

003.7

PUE

<sup>S</sup>  
TD-0011-04-01

TD-0011-04-01

**Universidad de Oriente**  
**Facultad de Matemática y computación**  
**Departamento de computación**



# **Sistema de Gestión De alojamiento**

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**INFORME TÉCNICO PARA OPTAR POR EL TÍTULO DE LICENCIADO EN CIENCIA DE LA  
COMPUTACIÓN.**

**Autor: Manuel Enrique Puebla Martínez**  
**Tutor: Ing. Ariadna Falcón López.**

**Ciudad de la Habana, Junio del 2004**

## **RESUMEN**

El centro actual del trabajo de la educación superior cubana está en consolidar la interrelación entre la formación de profesionales y la vida económica, política y social del país, a partir del concepto de integración de la docencia con la producción y la investigación en su concepción más amplia y más rica, que es aquella que revela que sus productos finales son competitivos.

En la educación superior cubana se entiende que sólo con un proceso docente que sea capaz de garantizar esta integración se podrán formar profesionales para afrontar los retos de un presente complejo y de un futuro de inserción en la competencia a nivel mundial.

El presente trabajo brinda una propuesta de aplicación Web para mejorar el proceso de distribución del personal en la Universidad de las Ciencias Informáticas (UCI), a través de la automatización de dicho proceso.

En estos momentos existe un sistema encargado de hacer la distribución de los estudiantes en la residencia. Dicho sistema se limita solamente a distribuir a los estudiantes, por este motivo se hace necesario concebir uno un poco más ambicioso, acorde a las tendencias del desarrollo de la informática actual, que sea capaz de distribuir eficientemente a todo tipo de huésped en la residencia de la Universidad de Ciencias Informáticas (UCI), además de ser implementado bajo la estructura de los Servicios Web, lo cual le permitirá un mayor nivel de integración con los demás sistemas implantados.

Esta herramienta propuesta, basada en un concepto de mucha actualidad: los servicios Web, contribuye al desarrollo de una eficiente organización de la universidad y por consiguiente, a la formación de profesionales integrales que constituyen el soporte de la informatización del país y la competitividad internacional de la industria cubana del software.

## ÍNDICE

<b>Agradecimientos</b> .....	<b>3</b>
<b>INTRODUCCIÓN</b> .....	<b>3</b>
<b>Fundamentación Teórica</b> .....	<b>5</b>
<b>Introducción</b> .....	<b>5</b>
1.2 Los Servicios Web.....	7
1.3 ¿Qué es XML? .....	8
1.4 Lenguaje de modelación y proceso de desarrollo del software. ....	9
1.5 Herramientas utilizadas (Rational Rose) .....	10
1.6 Algoritmo a seguir para realizar el proceso de distribución masiva de estudiantes....	10
<b>Conclusiones</b> .....	<b>13</b>
<b>Capítulo</b> .....	<b>14</b>
<b>Análisis del sistema</b> .....	<b>14</b>
<b>Introducción</b> .....	<b>14</b>
2.1 Objeto de estudio .....	14
Objeto de automatización.....	17
2.3 Problema a resolver. ....	17
2.4 Estudio de sistemas existentes. ....	18
2.5 Modelo del negocio: .....	19
2.6 Especificación de requisitos. ....	23
2.7 Requerimientos no funcionales .....	24
2.7 Casos de uso de alto nivel.....	25
<b>Conclusiones</b> .....	<b>30</b>
<b>Capítulo</b> .....	<b>31</b>
<b>Diseño del Sistema</b> .....	<b>31</b>
<b>Introducción</b> .....	<b>31</b>
3.2 Modelo Conceptual .....	46
.....	46
3.3 Diagramas de Secuencia.....	47
3.5 Diagramas del diseño web del sistema .....	55
3.7 Forma general y principios de la protección y seguridad.....	64
3.7.1 Forma general del tratamiento de errores .....	64
3.8 Diseño de la base de datos. ....	66
<b>Capítulo</b> .....	<b>68</b>
<b>Implementación del Sistema</b> .....	<b>68</b>
<b>Introducción</b> .....	<b>68</b>
4.1 Diagrama de Despliegue .....	68

<b>Recomendaciones .....</b>	<b>71</b>
<b>BIBLIOGRAFÍA.....</b>	<b>72</b>
<b>GLOSARIO DE TÉRMINOS Y SIGLAS .....</b>	<b>73</b>

## ***INTRODUCCIÓN***

En la actualidad la presencia de las tecnologías de la información y las comunicaciones se hace imprescindible en las actividades de la vida cotidiana. En la sociedad moderna este aspecto está adquiriendo un lugar trascendental e incuestionable y está cambiando las formas de aprender y transmitir los conocimientos. Nuestro país, además de estar inmerso en una verdadera revolución educacional, se ha dado a la tarea de automatizar al máximo la sociedad cubana.

En la Universidad de las Ciencias Informáticas (UCI), en correspondencia con el actual desarrollo y puesta en marcha de proyectos en el ámbito de la Informática y las Telecomunicaciones, se está desarrollando un proyecto de corte futurista que pretende automatizar al máximo todos los procesos, no solo relacionados con la actividad educacional dentro de la universidad, sino en el entorno de la sociedad misma. Este proyecto se ha denominado “UCI Ciudad Digital” y es la concepción de una sociedad totalmente informatizada que será punto de partida para la informatización de la sociedad cubana.

Con este propósito se desea que toda la gestión docente y social esté integrada y sea de fácil acceso para cualquier ciudadano. El presente trabajo -Sistema para la Gestión del Alojamiento- se enmarca dentro del área de los problemas de distribución de recursos y está vinculado a la distribución de los estudiantes, profesores y trabajadores en la residencia de acuerdo a una serie de restricciones que son de vital importancia para el buen funcionamiento de la Universidad.

En la actualidad, la distribución de los estudiantes se realiza utilizando una aplicación existente que se encarga de este proceso. Mientras que la distribución de los profesores y trabajadores se hace de forma manual. Para los profesores y trabajadores existe una persona que es la encargada de hacer la distribución, dicha persona trata de satisfacer de la mejor forma posible los intereses de ellos con el objetivo de que cada uno de ellos se sienta a gusto en el apartamento que le fue asignado.

En general, no se cuenta con un sistema que automatice por completo y de forma eficiente la distribución del personal en la residencia, lo cual implicaría elevar en un alto por ciento la eficiencia del proceso.

Como objetivo general de este proyecto, se define:

- Automatizar el proceso de la distribución del personal en la residencia.

A partir del análisis del objetivo general se derivan los siguientes objetivos específicos:

- Lograr una eficiente distribución de acuerdo a las restricciones que se tienen par cada caso.
- Garantizar la portabilidad del sistema y la facilidad para acoplarse a otras aplicaciones.

# Capítulo 1

## Fundamentación Teórica

### Introducción

En este capítulo se da una breve explicación sobre las características y potencialidades de la plataforma .NET. Además se muestran los aspectos más importantes de los servicios Web, obtenidos a través de un estudio detallado de la bibliografía existente sobre el tema.

También explicamos el algoritmo diseñado para realizar la distribución masiva de los estudiantes y una breve explicación sobre lo que es XML.

#### 1.1 ¿Qué es la plataforma .NET?

La plataforma .NET provee los cimientos para la nueva generación de software. Entre las características que presenta esta plataforma podemos mencionar que utiliza los Servicios Web como un medio para poder inter operar con distintas tecnologías, permite conectar distintos sistemas operativos, dispositivos físicos, información y usuarios, brinda 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 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.

Los desarrolladores por su parte tienen la infraestructura y herramientas para crear los servicios Web y hacer uso de ellos en programas. Es decir, se trata de aprovechar la capacidad de distribución a gran escala de Internet para acceder a servicios de software. También se trata de aprovechar el incremento en la capacidad de procesamiento de los nuevos dispositivos móviles llamados "Smart Devices" (dispositivos inteligentes) para que el usuario haga uso de la funcionalidad que proveen los servicios Web con interfaces cada vez más sencillas y naturales como la voz o la escritura.

El nuevo modelo de computación basado en Internet implica que las empresas no solamente tengan sitios donde el contenido puede ser accedido de manera visual como hasta ahora, con un explorador de Internet. Si estas empresas quieren ser exitosas deben crear componentes que implementen servicios relacionados con su actividad, para que usuarios o sitios, los integren y utilicen. Por ejemplo, una Aerolínea puede hacer componentes para la reserva de pasajes y desde una aplicación de una empresa de turismo llamar a este componente. O un usuario desde un dispositivo móvil (por ejemplo un celular) puede también invocar el componente de reserva de pasajes aéreos directamente para ver la disponibilidad y hacer reservas. La empresa turística puede exponer un servicio Web que incluya la llamada al servicio Web de la aerolínea. Cuando un servicio Web llama a otros se crea lo que se llama **federación de servicios Web** y las posibilidades funcionales se multiplican.



## **1.2 Los Servicios Web**

Con la arquitectura cliente-servidor, surge un modelo de desarrollo donde la aplicación se divide en una parte que interacciona con el usuario y otra parte destinada al procesamiento de la información. En este acercamiento se consigue que cada una de las partes que constituyen la aplicación pueda residir en computadoras distintas. Con el paso del tiempo, llega la era de las aplicaciones distribuidas en las cuales los procesos se realizan en diferentes unidades. De este paso surge la tecnología Internet para solventar las problemáticas asociadas al fallo de aplicación centralizado.

Con el surgimiento de los servicios Web, un ordenador ya no se considera como un núcleo de cómputo sino como un repositorio de servicios de N aplicaciones distribuidas por Internet.

Los servicios Web permiten que las aplicaciones compartan información y que además invoquen funciones de otras aplicaciones independientemente de cómo se hayan creado las aplicaciones, la plataforma en que se ejecutan, cuál sea el sistema operativo o software de servidor que utilicen. Aunque los servicios Web son independientes entre sí, pueden vincularse y formar un grupo de colaboración para realizar una tarea determinada.

Antes de la adopción del modelo de Servicios Web basados en XML los datos eran 'islas' que se encontraban dentro de las aplicaciones en las empresas. Era muy difícil y costoso implementar soluciones para acceder a la información desde afuera de la aplicación y la empresa. Las aplicaciones pueden ahora, comunicarse entre si y con los sistemas de sus socios, proveedores y clientes gracias a los Servicios Web y XML.

En resumen, con el uso de los servicios Web se integra la información que puede ser accedida desde distintos dispositivos, desde distintas plataformas de hardware o software y que puede estar guardada en distintos formatos. El lenguaje estándar para lograr esta integración es XML Todos los Servidores Corporativos de .NET entienden este lenguaje. Sigüientes versiones de estos servidores van a incorporar muchas mejoras en este

aspecto. Ejemplo de esto es la versión del SQL Server 2000 llamada Yukon. Este producto puede guardar datos en formato nativo XML, además permite hacer consultas al servidor no solamente en el lenguaje típico de bases de datos sino también en cualquier lenguaje compatible con la plataforma .NET.

### **1.3 ¿Qué es XML?**

XML no es más que un conjunto de reglas para definir etiquetas semánticas que organizan un documento en diferentes partes. XML es un metalenguaje que define la sintaxis utilizada para definir otros lenguajes de etiquetas estructurados.

XML es la base de los servicios Web. Está revolucionando la comunicación entre aplicaciones o, de forma más general, la comunicación entre equipos, pues ofrece un formato de datos universal que permite adaptar o transformar fácilmente la información.

A continuación se relacionan algunas de sus principales ventajas:

- Es una arquitectura más abierta y extensible. No se necesitan versiones para que puedan funcionar en futuros navegadores.
- Permite la integración de los datos de las fuentes más dispares. Se podrá hacer el intercambio de documentos entre las aplicaciones tanto en la propia computadora como en una red local o extensa.
- La extensibilidad y flexibilidad de este lenguaje brinda la posibilidad de agrupar una variedad amplia de aplicaciones, desde páginas Web hasta bases de datos.
- Permite la gestión y manipulación de los datos desde el propio cliente Web.

Los motores de búsqueda devuelven respuestas más adecuadas y precisas, ya que la codificación del contenido Web en XML consigue que la estructura de la información resulte más accesible.

#### **1.4 Lenguaje de modelación y proceso de desarrollo del software.**

La Tecnología de Orientación a Objetos constituye la base de la reutilización de código por medio de componentes. UML (Unified Modeling Language, Lenguaje de Modelado Unificado) es el lenguaje estándar adoptado por el OMG (Object Management Group) y mundialmente aceptado para la descripción de los "planos" de software. Utilizando UML se puede tener un lenguaje de comunicación que todos los integrantes de su equipo pueden entender, evitando la pérdida de claridad y concepto, y ahorrando tiempo y dinero en posibles correcciones y actualizaciones al reducir el tiempo de capacitación y desarrollo.

Ahora bien, un lenguaje indica la sintaxis para describir algo, pero no el método para hacer la descripción. RUP (Rational Unified Development Process, Proceso de Desarrollo Unificado de Rational) se erige como la metodología de construcción de software más poderosa por sus características de conducción por Casos de Uso y Orientación a Objetos.

Entre sus características principales están:

- a.) Guiado/Manejado por casos de uso: Los casos de uso reemplazan la antigua especificación funcional tradicional y constituyen la guía fundamental establecida para las actividades a realizar durante todo el proceso de desarrollo incluyendo el diseño, la implementación y las pruebas del sistema.
- b.)Centrado en arquitectura: La arquitectura involucra los elementos más significativos del sistema y está influenciada entre otros por plataformas de software, sistemas operativos, manejadores de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados y requerimientos no funcionales.
- c.) Iterativo e Incremental: Se divide el proyecto en ciclos. Para cada ciclo se establecen fases de referencia. Permite una comprensión creciente de los requerimientos, a la vez que se va haciendo crecer el sistema.
- d.) Utilización de un único lenguaje de modelación: UML.

### **1.5 Herramientas utilizadas (Rational Rose)**

Existen varias herramientas Case visuales que permiten realizar el modelado del desarrollo de los proyectos, como son: el Analise, el Designe, el Rational Rose. En este caso se utilizó el Rational Rose.

Rational Rose es la herramienta de modelación visual que provee el modelado basado en UML.

La Corporación Rational ofrece un Proceso Unificado (RUP) para el desarrollo de los proyectos de software, desde la etapa de Ingeniería de Requerimientos hasta la de pruebas. Para cada una de estas etapas existe una herramienta de ayuda en la administración de los proyectos.

### **1.6 Algoritmo a seguir para realizar el proceso de distribución masiva de estudiantes**

Con el objetivo de lograr en mayor medida que los estudiantes de un mismo grupo y un mismo sexo estén ubicados en un mismo apartamento de la residencia, se diseñó un algoritmo que, sin llegar a encontrar la distribución óptima, mejora en gran medida el cumplimiento de las restricciones.

Es importante señalar que diseñar un algoritmo que siempre encontrase la distribución óptima es algo bastante simple desde el punto de vista de implementación, sería esto:

Sea CantG la cantidad de grupos de la facultad y CantApar la cantidad de apartamentos del edificio donde se van a ubicar los estudiantes de un determinado sexo de la facultad.

**For i=1 to Factorial (CantApar) do**

**For K=1 to Factorial (CantG) do**

Por cada una de las permutaciones de la lista de apartamentos y para cada una de las permutaciones de la lista de grupos, hacer la distribución siguiendo el orden en que van apareciendo los apartamentos y grupos.

El problema de este algoritmo sería su tiempo de ejecución, el cual sería:  
 $\text{Factorial}(\text{CantApar}) * \text{Factorial}(\text{CantG}) * \text{TDistribución}$  (es el tiempo de ejecución para hacer una distribución siguiendo el orden en que aparecen los elementos).  
Para valores pequeños de CantApar y CantG no hay muchos problemas. Los problemas aparecen cuando nos enfrentamos a valores como los que se estiman que haya en la UCI, los cuales deben oscilar entre los 50 y 55 grupos por facultades y entre 25 y 30 apartamentos por edificios.

### **Algoritmo a seguir en el proceso de distribución masiva de estudiantes .**

Sean  $G_1, G_2, \dots, G_{\text{cantGrupo}}$  los grupos de una facultad y  $A_1, A_2, \dots, A_{\text{cantApart}}$  los apartamentos del edificio donde se va a ubicar a los estudiantes de esa facultad.

1) Agrupamos los grupos y los apartamentos en conjuntos que se caracterizan por la matrícula de estudiantes que tienen los grupos de cada conjunto de grupo y por la capacidad de los apartamentos que tiene cada conjunto de apartamentos respectivamente. O sea, cada uno de esos conjuntos de grupos o conjuntos de apartamentos van a tener una lista de grupos o de apartamentos, en dependencia de qué tipo de conjunto sea, y un número entero que representa la cantidad de de estudiantes de todos los grupos del conjunto, en el caso que sea un conjunto de grupos o la capacidad de todos los apartamentos del conjunto de apartamentos, en el caso que sea un conjunto de apartamentos.

$\text{ConjGrupK}_1, \text{ConjGrupK}_2, \dots, \text{ConjGrupK}_n \text{ -----} \rightarrow$  Conjunto de grupos.

$\text{ConjGrupK}_1 \text{-----} \rightarrow$  Es un conjunto de grupos donde todos los grupos de ese conjunto tienen una matrícula de estudiantes igual a  $K_1$ .

ConjApartP1, ConjApartP2,..., ConjApartPi ----→ Conjunto de apartamentos.

ConjApartP1----→ Conjunto de apartamentos donde todos los apartamentos de ese conjunto tienen una capacidad de P1 personas.

Calculamos todos los restos de las divisiones de las cantidades que caracterizan a los conjuntos de grupos entre las cantidades que caracterizan a los conjuntos de apartamentos.

Resto11=K1/P1, Resto12=K1/P2,..., Resto1i=K1/Pi, Resto21=K2/P1,  
Resto22=K2/P2,..., Resto2i=K2/Pi,..., Restoni=Kn/Pi.

1) Mientras haya algún resto de división que sea cero tomamos a los estudiantes de cada uno de los grupos del conjunto de grupo y los ubicamos en los apartamentos del conjunto de apartamentos, hasta que se acaben los estudiantes o se agoten los apartamentos de los conjuntos en cuestión.

2) Si se ejecutó el paso tres volvemos al paso número uno.

3) Si no se ejecutó el paso número tres, entonces:

Buscamos un conjunto de grupo y un conjunto de apartamentos, tal que al restarle a uno de los grupos la cantidad de integrantes de uno de los apartamentos del conjunto de apartamentos nos quede una cantidad que sí sea divisible por alguna de las cantidades de los demás conjuntos de apartamentos que existen. Si lo encontramos:

Sea X la capacidad de los apartamentos del conjunto de apartamentos y C la cantidad de apartamentos.

```
For i=1 to C do
{
  Para cada grupo del conjunto de grupo (siempre que i sea menor o igual que la
  cantidad de grupos del conjunto)

    Ubicar X estudiantes del grupo número i en el apartamento número i.

}
```

Con esto habremos logrado que nos quede un conjunto de grupos cuyas cantidades de estudiantes sí van a ser divisibles por unas de las capacidades de los conjuntos de apartamentos que ya existen. Luego volvemos al paso número uno.

Si no encontramos un conjunto de grupos y un conjunto de apartamentos que cumplan la condición planteada inicialmente en el paso número tres:

Comenzamos a movernos por los conjuntos de apartamentos y por los conjuntos de grupos y vamos ubicando a los estudiantes tal y como vayamos cogiéndolos, o sea, sin seguir ninguna lógica.

## **Conclusiones**

Según las características expuestas en este capítulo se puede concluir que los servicios Web son la revolución informática de la nueva generación de aplicaciones (la cual tiene su base en la plataforma .NET que se encarga de proveer los cimientos para la nueva generación de software) que trabajan en colaboración, en las cuales el software está distribuido en diferentes servidores. Además, el objetivo principal que se logra con el uso de estos servicios, es la interoperabilidad y la integración, pues permiten que las aplicaciones compartan información y que además invoquen funciones de otros sistemas. La idea anterior puede ser utilizada en el sistema a desarrollar para garantizar la integración con el resto de las aplicaciones que conforman el proyecto "UCI Ciudad

Digital". Además es válido aclarar que el éxito de esta tecnología radica en que se basa en estándares conocidos en los que ya se tiene una gran confianza, como el XML, por lo que en este sentido, también resulta conveniente utilizar los servicios Web.

## **Capítulo 2**

### **Análisis del sistema**

#### **Introducción**

A través de este capítulo se describen los objetos de estudio y automatización, los principales problemas existentes en cuanto a la distribución de los estudiantes, profesores y trabajadores de la Universidad de las Ciencias Informáticas, se hace un análisis comparativo de una solución existente en la Universidad, se analizan los requerimientos funcionales y no funcionales, y los casos de uso del sistema.

#### **2.1 Objeto de estudio**

##### **2.1.1 Situación Problemática**



En la Universidad de las Ciencias Informáticas (UCI) se realiza al inicio de cada curso escolar la distribución de estudiantes, profesores y trabajadores en la residencia. Aunque debemos aclarar que después de haber comenzado el curso también se puede realizar la ubicación individual o masiva de estudiantes, profesores o trabajadores. A este proceso pudiera incorporarse en un futuro la ubicación de visitantes en la residencia.

La situación actual en la Universidad de la Ciencias Informáticas reúne los siguientes aspectos:

- La matrícula comprende un número de alrededor de 4 mil estudiantes con perspectivas de ampliación, lo que provoca que la distribución de los mismos sea por facultades. Actualmente existen 6 facultades, cada una de ellas compuesta por 11 grupos, con excepción de la facultad 6 que tiene 12 grupos. En la facultad 1 se encuentran los grupos especiales de segundo año.
- Actualmente hay 287 profesores y 47 trabajadores ubicados en la residencia.
- El número elevado de personas a distribuir en la residencia hace que sea sumamente costoso lograr una distribución óptima del personal.

### **2.1.2. Problema**

¿Cómo ocurre actualmente este proceso?

La ubicación de los estudiantes, profesores y trabajadores en la residencia se hace de la siguiente manera:

Para los estudiantes:

- Se pretende lograr que todos los que conviven en un edificio sean de la misma facultad y los de un mismo cuarto que sean del mismo grupo.
- Todos tienen que ser del mismo sexo en un edificio.

Para los profesores y trabajadores:

La ubicación de los profesores se hace en dependencia de los siguientes criterios, teniendo en cuenta la disponibilidad de alojamiento en la residencia y el orden de llegada a la Universidad:

- Categoría docente y científica:

De la categoría de auxiliar hacia arriba deben de ir solos en un cuarto.

- Los matrimonios **oficiales entre profesores** deben de ir solos en un cuarto. Subrayando siempre el apellido oficiales.

- Los cuadros directivos de la universidad deben de ser ubicados solos en un apartamento para ellos y su familia.
- Las profesoras deben de ser ubicadas en las plantas bajas de los edificios.
- En caso de que la residencia no esté en condiciones de satisfacer alguna de las dos primeras condiciones, al menos se debe garantizar que sean ubicados en un apartamento donde convivan con personas con características similares a las suyas. Para el caso de los profesores con categoría, se deben ubicar en

apartamentos donde todos los integrantes tengan la misma categoría y para el caso de los matrimonios de forma equivalente.

Debemos aclarar que todo este proceso es realizado de forma manual, exceptuando la distribución de los estudiantes que es realizada mediante un sistema que ya está en funcionamiento.

### **Objeto de automatización.**

En el proceso de distribución en la residencia, se desea automatizar en estos momentos la generación de las ubicaciones en los apartamentos a partir de las experiencias que se tienen para lograr una excelente colocación. Para ello se debe crear una infraestructura que permita conocer a fondo todas las inquietudes y recomendaciones de los encargados en hacer actualmente esta tarea y de los ubicados.

### **2.3 Problema a resolver.**

El proceso actual de distribución del personal en la residencia presenta varios inconvenientes que dificultan una gestión a la altura de las tecnologías actuales de la informática. Fundamentalmente en la distribución de los profesores y trabajadores que se hace de forma completamente manual. La inexistencia de un software capaz de asumir todas las responsabilidades existentes tiene como consecuencia:

- ✓ Obtener las listas de personas por apartamentos que serán ubicados en un edificio en modelos impresos, con datos entrados manualmente en el caso de los profesores y trabajadores lo cual propicia la introducción de errores humanos.
- ✓ . La existencia de un gran volumen de información difícil de procesar, engorroso y poco eficiente a la hora de trabajar y gastos innecesarios de recursos y espacio.

- ✓ La no ubicación de las personas de la forma más óptima posible. Trayendo esto como resultado inconformidades por parte de los estudiantes, profesores y trabajadores.
- ✓ Ninguna integración a otros sistemas.

#### **2.4 Estudio de sistemas existentes.**

Actualmente en la Universidad de Ciencias Informáticas se cuenta con un sistema que es capaz de hacer sólo la distribución de los estudiantes en la residencia. A continuación haremos una pequeña comparación entre dicho sistema y el que se quiere desarrollar.

<b>Sistemas</b>	<b>Tipos de personas que distribuye</b>	<b>Eficiencia en el proceso de distribución</b>	<b>Facilidad de comunicación y acoplamiento con otros sistemas</b>
Sistema existente	Sólo distribuye a los estudiantes	Carece de un fuerte algoritmo capaz de satisfacer a cabalidad las restricciones existentes para la distribución de los estudiantes.	No presenta grandes posibilidades de comunicación y acoplamiento con otros sistemas.
Sistema a implantar	Distribuye a estudiantes, profesores y trabajadores	Es capaz de encontrar una buena distribución que disminuye el número de restricciones que quedan sin ser cumplidas.	Todos sus módulos están encapsulados dentro de servicios Web XML, lo cual proporciona una gran facilidad de comunicación y acoplamiento con otros sistemas.

¿Por qué decimos que el sistema a implantar posee grandes facilidades de comunicación y acoplamiento con otros sistemas?

#### **LOS SERVICIOS WEB XML:**

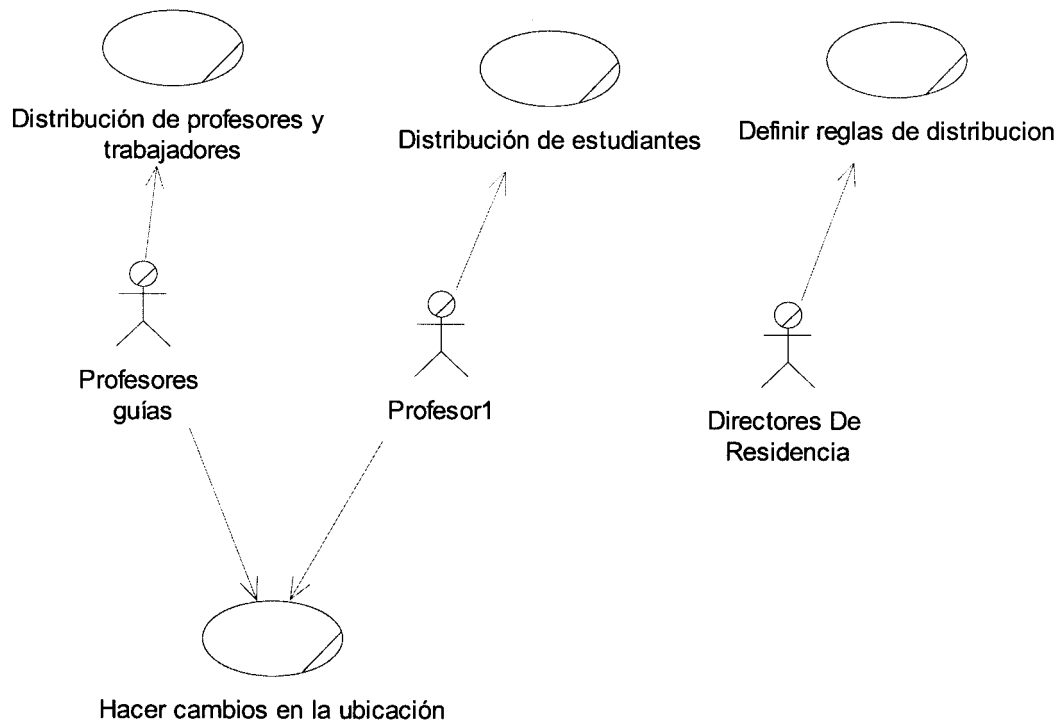
Los servicios Web XML permiten que las aplicaciones compartan información y que además invoquen funciones de otras aplicaciones independientemente de cómo se hayan creado las aplicaciones, cuál sea el sistema operativo o la plataforma en que se ejecutan y cuáles los dispositivos utilizados para obtener acceso a ellas. Aunque los servicios Web XML son independientes entre sí, pueden vincularse y formar un grupo de colaboración para realizar una tarea determinada.

¿Para qué sirve un Servicio Web? La respuesta puede ser otra pregunta: ¿Para que sirve en programación una rutina? Todos sabemos que una rutina es como una caja negra, que encierra cierto proceso o algoritmo, y que cumple una función clara. Muchas rutinas y un guión central componen un programa en lo que se llama "programación estructurada". Un Servicio Web viene a ser una rutina en Internet.

### **2.5 Modelo del negocio:**

<b>Actores del negocio</b>	<b>Justificación</b>
Profesor1	Representa a un compañero que es el responsable del alojamiento de los profesores en la residencia.
Directores de la residencia	Son los encargados de dictar las orientaciones y regulaciones por las cuales se va a regir la ubicación en la residencia.
Profesores guías.	Son los encargados de hacer la distribución de los estudiantes.

#### **2.5.1 Representación Gráfica del Diagrama de casos de uso del negocio**



### **2.5.2 Listado de los casos de uso del negocio**

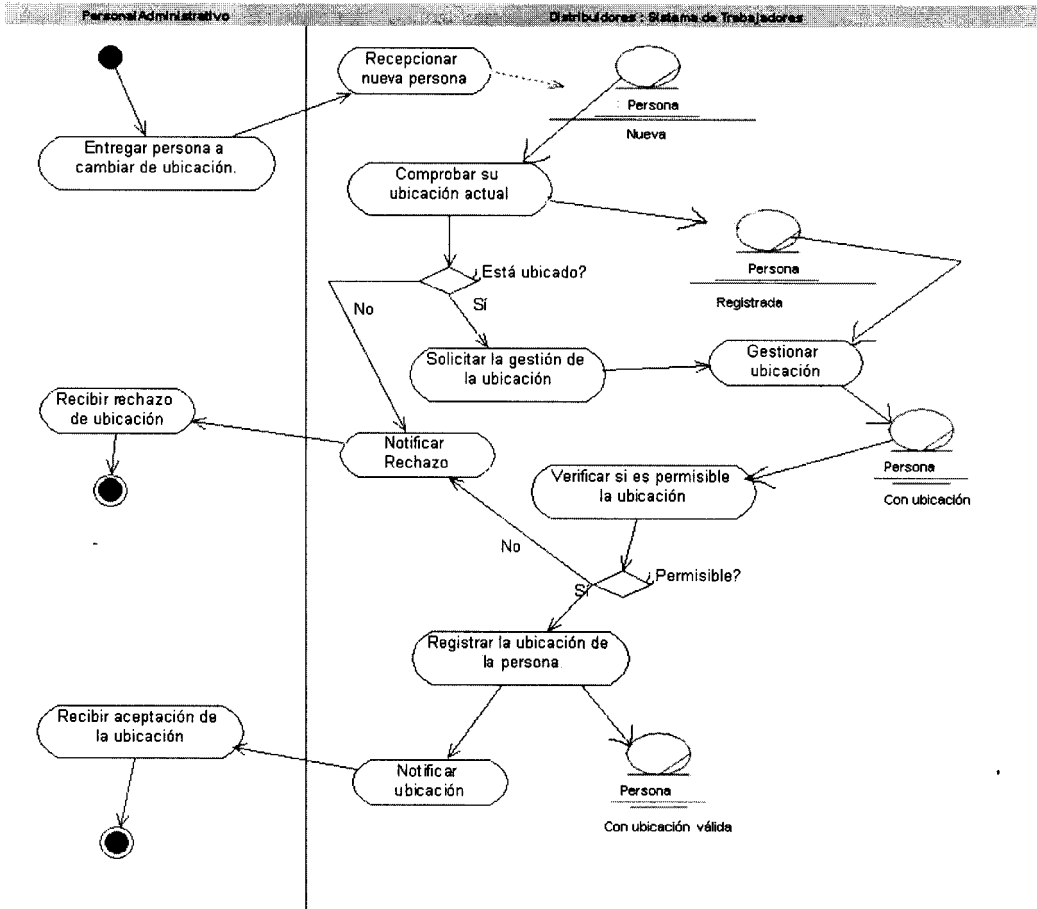
CU	NOMBRE
1	Distribución de estudiantes
2	Distribución de profesores y trabajadores
3	Definir reglas de distribución
4	Hacer cambios en la ubicación

### **2.5.3 Especificación del caso de uso: "Hacer cambios en la ubicación"**

Caso de uso	Hacer cambios en la ubicación	
Actores	Directores de la residencia	
Propósito	Realizar cambios en la ubicación de una persona.	
Flujo de trabajo Básico	Flujo de trabajo alternativo	
1)	Lo primero que se hace es	

comprobar que la persona que vamos cambiar de ubicación realmente está ubicada en la residencia.	
2) En caso afirmativo se pasa a gestionar la ubicación a partir de las características de la persona a ubicar. Se obtienen un listado de las posibles ubicaciones para esa persona.	Si no está se detiene la operación debido a que no podemos cambiar de ubicación a una persona que no está registrada en la residencia.
3) De todas las posibles ubicaciones se selecciona la mejor, la que se adapte en mayor medida a las características de dicha persona.	
4) Actualizamos la ubicación de la persona.	

#### **2.5.4 Diagrama de Actividad del Caso de Uso Hacer cambios en la ubicación**





## **2.6 Especificación de requisitos.**

### Requerimientos funcionales

Los requerimientos o requisitos funcionales se definen como aquellas opciones que debe ejecutar el software, las operaciones que se realizan sin que el usuario tenga que solicitarlas directamente o las condiciones de excepción o error que el sistema debe manejar.

Los requisitos del sistema se enuncian a continuación:

- 1) El sistema debe permitir hacer la distribución masiva de los estudiantes y profesores.
  - 1.1) El sistema debe seleccionar cuál es el edificio que más le conviene para ubicar a los estudiantes de una facultad en particular y también debe brindar la posibilidad de ser escogido por el usuario.
  - 1.2) El sistema debe permitir escoger cuáles son los estudiantes o profesores adiestrados para cada uno de los cuartos después de haber realizado la distribución debido a que el proceso de distribución devuelve las cantidades de estudiantes o profesores adiestrados por cuarto, que cumplen con ciertas restricciones y no los nombres de los integrantes.
  - 1.3) También debe brindar la posibilidad de hacer él mismo el proceso de selección después de haber realizado la distribución.
- 2) El sistema debe permitir la ubicación individual en la residencia de cualquiera de los individuos antes mencionados, además de los profesores con categorías, matrimonios, cuadros directivos y trabajadores.

Para ubicar una persona, el sistema debe permitir suministrar el número del carné de identidad de la persona.

- 3) El sistema de brindar una serie de reportes con los listados de las ubicaciones por edificios y cuartos.
- 4) Debe permitir actualizaciones de forma manual sobre la ubicación de un huésped en la residencia.
- 5) Gestionar identidad del usuario que va a manipular el sistema, mediante el nombre de sección y su contraseña.
- 6) El sistema debe brindar la posibilidad de agregar, eliminar o modificar los usuarios que tendrán permiso para trabajar con él.
- 7) El sistema debe permitir definir las características de un edificio y sus apartamentos.

### **2.7 Requerimientos no funcionales.**

Los requerimientos no funcionales son “características que describen alguna forma o restricción para la realización de algún requerimiento (funcionalidad) o conjunto de ellas e inclusive todos los requerimientos”.

Los requerimientos no funcionales se muestran a continuación:

#### **Apariencia o interfaz externa:**

La interfaz debe ser sencilla, intuitiva, amigable y mantener el formato en páginas similares. En general, fácil de usar.

#### **Requerimientos de Portabilidad:**

El sistema deberá funcionar sobre plataforma Windows, que es la que más usada hasta el momento en la universidad.

Requerimientos de Software:

Se utilizará Microsoft Visual Studio.Net 2003 y SQL Server como gestor de Base de Datos.

Requerimientos de Seguridad:

*Confabilidad* : la información manejada por el sistema está protegida de acceso no autorizado.

*Integridad* : la información será protegida contra corrupción y estados inconsistentes.

*Disponibilidad* : Los usuarios autorizados tendrán acceso a la información en todo momento.

Requerimiento de ayuda y documentación en línea.

Se deberá brindar ayuda en línea.

Restricciones de diseño

Herramientas de desarrollo gráfico.

Se utiliza el Rational Rose como herramienta de desarrollo gráfico para la modelación del sistema.

Soporte

*Extensibilidad*: se debe garantizar la inserción de módulos nuevos, sin negar lo realizado hasta el momento o afectar el buen funcionamiento.

*Mantenimiento*: El sistema debe estar bien documentado de forma tal que el tiempo de mantenimiento sea mínimo en caso de necesitarse.

**2.7 Casos de uso de alto nivel.**

En este epígrafe se abordarán los casos de uso de alto nivel, definiendo previamente los actores que intervienen en los mismos.

### 2.7.1 Tabla de Actores del sistema.

<b>ACTORES</b>	<b>DESCRIPCIÓN</b>
Administrador	Administrador del sistema.
Cliente del sistema	Usuario del sistema. Tiene privilegios para ejecutar cualquier consulta en el sistema excepto las de administración de los usuarios que son de uso exclusivo del administrador.
Directorio activo	Base de datos del dominio que almacena la información referente al dominio.
Sistema de Trabajadores	Sistema que maneja la información de los trabajadores de la UCI, excepto la de los profesores que está contenida en el Sistema Docente.
Sistema Docente	Sistema que gestiona la información de los estudiantes y profesores de la UCI.
Base de datos inmuebles	Base de datos donde se almacena las características físicas de los edificios.

### 2.7.2 Lista de casos de uso del sistema.

<b>CU</b>	<b>NOMBRE</b>
1	Validar identidad del usuario
2	Distribución Masiva
3	Distribución individual
4	Conformar los reportes
5	Administrar Usuarios
6	Definir características de los edificios y apartamentos
7	Hacer cambios en la ubicación

<b>CU-1</b>	<b>Validar identidad del usuario</b>
Actores	Cliente del sistema, Directorio activo.
Descripción	El usuario entra sus datos, donde se valida la entrada, y en dependencia del tipo el sistema le muestra una página con las opciones que puede realizar.
Referencia	5

<b>CU-2</b>	<b>Realizar distribución masiva</b>
Actores	Usuario, Sistema Docente, Base de datos inmueble.
Descripción	El sistema le muestra al usuario los tipos de distribuciones masivas posibles a realizar y en dependencia de la selección, el sistema hará el proceso de distribución.
Referencia	1

<b>CU-3</b>	<b>Realizar distribución individual</b>
Actores	Usuario, Sistema Docente, Sistema de Trabajadores.
Descripción	El sistema le muestra al usuario los tipos de distribuciones individuales posibles a realizar y en dependencia de la selección, el sistema le pedirá los datos necesarios para realizar dicha distribución.
Referencia	2

<b>CU-4</b>	<b>Conformar los reportes</b>
Actores	Usuario, Sistema Residencia.
Descripción	El sistema debe permitir escoger el tipo de reportes que desea mostrar
Referencia	3

<b>CU-5</b>	<b>Administrar usuarios.</b>
Actores	Administrador
Descripción	El administrador del sistema puede incorporar o eliminar usuarios al sistema.
Referencia	6

<b>CU-6</b>	<b>Definir características de los edificios y apartamentos</b>
Actores	Base de datos inmuebles
Descripción	Aquí se especificará una serie de características para los edificios y sus apartamentos que facilitarán el proceso de distribución.
Referencia	7

<b>CU-7</b>	<b>Hacer cambios en la ubicación</b>
Actores	Cliente del sistema.
Descripción	Aquí se podrá realizar cualquier tipo de cambios en la ubicación de una persona en la residencia.
Referencia	4

## **2.8 Selección de los Casos de Uso para cada ciclo**

En el núcleo central se representan los casos de usos que más profundamente influyen en la arquitectura básica del sistema, dando soporte al dominio y a las capas de servicio de alto nivel o los que representen el máximo riesgo, funciones urgentes o complejas.

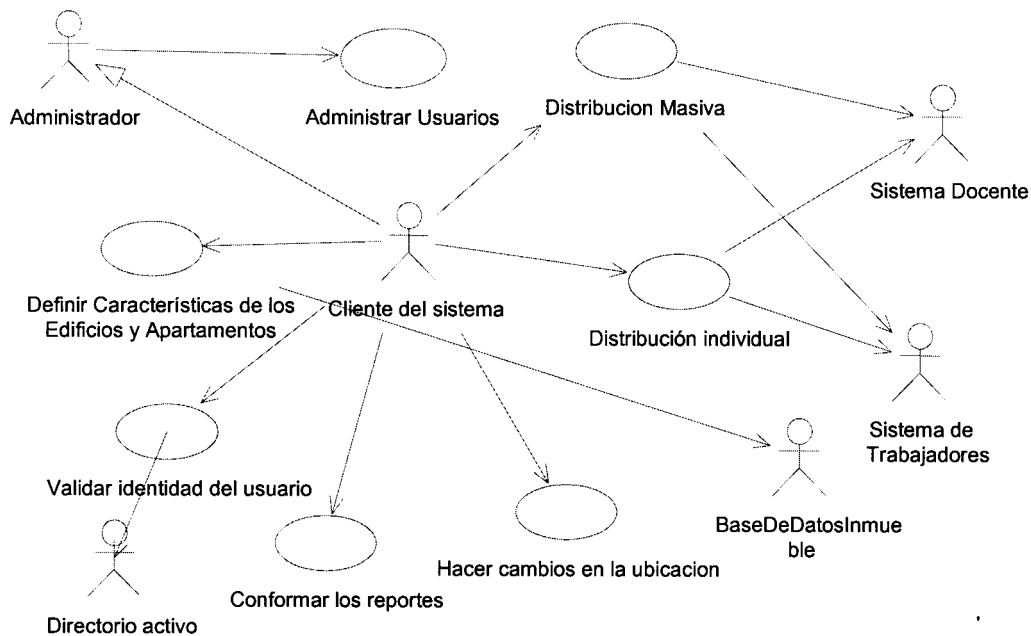
Mediante la siguiente tabla se representan los casos de uso del núcleo central y del segundo ciclo de desarrollo.

<b>PRIMER CICLO DE DESARROLLO</b>	<b>JUSTIFICACION</b>
Realizar distribución masiva	Representan los procesos primarios del soporte de la aplicación.
Realizar distribución individual	
Definir características de los edificios y apartamentos	
<b>SEGUNDO CICLO DE DESARROLLO</b>	No forman parte de las funciones prioritarias del sistema.
Autenticar el usuario	
Administrar usuarios	
Mostrar reportes	
Hacer cambios en la ubicación	
Realizar distribución masiva	
Realizar distribución individual	

**Nota:** Los casos de uso distribución masiva y distribución individual están incluidos en ambos ciclos de desarrollo porque en el primer ciclo no se va a culminar su implementación sino que se continua en el segundo ciclo.

## **2.9 Diagramas de Casos de Uso del Sistema**

Un diagrama de Casos de Uso muestra, de forma gráfica, las diferentes funciones que se esperan de una aplicación a través de los casos de uso y cómo se relacionan con su entorno (actores), para así definir las formas básicas en que estos utilizan el sistema.



## **Conclusiones**

Al finalizar la etapa de Estudio Preliminar han quedado definidos los problemas existentes actualmente en la distribución del personal en la residencia de la Universidad de Ciencias Informáticas, por lo que se evidencia la necesidad del mejoramiento de este proceso, con el uso de los mecanismos descritos en los siete casos de uso definidos.

Además se planteó un conjunto de requisitos funcionales y no funcionales que describen, respectivamente, la funcionalidad y las propiedades que debe tener el sistema a desarrollar.

Los casos de uso fueron distribuidos en dos ciclos de desarrollo, dejándose para el segundo ciclo, los casos de uso de menos importancia y la gestión de los usuarios que interactúan con la aplicación.



### **Introducción**

El análisis forma parte del proceso de desarrollo de software, cuyo propósito primario es formular el modelo del dominio del problema.

En el presente capítulo se procede a representar la expansión de los casos de uso, el modelo conceptual, y los diagramas de secuencia y de iteración, utilizando para su modelado el Lenguaje Unificado de Modelación (UML).

#### **3.1. Expansión de los casos de uso**

A través de la expansión de los casos de uso se describe paso a paso la secuencia de eventos que los actores utilizan para completar un proceso a través del sistema

Caso de uso	Realizar distribución masiva.	
Actores	Usuario, Sistema Docente.	
Propósito	Realizar la distribución masiva de estudiantes y profesores adiestrados.	
Resumen: El sistema le muestra al usuario los tipos de distribuciones masivas posibles a realizar y en dependencia de la selección, el sistema le pedirá los datos necesarios para realizar dicha distribución.		
	Acción del actor	Respuesta del sistema
	<p>1) El caso de uso comienza cuando el usuario selecciona uno de los tipos de distribuciones masivas posibles a realizar.</p> <p>a) Si la opción seleccionada se refiere a distribución masiva de estudiantes, véase la sección <i>Distribución masiva de Estudiantes</i>.</p> <p>b) Si la opción seleccionada se refiere a la distribución masiva de profesores, véase la sección <i>Distribución masiva de profesores adiestrados</i>.</p>	
<b>Sección: <i>Distribución masiva de Estudiantes.</i></b>		
	<b>Acción del actor</b>	<b>Respuesta del sistema</b>

	2) El sistema muestra un formulario para que se introduzcan algunos datos relacionados con las características de los estudiantes y para que se especifique el edificio donde se quiere alojar a dichos estudiantes.
3) El usuario introduce los datos solicitado por el sistema.	4) El sistema guarda la característica a considerar en el proceso de distribución.  5) El sistema solicita las cantidades de los estudiantes que poseen las características seleccionadas, al Sistema Base Única del Ciudadano y al Sistema Docente.
6) El Sistema Base Única del Ciudadano y el Sistema docente facilita las cantidades.	7) El sistema pasa a calcular la mejor ubicación posible del personal.  8) El sistema realiza la ubicación física.

**Sección: Distribución masiva de profesores adiestrados.**

<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	1) El sistema solicita las cantidades de profesoras y profesores adiestrados separados por centro de procedencia, al Sistema Base Única del Ciudadano.
2) El Sistema Base Única del	3) El sistema pasa a calcular la mejor

Ciudadano facilita las cantidades.	ubicación posible del personal. 4) El sistema realiza la asignación.
------------------------------------	---

Caso de uso	Conformar los reportes	
Actores	Usuario, Sistema Residencia.	
Propósito	Realizar la confección y muestra de los reportes	
Resumen: El sistema debe permitir escoger el tipo de reportes que desea mostrar el usuario		
	Acción del actor	Respuesta del sistema
	1) El caso de uso comienza cuando el Usuario selecciona uno de los reportes posibles a Mostar.	2) El sistema muestra un formulario para que el usuario introduzca las características necesarias para confeccionar el reporte.
	3) El usuario introduce los datos.	4) El sistema gestiona la lista de personas que poseen las características especificadas, para esto debe auxiliarse del Sistema Base Única del ciudadano.  6) El sistema muestra en un reporte la lista de personas.

Caso de uso	Realizar distribución individual.
Actores	Usuario, Sistema Docente.
Propósito	Realizar la distribución individual de cualquiera de los tipos de huéspedes
Resumen: El sistema le muestra al usuario los tipos de distribuciones individuales posibles a realizar y en dependencia de la selección, el sistema le pedirá los datos necesarios para realizar dicha distribución.	
Acción del actor	Respuesta del sistema
1) El caso de uso comienza cuando el usuario selecciona uno de los tipos de distribuciones individuales posibles a realizar.	2) El sistema muestra un formulario para que se introduzcan algunos datos relacionados con la distribución escogida.
3) El usuario introduce los datos solicitados por el sistema.	4) El sistema verifica que la o las personas estén registradas en la universidad y que no estén ubicados en la residencia y posteriormente gestiona las posibles ubicaciones. 5) El sistema muestra al usuario las posibles ubicaciones.
6) El usuario selecciona una.	7) El sistema almacena la selección.

Caso de uso	Administrar usuarios	
Actores	Administrador, Sistema Residencia.	
Propósito	Realizar el control sobre los usuarios que pueden trabajar con el sistema.	
<p>Resumen: El sistema debe permitir la inserción, eliminación y actualización de un usuario al sistema. Sólo hay dos grupos de usuarios, los administradores y los operadores. Los operadores no tienen permisos para hacer cambios en los grupos de usuarios pero sí pueden realizar cualquier otra operación en el sistema.</p>		
	Acción del actor	Respuesta del sistema
	<p>1) El caso de uso comienza cuando el Usuario selecciona que desea entrar al módulo de administración.</p> <p>3) En caso de tener permisos el usuario selecciona una de las acciones disponibles para realizar.</p> <p>a) Si la opción seleccionada se refiere a insertar usuario, véase la sección: <i>Inserción de usuarios.</i></p> <p>b) Si la opción seleccionada se refiere a eliminar usuario, véase la sección: <i>Eliminación de usuarios.</i></p> <p>c) Si la opción seleccionada se refiere</p>	<p>2) El sistema verifica que el usuario activo tiene permisos suficientes para administrar.</p>

a modificar usuario, véase la sección: <i>Actualización de usuarios.</i>	
<i>Sección: Inserción de usuarios..</i>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	4) El sistema muestra un formulario para que el usuario introduzca el nombre del usuario a insertar y el grupo donde desea insertarlo.
5) El usuario introduce el nombre.	6) El sistema muestra, usando el directorio activo, todos los usuarios de la Universidad que coinciden con el nombre suministrado para que se escoja cuál es.
7) El usuario escoge uno.	8) El sistema almacena el nombre de usuario.
<i>Sección: Eliminación de usuarios</i>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	4) El sistema muestra un formulario con los dos grupos de usuarios y cada uno de sus miembros para que el usuario activo seleccione cuál es el que desea eliminar.
5) El usuario selecciona uno.	6) El sistema realiza la eliminación.

<i>Sección: Actualización de usuarios</i>	
Acción del actor	Respuesta del sistema
	4) El sistema muestra un formulario con los dos grupos de usuarios y cada uno de sus miembros para que el usuario activo seleccione cuál es el que desea actualizar.
5) El usuario selecciona uno.	6) El sistema muestra los grupos de usuarios para que se seleccione uno.
7) El usuario actualiza el grupo al que pertenece.	8) El sistema almacena los datos del usuario.

<b>Caso de uso</b>	<b>Definir Características de los Edificios y Apartamentos</b>	
<b>Actores</b>	Cliente del sistema, Base de datos inmuebles.	
<b>Propósito</b>	Brindar la posibilidad de definir todas las características de los edificios y apartamentos.	
Resumen: Aquí se especificará una serie de características para los edificios y sus apartamentos que facilitarán el proceso de distribución.		
Acción del actor	Respuesta del sistema	
1) El caso de uso comienza cuando el Usuario selecciona que desea entrar al módulo de definición de las	2) El sistema muestra un formulario con la lista de edificios.	



características de los edificios y apartamentos.	
3) El usuario selecciona un edificio.	4) El sistema muestra un formulario para que el usuario introduzca las características que se desean incorporar al edificio.
5) El usuario introduce los datos	6) El sistema almacena las características para el edificio, en caso de que ya hayan características especificadas para este edificio el sistema envía una advertencia.  7) El sistema pregunta si desea especificar características para un apartamento en particular.
8) a) Si el usuario escoge que sí desea especificar características para un apartamento en particular, véase la sección: <i>Definir características de un apartamento.</i>  b) Si escoge que no, véase la sección: <i>Salir</i>	
<i>Sección: Definir características de un apartamento.</i>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>

	9) El sistema muestra un formulario con los apartamentos del edificio y las características que se han definido para cada uno de ellos para que el usuario las pueda modificar a su gusto.
10) El usuario realiza las modificaciones que estime.	11) El sistema almacena los datos suministrados por el usuario.
<b>Sección: Salir</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	9) El sistema sale del módulo encargado de la definición de las características de los edificios.

Caso de Uso	Validar identidad del usuario
Actores	Cliente del sistema, Directorio activo
<b>Descripción:</b> El usuario entra sus datos, donde se valida la entrada, para darle o no acceso al sistema.	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1) Comienza cuando el usuario solicita registrarse para entrar en el sistema.	2) El sistema muestra un formulario para que se introduzca el nombre de usuario del dominio y la clave.

3) El usuario suministra los datos.	4) El sistema verifica la autenticación con el directorio activo.
4) El directorio activo envía una respuesta.	5) En caso de ser positiva la respuesta del directorio activo el sistema verifica si tiene permisos para trabajar con él. En caso de tenerlos, verifica que tipo de usuario es, en caso de que sea un administrador debe almacenarlo para posteriormente brindarle acceso al módulo de administración de usuarios en caso de sea solicitado por él.

Caso de Uso	Hacer cambios en la ubicación
Actores	Cliente del sistema
<b>Descripción:</b> Aquí se podrá realizar cualquier tipo de cambios en la ubicación de una persona en la residencia.	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1) Comienza cuando el usuario solicita la opción de realizar cambios en la ubicación.	2) El sistema muestra un formulario con las diferentes formas de realizar actualizaciones en la residencia.

<p>3)</p> <p>a) Si la opción seleccionada se refiere a realizar cambios de una persona véase la sección: <i>Cambios de una persona</i></p> <p>b) Si la opción seleccionada se refiere a realizar cambios en un apartamento, véase la sección: <i>realizar cambios en un apartamento</i></p>	
<p>Sección: <i>Cambios de una persona</i></p>	
<p><b>Acción del actor</b></p>	<p><b>Respuesta del sistema</b></p>
	<p>4) El sistema muestra un formulario para que se introduzca el nombre de la persona a la que se le desea realizar cambios en la ubicación.</p>
<p>5) El usuario introduce el nombre</p>	<p>6) El sistema muestra todos los usuarios del dominio que coinciden con el nombre suministrado para que se seleccione uno de ellos.</p>
<p>7) El usuario selecciona uno.</p>	<p>8) El sistema muestra dos acciones a realizar con esa persona, una es eliminarlo del apartamento en que está ubicado actualmente y la otra es cambiarlo de ubicación.</p>

<p>9)</p> <p>a) Si opción seleccionada es eliminarlo véase la sección: <i>desubicar persona</i>.</p> <p>b) Si la opción seleccionada es cambiarlo de ubicación, véase la sección: <i>cambiar de ubicación</i></p>	
<p><i>Sección desubicar persona</i></p>	
<p><b>Acción del actor</b></p>	<p><b>Respuesta del sistema</b></p>
	<p>10) El sistema elimina a la persona del apartamento.</p>
<p><i>Sección: cambiar de ubicación</i></p>	
<p><b>Acción del actor</b></p>	<p><b>Respuesta del sistema</b></p>
	<p>10) El sistema muestra un formulario con la ubicación actual de la persona y una lista de los edificios con sus apartamentos para que se seleccione donde es que se desea reubicar.</p>
<p>11) El usuario selecciona un edificio y un apartamento.</p>	<p>12) El sistema verifica si es posible ubicar dicha persona en dicho apartamento, teniendo en cuenta las características definidas para ese apartamento. En caso de ser posible lo inserta ahí y lo elimina de su antigua ubicación.</p>

*Sección: realizar cambios en un apartamento*

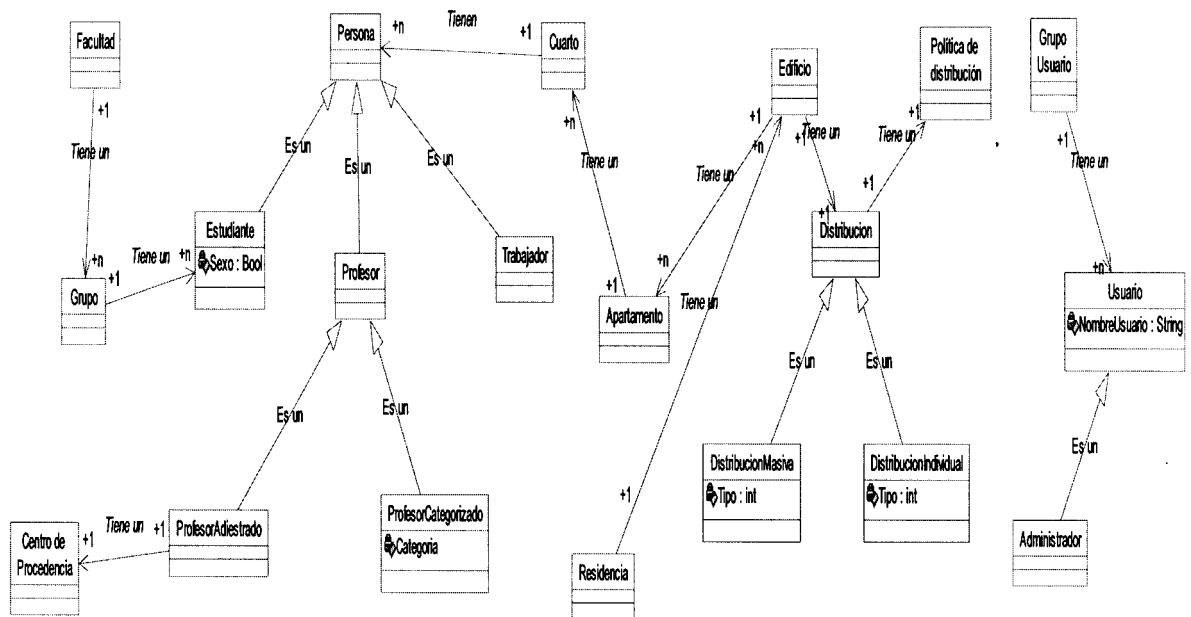
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	4) El sistema muestra la lista de edificios con sus apartamentos para que el usuario seleccione el apartamento al que desea realizarle cambios en el personal que lo habita.
5) El usuario selecciona un apartamento.	6) El sistema muestra un formulario con tres opciones a realizar. Una para eliminar personas del apartamento, la otra para incorporar personas al apartamento y la otra para cambiar personas de apartamento.

<p>7) El usuario selecciona una opción.</p> <p>a) Si la opción seleccionada es eliminarlo del apartamento, véase la sección: <i>desubicar persona</i>.</p> <p>b) Si la opción seleccionada es incorporar persona, véase la sección: <i>Incorporar persona</i>.</p> <p>c) Si la opción seleccionada es cambiarlo de apartamento, véase la sección: <i>cambiar de ubicación</i></p>	
<p>Sección: <i>Incorporar persona</i></p>	
<p><b>Acción del actor</b></p>	<p><b>Respuesta del sistema</b></p>
	<p>8) El sistema muestra un formulario para que se introduzca el nombre de la persona que se quiere incorporar en el apartamento.</p>
<p>9) El usuario introduce el nombre de la persona.</p>	<p>10) El sistema devuelve todas las personas registradas que coinciden con el nombre suministrado para que se escoja uno.</p>

11) El usuario escoge una.

12) El sistema verifica si es posible incorporarlo al apartamento, teniendo en cuenta las características de la persona y las del apartamento. De ser posible lo inserta, sino envía una alerta.

### 3.2 Modelo Conceptual

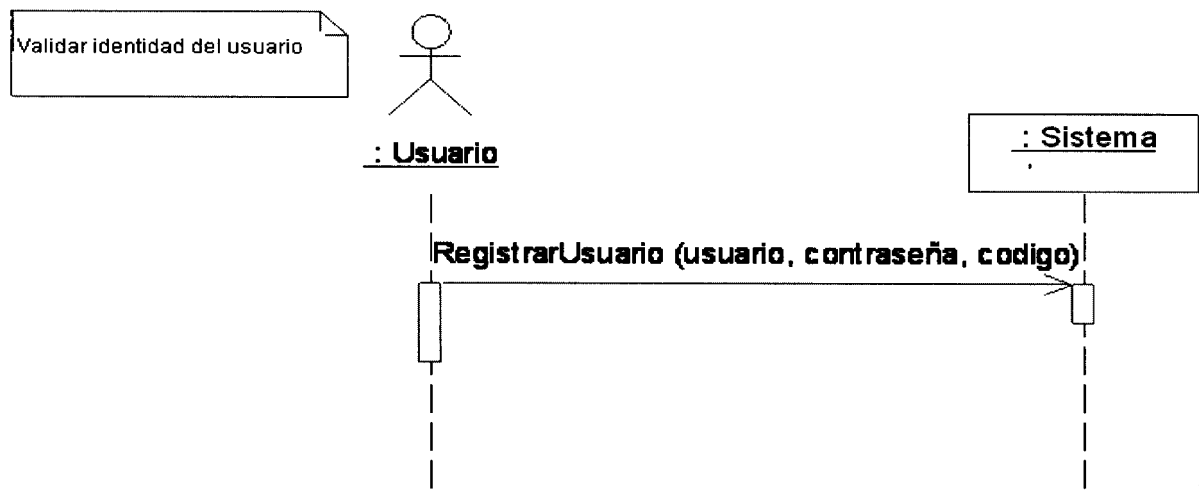




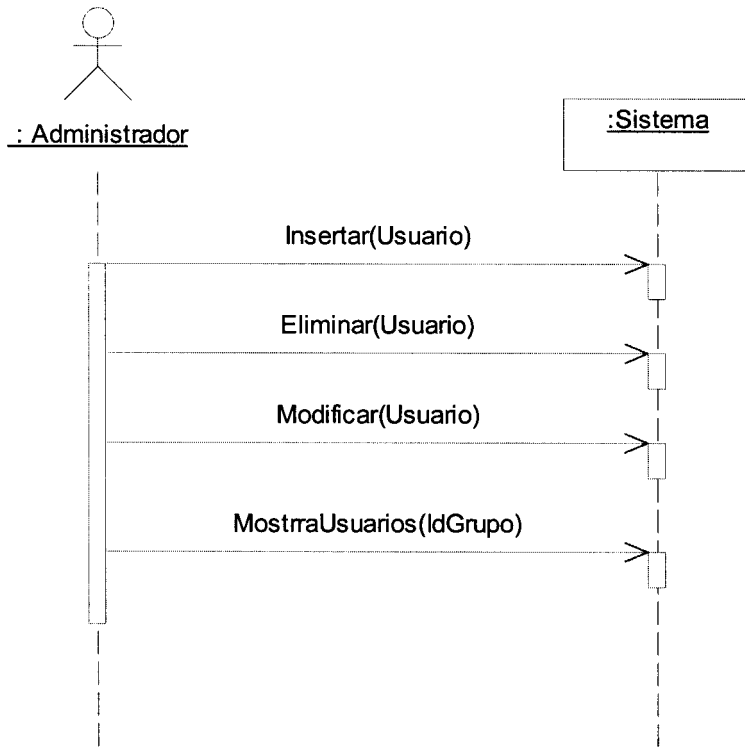
### 3.3 Diagramas de Secuencia

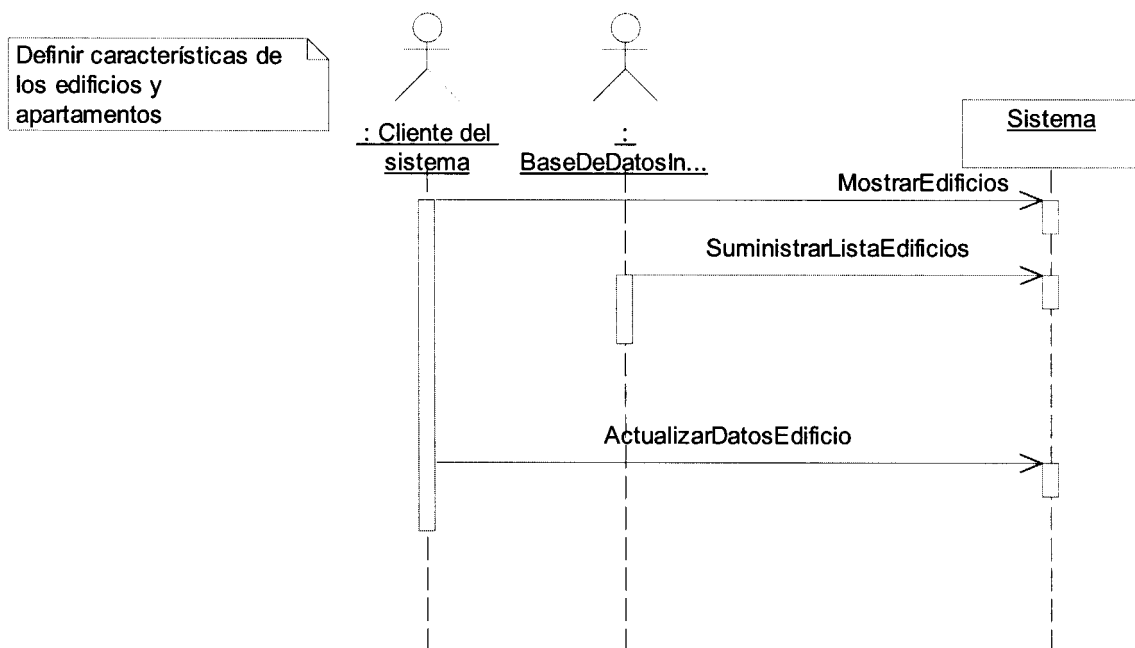
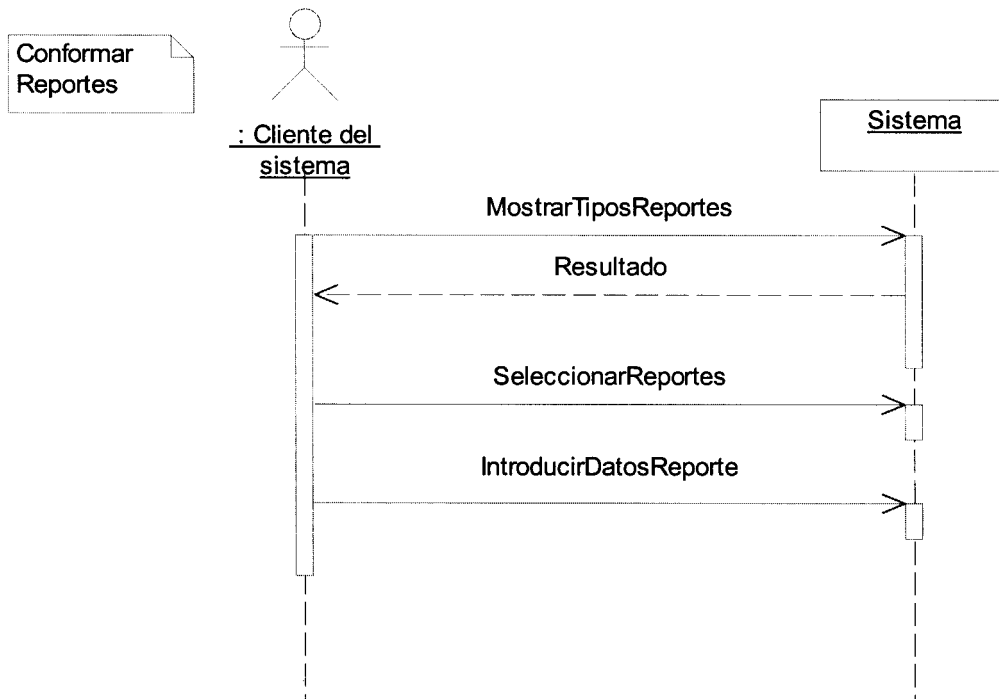
“El diagrama de secuencia es un artefacto de UML que muestra como los objetos se comunican unos con otros para llenar los requerimientos del sistema.” El diagrama de secuencia muestra gráficamente los eventos que fluyen de los actores al sistema.

Se crea un diagrama de secuencia del sistema para cada uno de los casos de uso. En estos diagramas intervienen los actores del caso de uso y un objeto que representa al sistema, y se reflejan los eventos que son enviados al sistema por cada uno de los actores.

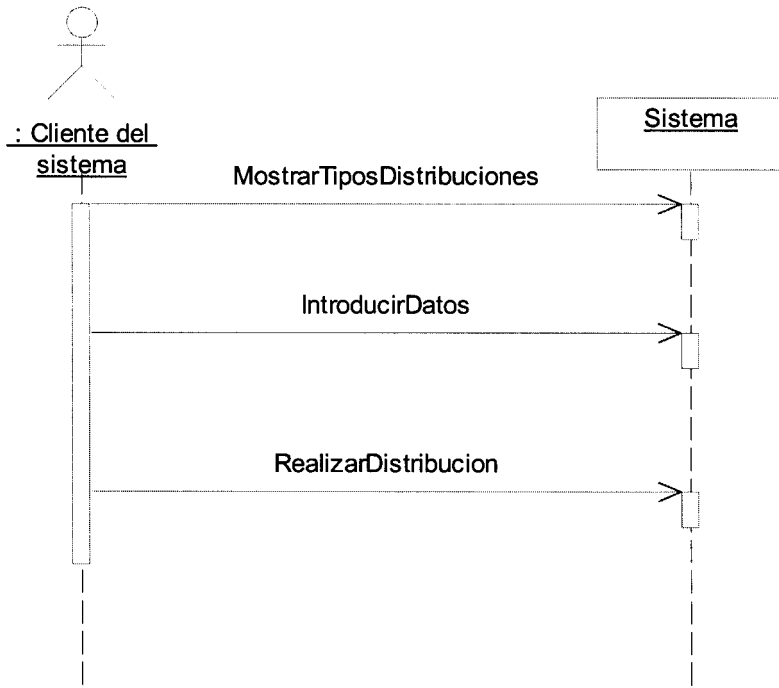


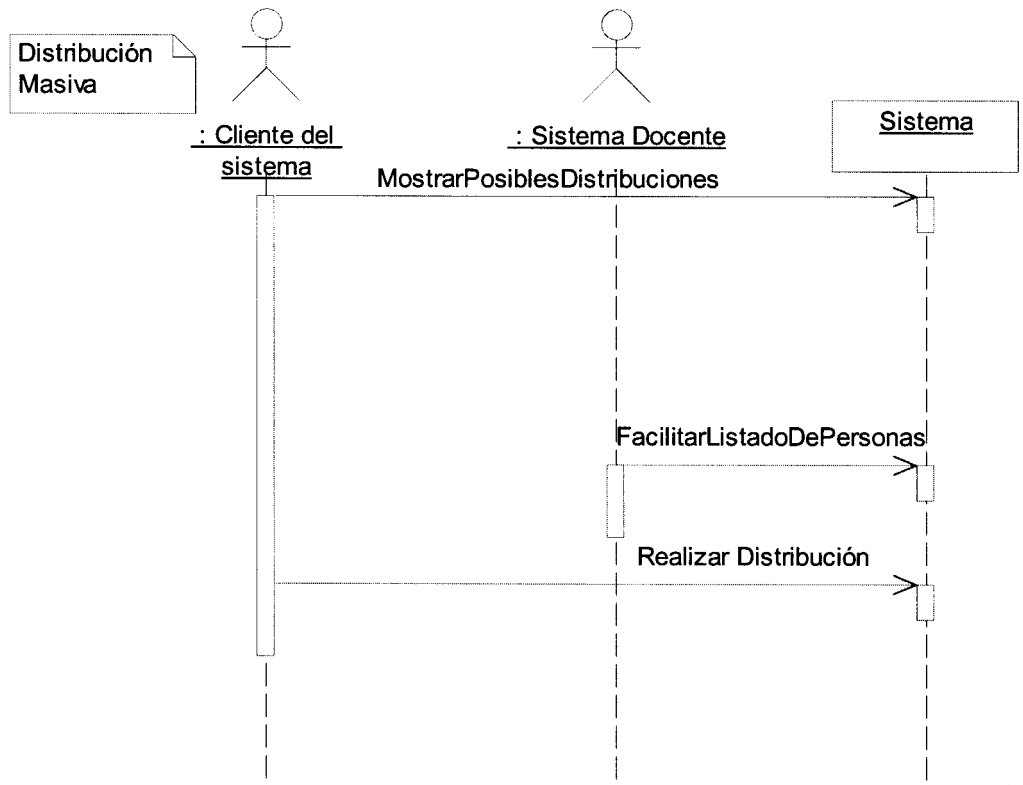
Administrar  
Usuarios

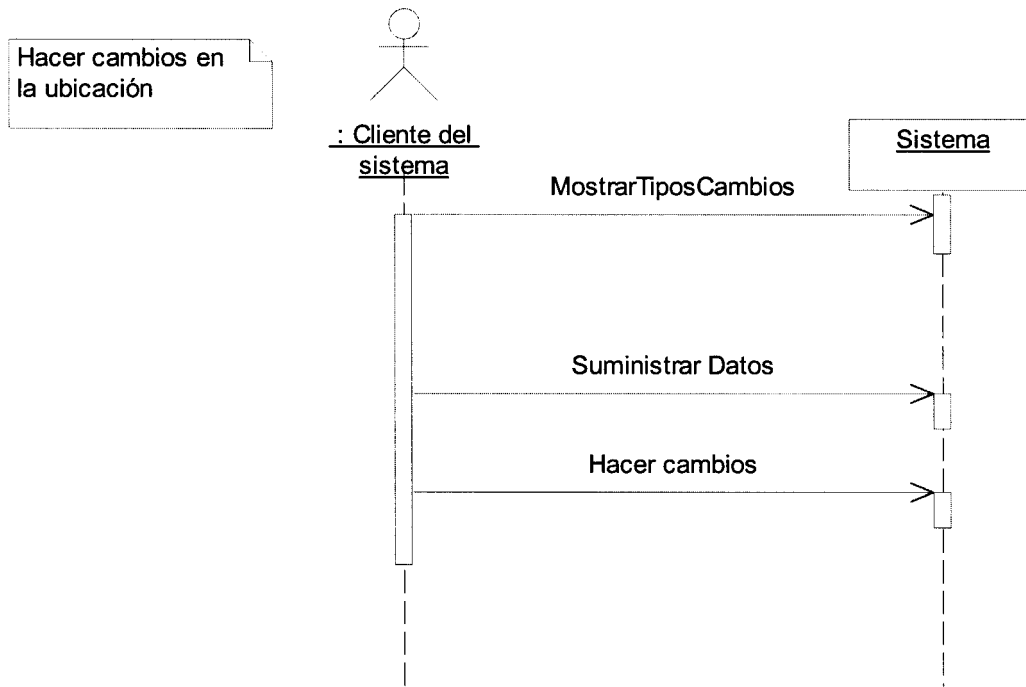




Distribución individual

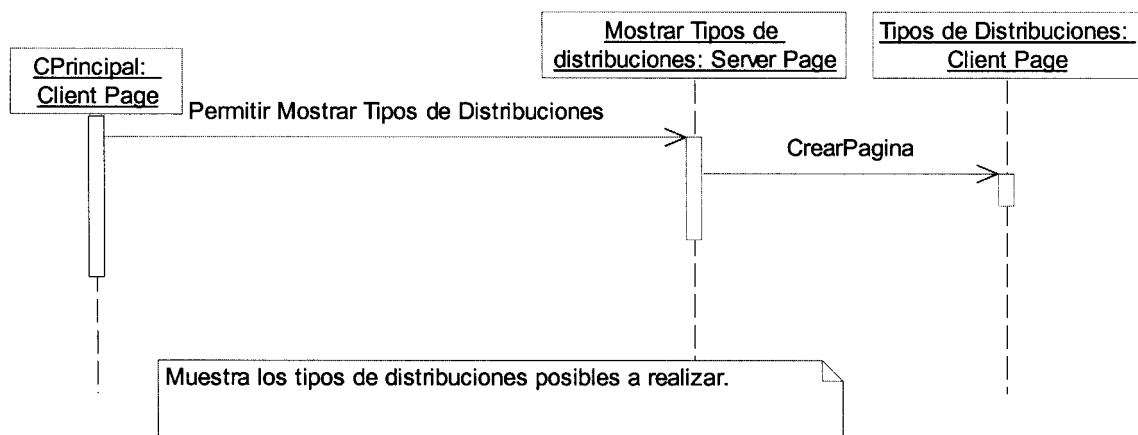


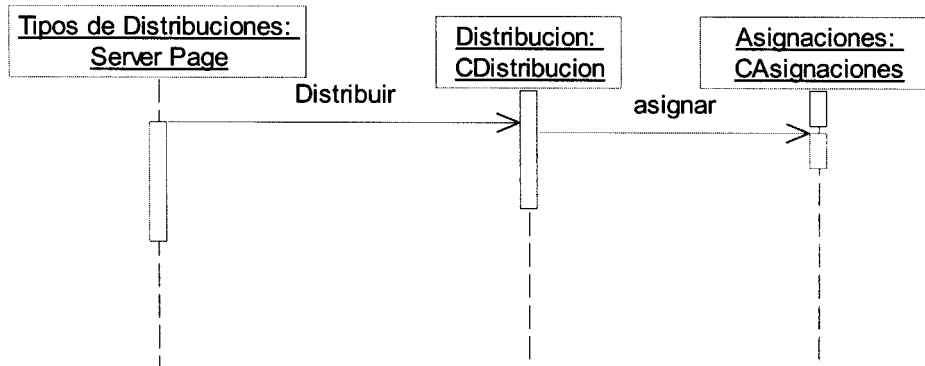
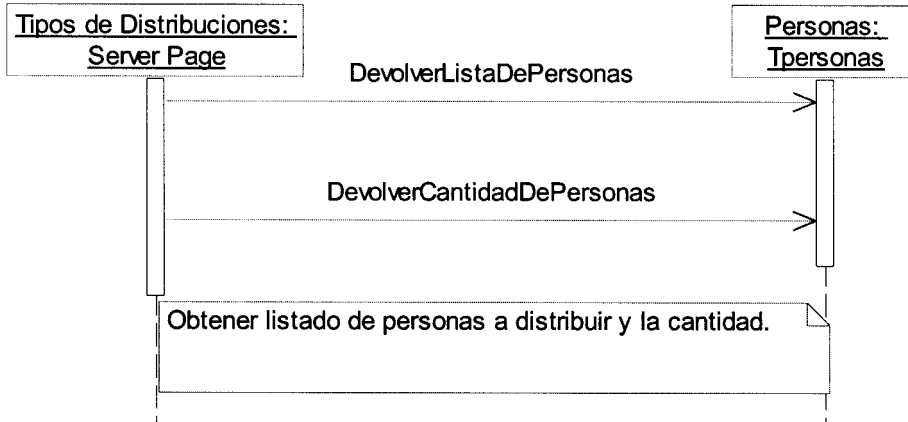




### 3.4 Diagramas de interacción del sistema

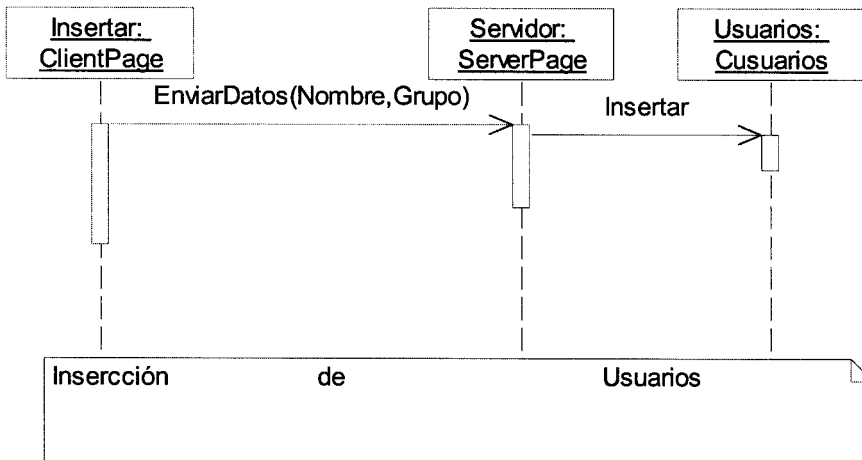
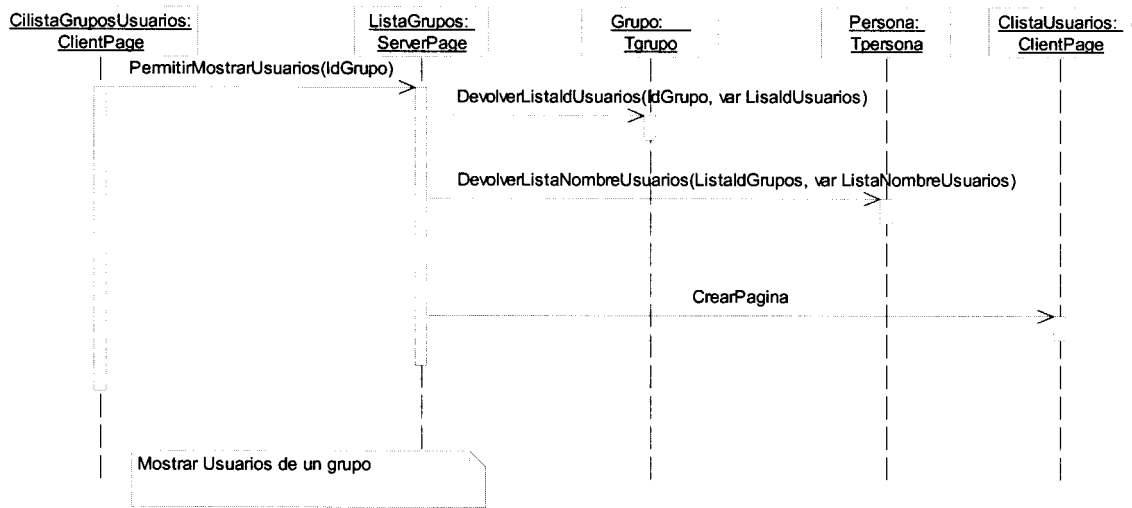
#### 3.4.1 Diagramas de iteración del caso de uso distribución masiva





Distribuir

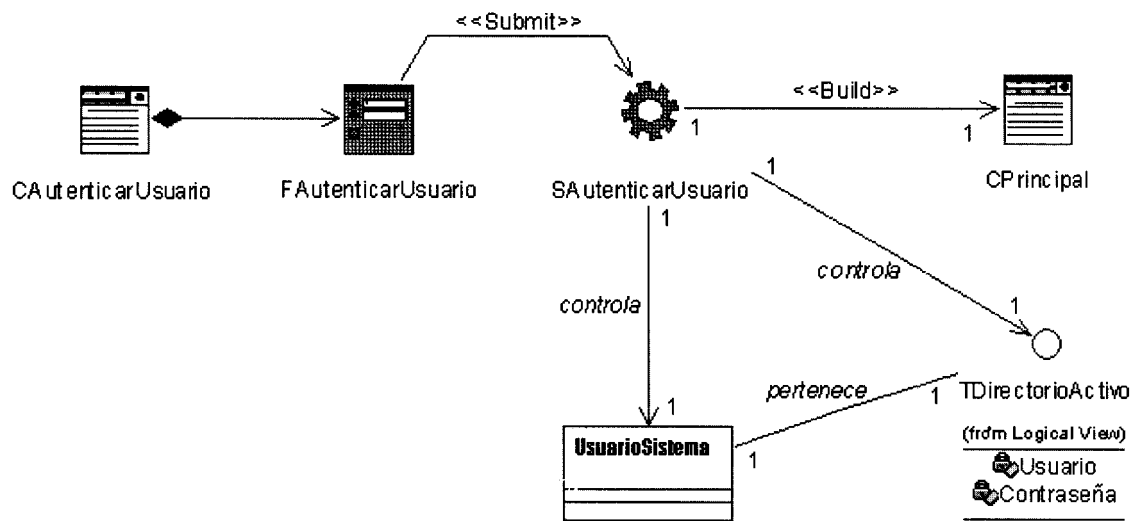
### 3.4.2 Diagramas de iteración del caso de uso Administrar Usuarios



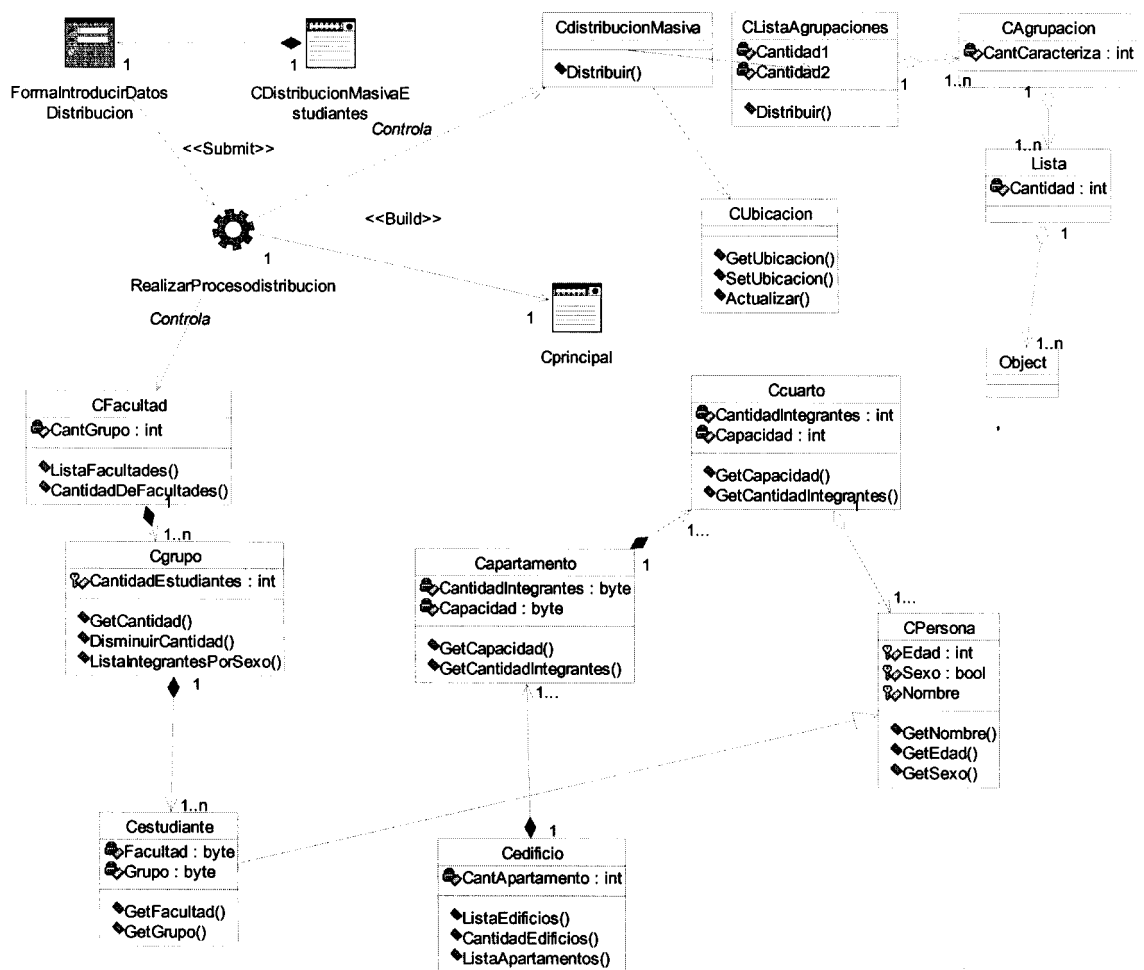


### 3.5 Diagramas del diseño web del sistema

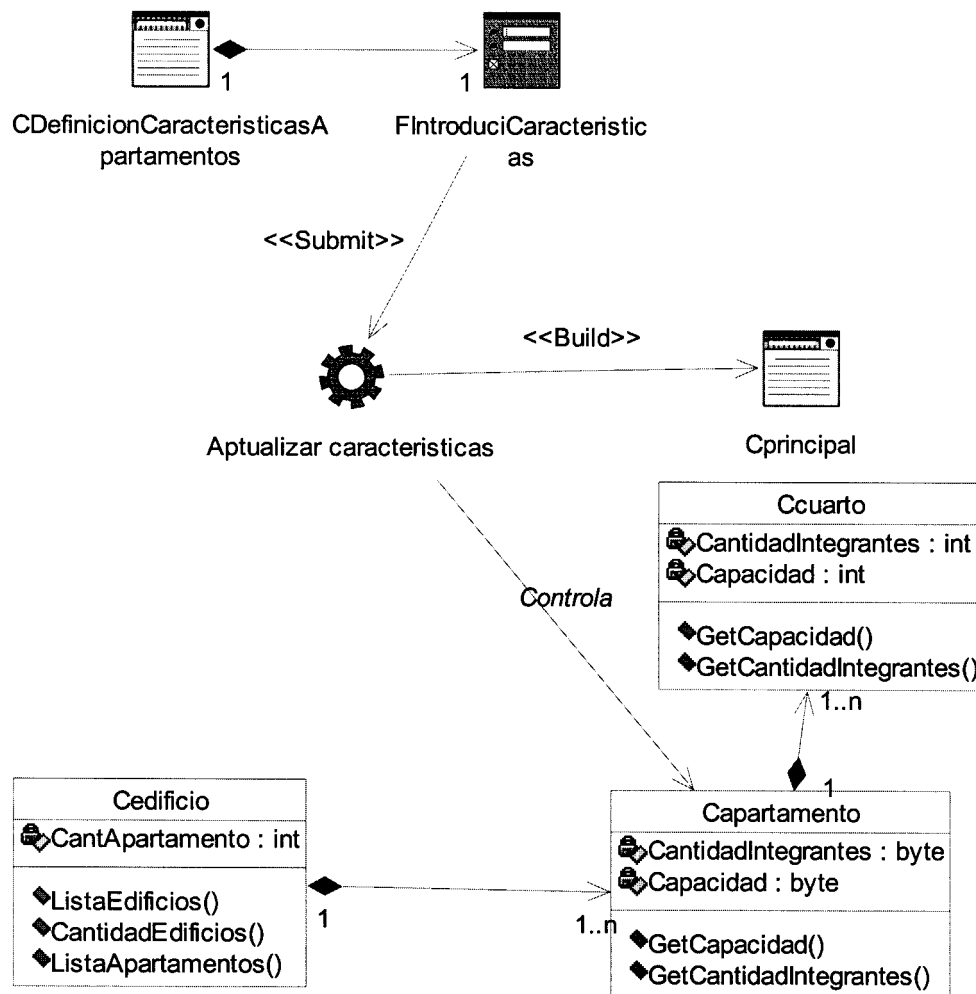
#### Diagrama de clases autentificar usuario



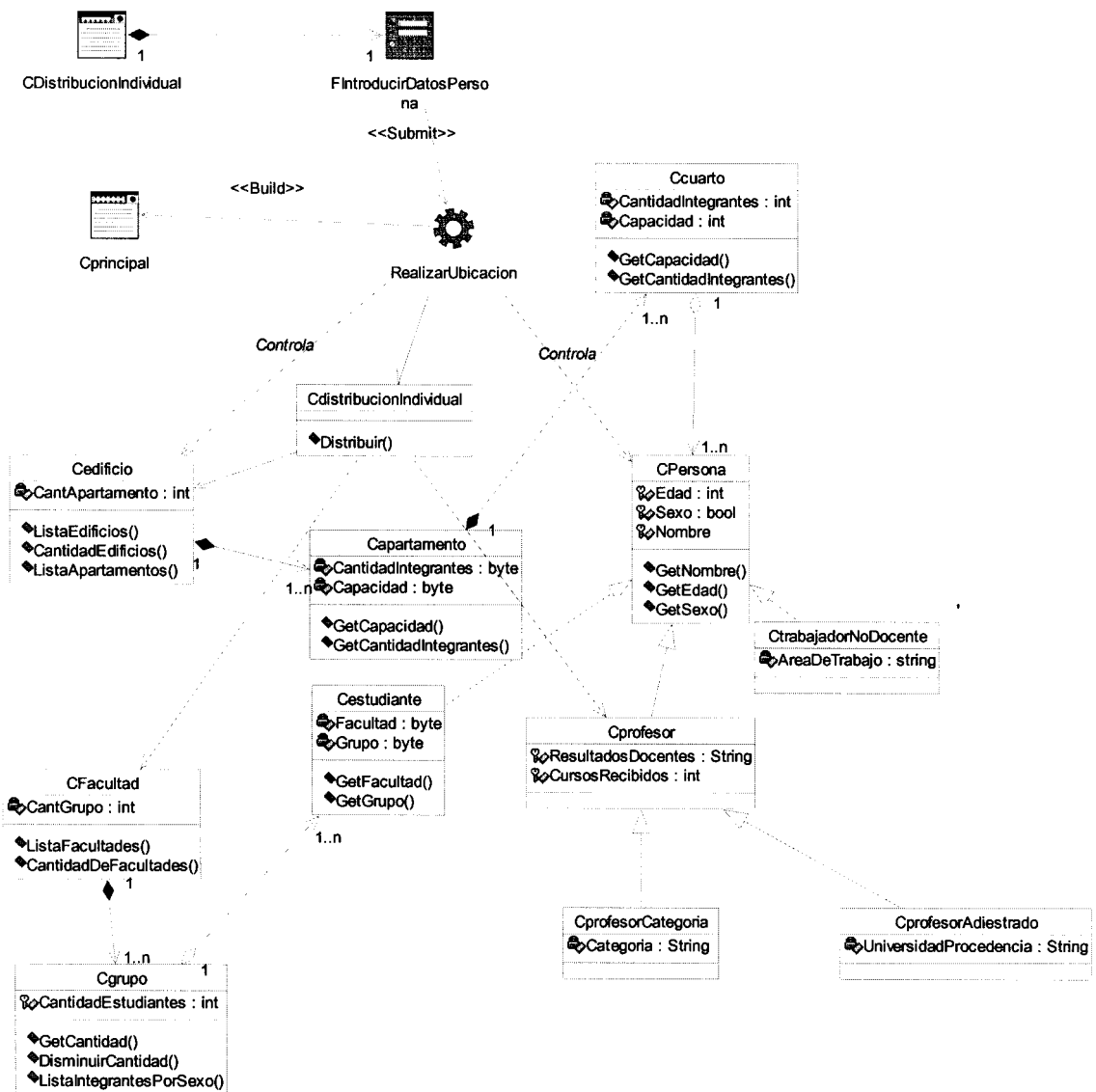
## Diagrama de clases: Distribución masiva de estudiantes.



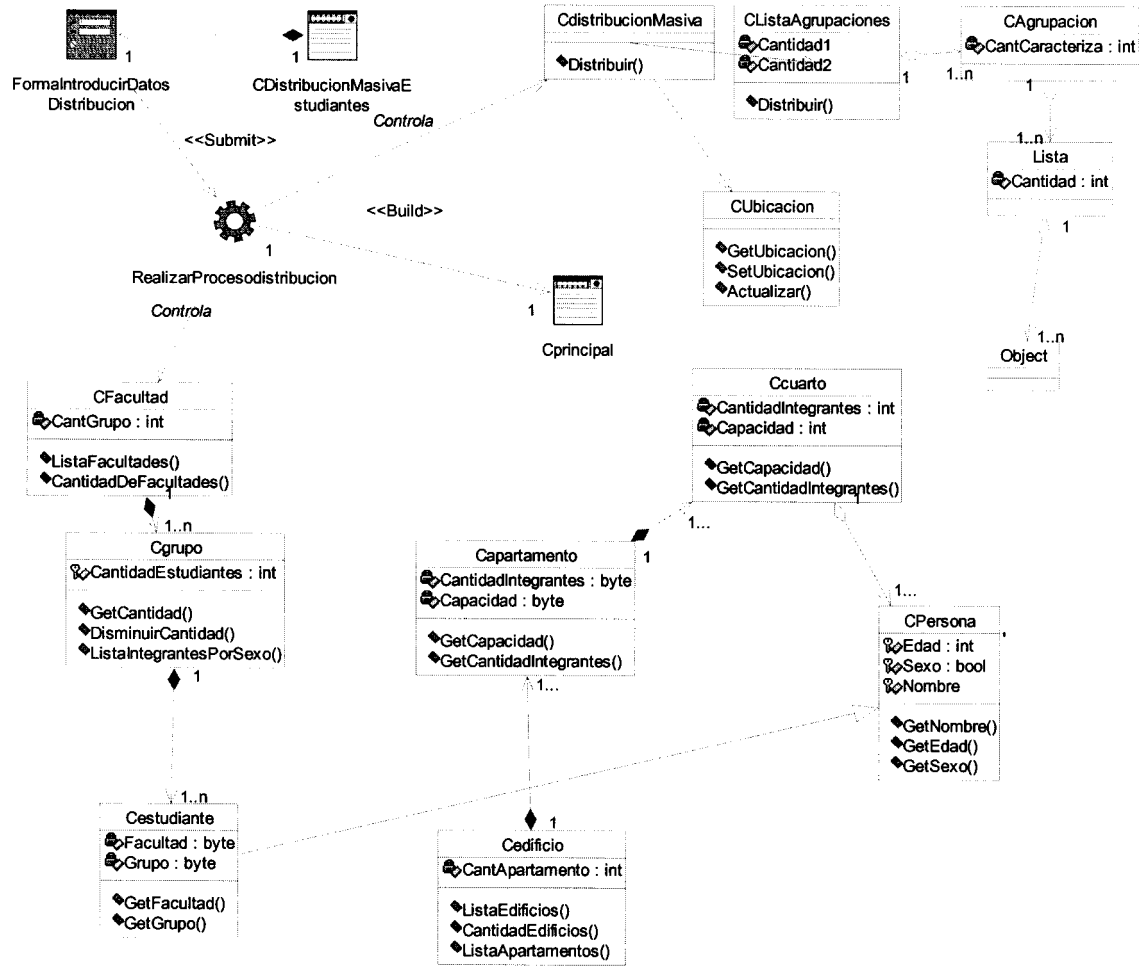
## Diagrama de clases: Definición de características de los apartamentos.



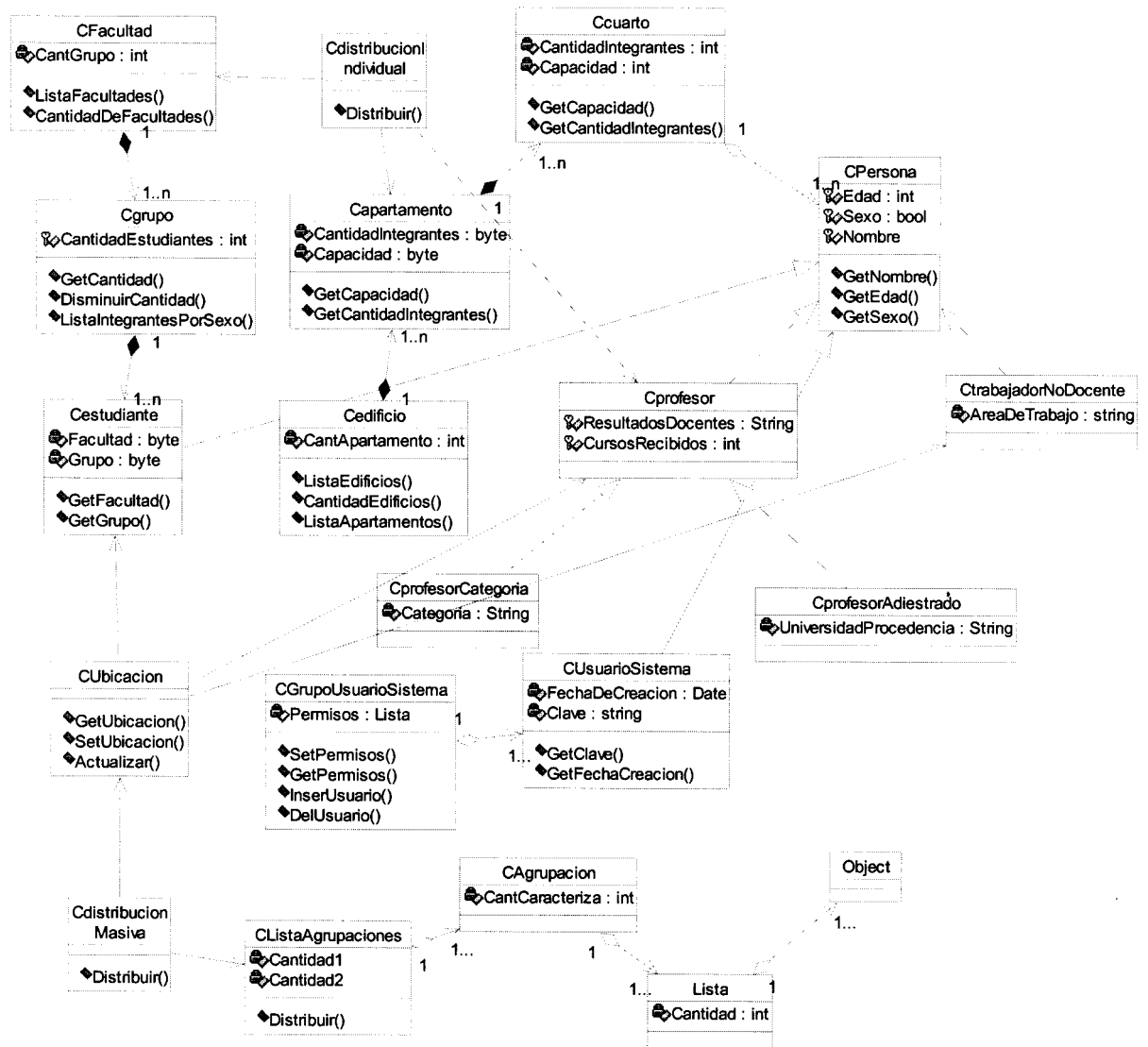
## Diagrama de clase: Distribución individual.



## Diagrama de clases: Distribución masiva de profesores.



## Diagrama de clases



### 3.6 Descripción de las clases

<b>Nombre: CFacultad</b>	
<b>Tipo Clase: Controladora.</b>	
Responsabilidades	
Nombre	ListaFacultades()
Descripción	Devuelve el listado de las facultades.
Nombre	CantidadDeFacultades()
Descripción	Devuelve la cantidad de facultades.

<b>Nombre: Cgrupo</b>	
<b>Tipo Clase: Controladora.</b>	
Responsabilidades	
Nombre	GetCantidad()
Descripción	Devuelve la cantidad de estudiantes.
Nombre	ListaIntegrantesPorSexo(Sexo: Tsexo)
Descripción	Devuelve el listado de los estudiantes de un determinado sexo.

<b>Nombre: Cestudiante</b>	
<b>Tipo Clase: Controladora.</b>	
Responsabilidades	
Nombre	GetFacultad()
Descripción	Devuelve la facultad del estudiante.
Nombre	GetGrupo()
Descripción	Devuelve el grupo del estudiante.

<b>Nombre: CEdificio</b>	
<b>Tipo Clase: Controladora.</b>	
Responsabilidades	
Nombre	ListaEdificios().
Descripción	Devuelve el listado de los edificios.
Nombre	CantidadEdificios()
Descripción	Devuelve la cantidad de edificios.
Nombre	ListaApartamentos()
Descripción	Devuelve el listado de los apartamentos.



<b>Nombre: Capartamento</b>	
<b>Tipo Clase: Controladora.</b>	
Responsabilidades	
Nombre	GetCapacidad()
Descripción	Devuelve la capacidad del apartamento.
Nombre	GetCantidadIntegrantes()
Descripción	Devuelve la cantidad de estudiantes.

<b>Nombre: Cubicacion</b>	
<b>Tipo Clase: Entidad.</b>	
Responsabilidades	
Nombre	GetUbicacion( IdPersona )
Descripción	Devuelve el identificador del apartamento donde está ubicado la persona.
Nombre	SetUbicacion(IdPersona, ubicacion)
Descripción	Escribe en la base de datos del sistema la ubicación.
Nombre	Actualizar()
Descripción	Pasa todas las ubicaciones de la base de datos del sistema para los sistemas a nivel de universidad.

<b>Nombre: CDistribucionIndividual</b>	
<b>Tipo Clase: Controladora.</b>	
Responsabilidades	
Nombre	Distribuir()
Descripción	Se encarga de realizar la distribución individual.

### **3.7 Forma general y principios de la protección y seguridad.**

El sistema analizado debe tener determinado nivel de seguridad ya que alguna modificación indebida en la ubicación de una persona implicaría grandes problemas. Para garantizar la seguridad de la información que se encuentra en las bases de datos, se han creados determinados permisos para el acceso a las funcionalidades del sistema.

En este caso no será necesaria la creación de nuevas cuentas de usuario ya que se podrán utilizar para la autenticación del sistema las cuentas del dominio. Solo es necesario asignar a cada usuario autorizado los permisos o niveles de acceso. Por el momento serán empleados dos niveles: administrador, supervisor.

- ✓ Supervisor: Podrá tener acceso a todos los reportes que brinda el sistema pero no podrá efectuar ningún tipo de cambios en las bases de datos.
- ✓ Administrador: tendrá control total sobre el sistema.

#### **3.7.1 Forma general del tratamiento de errores**

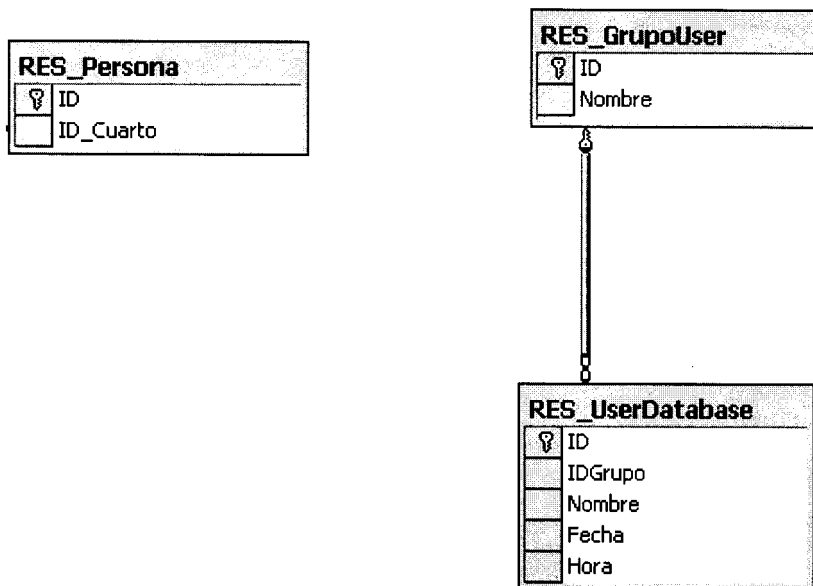
El tratamiento de los errores está referido a aquellos errores que ocurren cuando el usuario interactúa con el sistema al proporcionar el conjunto de datos necesarios para la ejecución del mismo.

Primeramente debe pensarse que una aplicación bien diseñada debe disminuir la posibilidad de la ocurrencia de errores, pero además el sistema debe ser capaz de, una vez que se produce el error, detectarlo y tratar de corregirlo. Por lo tanto es evidente que una de las mejores prácticas es la prevención de los errores, aunque en muchas ocasiones pueden ocurrir, incluso por parte de usuarios que poseen gran experiencia en los procesos que automatiza la aplicación y en el manejo del propio sistema.

Para tratar de evitar los errores que se puedan cometer durante la ejecución de la aplicación, al realizar la operación de inserción se le presenta al usuario los elementos que existen con el objetivo de que no se repitan los mismos. Además para la eliminación o la modificación de elementos se visualiza el conjunto de datos de los cuales el usuario podrá seleccionar el elemento específico que intervendrá en la ejecución de las operaciones antes mencionadas. Esta concepción disminuye la búsqueda de elementos introducidos por teclado, con el objetivo de evitar la ocurrencia de errores por una rápida e indebida manipulación de la entrada de datos.

En el caso de que no se puedan evitar los errores, es necesario lograr minimizar las consecuencias futuras que pueden acarrear y ayudar a los diferentes usuarios a recuperarse ante esta situación. Para garantizar esto, se validan las entradas de datos de forma tal que si el dato entrado no coincide con el formato debido, este error es detectado por el sistema, visualizándose en pantalla, una caja de diálogo de mensaje para informarle al usuario el formato o tipo del dato requerido. Además, se muestra un mensaje de error cuando se trata de insertar un elemento que ya existe, para notificarle al usuario el motivo por el cual no se puede efectuar la inserción solicitada. Posteriormente, cuando se obtiene la confirmación de lectura del mensaje de error por parte del usuario, desaparece la caja de diálogo mostrada y la aplicación continúa con su ejecución.

### 3.8 Diseño de la base de datos.



Nombre: RES_Persona		
<b>Descripción:</b> Contiene el identificador de la persona en la base única del ciudadano y el identificador del apartamento donde está ubicado.		
Nombre	Tipo	Descripción
ID(llave)	Bigint	Identificador de la persona en la base única del ciudadano.
ID_Cuarto	Bigint	Identificador del cuarto donde está ubicado

<b>Nombre: RES_GrupoUser</b>		
<b>Descripción:</b> Contiene los grupos de usuarios que existen en el sistema.		
<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
ID(llave)	smallint	Identificador del grupo.
Nombre	nvarchar	Nombre del grupo.

<b>Nombre: RES_UserDatabase</b>		
<b>Descripción:</b> Contiene los usuarios del sistema.		
<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
ID(llave)	int	Identificador del usuario.
IDGrupo	smallint	Identificador del grupo al que pertenece.
Nombre	nvarchar	Nombre de usuario.
Fecha	datetime	Fecha en que se creó la cuenta.
Hora	datetime	Hora en que se creó la cuenta.

# Capítulo 4

## Implementación del Sistema

### Introducción

En este capítulo sólo se abordará lo referido al diagrama de despliegue. El cual nos permitirá tener una idea de las relaciones entre los componentes de hardware y software.

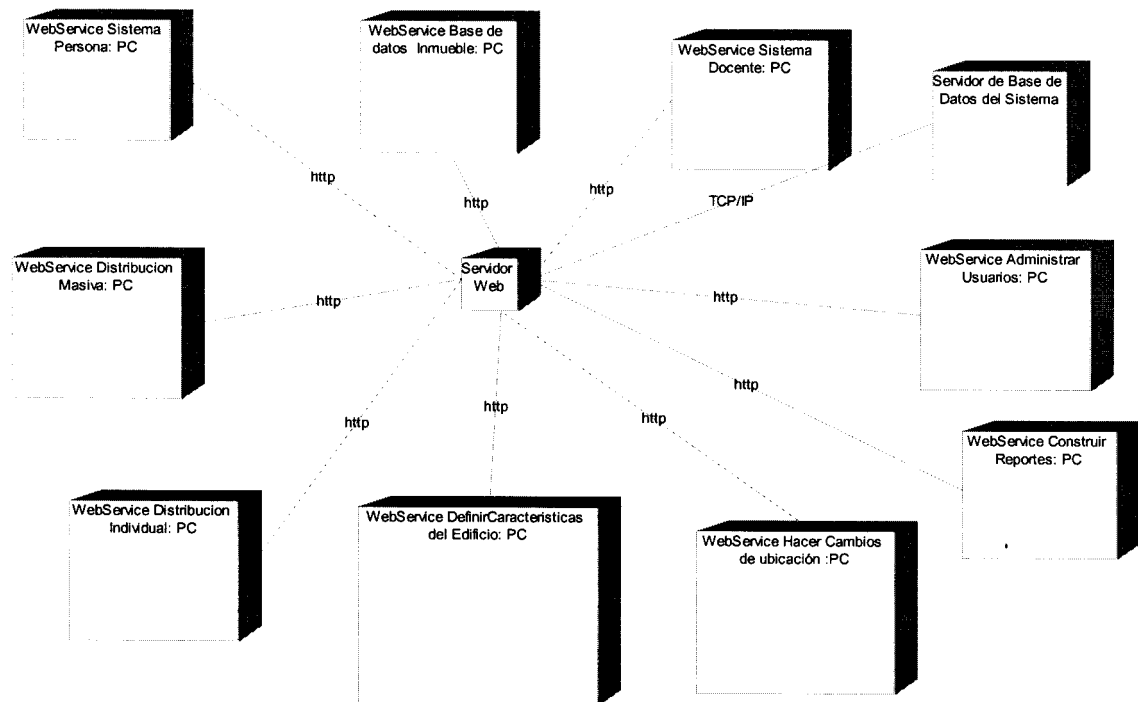
### 4.1 Diagrama de Despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes **hardware** y **software** en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes **software** (procesos y objetos que se ejecutan en ellos).

Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación. En general un nodo será una unidad de computación de algún tipo, desde un sensor a un *mainframe*.

Un nodo es: un objeto físico en tiempo de ejecución que representa un recurso computacional, generalmente con memoria y capacidad de procesamiento.

## 4.2.1 Diagrama de despliegue



## Conclusiones

Podemos concluir que con la realización del diagrama de despliegue llegamos a una representación gráfica de las relaciones entre los diferentes componentes del sistema.

---

## ***Conclusiones***

Con el estudio realizado se ha confirmado que es verdaderamente necesaria la existencia de un software que automatice el proceso gestión del alojamiento en la residencia, solucionando los problemas actuales. Con este trabajo se ha logrado el objetivo trazado: llegar hasta el primer ciclo de desarrollo de un software capaz de gestionar las ubicaciones del personal, reduciendo en gran medida el volumen de información en papel, así como los errores introducidos por las personas que intervienen en el proceso.

El sistema resultante logrará un procesamiento óptimo de la información y disponibilidad de la misma en todo momento para las personas autorizadas. Además será punto de partida para la incorporación de nuevas funcionalidades. También permitirá la interacción con otros sistemas teniendo acceso a la información actualizada a través de servicios web y evitando duplicidad de información en distintas áreas dentro de la universidad.



Se recomienda:

- Realizar la documentación referida al caso de uso del segundo ciclo de desarrollo, para profundizar en el proceso de gestión de los usuarios del sistema y la generación de los reportes.
- Para futuras versiones tratar de optimizar un poco más el algoritmo a seguir en el proceso de distribución.
- Continuar con la investigación para aumentar las funcionalidades del sistema, con el objetivo de obtener nuevas mejoras en futuras versiones del mismo.

## **BIBLIOGRAFÍA**

1. Tutoriales de Microsoft ASP.NET  
<http://es.gotdotnet.com/quickstart/aspplus/doc/wsbehavior.aspx> (02/03/2004)
2. [PDF] Introducción a Web Services con herramientas de desarrollo ... (20/05/2004)
3. <http://www.webservice.com.mx/> (15/05/2004)
4. Craig Larman. UML y Patrones, segunda edición.
5. Tutorial de UML - Casos de Uso: <http://www.dcc.uchile.cl/~psalinas/uml/casosuso.html>
6. La Ingeniería de Software (10/06/2004)  
<http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html>
7. <http://geneura.ugr.es/~maribel/xml/introduccion/index.shtml> (24/06/2004)

## **GLOSARIO DE TÉRMINOS Y SIGLAS**

**Mainframe:** Supercomputadora con elevadas prestaciones en término de capacidad de cálculo y de memoria, usado en las redes como punto central de distribución.

**Software:** (componentes lógicos, programas, software). -- Programas o elementos lógicos que hacen funcionar un ordenador o una red, o que se ejecutan en ellos, en contraposición con los componentes físicos del ordenador o la red.

**Http:** (HyperText Transfer Protocol; Protocolo de transferencia de hipertexto): Es un protocolo del nivel de aplicación usado para la transferencia de información entre sistemas, de forma clara y rápida. Ha sido usado por el World-Wide Web desde 1990.

**Internet:** Red de redes. Sistema mundial de redes de computadoras interconectadas. Fue concebida a fines de la década de 1960 por el Departamento de Defensa de los Estados Unidos; más precisamente, por la ARPA. Se la llamó primero ARPAnet y fue pensada para cumplir funciones de investigación. Su uso se popularizó a partir de la creación de la World Wide Web . Actualmente es un espacio público utilizado por millones de personas en todo el mundo como herramienta de comunicación e información.

**Red:** en tecnología de la información, una red es un conjunto de dos o más computadoras interconectadas.