

004.21  
Leo  
S  
TD\_0088\_05-02

TD-0088-05-02



**Universidad de Ciencias Informáticas**  
**Departamento de Informatización**

# **SISTEMA INTEGRAL DE RESERVACIONES**

**Documento de tesis para optar por el título de Ingeniero Informático**

**Autor: Nazario León Pérez**

**Tutor: Ing. Ariadna Falcón López**

## **Resumen**

El proyecto que aquí se desarrolla consiste en la creación de un sistema encargado de automatizar el proceso de reservación de recursos varios en la Universidad de las Ciencias Informáticas (UCI).

Existían en la universidad previos a la creación del Sistema Integral de Reservaciones algunos sistemas que resolvían en alguna medida la gestión de la reservación de los pases y los tiempos de máquina pero estos no contaban con el grado de integración que requieren los planes de informatización de la universidad.

El trabajo surge a partir de la necesidad de lograr la integración de todos los sistemas con el fin de aprovechar la infraestructura de redes y equipamiento informático, dando solución a un problema de primer orden, garantizar el acceso a los recursos, actividades y servicios a todo el personal y estudiantado de una forma eficiente y organizada a través de los medios antes mencionados.

El objetivo de este trabajo es realizar el análisis, diseño e implementación de un sistema informático que permita realizar reservaciones de recursos de diversa índole, que provea con la información necesaria a los demás sistemas que así lo requieran y que esté al alcance de todo el personal de la UCI en todo momento.

Este sistema, implementado en tecnologías de punta como .Net y los Servicios Web, permitirá el acceso de forma cómoda, segura y eficiente a reservar cualquier recurso de los actualmente estipulados y los que puedan surgir en el transcurso del desarrollo de la institución. Además se integra a los sistemas que necesiten nutrirse de la información que éste maneja.

<b>Introducción</b> .....	8
<b>CAPITULO 1. Fundamentación teórica</b> .....	11
Introducción .....	11
1.1 Actualidad .....	11
1.1.1 Servicios que oferta una agencia de reservaciones.....	13
1.2 Marco de trabajo y herramientas de apoyo para la elaboración.....	14
1.2.1 RUP: .....	14
1.2.2 UML: .....	17
1.2.3 Rational Rose: .....	18
1.2.4 .Net: .....	19
1.2.5 Servicios Web: .....	20
1.2.5 Asp.net: .....	21
1.2.6 SQL Server:.....	22
Conclusiones.....	23
<b>CAPITULO 2. Análisis del Sistema</b> .....	24
Introducción.....	24
2.1 Situación problemática .....	24
2.2 Objeto de automatización.....	25
2.2.1 Reservación de Pase:.....	25
2.2.2 Reservación de Pase Masivo:.....	26
2.2.3 Reservación de Viajes Nacionales: .....	27
2.2.4 Reservación de Locales: .....	27
2.2.5 Reservación de Tiempo de Máquinas: .....	28
2.2.6 Reservación de Excursiones:.....	29
2.3 Propuesta de sistema .....	30
2.3.1 Objetivos Generales. ....	30
2.3.2 Objetivos Específicos. ....	30
2.3.3 Sistema propuesto (Software).....	30
2.4 Modelo de negocio .....	33
2.4.1 Modelo del negocio .....	33
2.4.2 Especificación de los requisitos de software .....	33

2.4.3 Requerimientos Funcionales.....	33
2.4.4 Requerimientos no Funcionales. ....	35
2.4.5 Definición de los casos de uso del sistema. ....	36
Conclusiones.....	37
<b>CAPITULO 3. Análisis y diseño del sistema .....</b>	<b>38</b>
Introducción.....	38
3.1 Diagrama de clases del análisis.....	39
3.2 Diagramas de interacción.....	40
3.3 Diseño de las clases .....	45
3.4.1 Capa de datos.....	46
3.4.2 Capa de negocio .....	47
3.5 Diagramas de diseño Web.....	48
3.7 Modelo de datos .....	62
3.8 Tratamiento de errores .....	67
3.9 Concepción de la ayuda .....	68
Conclusiones.....	68
<b>Conclusiones generales.....</b>	<b>70</b>
<b>Recomendaciones .....</b>	<b>71</b>
<b>Bibliografía Consultada.....</b>	<b>72</b>
<b>Glosario de términos.....</b>	<b>73</b>
<b>Anexo 1. Modelo del negocio.....</b>	<b>79</b>
Actores que participan en el proceso del negocio .....	79
Diagramas de casos de uso del negocio .....	79
Descripción de los casos de uso .....	82
Diagramas de actividades del negocio .....	84
<b>Anexo 2. Definición de los casos de uso del sistema.....</b>	<b>87</b>
Descripción de los actores.....	87
Descripción de los casos de uso del sistema.....	87
Diagrama de casos de uso del sistema.....	92
Modelo del dominio .....	93
Casos de uso expandidos.....	94

### **Introducción**

Dada la creciente demanda de recursos y servicios que surge como parte del proceso de construcción de la UCI, la formación del estudiantado y del funcionamiento en general de la entidad se hace necesario el acceso a estos de forma práctica y organizada. Pero no siempre están en disponibilidad en todo momento y para todo el personal. Entonces surge la necesidad de realizar una planificación personal del acceso a estos elementos en función de las necesidades propias, el momento deseado para tal y la disponibilidad; para lograr esto de una forma ordenada y segura se implementan mecanismos de reservación.

La Universidad de Ciencias Informáticas UCI cuenta con una infraestructura de equipamiento informático y redes que le permite un acceso a toda la creciente población de estudiantes y trabajadores a este tipo de tecnología. Debido a esto se hace sumamente factible la creación de sistemas informáticos para garantizar el acceso a los elementos a reservar de una forma cómoda, rápida y segura. En la actualidad se ha informatizado la reservación de algunos recursos con la creación de pequeños sistemas que prestan este servicio pero la mayoría de las reservaciones se continúa realizando "a mano", además los sistemas implementados y en fase de implementación para este fin son sistemas independientes entre si que necesitan ser accedidos por separado para cada recurso a reservar.

El proceso de reservación de recursos y servicios es una de las actividades mas importantes que se realizan en la UCI por lo que surgió la necesidad de integrar todo el proceso de reservación en un único sistema informático que permita realizar cualquier tipo de reservación de los recursos y servicios preestablecidos.

Se han creado durante la etapa inicial de informatización de la UCI algunos sistemas que de alguna medida han automatizado el proceso de reservación, algunos en funcionamiento y otros en fase de implementación o diseño, pero no han logrado el nivel de integración que se requiere en un centro con las características de la UCI.

## INTRODUCCION

---

El sistema que aquí se desarrolla está orientado a satisfacer las necesidades de reserva de recursos que actualmente se realiza de forma manual o automatizada pero además está pensado para lograr esto de una forma única e integrada nutriéndose de información de los sistemas adyacentes que gestionan los recursos dados y proveyendo de información a los sistemas que así lo requieran. El sistema será capaz de brindar la información del movimiento y estado de todo un cúmulo de recursos, servicios y personal, básico esto para el correcto funcionamiento de la entidad. Dado esto, es posible definir un objetivo general que sería:

Realizar el análisis, diseño e implementación de un sistema informático que automatice el proceso de reservación de recursos, actividades y servicios.

Este sistema debe gestionar, como punto de partida y serán objetivos específicos a lograr, los servicios siguientes:

- Reservación de pases, cortos y de fin de semana.
- Las actividades recreativas como excursiones y viajes.
- La reserva de locales.
- La reserva de tiempos de máquinas.

Pero permitirá además la adición de cualquier otro recurso o servicio (en lo adelante todos los elementos reservables serán llamados recursos o recursos reservables) que sea de interés poner a consideración para ser reservado. La gestión de la heterogeneidad de los recursos reservables es el objeto fundamental en el que basa este análisis y para el cual se trabajó en función de su solución.

El sistema está destinado a ser usado por la mayoría de los estudiantes, profesores y trabajadores de la Universidad y estará además al servicio de otras aplicaciones que se nutrirán de la información que este maneja, para la toma de decisiones y la planificación en general.

En este trabajo se realizó el estudio del negocio actual de gestión de reservas en la forma que se maneja y se hizo un análisis del proceso para lograr determinar un diseño

## INTRODUCCION

---

que satisfaga las expectativas que se tienen de un sistema tan integrador e innovador. Se profundizó en el tema del manejo del cúmulo de recursos reservables posibles, tratando de prever todas sus particularidades y características que se pudieran presentar, para no perder así la riqueza de información que pudiera extraerse de estos; además se trabajó en lograr un sistema lo suficientemente escalable capaz de adaptarse al número variable de requerimientos que se espera de una institución en crecimiento como la UCI.

Este trabajo está dividido en tres capítulos que contienen toda la información referente a la investigación realizada y al análisis y diseño del sistema a desarrollar, la información está distribuida de la siguiente manera: Capítulo I, Fundamentación Teórica; Capítulo II, Características del Sistema, en este capítulo se realiza un estudio preliminar del sistema y el proceso de negocio, en el cual se especifican los objetivos y los requerimientos esperados así como las tecnologías elegidas para el fin; Capítulo III, Análisis y Diseño del Sistema: Se definen las expansiones de los casos de usos, se define el modelo del dominio tratado, se desarrollan la secuencia del proceso y el diseño como tal de la arquitectura del sistema.

## **CAPITULO 1. Fundamentación teórica**

### **Introducción**

En este capítulo se aborda el estado actual de desarrollo de las aplicaciones de gestión de reservaciones en el ámbito mundial, nacional y en la Universidad, además de un estudio de las tecnologías en las que se apoya el desarrollo del sistema en función de un análisis de las tendencias actuales.

### **1.1 Actualidad**

“Actualmente se vive en una economía acelerada y que demanda respuestas rápidas a cualquier estímulo de cambio. Ahora no es sólo importante manejar productos y servicios de calidad y bajo ciertos estándares, sino responder a las demandas del mercado más rápido y a menor costo que el resto de los competidores. No sólo es una economía más exigente sino que además está sufriendo una metamorfosis al pasar de un mercado basado en bienes y servicios tangibles a uno donde predominan los intangibles, donde la información y el conocimiento son los activos más estratégicos.”  
Urdiales Salazar (2000)

Respecto a los sistemas informáticos de reservaciones se ha hecho mucho en el mundo, fundamentalmente en el ámbito de Internet como se puede constatar en estos datos:

<b>DATOS GENERALES</b>
24 millones de americanos hacen reservaciones On-line (duplicándose cada año)
E-commerce representará el 9% de las ventas globales para el 2004
99.8% de los viajeros internacionales de negocios tienen acceso al Internet y al e-mail
El 63.5% pasan de 30 minutos a dos horas en el Internet por día.
El 91% compraron algo en el INTERNET el año anterior.











## CAPITULO I “FUNDAMENTACION TEORICA”

### GLOBAL DATA SYNCHRONIZATION(GDS) Y ELECTRONIC DISTRIBUTION SYSTEM(EDS)

Ofrecen el hardware, el software, y las líneas de cómputo por las cuales las agencias de viajes, líneas aéreas, redes de suscripción On-line y otros pueden acceder el sistema central de reservaciones.

Algunas de las compañías y grupos que brindan servicios On-line de reservaciones.

COMPAÑÍA AMADEUS	
54.000 Agencias de viajes	
8,400 oficinas de venta de líneas aéreas en 199 mercados	
Reservaciones a 500 compañías aéreas.	
Acceso a más de 54,000 hoteles	
46 empresas de alquiler de autos	

En Cuba, a partir de la explosión del desarrollo del polo turístico y la introducción de las nuevas tecnologías se ha trabajado mucho en función de la informatización del proceso de acceder a los recursos y servicios de esta rama de nuestra economía. Debido al

## **CAPITULO I “FUNDAMENTACION TEORICA”**

---

bloqueo económico nos es difícil llevar una labor de marketing que se corresponda a las necesidades de la competencia actual en el sector turístico más las dificultades que se presentan al cliente potencial a la hora de intentar acceder personalmente a nuestras ofertas. Internet, y en particular la Web, se presta para aliviar la situación con la introducción de las nuevas posibilidades de los sistemas informáticos de reservaciones, accesibles a todo el mundo y fácilmente soportados por cualquier sitio Web. La mayoría de nuestros hoteles y establecimientos de servicios del sector turístico cuentan con sitios Web publicados en Internet y cuentan con el servicio de reservaciones on-line.

A pesar del gran desarrollo que ha alcanzado el mundo de las reservaciones empleando sistemas informáticos e Internet no se encontraron precedentes de la existencia de alguna agencia de reservaciones que fuera capaz de integrar en si misma el proceso de reservación de recursos variados sino que la tendencia ha sido siempre la de ofrecer acceso a los sistemas especializados en un recurso en particular a reservar.

### **1.1.1 Servicios que oferta una agencia de reservaciones.**

Información, orientación y asistencia al público y a sus clientes:

Las agencias de reservaciones en el ejercicio de sus actividades ofrecen un servicio de información permanente, sin costo alguno (gratuito) al público en general sobre todo lo relacionado a los recursos a reservar. Igualmente realiza y/o presta un servicio de gestión a sus usuarios como puede ser la canalización de visado, cuando tiene un grupo o un paquete para determinado país por lo cual no percibe ningún tipo de remuneración.

Ventas de Servicios Suelto:

Las agencias se encargan de la mediación en la venta de boletos y reserva de plaza en toda clase de medios de transporte, las reservas de habitaciones y servicios en los

## **CAPITULO I “FUNDAMENTACION TEORICA”**

---

establecimientos hoteleros y demás alojamientos turísticos, alquiler de automóvil, la venta de seguro de viajes o de asistencia turística etc.

Por las ventas de estos servicios las agencias de viajes perciben una comisión de antemano acordada con el proveedor de los mismos la cual varía según el tipo de servicios que se ofrezca y el volumen que genere.

Venta de Servicios Organizados:

En esta concepción están comprendidos todos los servicios que se comercializan, que se venden en forma de paquetes los cuales el cliente puede adquirir en un solo acto de compra. (Transporte, Alojamiento, Alimentación, Traslado, Etc.), evitándose así todo el complicado proceso de contratación de dichos servicios por separados. Los servicios organizados que vende una agencia de viajes, pueden ser preparados por la misma agencia, a petición de un cliente o en forma de programas regulares preparados por los mayoristas de los tours operadores.

### **1.2 Marco de trabajo y herramientas de apoyo para la elaboración**

En el análisis y diseño de este trabajo se emplea una metodología RUP usando como herramienta el Rational Rose y el lenguaje UML. El sistema se programará empleando Asp.net y los Servicios Web de la tecnología .Net y se usará el Visual Estudio .Net como herramienta de desarrollo. La base de datos se implementará según el modelo relacional sobre SQL Server. Esta decisión está determinada, fundamentalmente por el grado de integración, compatibilidad y productividad que estas herramientas y tecnologías manifiestan.

#### **1.2.1 RUP:**

Evolución:

1967	Método Ericsson	Desarrollo Basado en Componentes (I. Jacobson)
------	-----------------	---

## CAPITULO I “FUNDAMENTACION TEORICA”

---

1987	Proceso Objectory	Incorpora “Casos de Uso” y diseño OO.
1995-1997	P. Objectory de Rational (ROP)	Añade fases: Inicio, Elaboración, Construcción y Transición. Utiliza UML.
1998	P.Unificado de Rational 5.0	Método Unificado de I. Jacobson, G. Booch, J. Rumbaugh.
2000	P.Unificado de Rational 2000	Incorpora Método OO en Tiempo Real

Descripción:

El proceso unificado es un conjunto de actividades que transforman requisitos de usuario en un sistema software.

Consta de:

a) Flujos de Trabajo Fundamentales:

- Modelado del Negocio
- Requisitos
- Análisis y Diseño
- Implementación
- Pruebas
- Despliegue
- Configuración y Gestión de Cambios, Gestión del Proyecto, Entorno.

b) Fases: Inicio, Elaboración, Construcción, Transición

- Una Fase tiene 1 o varias Iteraciones y finaliza con un Hito.
- Las 4 Fases producen una Generación.

RUP es:

- Iterativo e Incremental
- Dirigido por los Casos de Uso
- Centrado en la Arquitectura

## CAPITULO I “FUNDAMENTACION TEORICA”

---

### a) Iterativo e Incremental:

- Compuesto por pequeños proyectos que incorporan incrementalmente nueva funcionalidad y cuyo desarrollo es una iteración”
  - Inicio: Comprender el Problema y Determinar su Ámbito
  - Elaboración: Capturar Requisitos y Producir un Prototipo (Línea Base)
  - Construcción: Diseño e Implementación. 1º Producto Operacional (Beta)
  - Transición: Producir y Suministrar el Producto Final.
- Características de un Proyecto Iterativo e Incremental
  - Obtiene un Sistema Robusto
  - Reduce el Riesgo de tener un mal producto
  - Reduce el Riesgo de no obtener el producto en el tiempo previsto
  - Permite atacar problemas con requisitos incompletos.

### b) Dirigido por los Casos de Uso:

- Casos de Uso:
  - Servicios que un actor requiere del sistema y le proporcionan un resultado
  - Proporcionan los Requisitos Funcionales del Sistema
- Modelo de Casos de Uso:
  - Describen toda la funcionalidad del Sistema
- Filosofía de Desarrollo:
  - Modelo Casos de Uso → Modelo de Análisis → Modelo de Diseño → Modelo Implementación.
- Trazabilidad del Sistema:

## CAPITULO I “FUNDAMENTACION TEORICA”

---

- Cambios en Requisitos de un Caso de Uso fácil detectar las clases y componentes que afectan.

### c) Centrado en la Arquitectura

- Casos de Uso describen la Funcionalidad del Sistema
- Arquitectura define la Forma del Sistema
- Se describe mediante Vistas que incorporan el 5-10% de los casos de uso más relevantes:
  - Casos de Uso: Actores y Casos de Uso relevantes.
  - Análisis: Clases de Análisis (Interfaz, Control, Entidad) de C. Uso anteriores.
  - Diseño: Subsistemas y Clases de Diseño derivadas de las Clases de Análisis.
  - Despliegue: Arquitectura Física por medio de Nodos
  - Implementación: Componentes de las clases relevantes.

### 1.2.2 UML:

El Lenguaje de Modelado Unificado UML es una especificación de notación orientada a objetos. Es un lenguaje de propósito general para el modelado orientado a objetos. UML es también un lenguaje de modelación visual que permite una abstracción del sistema y sus componentes, que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir.

UML no es un lenguaje de programación. Las herramientas pueden ofrecer generadores de código de UML para una gran variedad de lenguaje de programación, así como construir modelos por ingeniería inversa a partir de programas existentes.

UML también intenta solucionar el problema de propiedad de código que se da con los desarrolladores, al implementar un lenguaje de modelado común para todos los

## CAPITULO I “FUNDAMENTACION TEORICA”

---

desarrollos se crea una documentación también común, que cualquier desarrollador con conocimientos de UML será capaz de entender, independientemente del lenguaje utilizado para el desarrollo.

Permite la modificación de todos sus miembros mediante estereotipos y restricciones. Un estereotipo nos permite indicar especificaciones del lenguaje al que se refiere el diagrama de UML. Una restricción identifica un comportamiento forzado de una clase o relación, es decir mediante la restricción estamos forzando el comportamiento que debe tener el objeto al que se le aplica.

UML es ahora un Standard, no existe otra especificación de diseño orientado a objetos. Su utilización es independiente del lenguaje de programación y de las características de los proyectos, ya que UML ha sido diseñado para modelar cualquier tipo de proyectos, tanto informáticos como de arquitectura, o de cualquier otro ramo.

### 1.2.3 Rational Rose:

*Rational Rose* es la herramienta CASE que comercializan los desarrolladores de UML y que soporta de forma completa la especificación del UML.

Esta herramienta propone la utilización de cuatro tipos de modelos para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software.

*Rational Rose* utiliza un proceso de desarrollo iterativo controlado (*controlled iterative process development*), donde el desarrollo se lleva a cabo en una secuencia de iteraciones. Cada iteración comienza con una primera aproximación del análisis, diseño e implementación para identificar los riesgos del diseño, los cuales se utilizan para conducir la iteración, primero se identifican los riesgos y después se prueba la aplicación para que éstos se hagan mínimos.

## **CAPITULO I “FUNDAMENTACION TEORICA”**

---

Cuando la implementación pasa todas las pruebas que se determinan en el proceso, ésta se revisa y se añaden los elementos modificados al modelo de análisis y diseño. Una vez que la actualización del modelo se ha modificado, se realiza la siguiente iteración.

Rose permite que haya varias personas trabajando a la vez en el proceso iterativo controlado, para ello posibilita que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo.

También es posible descomponer el modelo en unidades controladas e integrarlas con un sistema para realizar el control de proyectos que permite mantener la integridad de dichas unidades.

Se puede generar código en distintos lenguajes de programación a partir de un diseño en UML.

Rational Rose proporciona mecanismos para realizar la denominada Ingeniería Inversa, es decir, a partir del código de un programa, se puede obtener información sobre su diseño.

Rational también ofrece una alternativa a la plataforma .NET que permite crear aplicaciones de nuevas generaciones fiables y productivamente.

### **1.2.4 .Net:**

Utiliza los Servicios Web como un medio para poder interoperar a distintas tecnologías. Permite conectar distintos sistemas operativos, dispositivos físicos, información y usuarios. Les da a los desarrolladores las herramientas y tecnologías para hacer rápidamente soluciones de negocios que involucran distintas aplicaciones, dispositivos físicos y organizaciones.

La idea central detrás de la plataforma .NET es la de servicio. Más concretamente software como servicio y de cómo construir, instalar, consumir, integrar o agregar estos



## **CAPITULO I “FUNDAMENTACION TEORICA”**

---

servicios para que puedan ser accedidos mediante Internet. Esto es posible debido a que tenemos la infraestructura de comunicación global que es Internet cada vez mas rápida y a un costo cada vez menor y además, a la capacidad de los procesadores que continúa incrementándose año tras año. El usuario de Internet puede con un explorador de Internet no solamente acceder a contenido como texto, imágenes o sonido, también puede hacer uso de servicios Web. Estos son los bloques de construcción o componentes sobre los cuales se basa el modelo de computación distribuida en Internet. La plataforma .NET permite usar Internet y su capacidad de distribución para que los usuarios accedan desde cualquier dispositivo, en cualquier sistema operativo y lugar a la funcionalidad que los servicios Web proveen.

Tanto la invocación de los servicios como su ejecución pueden ser hechas en cualquier dispositivo y sistema operativo, y accedido desde Internet. Los sitios se comunican entre sí y acceden a servicios y contenidos sin la intervención humana. Por eso se llama a la nueva generación de Internet "Internet inteligente".

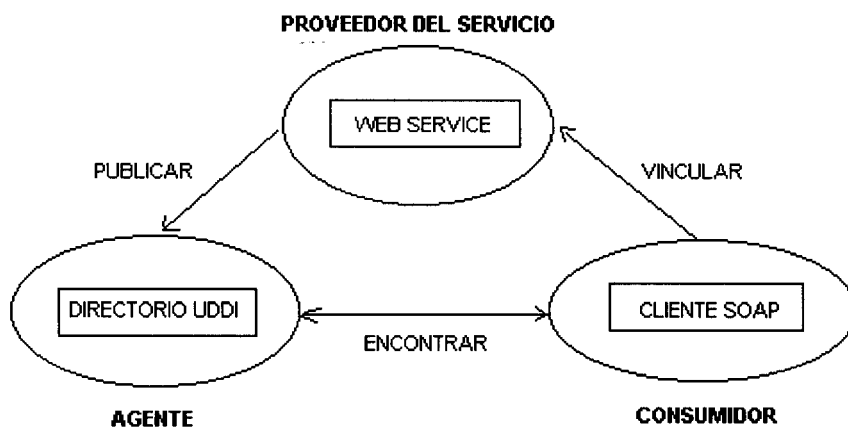
### **1.2.5 Servicios Web:**

Los clientes actuales que existen en la Web, ya sean usuarios físicos o aplicaciones de terceros, son consumidores de servicios. Conseguir que la red, compuesta principalmente de páginas HTML y archivos, se convierta en una red de servicios es uno de los objetivos de la plataforma .Net. Un servicio Web es como su nombre lo indica, un servicio que una determinada empresa pone a disposición de los usuarios. Dependiendo de las necesidades, puede tratarse de un servicio interno o accesible para el público de Internet en general. Para crear un servicio es posible utilizar cualquiera de los lenguajes disponibles en la plataforma .Net, mientras que los consumidores pueden potencialmente estar en cualquier lugar y ejecutándose sobre cualquier hardware y sistema operativo. Realmente cualquiera puede ofrecer o consumir un Servicio Web si utiliza el protocolo adecuado, no es necesario disponer de Windows o la plataforma .Net.

## CAPITULO I “FUNDAMENTACION TEORICA”

---

Los Servicios Web son componentes software que permiten a los usuarios usar aplicaciones de negocio que comparten datos con otros programas modulares, vía Internet. Son aplicaciones independientes de la plataforma que pueden ser fácilmente publicadas, localizadas e invocadas mediante protocolos Web estándar, como XML, SOAP, UDDI o WSDL. El objetivo final es la creación de un directorio de online de Web services, que pueda ser localizado de un modo sencillo y que tenga una alta fiabilidad.



### 1.2.5 Asp.net:

Cada vez es mayor el número de empresas que optan por usar la arquitectura de Internet, sus protocolos y servicios, como medio para conectar los sistemas de su propia empresa, así como la propia Internet como infraestructura de comunicación para conseguir esa conexión. En un entorno como el de Internet no podemos asumir que todos los clientes tengan un PC, funcionando con Windows como sistema operativo, para poder ejecutar las interfaces de usuario basadas en formularios Windows. Existen otros dispositivos y sistemas operativos, potenciales clientes a los que hay que hay que ofrecer otro tipo de interfaz de acceso a nuestros servicios.

Una de las novedades más interesantes de la plataforma .Net es Asp.Net, el sucesor de Asp (Active Server Pages). Mediante Asp.Net podemos desarrollar aplicaciones, generalmente interfaces de usuarios, accesibles desde cualquier cliente que sea capaz de acceder a un servidor Web. Asp.Net se encarga de ejecutar la lógica en el servidor y

## **CAPITULO I “FUNDAMENTACION TEORICA”**

---

generar el código HTML apropiado para el cliente que efectúe la solicitud, si importar el tipo de dispositivo, hardware o sistema operativo en que esté ejecutándose.

ASP.NET es un marco de trabajo de programación generado en CLR (Common Language Runtime) que puede utilizarse en un servidor para generar eficaces aplicaciones Web. ASP.NET ofrece varias ventajas importantes acerca de los modelos de programación Web anteriores:

- Mejor rendimiento.
- Compatibilidad con herramientas de primer nivel.
- Eficacia y flexibilidad.
- Simplicidad.
- Facilidad de uso.
- Escalabilidad y disponibilidad.
- Posibilidad de personalización y extensibilidad.
- Seguridad.

### **1.2.6 SQL Server:**

En la actualidad, las compañías demandan una clase diferente de solución de base de datos. El rendimiento, la escalabilidad y la confiabilidad son esenciales y la anticipación al mercado es crítica. Aparte de estas cualidades empresariales fundamentales, SQL Server 2000 proporciona agilidad a sus operaciones de análisis y administración de datos al permitir a su organización adaptarse rápida y fácilmente para obtener ventaja competitiva en un entorno de cambios constantes. Desde una perspectiva de administración de datos y análisis, resulta crítico transformar los datos sin procesar en inteligencia empresarial y aprovechar las oportunidades que presenta el Web. SQL Server 2000 es un paquete completo de base de datos y análisis de datos que abre las puertas al rápido desarrollo de una nueva generación de aplicaciones comerciales de nivel empresarial, que pueden proporcionar a su compañía una ventaja competitiva crítica. SQL Server 2000 ha obtenido importantes galardones en pruebas de referencia por su escalabilidad y velocidad. Es un producto de base de datos totalmente habilitado para Web que proporciona una compatibilidad fundamental con el Lenguaje de

## **CAPITULO I “FUNDAMENTACION TEORICA”**

---

marcado extensible (XML, Extensible Markup Language) y la capacidad para realizar consultas en Internet y por encima del servidor de seguridad.

### **Conclusiones**

Es posible concluir, según los argumentos antes expuestos, que las tecnologías seleccionadas para el desarrollo del sistema cumplen con los requerimientos esperados. Existen en la actualidad un sinnúmero de alternativas para el desarrollo de aplicaciones, pero tratando de seguir una línea específica de tecnologías afines y capaces de garantizar un nivel elevado de productividad e integración no se optó por tecnologías de la categoría llamada Open Source dada la relación tiempo/esfuerzo manifestada en éstas no factible para la premura de la informatización de la UCI. No obstante, como se podrá constatar en los siguientes capítulos, se diseñó el sistema de una forma flexible pensando en una posible migración de tecnologías, ya sean Open Source o Propietario.

### **CAPITULO 2. Análisis del Sistema.**

#### **Introducción**

En este capítulo se realiza un estudio del problema dado a solucionar que está orientado a fundamentar la necesidad de la creación de un sistema que automatice el proceso de reservación de recursos en la UCI. Contiene además un análisis del negocio funcional actual y de los requerimientos esperados para el desarrollo del sistema. Se desarrollan los casos de uso previamente definidos y se llega a la selección de los casos a implementar para este primer ciclo.

#### **2.1 Situación problémica**

La actual situación del estado de automatización de los procesos conlleva a alejar la posibilidad de crear lo que sería una ciudad digital con la mayoría de los servicios completamente informatizados. Los factores que más influencia tienen en esta situación son: La poca integración de los sistemas actualmente existentes, el uso de tecnologías de forma heterogénea que afecta aun más la posible futura integración y la aun insuficiente automatización de los procesos y servicios.

Sería útil, por ejemplo, poder determinar, de forma automática y fluida, el estado del flujo del personal, fundamentalmente del estudiantado, que por determinadas razones se mueve fuera de la entidad. Esta información podría servir para la planificación de la alimentación y un sistema que controlara los pases planificados y las salidas extraoficiales podría ser bastante determinante; La planificación de las actividades docentes lleva consigo la utilización de medios materiales que, en el caso de la UCI, se mueven en grandes volúmenes lo que hace difícil su control, un sistema que sea capaz de gestionar la reservación de los recursos podría aportar gran comodidad y control a este proceso.

### **2.2 Objeto de automatización**

Se desea garantizar un sistema que permita realizar cualquier tipo de reservación, ya sea un pase de fin de semana, un pase masivo, un viaje nacional, un local, un tiempo de máquina o una excursión de carácter recreativo.

Existe en la actualidad un sistema implementado que controla el pase del estudiantado para los fines de semana. Existe además un sistema, en fase de implementación aun, que gestiona las reservaciones de tiempo de máquina y de los laboratorios de computación. Estos sistemas resuelven de alguna medida la necesidad de informatización que se necesita, pero son independientes y no cumplen con los requerimientos para formar parte de un sistema integrado como lo sería el proyecto “UCI Ciudad Digital”. Además de esto, restan todavía por implementar los sistemas de reservaciones para los viajes nacionales, las excursiones y actividades extra docentes, los pases masivos y la reservación de locales.

A continuación se muestra como se realizan estos procesos actualmente.

#### **2.2.1 Reservación de Pase:**

Existe un sistema informático que permite la reservación de los pases de fin de semana, fuera de este tiempo tanto el pase diario como el del domingo, se tienen que informar a la persona encargada de controlar estos y la misma tiene que tramitarlo.

Para reservar el pase de fin de semana, el estudiante solicita hasta una fecha fijada, salir el fin de semana. En estos casos se recogen los datos del estudiante que va a viajar, tales son: lugar (origen y destino), fecha, motivo, nombre, apellidos, facultad, y si el estudiante no es de la habana: el nombre del familiar, el parentesco, la dirección y el teléfono. Luego se hace una asignación del transporte, en estos casos ómnibus, donde van a viajar.

Para el pase diario, el estudiante que presenta alguna situación por la que necesita salir de pase un día en específico lo tiene que solicitar al decano o profesor autorizado

## **CAPITULO II “CARACTERÍSTICAS DEL SISTEMA”**

---

a dar este tipo de pase, en correspondencia con su facultad. El profesor o decano indistintamente, analizan el motivo por el cual el estudiante solicita y acepta o rechaza el pase según su criterio, seguido de esto tiene que generarle un ticket o autorización que justifique al estudiante a subir a los ómnibus. En este caso se recogen los mismos datos que en el pase de fin de semana, todo este proceso se realiza de forma manual.

El pase del domingo es una oferta para los estudiantes del interior que no tienen familiar en la Habana, en estos casos existen dos destinos: el Vedado (Coopelia) y la Habana Vieja (Parque central), los estudiantes tienen derecho a hacer una reservación para cualquiera de estos dos destinos, y escoger el punto desde donde van a regresar, para lo cual se requiere que los estudiantes notifiquen con tiempo si van a salir en esta ocasión para hacer una planificación temprana de los ómnibus que se asignarán a esta tarea.

### **2.2.2 Reservación de Pase Masivo:**

En dos fechas del año la escuela planifica los pases masivos, en estos casos se asumirá que tanto estudiantes como trabajadores y profesores, viajarán hacia sus provincias de origen, en caso de que deseen un destino diferente deben aclarar el lugar (provincia) hacia el que van a viajar, desde donde van a regresar, el motivo por el cual toman esa decisión para, en caso de aprobarse dicho cambio, poder hacer una planificación de los pasajes que se van a reservar. Los estudiantes deben notificarlos a los encargados de cada una de sus facultades y los profesores y trabajadores deben notificarlo al encargado de la planificación o tramitador como también se le llama, en estos casos si el personal a viajar no ha notificado en tiempo se le hace una planificación para su provincia de procedencia.

Además puede darse el caso de que la persona no vaya a viajar en el transporte que se destina y se necesita de esta aclaración para así no hacer reservaciones de más.

Una vez hecha la planificación el tramitador negocia con la agencia central las cifras y las fechas (de ida y regreso) y estas se publican en la Intranet para conocimiento de todos los interesados.

## **CAPITULO II “CARACTERÍSTICAS DEL SISTEMA”**

---

Para poder salir en el transporte se exige un ticket o pasaje en el que se especifican el nombre y apellidos, carne de identidad, fecha, número del tren, hora de salida, destino y una asignación coche-asientos, etc.

### **2.2.3 Reservación de Viajes Nacionales:**

Un profesor o trabajador (albergado), tiene derecho a dos reservaciones (de ida y regreso) por semestre para viajar a provincia, incluyendo el viaje del pase masivo.

Para hacer esta reservación el interesado solicitará al tramitador, en el modelo de Solicitud de Pago, la fecha en que desean viajar, el nombre y apellidos y el área a que pertenece.

Una vez hecha la reservación el tramitador le entrega el pasaje al solicitante, donde consta la fecha, el destino, nombre, apellidos, y asignación de lugar según el tipo de vehículo.

A los estudiantes que por un motivo justificado necesite viajar a su casa el centro le garantizará el pasaje de ida, el cual se le entregara previa solicitud, en el modelo de Solicitud de Pago, que debe especificar, nombre y apellidos y el destino de viaje.

Este tipo de reservación debe ser aprobada por el rector.

### **2.2.4 Reservación de Locales:**

Durante el transcurso del curso existe una planificación docente que define los horarios y locales, entre otras cosas, en los que se va a concretar el proceso docente como tal, pero es necesario lograr un mecanismo que sea capaz de gestionar las actividades no planificadas en transcurso del curso docente normal, las cuales requieren de medios y locales para su desempeño. No todos los locales permanecen ocupados todo el tiempo y este tiempo en que permanecen libres se puede emplear para actividades como conferencias o repasos dirigidos por profesores. He aquí donde surge la necesidad de implementar un mecanismo para garantizar la reservación de locales según su



## **CAPITULO II “CARACTERÍSTICAS DEL SISTEMA”**

---

particularidades (climatización, capacidad, número de televisores, etc.) para diversos fines por un personal autorizado, fundamentalmente por profesores.

Actualmente esto se realiza de forma manual, mediante coordinación con los encargados de los edificios y persiguiendo los horarios en los cuales el local se encuentre libre.

### **2.2.5 Reservación de Tiempo de Máquinas:**

En un centro de estudio especializado en las ciencias informáticas como lo es la Universidad de Ciencias Informáticas (UCI) la utilización de los medios informáticos juega un papel preponderante en lo que se refiere al aprovechamiento de los recursos que el estado ha puesto a disposición de la formación del estudiantado. He aquí donde entra a jugar un papel fundamental la gestión del uso que se le dé a los laboratorios y a las computadoras en particular. El uso planificado de estos medios está fundamentalmente centrado en los horarios docentes, pero no es en este tiempo solamente donde se logra el aprovechamiento del tiempo útil de las PC y el tiempo de preparación del estudiantado. Es entonces cuando aparece la necesidad de implementar un mecanismo que permita la reservación de tiempo (tiempo de máquina) para aprovechar al máximo las posibilidades que la presencia de estos medios brinda.

El proceso actual de reservación de tiempos de máquinas y laboratorios se realiza de forma manual, cuando un estudiante desea reservar un tiempo para trabajar en un laboratorio en particular se dirige a un técnico que es el encargado de llevar estos controles y efectúa la reservación de forma libre en el laboratorio deseado solo en los horarios de la noche (7<sup>mo</sup>, 8<sup>vo</sup> y 9<sup>no</sup> turnos) y por un tiempo equivalente a un turno de clase normal. El mecanismo para que un profesor reserve un laboratorio no está oficialmente definido, simplemente se resuelve el problema según un acuerdo con los técnicos y la disponibilidad del recurso.

En la actualidad existe un sistema en fase de implementación para gestionar el proceso de reservación de los tiempos de máquinas pero aún no está público. Este sistema comprende la reservación de los tiempos de máquina y los laboratorios de una manera

## **CAPITULO II “CARACTERÍSTICAS DEL SISTEMA”**

---

ordenada y planificada. Se comprende que un estudiante tiene derecho a reservar un solo tiempo de máquina al día en horarios nocturnos, también 7<sup>mo</sup>, 8<sup>vo</sup> y 9<sup>no</sup> turnos, cualquier día de la semana en un laboratorio de su facultad. Para esto se considera que las máquinas tienen un estándar de características de hardware y software instalados por lo que no se considera la posibilidad de elegir la máquina en la que se desea trabajar sino solamente el laboratorio. Aun no está previsto el mecanismo para permitir la reservación del tiempo de máquina en horarios diurnos en que no se esté empleando el laboratorio ni tampoco el que los profesores puedan reservar un laboratorio. Este sistema está siendo implementado sobre una tecnología ASP y se nutre de las bases de datos que poseen los técnicos para el control de los medios que interesa a su labor.

### **2.2.6 Reservación de Excursiones:**

Como parte del proceso de convivencia del estudiantado y el personal dentro de la universidad se hace necesaria la oferta de actividades recreativas, es entonces cuando se hace necesaria la posibilidad de acceder a éstas. El personal de la Casona es el encargado de llevar esta actividad, las actividades fundamentales que se llevan a cabo son las excursiones a lugares de carácter recreativo, histórico y cultural. Estas actividades generalmente tienen incluida la transportación que suele ser el parámetro limitante de la capacidad de oferta. Este proceso de reservación se realiza generalmente mediante la coordinación de los profesores con el personal de la Casona o mediante ofertas abiertas al estudiantado.

### **2.3 Propuesta de sistema**

#### **2.3.1 Objetivos Generales.**

- Crear un sistema único de reservación de recursos para la universidad.

#### **2.3.2 Objetivos Específicos.**

- Gestionar de forma automatizada el proceso de reservación de pases, viajes nacionales, locales, tiempo de máquina, pases masivos y excursiones.
- Brindar información de todo tipo referente a la presencia o no del personal en la institución para fines como la planificación de la cocina y los comedores.
- Brindar información del uso de los locales y máquinas por parte del personal.
- Realizar análisis estadísticos respecto a las preferencias del personal y el estudiantado.
- Realizar la distribución de personal según la disponibilidad de los medios de transporte en el caso de los pases cortos y masivos.

#### **2.3.3 Sistema propuesto (Software).**

Se considera la implementación de un software que sea capaz de gestionar la solicitud de reservaciones de recursos varios desde cualquier momento y lugar conectado a la red con una interfaz única que permita acceder a cualquier tipo de recurso a reservar.

El Sistema Integral de Reservaciones de la UCI debe gestionar todo el proceso de reservación de recursos variados. Para lograr este propósito se creará un sistema capaz de permitir la adición y configuración de cualquier recurso que se desee poner a consideración de los usuarios. Los recursos a reservar cuentan con atributos comunes que permitirán crear un mecanismo para gestionarlos de forma común aparte se proveerá un conjunto de módulos que realicen acciones específicas con fines de realizar cálculos para la confección de informes, etc.

## CAPITULO II “CARACTERÍSTICAS DEL SISTEMA”

---

Los atributos comunes son:

- Nombre del recurso a reservar.
- Descripción.
- Tiempo de vigencia para realizar y/o cancelar la reservación.

Algunos de los atributos dependen de factores externos y son diferentes para cada caso en particular debiéndose de implementar módulos de carácter y uso generales que sean capaces de proveer la información necesaria para los mismos. Ejemplo de esto sería la reservación de los locales y los tiempos de máquinas cuya disponibilidad está dada por sistemas externos, la reservación de los pases especiales y los viajes nacionales está también dentro de este grupo pues para que tenga efecto la reservación debe de ser aprobada por la persona destinada a este fin.

Estos son:

- Disponibilidad del recurso.
- Aprobación en caso de necesitarla.

Otros recursos estarán permanentemente disponibles para reservar pues a pesar de poseer los atributos comunes se reimplantan después de cada cierre al vencerse el plazo de vencimiento como disponibles para su reservación siguiendo una secuencia periódica, diaria, semanal, etc. Ejemplo de esto sería la reservación de los pases de fin de semana los cuales tienen un plazo determinado para ser efectuado pero cada vez que transcurre la semana se vuelve a poner el mismo a consideración de los usuarios y así de forma periódica. Además es necesario tener en cuenta la posibilidad de que un recurso esté publicado para su reservación de forma periódica pero que solo sea vigente un tiempo determinado.

Estos atributos se podrían denominar:

- Periodicidad.
- Tiempo de publicación.

## **CAPITULO II “CARACTERÍSTICAS DEL SISTEMA”**

---

Existen además recursos cuya disponibilidad se pondrá a consideración del usuario en forma de grupos debido a que están compuestos por elementos idénticos, ejemplo de esto serían los asientos de un ómnibus, las capacidades para una excursión o las entradas para un espectáculo. Sin embargo otros, a pesar de tener una cierta agrupación, se deben de gestionar de forma independiente debido a que poseen cualidades que los diferencia entre si, ejemplo de esto serían las computadoras, las cuales están comprendidas dentro del recurso laboratorio pero presentan diferencias que las hace singulares ante la consideración del usuario, está también el caso de los locales, los cuales están marcados por su diferencias particulares, capacidad, climatización, etc.

Se podría considerar un atributo que seria:

- Contenedor.

Siguiendo esta filosofía se permitirá gestionar todos los procesos de reservaciones que se realizan actualmente pero además permitirá adicionar en cualquier momento cualquier nuevo recurso que se desee poner a consideración de los usuarios con solo configurar sus atributos que estarán previstos con anterioridad. Solo en caso de necesitar algún atributo o acción especial se procederá a adicionar otro módulo que de respuesta al requerimiento enriqueciendo de esta forma al sistema, opción que estará prevista a la hora de diseñar el software.

### **2.4 Modelo de negocio**

Dada la descripción expuesta anteriormente respecto al funcionamiento de los procesos de gestión de reservación de recursos en la forma en que se resuelve actualmente, es posible aseverar que la solución a este problema sería la informatización de todo el proceso, de manera que el flujo no pese en las manos del personal encargado de controlar este. La idea de lograr la unificación de los procesos, que actualmente se realizan de forma separada es la forma más factible de lograrlo pero sería imposible pretender realizar esto de forma manual por lo que una nueva propuesta de negocio deberá estar soportada indiscutiblemente sobre un sistema informático.

#### **2.4.1 Modelo del negocio**

Ver Anexo 1.

#### **2.4.2 Especificación de los requisitos de software**

El sistema se debe nutrir de otros sistemas que brindarán la información necesaria para el funcionamiento correcto, proveerá además a otros sistemas que lo necesiten.

El sistema Personas de la UCI deberá proveer la información sobre las personas que operarán el sistema, así como el Sistema de Matricula lo hará respecto a los estudiantes.

El sistema Inventarios deberá proveer la información de los muebles e inmuebles para tomar información como los locales y computadoras que serán algunos de los recursos que podrán ser reservados.

El sistema de transporte permitirá consultar la información referente a la disponibilidad de ómnibus según la cual se podrá realizar la distribución de la transportación para los pases y las actividades.

#### **2.4.3 Requerimientos Funcionales.**

## CAPITULO II “CARACTERÍSTICAS DEL SISTEMA”

---

### 1. Autenticación de usuario.

- Verificar usuario y contraseña.
- Otorgar los accesos al usuario.

### 2. Reservación de recursos.

- Elegir el recurso a reservar.
- Personalizar la reservación si es requerido.
- Verificar el plazo establecido para reservar.
- Adicionar a la base de datos la nueva reservación.
- Mostrar mensaje de confirmación.

### 3. Modificar una reservación.

- Mostrar un listado con los recursos reservados concernientes al usuario.
- Verificar el plazo establecido para modificar.
- Modificar los parámetros de interés.
- Actualizar la base de datos.
- Mostrar mensaje de confirmación.

### 4. Cancelación de una reservación.

- Mostrar un listado con los recursos reservados concernientes al usuario.
- Verificar el plazo establecido para cancelar.
- Marcar la reservación como cancelada en la base de datos.
- Mostrar mensaje de confirmación.

### 5. Tramitar solicitud en caso que lo requiera, puede se autorizada o rechazada.

### 6. Permitir la adición de nuevos módulos con nuevas funcionalidades sin la necesidad de recompilar el sistema.

- Debe contener un módulo capaz de realizar una distribución del personal por ómnibus para una actividad específica que así lo requiera.

## **CAPITULO II “CARACTERÍSTICAS DEL SISTEMA”**

---

- Debe poseer un módulo que consulte el sistema de inventarios y brinde la información necesaria respecto a los medios materiales.
- Debe contener un módulo que consulte y brinde información respecto al transporte por lo que deberá interactuar con el sistema que gestiona este recurso.
- Debe poseer un módulo que consulte al sistema de horarios y extraiga información respecto a la utilización de los medios de carácter docente.

### **7. Presentación de informes y estadísticas.**

- Brindar informe de estado actual e histórico del estado de las reservaciones.
- Permitir la búsqueda de reservaciones actuales e históricas según criterios avanzados.
- Brindar información sobre el movimiento del personal respecto a pases, viajes, etc.

### **2.4.4 Requerimientos no Funcionales.**

1. El sistema será integrado, debe permitir realizar todo el proceso de reservaciones que actualmente se realiza por sistemas independientes o a mano.
2. Contará con niveles de acceso para restringir la operación respecto a los recursos a reservar.
3. El sistema será implementado usando una tecnología Web.
4. La información será almacenada en bases de datos, dejando registro de toda operación realizada.
5. El sistema debe permitir reservar todos los recursos que actualmente están implicados. Los recursos son:
  - Pase corto individual.
  - Aulas.
  - Laboratorios.
  - Máquinas.
  - Actividades extra-docentes.



## **CAPITULO II “CARACTERÍSTICAS DEL SISTEMA”**

---

- Viajes nacionales.
  - Pase masivo.
6. La reservación del pase corto tendrá en cuenta la distribución del personal según las rutas establecidas y los ómnibus disponibles.
  7. La reservación de las aulas y los laboratorios se hará teniendo en cuenta la planificación docente y los horarios.
  8. La reservación de las máquinas se subordina a los laboratorios.
  9. El sistema deberá poseer una interfaz al usuario agradable y en concordancia al estilo de las aplicaciones de la UCI
  10. El sistema restringirá el acceso a los usuarios por lo que el uso y manejo del sistema estará controlado. Se realizará una autenticación con el directorio activo y se cargarán los niveles de accesos según los criterios que el administrador del sistema haya definido para el usuario en particular y según el tipo de persona y el rol que juegue en el sistema.

### **2.4.5 Definición de los casos de uso del sistema.**

Ver Anexo 2.

### **Conclusiones**

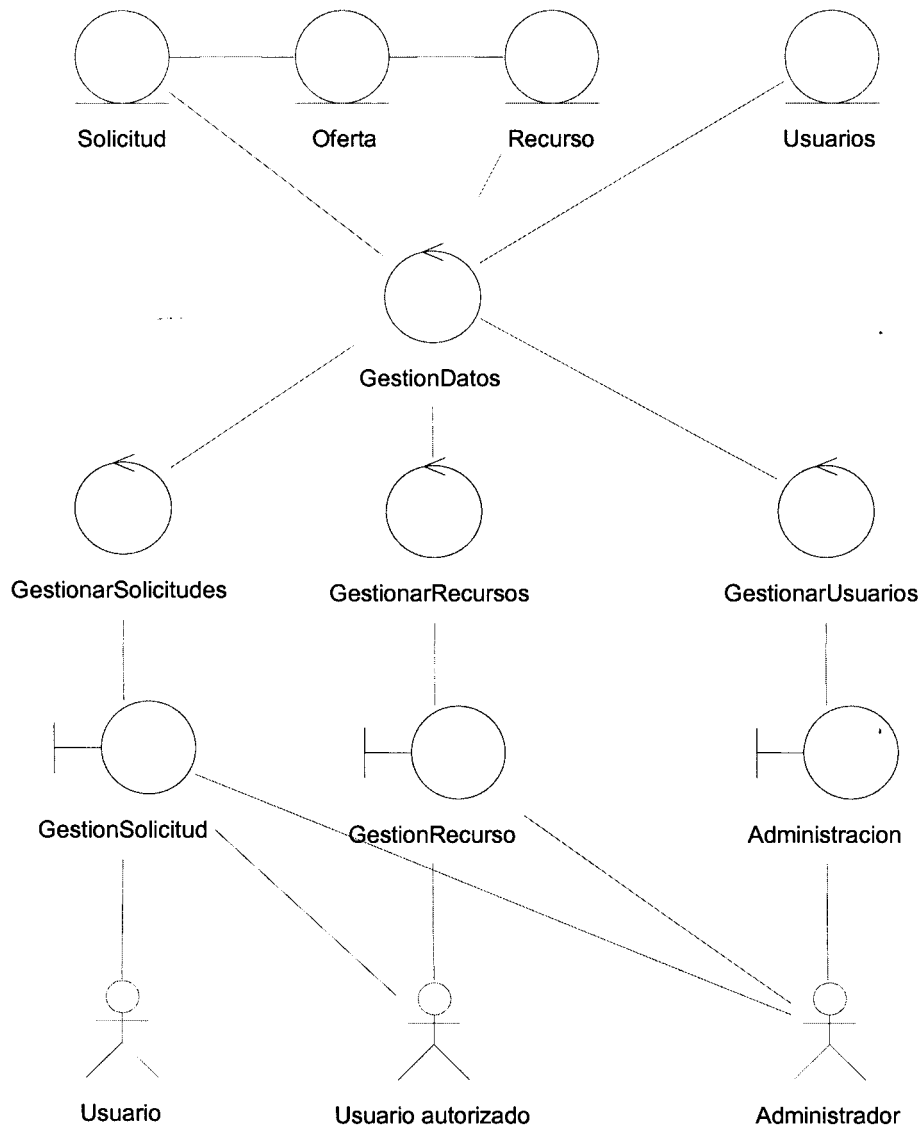
En este capítulo se realizó un estudio detallado de la reglas del negocio involucrado en el campo de acción y se completó su análisis y modelado. Se realizó una propuesta de sistema que satisfaga las especificaciones de automatización de los procesos del negocio actuales y se completó el levantamiento de los requerimientos esperados para el sistema.

## **CAPITULO 3. Análisis y diseño del sistema**

### **Introducción**

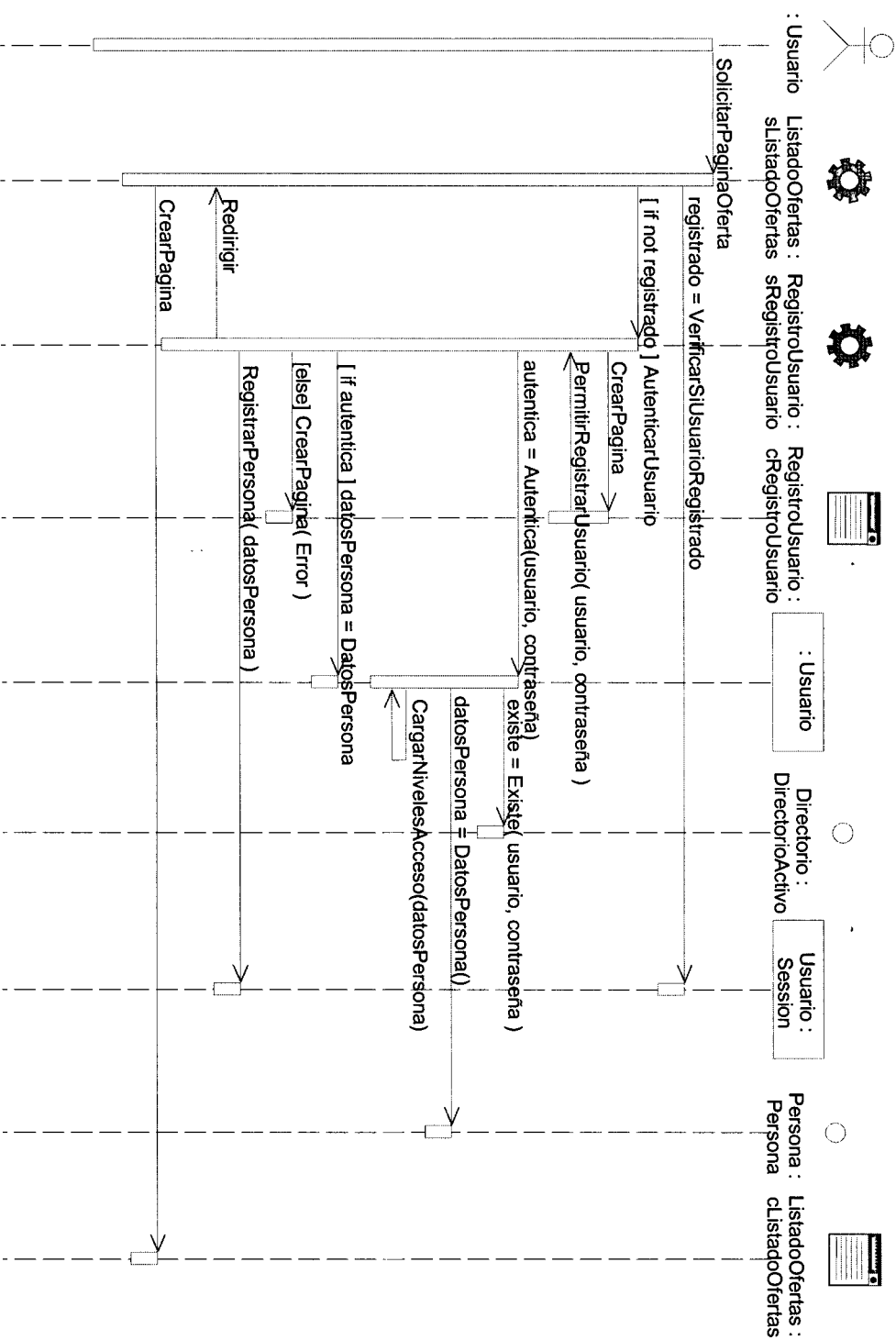
En este capítulo se abordará el análisis y diseño de la aplicación en los aspectos más importantes en función de satisfacer los requerimientos esperados, se confecciona el diagrama de clases de análisis y los diagramas de interacción, se completa el modelo de clases de diseño y se aplica el diseño Web para las especificaciones de la aplicación. Se realiza además el diseño del modelo de datos para el sistema.

**3.1 Diagrama de clases del análisis**

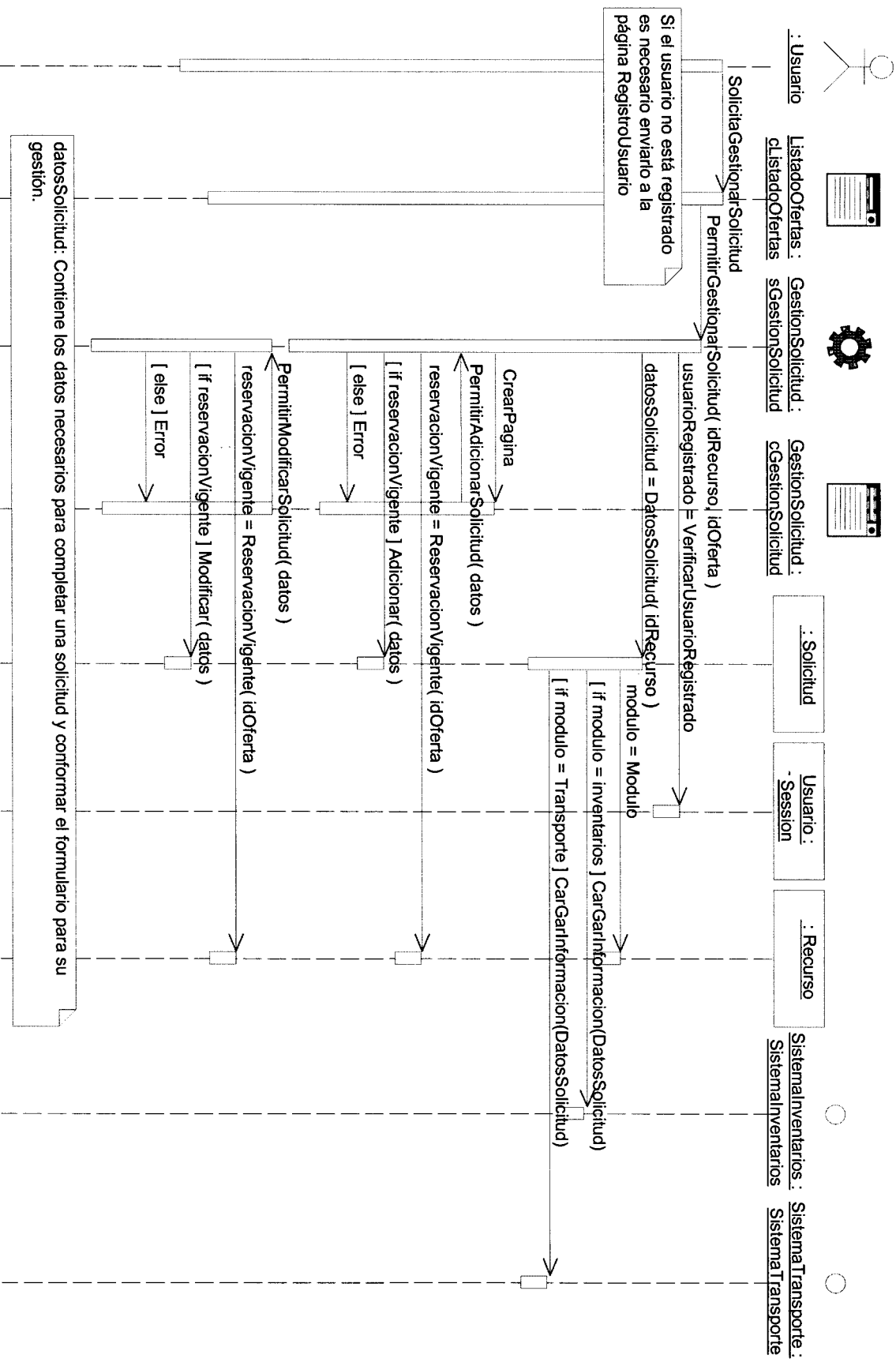


**3.2 Diagramas de interacción**

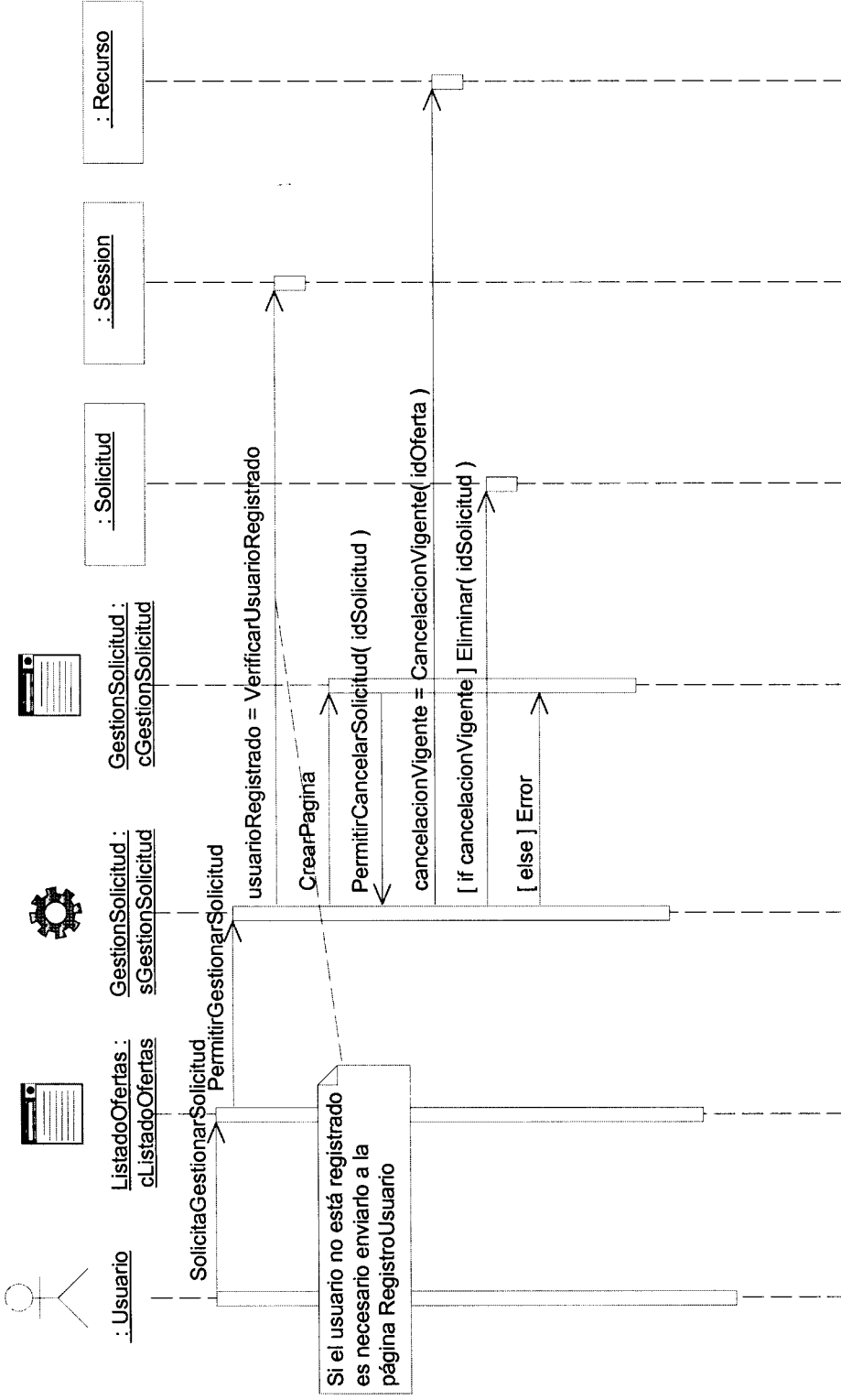
DIS Autenticación de usuarios



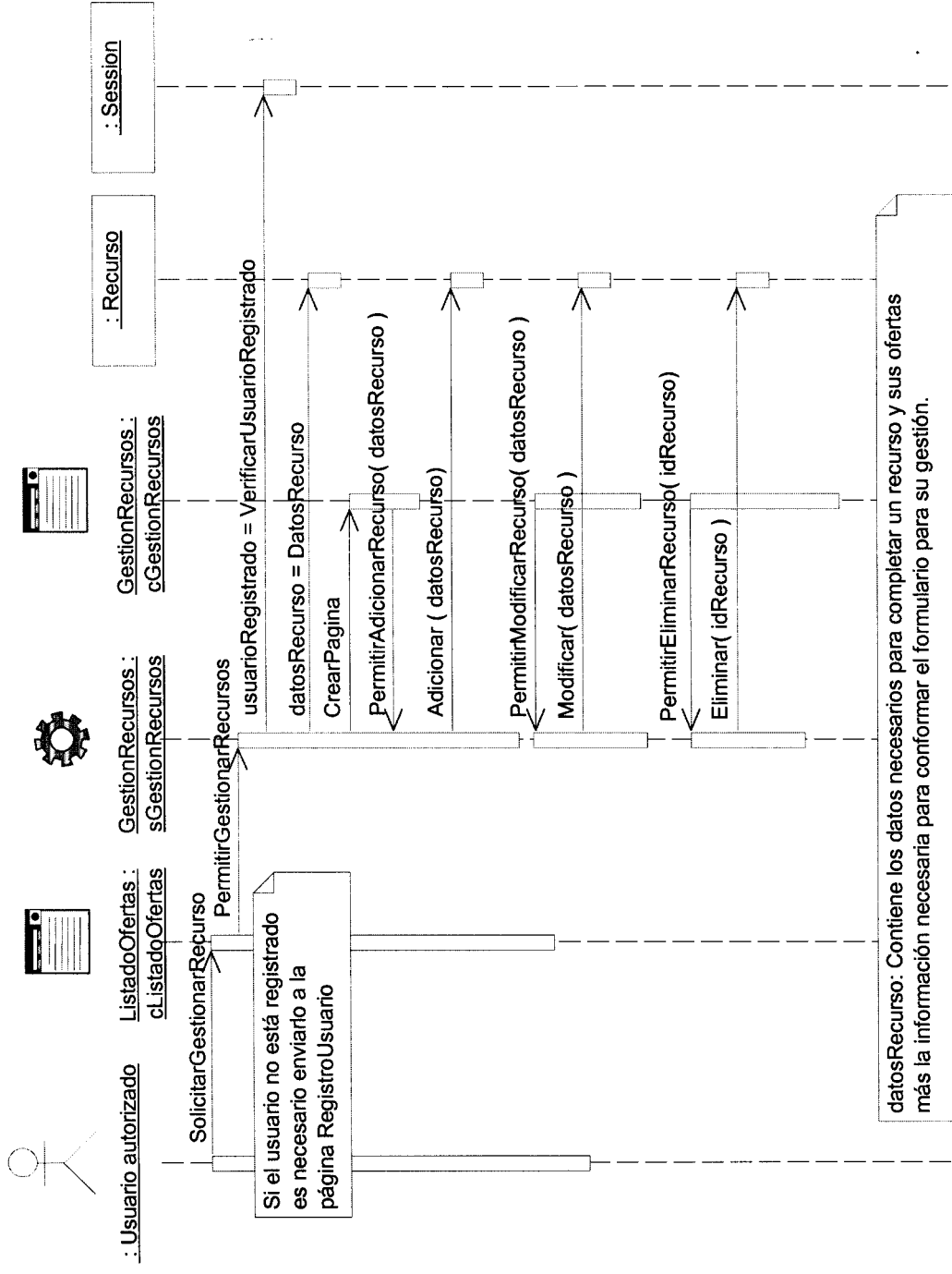
DIS Solicitar reservación



DIS Cancelar reservación

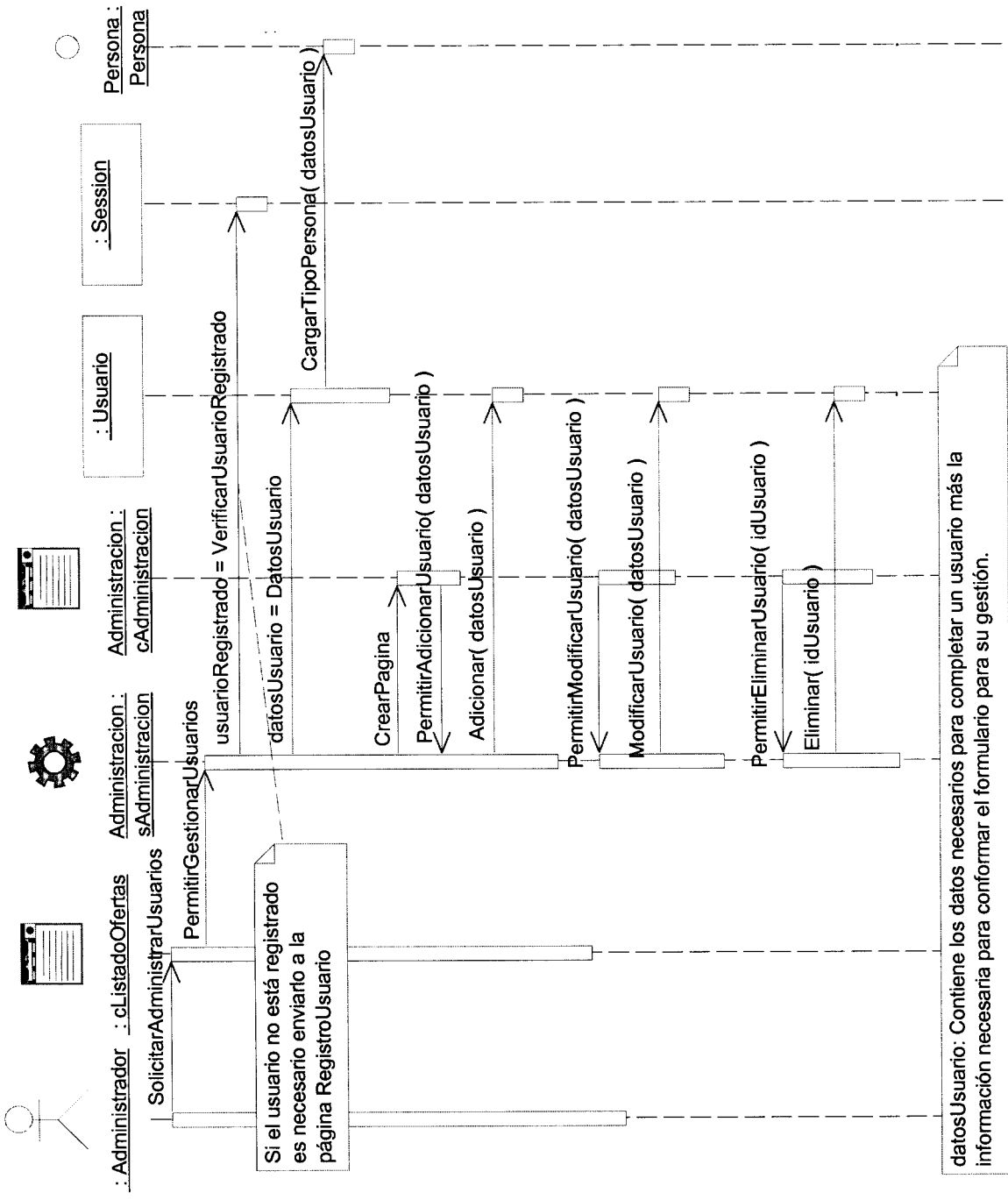


DIS Gestionar recursos





DIS Administrar usuarios



### **3.3 Diseño de las clases**

En el diseño del sistema se tomaron algunas consideraciones que ayudarán a darle claridad para el posterior desarrollo. Se ha diseñado una jerarquía de clases que será la encargada de gestionar toda la data de forma única teniendo en cuenta la posibilidad de emplear varios gestores de diferentes tecnologías por lo que no estarán ligadas a ninguna base de datos en específico. Estas clases se limitan a implementar métodos para realizar consultas, ya sea de acción o de selección pero en este caso solo manipularán procedimientos almacenados dada su probada eficiencia, no obstante no sería difícil realizar pequeñas modificaciones para emplear consultas SQL. Serán las encargadas además de manipular las conexiones y las transacciones.

Dado que la aplicación que nos ocupa limita su funcionamiento prácticamente a manipular datos almacenados en una base de datos se creó una jerarquía que permita un comportamiento de los objetos como si fueran entidades lógicas con sus funciones de proveer datos y almacenarlos en función de las especificaciones de la entidad representada. Para este propósito se implementó una clase Entidad que se comportará como cualquier entidad de datos con sus posibles métodos de devolución, modificación e inserción de datos. Para que esta filosofía de diseño funcione correctamente cada método de la clase deberá tener implementado su contraparte en consultas SQL que en este caso estará contenido en procedimientos almacenados, siguiendo una nomenclatura específica, pero es posible con la realización de pequeños cambios implementar un mecanismo para emplear consultas SQL desde el lado de la aplicación, este tema de la implementación de las consultas se retomará con mas detalle en la sección del diseño del modelo de datos. Todas las clases que manejan la lógica del sistema heredarán de esta clase Entidad, implementando los mecanismos específicos para sus responsabilidades particulares.

Según esta filosofía de diseño la aplicación quedará perfectamente dividida en tres capas mínimas, capa de datos, capa de negocio y capa de presentación.

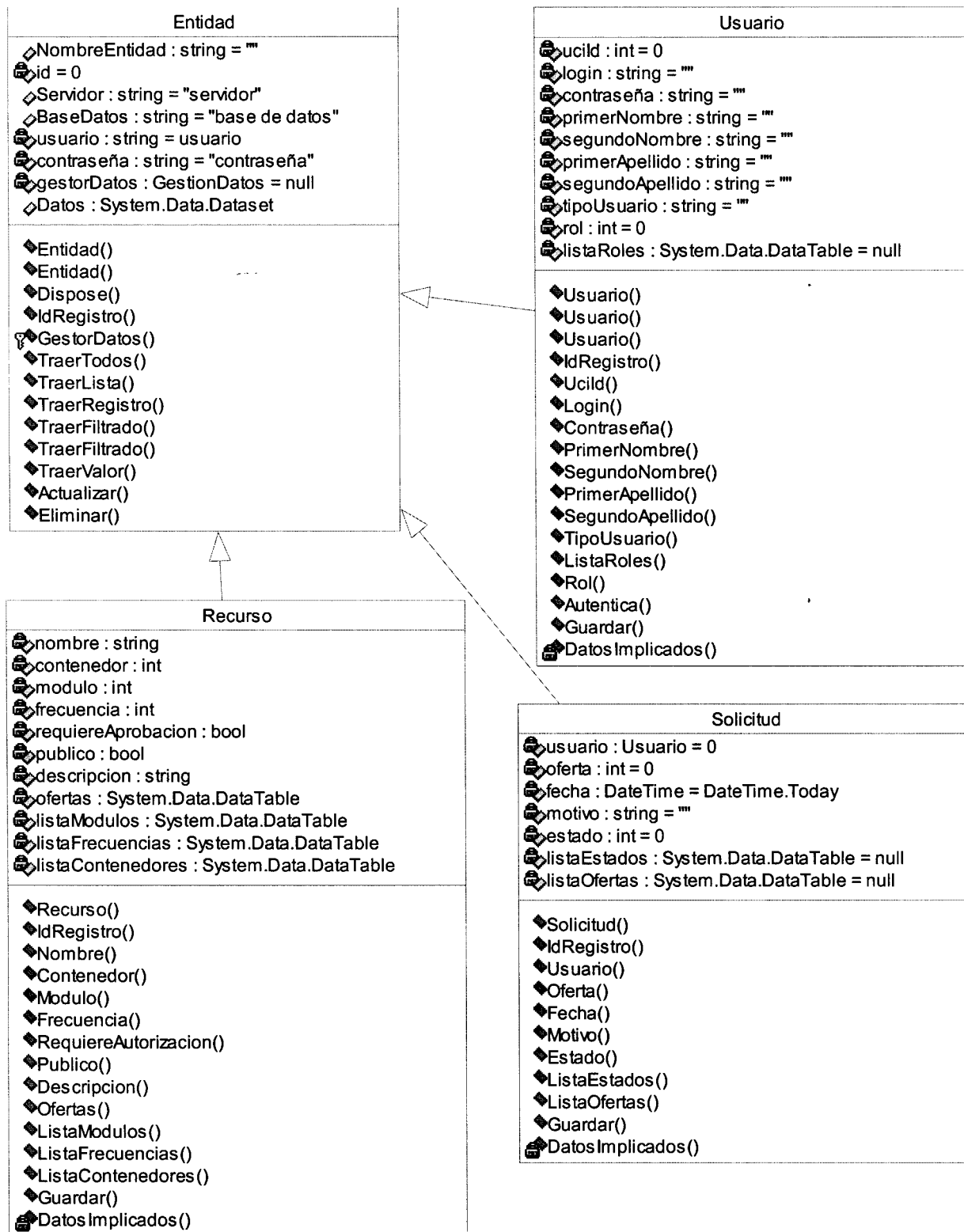
**3.4 Diagrama de clases**

**3.4.1 Capa de datos**



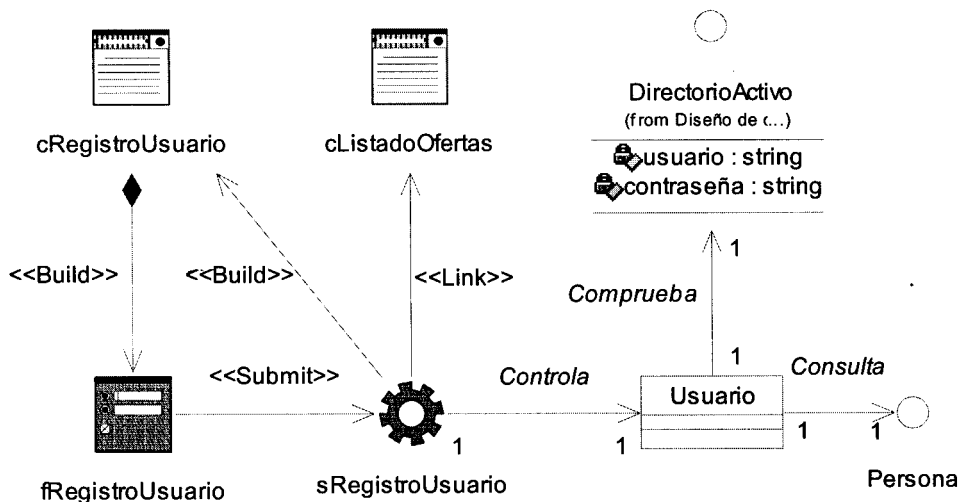
## CAPITULO III “ANÁLISIS Y DISEÑO DEL SISTEMA”

### 3.4.2 Capa de negocio

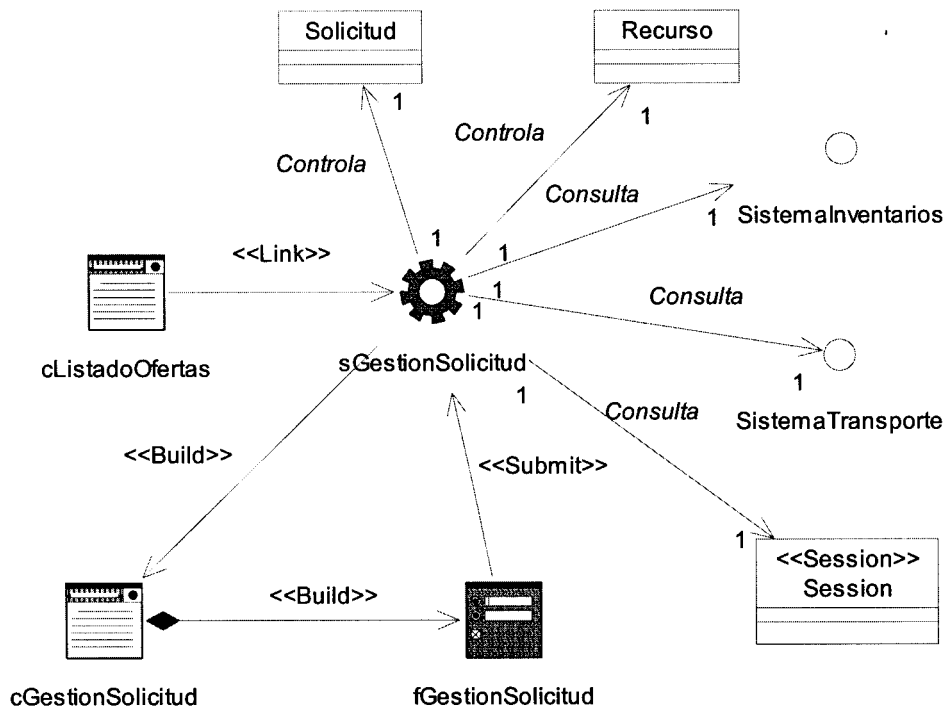


### 3.5 Diagramas de diseño Web

#### DDW Autenticar usuarios

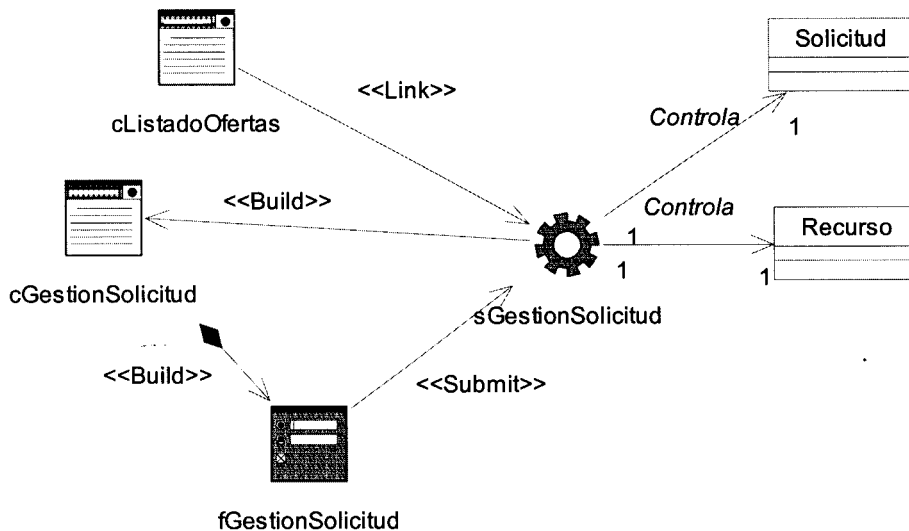


#### DDW Solicitar reservación

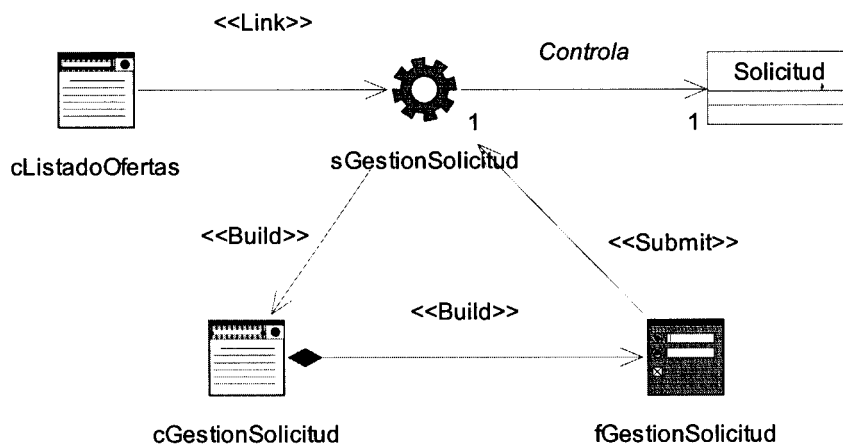


### CAPITULO III "ANÁLISIS Y DISEÑO DEL SISTEMA"

#### DDW Modificar solicitud

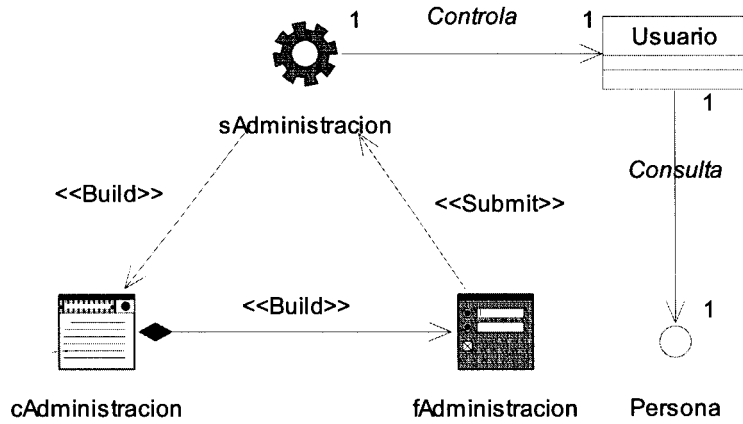


#### DDW Cancelar solicitud

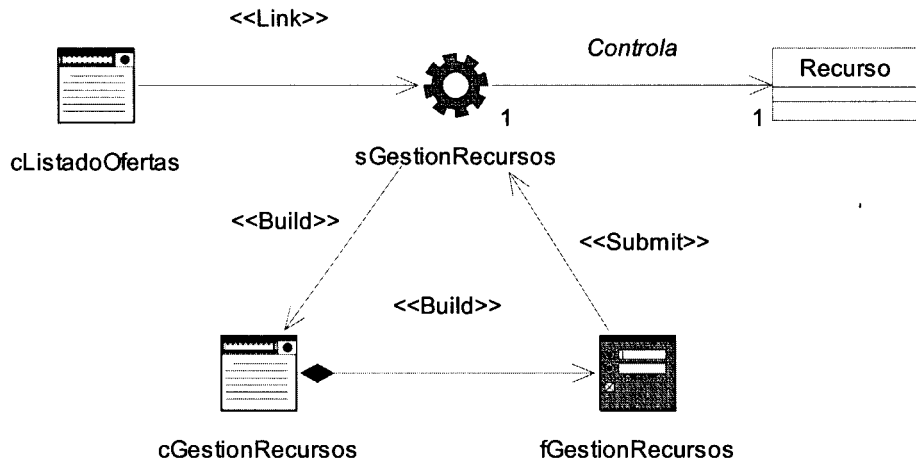


# CAPITULO III "ANÁLISIS Y DISEÑO DEL SISTEMA"

## DDW Administrar usuarios



## DDW Gestionar recursos



**3.6 Descripción de las clases**

<b>Nombre:</b> GestionDatos (abstracta)	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
servidor	string
baseDatos	string
cadenaConexion	string
conexion	System.Data.IDbConnection
transaccion	System.Data.IDbTransaction
enTransaccion	bool
<b>Para cada responsabilidad:</b>	
Nombre:	GestionDatos()
Descripción:	Es el constructor por defecto de la clase.
Nombre:	string Servidor
Descripción:	Es la propiedad que encapsula al atributo servidor
Nombre:	string BaseDatos
Descripción:	Es la propiedad que encapsula al atributo baseDatos
Nombre:	string CadenaConexion
Descripción:	Es la propiedad abstracta que encapsula al atributo cadenaConexion
Nombre:	System.Data.IDbConnection Conexion
Descripción:	Es la propiedad que encapsula al atributo conexión, es de solo lectura y se encarga, en caso de ser accedida, de crear la conexión si no existe.
Nombre:	System.Data.IDbTransaction Transaccion
Descripción:	Es la propiedad que encapsula al atributo transaccion
Nombre:	System.Data.DataSet TraerDataSet(string procedimientoAlmacenado)
Descripción:	Método que devuelve un dataset dado un procedimiento almacenado existente con una consulta de selección.
Nombre:	System.Data.DataSet TraerDataSet(string procedimientoAlmacenado, params object[] argumentos)
Descripción:	Sobrecarga del método anterior que permite emplear procedimientos almacenados con parámetros.
Nombre:	System.Data.DataTable TraerDataTable(string procedimientoAlmacenado)



### CAPITULO III “ANÁLISIS Y DISEÑO DEL SISTEMA”

Descripción:	Método que devuelve un datatable dado un procedimiento almacenado existente con una consulta de selección.
Nombre:	System.Data.DataTable TraerDataTable(string procedimientoAlmacenado, params object[] argumentos)
Descripción:	Sobrecarga del método anterior que permite emplear procedimientos almacenados con parámetros.
Nombre:	object TraerValor(string procedimientoAlmacenado)
Descripción:	Método que devuelve un valor dado un procedimiento almacenado existente con un parámetro de salida.
Nombre:	object TraerValor(string procedimientoAlmacenado, params object[] argumentos)
Descripción:	Sobrecarga del método anterior que permite emplear procedimientos almacenados con parámetros.
Nombre:	System.Data.IDbConnection CrearConexion(string cadenaConexion)
Descripción:	Método abstracto que debe encargarse de crear la conexión para el gestor de datos específico.
Nombre:	System.Data.IDbCommand Comando(string procedimientoAlmacenado)
Descripción:	Método abstracto que debe encargarse de crear el comando para el gestor de datos específico.
Nombre:	System.Data.IDataAdapter CrearDataAdapter(string procedimientoAlmacenado, params object[] argumentos)
Descripción:	Método abstracto que debe encargarse de crear el dataadapter para el gestor de datos específico.
Nombre:	CargarParametros(IDbCommand Comando, params object[] argumentos)
Descripción:	Método abstracto que debe encargarse de la carga de los parámetros en el comando para el gestor de datos específico.
Nombre:	int Ejecutar(string procedimientoAlmacenado)
Descripción:	Método encargado de ejecutar un procedimiento almacenado que contenga una consulta de acción, devuelve el número de registros afectados.
Nombre:	int Ejecutar(string procedimientoAlmacenado, params object[] argumentos)
Descripción:	Sobrecarga del método anterior que permite emplear un

### CAPITULO III “ANÁLISIS Y DISEÑO DEL SISTEMA”

	procedimiento almacenado con parámetros.
Nombre:	IniciarTransaccion()
Descripción:	Método encargado de crear e iniciar una transacción.
Nombre:	TerminarTransaccion()
Descripción:	Método encargado de completar la transacción iniciada.
Nombre:	AbortarTransaccion()
Descripción:	Método encargado de cancelar una transacción iniciada.

<b>Nombre:</b> GestionDatosSQL (hereda de GestionDatos)	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
usuario	string
contraseña	string
<b>Para cada responsabilidad:</b>	
Nombre:	GestionDatosSql()
Descripción:	Es el constructor por defecto de la clase.
Nombre:	GestionDatosSql(string cadenaConexion)
Descripción:	Sobrecarga del constructor que inicializa el atributo cadenaConexion heredado.
Nombre:	GestionDatosSql(string servidor, string baseDatos, string usuario, string contraseña)
Descripción:	Sobrecarga del constructor que inicializa los atributos usuario y contraseña de la clase y servidor y baseDatos heredados.
Nombre:	string CadenaConexion
Descripción:	Implementación de la propiedad abstracta heredada del mismo nombre.
Nombre:	string Usuario
Descripción:	Es la propiedad de solo escritura que encapsula el atributo usuario.
Nombre:	string Contraseña
Descripción:	Es la propiedad de solo escritura que encapsula el atributo contraseña.
Nombre:	System.Data.IDbConnection CrearConexion(string cadenaConexion)
Descripción:	Implementación de la propiedad abstracta heredada del mismo nombre.

### CAPITULO III “ANÁLISIS Y DISEÑO DEL SISTEMA”

Nombre:	CargarParametros(IDbCommand comando, params object[] argumentos)
Descripción:	Implementación de la propiedad abstracta heredada del mismo nombre.
Nombre:	System.Data.IDbCommand Comando(string procedimientoAlmacenado)
Descripción:	Implementación de la propiedad abstracta heredada del mismo nombre.
Nombre:	System.Data.IDataAdapter CrearDataAdapter(string procedimientoAlmacenado, params object[] argumentos)
Descripción:	Implementación de la propiedad abstracta heredada del mismo nombre.
Nombre:	object TraerValor(string procedimientoAlmacenado)
Descripción:	Método que devuelve un valor dado un procedimiento almacenado existente con un parámetro de salida.
Nombre:	object TraerValor(string procedimientoAlmacenado, params object[] argumentos)
Descripción:	Sobrecarga del método anterior que permite emplear procedimientos almacenados con parámetros.
Nombre:	System.Data.IDbConnection CrearConexion(string cadenaConexion)
Descripción:	Método abstracto que debe encargarse de crear la conexión para el gestor de datos específico.
Nombre:	System.Data.IDbCommand Comando(string procedimientoAlmacenado)
Descripción:	Método abstracto que debe encargarse de crear el comando para el gestor de datos específico.
Nombre:	System.Data.IDataAdapter CrearDataAdapter(string procedimientoAlmacenado, params object[] argumentos)
Descripción:	Método abstracto que debe encargarse de crear el dataadapter para el gestor de datos específico.

<b>Nombre:</b> Entidad	
<b>Clase controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
NombreEntidad	string
id	int
Servidor	string

### CAPITULO III “ANÁLISIS Y DISEÑO DEL SISTEMA”

BaseDatos	string
usuario	string
contraseña	string
gestorDatos	GestionDatos
Datos	System.Data.DataSet
<b>Para cada responsabilidad:</b>	
Nombre:	Entidad()
Descripción:	Es el constructor por defecto de la clase.
Nombre:	Entidad(string nombreEntidad)
Descripción:	Sobrecarga del constructor que inicializa el atributo nombreEntidad.
Nombre:	int IdRegistro
Descripción:	Propiedad que encapsula el atributo idRegistro, a asignar un identificador válido se cargan los datos referentes a la entidad creada correspondientes al idRegistro en la tabla correspondiente.
Nombre:	GestionDatos GestorDatos()
Descripción:	Crea una nueva instancia de la clase GestorDatos, en caso de no existir, según el gestor de datos correspondiente.
Nombre:	System.Data.DataTable TraerTodos()
Descripción:	Método que devuelve todos los registros de la entidad especificada correspondiente a la tabla del mismo nombre.
Nombre:	System.Data.DataTable TraerLista ()
Descripción:	Método que devuelve todos los registros de la entidad especificada correspondiente a la tabla del mismo nombre, tomando solamente los dos campos mas significativos en la forma Id y Descripción.
Nombre:	System.Data.DataRow TraerRegistro(int id)
Descripción:	Retorna el registro especificado por id.
Nombre:	System.Data.DataTable TraerFiltrado(string filtro)
Descripción:	Retorna un conjunto de registros seleccionados según el criterio especificado en filtro.
Nombre:	System.Data.DataTable TraerFiltrado(string filtro, params object[] argumentos)
Descripción:	Sobrecarga del método anterior que permite el paso de argumentos aparte del criterio de selección.
Nombre:	object TraerValor(string condicion, params object[] argumentos)
Descripción:	Devuelve un valor que cumpla con la condición especificada en condición.

### CAPITULO III “ANÁLISIS Y DISEÑO DEL SISTEMA”

Nombre:	Actualizar()
Descripción:	Actualiza los cambios realizados al conjunto de datos en la base de datos.
Nombre:	Eliminar()
Descripción:	Se encarga de eliminar la actual entidad específica a la que se está haciendo referencia.

<b>Nombre:</b> Recurso (hereda de entidad)	
<b>Clase controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
nombre	string
contenedor	int
modulo	int
frecuencia	int
requiereAprobacion	bool
publico	bool
descripcion	string
ofertas	System.Data.DataTable
listaModulos	System.Data.DataTable
listaFrecuencias	System.Data.DataTable
listaContenedores	System.Data.DataTable
<b>Para cada responsabilidad:</b>	
Nombre:	Recurso()
Descripción:	Es el constructor por defecto de la clase.
Nombre:	int IdRegistro
Descripción:	Propiedad que sobre escribe la propiedad heredada del mismo nombre adicionándole funcionalidades mas específicas.
Nombre:	string Nombre
Descripción:	Propiedad que encapsula el atributo nombre.
Nombre:	int Contenedor
Descripción:	Propiedad que encapsula el atributo contenedor.
Nombre:	int Modulo
Descripción:	Propiedad que encapsula el atributo modulo.
Nombre:	int Frecuencia
Descripción:	Propiedad que encapsula el atributo Frecuencia.
Nombre:	bool RequiereAutorizacion

### CAPITULO III “ANÁLISIS Y DISEÑO DEL SISTEMA”

Descripción:	Propiedad que encapsula el atributo requiereAutorizacion.
Nombre:	bool Publico
Descripción:	Propiedad que encapsula el atributo publico.
Nombre:	string Descripcion
Descripción:	Propiedad que encapsula el atributo descripcion.
Nombre:	System.Data.DataTable Ofertas
Descripción:	Propiedad que encapsula el atributo Ofertas.
Nombre:	System.Data.DataTable ListaModulos
Descripción:	Propiedad que encapsula el atributo listaModulos.
Nombre:	System.Data.DataTable ListaFrecuencias
Descripción:	Propiedad que encapsula el atributo listaFrecuencias.
Nombre:	System.Data.DataTable ListaContenedores
Descripción:	Propiedad que encapsula el atributo listaContenedores.
Nombre:	Guardar()
Descripción:	Adiciona o modifica la información de un recurso en la base de datos según sea el caso.
Nombre:	DatosImplicados(int id)
Descripción:	Devuelve el conjunto de datos implicados en la entidad Recurso.

<b>Nombre:</b> Solicitud (hereda de entidad)	
<b>Clase controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
usuario	int
oferta	int
fecha	DateTime
motivo	string
estado	int
listaEstados	System.Data.DataTable
listaOfertas	System.Data.DataTable
<b>Para cada responsabilidad:</b>	
Nombre:	Solicitud ()
Descripción:	Es el constructor por defecto de la clase.
Nombre:	int IdRegistro
Descripción:	Propiedad que sobre escribe la propiedad heredada del mismo nombre adicionándole funcionalidades mas específicas.

### CAPITULO III “ANÁLISIS Y DISEÑO DEL SISTEMA”

Nombre:	string Usuario
Descripción:	Propiedad que encapsula el atributo usuario.
Nombre:	int Oferta
Descripción:	Propiedad que encapsula el atributo oferta.
Nombre:	DateTime Fecha
Descripción:	Propiedad que encapsula el atributo fecha.
Nombre:	string Motivo
Descripción:	Propiedad que encapsula el atributo motivo.
Nombre:	int Estado
Descripción:	Propiedad que encapsula el atributo estado.
Nombre:	System.Data.DataTable ListaEstados
Descripción:	Propiedad que encapsula el atributo listaEstados.
Nombre:	System.Data.DataTable ListaOfertas
Descripción:	Propiedad que encapsula el atributo listaOfertas.
Nombre:	Guardar()
Descripción:	Adiciona o modifica la información de una solicitud en la base de datos según sea el caso.
Nombre:	DatosImplicados(int id)
Descripción:	Devuelve el conjunto de datos implicados en la entidad Solicitud.

<b>Nombre:</b> Usuario (hereda de entidad)	
<b>Clase controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
uclid	int
login	string
contraseña	string
primerNombre	string
segundoNombre	string
primerApellido	string
segundoApellido	string
tipoUsuario	string
rol	int
listaRoles	System.Data.DataTable
<b>Para cada responsabilidad:</b>	
Nombre:	Usuario ()

### CAPITULO III “ANÁLISIS Y DISEÑO DEL SISTEMA”

Descripción:	Es el constructor por defecto de la clase.
Nombre:	Usuario (int id)
Descripción:	Sobrecarga del constructor que inicializa la propiedad Ucild.
Nombre:	Usuario(string usuario)
Descripción:	Sobrecarga del constructor que inicializa la propiedad Usuario.
Nombre:	int IdRegistro
Descripción:	Propiedad que sobre escribe la propiedad heredada del mismo nombre adicionándole funcionalidades mas específicas.
Nombre:	Int Ucild
Descripción:	Propiedad que encapsula el atributo ucild.
Nombre:	string Login
Descripción:	Propiedad que encapsula el atributo login.
Nombre:	string Contraseña
Descripción:	Propiedad de solo escritura que en capsula el atributo contraseña.
Nombre:	string PrimerNombre
Descripción:	Propiedad que encapsula el atributo primerNombre.
Nombre:	string SegundoNombre
Descripción:	Propiedad que encapsula el atributo segundoNombre.
Nombre:	string PrimerApellido
Descripción:	Propiedad que encapsula el atributo primerApellido.
Nombre:	string SegundoApellido
Descripción:	Propiedad que encapsula el atributo segundoApellido.
Nombre:	System.Data.DataTable ListaRoles
Descripción:	Propiedad que encapsula el atributo listaRoles.
Nombre:	string TipoUsuario
Descripción:	Propiedad que encapsula el atributo tipoUsuario.
Nombre:	bool Autentica()
Descripción:	Método que autentica al usuario contra el directorio activo.
Nombre:	Guardar()
Descripción:	Adiciona o modifica la información de un usuario en la base de datos según sea el caso.



### CAPITULO III “ANÁLISIS Y DISEÑO DEL SISTEMA”

---

Nombre:	DatosImplicados(int id)
Descripción:	Devuelve el conjunto de datos implicados en la entidad Usuario.

En esta etapa del diseño, y estando ya elegida la tecnología a emplear para la implementación, en este caso .Net, se ha tomado la determinación de no describir las clases que físicamente tienen su equivalente como páginas Web, ya sean de cliente o servidor, siguiendo la misma metodología. Las clases (páginas) empleadas en el diseño no contienen lógica del negocio en su contenido dada su separación en la capa de presentación o de cliente, por este motivo solo se limitan a contener manipuladores para los eventos correspondientes y como atributos emplean las instancias de los componentes de Web Forms de .Net. La descripción se basará en los roles que juegan estas páginas dentro de la aplicación.

**Nombre:** GestionRecursos

**Descripción:**

Es la clase encargada de permitir al usuario autorizado la manipulación de los recursos, inserción, modificación y eliminación. Deberá permitir además la gestión de las ofertas realizadas para cada recurso. Es la encargada de generar una interfaz que permita manipular a la información en la forma más amigable posible dada la complejidad de la configuración de los tan heterogéneos recursos.

**Nombre:** GestionSolicitud

**Descripción:**

Es la clase encargada de permitir al usuario la gestión de sus solicitudes. Debe permitir la creación de una nueva solicitud, modificar una previamente realizada, consultar el estado de sus solicitudes y cancelar alguna no deseada. Debe mostrar, además, información que ayude al usuario a definir sus solicitudes, esto significa que para los recursos reservables que posean módulos de consulta de información externa, deberán ser consultados estos. Es la encargada de generar una interfaz que permita manipular a la información en la forma más amigable posible dado que esta será la página principal donde los usuarios podrán conformar sus solicitudes.

### CAPITULO III “ANÁLISIS Y DISEÑO DEL SISTEMA”

---

**Nombre:** RegistroUsuario

**Descripción:**

Es la clase encargada de permitir a los usuarios registrarse al sistema. Deberá realizar un registro del usuario autenticado para la confirmación de las páginas restringidas. Es la encargada de generar un sencillo formulario para la captura de los datos del usuario.

**Nombre:** Administración

**Descripción:**

Es la clase encargada de gestionar los usuarios que accederán al sistema con un nivel de acceso diferente al del usuario común. Deberá permitir la adición de un nuevo usuario, la edición de los datos de uno ya existente y la eliminación. Debe permitir además permitir la edición de los roles que estos usuarios jugarán dentro del sistema así como los niveles de acceso pertenecientes a estos. Es la encargada de generar una interfaz que permita manipular a la información de forma sencilla y rápida.

**Nombre:** ListadoOfertas

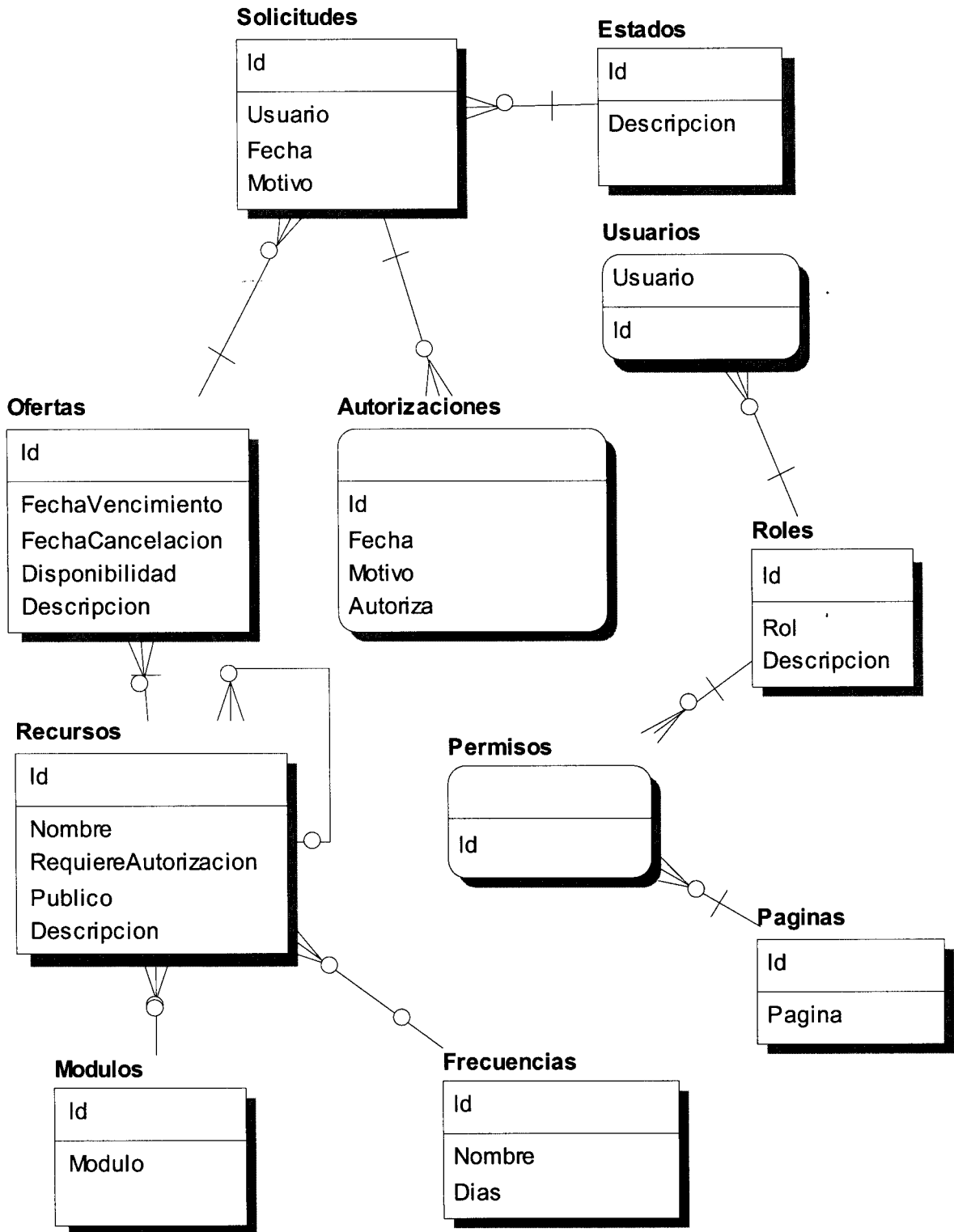
**Descripción:**

Es la clase encargada de mostrar a los usuarios un listado de las posibles reservaciones que podrán realizar. Deberá mostrar las opciones para los fines administrativos del sistema y proveerá el acceso a la autenticación de los usuarios.

### **3.7 Modelo de datos**

En el modelo de datos es necesario realizar algunas consideraciones a la hora de la implementación, es necesario conformar una serie de consultas para cada una de las entidades participantes en el modelo que jugarán un papel determinante para el funcionamiento de la aplicación, estas consultas comprenderán las responsabilidades básicas fundamentales de cada entidad con el fin de facilitar la implementación del lado del negocio y del cliente. Estas responsabilidades estarán encapsuladas en procedimientos almacenados empleando una nomenclatura específica en la forma <nombre de la tabla> más la acción que realiza, además se nombrarán las tablas con nombres que representen al objeto del mundo real cuyos datos almacenan, de esta forma solo es necesario conocer el nombre del objeto que queremos gestionar y adicionándole la responsabilidad de manejo de datos que nos interese tendremos el procedimiento que responderá a esa consulta específica. Otra consideración consiste en que cada tabla contendrá un campo Id que será auto incremental, es necesario este campo para lograr funcionalidades que agilicen el proceso de implementación del software.

3.7.1 Diagrama entidad-relación



## CAPITULO III “ANÁLISIS Y DISEÑO DEL SISTEMA”

### 3.7.2 Descripción de las tablas

<b>Nombre:</b> Solicitudes		
<b>Descripción:</b> Contiene los datos de las solicitudes realizadas.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id	bigint	Identificador de la solicitud.
Usuario	int	Identificador del usuario que realizó la solicitud.
Oferta	int	Identificador de la oferta solicitada.
Fecha	smalldatetim e	Fecha de realización de la solicitud.
Motivo	nvarchar	Motivo por el cual se realiza la solicitud
Estado	tinyint	Estado en el que se encuentra la solicitud.

<b>Nombre:</b> Recursos		
<b>Descripción:</b> Contiene los datos de los recursos reservables.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id	int	Identificador del recurso.
Nombre	nvarchar	Nombre del recurso.
Contenedor	int	Identificador del recurso padre en el que está contenido.
Modulo	int	Identificador del módulo que gestiona la información externa referente al recurso.
Frecuencia	int	Identificador de la frecuencia con que el recurso es republicado después del vencimiento.
RequiereAutorizacio n	bit	Indica si el recurso requiere autorización para su reserva.
Publico	bit	Indica si está disponible para ser reservado.
Descripcion	nvarchar	Descripción del recurso.

<b>Nombre:</b> Ofertas		
<b>Descripción:</b> Contiene de las ofertas realizadas a los recursos.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>

### CAPITULO III “ANÁLISIS Y DISEÑO DEL SISTEMA”

Id	int	Identificador de la oferta.
Recurso	int	Identificador del recurso ofertado.
FechaVencimiento	smalldatetim e	Fecha en que vence la posibilidad de reservar.
FechaCancelacion	smalldatetim e	Fecha en que vence la posibilidad de cancelar una reservación hecha.
Disponibilidad	int	Disponibilidad de la oferta hasta el momento.
Descripcion	nvarchar	Descripción de la oferta.

<b>Nombre:</b> Frecuencias		
<b>Descripción:</b> Contiene los datos de los tipos de frecuencias posibles.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id	int	Identificador de la frecuencia.
Nombre	nvarchar	Nombre de la frecuencia.
Dias	int	Número de días que comprende la frecuencia.

<b>Nombre:</b> Autorizaciones		
<b>Descripción:</b> Contiene los datos de las autorizaciones realizadas.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id	int	Identificador de la autorización.
Solicitud	int	Identificador de la Solicitud.
Fecha	smalldatetim e	Fecha en que fue autorizada la solicitud.
Motivo	nvarchar	Motivo por el que fue denegada la autorización.
Autoriza	int	Identificador de la persona que autoriza.

<b>Nombre:</b> Estados		
<b>Descripción:</b> Contiene los datos de los posibles estados.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id	int	Identificador del estado.

### CAPITULO III “ANÁLISIS Y DISEÑO DEL SISTEMA”

Descripción	nvarchar	Descripción del estado.
-------------	----------	-------------------------

<b>Nombre: Modulos</b>		
<b>Descripción:</b> Contiene los datos de los posibles módulos implementados.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id	int	Identificador del módulo.
Modulo	nvarchar	Nombre del módulo.

<b>Nombre: Usuarios</b>		
<b>Descripción:</b> Contiene la asociación de los usuarios con un rol específico.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id	int	Identificador de registro.
Usuario	int	Identificador del usuario.
Rol	int	Identificador del rol que juega el usuario.

<b>Nombre: Roles</b>		
<b>Descripción:</b> Contiene los datos de los posibles roles.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id	int	Identificador del rol.
Rol	nvarchar	Nombre del rol.
Descripción	Nvarchar	Descripción del rol.

<b>Nombre: Paginas</b>		
<b>Descripción:</b> Contiene los datos de las páginas que contiene el sitio.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id	int	Identificador de la página.
Pagina	nvarchar	Nombre de la página.

<b>Nombre: Permisos</b>		
<b>Descripción:</b> Contiene la asociación entre los roles y las páginas.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>

## CAPITULO III “ANÁLISIS Y DISEÑO DEL SISTEMA”

---

Id	int	Identificador del permiso.
Pagina	int	Identificador de la página que se asocia.
Rol	int	Identificador del rol que se asocia.

### **3.8 Tratamiento de errores**

El tratamiento de errores está referido a los errores que pueden ocurrir tanto durante la interacción del usuario con el sistema a la hora de introducir datos como a los errores y excepciones que pudieran generarse a la hora de correr la aplicación como producto de su funcionamiento e interacción con sistemas y mecanismos externos.

Debe tratarse de evitar la posibilidad de ocurrencia de errores por la parte del cliente por lo que los formularios, siempre que sea posible, mostrarán la información en forma de listas para que el cliente elija y disminuir de esta forma la posibilidad de errores de teclado. En los casos en que no sea posible realizar esta concesión los campos que implican datos con formatos específicos así como los valores numéricos deberán estar completamente validados, informándole al usuario la ocurrencia del error e imposibilitar la entrada de los datos al sistema para ser procesados.

Por otra parte la ocurrencia de excepciones a nivel del sistema es algo que deberá reducirse al mínimo, pero no es posible eliminar absolutamente la posibilidad de ocurrencia de estas, más aun en un sistema que trabajará dependiente de otros y realizando accesos a bases de datos externas. En este caso es necesario minimizar las consecuencias de estos realizando una captura de excepciones desde los niveles más bajos de la implementación con el fin de poder procesarlas y poder brindarle una información lo suficientemente explícita al usuario que le permita recuperarse del fallo sin interrumpir la ejecución de la aplicación.



### **3.9 Concepción de la ayuda**

El sistema mostrará una interfaz Web amigable que sea capaz de dirigir al usuario durante todo el proceso de solicitud de una reservación de forma tal que la utilización del sistema se convierta en algo cotidiano y sencillo y no en una tarea pesada. En las secciones dedicadas a la administración es donde se deberá hacer énfasis pues en estas se realizan funciones más complejas como la gestión de los recursos y sus ofertas, tarea de la cual depende el correcto funcionamiento y la aceptación que pueda llegar a tener el sistema. Dada la heterogeneidad de los posibles recursos se hizo necesario confeccionar un mecanismo para permitir gestionar los recursos en forma que toda la gama de particularidades se mantenga en la mayor manera posible para no perder riqueza en la información que se le oferte al usuario. De esta manera se puede hacer algo compleja la manipulación de los recursos teniendo poco que ver, por ejemplo un pase con un salón de conferencias de aquí que sea necesario lograr un formulario que en forma clara guíe al usuario autorizado durante el proceso de gestión de un recurso. La inclusión de una ayuda documentada no está descartada, más, se incluirá en este trabajo como una recomendación a implementar.

### **Conclusiones**

En este capítulo se abordó el análisis y el diseño del sistema a implementar, fue desarrollado el modelo de clases del análisis y se confeccionaron los diagramas de interacción como evolución de la expansión de los casos de uso. Se obtuvo el diagrama de clases del diseño de la aplicación y se describieron las clases teniendo en cuenta sus atributos y responsabilidades, como concesión a las particularidades de la aplicación a implementar y la tecnología elegida se desarrollaron los diagramas de diseño Web y se realizó la descripción de las páginas en función de sus roles específicos.

Se obtuvo el diagrama entidad relación y se realizó la descripción de las tablas teniendo en cuenta la información que contiene y los campos que contiene.

### **CAPITULO III “ANÁLISIS Y DISEÑO DEL SISTEMA”**

---

En este capítulo se realizaron además algunas consideraciones acerca del tratamiento de errores y el manejo de excepciones con el fin de lograr una aplicación robusta.

## CONCLUSIONES GENERALES

---

### **Conclusiones generales**

Durante el desarrollo de este trabajo se realizaron estudios demuestran la necesidad de implantar un sistema de reservaciones configurable, robusto y capaz de adaptarse a las crecientes necesidades de la UCI. Para llegar a esto se realizó un profundo análisis y diseño de aspectos trascendentales con la finalidad de lograr un modelo diferente al clásico sistema aislado e independiente no aplicable al centro.

Se logró un innovador mecanismo de reservaciones capaz de gestionar el proceso de reserva de todos los recursos y servicios que necesiten ser accedidos de forma planificada y controlada.

El diseño multi-capas permite la integración a los sistemas existentes en la UCI logrando un elevado nivel de escalabilidad facilitando la aplicación de servicios Web que proveerían de información a aquellos sistemas que lo requieran.

Al culminar este trabajo se cumple con el objetivo fundamental trazado, diseñar e implementar un sistema integrador de reservaciones que satisfaga las necesidades de la UCI.

## **RECOMENDACIONES**

---

### **Recomendaciones**

- Realizar el análisis y el diseño del sistema para los casos de uso que solo se definieron sin expandirlos.
- Realizar el diseño del sistema para la posibilidad de no emplear procedimientos almacenados. Se sugiere implementar un contenedor XML para las consultas SQL siguiendo la misma filosofía de estandarizar las consultas.
- Confeccionar una ayuda documental que sirva de guía al usuario autorizado a la hora de gestionar los recursos pues de ello depende el correcto funcionamiento del sistema.

## **BIBLIOGRAFÍA CONSULTADA**

---

### **Bibliografía Consultada**

1. Fowler, Martín. "UML Gota a Gota". Primera Edición. Addison Wesley Longman. 1999.
2. Larman, Craig. *UML y Patrones, Introducción al análisis y diseño orientado a objetos*. Prentice-Hall, 2002.
3. Ceria, Santiago. *Ingeniería de Software I. Casos de Uso. Un Método Práctico para Explorar Requerimientos*.
4. Desarrollo basado en RUP bajo la herramienta Rational Rose  
<http://lml.ls.fi.upm.es/mdp/si/> (4/4/2004)
5. Universidad .NET <http://www.microsoft.com/spanish/msdn/comunidad/uni.net/>  
(4/4/2004)
6. Curso práctico de desarrollo de aplicaciones con Visual Studio .NET  
<http://www.microsoft.com/spanish/msdn/comunidad/uni.net/> (4/4/2004)
7. Villariño, Luis. *UML para Web*. 2002.
8. Booch, G., Rumbaugh, J., Jacobson, I. *El Lenguaje Unificado de Modelado*. Addison-Wesley. 1999.
9. Booch, G., Rumbaugh, J., Jacobson, I. *The Unified Software Development Process*. Addison-Wesley. 1999.
10. Cockburn, A. *Using Goal-Based Use Cases*. JOOP, 1997
11. García Molina, J. *Towards Use Case and Conceptual Models through Business Modeling*. Conference on Conceptual Modelling. 2000
12. Larman, C. *Applying UML and Patterns. An Introduction to Object-Oriented Analysis and Design*. Prentice-Hall, 1998.
13. Gamma, E. *Design Patterns, Elements of Reusable Object-Oriented Software*. Addison-Wesley. 1995.
14. Meyer, B. *Construcción de software orientado a objetos*. Prentice-Hall. 1998.
15. Riel, A. *Object-Oriented Design Heuristics*. Addison-Wesley. 1996.
16. Jim Conallen. *Building Web Applications with UML*. Addison-Wesley. 1999.

## GLOSARIO DE TÉRMINOS

---

### Glosario de términos

#### SIGLAS

- CPU (Unidad Central de Proceso)
- DHTML (*Dynamic Hypertext Markup Language*)
- HTML (*Hypertext Markup Language*)
- UCI (Universidad de las ciencias informáticas)
- Open Source (Código abierto)
- PHP (Personal Home Page).
- RUP: Rational Unified Process (Proceso Unificado de Rational).
- SQL (*Structure Query Language*)
- UML (*Unified Modeling Language*)

#### Términos

##### **B:**

##### **Base de datos**

Una base de datos de consta de una colección de tablas que contienen datos y otros objetos, como vistas, índices, procedimientos almacenados y desencadenadores, que se definen para poder llevar a cabo distintas operaciones con datos.

##### **C:**

##### **Cliente-Servidor**

Cuando se menciona este término no se piensa en otra cosa más que en bases de datos, dado que generalmente (Y de manera incorrecta) este término se usa como sinónimo de esto. Este término, en su más amplia definición, se usa para describir una aplicación en la cual dos o más procesos separados trabajan juntos para completar una tarea. El proceso cliente solicita al proceso servidor la ejecución de alguna acción en particular. Esta operación se conoce como Proceso Cooperativo, dado que dos procesos separados cooperan para completar la tarea en particular.

## **GLOSARIO DE TÉRMINOS**

---

### **H:**

#### **Hardware**

Componentes electrónicos, tarjetas, periféricos y equipo que conforman un sistema de computación; se distinguen de los programas (software) porque éstos le indican a los componentes mencionados lo que deben hacer.

#### **HTML (*Hypertext Markup Language*)**

Lenguaje usado para escribir documentos para servidores World Wide Web. Es una aplicación de la ISO Standard 8879:1986.

### **I:**

#### **Interface**

Conexión entre dos dispositivos de hardware, entre dos aplicaciones o entre un usuario y una aplicación que facilita el intercambio de datos.

#### **Internet**

Sistema de redes de computación ligadas entre sí, con alcance mundial, que facilita servicios de comunicación de datos como registro remoto, transferencia de archivos, correo electrónico y grupos de noticias. Internet es una forma de conectar las redes de computación existentes que amplía en gran medida el alcance de cada sistema participante.

#### **Intranet**

Una red interna, conectando a múltiples usuarios a través de la tecnología de Internet, usualmente dentro de una organización. Una Intranet es típicamente usada para incrementar el acceso a la información, comunicación, y trabajo entre usuarios individuales y la organización.

## GLOSARIO DE TÉRMINOS

---

**J:**

### **JSCRIPT (JavaScript)**

Otra importante aplicación desarrollada por Netscape, se trata de programas muy parecidos a los de Java, la diferencia estriba en que los Javascripts se encuentran incorporados dentro del HTML. Un lenguaje scripting basado en Java que le permite al web browser correr simples programas en la computadora del usuario. JavaScript es usado a menudo para la validación de formularios.

**M:**

### **Microsoft**

Compañía que manufactura los sistemas de operación DOS y Windows. Compañía de software más grande del mundo. Fue fundada en 1975 por Paul Allen y Bill Gates. Aunque también se conoce por sus lenguajes de programación y aplicaciones para computadores personales, el éxito sobresaliente de Microsoft se debe a sus sistemas operativos DOS y Windows.

### **Microsoft SQL Server**

Un servidor de Structured Query Language (SQL), SQL Server es usado para interactuar con una base de datos. Recientemente, ha sido ampliamente usado en las aplicaciones web.

**O:**

### **Open Source(Código Abierto)**

En el mundo del Software Libre existen, por decirlo de alguna manera, dos “políticas” o corrientes sobre cuyas definiciones de licenciamiento están englobadas la mayoría de aplicaciones de Código Abierto. Estas dos corrientes son: Free Software Foundation con la licencia GNU/GPL y la Open Source Initiative con la licencia OpenSource.

En esencia los criterios de ambas posiciones son muy similares en lo que respecta a la posesión del código fuente, la libertad de uso, copia, modificación y distribución. Sin embargo es bueno conocer que la licencia de OpenSource incorpora definiciones que la hacen más aceptable para la incorporación de empresas al desarrollo de Software Libre.



## GLOSARIO DE TÉRMINOS

---

**P:**

### **PHP**

Es un ambiente *script* del lado del servidor que permite crear y ejecutar aplicaciones Web dinámicas e interactivas. Con PHP se pueden combinar páginas HTML y *scripts*. Con el objetivo de crear aplicaciones potentes. Se caracterizan por ser tecnología (open source), y por su fácil desarrollo y mantenimiento. Existe mucha documentación en Internet.

### **Proceso**

Un proceso puede ser definido como un conjunto de actividades interrelacionadas entre sí que, a partir de una o varias entradas de materiales o información, dan lugar a una o varias salidas también de materiales o información con valor añadido.

**R:**

### **Red**

Una red es dos o más computadoras que están físicamente conectadas con las otras y capaces de compartir información

### **Reglas del CLS**

Common Language Specification. Conjunto de reglas que han de seguir las definiciones de tipos que se hagan usando un determinado lenguaje gestionado si se desea que sean accesibles desde cualquier otro lenguaje gestionado.

**S:**

### **SDK de Windows**

También denominado API de Windows

Librería de rutinas y funciones (SDK - Kit de desarrollo de software) que permiten gestionar componentes como menús, diálogos, ventanas, etc. Proveída por Windows.

## **GLOSARIO DE TÉRMINOS**

---

### **Scripts**

En inglés significa "guión". De hecho, el uso es exactamente éste: el navegador lee una línea, la interpreta y la ejecuta, después pasa a la sucesiva y hace lo mismo, y así hasta el cierre del script.

### **Sistema**

Conjunto de cosas que ordenadamente relacionadas entre sí contribuyen a determinado objeto.

### **Sitio Web**

Sistema de computación que corre un servidor Web y que se ha establecido para editar documentos en Web.

### **Sistema distribuido**

Un sistema distribuido es aquel en el que dos o más máquinas colaboran para la obtención de un resultado. En todo sistema distribuido se establecen una o varias comunicaciones siguiendo un protocolo prefijado mediante un esquema cliente-servidor.

### **SOAP (Simple Object Access Protocol)**

Una aplicación XML que emplea .NET como protocolo de comunicación.

### **Software**

Programas de sistema, utilerías o aplicaciones expresados en un lenguaje de máquina.

### **U:**

### **UML**

Unified Modeling Language. Es una notación standard para modelar objetos del mundo real como primer paso en el desarrollo de programas orientados a objetos.

## **GLOSARIO DE TÉRMINOS**

---

### **W:**

#### **WEB (WWW)**

Red de documentos HTML intercomunicados y distribuidos entre servidores del mundo entero.

#### **Web browser**

Una aplicación que localiza un documento en Internet usando un URL (Uniform Resource Locator), lo localiza, y visualiza el documento. Netscape Navigator, Spyglass Mosaic, y Microsoft Internet Explorer son ejemplos de Web browsers.

#### **Windows**

Actualmente el sistema operativo más popularizado del planeta desarrollado por la empresa Microsoft, el cual funciona a través de ventanas (windows) desplegadas. Es el nombre del popular entorno (no es un sistema operativo y no es una aplicación) software creado por Microsoft.

### **X:**

#### **XML**

Extensible Markup Language. Lenguaje orientado al almacenamiento, procesamiento y transmisión de mensajes.

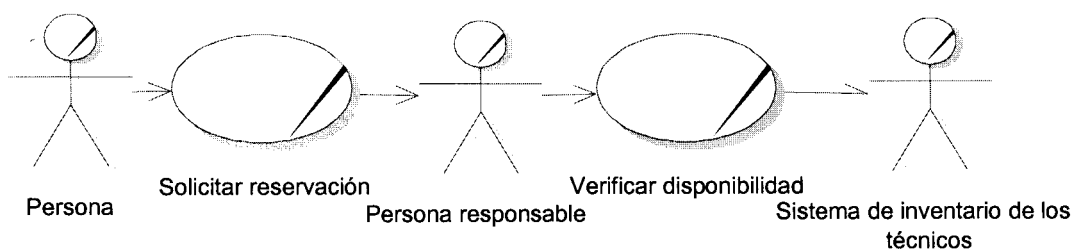
**Anexo 1. Modelo del negocio**

**Actores que participan en el proceso del negocio**

En este proceso de negocio los actores que participan en función de sus roles específicos son:

Actores del negocio	Justificación
Persona	Inicia los procesos y es quien se beneficia con los resultados.
Persona responsable	Es la encargada de tomar determinaciones respecto a los casos de uso y intercambia información con ellos a la hora de decepcionar las solicitudes.
Dpto. Relaciones internacionales	Es el encargado de controlar el proceso de gestión de los viajes nacionales.
Sistema de inventario de los técnicos	Nutre al proceso con la información de los medios materiales con que cuentan los técnicos (laboratorios, PC, ETC)
Sistema de reservación de pases	Es actualmente el encargado de gestionar el proceso de reservación de los pases.
Casona	Es quien organiza las actividades recreativas y excursiones.

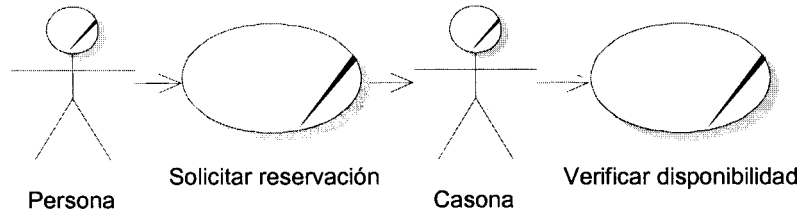
**Diagramas de casos de uso del negocio**



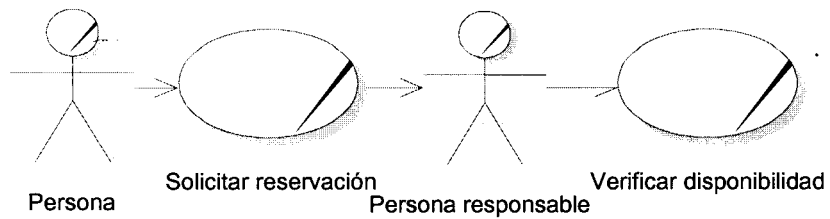
DCU Reservación de tiempos de máquina

## ANEXO I "MODELO DEL NEGOCIO"

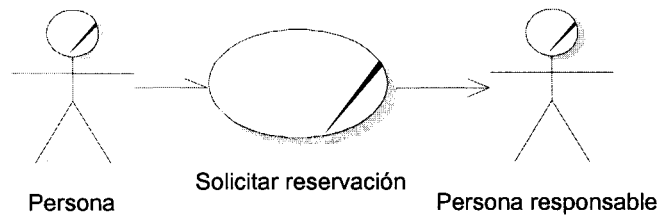
---



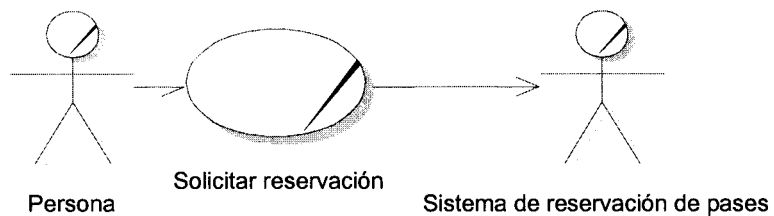
### DCU Reservación de excursiones



### DCU Reservación de locales



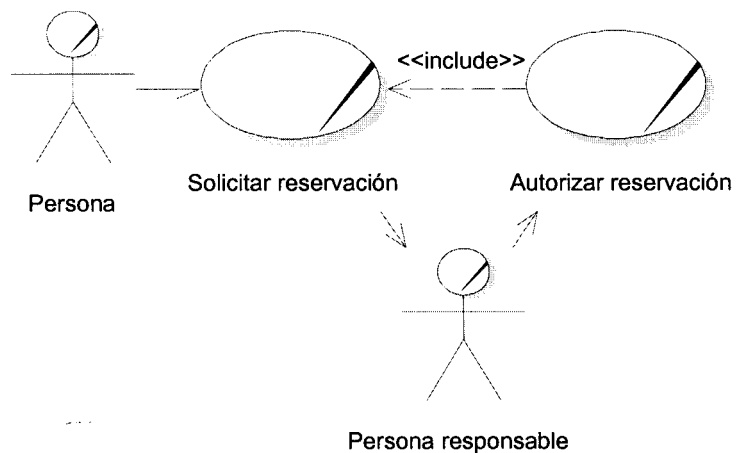
### DCU Reservación de pases de domingo



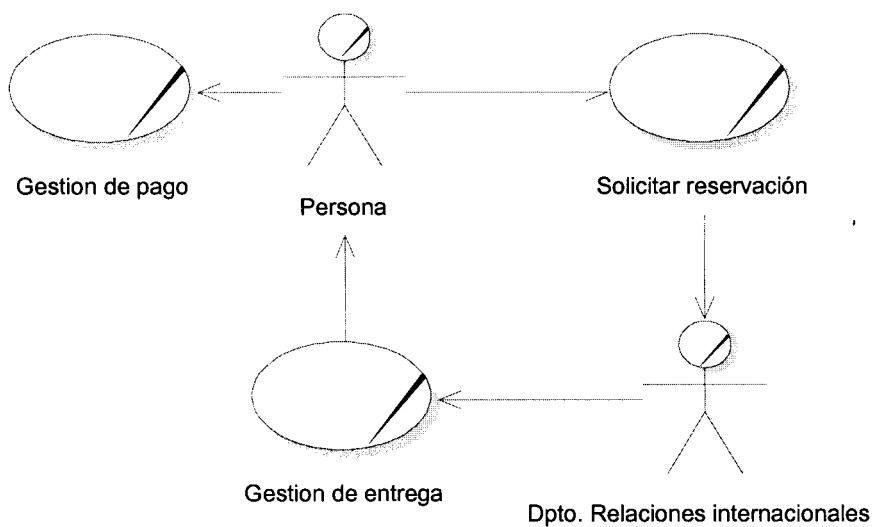
### DCU Reservación de pases de fin de semana

## ANEXO I "MODELO DEL NEGOCIO"

---



### DCU Reservación de pases diarios



### DCU Reservación de viajes nacionales

## ANEXO I “MODELO DEL NEGOCIO”

---

### Descripción de los casos de uso

<b>Caso de Uso</b>	<b>Autorizar reservación</b>
<b>Actores</b>	Persona responsable
<b>Descripción:</b>	La solicitud hecha por la persona implicada requiere ser autorizada y la persona responsable para este fin considera la justificación y determina si autoriza o no la solicitud.
<b>Referencias</b>	

<b>Caso de Uso</b>	<b>Solicitar reservación</b>
<b>Actores</b>	Persona, Persona responsable, Sistema de Reservación de pase, Dpto. de Relaciones Internacionales.
<b>Descripción:</b>	La persona establece la solicitud de la reservación a la persona responsable para este fin o al sistema implicado tal como sea el caso, la solicitud es procesada.
<b>Referencias</b>	

<b>Caso de Uso</b>	<b>Verificar disponibilidad</b>
<b>Actores</b>	Persona responsable, Sistema de inventario de los técnicos.
<b>Descripción:</b>	Al establecerse una solicitud, la persona responsable de esta actividad verificará la disponibilidad del recurso o servicio solicitado para decidir si puede o no realizar la reservación.
<b>Referencias</b>	

<b>Caso de Uso</b>	<b>Gestión de entrega</b>
<b>Actores</b>	Dpto. de Relaciones Internacionales.
<b>Descripción:</b>	Después de que una reservación es gestionada por el Dpto. de Relaciones Internacionales, el pasaje obtenido mediante esta gestión es hecho llegar al interesado.
<b>Referencias</b>	

<b>Caso de Uso</b>	<b>Gestión de pago</b>
<b>Actores</b>	Persona

## ANEXO I “MODELO DEL NEGOCIO”

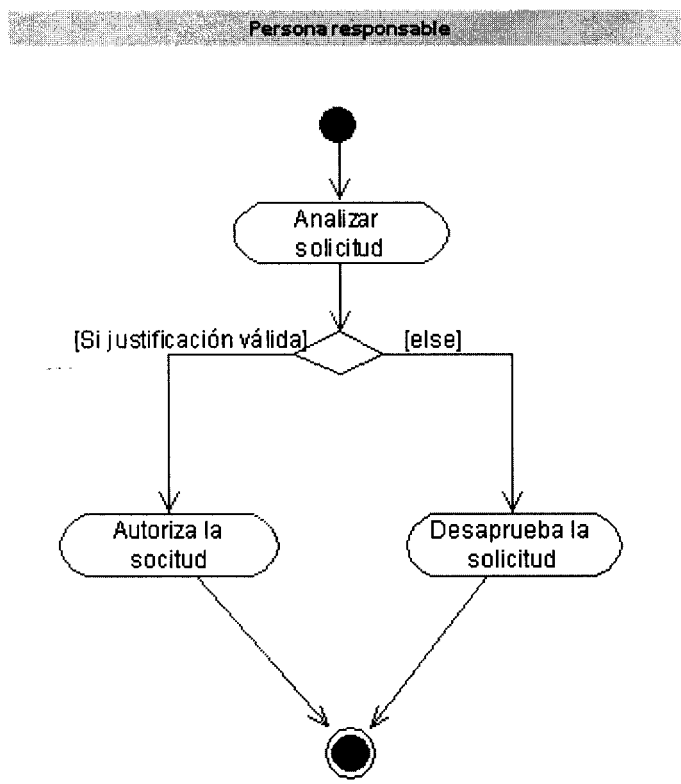
---

<b>Descripción:</b>
Una persona al realizar una solicitud la hace oficial mediante el modelo de solicitud de pago en donde remunera el monto del pasaje que solicita.
<b>Referencias</b>

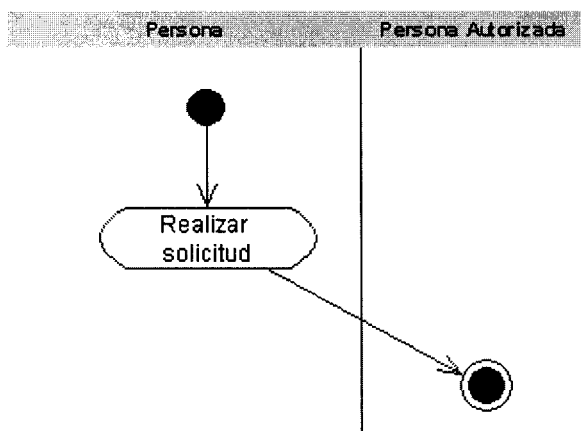


## ANEXO I “MODELO DEL NEGOCIO”

### Diagramas de actividades del negocio

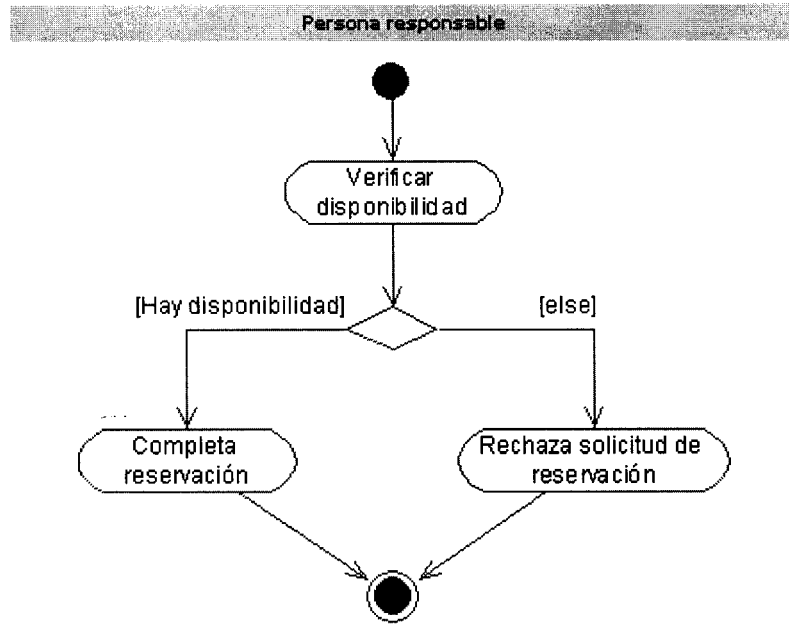


### DAN Autorizar reservación

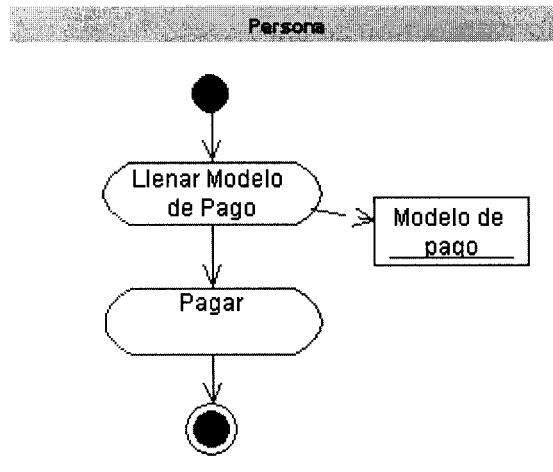


### DAN Solicitar reservación

# ANEXO I “MODELO DEL NEGOCIO”



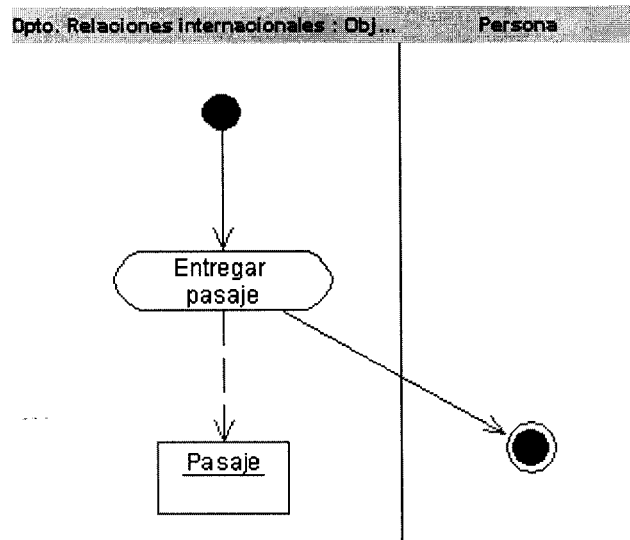
DAN Consultar disponibilidad



DAN Gestión de pago

## ANEXO I "MODELO DEL NEGOCIO"

---



DAN Gestión de entrega

## **ANEXO II. DEFINICIÓN DE LOS CASOS DE USO DEL SISTEMA**

---

### **Anexo 2. Definición de los casos de uso del sistema**

#### **Descripción de los actores**

<b>Nombre del actor</b>	<b>Descripción</b>
Usuario	Son todos los usuarios del sistema que solicitan reservaciones y consultan información.
Usuario autorizado	Es el usuario que tiene privilegios para autorizar una reservación, y consultar información para hacer análisis estadísticos.
Sistema UCI Personas	Es el sistema donde se guarda la información de las personas.
Sistema Dominio.	Es el sistema donde se encuentran todos los usuarios del dominio.
Sistema Correo Postal	El sistema con el que se interactúa a la hora de la gestión de entrega de la reservación.
Administrador	Es el encargado de realizar las actualizaciones del sistema, declarar los niveles de usuarios y los permisos para cada uno.
Sistema UCI Horarios	Es el sistema que se encargará de gestionar y generar los horarios docentes.
Sistema UCI Transporte	Es el sistema encargado de gestionar Transporte en la UCI.
Sistema UCI Inventarios	Es el sistema encargado de llevar la gestión de los inmuebles y muebles.

#### **Descripción de los casos de uso del sistema**

<b>CU-1</b>	<b>Autenticar usuario</b>
<b>Actores</b>	Usuario, Sistema Dominio, Sistema UCI Personas.
<b>Descripción:</b>  El usuario introduce sus datos al sistema en la forma de usuario y contraseña, donde	

## ANEXO II. DEFINICIÓN DE LOS CASOS DE USO DEL SISTEMA

---

se valida la entrada, y en dependencia del tipo de usuario el sistema le asigna su nivel de acceso.

**Referencias**

**CU-2 Solicitar reservación.**

**Actores** Usuario

**Descripción:**

El usuario hace una solicitud de reservación que genera un registro que el usuario puede modificar para poder tramitar la reservación, para todos los recursos a reservar es necesario activar el caso de uso Consultar Disponibilidad. Existen ciertos tipos de recursos cuya reservación requiere autorización, en este caso se activa el caso de uso Autorizar Reservación, otros requieren de pago, en este caso el implicado es el caso Gestión de Pago.

**Referencias**

**CU-3 Cancelar solicitud.**

**Actores** Usuario.

**Descripción:**

El usuario solicita cancelar una o varias reservaciones previamente realizadas por el.

**Referencias**

**CU-4 Autorizar Reservación**

**Actores** Usuario autorizado

**Descripción:**

El usuario autorizado autoriza o rechaza las reservaciones que están en espera de autorización.

**Referencias**

**CU-5 Administrar usuarios.**

**Actores** Administrador.

## ANEXO II. DEFINICIÓN DE LOS CASOS DE USO DEL SISTEMA

---

<b>Descripción:</b> El administrador del sistema define los niveles de usuarios, crea y modifica los usuarios que están en cada uno de estos niveles.	
<b>Referencias</b>	

<b>CU-6</b>	<b>Gestión de entrega.</b>
<b>Actores</b>	Usuario, Sistema Correo Postal
<b>Descripción:</b> En dependencia del tipo de reservación el sistema genera los ticket que serán entregados personalmente o envía al sistema de correo postal una relación de modelos que deberán ser entregados por medio de este servicio.	
<b>Referencias</b>	

<b>CU-7</b>	<b>Consultar información</b>
<b>Actores</b>	Usuario, Usuario autorizado, Administrador
<b>Descripción:</b> Cuando el usuario hace una petición de servicio al sistema se le brinda información sobre los recursos disponibles, el estado de las reservaciones solicitadas por él, la disponibilidad de los recursos a reservar además de informaciones estadísticas de interés. Este caso siempre incluye las acciones del caso de uso Consultar Disponibilidad.	
<b>Referencias</b>	

<b>CU-8</b>	<b>Gestión de pago.</b>
<b>Actores</b>	
<b>Descripción:</b> Dada una serie de opciones de pago el usuario debe decidir la forma en que pagará la reservación en el caso de requerirlo.	
<b>Referencias</b>	

## ANEXO II. DEFINICIÓN DE LOS CASOS DE USO DEL SISTEMA

---

<b>CU-9</b>	<b>Consultar información externa.</b>
<b>Actores</b>	
<b>Descripción:</b>	Ante una solicitud de reservación el sistema muestra en la información de los recursos la disponibilidad de estos según como fueron definidos, además el sistema debe consultar esta información para su funcionamiento.
<b>Referencias</b>	

<b>CU-10</b>	<b>Consultar información transporte.</b>
<b>Actores</b>	Sistema UCI Transporte
<b>Descripción:</b>	Esta especialización de Consultar Disponibilidad verifica en el Sistema de Transporte la disponibilidad de este recurso. Este caso es una especialización del caso de uso Consultar Disponibilidad.
<b>Referencias</b>	

<b>CU-11</b>	<b>Consultar información inventarios.</b>
<b>Actores</b>	Sistema UCI Inventarios, Sistema UCI Horarios.
<b>Descripción:</b>	Verifica en el Sistema de Inventarios la disponibilidad de los locales, salones y laboratorios así como los horarios de disponibilidad de estos. Este caso es una especialización del caso de uso Consultar Disponibilidad.
<b>Referencias</b>	

<b>CU-12</b>	<b>Gestionar recursos.</b>
<b>Actores</b>	Usuario autorizado
<b>Descripción:</b>	Permite adicionar nuevos recursos reservables al sistema, así como eliminar o modificar los ya existentes.
<b>Referencias</b>	

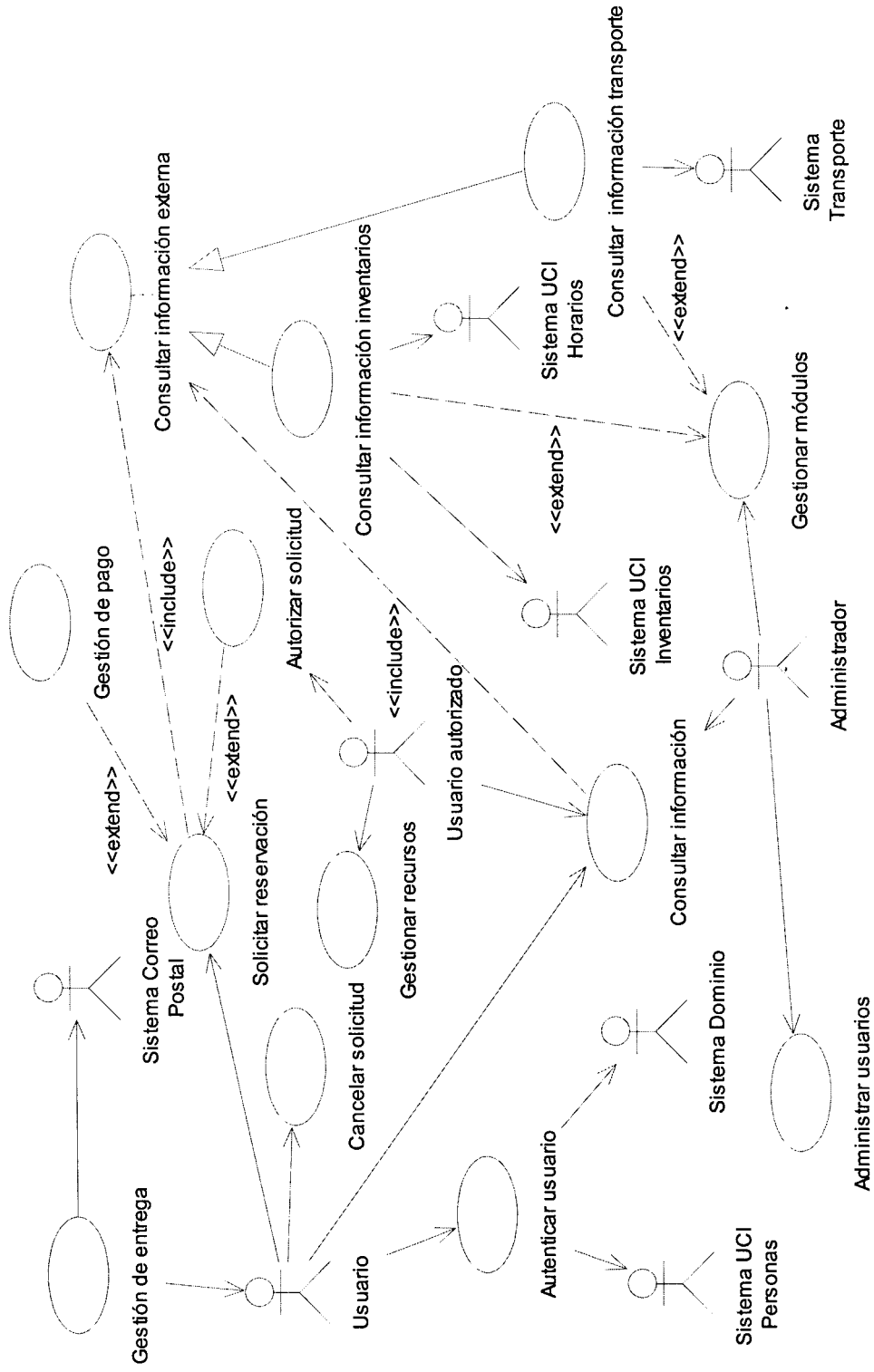
## **ANEXO II. DEFINICIÓN DE LOS CASOS DE USO DEL SISTEMA**

---

<b>CU-13</b>	<b>Gestionar módulos.</b>
<b>Actores</b>	Administrador
<b>Descripción:</b>	Permite manipular módulos de consulta de información externa, se encarga además de la instalación, modificación y enlace con los elementos que lo necesiten.
<b>Referencias</b>	



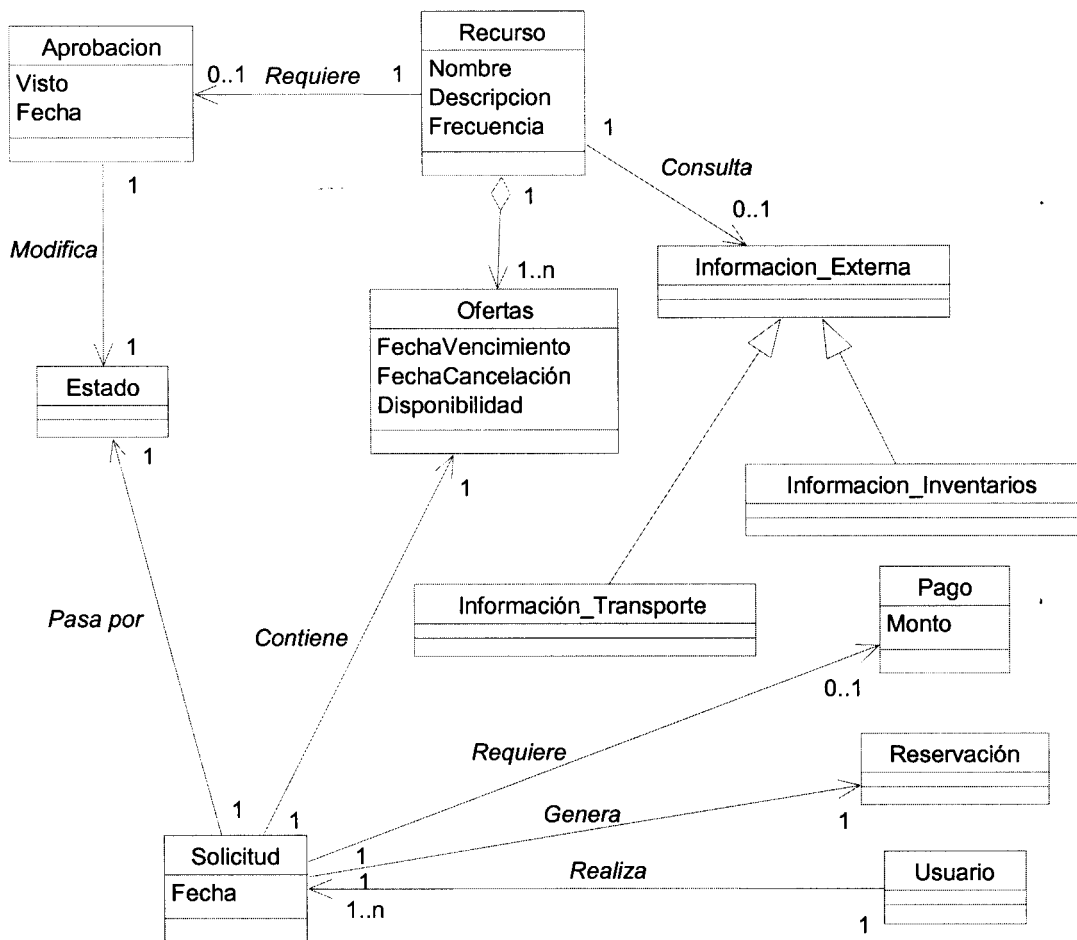
**Diagrama de casos de uso del sistema**



## ANEXO II. DEFINICIÓN DE LOS CASOS DE USO DEL SISTEMA

### Modelo del dominio

Con el fin de lograr una mayor claridad al la hora de entender el dominio del problema se propone un diagrama de dominio.



Modelo conceptual del dominio

## ANEXO II. DEFINICIÓN DE LOS CASOS DE USO DEL SISTEMA

---

### Casos de uso por ciclo de desarrollo

Ciclo	Nombre de caso de uso	Justificación de la selección
1	Solicitar reservación	Es el caso de uso básico, contiene la esencia del funcionamiento del sistema.
	Cancelar reservación	Es un caso fundamental, su no implementación implica que el sistema no satisfaga completamente los requerimientos.
	Gestionar recursos	Es el caso de uso más complejo, es el que permite el manejo de los recursos reservables que serán puestos a consideración de los usuarios.
2	Autenticar usuario	Es un caso de uso que no interviene directamente en el funcionamiento del sistema pero es fundamental para garantizar el control de los accesos.
	Administrar usuarios	Este caso se encarga de distribuir los niveles de acceso según los roles que jueguen los usuarios dentro del sistema.

### Casos de uso expandidos

<b>CU-1</b>	<b>Autenticar usuario</b>
<b>Actores</b>	Usuario, Sistema Dominio, Sistema UCI Personas.
<b>Descripción:</b> El usuario introduce sus datos al sistema en la forma de usuario y contraseña, donde se valida la entrada, y en dependencia del tipo de usuario el sistema le asigna su nivel de acceso.	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El usuario solicita acceder a una página restringida.	2. La página detecta que el usuario no está registrado y lo redirige a la página de autenticación y registro.

## ANEXO II. DEFINICIÓN DE LOS CASOS DE USO DEL SISTEMA

<p>3. El usuario se registra para entrar al sistema.</p>	<p>4. El sistema recepciona los datos (usuario, contraseña)</p> <p>5. Verifica la autenticación con el directorio activo.</p> <p>7. El sistema verifica que tipo de usuario con el Sistema UCI Personas.</p> <p>8. Se comprueba en la base de datos del sistema y genera los niveles de acceso para el.</p>
--	---

<b>CU-2</b>	<b>Solicitar reservación.</b>	
<b>Actores</b>	Usuario	
<p><b>Descripción:</b></p> <p>El usuario hace una solicitud de reservación que genera un registro que el usuario puede modificar para poder tramitar la reservación, para todos los recursos a reservar es necesario activar el caso de uso Consultar Disponibilidad. Existen ciertos tipos de recursos cuya reservación requiere autorización, en este caso se activa el caso de uso Autorizar Reservación, otros requieren de pago, en este caso el implicado es el caso Gestión de Pago.</p>		
<b>Acción del actor</b>	<b>Respuesta del sistema</b>	

## ANEXO II. DEFINICIÓN DE LOS CASOS DE USO DEL SISTEMA

<p>1. El usuario selecciona la opción de realizar una solicitud de reservación.</p>	<p>2. El sistema verifica que el usuario tenga acceso a la página solicitada</p> <p>3. El sistema carga la información necesaria para conformar una solicitud</p> <p>4. El sistema verifica, en caso de requerirlo, la información externa del recurso según el módulo asociado a este.</p> <p>5. El sistema muestra un formulario para completar la solicitud.</p> <p>6. El sistema verifica el cumplimiento de las políticas determinadas para el recurso en particular (tiempo establecido, disponibilidad, etc.).</p>
<p>7. El usuario completa el llenado de la solicitud.</p>	<p>8. En caso de seleccionar un recurso que requiere autorización el sistema lo marca como pendiente para ser autorizado.</p> <p>9. El sistema registra la solicitud.</p>
<p><b>Sección 1- Modificar solicitud</b></p>	
<p>1. El usuario selecciona modificar la solicitud.</p>	<p>2. El sistema le brinda la posibilidad de modificar los datos.</p>
<p>3. El usuario introduce al sistema los nuevos datos.</p>	<p>4. El sistema registra los nuevos datos entrados.</p>

<b>CU-3</b>	<b>Cancelar solicitud.</b>
<b>Actores</b>	Usuario.
<p><b>Descripción:</b> El usuario solicita cancelar una o varias solicitudes previamente realizadas por el.</p>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>

## ANEXO II. DEFINICIÓN DE LOS CASOS DE USO DEL SISTEMA

1. El usuario selecciona cancelar una solicitud.	2. El sistema verifica que el usuario tenga acceso a la página solicitada 3. El sistema le muestra un listado de las solicitudes previamente realizadas por el usuario.
3. El usuario selecciona las solicitudes que desea cancelar.	5. El sistema verifica el cumplimiento de las políticas definidas para el recurso seleccionado. 6. El sistema elimina las solicitudes escogidas. 7. Se actualizan las disponibilidades de los recursos afectados.

<b>CU-5</b>	<b>Administrar usuarios.</b>	
<b>Actores</b>	Administrador.	
<b>Descripción:</b> El administrador del sistema define los niveles de usuarios, crea y modifica los usuarios que están en cada uno de estos niveles.		
<b>Acción del actor</b>		<b>Respuesta del sistema</b>
1. El administrador selecciona la opción de administrar usuarios.		2. El sistema verifica que el usuario tenga acceso a la página solicitada 3. El sistema carga un formulario para el fin.
4. El usuario modifica o adiciona la información pertinente y elije guardar.		5. El sistema actualiza los cambios realizados en la base de datos.

<b>CU-12</b>	<b>Gestionar recursos.</b>	
<b>Actores</b>	Usuario autorizado	

## ANEXO II. DEFINICIÓN DE LOS CASOS DE USO DEL SISTEMA

<b>Descripción:</b>	
Permite adicionar nuevos recursos reservables al sistema, así como eliminar o modificar los ya existentes.	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El usuario autorizado selecciona adicionar un nuevo recurso a reservar.	2. El sistema verifica que el usuario tenga acceso a la página solicitada 3. Se cargan los datos para conformar un recurso. 4. El sistema muestra el formulario para este fin.
5. El usuario autorizado introduce la información pertinente para cada recurso y las ofertas que van a ser publicadas.	6. El sistema registra las nuevas ofertas. 7. El sistema registra los datos del nuevo recurso verificando que haya correspondencia entre el recurso y sus ofertas.
<b>Sección 1- Modificar Recurso</b>	
1. El usuario autorizado selecciona modificar un recurso reservable.	2. Se muestra un listado con los recursos existentes.
3. El usuario autorizado selecciona un recurso.	4. El sistema carga un formulario con los datos del recurso seleccionado.
5. El usuario autorizado modifica el recurso y selecciona actualizar los cambios.	6. El sistema verifica los requerimientos de los datos introducidos y actualiza la base de datos.
<b>Sección 1- Eliminar Recurso</b>	
1. El usuario autorizado selecciona eliminar un recurso reservable.	2. Se muestra un listado con los recursos existentes.
3. El usuario autorizado selecciona un recurso.	4. El sistema elimina las ofertas asociadas al recurso de la base de datos. 5. El sistema elimina el recurso de la base de datos.