donde se describe detalladamente como funcionan actualmente los procesos de negocio dentro del SAT. Además, se describe la situación problemática o problemas que presenta actualmente la asignación de transporte por no contar con un sistema para su gestión, con el objetivo de justificar el por que de este proyecto, planteándose los objetivos generales y específicos que se persiguen a través del mismo.

Finalmente se describen las tendencias y tecnologías actuales, justificando la selección de aquellas que se consideran adecuadas para dar solución a la situación problemática existente; se fundamenta la metodología utilizada para el desarrollo del sistema, así como el uso de lenguajes y del gestor de bases de datos.

### **1.2 Objeto de estudio**

#### **1.2.1 Objetivos estratégicos de la organización.**

La creación de la Universidad de las Ciencias Informáticas, con una fuerte base tecnológica y un amplio perfil productivo, donde el estudiante se forma desde la producción, garantiza una amplia capacidad investigativa que facilita las condiciones necesarias para que esta universidad se convierta en el centro de la enseñanza superior de mayor fortaleza científica del país en el área de la informática y tenga un peso importante dentro del sistema científico cubano, lo que

impone a la universidad una participación destacada en las investigaciones dirigidas a buscar soluciones a los problemas que se presentan en el proceso productivo, lo que se corresponde con el concepto de universalización de la enseñanza y el principio de la unidad de la docencia, la investigación y la producción, facilitando la vinculación investigación-producción y una formación más integral del estudiante.

La UCI cuenta con una Infraestructura Productiva (IP) que se dedica a la producción de software en la que los estudiantes juegan un papel esencial formando parte de los proyectos que se llevan a cabo vinculando el estudio con la producción.

Las perspectivas apuntan al papel de esta Universidad como decisivo en el desarrollo de la industria nacional del *software* y del programa de informatización. Se están dando grandes pasos en los sectores de la salud y educación, con la realización de proyectos productivos en los mismos. Para ello hacen uso de una moderna infraestructura tecnológica y un método novedoso de organización.

### **1.2.2 Flujo actual de los procesos.**

El área de Infraestructura Productiva (IP), por su estado funcional requiere de la necesidad y obligación de realizar numerosos viajes fuera de la universidad para dar cumplimiento a las estrategias trazadas para obtener resultados satisfactorios en el desarrollo de sus proyectos.

Actualmente existe una secretaria que es la encargada de recepcionar las solicitudes, definir los viajes y asignarle transporte realizándolo de forma manual, para ello consta con los automóviles de la infraestructura productiva (IP), cuando estos no son suficientes utiliza autos de otras áreas que son controlados y planificados por sus directivos, para solicitar transporte para un viaje es necesario dirigirse a la secretaria de la infraestructura productiva ya sea por correo, teléfono o personalmente.

### **1.2.3 Análisis crítico de la ejecución de los procesos.**

Los procesos descritos anteriormente a pesar de lograr cada uno sus objetivos, no son todo lo eficiente que pudiera, pues es evidente la pérdida de tiempo y descentralización de la información.

En los casos que haya usar automóviles de otras áreas tiene que ser coordinado con anterioridad con sus directivos que son los responsables de planificar sus autos, sin conocer si estos tienen disponibilidad, por lo que la planificación se hace difícil y tediosa.

*Trayendo como consecuencias el desconocimiento de:*

- El total de autos a disponer en un momento determinado.
- La capacidad de los autos.
- La planificación de los automóviles de otras áreas.
- El estado técnico de los automóviles.
- El estado de las solicitudes.
- Estado de los viajes.

### **1.3 Procesos objeto de automatización**

Se desean automatizar todos los procesos citados anteriormente de forma tal que cada directivo planifique sus viajes y tenga acceso total a la parte que le corresponda del sistema que se propone, lo que incluye derechos de publicación y modificación de datos.

El éxito de los procesos en la elaboración del sistema va a depender de un reparto efectivo de las tareas entre dicho sistema y los usuarios del mismo. Se necesitará considerar diferentes opciones en la asignación de transporte antes de especificar una frontera bien definida del sistema, han de establecer, además, una variedad de opciones para identificar una división óptima de la labor a realizar, para asegurar la satisfacción en el trabajo, así como una operatividad eficiente en el proceso.

## **1.4 Sistemas automatizados existentes vinculados al campo de acción.**

La Universidad cuenta con sistemas como el de Acreditación, Akademos y Control de Acceso pero no están vinculados al campo de acción de sistema que se propone.

Luego de realizar una búsqueda de información sobre otros sistemas de asignación de transporte en Internet, se llegó a la conclusión de que en el mundo existen numerosos sistemas similares y algoritmos inteligentes para hacer estos más eficientes los cuales no se adaptan a las condiciones y requerimientos de la Universidad.

*Ejemplo de sistema de asignación de Transporte.*

- Sistema de Asignación Interactiva de Camiones, disponible en [https://e-servicio.ahmsa.com/sip/e-procurement/Portal\\_login.aspx](https://e-servicio.ahmsa.com/sip/e-procurement/Portal_login.aspx).

## **1.5 Fundamentación de los objetivos**

Sobre la base de darle solución a la problemática planteada el *objetivo general* propuesto es:

- Diseñar e implementar una aplicación Web que permita garantizar la transportación de IP con el control del estado de los autos de la universidad.

El cual está acompañado de los siguientes *objetivos específicos*:

- Generar una interfaz Web adecuada mediante el control de acceso de los usuarios al sistema.
- Definir sección editable que muestre la asignación de autos.
- Definir sección editable que muestre solicitudes de viajes
- Crear servicios informativos que brinden datos del estado de un vehiculo en la Web.

- Gestionar los espacios informativos de los autos de acuerdo a las áreas organizacionales.
- Enviar avisos (e-mail, beeper, llamada telefónica) a directivos de la transportación y uso del vehiculo.

## **1.6 Tendencias y tecnologías actuales.**

Los cambios que están aconteciendo en el mercado de la informática y las telecomunicaciones tienen una incidencia directa en la gestión empresarial. La creación de proyectos aplicado las tecnológicas de redes e intercambio de datos tiene grandes ventajas en cuanto a reducción de errores y la creación de nuevas funcionalidades, incuestionables en cualquier sector.

En los términos Tecnologías de Información y Comunicación (TIC), está implícito un elemento fundamental: el de la "Comunicación". Esto porque el intercambio de información entre un emisor y receptor, denominado "mensaje", permite interrelacionarse para trabajar en conjunto.

### **1.6.1 Internet**

Internet ha supuesto una revolución sin precedentes en el mundo de la informática y de las comunicaciones. Los inventos del telégrafo, teléfono, radio y la computadora sentaron las bases para esta integración de capacidades nunca antes vivida. Internet es a la vez una oportunidad de difusión mundial, un mecanismo de propagación de la información y un medio de colaboración e interacción entre los individuos y sus ordenadores independientemente de su localización geográfica.

Internet tiene un impacto profundo en el trabajo, el ocio y el conocimiento. Gracias a la web, millones de personas tienen acceso fácil e inmediato a una cantidad extensa y diversa de información en línea. Un ejemplo de esto es el desarrollo y la distribución de colaboración del software de Free/Libre/Open-Source (SEDA).[1]

## **1.6.2 Soluciones Web**

### **1.6.2.1 ¿Qué es una solución Web?**

Una solución web es aquella que los usuarios usan accediendo a un servidor web a través de Internet o de una intranet. Las aplicaciones web son populares debido a la practicidad del navegador web como cliente ligero. La habilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software es otra razón de su popularidad. Diferente a un sitio Web, una solución puede controlar y administrar todo su negocio, siendo su sistema principal de trabajo.[2]

### **1.6.2.2 ¿Necesito una solución Web?**

Cuando se cuenta con una solución Web, se puede trabajar desde cualquier parte del mundo. Toda la información de su empresa puede estar centralizada en un solo sistema seguro y confiable, operado desde Internet. No importa cual sea el giro de su negocio, seguramente se puede optar por una solución Web. Desde un inventario, base de datos, fotografías y registros de productos hasta su agenda, correo personal y documentos privados.[2]

### **1.6.2.3 ¿Que ventajas brinda una solución Web? [2]**

- **Multiplataforma.** Internet es global, todos los sistemas operativos que acceden a la red, lo hacen mediante el mismo lenguaje estándar. Gracias a esto, no importa si su empresa cuenta con equipo de cómputo de diferentes sistemas operativos (Windows, Mac OS, Unix, Linux, Solaris, etc.), una solución Web será accesible para todos sus usuarios.
- **Omnipresencia.** Es una característica que permite Internet. No importa la distribución física de su negocio, en realidad, no es necesario contar con instalaciones cuando se cuenta con una solución Web. Si tiene colaboradores o trabajadores en diferentes partes del globo, su trabajo puede estar centralizado en el mismo sistema.

- *Interfaz amigable.* Debido a que el ambiente es un navegador, los componentes visuales utilizados son conocidos por la mayoría de usuarios, y por todos los navegadores. Los controles y la manera de operar son comunes al Internet, lo que muchas veces facilita su uso, ganando espacio contra aquellas complejas aplicaciones llenas de botones, menús y comandos mal distribuidos en pantalla.

*Otras Ventajas.* [3]

- Acceso en cualquier momento, desde cualquier lugar, simple a través de portal y sin necesidad de conocimientos especializados.
- Gestión más eficiente y centralizada de los recursos informáticos.

#### **1.6.2.4 ¿Es confiable una solución Web?**

Años de trabajo en seguridad lo demuestran. Los navegadores de Internet más usados contienen toda la seguridad que se puede brindar a los usuarios. Bancos de gran importancia a nivel global realizan transacciones a través de este medio.  
[2]

#### **1.6.3 Comparación entre Web dinámicas y estáticas.**

En Internet se puede encontrar dos tipos de páginas:

- Las que se presentan sin movimiento y sin funcionalidades más allá de los enlaces y animaciones.
- Las páginas que tienen efectos especiales y en las que podemos interactuar.

Las primeras páginas son las que denominamos páginas estáticas, se construyen con el lenguaje HTML (lenguaje de etiquetado de documentos hipertextual), que no permiten realizar un simple cálculo matemático o crear una página a partir de una base de datos. A decir verdad, el HTML, aunque muy útil a pequeña escala, resulta bastante limitado a la hora de concebir grandes sitios o portales. Estas páginas son muy sencillas de crear, aunque ofrecen pocas ventajas tanto a los

desarrolladores como a los visitantes, ya que sólo se pueden presentar textos planos acompañados de imágenes y a lo sumo contenidos multimedia como pueden ser videos o sonidos. [4]

El segundo tipo de páginas se denomina página dinámica. Una página es dinámica cuando se incluye cualquier efecto especial o funcionalidad y para ello es necesario utilizar otros lenguajes de programación, aparte del simple HTML.[4]

Es esta deficiencia del HTML la que ha hecho necesario el empleo de otros lenguajes mucho más versátiles y de un aprendizaje relativamente más complicado, capaces de responder de manera inteligente a las demandas del navegador y que permiten la automatización de determinadas tareas tediosas e irremediables como pueden ser las actualizaciones, el tratamiento de pedidos de una tienda virtual, o la aplicación que es tema de este trabajo. Estos lenguajes capaces de recrear a partir de ciertos "scripts" un sinfín de páginas automatizadas son los protagonistas de este concepto de páginas dinámicas. [4]

*Una página Web estática presenta las siguientes características:[5]*

- Ausencia de movimiento y funcionalidades.
- Absoluta opacidad a los deseos o búsquedas del visitante a la página.
- Para cambiar los contenidos de la página, es imprescindible acceder al servidor donde está alojada la página.
- El usuario no tiene ninguna posibilidad de seleccionar, ordenar o modificar los contenidos o el diseño de la página a su gusto.
- El proceso de actualización es lento, tedioso y esencialmente manual.
- No se pueden utilizar funcionalidades tales como bases de datos, foros, etc.

*Por el contrario, una página Web dinámica tiene las siguientes características:[5]*

- Gran número de posibilidades en su diseño y desarrollo.
- El visitante puede alterar el diseño, contenidos o presentación de la página a su gusto.
- En su realización se utilizan diversos lenguajes y técnicas de programación.
- El proceso de actualización es sumamente sencillo, sin necesidad de entrar en el servidor.
- Permite un gran número de funcionalidades tales como bases de datos, foros, contenido dinámico, etc.
- Pueden realizarse íntegramente con software de libre distribución.
- Cuenta con un gran número de soluciones prediseñadas de libre disposición.

### **1.6.4 Servidor.**

Un servidor es una computadora que maneja peticiones de data, email, servicios de redes y transferencia de archivos de otras computadoras (clientes). También puede referirse a un software específico, como lo es el servidor WWW.

Una computadora puede tener distintos software de servidor, proporcionando muchos servidores a clientes en la red. Por ejemplo, las computadoras que contienen sitios web se llaman servidores ya que “sirven” recursos de web para aplicaciones cliente como los navegadores.[6]

Se denomina servidor en informática a una computadora que realiza algunas tareas en beneficio de otras aplicaciones llamadas clientes, tanto si se trata de un ordenador central (mainframe), un miniordenador, un ordenador personal, un PDA (Ayudante personal digital), o un sistema integrado; sin embargo, hay ordenadores destinados únicamente a proveer los servicios de estos programas. Algunos servicios habituales son los servicios de archivos, que permiten a los usuarios almacenar y acceder a los archivos de un ordenador y los servicios de aplicaciones, que realizan tareas en beneficio directo del usuario final.[7]

*Torre de servidores.*

Una torre de servidores es un grupo de servidores, normalmente mantenidos por una empresa o universidad para ejecutar tareas que van más allá de la capacidad de una sola máquina corriente, como alternativa, generalmente más económica, a un superordenador.

También hace posible la distribución de tareas, de forma que el sistema gana cierta tolerancia a fallos, ya que si uno de los servidores se estropea, el sistema continúa trabajando, notando únicamente una pérdida de rendimiento.[8]

**1.6.4.1 Servidor Web**

Un servidor Web es un programa que implementa el protocolo HTTP (protocolo de transferencia de hipertexto). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas Web o páginas HTML(lenguaje de marcado de hipertexto).[9]

La Figura 1.1 muestra el esquema de un Servidor Web

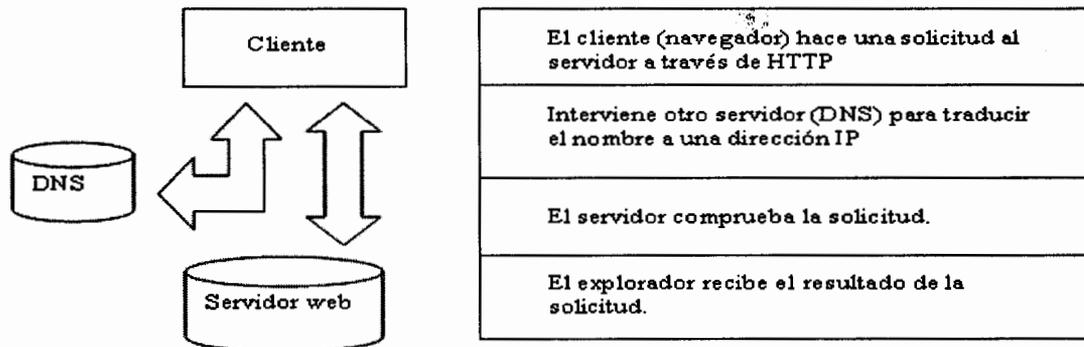


Figura1.1 Servidor Web.[10]

Sobre el servicio Web clásico podemos disponer de aplicaciones Web. Éstas son fragmentos de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Hay que distinguir entre:

*Aplicaciones en el lado del cliente:* el cliente Web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java o Javascript: el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas *scripts*). Normalmente, los navegadores permiten ejecutar aplicaciones escritas en lenguaje *javascript* y *java*, aunque pueden añadirse mas lenguajes mediante el uso de *plugins*.

*Aplicaciones en el lado del servidor:* el servidor Web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.

### 1.6.4.2 Servidores Web

#### Apache

Apache es un servidor Web, que permite el alojamiento de paginas Web en una máquina específica.

Algunas de sus principales características son:

- Se ejecuta en una multitud de Sistemas operativos Windows, la mayoría de versiones de Unix y sus derivados, haciéndolo prácticamente universal.
- Es una tecnología gratuita de código fuente abierto, entiéndase por esto que da una transparencia ha este software de manera tal que si se quiere ver lo que se está instalando como servidor se puede saber sin ningún secreto.
- Es un servidor altamente configurable de diseño modular.
- Trabaja con Perl, PHP, Java y otros lenguajes de script.
- Permite personalizar la respuesta ante posibles errores que se puedan dar en el servidor. [11]

### Internet Information Server

Internet Information Services o Server (IIS), es el servicio Web con el que cuenta la familia de Sistemas Operativos Windows 2000 y XP para la publicación de información en Intranets e Internet.

Este servicio convierte a un computador en un servidor de internet o Intranet es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas web tanto local como remotamente (servidor web). [12]

El servidor Web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de ASP (Active Server Pages) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP (Hypertext Preprocess) o Perl.[12]

#### Las principales características de este servicio Web son:

*Sistema distribuido de creación y controles de versiones Web (WebDay):* Permite a los usuarios remotos crear, mover o borrar archivos, directorios y propiedades de directorio en el servidor a través de una conexión HTTP.

*Administración personal del Web:* Incluye una herramienta de Administrador personal del Web, que ayuda a administrar y publicar un sitio de publicación personal.

*Servicios incluidos:* Brinda los servicios de FTP (Protocolo de Transferencia de Ficheros) y WWW (World Wide Web); al igual que ofrece servicios de envío de correo electrónico y noticias, a través del servicio SMTP de Microsoft, que usa el Protocolo Simple de transferencia de correo (SMPT), un estándar de Internet.

*Protección de aplicaciones:* Ofrece mayor protección y confiabilidad de las aplicaciones Web. De manera predeterminada, IIS ejecutará todas las aplicaciones en un proceso común o agrupado que está separado de los procesos del núcleo de IIS. Además, también puede aislar aplicaciones cuyas misiones sean críticas y que se deban ejecutarse fuera tanto de los procesos de núcleo de IIS como de los agrupados. [12]

### Apache vs. IIS

Los dos servidores de la red más difundidos son, sin duda alguna, IIS de Microsoft y el adversario Apache. La diferencia principal entre los dos apliques es la plataforma para las que fueron pensadas: IIS para servidores que se basan en los sistemas Windows, Apache para la gran familia de los Unix. Sin embargo, un punto a favor de Apache puede ser el esfuerzo (en parte también logrado) para llevar el servidor de la red a plataformas no nativas, como, por ejemplo, los sistemas Windows. Obviamente, los mejores resultados se obtienen en plataformas nativas, aunque sin lugar a duda tiene excelentes prestaciones en otras plataformas. [13]

Windows dispone de una interfaz gráfica que facilita el manejo de los procedimientos; cada comando puede ser visualizado en pantalla mediante una imagen que lo representa. Si bien, es una característica que comparte con otros sistemas, Windows es el Sistema Operativo con mayor difusión en el mercado actual; y a su vez, dispone de la compatibilidad con los productos Microsoft. [13]

### **1.6.5 Sistemas de Bases de Datos.**

Una base de datos es un conjunto de datos almacenados entre los que existen relaciones lógicas y ha sido diseñada para satisfacer los requerimientos de información de una empresa u organización. En una base de datos, además de los datos, también se almacena su descripción; es un gran almacén de datos que se define una sola vez y que se utiliza al mismo tiempo por muchos departamentos y usuarios. En lugar de trabajar con ficheros desconectados e información redundante, todos los datos se integran con una mínima cantidad de duplicidad; la base de datos no sólo contiene los datos de la organización, también almacena una descripción de dichos datos. Esta descripción es lo que se denomina metadatos, se almacena en el diccionario de datos o catálogo y es lo que permite que exista independencia de datos lógica-física. [14]

El modelo seguido con los sistemas de bases de datos, en donde se separa la definición de los datos de los programas de aplicación, es muy similar al modelo que se sigue en la actualidad para el desarrollo de programas, en donde se da una definición interna de un objeto y una definición externa separada. Los usuarios del objeto sólo ven la definición externa y no se deben preocupar de cómo se define internamente el objeto y cómo funciona. Una ventaja de este modelo, conocido como abstracción de datos, es que se puede cambiar la definición interna de un objeto sin afectar a sus usuarios ya que la definición externa no se ve alterada. Del mismo modo, los sistemas de bases de datos separan la definición de la estructura de los datos, de los programas de aplicación y almacenan esta definición en la base de datos. Si se añaden nuevas estructuras de datos o se modifican las ya existentes, los programas de aplicación no se ven afectados ya que no dependen directamente de aquello que se ha modificado. [14]

### *Ventajas de los Sistemas de Bases de Datos.*

Los sistemas de bases de datos presentan numerosas ventajas que se pueden dividir en dos grupos: las que se deben a la integración de datos y las que se deben a la interfaz común que proporciona el SGBD (Sistemas Gestores de Bases de Datos).[14]

#### **1.6.5.1 Motores de Bases de Datos.**

##### PostgreSQL

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) que ha sido desarrollado de varias formas desde 1977. Comenzó como un proyecto denominado Ingres en la Universidad Berkeley de California.

PostgreSQL está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo, es un servidor de base de datos relacional libre. Es una alternativa a otros sistemas de bases de datos de código abierto (como MySQL, Firebird y MaxDB), así como sistemas propietarios como Oracle o DB2.[15]

*Algunas de sus principales características son:*

- Claves ajenas también denominadas Llaves ajenas o Llaves Foráneas (foreign keys).
- Disparadores (triggers).
- Vistas.
- Integridad transaccional.
- Acceso concurrente multiversión (no se bloquean las tablas, ni siquiera las filas, cuando un proceso escribe).
- Capacidad de albergar programas en el servidor en varios lenguajes.
- Herencia de tablas.
- Tipos de datos y operaciones geométricas.[15]

### MySQL.

Es un servidor de bases de datos relacionales muy rápido, multiusuario y multihilo, usado sobre todo en Internet en conjunción con PHP. Es software libre (licencia GPL) y es mantenido por la compañía sueca MySQL AB.[16]

Una característica importante del servidor de bases de datos MySQL es que consume muy pocos recursos, tanto de CPU como de memoria. Fue desarrollado originalmente para manejar grandes bases de datos mucho más rápido que las soluciones existentes, con este fin se sacrificaron algunas características esenciales en sistemas más "serios", carencia que no lo hace adecuado para aplicaciones críticas. Son apropiadas su conectividad y velocidad para acceder a bases de datos en Internet. [17]

Resulta útil para la administración (backup, recuperación de errores), no tiene límites en el tamaño de los registros, es portable de modo que se puede emplear tanto en servidores Linux como Windows y realiza control de acceso de usuarios y permisos. Tiene como inconvenientes, que no soporta transacciones por lo que ante un problema que interrumpiese una serie de comandos podría dejar la base de datos en un estado inconsistente, no considera llaves extranjeras e ignora la integridad referencial, dejándola en manos del programador de la aplicación. [17]

*Entre las características disponibles en las últimas versiones se puede destacar:*

- Amplio subconjunto del lenguaje SQL.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.[18]

### ORACLE

Es un sistema gestor de bases de datos que implementa el modelo relacional y una versión del modelo relacional orientado a objetos. Es un sistema multiplataforma que admite diferentes modelos de ejecución:

*Cliente/servidor.* El servidor realiza gran parte de las tareas de acceso a los datos en respuesta a la solicitud del cliente, que puede ser una plataforma diferente mediante comunicación soportada con TCP/IP, Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP).

*Centralizado o basada en mainframe.* El servidor realiza todas las tareas de acceso a los datos e incluso devuelve los datos con formato para la presentación en informes o formularios. El cliente no realiza prácticamente ninguna función. Se usa en grandes ordenadores centrales (mainframes) con terminales soportados directamente por su propio sistema de comunicaciones

*Procesamiento distribuido.* La base de datos está repartida en diferentes servidores y los usuarios no son conscientes de la ubicación física de los datos. El servidor distribuido maneja los datos de manera transparente al usuario.

*Procesamiento paralelo.* La base de datos está en una única plataforma con varios procesadores y las consultas se pueden ejecutar en paralelo.[19]

Está previsto para grandes demandas, por lo que dispone de todas las características que se requieren de un SGBD aplicable a grandes empresas, tales como: mecanismos de seguridad mediante un lenguaje de control de datos que permite definir derechos de consulta de seguridad; copias de seguridad y recuperación que permite realizar con el servidor parado o funcionando por lo que es posible un servicio ininterrumpido; conectividad abierta que admite acceder a datos de otros SGBD; herramientas de desarrollo para la generación de informes, formularios; soporte a grandes cantidades de datos que alcanzan el orden de los gigabytes y acceso rápido a los mismos. Se considera muy eficaz para aplicaciones sumamente exigentes, por su robustez y capacidad de soportar más del millón de transacciones.[19]

Su interfaz no resulta muy amigable comparada con la de Microsoft SQL Server y esto atenta contra la facilidad de configuración. Tiene la ventaja de ser un producto multiplataforma pues más del 80% de los códigos internos de Oracle son iguales a los establecidos en todas las plataformas de sistemas operativos. [19]

### Microsoft SQL Server 2000.

SQL Server es un sistema de gestión de bases de datos relacionales utilizado por una amplia gama de clientes corporativos y proveedores independientes de software que construyen aplicaciones de negocios. Se basa en la arquitectura cliente-servidor y utiliza el lenguaje de bases de datos normalizado SQL. No es multiplataforma, está diseñado para Windows.[20]

Cuenta además con las siguientes ventajas:

**Escalabilidad:** Se adapta a las necesidades de la empresa, soportando desde unos pocos usuarios a varios miles. Empresas centralizadas u oficinas distribuidas, replicando cientos de sitios.

- **Gestión:** Con un completo interfaz gráfico que reduce la complejidad innecesaria de las tareas de administración y gestión de la base de datos.

- Orientada al desarrollo: Visual Basic, Visual C++, Visual J++, Visual Interdev, Microfocus Cobol y muchas otras herramientas son compatibles con Microsoft SQL Server.
- Seguridad.

La estrategia de Microsoft es la de hacer que SQL Server sea la base de datos más fácil de utilizar para construir, administrar e implementar aplicaciones de negocios. Esto significa tener que poner a disposición un modelo de programación rápido y sencillo para desarrolladores, eliminando la administración de base de datos para operaciones estándar, y suministrando herramientas sofisticadas para operaciones más complejas.[20]

SQL Server puede manejar perfectamente bases de datos de terabytes con millones de registros y funciona sin problemas con miles de conexiones simultáneas a los datos. Es un servidor de bases de datos pensado para gestionar tantos clientes simultáneos como admita la potencia del hardware del equipo en el que esté instalado.[20]

### 1.6.6 Arquitectura Cliente/Servidor.

La arquitectura Cliente / Servidor es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en elementos independientes que cooperan entre sí para intercambiar información, servicios o recursos.[21] (Ver Figura 1.2)

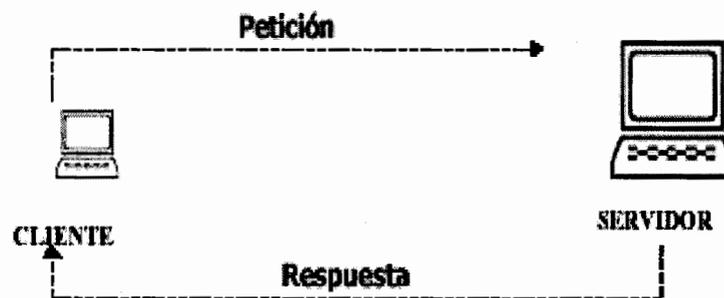


Figura 1.2. Modelo Cliente Servidor.

Entre las principales características de la arquitectura Cliente / Servidor, se pueden destacar las siguientes:

- El servidor presenta a todos sus clientes una interfaz única, amigable y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

La arquitectura Cliente / Servidor es una infraestructura versátil modular y basada en mensajes que pretende mejorar la portabilidad, la interoperabilidad y la escalabilidad del cómputo; además es una apertura del ramo que invita a participar a una variedad de plataformas, hardware y software del sistema. [21]

Conceptualmente, los componentes de la arquitectura Cliente / Servidor son el cliente, el servidor y la infraestructura de comunicaciones (middleware). [17] (Ver figura 1.3)

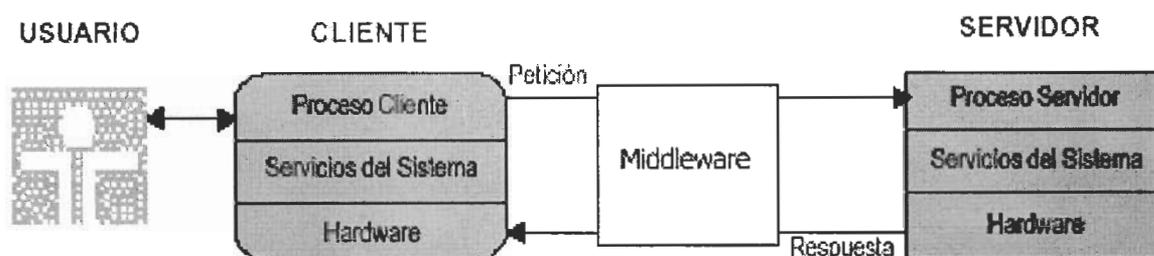


Figura 1.3. Componentes de la Arquitectura Cliente/Servidor.

### **1.6.7 Tecnologías de Desarrollo.**

Las tecnologías que compiten por la creación de aplicaciones Cliente / Servidor han reinventado la computación Cliente / Servidor, no solo en cuanto a bases de datos, sino también con los objetos distribuidos y el Web.

En las tecnologías de diseño HTML fue el lenguaje inicial para incluir funcionalidades que sus usuarios iban demandando, como la capacidad de mostrar contenido multimedia. Esto fue dando lugar a la aparición de sucesivas versiones del lenguaje cada vez más potentes hasta llegar a la versión actual.

La principal debilidad que inicialmente tenía este sistema consistía en que las páginas HTML eran simples ficheros colocados en un servidor Web que se las enviaba a sus clientes tal cuales, sin ninguna posibilidad interacción con los mismos excepto, la utilización de sus hipervínculos y forzando a que cualquier actualización de ellos tuviese que ser realizada editando su contenido en el servidor. Esto, a medida que fue creciendo la red y utilizándose profesionalmente como vehículo con el que realizar operaciones de comercio electrónico, fue volviéndose inadmisibles, y se empezaron a pedir mayores posibilidades de interacción que permitiesen realizar aplicaciones Web con capacidades similares a las de las aplicaciones tradicionales, así como mecanismos con que actualizar su contenido. Por eso, a lo largo de las diferentes revisiones que se fueron haciendo de HTML, se fue introduciendo en el lenguaje y en el funcionamiento de los servidores Web diferentes mecanismos destinados a permitir la modificación y generación dinámica del contenido de las páginas Web, obteniéndose como resultado la página dinámica. [22]

#### **1.6.7.1 Tecnologías del lado del Servidor.**

Estas tecnologías son interpretadas y ejecutadas por el servidor. Se consideran de mayor importancia ya que son las que garantizan el intercambio de datos en ambos sentidos, entre el navegador del cliente y el servidor Web, y a su vez las que soportan la funcionalidad del sitio.[23]

### PHP (Professional Home Pages)

PHP es un lenguaje de programación que se ejecuta del lado del servidor, cuyo código se combina con el HTML, para procesar la información de formularios, generar páginas con contenidos dinámicos, o mandar y recibir cookies. Dispone de múltiples herramientas que permiten acceder a bases de datos de forma sencilla, por lo que es ideal para crear aplicaciones en Internet.[24]

*Los principales usos del PHP son los siguientes:*

- Programación de páginas Web dinámicas, habitualmente en combinación con el motor de base datos MySQL, aunque cuenta con soporte nativo para otros motores, incluyendo el estándar ODBC, lo que amplía en gran medida sus posibilidades de conexión.
- Programación en consola, al estilo de Perl, en Linux, Windows y Macintosh.
- Creación de aplicaciones gráficas independientes del navegador, por medio de la combinación de PHP y GTK, que permite desarrollar aplicaciones de escritorio tanto para los sistemas operativos basados en Unix, como para Windows y Macintosh.[24]

### *Ventajas de PHP*

- La principal ventaja se basa en ser un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad.
- Leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una muy buena documentación en su página oficial.
- Es Libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos. [24]

### ASP (Active Server Pages)

Es una tecnología del lado servidor de Microsoft para páginas Web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Server (IIS).

ASP ha pasado por cuatro iteraciones mayores, ASP 1.0 (distribuido con IIS 3.0), ASP 2.0 (distribuido con IIS 4.0), ASP 3.0 (distribuido con IIS 5.0) y ASP.NET (parte de la plataforma .NET de Microsoft). Las páginas pueden ser generadas mezclando código de scripts del lado del servidor, permite acceso a base de datos para extraer información y visualizarla en el navegador, envío de correo electrónico, acceso al sistema de archivos del servidor. Una potencialidad en ASP está en el uso de controles ActiveX construidos para ser ejecutados en el servidor.[25]

### Perl (Practical Extraction and Report Language)

Es un lenguaje de programación desarrollado por Larry Wall (lwall at netlabs.com) inspirado en otras herramientas de UNIX. No establece ninguna filosofía de programación concreta. No se puede decir que sea orientado a objetos, modular o estructurado aunque soporta directamente todos estos paradigmas; su punto fuerte son las labores de procesamiento de textos y archivos.

No es ni un compilador ni un intérprete, está en un punto intermedio, cuando mandamos a ejecutar un programa en Perl, se compila el código fuente a un código intermedio en memoria que se optimiza como si se fuera a elaborar un programa ejecutable pero es ejecutado por un motor, como si se tratase de un intérprete.[26]

Es un lenguaje de programación basado en scripts portable a casi cualquier plataforma. Uno de sus elementos más potentes son las expresiones regulares, que a partir de su versión en Perl han sido adoptadas por otros lenguajes y plataformas como .NET o Javascript.[26]

### JSP (Java Server Pages)

La tecnología JSP, o de Java Server Pages, es una tecnología Java que permite a los programadores generar dinámicamente HTML, XML (sigla en inglés de eXtensible Markup Language, 'lenguaje de marcado extensible') o algún otro tipo de página web. Esta tecnología permite al código Java y a algunas acciones predefinidas ser embebidas en el contenido estático. En las JSP, se escribe el texto que va a ser devuelto en la salida (normalmente código HTML) incluyendo código java dentro de él para poder modificar o generar contenido dinámicamente.

La principal ventaja de JSP frente a otros lenguajes es que permite integrarse con clases Java (class) lo que permite separar en niveles las aplicaciones web, almacenando en clases java las partes que consumen más recursos así como las que requieren más seguridad, y dejando la parte encargada de formatear el documento html en el archivo jsp.[27]

Además Java se caracteriza por ser un lenguaje que puede ejecutarse en cualquier sistema, lo que sumado a jsp le da mucha versatilidad.

Sin embargo JSP no se puede considerar un script al 100% ya que antes de ejecutarse el servidor web compila el script y genera un servlet, por lo tanto se puede decir que aunque este proceso sea transparente para el programador no deja de ser una aplicación compilada. La ventaja de esto es algo más de rapidez y disponer del API de Java en su totalidad.[27]

### Asp.NET

Es la tecnología de Microsoft .Net para el desarrollo de páginas Web dinámicas. El cual trae diversas mejoras con respecto al ASP tradicional entre las que se destacan:

*Rapidez en programación:* Mediante diversos controles se puede con unas pocas líneas y en menor tiempo mostrar toda una base de datos y hacer rutinas complejas.

*Programación Multilenguaje:* El programador ya no se encuentra restringido a la utilización de VBScript o JScript para programar las páginas ASP. Puede utilizar cualquiera admitido por la plataforma .NET, lo que incluye un gran abanico de lenguajes como C#, Visual Basic.NET, JScript.NET, COBOL.NET, C++.NET etc.

*Programación visual:* La arquitectura esta diseñada de forma que el desarrollo de las aplicaciones Web sea análogo a la forma en que se hace en las aplicaciones de ventanas estándar, representando los controles como objetos que se pueden manipular en la programación, al igual que se hace en las librerías de ventanas orientadas a objetos. Además permitiendo que con herramientas visuales, diseñarlas sea fácil como arrastrar los componentes necesarios sobre la página y hacer doble clic, sobre cada componente para escribir la lógica de respuesta a los eventos que se puedan realizar sobre ellos, de forma parecida a como se hace en entornos RAD (Rapid Application Development) como Delphi.[28]

### **1.6.7.2 Tecnologías en el lado cliente.**

Una forma de conseguir aumentar el dinamismo de los sitios web consistió en dotar al lenguaje HTML de la capacidad de poder incluir *scripts* de código escritos en diversos lenguajes (como JavaScript o VBScript) que los navegadores interpretarían para, a través de ellos, modificar el aspecto de las páginas e interactuar con el navegador. A esto se le conoce como **DHTML** (*Dynamic HTML*) y forma parte de las denominadas **tecnologías en el lado del cliente**, pues requiere de un soporte directo por parte del navegador del cliente. Ahí es precisamente donde se encuentra su principal problema: los fabricantes de navegadores nunca han llegado a un acuerdo a la hora de implementar esto, y cada uno soporta tanto diferentes lenguajes de *scripts* como diversas capacidades dentro de un mismo lenguaje, lo que hace que sea difícil la creación y mantenimiento de páginas DHTML compatibles con los navegadores más utilizados.[29]

### DHTML (HTML dinámico).

No es un lenguaje de programación, sino que, más bien se lo puede definir como el conjunto de técnicas que permiten controlar la apariencia de una página, interactuar con el usuario e incluir elementos activos en ella. (Se basa en convertir las etiquetas tradicionales del HTML en objetos programables, que se pueden manipular con JavaScript u otros lenguajes.) Permite actualizar los elementos de la página en cualquier momento, incluso después de que la página se haya cargado, sin tener que establecer una nueva conexión con el servidor.

*Los componentes más importantes son:*

- Hojas de estilo en cascada (CSS: *Cascading Style Sheets*)
- Lenguajes de Script
- Modelo de objetos del documento: DOM (*Document Object Model*)
- Además podemos añadir algunas otras posibilidades como acceso a bases de datos, controles multimedia o carga automática de fuentes.[30]

### JavaScript.

Es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página Web. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Se pueden crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones JavaScript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador. [31]

Entre las acciones típicas que se pueden realizar en JavaScript tenemos dos vertientes. Por un lado los efectos sobre páginas Web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. Por el otro, JavaScript nos permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que podemos

crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo. No solo permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones, estructuras de datos complejas, etc. [31]

### VBScript.

Visual Basic Script es un lenguaje de script del lado del cliente, directamente derivado de Visual Basic, pero solo compatible con Internet Explorer. Su modo de funcionamiento para construir efectos en páginas Web es muy similar al utilizado en JavaScript, y los recursos a los que se puede acceder también son los mismos: el navegador.

Los lenguajes de script se ejecutan interpretados, no compilados. Esto significa que el código escrito en un lenguaje de script no sufre ninguna transformación previa a su ejecución. Cada línea de código es traducida a lenguaje máquina justo antes de su ejecución. Después es ejecutada y la traducción no es almacenada. Si es necesaria otra ejecución, el intérprete se verá obligado a realizar una nueva traducción de cada línea de código. [32]

### **1.6.8 Tecnología utilizada en el desarrollo de la aplicación.**

#### **1.6.8.1 Microsoft .NET:**

Microsoft .NET es un conjunto de tecnologías de software de Microsoft para conectar su mundo de información, gente, sistemas y dispositivos, con el objetivo de obtener una plataforma sencilla y potente para distribuir el software en forma de servicios que puedan ser suministrados remotamente y que puedan comunicarse y combinarse unos con otros de manera totalmente independiente de la plataforma, lenguaje y modelo de componentes con los que hayan sido desarrollados. Esta es la llamada plataforma.NET y a los servicios antes mencionados se les denomina servicios Web.[33]

Para crear aplicaciones para la plataforma .NET, tanto de servicios como aplicaciones tradicionales (aplicaciones de consola, aplicaciones de ventanas, servicios de Windows NT, etc.), Microsoft ha publicado el denominado kit (SDK o Kit, 'Software Development Kit') de desarrollo de software conocido como .NET Framework SDK, que incluye las herramientas necesarias tanto para su desarrollo como para su distribución y ejecución y Visual Studio.NET, que permite hacer todo lo anterior desde una interfaz visual basada en ventanas.[31, 33]

A largo plazo Microsoft pretende reemplazar la API Win32 o Windows API con la plataforma .NET. Esto debido a que la API Win32 o Windows API fue desarrollada sobre la marcha, careciendo de documentación detallada, uniformidad y cohesión entre sus distintos componentes, provocando múltiples problemas en el desarrollo de aplicaciones para el sistema operativo Windows.

La plataforma .NET pretende solventar la mayoría de estos problemas proveyendo un conjunto único y expandible con facilidad, de bloques interconectados, diseñados de forma uniforme y bien documentados, que permitan a los desarrolladores tener a mano todo lo que necesitan para producir aplicaciones sólidas.[34]

Con esta plataforma Microsoft incursiona de lleno en el campo de los Servicios Web y establece el XML como norma en el transporte de información en sus productos y lo promociona como tal en los sistemas desarrollados utilizando sus herramientas.[34]

.NET intenta ofrecer una manera rápida y económica pero a la vez segura y robusta de desarrollar aplicaciones, permitiendo a su vez una integración más rápida y ágil entre sus componentes.[34]

.NET Framework

Constituye la plataforma que simplifica el desarrollo de aplicaciones en un entorno altamente distribuido como es Internet.

.NET Framework contiene tres componentes principales: el Entorno de Ejecución Común de Lenguajes o Common Language Runtime (CLR); la jerarquía de clases básicas de la plataforma o .NET Framework Base Classes y el motor de generación de interfaz de usuario, que permite crear interfaces para la Web o para el tradicional entorno Windows.[35]

Este constituye la base de la plataforma .Net y denota la infraestructura sobre la cual se reúnen un conjunto de lenguajes, herramientas y servicios que simplifican el desarrollo de aplicaciones en entorno de ejecución distribuido. [34]

La Figura 1.4 muestra un esquema con los elementos que conforman el entorno .NET Framework.

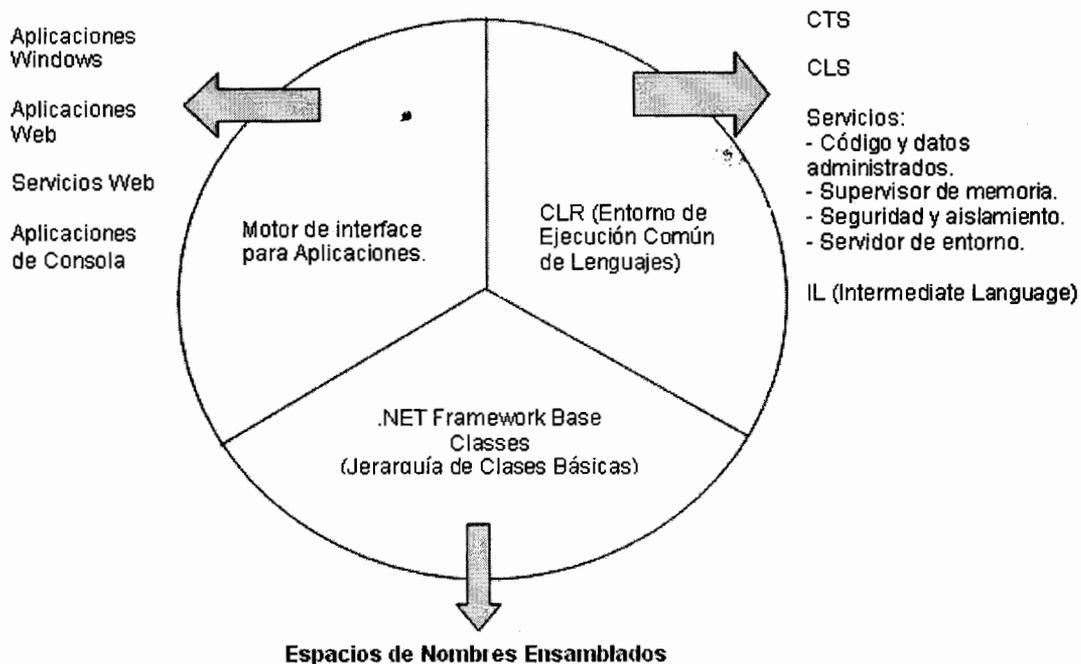


Figura1.4. Entorno .NET Framework

Common Language Runtime (CLR)

El CLR es el verdadero núcleo del Framework de .Net, entorno de ejecución en el que se cargan las aplicaciones desarrolladas en los distintos lenguajes, ampliando el conjunto de servicios del sistema operativo (W2k y W2003). La Figura 1.5 muestra un esquema con los elementos que conforman el CLR.

La herramienta de desarrollo compila el código fuente de cualquiera de los lenguajes soportados por .Net en un código intermedio (MSIL, Microsoft Intermediate Lenguaje), similar al BYTECODE de Java. Para generar dicho código el compilador se basa en el Common Language Specification (CLS) que determina las reglas necesarias para crear ese código MSIL compatible con el CLR. Para ejecutarse se necesita un segundo paso, un compilador JIT (Just-In-Time) es el que genera el código máquina real que se ejecuta en la plataforma del cliente.

La compilación JIT la realiza el CLR a medida que el programa invoca métodos, el código ejecutable obtenido, se almacena en la memoria caché del ordenador, siendo recompilado de nuevo sólo en el caso de producirse algún cambio en el código fuente.[35]



Figura 1.5 Entorno Común de Ejecución para lenguajes (CLR)

### Biblioteca de Clases Base de .Net

La Biblioteca de Clases Base (BCL) maneja la mayoría de las operaciones básicas que se encuentran involucradas en el desarrollo de aplicaciones, incluyendo entre otras. (Ver Figura 1.6)

- Interacción con los dispositivos periféricos.
- Manejo de datos (ADO.NET).
- Administración de memoria.
- Cifrado de datos.
- Transmisión y recepción de datos por distintos medios (XML, TCP/IP).
- Administración de componentes Web que corren tanto en el servidor como en el cliente (ASP.NET).
- Manejo y administración de excepciones.
- Herramientas de despliegue de gráficos (GDI+).
- Herramientas de seguridad e integración con la seguridad del sistema operativo.
- Manejo de tipos de datos unificado.
- Interacción con otras aplicaciones.
- Manejo de cadenas de caracteres y expresiones regulares.
- Operaciones aritméticas.
- Manipulación de fechas, zonas horarias y periodos de tiempo.
- Manejo de arreglos de datos y colecciones.
- Manipulación de archivos de imágenes.
- Aleatoriedad.
- Generación de código.
- Manejo de idiomas.
- Auto descripción de código.
- Interacción con el API Win32 o Windows API.
- Compilación de código.

Esta funcionalidad se encuentra organizada por medio de espacios de nombres jerárquicos.[34]

La Biblioteca de Clases Base se clasifica, en tres grupos clave:

1. ASP.NET y Servicios Web XML.
2. Windows Forms.
3. ADO.NET

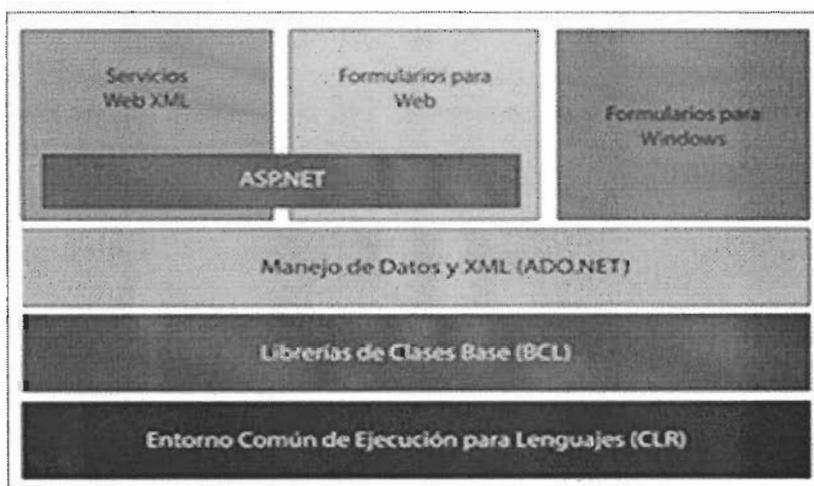


Figura1.6 Biblioteca de Clases Base (BCL)

### Ensamblados



Figura1.7 Ensamblados

Los ensamblados son ficheros con forma de EXE o DLL que contienen toda la funcionalidad de la aplicación de forma encapsulada. Con ellos ya no es necesario registrar los componentes de la aplicación.[34] (Ver figura 1.7)

### Servicio Web ASP.NET

Arquitectura para el desarrollo de sitios Web que pone a disposición formularios Web Forms y servicios Web XML. A través del servidor Web IIS las aplicaciones ASP.NET se ejecutarán bajo el CLR. Se puede usar el conjunto de clases del .NET Framework para desarrollar dichas aplicaciones.

Los programadores tradicionales de Windows tienen a su disposición las clases de Windows Forms, que simplifican enormemente el desarrollo de la interfaz GUI (Interfaz gráfica de usuario) para Windows. [35, 36]

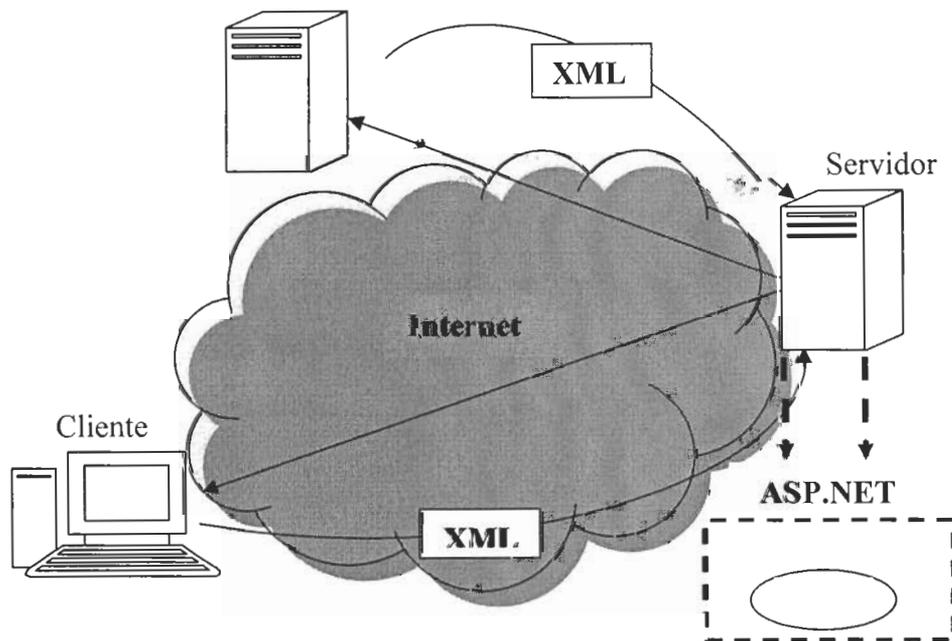


Figura 1.7. Uso de un servicio Web.

Los servicios Web emplean formato de lenguaje de marcación XML (es un formato de texto que cumple con ciertas reglas, el cual puede ser escrito y leído por cualquier lenguaje o sistema operativo) en todos aquellos puntos que involucren la red, aunque con posterioridad, dentro de la microcomputadora la tarea se gestione utilizando los mismos medios y recursos empleados con ensamblados locales. La Figura 1.7 muestra el uso de un servicio Web.[35, 36]

### **1.6.9 Fundamentación del lenguaje, gestor de BD y servidor Web a utilizar.**

Para el desarrollo del Sistema de Asignación de Transportes (SAT) se utiliza un servidor Web con IIS, PostgreSQL como servidor de base de datos y tecnología ASP. NET con C# como lenguaje de programación en el servidor y javascript en el cliente. La decisión se fundamenta claramente con el estudio y comparación de las tendencias y tecnologías que existen en la actualidad sobre las que se apoya la propuesta, realizado en el epígrafe.

#### **1.6.9.1 PostgreSQL**

Los sistemas de mantenimiento de Bases de Datos relacionales tradicionales (DBMS,s) soportan un modelo de datos que consisten en una colección de relaciones con nombre, que contienen atributos de un tipo específico. En los sistemas comerciales actuales, los tipos posibles incluyen numéricos de punto flotante, enteros, cadenas de caracteres, cantidades monetarias y fechas. Está generalmente reconocido que este modelo será inadecuado para las aplicaciones futuras de procesamiento de datos. El modelo relacional sustituyó modelos previos en parte por su "simplicidad espartana". Sin embargo, como se ha mencionado, esta simplicidad también hace muy difícil la implementación de ciertas aplicaciones. Postgres ofrece una potencia adicional sustancial al incorporar los siguientes cuatro conceptos adicionales básicos en una vía en la que los usuarios pueden extender fácilmente el sistema. [37]

- Clases.
- Herencia.
- Tipos.
- Funciones.

*Otras características aportan potencia y flexibilidad adicional:*

- Restricciones (Constraints).
- Disparadores (triggers).
- Reglas (rules).
- Integridad transaccional

Estas características colocan a Postgres en la categoría de las Bases de Datos identificadas como objeto relacionales. Nótese que éstas son diferentes de las referidas como orientadas a objetos, que en general no son bien aprovechables para soportar lenguajes de Bases de Datos relacionales tradicionales. Postgres tiene algunas características que son propias del mundo de las bases de datos orientadas a objetos. De hecho, algunas Bases de Datos comerciales han incorporado recientemente características en las que Postgres fue pionera. [37]

### **1.6.9.2 C#**

Visual C# .NET proporciona a los programadores un lenguaje moderno orientado a componentes con el que pueden crear con rapidez soluciones interesantes controladas por datos. Gracias a la rapidez de diseño, programación e implementación de servicios Web XML, las soluciones controladas por datos de alto rendimiento y los diseñadores visuales de servidor, Visual C# .NET ofrece funcionalidad superior para optimizar procesos empresariales. Permite a los programadores generar soluciones para una gama amplísima de clientes, incluidas aplicaciones Web, aplicaciones basadas en Windows Forms y otras. Además, con un lenguaje de programación elegante y moderno, los programadores de C# pueden aprovechar sus conocimientos de C++ y Java para disfrutar de una experiencia satisfactoria con la plataforma Microsoft .NET.[38]

*El lenguaje proporciona la capacidad de generar componentes de sistema duraderos en virtud de las siguientes características:*

- Total compatibilidad entre **COM** y plataforma para integración de código existente.
- Gran robustez, gracias a la recolección de elementos no utilizados (liberación de memoria) y a la seguridad en el tratamiento de tipos.

- Seguridad implementada por medio de mecanismos de confianza intrínsecos del código.
- Plena compatibilidad con conceptos de metadatos extensibles.

*Además, es posible interactuar con otros lenguajes, entre plataformas distintas, y con datos heredados, en virtud de las siguientes características:*

- Plena interoperabilidad por medio de los servicios de **COM+ 1.0** y **.NET Framework** con un acceso limitado basado en bibliotecas.
- Compatibilidad con XML para interacción con componentes basados en tecnología Web.
- Capacidad de control de versiones para facilitar la administración y la implementación.[38]

### **1.6.9.3 Internet Information Server 6.0**

IIS es un servidor Web de funciones completas que posibilita la creación de aplicaciones Web y servicios Web XML. La arquitectura de IIS 6.0 ha sido completamente reconstruida, con un nuevo modelo de proceso de tolerancia a errores que mejora significativamente la confiabilidad de las aplicaciones y los sitios Web.

Ahora, IIS puede aislar una aplicación Web individual o varios sitios en un proceso autocontenido (llamado un grupo de aplicaciones) que se comunica directamente con el núcleo del sistema operativo. Esta función aumenta el rendimiento y la capacidad de las aplicaciones, proporcionando a la vez más espacio libre en los servidores, con lo que se reducen de forma efectiva los requisitos de hardware. Estos grupos de aplicaciones autocontenidos impiden que una aplicación o un sitio interrumpan los servicios Web XML u otras aplicaciones Web del servidor.

IIS también ofrece capacidades de supervisión del estado con el fin de descubrir, recuperar e impedir errores en las aplicaciones Web. En Windows Server 2003, Microsoft ASP.NET usa de forma nativa el nuevo modelo de proceso de IIS.

### **1.6.10 Fundamentación de la metodología utilizada.**

¿Cuál metodología a seguir es la mejor? Todos en algún momento se han hecho esta pregunta, cuando han tenido que desarrollar un software. Y de hecho esta pregunta se torna muy importante pues debemos tener un plano en que apoyarse. Todo desarrollo de software es riesgoso y difícil de controlar, pero si no llevamos una metodología de por medio, lo que obtenemos es clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos.

El mundo de la informática no para de hablar de procesos de desarrollo, el modo de trabajar eficientemente para evitar catástrofes que llevan a que un gran porcentaje de proyectos se terminen sin éxito.

El objetivo de un proceso de desarrollo es garantizar la calidad del software (en todas las fases por las que pasa) a través de una mayor transparencia y control sobre el proceso. Da igual si es un producto genérico o para un cliente, hay que producir lo esperado en el tiempo esperado y con el coste esperado. Es labor del proceso de desarrollo hacer que esas medidas para aumentar la calidad sean reproducibles en cada desarrollo.

Muchas veces no se toma en cuenta el utilizar una metodología adecuada, sin pensar lo importante de buscar cuál sería la metodología de desarrollo más apropiada. Muchas veces el éxito de un software radica en haber encontrado la metodología correcta que sea capaz de cumplir con el objetivo del sistema.

#### **1.6.10.1 Rational Unified Process (RUP)**

El Proceso Unificado Racional es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo)
- Pretende implementar las mejores prácticas en Ingeniería de Software.
- Desarrollo iterativo.
- Administración de requisitos.
- Uso de arquitectura basada en componentes.
- Control de cambios.
- Modelado visual del software.
- Verificación de la calidad del software.

El RUP es un producto de Rational (IBM). Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).[39]

*El RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al final de cada ciclo, cada ciclo se divide en fases que finalizan con un hito donde se debe tomar una decisión importante:*

- Inicio: se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos.
- Elaboración: se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.
- Construcción: se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario.
- Transición: se implementa el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requerimientos a ser analizados.

### *Fases del RUP*

- Establece oportunidad y alcance.
- Identifica las entidades externas o actores con las que se trata.
- Identifica los casos de uso. [39]

### **1.6.10.2 Unified Modeling Language(UML)**

Lenguaje Unificado de Modelado (UML), es el lenguaje de modelado de sistemas de software más conocido en la actualidad; aún cuando todavía no es un estándar oficial, está apoyado en gran manera por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

El punto importante para notar aquí es que UML es un "lenguaje" para especificar y no un método o un proceso. UML se usa para definir un sistema de software; para detallar los artefactos en el sistema; para documentar y construir -es el lenguaje en el que está descrito el modelo. UML se puede usar en una gran variedad de formas para soportar una metodología de desarrollo de software (tal como el Proceso Unificado de Rational), pero no especifica en sí mismo qué metodología o proceso usar.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas.

### *Estandarización de UML*

Además de haberse convertido en un estándar de facto, UML es un estándar industrial promovido por el grupo OMG al mismo nivel que el estándar CORBA para intercambio de objetos distribuidos. Para la revisión de UML se formaron dos "corrientes" que promovían la aparición de la nueva versión desde distintos puntos

de vista. Finalmente se impuso la visión más industrial frente a la académica. Recientemente se ha publicado la versión 2.0 en la que aparecen muchas novedades y cambios que, fundamentalmente, se centran en resolver carencias prácticas. Además, esta versión recibe diversas mejoras que provienen del lenguaje SDL.[40]

### Tipos de Diagrama en UML

- 1) Diagramas de estructura enfatizan en los elementos que deben existir en el sistema modelado:
  - a) Diagrama de clases.
  - b) Diagrama de componentes.
  - c) Diagrama de objetos.
  - d) Diagrama de estructura compuesta.
  - e) Diagrama de despliegue.
  - f) Diagrama de paquetes.
- 2) Diagramas de comportamiento enfatizan en lo que debe suceder en el sistema modelado:
  - a) Diagrama de actividades.
  - b) Diagrama de casos de uso.
  - c) Diagrama de estados.
- 3) Diagramas de Interacción, un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado:
  - a) Diagrama de secuencia.
  - b) Diagrama de comunicación.
  - c) Diagrama de tiempos (UML 2.0)

### **1.7 Conclusiones**

En este capítulo se logró definir los objetivos del sistema propuesto, después de identificar el dominio del problema a resolver partiendo del estudio de la organización y los intereses que se persiguen con el proyecto. Se analizó la situación actual y los problemas que enfrenta la asignación de transporte, sirviendo esto como base para la elaboración de la solución propuesta.

La variada gama de tecnologías existentes actualmente para el desarrollo de aplicaciones Web constituye un factor importante a la hora de tomar la decisión de cuál es la que se hace necesario emplear. Teniendo en cuenta los requerimientos del sistema propuesto, se selecciona para la implementación del SAT la plataforma .Net y el lenguaje C#, por su facilidad y rapidez en la confección de sistemas, con PostgreSQL como gestor de base de dato.

Además para el análisis y el diseño del sistema se sigue la Metodología de desarrollo con tecnología orientada a objetos Rational Unified Process (RUP) que utiliza notación UML (Unified Modeling Language), la cual garantiza la elaboración de todas las fases de un producto de software.

### **2.1 Introducción**

Este capítulo abordará todo lo que tiene que ver con el modelo de negocio y el sistema propuesto. Los procesos propuestos partiendo de las reglas del negocio, así como los procesos propuestos para la realización del sistema. Para una mejor comprensión del negocio se ilustrarán diagramas de casos de uso del negocio y del sistema. En el caso de los procesos del negocio están en formato expandido para una mejor comprensión, ya que el funcionamiento de SAT esta regido por los mismos.

### **2.2 Reglas del negocio a considerar**

El área de Infraestructura Productiva, cuenta con un pequeño número de vehículos que no son capaces de cubrir las solicitudes de transporte en una jornada, por lo que se hace necesario pedir refuerzo a otras direcciones de la universidad y contar con sus autos para poder dar respuesta a la demanda planteada.

El Solicitante es la persona interesada en solicitar un medio de transporte para realizar un viaje vinculado a las actividades de la infraestructura productiva (IP), la solicitud presentada se le otorga una prioridad en dependencia inmediata. El solicitante contacta con la Planificadora, especie de secretaria de IP quien es la encargada de comunicarle la respuesta indicada.

Si el lugar que se desea ser trasladado no es contemplado en los autorizados, se necesita contactar con un superior que autorice tal situación.

La Responsable de Área le ofrece la información a la planificadora en tiempo puede programar un viaje en auto del área.

Para realizar un viaje en un vehículo que no pertenezca al área IP tiene que ser autorizado por el directivo al área que pertenece.

### 2.3 Descripción de los procesos del negocio propuesto

#### Actores del Negocio

Tabla 2.1. Descripción de los actores del negocio

Nombre del actor	Descripción
Solicitante	Persona que solicita transporte para cumplir una función del área Infraestructura Productiva. Actúa en el caso de uso Controlar Solicitudes.
Directivo	Persona encargada de otorgarle un plan de trabajo del auto. Solicita información de los viajes y estado de los autos. Participa en los casos de uso Controlar Estado de Autos, Controlar Estado de Viajes y Reportes.

### 2.4 Diagrama de casos de uso del negocio

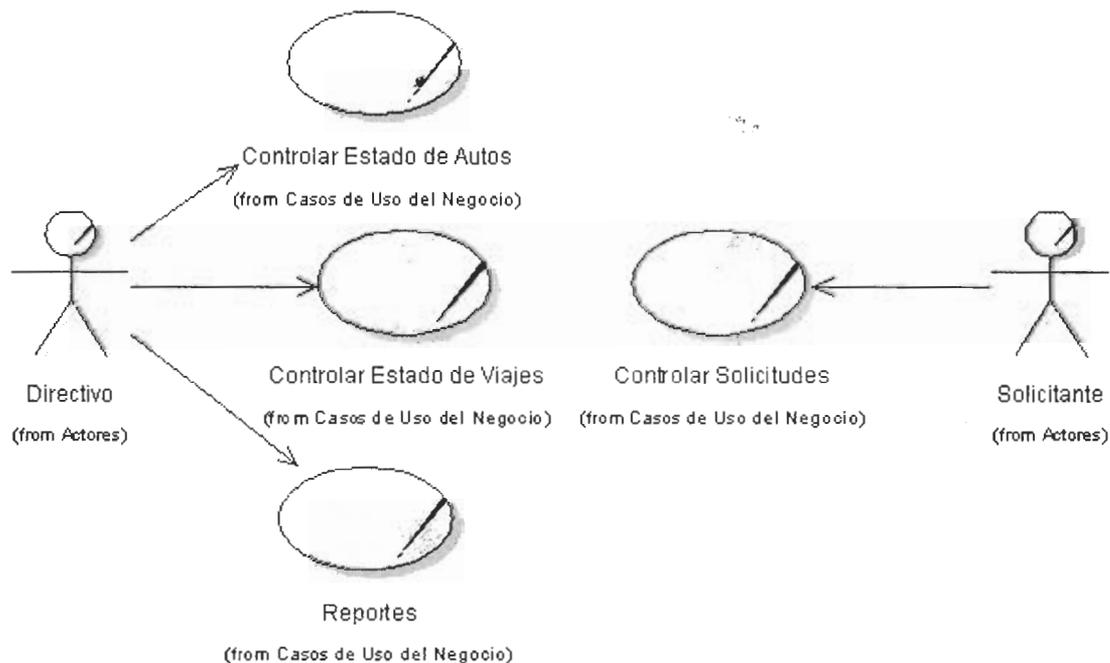


Figura 2.1. Diagrama de casos de uso del negocio

## 2.5 Trabajadores del negocio

Tabla 1.2. Descripción de los trabajadores del negocio

Nombre del trabajador	Descripción
Responsable de área	Trabajador que mantiene actualizado el estado y control de los autos del área. Participa en el caso de uso Controlar Estado del Auto.
Planificadora	Trabajador que procesa solicitudes, lleva el control de los viajes. Emite los avisos de la solicitud. Participa en el caso de uso Controlar Solicitudes y Controlar Estado de Viajes.

## 2.6 Casos de uso del negocio

### 2.6.1 Caso de uso Controlar Estado de Autos

Tabla 2.3. Descripción del caso de uso Controlar Estado de Autos.

Caso de Uso	Controlar Estado de Autos
<b>Actores</b>	Directivo(Inicia)
<b>Propósito</b>	Controlar el estado y plan de trabajo del auto.
<b>Resumen:</b> El caso de uso se inicia cuando un Directivo contacta con su Responsable de Área, encargada de que haga la planificación de trabajo de los autos del área según agenda de trabajo. La Responsable de Área selecciona el auto, busca su estado de funcionamiento, en dependencia del resultado le planifica plan de trabajo. El caso de uso termina cuando el directivo recibe notificación de la planificación de los autos.	
<b>Acción del actor</b>	<b>Respuesta del proceso de negocio</b>

<p>1- El Directivo comunica realizar planificación de los autos a la Responsable de área.</p> <p>3- El Directivo recibe notificación de no existencia del auto.</p>	<p>2- La Responsable de área lista los autos del área. En caso de existencia, se procede al paso 4, de lo contrario se le comunica al Directivo la notificación de usuario existente.</p> <p>4- La Responsable de área planifica plan de trabajo del auto.</p>
<b>Prioridad</b>	-----
<b>Mejoras</b>	
<b>Cursos alternos</b>	-----

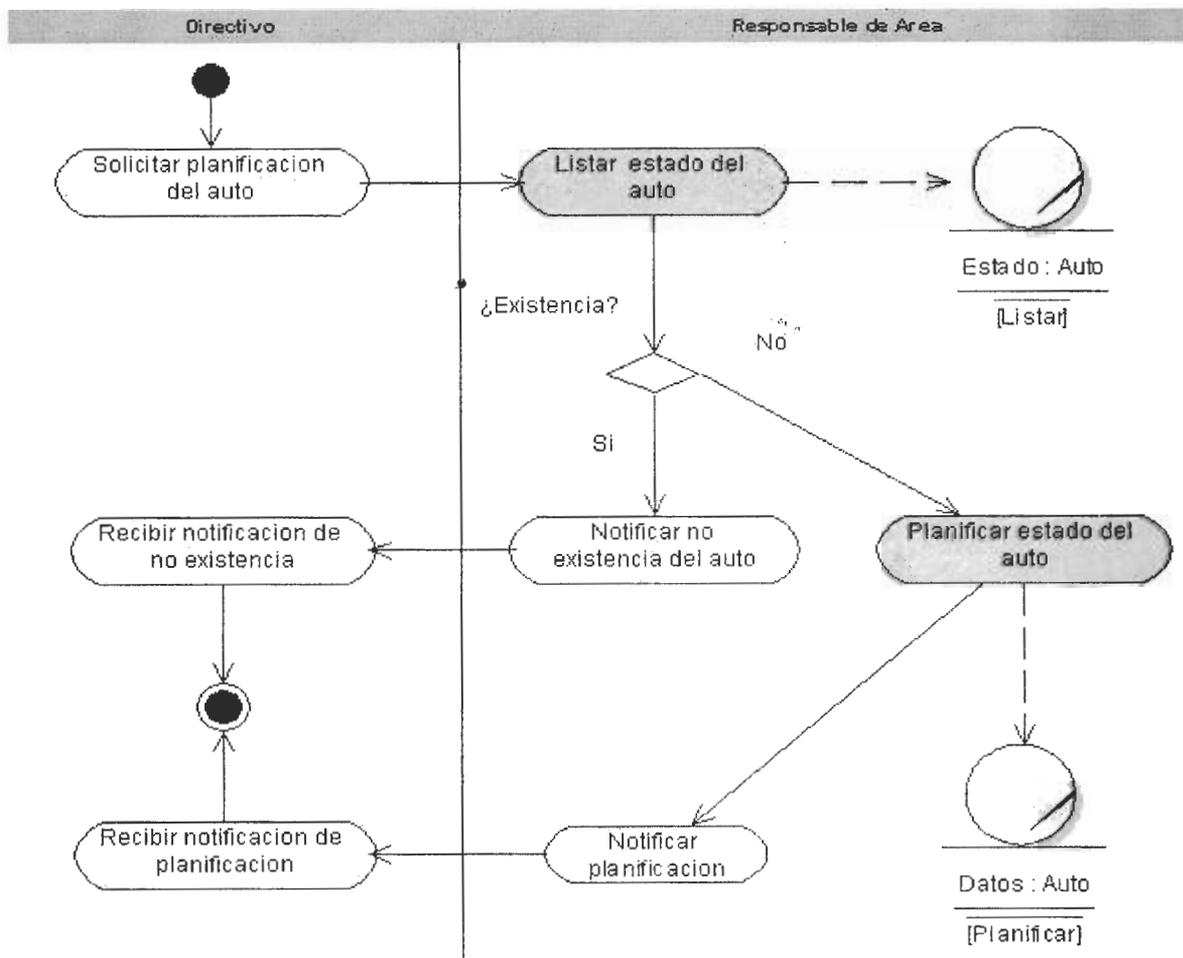


Figura 2.2. Diagrama del caso de uso Controlar Estado de Autos

2.6.2 Caso de uso Controlar Estado de Viajes

Tabla 2.4. Descripción del caso de uso Controlar Estado de Viajes

Caso de Uso		Controlar Estado de Viajes	
<b>Actores</b>	Directivo(Inicia)		
<b>Propósito</b>	Controlar el estado de los viajes.		
<b>Resumen:</b> El caso de uso se inicia cuando el Directivo envía solicitud de control de los viajes. La Planificadora lista los posibles viajes a efectuar. Si el lugar destino a viajar ya esta contemplado en los autorizados se crean los viajes con ese lugar, de lo contrario se crea el lugar destino para crear el viaje solicitado. El caso de uso termina con la notificación de la solicitud de control de viajes.			
Acción del actor		Respuesta del proceso de negocio	
1- El Directivo envía solicitud de control de viaje.		2- La Planificadora lista los viajes, si el lugar no existe se procede al paso 4, de lo contrario notifica al Directivo de que el viaje se encuentra entre los autorizados.	
3- El Directivo recibe notificación de viaje realizada.		4- La Planificadora crea el lugar para poder crear el viaje autorizado.	
<b>Prioridad</b>	-----		
<b>Mejoras</b>			
<b>Cursos alternos</b>	-----		

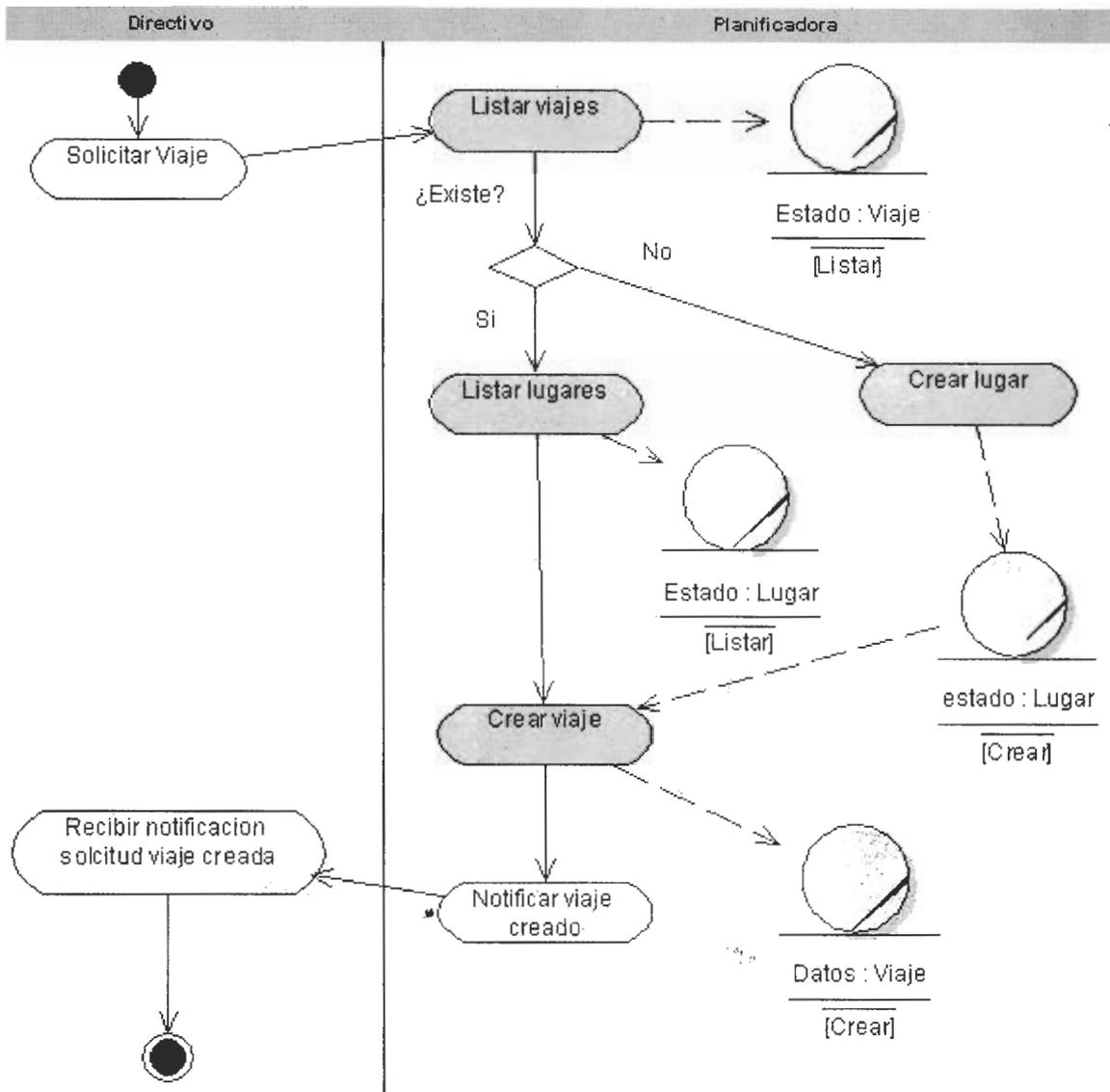


Figura 2.3. Diagrama del caso de uso Controlar Estado de Viajes

2.6.3 Caso de uso Controlar Solicitudes

Tabla 2.5. Descripción del caso de uso Controlar Solicitudes

Caso de Uso		Controlar Solicitudes	
<b>Actores</b>	Solicitante(Inicia)		
<b>Propósito</b>	Controlar el estado de las solicitudes.		
<b>Resumen:</b> El caso de uso se inicia cuando el Solicitante hace una solicitud de transporte. La Planificadora lista las solicitudes para verificar que la solicitud planteada no se encuentra entre las que se están procesando. Se crea la solicitud de transporte en caso de que no exista. El caso de uso termina con la notificación a la solicitante de la propuesta planteada.			
Acción del actor		Respuesta del proceso de negocio	
1- El Solicitante envía solicitud de viaje.		2- La Planificadora lista los viajes, si la solicitud no existe se procede al paso 4, de lo contrario se notifica al solicitante de que su solicitud ya ha sido planteada por otro usuario.	
3- El Solicitante recibe notificación de solicitud de viaje planteada.		4- La Planificadora crea solicitud de viaje.	
<b>Prioridad</b>	-----		
<b>Mejoras</b>			
<b>Cursos alternos</b>	-----		

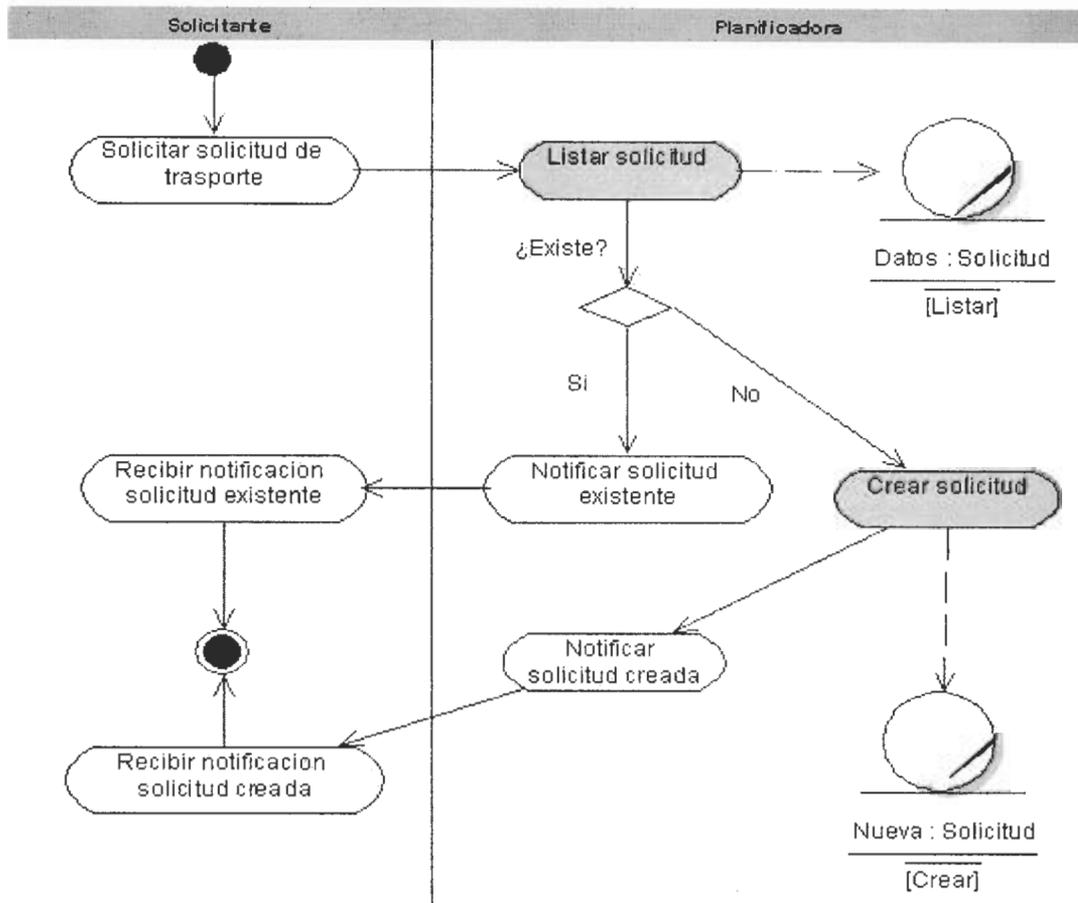


Figura 2.4. Diagrama del caso de uso Controlar Solicitudes

### 2.6.4 Caso de uso Controlar Reportes

Tabla 2.5. Descripción del caso de uso Controlar Reportes

Caso de Uso	Reportes
<b>Actores</b>	Directivo (Inicia)
<b>Propósito</b>	Obtener información de la actividad del área.
<b>Resumen:</b> El caso de uso se inicia cuando el Directivo le hace la solicitud de obtener los reportes de información de los servicios que presta el área a IP. La Responsable de área lista los reportes para obtener el indicado, en caso que no exista procede a crear el tipo de reporte solicitado. El caso de uso termina con la notificación y respuesta de la solicitud de reportes.	
<b>Acción del actor</b>	<b>Respuesta del proceso de negocio</b>

<p>1- El Directivo envía solicitud de reportes.</p> <p>3- El Directivo recibe notificación de solicitud de reportes procesada.</p>	<p>2- La Responsable de área lista los reportes, si el reporte solicitado no existe se procede al paso 4, de lo contrario se notifica al Directivo de que su reporte ha sido procesado.</p> <p>4- La Responsable de área crea reportes del área.</p>
<p><b>Prioridad</b></p>	<p>-----</p>
<p><b>Mejoras</b></p>	<p></p>
<p><b>Cursos alternos</b></p>	<p>-----</p>

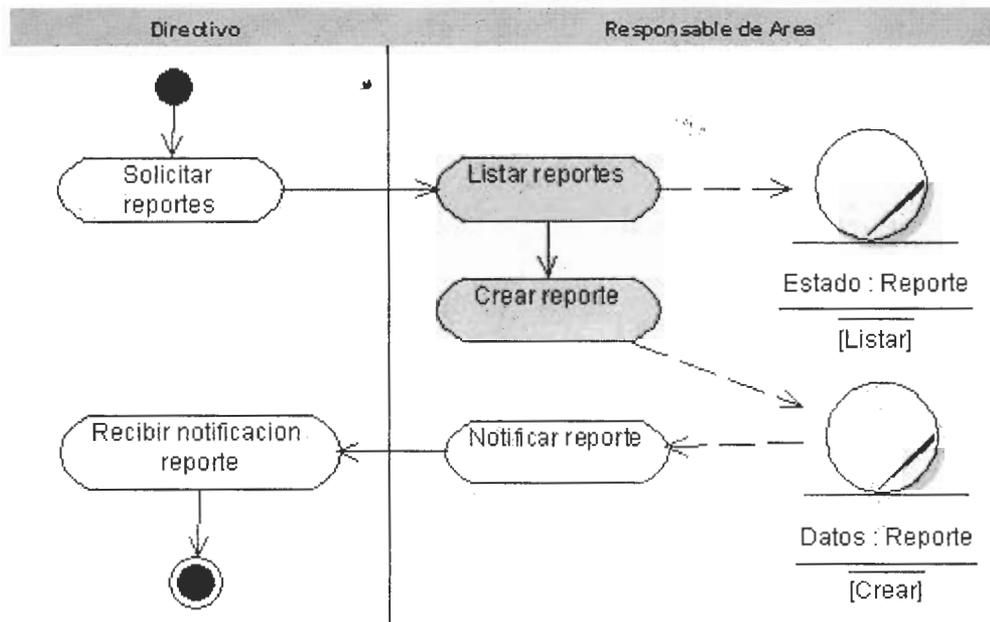


Figura 2.5. Diagrama del caso de uso Controlar Reportes

## 2.7 Modelo de objetos

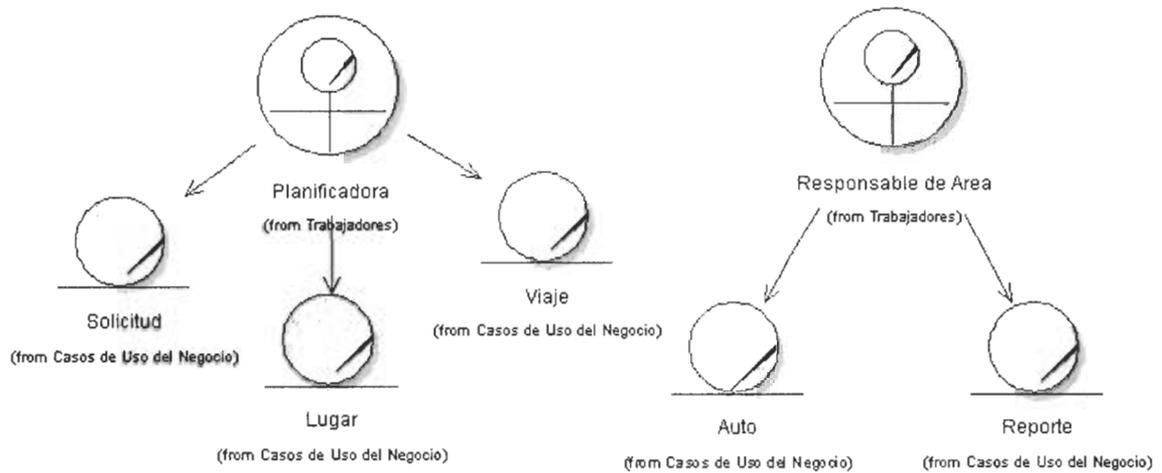


Figura 2.6. Diagrama de clases del modelo de objetos

## 2.8 Conclusiones

Con la culminación de este capítulo queda definido como es el modelo de negocio, los actores y trabajadores, el diagrama de casos de uso del negocio. Se hacen las descripciones de los casos de uso y se obtiene el diagrama de clases del modelo de objetos.

### 3.1 Introducción

### 3.2 Actores del sistema a automatizar

Tomando en consideración que los actores se caracterizan por lo siguiente:

- No son parte del sistema, son roles de un usuario.
- Pueden intercambiar información con el sistema.
- Pueden ser un recipiente pasivo de información.
- Pueden representar a un humano a una máquina o un software.

Se definen a continuación los actores del sistema

Tabla 2.1. Definición de actores del sistema a automatizar

Nombre del actor	Descripción
Administrador	Es el encargado de que el sistema funcione correctamente, además se encarga de las actualizaciones del mismo.
Planificadora	Opera la aplicación Web para los procesos de viajes solicitudes y lugares.
Responsable de Área	Encargada de operar en la Web para mantener actualizado el estado de los autos. Obtiene los reportes de información del área.

### 3.3 Paquetes y sus relaciones

La aplicación Web SAT necesita información de la Base de Datos UCI Usuarios para el acceso al sistema.

### **3.4 Requerimientos funcionales.**

Los requerimientos funcionales son aquellos requisitos que, desde el punto de vista de las necesidades del usuario, debe cumplir el sistema y que están fuertemente ligados a las opciones del programa.

1. Gestionar usuarios.
  - 1.1 Listar Usuarios.
  - 1.2 Insertar usuario.
  - 1.3 Eliminar usuario.
  - 1.4 Editar usuario.
2. Gestionar grupo.
  - 2.1 Listar grupos.
  - 2.2 Insertar grupo.
  - 2.3 Eliminar grupo.
  - 2.4 Editar grupo.
3. Gestionar viajes.
  - 3.1 Listar viajes.
  - 3.2 Insertar viaje.
  - 3.3 Eliminar viaje.
  - 3.4 Editar viaje.
4. Gestionar lugar.
  - 4.1 Listar lugares.
  - 4.2 Insertar lugar.
  - 4.3 Eliminar lugar.
  - 4.4 Editar lugar.
5. Gestionar solicitud.
  - 5.1 Listar solicitudes.
  - 5.2 Crear solicitud.
  - 5.3 Calendario de solicitud.
  - 5.4 Responder solicitud.
6. Gestionar auto.

- 6.1 Listado de autos.
- 6.2 Insertar auto.
- 6.3 Eliminar auto.
- 6.4 Editar auto.
- 6.5 Historial del auto.
- 7. Gestionar estados.
  - 7.1 Definir estado del viaje.
    - 7.1.1 Listar estados del viaje.
    - 7.1.2 Insertar estado del viaje.
    - 7.1.3 Eliminar estado del viaje.
    - 7.1.4 Editar estado del viaje.
  - 7.2 Definir estado del auto.
    - 7.2.1 Listar estados del auto.
    - 7.2.2 Insertar estado del auto.
    - 7.2.3 Eliminar estado del auto.
    - 7.2.4 Editar estado del auto.
  - 7.3 Definir estado de la solicitud.
    - 7.3.1 Listar estados de la solicitud.
    - 7.3.2 Insertar estado de la solicitud.
    - 7.3.3 Eliminar estado de la solicitud.
    - 7.3.4 Editar estado de la solicitud.
- 8. Gestionar accesos de la información.
  - 8.1 Listar accesos.
  - 8.2 Insertar acceso.
  - 8.3 Eliminar acceso.
  - 8.4 Editar acceso.
- 9. Gestionar bloques.
  - 9.1 Listar bloques.
  - 9.2 Insertar bloque.
  - 9.3 Eliminar bloque.
  - 9.4 Editar bloque.

10. Gestionar hipervínculos.

- 10.1 Listar hipervínculos.
- 10.2 Insertar hipervínculo.
- 10.3 Eliminar hipervínculo.
- 10.4 Editar hipervínculo.

11. Gestionar recursos.

- 11.1 Listar recursos
- 11.2 Insertar recurso.
- 11.3 Eliminar recurso.
- 11.4 Editar recurso.

12. Gestionar direcciones.

- 12.1 Listar direcciones.
- 12.2 Insertar dirección.
- 12.3 Eliminar dirección.
- 12.4 Editar dirección.
- 12.5 Detalles de la dirección.

### ***3.5 Definición de los requerimientos no funcionales.***

Se trata de características que describen alguna forma o restricción para la realización de algún requerimiento (funcionalidad) o conjunto de ellas e inclusive todos los requerimientos.

Los requerimientos no funcionales son los atributos del sistema, cualidades que debe tener el producto, los cuales aparecen a continuación.

#### **Apariencia o interfaz externa (ambiente Web):**

- Amigable al usuario.
- De fácil utilización.
- De opciones variables de acuerdo a los roles y accesos asignados a los usuarios.

**Usabilidad:**

- Facilidad de uso por personas sin experiencia previa con las computadoras.
- Consistencia en la interfaz de usuario.
- Alta productividad ganada con la introducción del producto para los trabajadores de la universidad ya que aumenta su desempeño laboral.

**Rendimiento:**

- El sistema debe soportar más de 100 transacciones por segundo.
- Respuesta rápida ante las solicitudes de los usuarios.
- Alto grado de eficiencia.
- Aprovechamiento máximo de los recursos en los clientes.
- Disponible en cualquier instante.
- Se utilizará almacenamiento duro para la información con que trabajará el sistema.

**Soporte:**

- El sistema será instalado y configurado por los especialistas de la dirección de Informatización, quienes se encargarán de darle mantenimiento.
- Fácil adaptabilidad para asumir nuevas funciones.

**Portabilidad:**

- Diseñado para Windows. No obstante puede llegar a ser multiplataforma, todo está en la portabilidad que alcance el .NET Framework SDK.

**Seguridad:**

Gran parte de los elementos de la seguridad están supeditados al SGBD PostgreSQL y al .NET Framework de la tecnología .NET, puesto que la propuesta del sistema está soportada básicamente por ellos. A continuación se exponen algunos elementos acerca de las opciones que brindan estas tecnologías para la

seguridad de las aplicaciones, haciendo énfasis principalmente en las que se disponen para aplicaciones Web.

- **Integridad de datos en PostgreSQL.**

Mediante PostgreSQL se puede garantizar la integridad de la información, mediante chequeos de la misma, para protegerla de estados inconsistentes. Existen diferentes formas de asegurar la integridad de los datos en PostgreSQL: integridad de dominio, integridad de entidad, integridad referencial, integridad definida por el usuario.

La integridad del dominio es la validación de las entradas para una columna dada. Se puede forzar la integridad del dominio restringiendo el tipo de dato, el formato o el rango de posibles valores.

La integridad referencial conserva las restricciones definidas entre las tablas cuando se insertan las filas o se borran. En PostgreSQL, la integridad referencial se basa en las restricciones entre las llaves extranjeras (foreign keys) y las llaves primarias (primary keys) o entre las llaves extranjeras y las restricciones únicas (UNIQUE). La integridad referencial asegura que los valores de la llave son consistentes a través de las tablas.

La integridad definida por el usuario le permite definir reglas de negocio específicas que no entran en una de las otras categorías de integridad. La herramienta que brinda en este caso son los procedimientos almacenados y disparadores.

- **Seguridad de datos en PostgreSQL.**

El PostgreSQL brinda: una arquitectura de seguridad basada en usuarios y grupos de usuarios; posibilidades para la elaboración de planes de seguridad que identifican cuáles usuarios pueden acceder a cuáles datos, y qué actividades pueden realizar en la base datos; creación y administración de cuentas de seguridad para la autenticación de los usuarios.

De forma general la seguridad de los datos se realiza mediante:

Login: Para lograr la conexión con una instancia del servidor PostgreSQL en cuestión.

Usuario: Permite determinado nivel de acceso en el ámbito de la base de datos para la que se crea, y que radica en el servidor accedido a través del “login” correspondiente.

Rol: Especifica determinado privilegio asociado a la operación sobre un servidor de bases de datos PostgreSQL o un conjunto determinado de objetos de este tales como base de datos, tablas. Se les denomina privilegios a las acciones que se permiten que un usuario realice sobre un objeto de la base de datos (SELECT, INSERT, DELETE, etc).

Se pueden conceder varios roles para un usuario, pero sólo uno determina el nivel de privilegios sobre los objetos del servidor PostgreSQL de un usuario, autenticado a través de un login determinado.

Los usuarios son los actores en la base de datos. Cada vez que el SGBD realiza una operación de consulta, actualización o ejecución de un procedimiento, lo hace en beneficio de un usuario. El SGBD controla las autorizaciones de acceso en dependencia del “login” del usuario en cuestión. [40] SQL Server Books Online.

- **Seguridad del .NET Framework.**

.NET Framework dispone de varios mecanismos para proteger recursos y código del acceso por parte de usuarios y código no autorizados:

Seguridad ASP.NET de aplicaciones Web: ofrece una manera de ayudar a limitar el acceso a un sitio comparando las credenciales autenticadas (o representaciones de las mismas) con los permisos del sistema de archivos de Microsoft Windows NT o con un archivo XML que contiene la lista de usuarios autorizados, funciones autorizadas o verbos HTTP autorizados.

Seguridad de acceso a código: utiliza permisos para ayudar a limitar el acceso que el código tiene a recursos y operaciones protegidos. Ayuda a proteger los sistemas de equipos de código móvil malicioso y ayuda a permitir la ejecución

segura del código móvil. La seguridad de acceso a código junto con las directivas que la rigen se denomina seguridad basada en evidencias.

Seguridad basada en funciones: proporciona la información necesaria para tomar decisiones acerca de lo que un usuario puede hacer. Estas decisiones se pueden basar en la identidad del usuario, en la pertinencia a una función, o en ambos. [33]

- **Seguridad ASP.NET de aplicaciones Web.**

ASP.NET funciona junto con .NET Framework e IIS para ayudar a proporcionar aplicaciones Web seguras. En el establecimiento de la seguridad de una aplicación ASP.NET, se deben realizar las dos funciones principales que se describen a continuación:

Autenticación: consiste en obtener las credenciales de identificación, como nombre y contraseña, de un usuario y validarlas consultando a una autoridad determinada. Si las credenciales son válidas, se considera a la entidad que ha enviado las credenciales como una entidad autenticada. ASP.NET implementa este proceso a través de proveedores de autenticación, que son módulos que contienen el código necesario para autenticar las credenciales del solicitante. Una vez autenticada la identidad, el proceso de autorización determina si esa identidad tiene acceso a un recurso específico.

Autorización: Limita los derechos de acceso mediante la concesión o negación de permisos específicos a una identidad autenticada. Una de las formas de autorizar el acceso a un recurso dado es la autorización de dirección URL, realizada mediante `URLAuthorizationModule`. Este módulo se puede utilizar para permitir o denegar el acceso de forma selectiva a partes arbitrarias del espacio de nombres URL a determinados conjuntos, usuarios o funciones.

Además, IIS puede conceder o negar el acceso en función de la dirección IP o del nombre de la computadora del usuario. Cualquier autorización de acceso posterior se realiza mediante la autorización de la dirección URL del permiso de acceso al sistema de archivos NTFS.

Puesto que ASP.NET se basa en Microsoft .NET Framework, el programador de aplicaciones ASP.NET también tiene acceso a todas las características de seguridad integradas de .NET Framework, como la seguridad de acceso a código y la seguridad de acceso basada en funciones. [33]

A continuación la propuesta de seguridad:

- Definición de roles: Administrador, Responsable de área organizacional (ResponsableAO), Responsable de área de interés (ResponsableAI), Especialista.
- Asignación de roles a usuarios, y límites de acceso a la información por grupos de trabajo y por usuarios, lo que garantiza la confidencialidad mostrándose la información a la persona indicada.
- Para preservar la integridad de los datos, las operaciones de actualización están a cargo de los roles definidos de modo que los cambios son realizados por personas autorizadas. Además se aprovecha la potencialidad que al respecto brinda el SGBD PostgreSQL, definiendo un usuario con acceso a la base de datos y estableciéndole el rol público con permisos a las operaciones necesarias. Se utiliza el control de concurrencia optimista para las actualizaciones.
- Para la autenticación, cada usuario tiene su identificador y contraseña que son verificados por la aplicación, así se puede acceder a información u opciones restringidas. Además, aprovecha el mecanismo de seguridad ASP.NET de aplicaciones Web brindada por .NET Framework para proteger recursos, configurando la autorización a ficheros para el rol Administrador de manera que sólo el usuario autenticado que responde a este rol podrá acceder a la página de administración. Se establece la conexión a la base de datos mediante un usuario y contraseña predefinidos, que son verificados por el SGBD PostgreSQL que registra este usuario.
- Una única identificación del usuario garantizará la disponibilidad de la información para la que esté autorizado.

- Cifrado de contraseñas para su transmisión segura por la red.
- En el servidor Web existe un cortafuego para impedir ataques de intrusos desde el exterior.
- El local donde está la computadora Servidor que alojará la aplicación cuenta con acceso restringido.
- Sólo se utilizará en la Universidad de Ciencias Informáticas.

### **Políticos-culturales:**

- El sistema estará disponible en idioma Español.

### **Confiabilidad:**

- Tolerancia a fallo correspondiente al PostgreSQL.

### **Legales:**

- Debe cumplir con las normas y leyes establecidas en el país, nos referimos en este caso a las leyes de nuestro sistema de seguridad y protección.

### **Ayuda y documentación en línea:**

- El sistema es muy fácil de usar por lo que no posee ayuda.
- Manual de usuario donde se exponen las restricciones de diseño a las que deben acogerse los usuarios que editen información a publicar.

### **Software:**

En el servidor:

- Sistema Operativo Windows 2000 Advanced Server.
- Servicios de Internet Information Server (IIS) 5.0 como servidor Web, o cualquier otro que presente soporte para ejecutar aplicaciones de la tecnología ASP.NET.
- PostgreSQL como gestor de base de datos relacionales.
- .NET Framework SDK 1.0.
- Microsoft Data Access Components (MDAC) 2.7.

- Servicio SSL para la transferencia de información cifrada.

En el cliente:

- Sistema Operativo Windows 98 o superior.
- Internet Explorer 5.01 o superior.

**Hardware:**

Para el servidor (mínimo):

- Pentium II con 128 MB de RAM y un microprocesador a 300 MHz, 6 Gb de disco duro.

Para el cliente (mínimo):

- Pentium II con 64 MB de RAM y un microprocesador a 300 MHz.

**Restricciones en el diseño y la implementación:**

- Como artefactos para el diseño se usan los que propone RUP apoyado en el estándar notacional UML.
- Sujeto a los estándares establecidos para una aplicación Web.
- PostgreSQL como gestor de bases de datos.
- Diseño prefijado por los diseñadores de Informatización.

### 3.6 Descripción de los casos de uso

#### 3.6.1 Descripción de los casos de uso del Paquete Control Usuarios

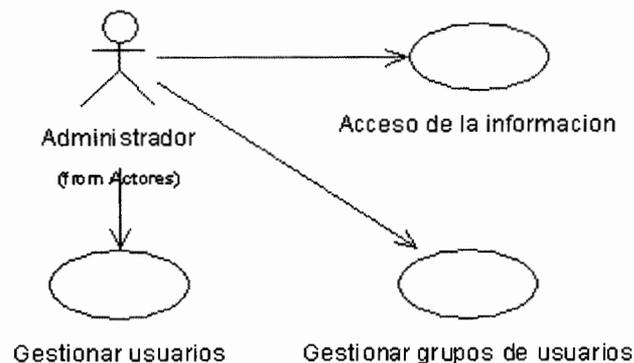


Figura 3.1. Diagrama de los casos de uso del Paquete Control Usuarios

Tabla 3.2. Descripción del caso de uso Gestionar Usuarios

Nombre del caso de uso		Gestionar Usuarios
<b>Actores</b>	Administrador (inicia)	
<b>Propósito</b>	Establecer el control de los usuarios con acceso al sistema.	
Resumen. El administrador es el encargado de garantizar la gestión de usuarios. La gestión de usuarios permite el acceso a la aplicación y desempeñar las funciones en correspondencia del mismo. Permite listar los usuarios con que cuenta el SAT, además de insertar, eliminar y editar un usuario.		
Referencias	R1.1, R1.2, R1.3, R1.4	
Precondiciones	El Administrador cuenta con permisos administrativos.	
Poscondiciones	Se crea el usuario.	

Tabla 3.3. Descripción del caso de uso Gestionar grupos de usuarios

Nombre del caso de uso		Gestionar grupos de usuarios
<b>Actores</b>	Administrador (inicia)	
<b>Propósito</b>	Establecer los permisos de los usuarios al sistema.	
Resumen. El administrador es el encargado de garantizar la gestión de grupos de usuarios. La gestión de grupos de usuarios permite controlar los permisos con que cuenta un usuario para el uso del SAT. Se puede listar los grupos de usuarios, además de insertar, eliminar y editar un grupo de usuario.		
Referencias	R2.1, R2.2, R2.3, R2.4	
Precondiciones	El Administrador cuenta con permisos administrativos.	
Poscondiciones	Se crea el grupo de usuario.	

Tabla 3.4. Descripción del caso de uso Acceso a la información

<b>Nombre del caso de uso</b>	Acceso a la información
<b>Actores</b>	Administrador (inicia)
<b>Propósito</b>	Establecer el acceso a la información disponible.
Resumen. El administrador es el encargado de garantizar el acceso a la información. El acceso a la información permite controlar la información que puede ser visualizada por los usuarios dependiendo al grupo de usuarios al que pertenece. Se puede listar los accesos a la información, además de insertar, eliminar y editar un acceso a la información.	
<b>Referencias</b>	R8.1, R8.2, R8.3, R8.4
<b>Precondiciones</b>	El Administrador cuenta con permisos administrativos.
<b>Poscondiciones</b>	Se crea el acceso a la información.

### 3.6.2 Descripción de los casos de uso del Paquete Control Viajes

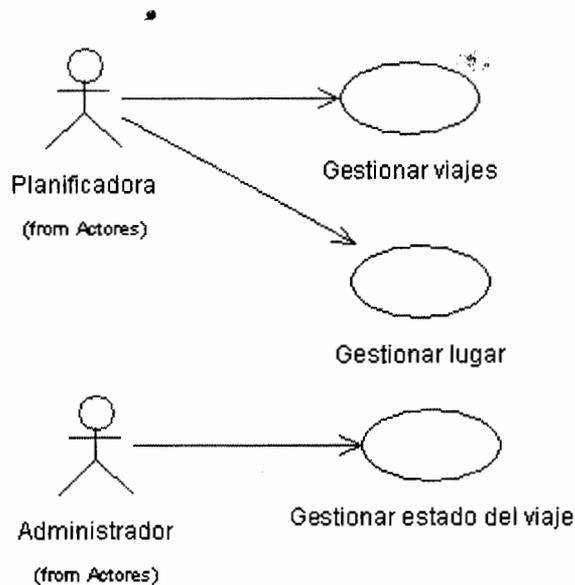


Figura 3.2. Diagrama de los casos de uso del Paquete Control Viajes

Tabla 3.5. Descripción del caso de uso Gestionar viajes

Nombre del caso de uso		Gestionar viajes
<b>Actores</b>	Planificadora (inicia)	
<b>Propósito</b>	Establecer la gestión de los viajes.	
Resumen. La planificadora es la encargada de garantizar el control de los viajes. Las opciones que brinda la gestión de un viaje es conocer el estado de un viaje. Otras opciones es listar todos los viajes, permitir insertar, eliminar y editar un viaje.		
Referencias	R3.1, R3.2, R3.3, R3.4	
Precondiciones	La Planificadora es la única que trabaja con los viajes.	
Poscondiciones	Se crea el viaje.	

Tabla 3.6. Descripción del caso de uso Gestionar lugar

Nombre del caso de uso		Gestionar lugar
<b>Actores</b>	Planificadora (inicia)	
<b>Propósito</b>	Establecer la gestión de los lugares.	
Resumen. La planificadora es la encargada de garantizar el control de los lugares de recogida y destino. Los lugares destino son los establecidos por la infraestructura productiva para hacer un viaje. Las opciones que brinda la gestión de lugar es conocer los lugares a los que se puede hacer un viaje por la opción listar lugares. Otras opciones son, permitir insertar, eliminar y editar un lugar.		
Referencias	R4.1, R4.2, R4.3, R4.4	
Precondiciones	La Planificadora es la única que trabaja con los lugares.	
Poscondiciones	Se crea el lugar.	

Tabla 3.7. Descripción del caso de uso Gestionar estado del viaje

<b>Nombre del caso de uso</b>	Gestionar estado del viaje
<b>Actores</b>	Administrador (inicia)
<b>Propósito</b>	Establecer la gestión de los estados del viaje.
<p>Resumen.</p> <p>El Administrador es el encargado de garantizar el control de la gestión de los estados que puede tener un viaje. Un viaje presenta diferentes estados que dan información en que fase se encuentran. Las opciones que brinda la gestión de los estados del viaje es listar todos los estados del viaje, insertar, eliminar y editar un estado de viaje.</p>	
<b>Referencias</b>	R7.1.1, R7.1.2, R7.1.3, R7.1.4
<b>Precondiciones</b>	El Administrador tiene permisos administrativos.
<b>Poscondiciones</b>	Se crea el estado del viaje.

### 3.6.3 Descripción de los casos de uso del Paquete Control Solicitud

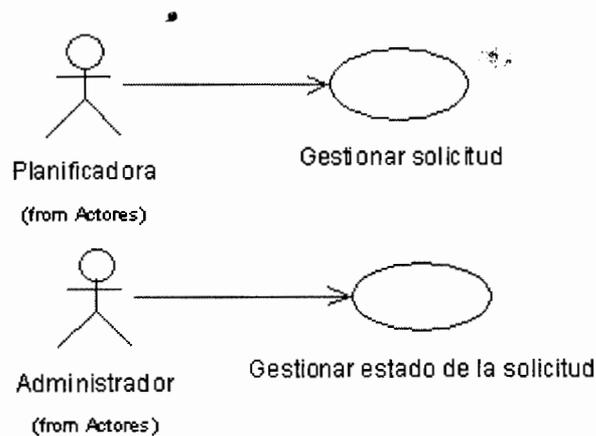


Figura 3.3. Diagrama de los casos de uso del Paquete Control Solicitudes

Tabla 3.8. Descripción del caso de uso Gestionar solicitud

Nombre del caso de uso		Gestionar solicitud
<b>Actores</b>	Planificadora (inicia)	
<b>Propósito</b>	Establecer la gestión de las solicitudes.	
Resumen. La Planificadora es la encargada de garantizar el control de la gestión de las solicitudes. Cuando recibe una solicitud es la encargada de darle el tratamiento que corresponde. La gestión de solicitud permite listar las solicitudes, crear una solicitud, mostrar el calendario de solicitudes y dar respuesta a una solicitud.		
Referencias	R5.1, R5.2, R5.3, R5.4	
Precondiciones	La Planificadora es la única que trabaja con la solicitud.	
Poscondiciones	Se crea la solicitud.	

Tabla 3.9. Descripción del caso de uso Gestionar estado de la solicitud

Nombre del caso de uso		Gestionar estado de la solicitud
<b>Actores</b>	Administrador (inicia)	
<b>Propósito</b>	Establecer los estados de la solicitud.	
Resumen. El Administrador es el encargado de garantizar la gestión de los estados que puede tener una solicitud. Una solicitud presenta diferentes estados que dan información de la fase que transita. Las opciones que brinda la gestión de estados de la solicitud es listar los estados de la solicitud, insertar, eliminar y editar un estado de la solicitud.		
Referencias	R7.3.1, R7.3.2, R7.3.3, R7.3.4	
Precondiciones	El Administrador tiene permisos administrativos.	
Poscondiciones	Se crea el estado de la solicitud.	

3.6.4 Descripción de los casos de uso del Paquete Control Autos

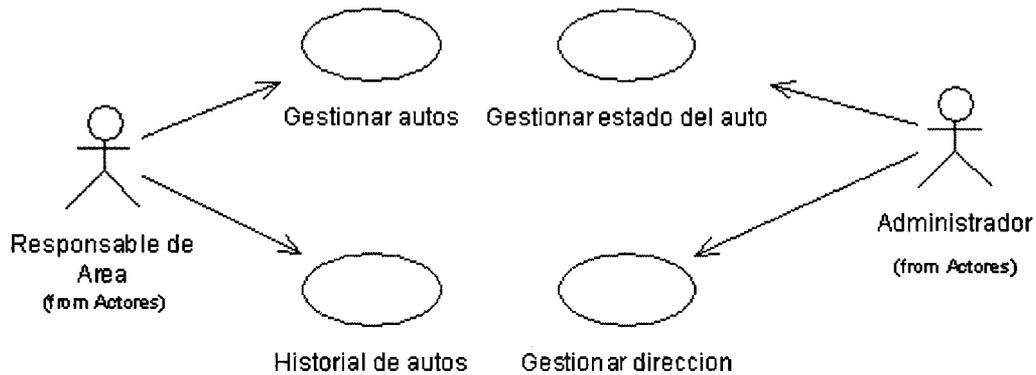


Figura 3.4. Diagrama de los casos de uso del Paquete Control Autos

Tabla 3.10. Descripción del caso de uso Gestionar autos

Nombre del caso de uso		Gestionar autos
Actores	Responsable de área (inicia)	
Propósito	Establecer el control de los autos.	
Resumen.		
La Responsable de área es la encargada de garantizar la gestión de los autos, consiste en controlar las características de los autos. La gestión de los autos permite listar los autos, insertar, eliminar y editar un auto.		
Referencias	R6.1, R6.2, R6.3, R6.4	
Precondiciones	La Responsable de área controla los autos.	
Poscondiciones	Se crea el auto.	

Tabla 3.11. Descripción del caso de uso Gestionar estado del auto

Nombre del caso de uso		Gestionar estado del auto
<b>Actores</b>	Administrador (inicia)	
<b>Propósito</b>	Establecer los estados del auto.	
Resumen. El Administrador es el encargado de llevar el control de los estados que puede presentar un auto. La gestión de estados de autos permite listar los estados, insertar, eliminar y editar un estado del auto.		
Referencias	R7.2.1, R7.2.2, R7.2.3, R7.2.4	
Precondiciones	El Administrador cuenta con permisos administrativos.	
Poscondiciones	Se crea el estado del auto.	

Tabla 3.12. Descripción del caso de uso Gestionar dirección

Nombre del caso de uso		Gestionar dirección
<b>Actores</b>	Administrador (inicia)	
<b>Propósito</b>	Establecer la información de las direcciones.	
Resumen. El Administrador es el encargado de gestionar las direcciones de la universidad. Una vez nacida una nueva dirección, el administrador deberá registrarla. La gestión de dirección permite además de insertar, listar, editar, eliminar y obtener los detalles de una dirección.		
Referencias	R12.1, 12.2,12.3,12.4,12.5	
Precondiciones	El Administración tiene permisos administrativos.	
Poscondiciones	Se crea la dirección.	

Tabla 3.13. Descripción del caso de uso Historial del auto

Nombre del caso de uso		Historial de autos
<b>Actores</b>	Responsable de área (inicia)	
<b>Propósito</b>	Establecer la información de los autos.	
Resumen.		
La Responsable de área es la encargada de reportar la información de los autos, se obtiene toda la información con los autos.		
Referencias	R6.5	
Precondiciones	El historial depende del control de autos.	
Poscondiciones	Se crea el historial.	

### 3.6.5 Descripción de los casos de uso del Paquete Control Autos

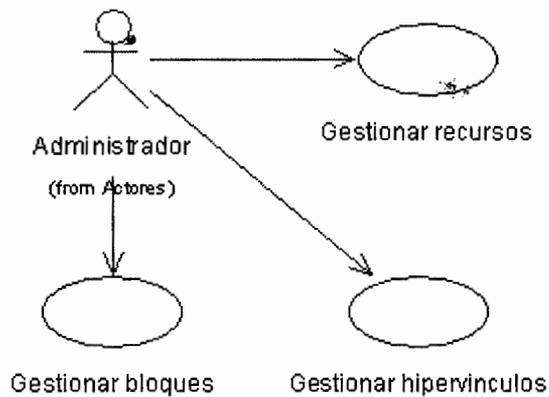


Figura 3.5. Diagrama de los casos de uso del Paquete Control Navegación

Tabla 3.14. Descripción del caso de uso Gestionar bloques

Nombre del caso de uso		Gestionar bloques
<b>Actores</b>	Administrador (inicia)	
<b>Propósito</b>	Establecer el control de los bloques de la página.	
Resumen. Consiste en llevar el control de los bloques que tiene la aplicación cuando es visualizada. La gestión de bloques es la forma dinámica de listar los bloques, insertar bloque, eliminar y editar un bloque		
Referencias	R9.1, R9.2, R9.3, R9.4	
Precondiciones	El Administrador tiene permisos administrativos.	
Poscondiciones	Se crea el bloque.	

Tabla 3.15. Descripción del caso de uso Gestionar hipervínculos

Nombre del caso de uso		Gestionar hipervínculos
<b>Actores</b>	Administrador (inicia)	
<b>Propósito</b>	Establecer el control de los hipervínculos de la página.	
Resumen. Consiste en llevar el control de los hipervínculos que tiene la aplicación cuando es visualizada. La gestión de hipervínculos es la forma dinámica de listar los hipervínculos, insertar hipervínculos, eliminar y editar un hipervínculos.		
Referencias	R9.1, R9.2, R9.3, R9.4	
Precondiciones	El Administrador tiene permisos administrativos.	
Poscondiciones	Se crea el hipervínculos.	

Tabla 3.16. Descripción del caso de uso Gestionar recursos

Nombre del caso de uso		Gestionar recursos
<b>Actores</b>	Administrador (inicia)	
<b>Propósito</b>	Establecer el control de los recursos de la página.	
Resumen. Consiste en llevar el control de los recursos con que cuenta la aplicación cuando es operada. La gestión de recursos es la forma dinámica de listar los recursos, insertar recursos, eliminar y editar un recurso.		
Referencias	R9.1, R9.2, R9.3, R9.4	
Precondiciones	El Administrador tiene permisos administrativos.	
Poscondiciones	Se crea el hipervínculos.	

### 3.7 Conclusiones

Con este capítulo se llegó a una idea general del sistema propuesto, representada en el modelo de casos de uso del sistema, y que responde a las necesidades del usuario.

### **4.1 Introducción**

En el presente capítulo se modelan los artefactos necesarios para la construcción de aplicaciones Web. Los componentes de la aplicación son tratados como clases y mediante la utilización de UML se podrán representar a través de diagramas de clases Web. Se presenta el modelo de datos que es la base para construir finalmente la base de datos que soportará el trabajo del sistema y el modelo de despliegue donde se representan los nodos en los que se distribuye la aplicación.

### **4.2 Diagrama de clases del diseño Web**

El diagrama de clases para las Aplicaciones Web difiere un poco del resto de las aplicaciones que estamos acostumbrados a construir puesto que en ellas son más importantes la modelación de la lógica y estado del negocio que los detalles de presentación. Para obtener un nivel correcto de abstracción y detalle que nos permita obtener un resultado final es mejor modelar los artefactos del sistema es decir: modelar las páginas, los enlaces entre estas, todo el código que irá creando las páginas, así como el contenido dinámico de estas, una vez que estén en el navegador del cliente; estos son los artefactos que necesitamos modelar para que el desarrollador los implemente luego y obtener así nuestro producto final.

Se elaboró un diagrama de clases Web para cada caso de uso de forma tal que se facilite la comprensión de cómo se relacionan los distintos elementos en la realización de cada uno de ellos. A continuación se muestran los diagramas:

4.2.1 Paquete Control Usuario

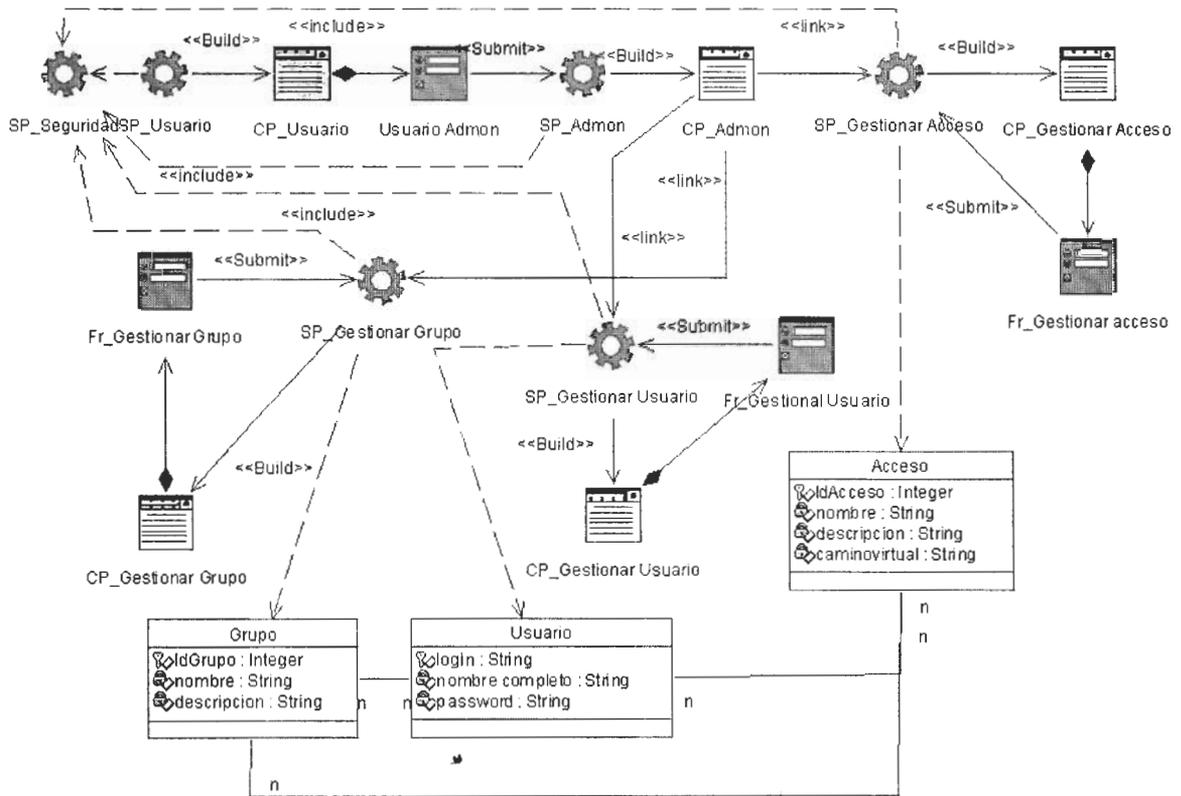


Figura. 4.1. Diagrama de Clases Web: Paquete Control Usuarios

4.2.2 Paquete Control Viajes

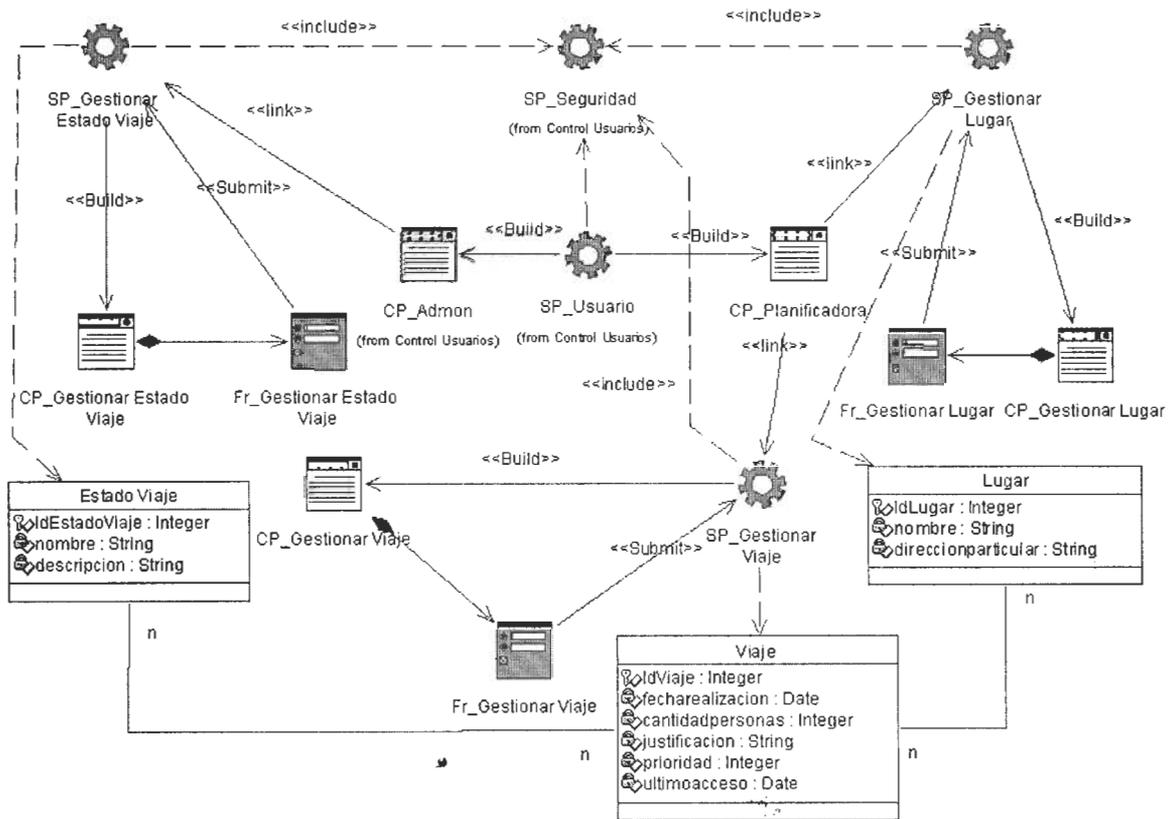


Figura. 4.2. Diagrama de Clases Web: Paquete Control Viajes

4.2.3 Paquete Control Solicitud

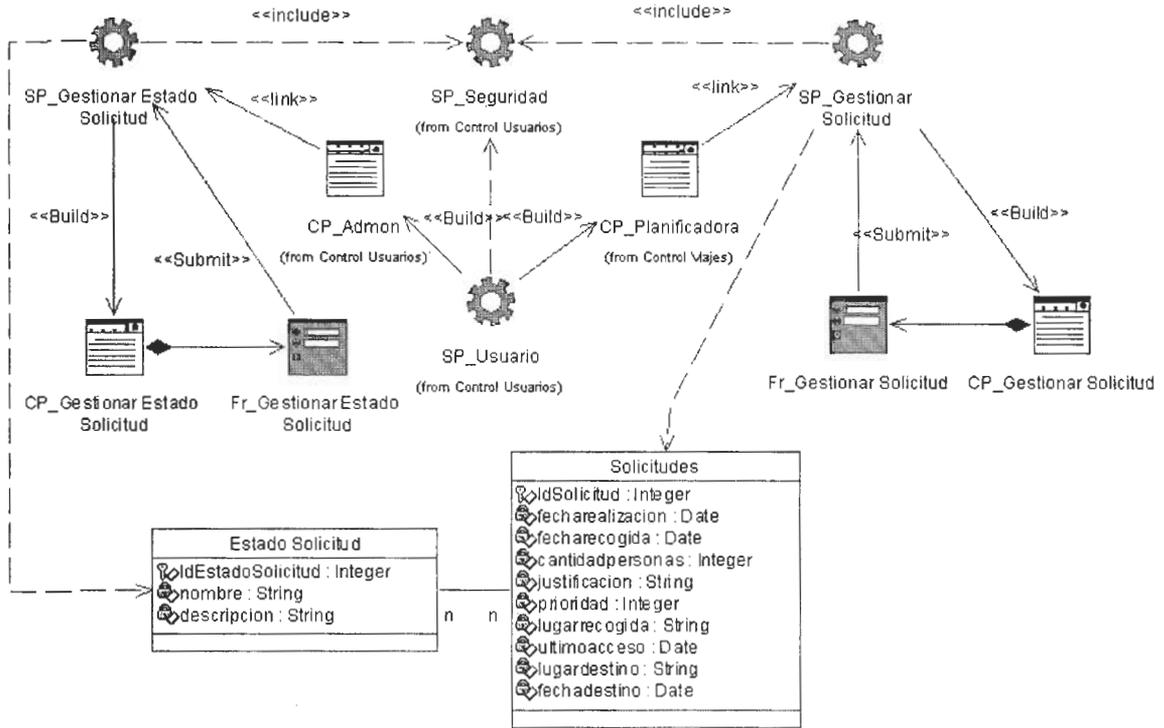


Figura 4.3. Diagrama de Clases Web: Paquete Control Solicitud

4.2.4 Paquete Control Auto

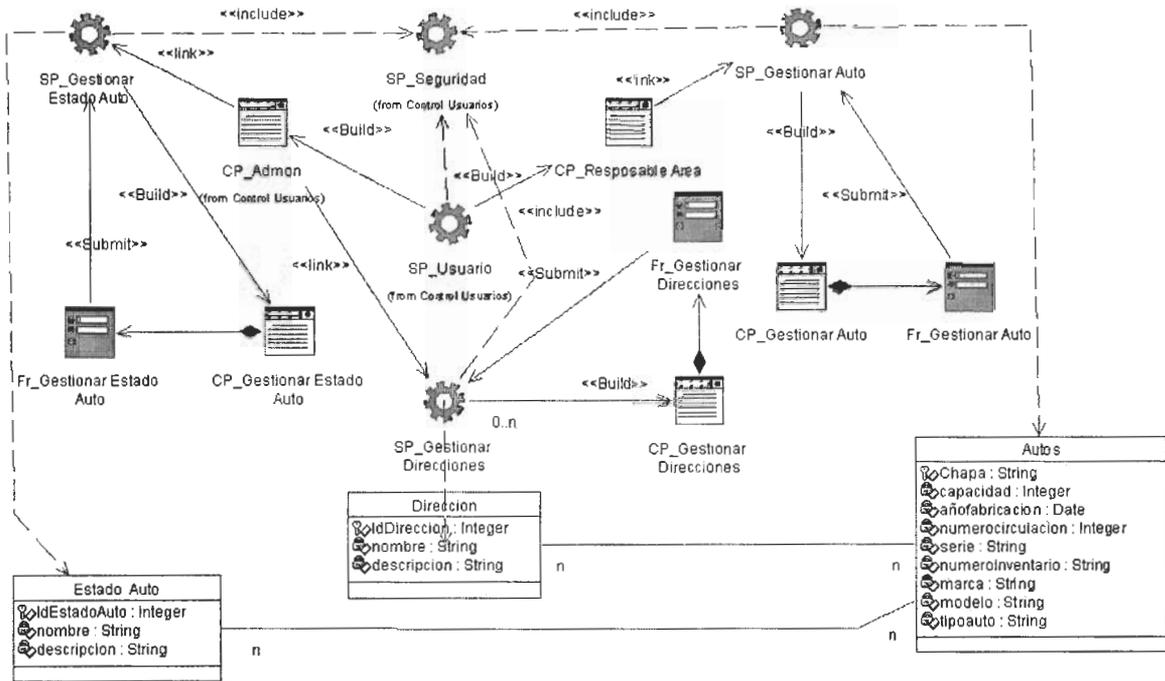


Figura 4.4. Diagrama de Clases Web: Paquete Control Aut

### 4.2.5 Paquete Control Navegación

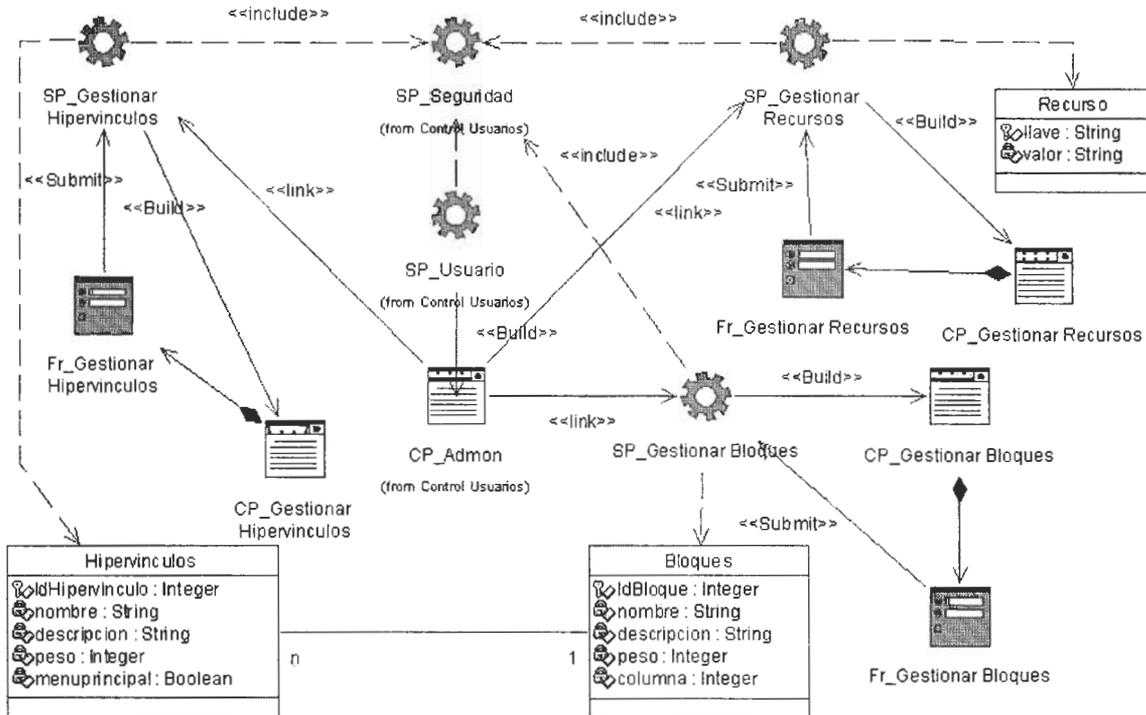


Figura 4.5. Diagrama de Clases Web: Paquete Control Navegación

## 4.3 Principios de diseño

### 4.3.1 Interfaz de usuario

Es importante aclarar que para el diseño de la interfaz del sistema analizado se tuvieron en cuenta las buenas prácticas para el diseño de una aplicación Web, teniendo en cuenta que el diseño del sitio es una herramienta fundamental para atraer a los usuarios al mismo y que el diseño de las páginas se debe realizar pensando en el cliente y no en el Web master, escogiendo cuidadosamente todos los elementos que se utilicen y empleando solamente los recursos necesarios para comunicar el mensaje que se desee, logrando que el sitio tenga una apariencia sencilla y nada recargada.

No se usaron marcos (frames) pues aunque estos mantienen el menú siempre a la vista dado que divide la pantalla en dos o más áreas, y utilizan diferentes barras para moverse en cada una de ellas; traen más problemas de los que resuelven:

## ***Capítulo 4. Descripción de la solución propuesta***

---

Resultan difíciles de indexar para los motores de búsqueda. Esta es la razón por la que muchos sitios ocupan puestos muy bajos en los buscadores solo por el uso de marcos.

- Dificultan la inclusión de una página en Favoritos.
- Una página llena de barras de desplazamiento da un aspecto muy poco estético al sitio.
- Algunos navegadores no imprimen correctamente el contenido de los frames.
- Los frames aparecen descompuestos en monitores con resolución de 640x 480.

Pensando en todo lo anteriormente mencionado, el sistema se diseñó utilizando un fondo gris pálido, con imágenes azul en un tono pálido, combinación esta que resulta muy agradable para los usuarios ya que no es irritante para la vista. Cuenta además con una imagen situado en la parte superior que da nombre a la aplicación Web. En la parte superior derecha de la imagen puede apreciarse el logotipo de la universidad.

El menú se situó en la parte izquierda de la pantalla, usando letra Verdana, tamaño 10, color azul grisáceo y con un fondo gris pálido, que es claramente legible.

### **4.3.2 Formato de salida de los reportes**

Uno de los propósitos de nuestra aplicación es ofrecerle un sistema basado en la plataforma Web con la utilización de un formato de letra claro y legible permitiéndolos contar con una aplicación que gestione el flujo de asignación de transporte y control de la flota de vehículos de las áreas de la universidad.

Los reportes implementados en el sistema son más bien los resultados obtenidos después de una petición realizada por el usuario según sus necesidades.

### **4.3.3 Ayuda**

El hecho de tratarse de una aplicación Web hace que el sistema de ayuda sea implementado usando el lenguaje HTML y las técnicas de hipertexto que este posee. Dicho sistema comprende los tópicos que pueden resultar dudosos o poco claros, así como la funcionalidad de los componentes de cada página.

Siguiendo los estándares de cualquier sistema de ayuda convencional, la información se puede alcanzar de dos formas: a través del índice general de sistema o alcanzando el tópico específico que se desea consultar utilizando para ello hipervínculos. De este modo se garantiza el acceso a la información necesaria ante cualquier problemática.

### **4.4 Tratamiento de errores**

El sistema ha sido diseñado partiendo de la premisa de evitar la ocurrencia de errores, apoyándose para esto en el empleo de estructuras que evitan el usuario teclee información que se puede codificar y mostrarse en una lista en la que solo tendrá que seleccionar el que desee en cada caso. De esta forma se evita que se introduzca datos erróneos no validos. Para cada formulario solo se muestran las opciones y los datos que son admisibles para cada entidad, evitando que el usuario, por una manipulación no adecuada, introduzca datos que no son validos para el sistema.

Entre los errores más comunes que se pueden detectar esta el hecho de que un usuario trate de ejecutar una operación a la cual no esté autorizado. En ambos casos el sistema muestra un mensaje para indicar que se cometió un error.

Los errores en la parte del cliente se validarán. En este punto se tienen en cuenta todos los posibles errores que puedan producirse, por parte del usuario, en alguna entrada de datos. Estos pueden ser la ausencia de las informaciones que sean requeridas por la aplicación, etc. En muchos casos, luego de ser validados en el cliente, pasan a un punto de chequeo en el servidor.

Si el error se produce en una página ASP, existirá la posibilidad de personalizar los mensajes de error con información mas detallada, de forma tal que el usuario posea un mayor conocimiento de lo que está ocurriendo. En dicha personalización podrá incluirse información útil para el usuario, por ejemplo, la forma de contactar al administrador a través de los servicios de administrar el sistema. Diseño de la base de datos

#### 4.4.1 Modelo lógico de datos

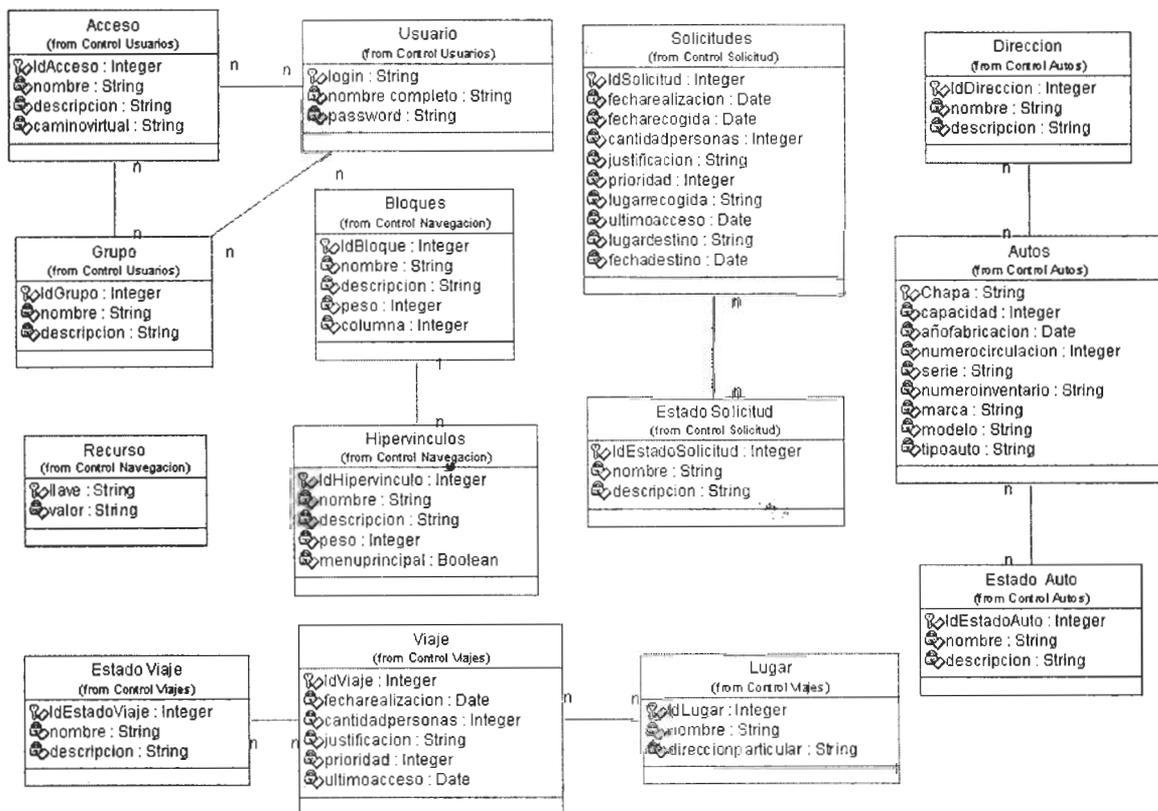


Fig. 4.6 Modelo lógico de datos

### 4.4.2 Modelo físico de datos

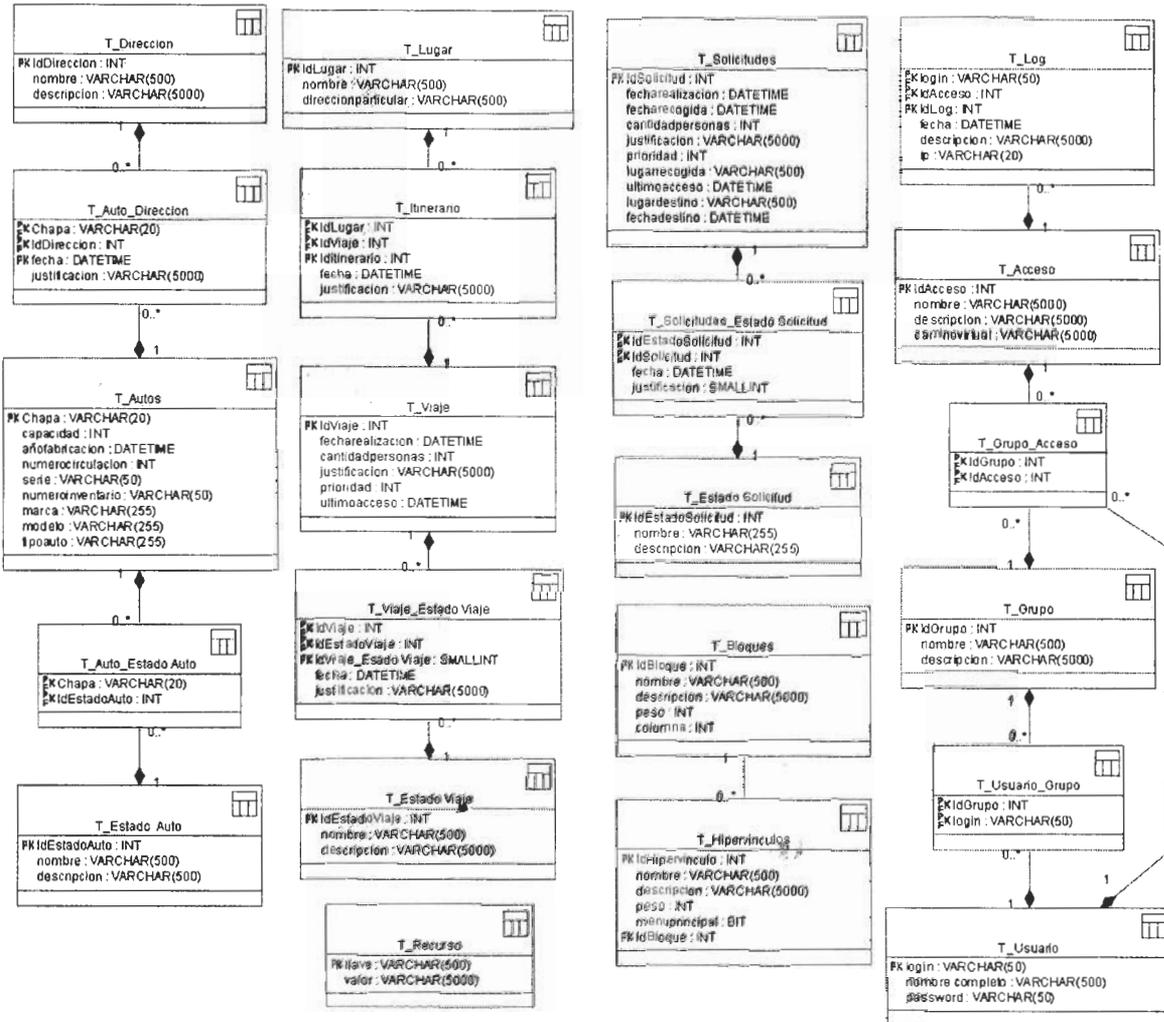


Figura 4.7. Modelo físico de datos

### 4.5 Diagrama de despliegue

La implementación de una aplicación en capas se basa en el envío de mensaje y representa una estructura modular que mejora la usabilidad, flexibilidad, interoperabilidad y la escalabilidad.

Independientemente de las capas que se implementen, la esencia de esta forma de construir aplicaciones es definir un cliente que solicita servicios y un servidor como proveedor de servicios.

Un modelo cliente-servidor simple define dos capas: capa cliente (ambiente de trabajo del usuario por lo que tiene la interfaz de la aplicación) y capa servidora (contiene la base de datos).

Este modelo de dos capas presenta limitaciones cuando el número de usuarios excede de 100. Además, cuando se implementan los servicios usando procedimientos propietarios de la base de datos (procedimientos almacenados y disparadores) se restringe la flexibilidad y la elección del Sistema de Gestión de Base de Datos (SGBD) con el que se construye la aplicación.

Estos problemas se resolvieron creando una tercera capa que implementa la lógica del negocio y que proporciona un ambiente donde miles de usuarios pueden estar conectados simultáneamente, pues el SGBD no tiene que resolver él solo la comunicación con los clientes.

De manera general las aplicaciones de n-capas ( $n > 2$ ) son fragmentadas en partes que pueden o no ejecutarse en computadoras diferentes, pero en las que pueden identificarse:

- Aplicación cliente que proporciona la interfaz con el usuario por lo que transfiere sus solicitudes.
- Servidor de aplicaciones a través del cual se actualiza y consulta la información porque tiene una relación directa con el servidor de datos.
- Servidor de datos como repositorio de la información.

Por las razones antes explicadas se utiliza en este sistema una arquitectura cliente-servidor de tres capas: la primera es la aplicación cliente, la segunda es el servidor que contiene los Script ASP encargados de realizar las actualizaciones y consultas a la base de datos y la tercera es el servidor de datos.

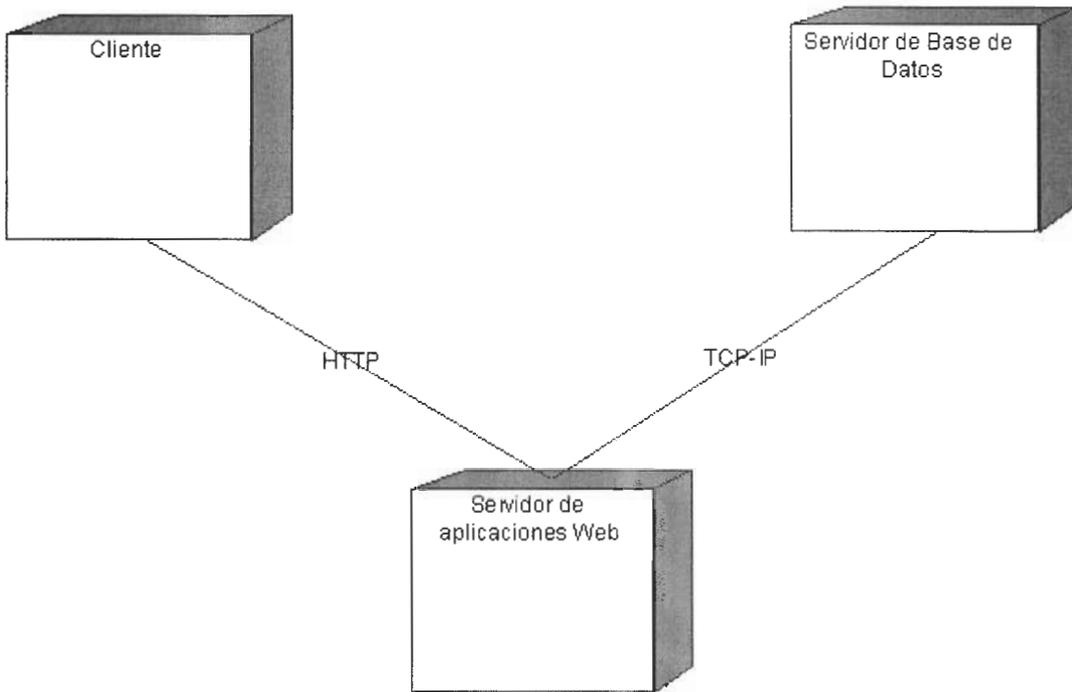


Figura 4.8. Diagrama de despliegue.

## 4.6 Conclusiones

Con la realización del diagrama de clases son especificadas las clases clientes y servidoras, con sus métodos correspondientes, necesarias para el desarrollo de la aplicación. Con el diseño de la base de datos se definen los objetos necesarios para el almacenamiento de la información.

### 5.1 Introducción

*"Para llevar a cabo un buen proyecto de desarrollo de software, debemos comprender el ámbito del trabajo a realizar, los recursos requeridos, las tareas a ejecutar, las referencias a tener en cuenta, el esfuerzo (COSTE) a emplear y la agenda a seguir."* (R. Pressman)

El estudio de factibilidad de un proyecto es un proceso muy importante a tener en cuenta antes de comenzar con la ejecución del mismo. Su importancia radica en que por medio de él podemos saber de antemano si lo que nos proponemos realizar tiene un costo de realización favorable en dependencia del esfuerzo y recursos necesarios a invertir. Además nos da un estimado del tiempo de desarrollo de la aplicación y la cantidad de personas necesarias para realizarlas, todo esto nos aporta argumentos concisos para la toma de decisiones.

### 5.2 Planificación basada en casos de uso.

La estimación mediante el análisis de Puntos de Casos de Uso es un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores, permitiendo documentar los requerimientos de un sistema en términos de Actores y Casos de Uso.

Un Actor típicamente representa a un usuario humano o a otro sistema que interactúa con el sistema bajo análisis. Un Caso de Uso representa un gránulo funcional del sistema bajo análisis, relatado como una secuencia de acciones que uno o más actores llevan a cabo en el sistema para obtener un resultado de valor significativo.

### 5.2.1 Calcular los Puntos de casos de uso desajustados (UUCP).

Se calcula a partir de la siguiente ecuación:  $UUCP = UAW + UUCW$  donde,

**UUCP:** Puntos de Casos de Uso sin ajustar

**UAW:** Factor de Peso de los Actores sin ajustar

**UUCW:** Factor de Peso de los Casos de Uso sin ajustar

#### 5.2.1.1 Calcular el factor de peso de los actores sin ajustar (UAW).

Este valor se calcula mediante un análisis de la cantidad de Actores presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Actores se establece teniendo en cuenta en primer lugar si se trata de una persona o de otro sistema, y en segundo lugar, la forma en la que el actor interactúa con el sistema.

La descripción y el tipo de los actores se relacionan en la siguiente tabla:

Tabla 5.1. Descripción y tipo de actores

Nombre del actor	Descripción	Tipo de Actor
Administrador	Persona que interactúa con el sistema a través de una interfaz gráfica.	Complejo
Planificadora	Persona que interactúa con el sistema a través de una interfaz gráfica.	Complejo
Responsable de área	Persona que interactúa con el sistema a través de una interfaz gráfica.	Complejo

Tenemos tres actores y por lo antes expuesto la relación de estos con su *factor de peso* la determinamos de la siguiente forma.

Tabla 5.2. Calculo del Factor de Peso de los Actores sin ajustar (UAW)

Descripción	Factor de peso	Actores	Total
Sistema con sistema a través de interfaz de programación.	1	0	0
Sistema con sistema mediante protocolo de interfaz basada en texto.	2	0	0
Persona que interactúa con el sistema mediante interfaz gráfica.	3	3	9
<b>UAW = <math>\Sigma</math> (Factor * Actores)</b>			<b>9</b>

**5.2.1.2 Calcular el factor de peso de los casos de uso sin ajustar (UUCW).**

Este valor se calcula mediante un análisis de la cantidad de *casos de uso* presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de *los casos de uso* se establece teniendo en cuenta la cantidad de transacciones efectuadas

Tabla 5.3. Casos de uso con *Tipo de caso de uso*

Gestionar usuarios	Medio
Gestionar grupo de usuarios	Medio
Acceso de la información	Simple
Gestionar viajes	Medio
Gestionar lugar	Medio
Gestionar estado del viaje	Medio
Gestionar Solicitud	Medio
Gestiona estado de la solicitud	Medio
Gestionar estado del auto	Medio
Gestionar autos	Medio
Historial de autos	Simple
Gestionar Recursos	Medio
Gestionar hipervínculos	Medio
Gestionar bloques	Medio
Gestionar direcciones	Medio

La relación de los Casos de Usos on su *factor de peso* la determinamos de la siguiente forma:

Tabla 5.4. Calculo del factor de peso de los casos de uso sin ajustar (UUCW)

Tipo de CU	Descripción	Peso	Cantidad de CU	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	2	10
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	13	130
Complejo	El caso de uso tiene más de 8 transacciones.	15	0	0
<b>UUCW = <math>\Sigma</math> (Factor * CantCU)</b>				<b>140</b>

Calculando los Puntos de Casos de Uso sin ajustar (UUCP):

$$\mathbf{UUCP} = \mathbf{UAW} + \mathbf{UUCW}$$

$$\mathbf{UUCP} = 9 + 140$$

$$\mathbf{UUCP} = 149$$

### 5.2.2 Calcular los puntos de casos de uso ajustados (UCP).

Una vez que se tienen los Puntos de Casos de Uso sin ajustar, se debe ajustar éste valor mediante la siguiente ecuación:

$$\mathbf{UCP} = \mathbf{UUCP} * \mathbf{TCF} * \mathbf{EF}$$

Donde:

**UCP:** Puntos de Casos de Uso ajustados

**UUCP:** Puntos de Casos de Uso sin ajustar

**TCF:** Factor de complejidad técnica

**EF:** Factor de ambiente

**5.2.2.1 Calcular el Factor de Complejidad Técnica (TCF).**

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema y cada uno de los factores se cuantifica con un valor.

$$TCF = 0.6 + 0.01 * \Sigma (\text{Peso}_i * \text{Valor}_i)$$

En la siguiente tabla se muestra el significado, el peso y el valor de cada uno de los factores:

Tabla 5.5. Sumatoria de los peso por el valor del factor de complejidad técnica

Factor	Descripción	Peso	Valor	Comentario	$\Sigma (\text{Peso}_i * \text{Valor}_i)$
T1	Sistema distribuido	2	0	El sistema es centralizado	0
T2	Objetivos de performance o tiempo de respuesta	1	2	La velocidad es limitada por las entradas provistas por el usuario	2
T3	Eficiencia del usuario final	1	3	Algunas restricciones de eficiencia	3
T4	Procesamiento interno complejo	1	1	Es sencillo	1
T5	El código debe ser reutilizable	1	1	No se requiere, aunque puede reutilizarse.	1
T6	Facilidad de instalación	0.5	1	Escasos requerimientos de facilidad de instalación	0.5
T7	Facilidad de uso	0.5	5	Es fácil de usar	2.5

T8	Portabilidad	2	0	No se requiere que el sistema sea portable	<b>0</b>
T9	Facilidad de cambio	1	3	Se requiere un costo medio de mantenimiento	<b>3</b>
T10	Concurrencia	1	0	No hay concurrencia	<b>0</b>
T11	Incluye objetivos especiales de seguridad	1	3	Seguridad normal	<b>3</b>
T12	Provee acceso directo a terceras partes	1	5	Los usuarios Web tienen acceso directo	<b>5</b>
T13	Se requieren facilidades especiales de entrenamiento a los usuarios	1	3	El sistema es fácil de usar.	<b>3</b>
<b><math>\Sigma (\text{Peso}_i * \text{Valor}_i)</math></b>					<b>24</b>

$$\text{TCF} = 0.6 + 0.01 * \Sigma (\text{Peso}_i * \text{Valor}_i)$$

$$\text{TCF} = 0.6 + 0.01 * 24$$

$$\text{TCF} = 0.84$$

### **5.2.2.2 Calcular el Factor de Ambiente (EF).**

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del Factor de ambiente. El cálculo del mismo es similar al cálculo del Factor de complejidad técnica, es decir, se trata de un conjunto de factores que se cuantifican con valores de 0 a 5.

$$\text{EF} = 1.4 - 0.03 * \Sigma (\text{Peso}_i * \text{Valor}_i)$$

Tabla 5.6. Sumatoria de los peso por el valor del factor de ambiente

Factor	Descripción	Peso	Valor	Comentario	$\Sigma (\text{Peso}_i * \text{Valor}_i)$
E1	Familiaridad con el modelo de proyecto utilizado	1.5	3	El grupo no esta totalmente familiarizado con el modelo	4.5
E2	Experiencia en la aplicación	0.5	3	El grupo no ha tenido tiempo suficiente para esta aplicación	1.5
E3	Experiencia en orientación a objetos	1	4	La mayoría del grupo programa en Objetos	4
E4	Capacidad del analista líder	0.5	2	No contamos con Especialista.	1
E5	Motivación	1	5	El grupo está altamente motivado	5
E6	Estabilidad de los requerimientos	2	2	Se esperan cambios	4
E7	Personal part-time	-1	3	El grupo tiene otras tareas.	-3
E8	Dificultad del lenguaje de programación	-1	2	Se usará lenguaje C#	-2
<b>Total</b>					<b>15</b>

$$EF = 1.4 - 0.03 * \Sigma (\text{Peso}_i * \text{Valor}_i)$$

$$EF = 1.4 - 0.03 * 15$$

$$EF = 0.95$$

Luego:

$$UCP = UUCP * TCF * EF$$

$$UCP = 149 * 0.84 * 0.95$$

$$UCP = 118.902$$

### 5.2.3 Esfuerzo (E).

Convertir los puntos de casos de uso ajustados a esfuerzo de desarrollo.

**E = UCP \* CF** Donde:

**E:** esfuerzo estimado en horas-hombre

**UCP:** Puntos de Casos de Uso ajustados

**CF:** Factor de conversión.

#### 5.2.3.1 Calcular el Factor de Conversión (CF)

Para calcular el factor de conversión:

**CF = 20 horas-hombre** (si  $Total_{EF} \leq 2$ )

**CF = 28 horas-hombre** (si  $Total_{EF} = 3$  ó  $Total_{EF} = 4$ )

**CF = abandonar o cambiar proyecto** (si  $Total_{EF} \geq 5$ )

$Total_{EF} = Cant_{EF < 3} (entre E1 - E6) + Cant_{EF > 3} (entre E7, E8)$

$Total_{EF} = 2 + 0$

$Total_{EF} = 2$

Por tanto se toma como factor de conversión (**CF**) = **20 horas-hombre** (porque  $Total_{EF} \leq 2$ )

Luego:

**E = UCP \* CF**

**E = 118.902 \* 20 horas-hombre**

**E = 2378.04 horas-hombre**

### 5.2.4 Esfuerzo Total del Proyecto (ET).

El valor de esfuerzo calculado representa el esfuerzo del FT implementación, por comparación salen el resto de los esfuerzo y la suma de ellos es el esfuerzo total

(**E<sub>T</sub>**).

Tabla 5.7. Esfuerzo Total

Actividad	% esfuerzo	Valor esfuerzo
Análisis	10%	594.51 horas-hombre
Diseño	20%	1189.02 horas-hombre
Implementación	40%	2378.04 horas-hombre
Prueba	15%	891.765 horas-hombre
Sobrecarga	15%	891.765 horas-hombre
<b>Esfuerzo total</b>	<b>100%</b>	<b>5941.1 horas-hombre</b>

Esfuerzo Total (Horas--Hombre) = **5945.1 HH**

#### Calcular Esfuerzo Total (Mes -- Hombre)

Suponiendo que una persona trabaje 8 horas por día, y un mes tiene como promedio 30 días; la cantidad de horas que puede trabajar una persona en 1 mes es 240 horas

Si **ET** (horas-hombre) = **5945.1** y por cada 240 horas yo tengo 1 mes eso daría un **ET** (mes-hombre) = **24.73**.

Como en la realización del esfuerzo participan dos personas el tiempo se reducirá a la mitad (aplicando ambos el mismo esfuerzo) luego el tiempo de desarrollo del proyecto sería de aproximadamente **12** meses, para un costo total de **\$ 4259.50** tomado un salario medio **\$ 225**.

### 5.3 Beneficios Tangibles e Intangibles

El propósito que conlleva al desarrollo del sistema propuesto es unificar y centralizar la asignación de transporte, permitiendo de este modo un rápido y eficiente acceso a la planificación.

Su principal objetivo es garantizar la transportación de IP con el control del estado de los autos de la universidad.

Por tanto, los beneficios inmediatos son mayormente intangibles:

- El acceso fácil a la información existente de los autos para la transportación.
- La organización y actualización del flujo de trabajo.
- Mejor integración y comunicación entre los trabajadores.
- Ahorro de tiempo en la búsqueda de información de los autos.
- Posibilidad de ver reportes de los viajes, autos con facilidad de búsqueda.
- Hacer más eficiente el proceso de solicitar autos con mayor control sobre los datos de la solicitud.
- Disponibilidad de los datos para estadísticas y toma de decisiones.

A pesar de que los beneficios son en su gran mayoría intangibles, con la utilización de esta aplicación Web se obtendrá considerable ahorro en gastos de papel e impresión que actualmente requiere llevar el adecuado control y reducción de costos en concepto de transportación.

Además por el carácter genérico que tiene este proyecto y su fácil implementación en cualquier plataforma puede comercializarse con cualquier centro en condiciones similares que lo requiera.

La puesta en funcionamiento de este servicio representa un aporte al desarrollo del plan de informatización de la sociedad y el empleo eficiente de las TICs. lo que contribuye con la batalla de ideas que se lleva a cabo en nuestro país y específicamente para apoyar el transporte de IP en la UCI.

Además, agrupar todos los materiales multimedia en un mismo sistema reporta notables beneficios tangibles aunque no cuantificables en cuanto a utilización del hardware debido a que se presenta una administración centralizada y por tanto una utilización del hardware centralizada. Esto posibilita una mayor capacidad de almacenamiento y una inversión mucho menor, pues existe solamente un punto de costo, con acceso de una manera más racional al hardware disponible, y no se duplican las funciones.

#### **5.4 Análisis de Costos y Beneficios.**

El desarrollo de este sistema no supone grandes gastos de recursos, ni tampoco de tiempo; la base de datos que contiene la información, puede ser alojada en los servidores existentes en la Universidad, ya que los mismos tienen buenas prestaciones y acceso rápido.

El costo para desarrollar la aplicación asciende a **\$ 4259.50**. Esto significa que se incurrirá en ese gasto fácilmente resarcible teniendo en cuenta los beneficios, para lograr una disminución considerable del tiempo empleado para controlar toda la información de los autos, chóferes y viajes, ya que el sistema permite planificar, buscar o bien consultar toda la información referente a la asignación de transporte.

En general, se puede concluir que es factible el desarrollo del sistema, ya que contribuye sin lugar a duda a un mejor y más eficiente servicio en la asignación de transporte en la infraestructura productiva.

#### **5.5 Conclusiones**

El estudio de factibilidad de un proyecto de desarrollo de software constituye un factor importante y una herramienta clave en el análisis de la estimación de los resultados del proyecto y de los valores de tiempo, costo y recursos requeridos.

El sistema propuesto se considera económicamente factible pues ya no sólo se trata de argumentos teóricos y prácticos que lo sostienen, sino de otros elementos cuantificables que subrayan el beneficio que reportaría para esta entidad y el considerable impacto social que conlleva, aportando al desarrollo del plan de informatización de la sociedad.

Una vez concluida la investigación y desarrollo de la aplicación Web, se ha llegado a la conclusión de que el objetivo fundamental de éste fue logrado. La automatización del “Sistema de Asignación de Transporte” permite garantizar un mejor registro, control, análisis y acceso a la información que se maneja en torno al proceso de la asignación de transporte para resolver los viajes de la Infraestructura productiva.

Se considera además que el presente trabajo ha cumplido con todos los objetivos propuestos, además se marcaron las pautas que permitieron:

- Identificar todas las necesidades de los clientes.
- Los procesos y especificaciones del sistema han sido descritos.
- Se alcanzó una definición de los conceptos más significativos en el dominio del problema.
- Al definirse las clases y sus relaciones en el Diagrama de clases del análisis, se obtuvo el diseño de la Base de Datos para implementar el núcleo central de la aplicación.
- La principal característica que se logró de la aplicación es su flexibilidad, a través de tablas.
- Con la implementación del primer ciclo de desarrollo quedan automatizados los principales aspectos para comenzar a trabajar con el Sistema de Asignación de Transporte.

Se puede trabajar con el Sistema de Asignación de Transporte desde varios puestos de trabajo (computadoras) concurrentemente y es muy fácil de utilizar.

Teniendo en cuenta lo anterior, este sistema se convertirá en una herramienta muy poderosa y útil para la gestión del proceso de asignación de transporte para los viajes de la Infraestructura Productiva.

Hechas las conclusiones del trabajo se recomienda:

- Para una próxima iteración considerar los cambios que puedan surgir en el negocio y que influyan en el comportamiento del sistema.
- Mantener sobre el sistema un estricto cumplimiento del proceso de mantenimiento y actualización periódica, logrando así que se mantenga la fiabilidad y funcionamiento óptimo del sistema y de la información que se gestiona a través de él.
- Continuar desarrollando la aplicación Web.
- Migrarlo a software libre.
- Elaborar el manual de usuarios del sistema.
- Implantarlo en otras empresas en un futuro.

## Referencias Bibliográficas

1. Foundation, W. *Internet* 2006 [cited; Available from: <http://es.wikipedia.org/wiki/Internet>].
2. Software, T. *Soluciones Web*. [Sitio Web] 2006 [cited; Available from: [http://www.menendezpoo.com/pags/tks/docs/it\\_soluciones\\_web.php](http://www.menendezpoo.com/pags/tks/docs/it_soluciones_web.php)].
3. Microsystems, S. *Sevicios Web*. [Sitio Web] 2005 [cited; Available from: <http://es.sun.com/soluciones/pdf/Technicalespanol3.pdf>].
4. Alvarez, M.A. *Páginas estáticas Vs. dinámicas*. 2006 [cited; Available from: <http://www.desarrolloweb.com/articulos/712.php>].
5. Mora, A. *Páginas Web Dinámicas vs. Estáticas*. 2005 [cited; Available from: <http://www.webmasters.org.mx/forosweb/index.php?showtopic=22>].
6. Panamacom. *Glosario de Informática e Internet* [cited; Available from: [http://glosario.panamacom.com/?id\\_c=19](http://glosario.panamacom.com/?id_c=19)].
7. Foundation, W. *Servidor informático*. 2006 [cited; Available from: <http://es.wikipedia.org/wiki/Servidor>].
8. Foundation, W. *Torre de servidores*. 2005 [cited; Available from: [http://es.wikipedia.org/wiki/%C3%81rea de servidores](http://es.wikipedia.org/wiki/%C3%81rea_de_servidores)].
9. Foundation, W. *Servidor web*. 2006 [cited; Available from: [http://es.wikipedia.org/wiki/Servidor\\_web](http://es.wikipedia.org/wiki/Servidor_web)].
10. Oviedo, U.d. *¿Qué es un servidor web?* 2006 [cited; Available from: <http://eo.ccu.uniovi.es/llamaquique/virtual/recursos/comun/webHTML/servidorweb/servidorweb.htm>].
11. Foundation, W. *Servidor HTTP Apache*. 2006 [cited; Available from: [http://es.wikipedia.org/wiki/Servidor\\_HTTP\\_Apache](http://es.wikipedia.org/wiki/Servidor_HTTP_Apache)].
12. Corporation, M., *Internet Information Server 5.0. Microsoft Corporation. Microsoft Windows 2000 Server Documentation*. 2000.
13. HTMLPOINT.COM. *¿Unix o NT? Depende*. 2006 [cited; Available from: <http://www.htmlpoint.com/apache/09a.htm>].
14. Andrés, M.M.M. *Sistemas de bases de datos*. 2001 [cited; Available from: <http://www3.uji.es/~mmarques/f47/apun/node4.html>].

15. Foundation, W. *PostgreSQL*. 2006 [cited; Available from: <http://es.wikipedia.org/wiki/Postgres>].
16. castellano, P.e. *MySQL*. 2005 [cited; Available from: <http://www.programacion.com/bbdd/tutoriales/idMySQL/>].
17. Suau, V.A.y.P. *MySQL vs. PostgreSQL*. 2005 [cited; Available from: <http://www.mmlabx.ua.es/mysql-postgres.html>].
18. Foundation, W. *MySQL*. 2006 [cited; Available from: <http://es.wikipedia.org/wiki/MySQL>].
19. Pérez, F.S. *Prontuario de Oracle*. [cited; Available from: [http://www.fdi.ucm.es/profesor/milanjm/bdsi0304/Manual\\_Oracle.pdf](http://www.fdi.ucm.es/profesor/milanjm/bdsi0304/Manual_Oracle.pdf)].
20. Urmachea, M.G. *SQL Server* [cited; Available from: <http://www.monografias.com/trabajos14/sqlserver/sqlserver.shtml>].
21. Informática, D.d.C.d.C.y.A. *Sistemas en Arquitectura Cliente/Servidor*. 2001 [cited; Available from: <http://sistemas.dgsca.unam.mx/publica/pdf/clienteservidor.PDF>].
22. Internet, A.E.d. *La WWW estática*. 2004 [cited; Available from: [http://www.ciberaula.com/curso/aspnet/que\\_es/](http://www.ciberaula.com/curso/aspnet/que_es/)].
23. Internet, A.E.d. *Tecnologías en el lado servidor*. 2004 [cited; Available from: [http://www.ciberaula.com/curso/aspnet/que\\_es/](http://www.ciberaula.com/curso/aspnet/que_es/)].
24. Foundation, W. *PHP*. 2006 [cited; Available from: <http://es.wikipedia.org/wiki/PHP>].
25. Foundation, W. *Active Server Pages*. 2006 [cited; Available from: [http://es.wikipedia.org/wiki/Active\\_Server\\_Pages](http://es.wikipedia.org/wiki/Active_Server_Pages)].
26. Foundation, W. *Perl*. 2006 [cited; Available from: <http://es.wikipedia.org/wiki/Perl>].
27. Foundation, W. *Java Server Pages*. 2006 [cited; Available from: [http://es.wikipedia.org/wiki/Java\\_Server\\_Pages](http://es.wikipedia.org/wiki/Java_Server_Pages)].
28. Internet, A.E.d. *Programación web con ASP.NET 2004* [cited; Available from: [http://www.ciberaula.com/curso/aspnet/que\\_es/](http://www.ciberaula.com/curso/aspnet/que_es/)].
29. Internet, A.E.d. *Tecnologías en el lado cliente*. 2004 [cited; Available from: [http://www.ciberaula.com/curso/aspnet/que\\_es/](http://www.ciberaula.com/curso/aspnet/que_es/)].

30. Ferrer, J. *Html Dinámico*. 1999 [cited; Available from: <http://leo.worldonline.es/joferrer/tweb/tutoriales/htmldinamico/htmldinamico.htm>].
31. Rodríguez, G.M. *Diferencias entre Java y JavaScript*. 2003 [cited; Available from: <http://www.gamarod.com.ar/articulos/32.asp>].
32. Quijado, J. *VBScript*. 2005 [cited; Available from: <http://www.programacion.com/asp/tutorial/vbscript/1/>].
33. Corporation, M. *ECMA C# and Common Language Infrastructure Standards*. 2006 [cited; Available from: <http://www.msdn.microsoft.com/net/ecma>].
34. Foundation, W. *.NET*. 2006 [cited; Available from: <http://es.wikipedia.org/wiki/.NET>].
35. *Documentación de .Net Framework SDK. Información General acerca de .Net Framework*. 2002.
36. Gracia, J. *Introducción al .NET Framework*. 2004 [cited; Available from: <http://www.webestilo.com/aspnet/aspnet00.phtml>].
37. Lockhart, T. *Tutorial de PostgreSQL*. [cited; Available from: <http://es.tldp.org/Postgresql-es/web/navegable/tutorial/tutorial.html>].
38. Corporation, M. *Preguntas más frecuentes*. 2003 [cited; Available from: <http://www.microsoft.com/spanish/msdn/vcsharp/productinfo/qa.asp>].
39. Foundation, W. *RUP*. 2006 [cited; Available from: <http://es.wikipedia.org/wiki/RUP>].
40. Foundation, W. *Lenguaje Unificado de Modelado*. 2006 [cited; Available from: <http://es.wikipedia.org/wiki/UML>].

**Código abierto:** Del inglés open source, es el término por el que se conoce al software distribuido y desarrollado en forma libre.

**Cookie:** Una cookie (en castellano, galleta) es un fragmento de información que se almacena en el disco duro del visitante de una página web a través de su navegador, a petición del servidor de la página. Esta información puede ser luego recuperada por el servidor en posteriores visitas.

**CSS:** Son grupos de estilo, que a su vez son grupos de propiedades, que definen la apariencia de un elemento en una página Web. Permiten definir los colores, el tamaño y la posición de cada elemento.

**DOM:** es una infraestructura construida que ve cada página Web como una colección independiente de objetos -imágenes...- cada uno de los cuales tiene sus propias propiedades.

**FTP:** Es uno de los diversos protocolos de la red Internet, concretamente significa File Transfer Protocol (Protocolo de Transferencia de Ficheros) y es el ideal para transferir grandes bloques de datos por la red.

**Gestión empresarial:** Consiste en la buena utilización de los recursos de una empresa en aras de obtener una mejor calidad.

**GUI:** Interfaz gráfica de usuario, es un método para facilitar la interacción del usuario con el ordenador a través de la utilización de un conjunto de imágenes y objetos pictóricos (iconos, ventanas..) además de texto.

**HTML:** Hypertext Markup Language (lenguaje de etiquetado de documentos hipertextual), es un lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

**HTTP:** HyperText Transfer Protocol (protocolo de transferencia de hipertexto), es el protocolo usado en cada transacción de la Web (WWW).

**Internet:** Es una red de redes a escala mundial de millones de computadoras interconectadas con un conjunto de protocolos, el más destacado, el TCP/IP.

**Lenguajes Script:** son versiones reducidas de otros lenguajes. Estas versiones se usan para su integración en páginas Web. Un código escrito en un lenguaje de script se incorpora directamente dentro de un código HTML y se ejecuta interpretado, no compilado. Son los que ayudan a modificar las propiedades de los diferentes objetos. Entiéndase JavaScript, VBScript u otro lenguaje. Es importante resaltar que DHTML es independiente del lenguaje.

**Metadatos:** Meta+datos, es un término que se refiere a datos sobre los propios datos. Un ejemplo es un folleto que nos informa sobre el lugar y el tipo de un libro. Nos está dando datos sobre otros datos: el libro al que se refiere el folleto.

**PDA:** Del inglés Personal Digital Assistant, (Ayudante personal digital) es un computador de mano originalmente diseñado como agenda electrónica. Hoy en día se puede usar como un ordenador doméstico (ver películas, crear documentos, navegar por Internet).

**Protocolo:** Se les llama protocolo de red o protocolo de comunicación al conjunto de reglas que controlan la secuencia de mensajes que ocurren durante una comunicación entre entidades que forman una red.

**Script:** Tipo de programa que consiste en una serie de instrucciones que serán utilizadas por otra aplicación.

**SDK:** Kit de desarrollo de software, un conjunto de aplicaciones para desarrollar programas en un determinado lenguaje o para un determinado entorno (Software Development Kit).

**Sistema operativo:** Es un conjunto de programas destinados a permitir la comunicación del usuario con un ordenador y gestionar sus recursos de manera eficiente.

**Sitio web:** En inglés: website, es una conjunto de páginas web, típicamente comunes a un dominio de Internet o subdominio en la World Wide Web en Internet.

**WWW:** La World Wide Web (del inglés, Telaraña Mundial), la Web o WWW, es un sistema arquitectura cliente/servidor creada por el CERN y permite la distribución y obtención de información en Internet basado en hipertexto e hipermedia.

**XML:** Lenguaje de descripción de páginas de Internet (eXtensible Markup Language), diseñado con la intención de reemplazar al estándar actual HTML.