

**UNIVERSIDAD MAYOR DE SAN ANDRES  
FACULTAD DE CIENCIAS PURAS Y NATURALES  
CARRERA DE INFORMATICA**



**PROYECTO DE GRADO**

**SISTEMA DE REGISTRO Y SEGUIMIENTO DE CORRESPONDENCIA Y  
ASIGNACION DE AULAS  
PARA EL CENTRO DE MULTISERVICIOS EDUCATIVOS CEMSE - CERPI**

PARA OPTAR AL TITULO DE LICENCIATURA EN INFORMATICA  
MENCIÓN: INGENIERIA DE SISTEMAS INFORMÁTICOS

**Postulante:** Silvia Gladys Huanca Rengel

**Tutor:** Lic. Mario Loayza Molina M.S.c.

**Revisor:** Lic. Javier Reyes Pacheco

**LA PAZ – BOLIVIA**

**2008**

## **DEDICATORIA**

*Este trabajo esta dedicado...*

*A mis padres Genaro e Isidora,  
como una forma de agradecimiento por su  
cariño, comprensión y apoyo a lo largo de  
mi carrera y de mi vida.*

## AGRADECIMIENTOS

*Deseo expresar mi agradecimiento y cariño a aquellas personas que durante todo este tiempo han sido mi apoyo, mi fuente de aprendizaje, mi motivación para seguir adelante con este proyecto de grado:*

- ❖ Agradecer primeramente a Dios por haberme dado fortaleza, salud y voluntad para culminar el presente proyecto de grado.*
- ❖ A mi Tutor Lic. Mario Loayza Molina por su colaboración y orientación en la realización de este proyecto.*
- ❖ A mi Revisor Lic. Javier Reyes Pacheco por toda su paciencia y tiempo que le brindo al revisar este proyecto.*
- ❖ A todas las personas de la Carrera de Informática que me brindaron su colaboración en cada momento.*
- ❖ Al CEMSE-CERPI, por colaborar y brindarme su amistad.*
- ❖ A mi familia: Isidora, Genaro, Toño, Lucho, Fanny y Milenka por su preocupación y apoyarme moralmente en todo momento.*
- ❖ A mis compañeros de la universidad los Pokaritos.*
- ❖ A mis compañeros de trabajo por aconsejarme en todo momento con su experiencia.*
- ❖ A todos mis amigos, en especial a Grover, Blanca y Hugo por su apoyo moral y animo para que culmine mi proyecto.*

**¡MUCHAS GRACIAS!**

# INDICE DE CONTENIDOS

---

## CAPITULO 1 PRESENTACION

1.1	Introducción.....	1
1.2	Antecedentes.....	2
1.2.1	Antecedentes de la Institución.....	2
1.2.2	Antecedentes del Proyecto.....	3
1.3	Planteamiento del Problema.....	4
1.3.1	Análisis del Problema.....	4
1.3.2	Definición del Problema.....	5
1.4	Objetivos.....	6
1.4.1	Objetivo General.....	6
1.4.2	Objetivos Específicos.....	6
1.5	Justificación.....	7
1.5.1	Justificación Social.....	7
1.5.2	Justificación Tecnológica.....	7
1.5.3	Justificación Económica.....	8
1.6	Alcances.....	8
1.7	Aportes.....	8
1.8	Metodología y Herramientas.....	8

## CAPITULO 2 MARCO TEORICO

2.1	Enfoque Orientada a Objetos.....	10
2.1.1	Conceptos Básicos de la Orientación a Objetos.....	10
2.1.2	Análisis Orientado a Objetos.....	11
2.1.3	Diseño Orientado a Objetos.....	12
2.2	Metodología Técnica de Modelado de Objetos OMT.....	12
2.3	Lenguaje Unificado para la Construcción de Modelos UML.....	15
2.3.1	Modelos.....	16
2.3.1.1	Diagrama de Estructura Estática.....	18

2.3.1.2	Diagrama de Casos de Uso.....	25
2.3.1.3	Diagramas de Interacción.....	26
2.3.1.4	Diagrama de Estados.....	29

### CAPITULO 3 GESTION DOCUMENTAL

3.1	Dato.....	31
3.2	Información.....	31
3.3	Documento.....	32
3.3.1	Documento Publico.....	32
3.3.2	Documento de Archivo.....	33
3.3.3	Documentaciones Publicas.....	33
3.4	Teoría del Valor Documental.....	33
3.4.1	Valores Primarios.....	34
3.4.2	Valores Secundarios.....	34
3.4.3	Valor Informativo.....	35
3.4.4	Valor Histórico.....	35
3.4.5	Valores Abstractos.....	36
3.4.6	Valor Probatorio.....	36
3.4.7	Valor Archivistito.....	36
3.5	Clasificación de la Correspondencia.....	37

### CAPITULO 4 MARCO APLICATIVO

4.1	Análisis de Objetos.....	39
4.1.1	Descripción del Problema.....	40
4.1.2	Modelo de Objetos.....	45
4.1.3	Modelo Dinámico.....	54
4.1.4	Modelo Funcional.....	56
4.2	Diseño.....	60
4.2.1	Diseño de Sistema.....	60
4.2.2	Diseño de Objetos.....	65

4.3	Implementación.....	66
4.3.1	Codificación.....	67
4.3.2	Diseño de la Interfaz.....	68

## CAPITULO 5 CALIDAD DEL SOFTWARE

5.1	Introducción.....	74
5.2	Confiabilidad.....	74
5.3	Funcionalidad.....	76
5.4	Portabilidad.....	78
5.5	Rendimiento.....	79
5.6	Mantenibilidad.....	79
5.7	Análisis Costo-Beneficio del Sistema.....	80
5.7.1	Costo de Análisis de Programación.....	80
5.7.2	Costos de Capacitación del Usuario.....	82
5.7.3	Costo de software de desarrollo y hardware.....	82
5.7.4	Costo de Instalación.....	82

## CAPITULO 6 CONCLUSIONES Y RECOMENDACIONES

6.1	Conclusiones.....	83
6.2	Recomendaciones.....	85

	<b>Referencias Bibliográficas.....</b>	<b>86</b>
--	--	-----------

**Anexo.**

## INDICE DE FIGURAS

---

Figura		Página
Figura2.1	Ciclo de vida OMT.....	13
Figura2.2	Historia de UML.....	16
Figura2.2	Ejemplo de nota.....	17
Figura2.3	Dependencias.....	18
Figura2.4	Notación para clases a distintos niveles de detalle.....	19
Figura 2.5	Ejemplos de objetos.....	19
Figura 2.6	Ejemplo de asociación con nombre y dirección.....	20
Figura 2.7	Ejemplos de multiplicidad en asociaciones.....	21
Figura 2.8	Ejemplo de roles en una asociación.....	21
Figura 2.9	Ejemplo de agregación.....	22
Figura 2.10	Ejemplo de clase asociación.....	22
Figura 2.11	Ejemplo de asociación ternaria.....	23
Figura 2.12	Ejemplo de herencia.....	24
Figura 2.13	Ejemplo de Atributo derivado.....	24
Figura 2.14	Diagrama de Casos de Uso.....	26
Figura 2.15	Diagrama de Secuencia.....	27
Figura 2.16	Diagrama de Colaboración.....	28
Figura 2.17	Diagrama de Estados.....	30
Figura. 4.1	Análisis de Objetos OMT.....	39
Figura. 4.2	Caso de Uso del Sist, de Registro, Seguimiento de Correspondencia y asignación de Aula.....	42
Figura: 4.3	Caso de uso Sistema de Registro y Seguimiento de Correspondencia.....	43
Figura: 4.4	Caso de uso Subsistema de Asignación de Aula.....	44
Figura. 4.5	Caso de Uso de Agenda.....	45
Figura: 4.6	Clases Identificadas.....	46
Figura 4.7	Agregación entre objetos del sistema .....	49

Figura 4.8	Diagrama de Relaciones y Asociaciones.....	50
Figura 4.9	Objeto de clases con atributos y operaciones.....	51
Figura 4.10	Herencia de la clase funcionario.....	52
Figura 4.11	Diagrama de Clases del Sistema de Registro y Seguimiento de Correspondencia y asignación de Aulas.....	53
Figura 4.12	Diagrama de secuencia para el seguimiento de correspondencia.....	54
Figura 4.13	Diagrama de secuencia de solicitud de aula.....	55
Figura 4.14	Diagrama de estado para Seguimiento de correspondencia.....	55
Figura 4.15	Diagrama de estado para la Asignación de aula.....	56
Figura 4.16	Diagrama de contexto del sistema.....	58
Figura 4.17.	Diagrama de Flujo de Datos de alto nivel de todo el sistema.....	59
Figura 4.18	Transformación de Objeto a Código.....	62
Figura 4.19	Transformación de la clase Correspondencia a la tabla correspondencia.....	63
Figura 4.20	Transformación de la clase correspondencia a la tabla correspondencia.....	65
Figura 4.21	Transformación de la clase usuario a la tabla usuario.....	66



Figura 4.22	Código MySQL para la implementación de la tabla correspondencia.....	67
Figura 4.23	Código MySQL para la implementación de la tabla usuario.....	67
Figura 4.24	Diseño de interfaz de la autenticación de usuario.....	68
Figura 4.25	Diseño de interfaz del Menú Principal.....	69
Figura 4.26	Diseño de interfaz del Menu Registro de Correspondencia.....	70
Figura 4.27	Formulario de Listado de Correspondencia Registrada.....	70
Figura 4.28	Formulario de Redacción de Correspondencia.....	71
Figura 4.29	Diseño de interfaz del Menú Usuario.....	71
Figura 4.30	Listado de los diferentes usuarios habilitados para manejar el sistema.....	72
Figura 4.31	Formulario Vista de Horario en la semana correspondiente....	72

## RESUMEN

El presente proyecto de grado, realizo los estudios en una solución de la denominada Gestión Documental el cual nos permite conservar el patrimonio documental en una institución, es así como se estudio en el CEMSE-CERPI (Centro de Multiservicios Educativos-Centro de Recursos Pedagógicos “Luís Espinal”). En esta Institución no se contaba con ningún libro de registro de entrada/salida de la Correspondencia, y es por esta razón principal que se ha realizado el presente proyecto.

En este sentido se presenta un sistema de registro y seguimiento y asignación de aulas para el Centro de Multiservicios Educativos CEMSE-CERPI, que brinda un entorno amigable y ayudara a organizar mejor el manejo de la información de correspondencia.

En el mismo se ha realizado el análisis de requerimientos, para encontrar las necesidades de la institución.

El Análisis y Diseño se baso en la metodología OMT (Técnica de Modelado de Objetos) apoyado por la herramienta UML.

Para la implementación se uso el Lenguaje de Programación PHP y el Gestor de Base de Datos MySQL.

Para medir la calidad del Software se uso la Norma ISO 9126 que considera los siguientes atributos: confiabilidad, funcionalidad, portabilidad, rendimiento y mantenibilidad.

# 1. PRESENTACIÓN

## 1.1 INTRODUCCIÓN

Las aplicaciones de sistemas de información tienen su origen en casi todas las áreas de una empresa o institución y están relacionadas con todos los problemas de la organización de documentos, información y otros datos importantes.

Uno de los principios fundamentales en el desarrollo de los sistemas de información por las empresas, es que las aplicaciones son una herramienta<sup>1</sup> y no un instrumento que debe tenerse para utilizar la tecnología de la información; en consecuencia los sistemas de información deben desarrollarse sobre la base de su propia capacidad para mejorar el desempeño de la organización, sin embargo estas razones no significan únicamente pérdidas y ganancias. La marcha de una empresa incluye el manejo de la información a tiempo que beneficie a la institución y su entorno.

En el transcurso de los años, se han logrado avances en la automatización de procesos de las diferentes empresas o instituciones (ej: administración de RRHH, facturación, contabilidad y todo manejo de información); los cuales proporcionan ayudas en el cumplimiento de las funciones dentro de una organización, sobre todo en esta época en la cual se necesita ahorrar tiempo en la obtención de la información, así como también, se requiere el control de los procesos.

Con este propósito se realizó este Proyecto, de manera tal que pueda solventar y mejorar los procesos manuales que se llevan a cabo en el Centro de Multiservicios Educativos CEMSE-CERPI<sup>2</sup>; el cual presenta retrasos, perdidas y equivocaciones a la hora del envío y recepción de correspondencia así como la administración de

---

<sup>1</sup> Herramienta: Aplicación empleada para la construcción (de ahí su nombre) de otros programas o aplicaciones

<sup>2</sup> CEMSE -CERPI Centro de Multiservicios Educativos –Centro de Recursos Pedagógicos

la solicitudes de uso de aulas, las cuales se evidencio mediante los procesos de observación y encuestas realizadas al personal correspondiente.

Detectando como posibles fallas, las humanas, ya que el personal que se encarga de cumplir con las funciones en las diferentes direcciones, llevan los procesos de forma manual, y teniendo problemas al momento de ordenar por ubicación las cartas que van hacer entregadas a las diferentes direcciones que correspondan en la institución.

Motivado a esta situación, se planteó su mejora, lo cual se quiere demostrar mediante este Proyecto, el que un sistema automatizado puede ayudar a obtener un mejor control de las actividades que se llevan a cabo en una institución y lograr así, una mejor calidad de los procesos que se realicen mediante su utilización.

En el cuerpo de trabajo, se ha elaborado el planteamiento del problema, de tal manera que identifica detalladamente la situación a estudiar. En él se han expuesto los objetivos o propósito de la investigación, la justificación del por qué la realización del proyecto y los alcances y delimitaciones, en la cual se destaca la relevancia del tema y de la investigación.

## **1.2 ANTECEDENTES**

### **1.2.1 Antecedentes de la institución.-**

El CEMSE <sup>3</sup> es una institución de la Iglesia Católica, dependiente de la Compañía de Jesús, con personería jurídica N° 32100 del 15 de febrero de 1949, bajo Resolución Ministerial N° 695 del 31 Julio de 1985, comenzando actividades el 14 de mayo de 1986. Su funcionamiento esta amparado bajo el Convenio Marco Estado Iglesia.

## **MISION**

---

<sup>3</sup> CEMSE Centro de Multiservicios Educativos

"Mejorar la calidad educativa en las unidades educativas de las redes fiscales, ofreciendo servicios modélicos y participativos en educación y en atención primaria en salud para estudiantes, docentes y sus respectivas comunidades".

## **VISION**

"Una sociedad que promueva el Desarrollo Humano con igualdad de oportunidades en educación y salud".

### **1.2.2 Antecedentes Del Proyecto.-**

Existen trabajos que están relacionados con el **“Sistema de Registro y Seguimiento de Correspondencia y Asignación de Aulas para El “Centro de Multiservicios Educativos” CEMSE-CERPI.** Por ejemplo:

- ✓ **“Sistema de Control y Seguimiento de Correspondencia SEDUCA – LA PAZ.** Utilizo la metodología estructurada ,apoyado con ASP y gestor de Base de Datos SQL Server 7.0.**[GCCP,2004]**
- ✓ **“Sistema de Registro de Correspondencia”** Compañía Boliviana de Energía Eléctrica S.A. Bolivian Power Company Limited COBEE-BPCo. **[ARJP,2004]**
- ✓ El **“Sistema de Seguimiento y Control de Correspondencia para la Prefectura del Departamento de La Paz”.** Se aplico el modelado UML, para lo cual utiliza como ayuda el lenguaje ASP, y gestor de Base de Datos SQL Server 2000.**[ACBP,2005]**

- ✓ “Sistema Automatizado de Registro y Seguimiento para la Correspondencia y/o Procesos Judiciales Departamento de Asesoría Jurídica UMSA.”, en la cual se usa como modelo de desarrollo el Proceso Racional Unificado (RUP). [TCMP,2005]
- ✓ “Sistema de Seguimiento y Control de la Documentación para la Escuela Naval Militar”, que utiliza el modelo de desarrollo el Proceso Racional Unificado (RUP). [ERJP,2007]

Para el desarrollo del presente Proyecto de Grado se usará la Metodología Orientada a Objetos (OMT), y el cual estará en entorno de Red, para lo cual se desarrollará en el lenguaje de programación PHP y gestor de Base de Datos MySql.

### **1.3 PLANTEAMIENTO DEL PROBLEMA.-**

#### **1.3.1 Análisis del Problema.-**

El desarrollo de una Institución depende del alto nivel de gestión y capacidad que presentan los departamentos que lo conforman, más aún si hablamos de una institución Educativa como el CEMSE-CERPI que debido a su crecimiento necesita optimizar los servicios que ofrece, cuya misión está orientada a mejorar la calidad de la educación impartida en los centros de educación primaria y secundaria de los estudiantes en el área de educación y salud donde juega papel importante la comunicación que se tenga entre sus dependencias y con el personal que trabaja dentro de ella, esta comunicación puede ser verbal o escrita.

Es por ello que el problema a plantear está enfocado hacia la comunicación, en este caso la escrita, que se establece entre los diferentes centros regionales, al presentar la correspondencia pérdidas y equivocaciones a la hora del envío y recepción de la misma.

Encontrando como posibles fallas: las personas que se encargan de ese proceso, bien sea por escribir mal o engorroso las direcciones de los envíos, o



por falta de control al momento de ordenar por ubicación los sobres que van a hacer entregados a los demás departamentos que trabajan en la institución.

Además no se optimiza los espacios libres de las aulas con las que cuenta el CEMSE-CERPI, teniendo problemas en la administración de los ambientes o aulas de la institución.

Después de haber realizado un análisis, utilizando la técnica de observación, cuestionarios y entrevistas podemos detectar los siguientes problemas:

- Manejo de la información manual.
- No se registra la entrada y salida de la documentación de correspondencia.
- No se tiene un seguimiento de la correspondencia así como la documentación generada en el interior de la institución.
- Se consumen innecesarios recursos humanos y de tiempo al momento de generar reportes.
- Correspondencia no entregada a tiempo.
- No se cuenta con información con respecto a las solicitudes de las aulas, con las que cuenta la institución.

### **1.3.2 Definición Del Problema.-**

El Centro de Multiservicios Educativos CEMSE y El Centro de Recursos Pedagógicos CERPI no cuenta con un sistema informático de Información y seguimiento de Correspondencia y asignación de Aulas, el manejo de la correspondencia es realizado manualmente el cual no es registrado, tampoco tiene un sistema que le brinde consultas interactivas que faciliten la generación de informes en menor tiempo.

¿Un sistema de Registro y Seguimiento de Correspondencia y Asignación de Aulas para El Centro de Multiservicios Educativos CEMSE y El Centro de Recursos Pedagógicos CERPI, generará informes confiables y oportunos, para mejorar los procesos de registro , seguimiento de correspondencia y asignación de aulas?

#### 1.4. OBJETIVOS.

##### 1.4.1 Objetivo General

Diseñar, desarrollar e implementar un sistema de registro y seguimiento de Correspondencia y asignación de aulas que facilite el proceso de registro, seguimiento y solicitudes de aula, para así solventar y mejorar los procesos manuales que se llevan a cabo en el Centro de Multiservicios Educativos CEMSE-CERPI.

##### 1.4.2 Objetivos Específicos

- Desarrollar el subsistema de Registro de Correspondencia, que mejore la gestión de correspondencia.
- Desarrollar el subsistema de seguimiento de la correspondencia, que proporcione información del estado actual de la correspondencia recibida.
- Generar informes y reportes de la correspondencia por áreas.
- Desarrollar un sistema que proporcione información al usuario mediante consultas, que tendrá una interfaz amigable.
- Desarrollar el subsistema de solicitud de aula que proporcione información sobre la disponibilidad de las aulas con las que cuenta la institución.



## **1.5. JUSTIFICACIÓN**

El presente proyecto de grado se justifica social, tecnológica y económicamente. Ya que de acuerdo al análisis realizado se puede observar problemas reales que se observaron en el Centro de Multiservicios Educativos CEMSE-CERPI, además el uso de técnicas y métodos para el desarrollo del mismo.

### **1.5.1 Justificación Social**

La implementación de este Proyecto de Grado ayudara a mejorar La información y Seguimiento de la correspondencia y las solicitudes de la red de colegios pertenecientes a la institución, recepcionadas y brindar un servicio más eficiente a la población estudiantil.

El sistema beneficiara directamente al Centro de Multiservicios Educativos CEMSE- CERPI en su totalidad.

### **1.5.2 Justificación Tecnológica.**

Este proyecto se justifica tecnológicamente porque será una herramienta de fácil uso, que ayudara a un adecuado seguimiento de la correspondencia. Además en la actualidad, con los avances tecnológicos en el área de la información; no se concibe que pequeñas y sobre todo grandes empresas e instituciones no tengan automatizado sino todos, la gran mayoría de sus procesos.

### **1.5.3 Justificación Económica**

Se justifica económicamente porque el nuevo sistema de información y Seguimiento de Correspondencia y Solicitud de Aula reduce costos, hablando en términos de tiempo y recursos humanos, así como también menos material de escritorio.

## 1.6. ALCANCES

El presente proyecto, abarcará los procesos de: Registro, Seguimiento de correspondencia y solicitudes de asignación de aulas, con el uso del análisis, diseño orientado a objetos, para el Centro de Multiservicios Educativos CEMSE-CERPI.

## 1.7. APORTES

- La implementación de los subsistemas: Registro, seguimiento y asignación de aulas para el Centro de Multiservicios Educativos CEMSE-CERPI.
- Con este proyecto se evita el uso innecesario de recursos humanos y de tiempo.
- La Aplicación de la Metodología Orientada a Objetos OMT.
- Desarrollo de un software eficiente y confiable para la institución
- Reportes oportunos.

## 1.8. METODOLOGIA Y HERRAMIENTAS.

Se usara la metodología orientada a objetos:

Análisis de Requerimientos

- Entrevistas y Cuestionarios
- Observación de las actividades desarrolladas
- Revisión de documentos (Manuales, Reglamento interno, leyes,...)

Etapas de Análisis

- Metodología Orientada a Objetos.
- Herramienta UML

Etapas de Diseño:

- Diseño Orientado a Objetos.
- Herramienta UML

Una de las técnicas para la construcción del sistema es la implementación de la arquitectura de tres niveles: interfaz de presentación, lógica de la aplicación y los datos, vía Internet, esto con el fin de no limitarnos a un control mediante de la intranet de la institución, sino tener acceso al sistema fuera de ella.

Para el desarrollo e implementación del sistema, es necesario contar con tres herramientas software orientados a Web las cuales son:

**PHP** (acrónimo de "PHP: Hypertext Preprocessor") es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor.

Apache Servidor de Internet de libre distribución de alto rendimiento, seguro y extensible creado inicialmente para funcionar bajo la plataforma Linux pero que es compatible también con la plataforma Windows.

Un sistema gestor de bases de datos (SGBD) MySQL, es una aplicación que permite manejar bases datos, son cómo archivos dónde guardamos datos ordenadamente para después recuperarlos fácilmente.

## 2. MARCO TEORICO

### 2.1 ENFOQUE ORIENTADA A OBJETOS

Vivimos en un mundo de objetos. Estos objetos existen en la naturaleza, en entidades hechas por el hombre, en los negocios y en los productos que usamos. Ellos pueden ser clasificados, descritos, organizados, combinados, manipulados y creados. Por esto no es sorprendente que se proponga una visión Orientada a Objetos para la creación de software de computadora, una abstracción que modela el mundo de forma tal que nos ayuda a entenderlo y gobernarlo mejor.

Un enfoque orientado a objetos (OO) para el desarrollo de software se propuso por primera vez a finales de los años 60. sin embargo, las tecnologías de objetos han necesitado casi veinte años para llegar a ser ampliamente usadas.

El software orientado a objetos es más fácil de mantener debido a que su estructura es inherentemente descompuesta. Esto nos lleva a menores efectos colaterales cuando se deben hacer cambios y provoca menos frustración en el ingeniero del software y el cliente. Adicionalmente, los sistemas orientados a objetos son más fáciles de adaptar y escalar (p. eje.: ejemplo, pueden crearse grandes sistemas ensamblando subsistemas reutilizables). [PRES, 1997]

#### 2.1.1 Conceptos Básicos de la Orientación a Objetos

Los conceptos de análisis y diseño orientado a objetos fueron desarrollados para dar soporte a la tecnología de programación (OO), por consiguiente es necesario citar algunas de las ideas fundamentales que subyacen en la tecnología (OO).

- a) **Clase**, descripción de un conjunto de objetos que comparten los mismos atributos operaciones, métodos, relaciones y significado.
- b) **Objeto**, instancia de una clase que encapsula el estado y el comportamiento. Más informalmente, ejemplo de una cosa.

- c) **Atributo**, es una característica o propiedad concreta de una clase.
- d) **Método** es una operación concreta de una determinada clase.
- e) **Instancia**, es una manifestación concreta de una clase(un objeto con valores concretos).
- f) **Herencia**, es una mecanismo por el cual se puede crear una nueva clase partiendo de una existente.
- g) **Abstracción**, acción de concentrar las cualidades esenciales o generales de cosas similares. También las características resultantes de una cosa.
- h) **Encapsulación**, Mecanismo con que se ocultan los datos, la estructura interna y los detalles de implementación de un objeto.

### 2.1.2 Análisis Orientado a Objetos

El objetivo del análisis orientado a objetos (AOO) es desarrollar una serie de modelos que describan el software de computadora al trabajar para satisfacer un conjunto de requisitos definidos por el cliente.

El proceso de AOO no comienza con una preocupación por los objetos. Más bien comienza con una comprensión de la manera en la que se usará el sistema: por las personas, si el sistema es de interacción con el hombre; por otras máquinas, si el sistema está envuelto en un control de procesos; o por otros programas; si el sistema coordina y controla otras aplicaciones. Una vez que se ha definido el escenario, comienza el modelado del software. [PRES, 1997]

### **2.1.3 Diseño Orientado a Objetos**

El Diseño orientado a objetos (DOO) transforma el modelo de análisis creado usando el análisis orientado a objetos, en un modelo de diseño que sirve como un anteproyecto para la construcción del software. A diferencia de los métodos convencionales de diseño del software, el DOO constituye un tipo de diseño que logra un cierto número de diferentes niveles de modularidad. Los componentes principales del sistema están organizados en “módulos” denominados subsistemas. Los datos y las operaciones que manipulan los datos están encapsulados en objetos, una forma modular que es el bloque de construcción de un sistema OO. [PRES, 1997]

## **2.2 METODOLOGIA TECNICA DE MODELADO DE OBJETOS OMT**

La metodología OMT (Object Modeling Technique) fue creada por James Rumbaugh y Michael Blaha en 1991, mientras James dirigía un equipo de investigación de los laboratorios General Electric.

OMT es una de las metodologías de análisis y diseño orientados a objetos, más maduros y eficientes que existen en la actualidad. La gran virtud que aporta esta metodología es su carácter de abierta (no propietaria), que le permite ser de dominio público y, en consecuencia, sobrevivir con enorme vitalidad. Esto facilita su evolución para acoplarse a todas las necesidades actuales y futuras de la ingeniería de software.

Es una metodología orientada a objeto muy difundida que se hace cargo de todo el ciclo de vida del software. Parte de la idea de utilizar los mismos conceptos y la misma notación a lo largo de todo el ciclo de vida.

Tiene una fase de diseño no muy compleja y se centra mucho en un análisis. Es de las denominadas “dirigidas por los datos”.

Divide el ciclo de vida del software en cuatro fases consecutivas: análisis de objetos diseño del sistema, diseño de objetos e implementación.



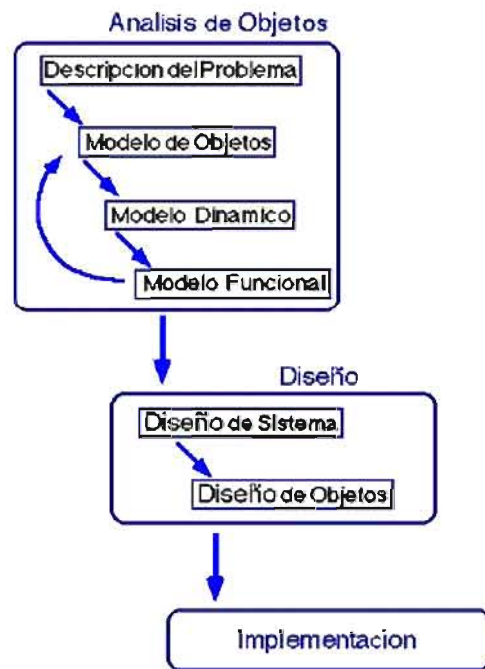


Figura: 2.1 Ciclo de vida OMT [OMTR,ICR]<sup>4</sup>

- ✓ **Análisis de Objetos**, el modelo de análisis es una abstracción resumida y precisa de lo que debe hacer el sistema deseado y no de la forma en que se hará. Los objetos del modelo deberán ser conceptos del dominio de la aplicación, y no conceptos de implementación de la computadora tales como estructura de datos. Un buen modelo podrá ser comprendido y criticado por expertos de la aplicación que no sean programadores. El modelo de Análisis no deberá contener ninguna decisión de implementación.
  
- ✓ **Diseño del Sistema**, el diseñador de sistemas toma decisiones de alto nivel acerca de la arquitectura global durante el diseño, el sistema de destino se organiza en subsistemas basados tanto en la estructura de análisis como en la arquitectura propuesta. El diseñador de sistemas deberá decidir que características de rendimiento hay que optimizar. Seleccionando una estrategia para atacar el problema y efectuando una reserva de recursos tentativas.
  
- ✓ **Diseño de Objetos**, el diseñador de objetos construye un modulo de diseño basándose en el modelo de análisis que lleven incorporados detalles de implementación. El diseñador añade detalles al modelo de acuerdo con la

<sup>4</sup> [http://pisuerga.inf.ubu.es/icruzado/ffc/OMT\\_res.pdf](http://pisuerga.inf.ubu.es/icruzado/ffc/OMT_res.pdf)

estrategia establecida durante el diseño del sistema. El foco de atención del diseño de objetos son las estructuras de datos y los algoritmos necesarios para implementar cada una de las clases. Las clases de objetos procedentes del análisis siguen siendo significativas pero se aumentan con estructuras de datos y algoritmos del dominio de la computadora seleccionados para optimizar medidas importantes de rendimiento. Tanto los objetos del dominio de aplicación como de los objetos del dominio de la computadora se describen utilizando los mismos y una misma notación orientada a objetos aun cuando existan en planos conceptuales diferentes.

- ✓ **Implementación**, Las clases de objetos y las relaciones desarrolladas durante su diseño se traducen finalmente a un lenguaje de programación concreto, a una base de datos o a una implementación en hardware. La programación debería ser una parte relativamente pequeña del ciclo de desarrollo y fundamentalmente mecánica porque todas las decisiones importantes deberán hacerse durante el diseño. El lenguaje de diseño influye en cierta medida sobre las decisiones de diseño pero este no debería depender de la estructura final de un lenguaje de programación. Durante la implementación es importante respetar las buenas ideas de la ingeniería del software, de tal manera que el seguimiento hasta el diseño sea sencillo y de tal forma que el sistema implementado siga siendo flexible y extensible.

La metodología OMT emplea tres clases de modelos para describir el sistema.

**El Modelo de Objetos**, que describe los objetos del sistema y sus relaciones. El modelo de objetos contiene diagramas de objetos.

**El modelo dinámico**, que describe las interacciones existentes entre objetos del sistema y los aspectos de un sistema que cambian con el tiempo. El modelo dinámico se utiliza para especificar e implementar los aspectos de control del sistema. Los modelos dinámicos contienen diagramas de estado.

**El modelo funcional**, que describe las transformaciones de datos del sistema. El modelo funcional contiene diagramas de flujo de datos.



Todos los modelos son aplicables en la totalidad de las fases del desarrollo y van adquiriendo detalles de implementación a medida que progresa el desarrollo. Los tres modelos son partes ortogonales de la descripción del sistema y están enlazados entre si. [RUMB, 96].

### **2.3 LENGUAJE UNIFICADO PARA LA CONSTRUCCIÓN DE MODELOS UML**

Es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Se ha convertido en el estándar de facto de la industria, debido a que ha sido concebido por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh. Estos autores fueron contratados por la empresa Rational Software Co. para crear una notación unificada en la que basar la construcción de sus herramientas CASE. En el proceso de creación de UML han participado, no obstante, otras empresas de gran peso en la industria como Microsoft, Hewlett-Packard, Oracle o IBM, así como grupos de analistas y desarrolladores.

Esta notación ha sido ampliamente aceptada debido al prestigio de sus creadores y debido a que incorpora las principales ventajas de cada uno de los métodos particulares en los que se basa: Booch, OMT y OOSE. UML ha puesto fin a las llamadas “guerras de métodos” que se han mantenido a lo largo de los 90, en las que los principales métodos sacaban nuevas versiones que incorporaban las técnicas de los demás. Con UML se fusiona la notación de estas técnicas para formar una herramienta compartida entre todos los ingenieros software que trabajan en el desarrollo orientado a objetos.

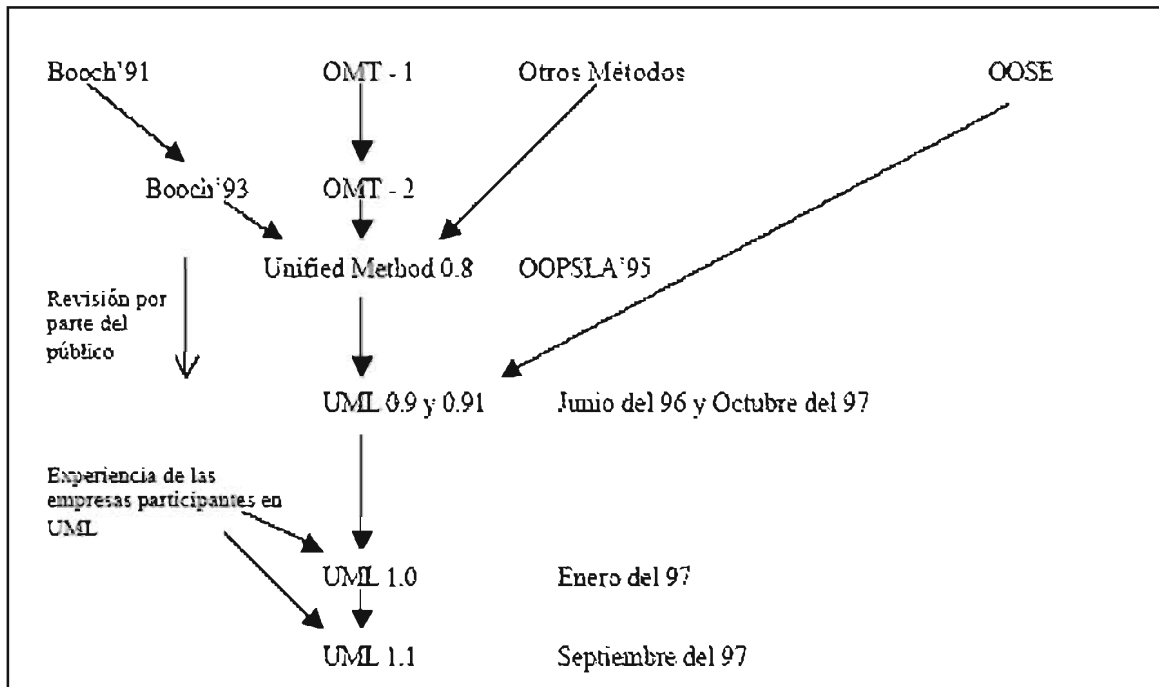


Figura: 2.2 Historia de UML [XFG,UML]<sup>5</sup>

Hay que tener en cuenta que el estándar UML no define un proceso de desarrollo específico, tan solo se trata de una notación. En el presente Proyecto de Grado se sigue el método propuesto por Craig Larman [LARM,99]<sup>6</sup> que se ajusta a un ciclo de vida iterativo e incremental dirigido por casos de uso.

### 2.3.1 Modelos

Un modelo representa a un sistema software desde una perspectiva específica. Al igual que la planta y el alzado de una figura en dibujo técnico nos muestran la misma figura vista desde distintos ángulos, cada modelo nos permite fijarnos en un aspecto distinto del sistema.

Los modelos de UML utilizados son los siguientes:

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

<sup>5</sup> Xavier Ferré Grau, Maria Isabel Sanchez Segura "Desarrollo Orientado a Objetos con UML".

<sup>6</sup> Larman C.: *UML y Patrones*. Prentice Hall, 1999.

## Elementos Comunes a Todos los Diagramas

### a) Notas

Una nota sirve para añadir cualquier tipo de comentario a un diagrama o a un elemento de un diagrama. Es un modo de indicar información en un formato libre, cuando la notación del diagrama en cuestión no nos permite expresar dicha información de manera adecuada. Una nota se representa como un rectángulo con una esquina doblada con texto en su interior.

Puede aparecer en un diagrama tanto sola, como unida a un elemento por medio de una línea discontinua. Puede contener restricciones, comentarios, el cuerpo de un procedimiento, etc.

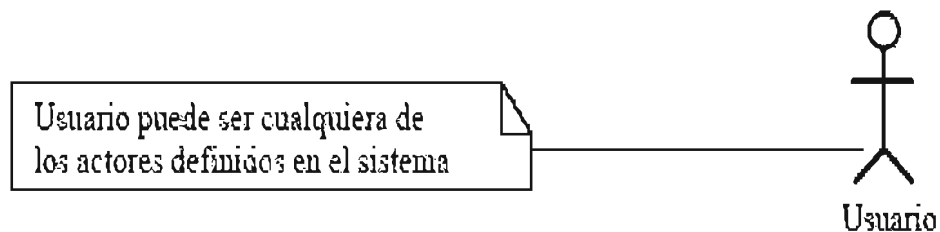


Figura 2.2 Ejemplo de nota [XFG,UML]<sup>7</sup>

### b) Dependencias

La relación de dependencia entre dos elementos de un diagrama significa que un cambio en el elemento destino puede implicar un cambio en el elemento origen (por tanto, si cambia el elemento destino habría que revisar el elemento origen).

Una dependencia se representa por medio de una línea de trazo discontinuo entre los dos elementos con una flecha en su extremo. El elemento dependiente es el origen de la flecha y el elemento del que depende es el destino (junto a él está la flecha).

<sup>7</sup> Xavier Ferré Grau, Maria Isabel Sanchez Segura "Desarrollo Orientado a Objetos con UML".

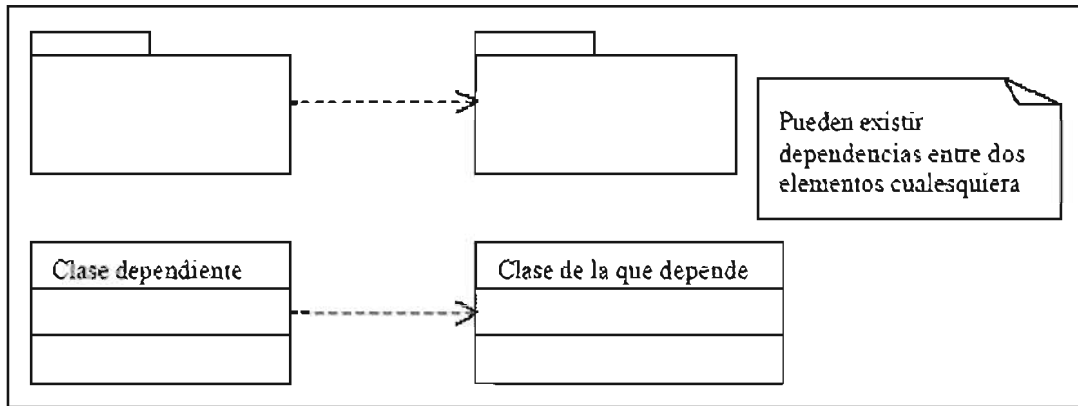


Figura 2.3 Dependencias [XFG,UML]<sup>8</sup>

### 2.3.1.1 Diagramas De Estructura Estática

Con el nombre de Diagramas de Estructura Estática se engloba tanto al Modelo Conceptual de la fase de Análisis como al Diagrama de Clases de la fase de Diseño. Ambos son distintos conceptualmente, mientras el primero modela elementos del dominio el segundo presenta los elementos de la solución software. Sin embargo, ambos comparten la misma notación para los elementos que los forman (clases y objetos) y las relaciones que existen entre los mismos (asociaciones).

#### a) Clases

Una clase se representa mediante una caja subdividida en tres partes: En la superior se muestra el nombre de la clase, en la media los atributos y en la inferior las operaciones. Una clase puede representarse de forma esquemática (plegada), con los detalles como atributos y operaciones suprimidos, siendo entonces tan solo un rectángulo con el nombre de la clase. En la Figura 5 se ve cómo una misma clase puede representarse a distinto nivel de detalle según interese, y según la fase en la que se esté.

<sup>8</sup> Xavier Ferré Grau, Maria Isabel Sanchez Segura “Desarrollo Orientado a Objetos con UML”.

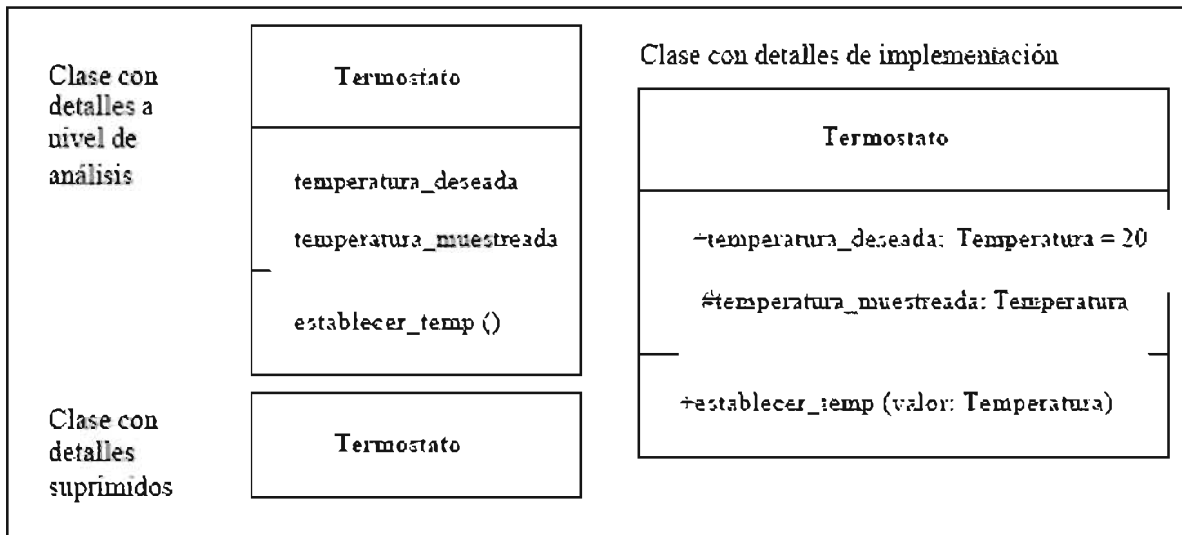


Figura 2.4 Notación para clases a distintos niveles de detalle [XFG,UML]<sup>9</sup>

### b) Objetos

Un objeto se representa de la misma forma que una clase. En el compartimento superior aparecen el nombre del objeto junto con el nombre de la clase subrayados, según la siguiente sintaxis: *nombre\_del\_objeto: nombre\_de\_la\_clase*. Puede representarse un objeto sin un nombre específico, entonces sólo aparece el nombre de la clase.

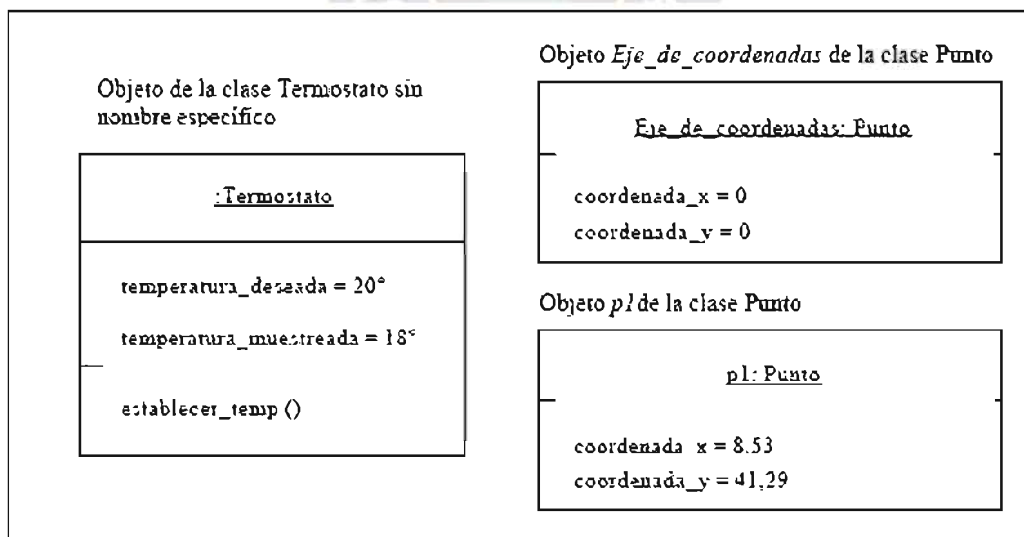


Figura 2.5 Ejemplos de objetos [XFG,UML]

<sup>9</sup> Xavier Ferré Grau, Maria Isabel Sanchez Segura “Desarrollo Orientado a Objetos con UML”.

### c) Asociaciones

Las asociaciones entre dos clases se representan mediante una línea que las une. La línea puede tener una serie de elementos gráficos que expresan características particulares de la asociación. A continuación se verán los más importantes de entre dichos elementos gráficos.

- **Nombre de la Asociación y Dirección**

El nombre de la asociación es opcional y se muestra como un texto que está próximo a la línea. Se puede añadir un pequeño triángulo negro sólido que indique la dirección en la cual leer el nombre de la asociación. En el ejemplo de la Figura 7 se puede leer la asociación como “Director manda sobre Empleado”.



Figura 2.6 Ejemplo de asociación con nombre y dirección [XFG,UML]<sup>10</sup>

Los nombres de las asociaciones normalmente se incluyen en los modelos para aumentar la legibilidad. Sin embargo, en ocasiones pueden hacer demasiado abundante la información que se presenta, con el consiguiente riesgo de saturación. En ese caso se puede suprimir el nombre de las asociaciones consideradas como suficientemente conocidas. En las asociaciones de tipo agregación y de herencia no se suele poner el nombre.

<sup>10</sup> Xavier Ferré Grau, Maria Isabel Sanchez Segura “Desarrollo Orientado a Objetos con UML”.

- **Multiplicidad**

La multiplicidad es una restricción que se pone a una asociación, que limita el número de instancias de una clase que pueden tener esa asociación con una instancia de la otra clase. Puede expresarse de las siguientes formas:

- Con un número fijo: 1.
- Con un intervalo de valores: 2..5.
- Con un rango en el cual uno de los extremos es un asterisco. Significa que es un intervalo abierto. Por ejemplo, 2..\* significa 2 o más.
- Con una combinación de elementos como los anteriores separados por comas: 1, 3..5, 7, 15..\*.
- Con un asterisco: \*. En este caso indica que puede tomar cualquier valor (cero o más).

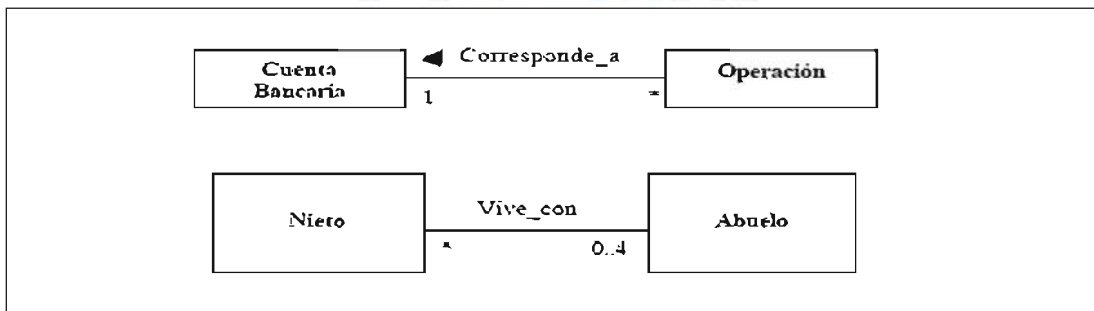


Figura 2.7 Ejemplos de multiplicidad en asociaciones [XFG,UML]<sup>11</sup>

### Roles

Para indicar el papel que juega una clase en una asociación se puede especificar un nombre de rol. Se representa en el extremo de la asociación junto a la clase que desempeña dicho rol.

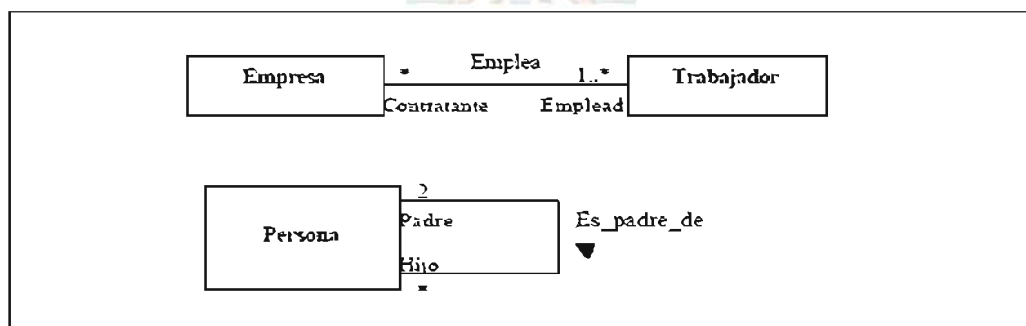


Figura 2.8 Ejemplo de roles en una asociación [XFG,UML]

<sup>11</sup> Xavier Ferré Grau, Maria Isabel Sanchez Segura “Desarrollo Orientado a Objetos con UML”.



## Agregación

El símbolo de agregación es un diamante colocado en el extremo en el que está la clase que representa el “todo”.

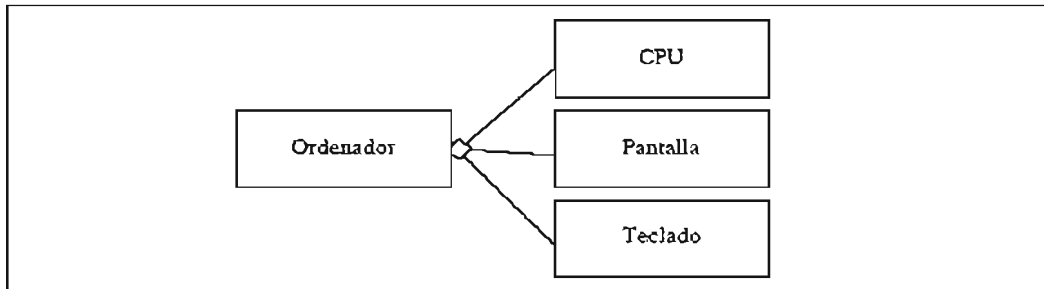


Figura 2.9 Ejemplo de agregación [XFG,UML]<sup>12</sup>

## Clases Asociación:

Cuando una asociación tiene propiedades propias se representa como una clase unida a la línea de la asociación por medio de una línea a trazos. Tanto la línea como el rectángulo de clase representan el mismo elemento conceptual: la asociación. Por tanto ambos tienen el mismo nombre, el de la asociación. Cuando la clase asociación sólo tiene atributos el nombre suele ponerse sobre la línea (como ocurre en el ejemplo de la Figura 11). Por el contrario, cuando la clase asociación tiene alguna operación o asociación propia, entonces se pone el nombre en la clase asociación y se puede quitar de la línea.

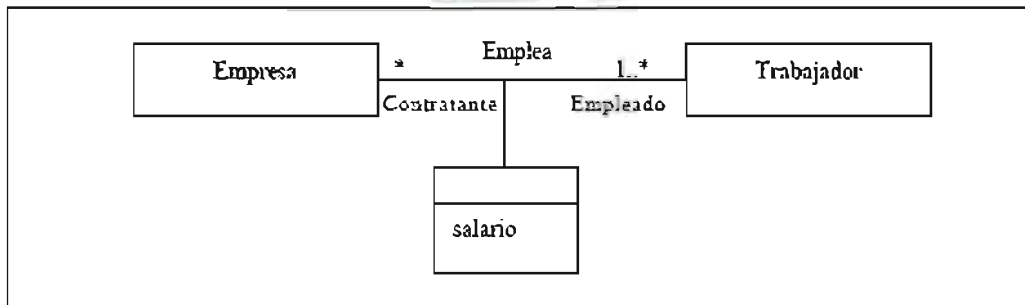


Figura 2.10 Ejemplo de clase asociación [XFG,UML]

<sup>12</sup> Xavier Ferré Grau, Maria Isabel Sanchez Segura “Desarrollo Orientado a Objetos con UML”.



## Asociaciones N-Arias

En el caso de una asociación en la que participan más de dos clases, las clases se unen con una línea a un diamante central. Si se muestra multiplicidad en un rol, representa el número potencial de tuplas de instancias en la asociación cuando el resto de los N-1 valores están fijados. En la Figura 12 se ha impuesto la restricción de que un jugador no puede jugar en dos equipos distintos a lo largo de una temporada, porque la multiplicidad de "Equipo" es 1 en la asociación ternaria.

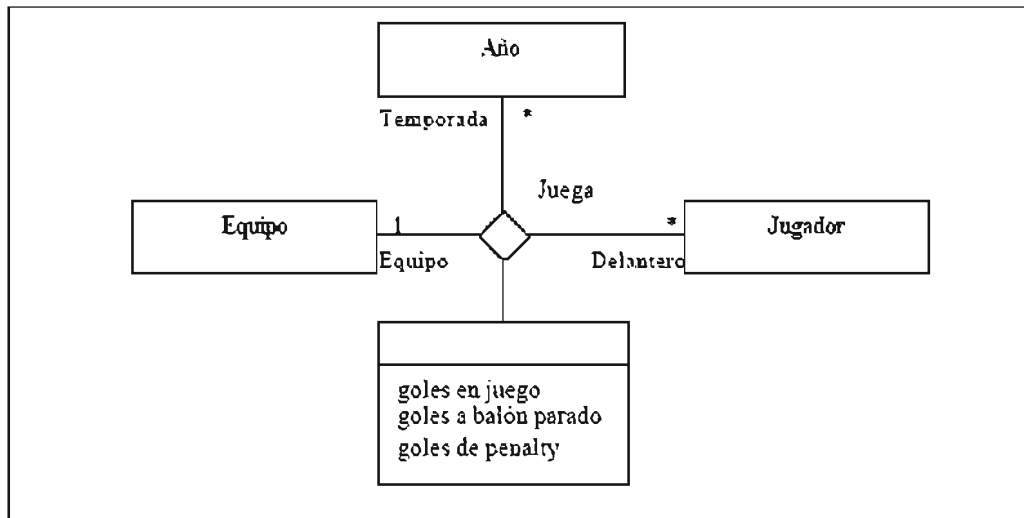


Figura 2.11 Ejemplo de asociación ternaria [XFG,UML]

## Navegabilidad

En un extremo de una asociación se puede indicar la navegabilidad mediante una flecha.

Significa que es posible "navegar" desde el objeto de la clase origen hasta el objeto de la clase destino. Se trata de un concepto de diseño, que indica que un objeto de la clase origen conoce al (los) objeto(s) de la clase destino, y por tanto puede llamar a alguna de sus operaciones.

#### d) Herencia

La relación de herencia se representa mediante un triángulo en el extremo de la relación que corresponde a la clase más general o clase “padre”. Si se tiene una relación de herencia con varias clases subordinadas, pero en un diagrama concreto no se quieren poner todas, esto se representa mediante puntos suspensivos. En el ejemplo de la Figura 13, sólo aparecen en el diagrama 3 tipos de departamentos, pero con los puntos suspensivos se indica que en el modelo completo (el formado por todos los diagramas) la clase “Departamento” tiene subclases adicionales, como podrían ser “Recursos Humanos” y “Producción”.

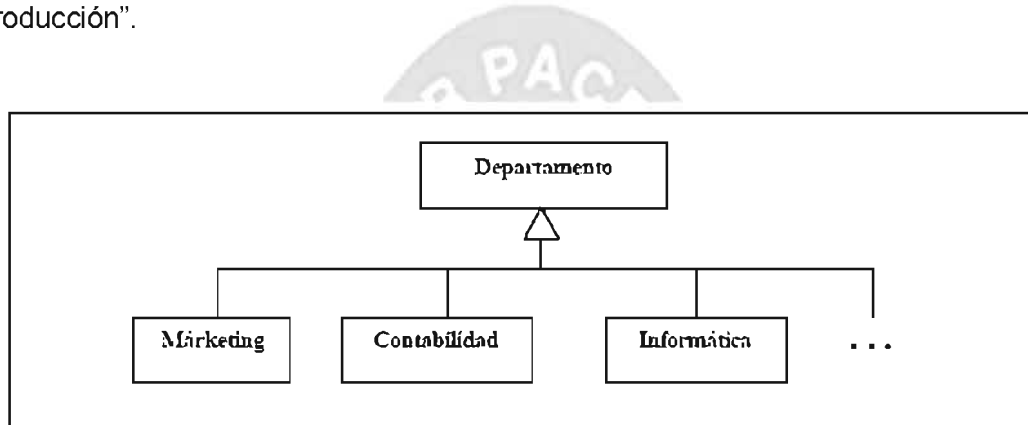


Figura 2.12 Ejemplo de herencia [XFG,UML]

#### e) Elementos Derivados

Un elemento derivado es aquel cuyo valor se puede calcular a partir de otros elementos Presentes en el modelo, pero que se incluye en el modelo por motivos de claridad o como decisión de diseño. Se representa con una barra “/” precediendo al nombre del elemento Derivado.

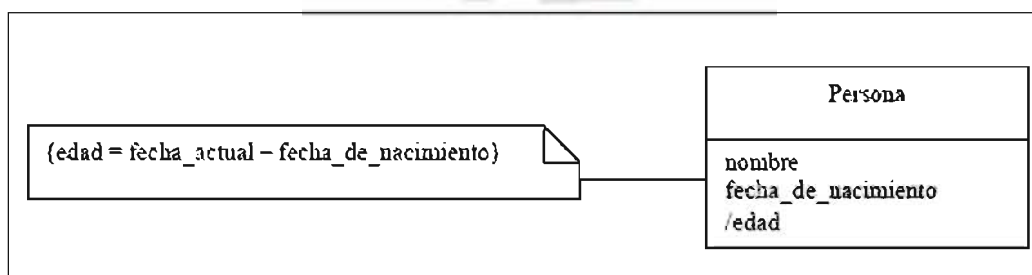


Figura 2.13 Ejemplo de Atributo derivado [XFG,UML]

### 2.3.1.2 Diagrama de Casos de Uso

Un Diagrama de Casos de Uso muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa.

#### a) Elementos

Los elementos que pueden aparecer en un Diagrama de Casos de Uso son: actores, casos de uso y relaciones entre casos de uso.

##### **Actores**

Un actor es una entidad externa al sistema que realiza algún tipo de interacción con el mismo. Se representa mediante una figura humana dibujada con palotes. Esta representación sirve tanto para actores que son personas como para otro tipo de actores (otros sistemas, sensores, etc.).

##### **Casos de Uso**

Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica. Expresa una unidad coherente de funcionalidad, y se representa en el Diagrama de Casos de Uso mediante una elipse con el nombre del caso de uso en su interior. El nombre del caso de uso debe reflejar la tarea específica que el actor desea llevar a cabo usando el sistema.

##### **Relaciones entre Casos de Uso**

Entre dos casos de uso puede haber las siguientes relaciones:

- Extiende: Cuando un caso de uso especializa a otro extendiendo su funcionalidad.
- Usa: Cuando un caso de uso utiliza a otro.

Se representan como una línea que une a los dos casos de uso relacionados, con una flecha en forma de triángulo y con una etiqueta <<extiende>> o <<usa>> según sea el tipo de relación.

En el diagrama de casos de uso se representa también el sistema como una caja rectangular con el nombre en su interior. Los casos de uso están en el interior de la caja del sistema, y los actores fuera, y cada actor está

unido a los casos de uso en los que participa mediante una línea. En la Figura 15 se muestra un ejemplo de Diagrama de Casos de Uso para un cajero automático.

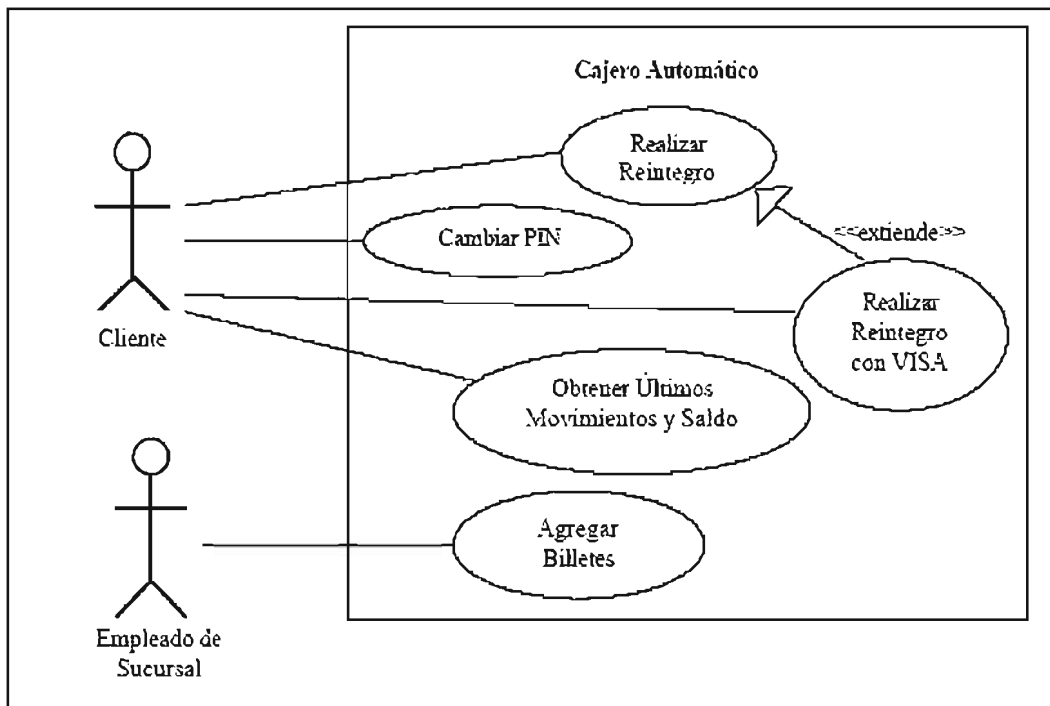


Figura 2.14 Diagrama de Casos de Uso [XFG,UML]

### 2.3.1.3 Diagramas de Interacción

En los diagramas de interacción se muestra un patrón de interacción entre objetos. Hay dos tipos de diagrama de interacción, ambos basados en la misma información, pero cada uno enfatizando un aspecto particular: Diagramas de Secuencia y Diagramas de Colaboración.

#### a) Diagrama de Secuencia

Un diagrama de Secuencia muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo.

El eje vertical representa el tiempo, y en el eje horizontal se colocan los objetos y actores participantes en la interacción, sin un orden prefijado. Cada objeto o actor tiene una línea vertical, y los mensajes se representan mediante flechas entre los distintos objetos. El tiempo fluye de arriba abajo.

Se pueden colocar etiquetas (como restricciones de tiempo, descripciones de acciones, etc.) bien en el margen izquierdo o bien junto a las transiciones o activaciones a las que se refieren.



Figura  
2.15  
Diagram

Diagrama de Secuencia [XFG,UML]<sup>13</sup>

### b) Diagrama de Colaboración

Un Diagrama de Colaboración muestra una interacción organizada basándose en los objetos que toman parte en la interacción y los enlaces entre los mismos (en cuanto a la interacción se refiere). A diferencia de los Diagramas de Secuencia, los Diagramas de Colaboración muestran las relaciones entre los roles de los objetos. La secuencia de los mensajes y los flujos de ejecución concurrentes deben determinarse explícitamente mediante números de secuencia. En cuanto a la representación, un Diagrama de Colaboración muestra a una serie de objetos con los enlaces entre los mismos, y con los mensajes que se intercambian dichos objetos. Los mensajes son flechas que van junto al enlace por el que “circulan”, y con el nombre del mensaje y los parámetros (si los tiene) entre paréntesis.

Cada mensaje lleva un número de secuencia que denota cuál es el mensaje que le precede, excepto el mensaje que inicia el diagrama, que no lleva número de secuencia. Se pueden indicar alternativas con condiciones entre corchetes (por ejemplo 3 [condición\_de\_test]: nombre\_de\_método() ), tal y como aparece en el ejemplo de la Figura 17. También se puede mostrar el anidamiento de mensajes con números de secuencia como 2.1, que

<sup>13</sup> Xavier Ferré Grau, Maria Isabel Sanchez Segura “Desarrollo Orientado a Objetos con UML”.

significa que el mensaje con número de secuencia 2 no acaba de ejecutarse hasta que no se han ejecutado todos los 2. x .

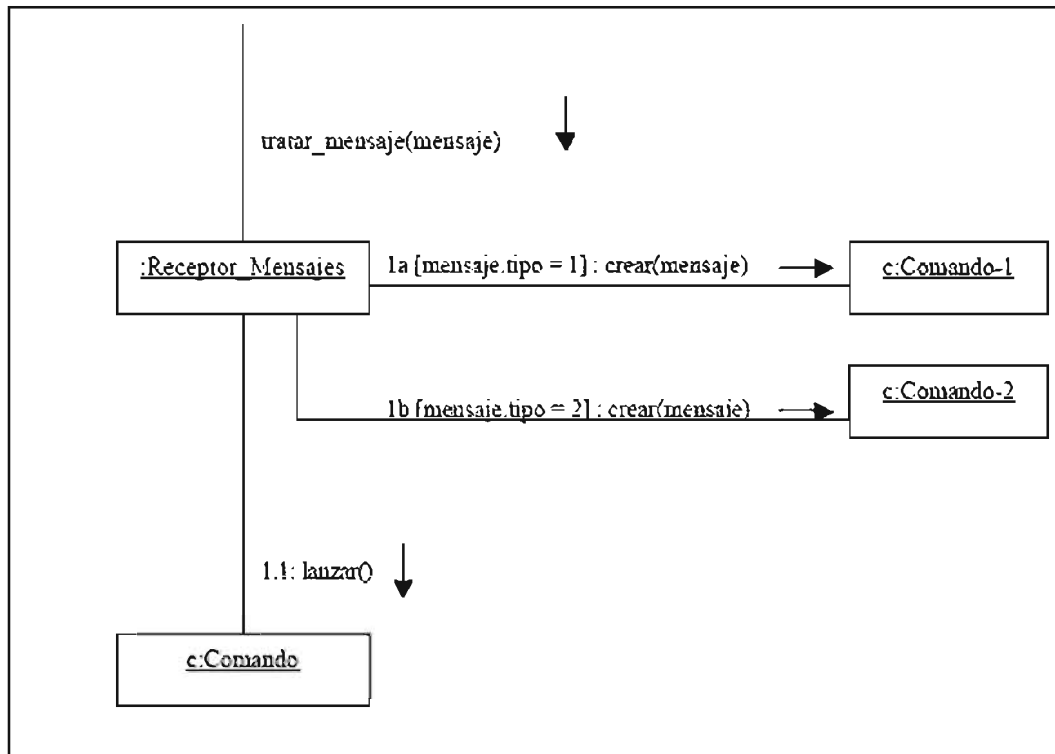


Figura 2.16 Diagrama de Colaboración [XFG,UML]<sup>14</sup>

#### 2.3.1.4 Diagrama de Estados

Un Diagrama de Estados muestra la secuencia de estados por los que pasa un caso de uso o un objeto a lo largo de su vida, indicando qué eventos hacen que se pase de un estado a otro y cuáles son las respuestas y acciones que genera.

<sup>14</sup> Xavier Ferré Grau, Maria Isabel Sanchez Segura “Desarrollo Orientado a Objetos con UML”.

En cuanto a la representación, un diagrama de estados es un grafo cuyos nodos son estados y cuyos arcos dirigidos son transiciones etiquetadas con los nombres de los eventos.

Un estado se representa como una caja redondeada con el nombre del estado en su interior. Una transición se representa como una flecha desde el estado origen al estado destino. La caja de un estado puede tener 1 o 2 compartimentos. En el primer compartimento aparece el nombre del estado. El segundo compartimento es opcional, y en él pueden aparecer acciones de entrada, de salida y acciones internas.

Una acción de entrada aparece en la forma *entrada/acción\_asociada* donde *acción\_asociada* es el nombre de la acción que se realiza al entrar en ese estado. Cada vez que se entra al estado por medio de una transición la acción de entrada se ejecuta.

Una acción de salida aparece en la forma *salida/acción\_asociada*. Cada vez que se sale del estado por una transición de salida la acción de salida se ejecuta.

Una acción interna es una acción que se ejecuta cuando se recibe un determinado evento en ese estado, pero que no causa una transición a otro estado. Se indica en la forma *nombre\_de\_evento/acción\_asociada*

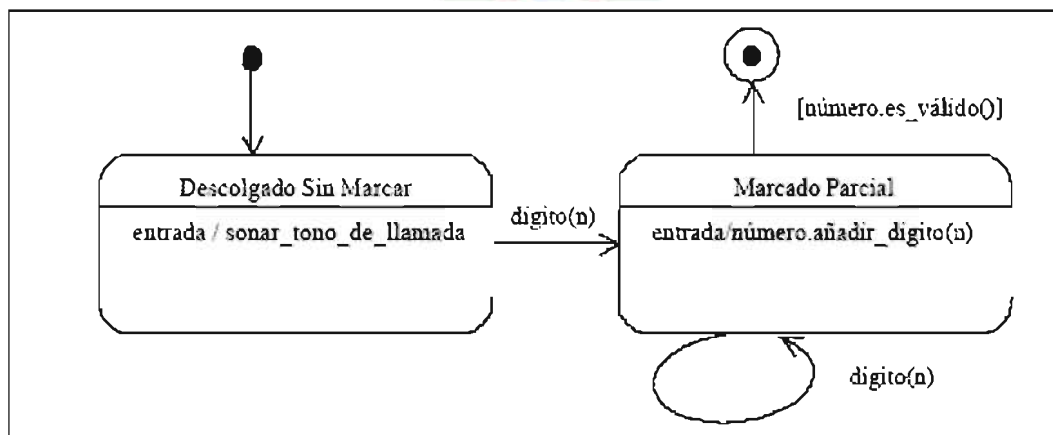


Figura 2.17 Diagrama de Estados [XFG,UML]

Un diagrama de estados puede representar ciclos continuos o bien una vida finita, en la que hay un estado inicial de creación y un estado final de destrucción (del caso de uso o del objeto). El estado inicial se muestra como un círculo sólido y el estado final como un círculo sólido rodeado de otro círculo. En realidad, los estados inicial y final son pseudoestados, pues un objeto no puede “estar” en esos estados, pero nos sirven para saber cuáles son la transición inicial y final(es).





### **3. GESTION DOCUMENTAL**

Antes de abordar las características de la administración documentaria, es preciso analizar el concepto y alcances de la base esencial de ese proceso: el documento es la célula inicial que da lugar a la gestión, embrión del archivo, y por ende a su desarrollo último que se verifica ya en el archivo de gestión u oficina.

Los documentos de archivo contienen información, pero es preciso antes dos palabras sobre el origen del archivo, analizando el dato y la información, insumos básicos de los documentos.

#### **3.1 DATO**

El dato es un hecho simple, un “data-ítem” o elemento de dato, que se encarga de describir sucesos y entidades. Su caracterización es que como tales no tienen la capacidad de comunicar un significado. Por ende no afectan el comportamiento de quien los recibe. En otras palabras, no contienen información.

Sin embargo, cuando se ordenan y reordenan de forma utilizable, adquieren el atributo de información. Se convierten en dato significativo gracias a una serie de símbolos reconocibles, completos que expresan una idea, condiciones, hechos, situaciones o valores. Esa capacidad de asociación dentro de un contexto para ofrecer un significado, conocimiento, ideas o conclusiones, los convierte en información.

#### **3.2 INFORMACIÓN**

Es el conocimiento basado en los datos, a los cuales se les ha dado significado, propósito y utilidad. Es un mensaje con contenido determinado emitido por una persona o entidad hacia otra; en ese proceso se transforma en acción y por ello representa un papel primordial en el proceso de la comunicación y posee una evidente función social, pues se torna en el insumo

básico de las ciencias de la información, y por ende del documento. Para su transmisión requiere, necesariamente, de un soporte que contiene los datos significativos, es decir, la información.

### 3.3 DOCUMENTO

Deriva de *documentum*, que a su vez procede del verbo *docere*, que quiere decir “enseñar”. El sufijo *mentum* viene de *mento* que quiere decir “elemento material en el que se ha hecho visible o palpable una idea o un hecho”.

Ese concepto es amplio, porque abarca a un conjunto de datos e impulsos significativos, en lenguaje natural o convencional, gráfico, en imagen o sonoro, registrados en cualquier tipo de soporte, incluso de tipo informático, es decir:

*“Documento es todo registro de información independiente de su soporte físico. Abarca todo lo que puede transmitir el conocimiento humano”.*

#### 3.3.1 Documento Público

Este concepto fue acuñado por el Programa de Desarrollo de Servicio de los Archivos de Bolivia, y se refiere a la base esencial que sustenta toda teoría archivística, pues *sin documentos no hay Servicio*; y menos aun puede pensarse siquiera en comprender los problemas del desarrollo del Servicio. El concepto señala que:

*“Documento público es cualquier documento, formulado por cualquier medio expresivo, como medio específico para probar, describir, comunicar y/o informar algo dentro de la estructura, la función, la actividad, y los trámites de cualquier entidad pública en cualquier parte y en cualquier época”.*

### 3.3.2 Documento de Archivo

En función de lo anterior, se puede definir al documento de archivo como el soporte original, recibido o producido por una persona o entidad, en el ejercicio de sus funciones o actividades, que se caracteriza por su legitimidad, y está dotado de valores primarios (administrativo, fiscal/contable, y legal) desde el momento de su generación. Se lo conoce también como “pieza documental”, documento individual que generalmente viene en una o mas hojas de papel. Puede ser una pieza simple (puede constar de una o mas hojas), o compleja (puede estar seguida de anexos, adjuntos e impresos).

### 3.3.3 Documentaciones Públicas.

Son los documentos resultantes de la función, actividades y tramites de cualquier estructura administrativa central, descentralizada, desconcentrada, local, municipal, judicial y/o universitaria. Constituyen bienes y recursos indispensables para la administración pública. La información e investigación científica, promoción de la conciencia cívica y el desarrollo nacional.

Estas documentaciones forman la base del patrimonio documental de la nación, una vez que pierden sus valores primarios y pasan la prueba de valoración

## 3.4 TEORIA DEL VALOR DOCUMENTAL

Una de las características peculiares de la documentación de archivo reside en sus valores. La diplomática ha establecido con propiedad que la documentación de archivo nace con valores intrínsecos, que le otorgan en primera instancia su carácter de originalidad o único en su genero (*no existen dos documentos originales y/o auténticos iguales*) y legítimo (*por su autenticidad*).

A partir de estas características surgen un conjunto de valores que le asignan a la documentación el carácter de información estratégica para el desarrollo socioeconómico de una comunidad o una Nación. Metodológicamente parece mas apropiada una primera clasificación de los valores en un nivel genérico, temporalmente aplicables en primera instancia (*valor primario y secundario*), a partir del cual tendremos un conjunto de valores específicos, especialmente aplicables (en los archivos, de acuerdo al ciclo vital de los documentos).

#### **3.4.1 Valores Primarios**

Conjunto de valores específicos que son de utilidad exclusiva para los productores de la documentación, y por ello son mediatizados en su acceso para el conjunto de la sociedad, por el imperio del Corpus jurídico nacional, en base a la calidad de prueba de los asuntos tramitados por las partes interesadas, que se conoce como el interés legitimo de los administrados. Estos valores tienen vigencia tanto en cuanto prescriban los valores específicos de la etapa activa, comprendidos bajo la abstracción del *Valor Probatorio*.

#### **3.4.2 Valores Secundarios**

Conjunto de valores específicos informativos e históricos que adquieren los documentos al perder sus valores primarios. Es decir, cuando se establece la perdida de los valores primarios, desaparece el carácter mediatizador de la accesibilidad plena; por ende, son liberados al uso publico por efectos del plazo de conservación, que son determinados, siempre, por normas legales. Se dice entonces que al perder sus valores primarios, las documentaciones son de acceso inmediato para el conjunto de la sociedad.

### **3.4.3 Valor Informativo**

El valor informativo expresa el carácter y potencial informativo intrínseco de cualquier documento, otorgándole gran utilidad para la investigación, pues los documentos a menudo contienen información con alto valor para la investigación, sin que hubiera sido considerado como tal por los creadores de la documentación. En este caso, a pesar de ser un valor intrínseco o natural, el rigor del carácter mediato de la documentación activa, impide su uso hasta que es liberado de ese óbice por efectos de los plazos de conservación. En términos estrictamente archivísticos, solo en tanto y en cuanto alcancen el carácter inmediato podrá explotarse el valor potencial informativo por el conjunto de la sociedad de manera irrestricta.

Sin embargo de lo anterior, no cabe duda que se trata de un valor permanente que esta dado por las características intrínsecas y extrínsecas del documento. Los documentos de archivo adquieren el carácter de fuente oficial. Las copias simples de aquellas legales, requieren de la compulsión documental previa y la certificación de copia fiel del original. La siguiente máxima se aplica en este caso: todos los documentos de archivo, de cualquier época y tipo tienen valor informativo.

### **3.4.4 Valor Histórico**

En primera instancia, es aplicable a las documentaciones que han pasado por la fase de la valoración y son preservadas para su conservación permanente, por su importancia para los intereses de la sociedad civil, el gobierno y el país.

### **3.4.5 Valores Abstractos.**

El conjunto de valores aplicables en la etapa activa de la documentación y en tanto esté vigente la institución y/o individuo creador, le dotan al documento del Valor Probatorio, que viene a ser una abstracción de uno o mas valores primarios, pues expresa de manera fehaciente la actuación administrativa. Este valor probatorio convierte al documento de valores es aplicado en la etapa inactiva y se conceptúa bajo la abstracción de Valor Archivístico.

### **3.4.6 Valor Probatorio.**

Abstracción conceptual que deviene del conjunto valores propios de la etapa administrativa (valor administrativo, judicial, legal, o contable). El plazo de la prescripción del valor legal, es de diez años en Bolivia (15 años para fines archivísticos). Sus alcances son sorprendentes, pues sirve tanto para fines administrativos institucionales, como los propios del control fiscal, y en los estrados judiciales.

El valor probatorio provee evidencia del origen, estructura, funciones y actividades de la Administración o la persona por las que se ha creado e documento.

### **3.4.7 Valor Archivístico.**

Este concepto resulta de la abstracción del conjunto de valores del documento. Dicho de otra manera: el valor archivístico como tal no existe, sino como un valor abstracto, en el que se basa la justificación de última instancia sobre la pertinencia de la conservación permanente de los documentos.



VALORES GENERICOS	VALORES ESPECIFICOS	VALORES ABSTRACTOS
Valores primarios  Valores Secundarios	Administrativo  Fiscal(Contable)  Legal(Jurídico)  Valor Informativo  Valor Histórico	Valor Probatorio  Valor Archivístico

**Tabla 3.1:** Síntesis de la Teoría del Valor Documental. [GEST,05]

### 3.5 CLASIFICACION DE LA CORRESPONDENCIA

La clasificación de correspondencia corriente tiene los siguientes aspectos:

- ❖ Las carpetas o archivadores de palanca organizan su documentación, considerando tanto la que se reciba como la respuesta que diera a lugar.
- ❖ Se prohíbe clasificar el archivo de correspondencia corriente por “Recibida” y “Despachada”, por que rompe la unidad de la comunicación e impide ubicar oportunamente las respuestas.
- ❖ El archivo de correspondencia corriente se subdivide en tres subseries y se ordena como sigue:



- **Correspondencia externa.** Agrupa la correspondencia con instituciones externas en general, sean de carácter nacional o internacional.
- **Correspondencia interna.** Agrupa la correspondencia institucional entre las oficinas de la institución.
- **Correspondencia de personas.** Agrupa la correspondencia mantenida con personas naturales.



## 4. MARCO APLICATIVO

Para el desarrollo de este capítulo se hace la conjunción del análisis y el diseño del Sistema de Registro y Seguimiento de Correspondencia y asignación de aulas, empleando la metodología OMT de Rumbaugh como base para el desarrollo del proyecto y la herramienta de apoyo UML, como un lenguaje de modelado en sus distintas fases de OMT para una mejor visualización y construcción de los artefactos del sistema.

### 4.1 ANALISIS DE OBJETOS

En el Análisis de Objetos descritos en la Técnica de Modelado de Objetos se tiene lo siguiente:

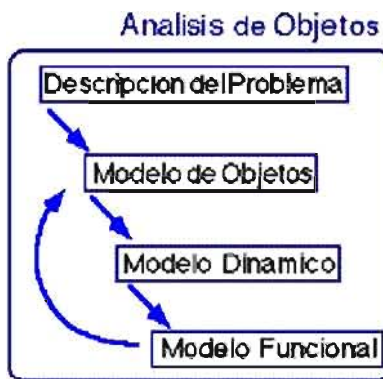


Figura. 4.1 Análisis de Objetos OMT. [OMTR,ICR]<sup>15</sup>

<sup>15</sup> [http://pisuerga.inf.ubu.es/icruzado/tfc/OMT\\_res.pdf](http://pisuerga.inf.ubu.es/icruzado/tfc/OMT_res.pdf)

#### 4.1.1 Descripción del Problema

En base al análisis del problema realizado, se pudo obtener los siguientes problemas en los cuales se basa el análisis de requerimientos y para los cuales se usara los casos de uso para su especificación.

##### Casos de Uso

El diagrama de casos de uso representa la forma como un usuario (Actor) opera con el sistema en desarrollo, además de la forma , tipo y orden en como los elementos interactúan.

En el presente sistema se identificaron los siguientes actores.

- **Entidades Externas** (Empresas, instituciones, personas, etc.), son las personas e instituciones que envían correspondencia al CEMSE-CERPI, los cuales no pertenecen a la institución.
- **Entidades Internas** (Unidades o áreas), estas entidades pertenecen al CEMSE-CERPI las cuales manejan la documentación de la correspondencia y requieren de la misma internamente con la institución.
- **Administrador.**, es la persona encargada de la administración del sistema, el cual manipula y asigna los privilegios a los distintos usuarios.
- **Usuarios.**, son todas las personas que trabajan en el CEMSE-CERPI , los cuales tienen acceso al sistema de acuerdo a las necesidades y puestos que ocupan dentro de la institución.

**Caso de Uso General: Sistema de Registro y Seguimiento de Correspondencia y Asignación de Aulas para el CEMSE-CERPI**

**Caso de uso:** Sistema de Registro y Seguimiento de Correspondencia y Asignación de Aula.

**Actores** : Entidades externas, Entidades internas, Administrador, Usuarios.

**Propósito** : Identificar los usuarios que interactúan con el sistema.

**Descripción** :Se debe registrar la correspondencia Interna/externa y hacer su respectivo seguimiento de la misma , en cuanto a las solicitudes que se tenga de aula se debe registrar y a su vez verificar los horarios disponible , y los usuarios (CEMSE-CERPI) deberán registrar en su agenda las reuniones importantes u otros.

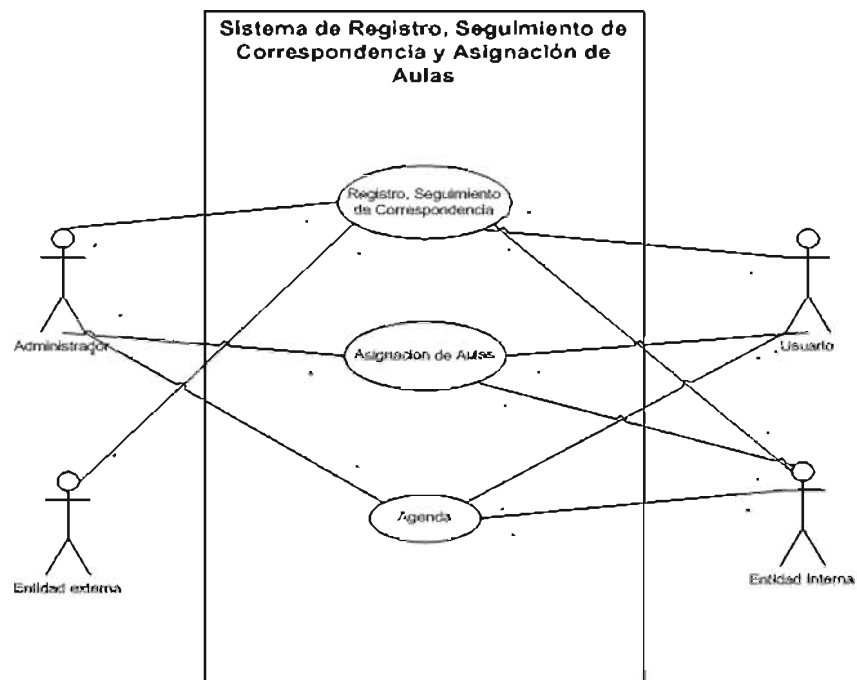


Figura. 4.2 Caso de Uso del Sistema de Registro, Seguimiento de Correspondencia y asignación de Aula.[Elaboración Propia]

### Caso de Uso Subsistemas

**Caso de uso:** Subsistema de Registro y Seguimiento de Correspondencia.

**Actores:** Entidades externas, Entidades internas, Administrador, Usuarios.

**Propósito:** Registrar y hacer el seguimiento de la correspondencia.

**Descripción:** Se realiza la recepción y registro de la correspondencia para su respectiva derivación a la unidad o área de destino correspondiente.

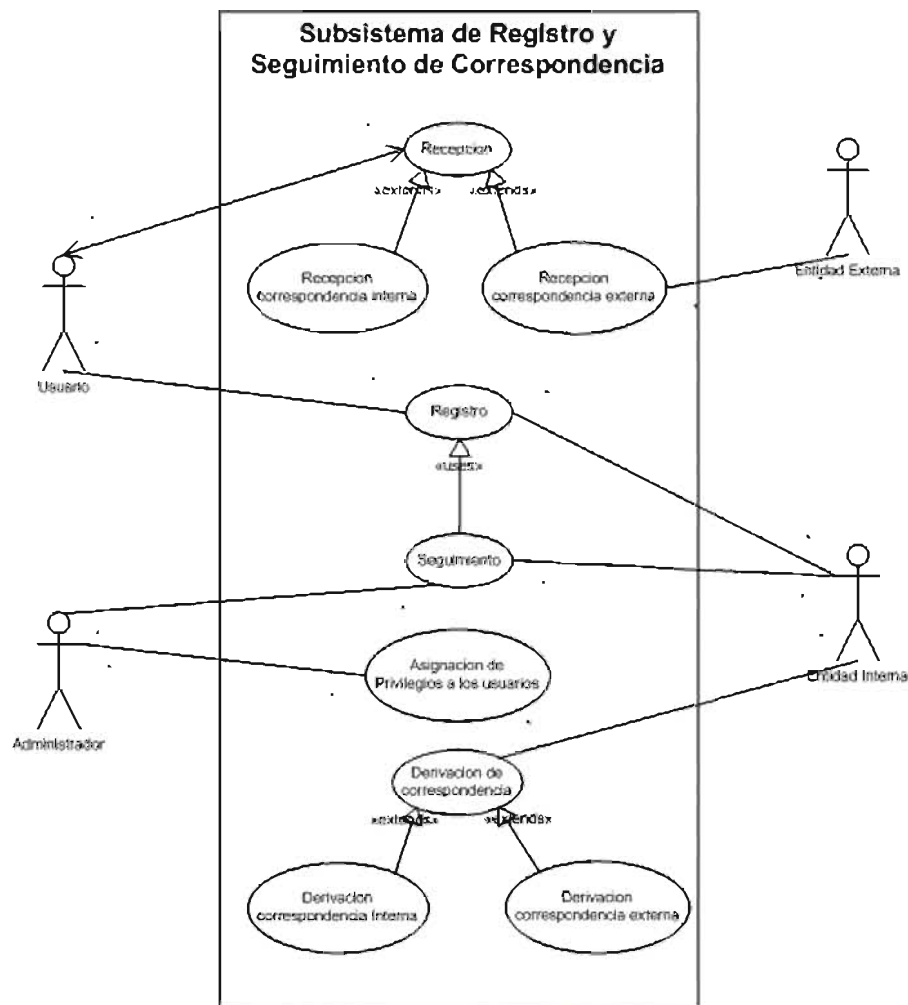


Figura: 4.3 Caso de uso Sistema de Registro y Seguimiento de Correspondencia. [Elaboración Propia]

**Caso de uso:** Subsistema de Asignación de Aula.

**Actores** : Entidades externas, Entidades internas, Administrador, Usuarios.

**Propósito** : Asignar aula correspondiente a las solicitudes que llegan a la institución.

**Descripción** : Se realiza la recepción y registro de la solicitud de aula que se requiere, se verifica disponibilidad y se eleva una respuesta.

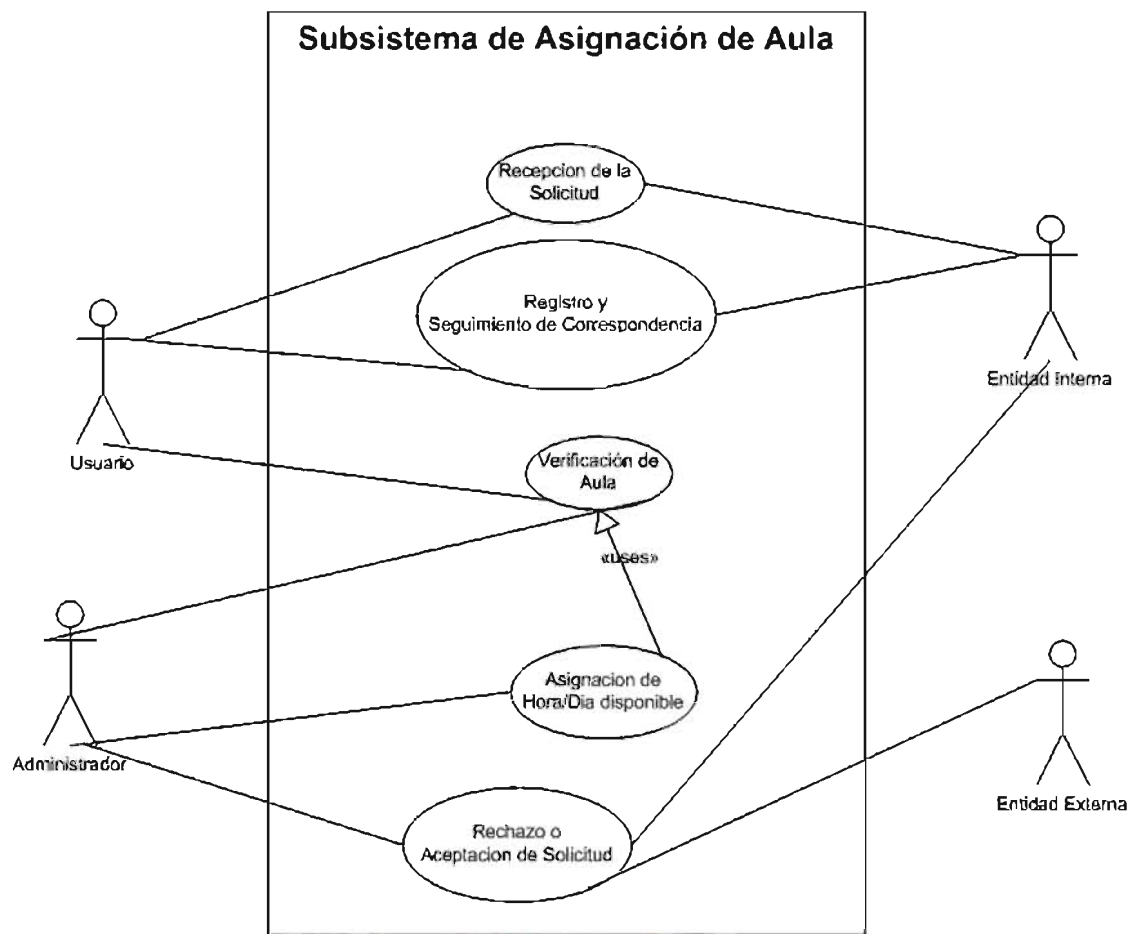


Figura: 4.4 Caso de uso Subsistema de Asignación de Aula. [Elaboración Propia]

**Caso de uso:** Subsistema de Agenda.

**Actores** : Administrador, Usuarios.

**Propósito** : Ayudar a los Usuarios a recordar las diversas actividades que tienen sin dejar de lado a ninguna por falta de comunicación.

**Descripción** : Se realiza el registro de las diferentes actividades que cada usuario tenga y colocando una hora y fecha de recordatorio para activar la alarma a tiempo de emprender dicha tarea.



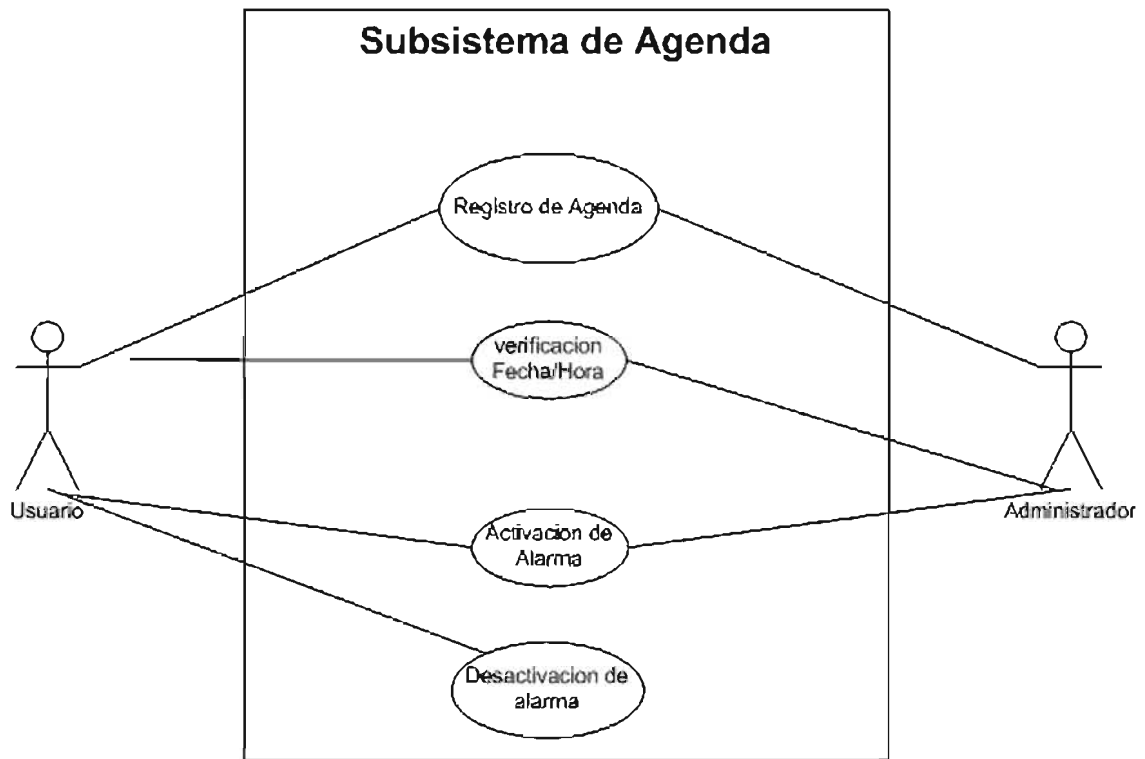


Figura. 4.5 Caso de Uso de Agenda [Elaboración Propia]

#### 4.1.2 Modelo de Objetos

Un modelo de objetos describe la estructura de los objetos de un sistema (identidad, relaciones con otros objetos, atributos y operaciones).

Se representa gráficamente mediante diagramas que contienen clases de objetos, para lo cual se usará la herramienta de UML.

### a) Identificación de clases y objetos



Figura: 4.6 Clases Identificadas [Elaboración Propia]

### b) Preparación del Diccionario de Datos

Esta actividad se utilizó para tener una mejor concepción de la problemática que encierra el proyecto, en la cual se realiza la descripción de cada clase de objeto identificados del sistema.

**Usuarios:** son todas las personas que trabajan en el CEMSE-CERPI , los cuales tienen acceso al sistema de acuerdo a las necesidades y puestos que ocupan dentro de la institución.

Relaciones:	Atributos:	Operaciones:
Realiza Seguimiento	Id_puesto	Autenticación
Introduce Datos Corresp.	Nivel_usuario	Registrar correspondencia
Tiene Agenda	Estado_usuario	
Tiene Puesto	Id_usuario	

**Correspondencia:** Son aquellos documentos que llegan a la institución (cartas, solicitudes, invitaciones, formularios, comunicados, etc.) .

Tiene hoja de ruta	id_corr	Crear _correspondencia
Pertenece a un tipo_corresp.	No_cite	Registrar_corresp
Tiene derivación	Fecha	Listar_corresp
Tiene seguimiento	Referencia	Borrar_corresp
	Destinatario_reg	Asignar_derivación
	Rec_env	
	Int_ext	
	Tipo	
	No_adj	
	Adjunto	
	Archivo	
	Ide_der	
	Id_usuario	

**Seguimiento:** se encarga de almacenar el estado de la correspondencia en que se encuentra indicando su ubicación y su proceso.

<b>Relaciones:</b>	<b>Atributos:</b>	<b>Operaciones:</b>
Pertenece_derivacion	No_corres	Verificar_estado
Pertenece_corresp	Estado_corresp	Listar_estados
	Descripcion	
	observaciones	

**Hoja de Ruta:** Registro que se adjunta a la correspondencia para el respectivo seguimiento que se debe realizar.

<b>Relaciones:</b>	<b>Atributos:</b>	<b>Operaciones:</b>
Pertenece a Corresp.	Num	Registrar_HR
Tiene derivación	Fecha	Cambios_HR
Tiene Agenda	Hora	Borrar_HR
Tiene Puesto	Origen	
	cite	
	ref	
	No_fojas	
	Tipo	
	Id_corresp	

**Administrador.**, es la persona encargada de la administración del sistema, el cual manipula y asigna los privilegios a los distintos usuarios.

<b>Relaciones:</b>	<b>Atributos:</b>	<b>Operaciones:</b>
Realiza Seguimiento	Id_puesto	Autenticación
Introduce Datos Corresp.	Nivel_usuario	Registrar correspondencia
Tiene Agenda	Estado_usuario	Asigna Privilegios
Tiene Puesto	Id_usuario	

### c) Identificar asociaciones entre objetos

#### ➤ Agregación

Las agregaciones que se identificaron entre objetos del sistema se observa en la fig.4.7.

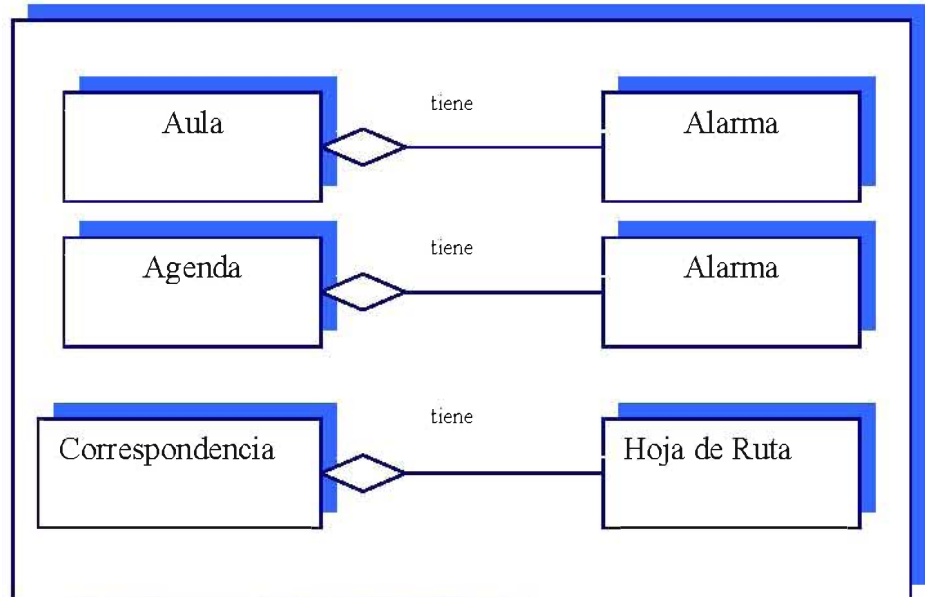


Figura 4.7 Agregación entre objetos del sistema

[Elaboración Propia]

## ➤ Multiplicidad

La multiplicidad entre objetos del sistema se observa en la figura siguiente:

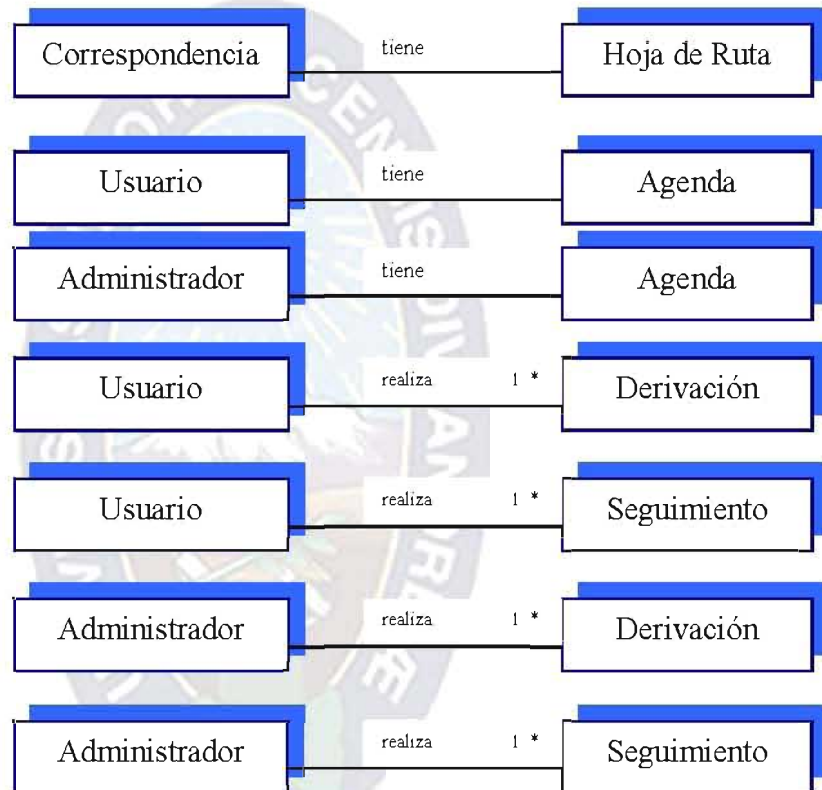


Figura 4.8 Diagrama de Relaciones y Asociaciones  
[Elaboración Propia]

**d) Identificar atributos de objetos**

En la figura 4.9 se observa a todas las clases identificadas del sistema con sus atributos y operaciones.

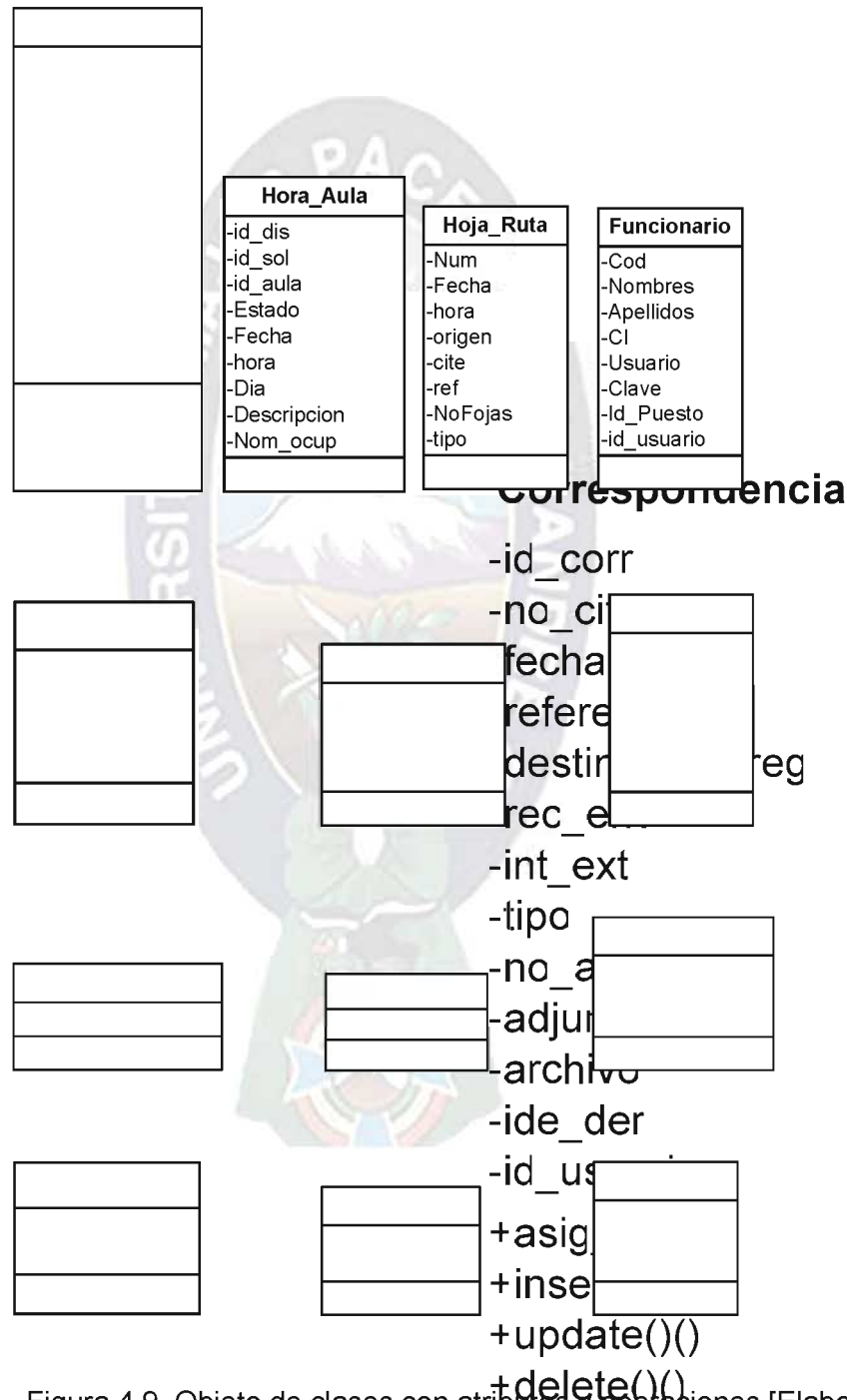


Figura 4.9 Objeto de clases con atributos y operaciones [Elaboración Propia]



**e) Organizar y simplificar las clases de objetos utilizando herencia.**

En la figura 4.10 se observa la herencia de la clase Funcionario donde la super clase funcionario hereda sus atributos y operaciones a las clases: usuario y administrador.

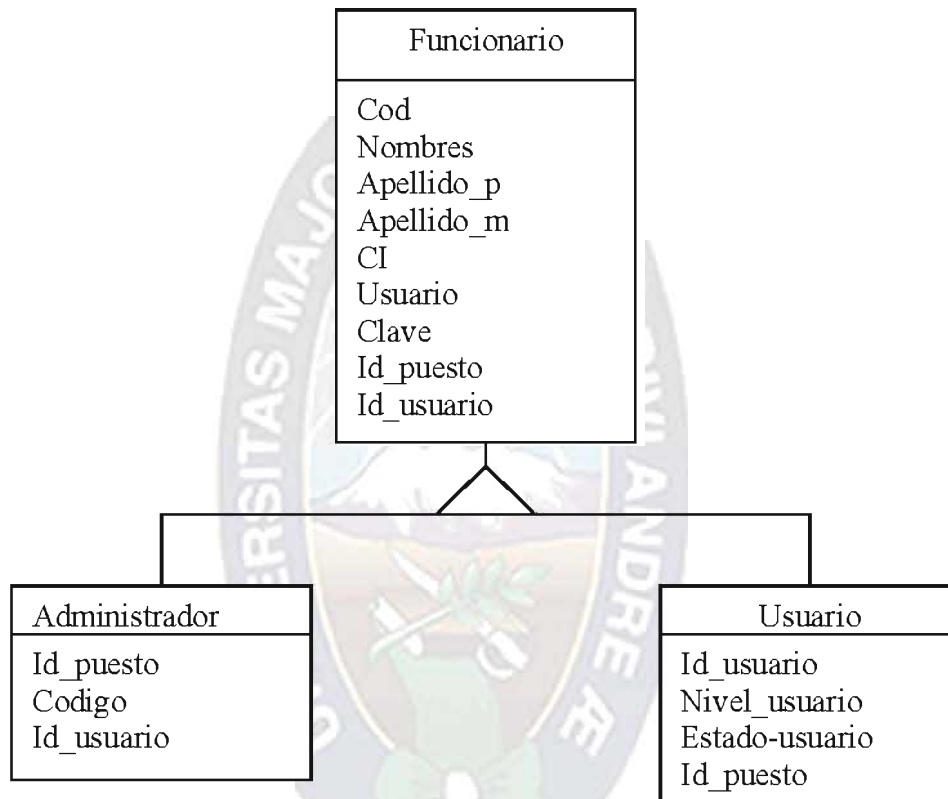


Figura 4.10 Herencia de la clase funcionario [Elaboración Propia]

**f) Iterar y refinar el modelo**

En este paso se realizó la iteración y se refinamiento de todas las etapas para la obtención del modelado de objetos. En la figura 4.11 se observa el diagrama general de objetos donde se aplicó todos los conceptos utilizados en la etapa de modelo de objetos.

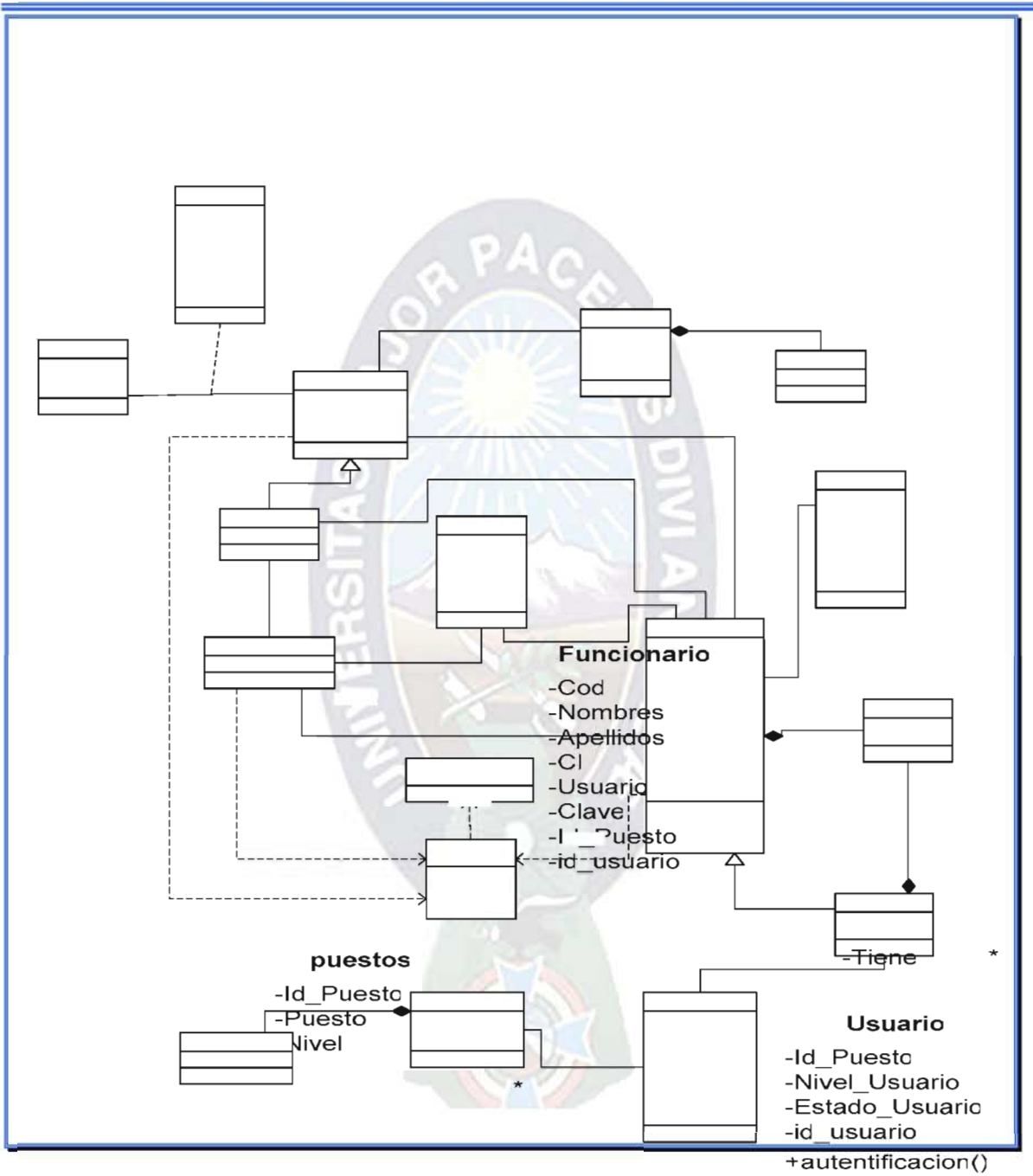


Figura 4.11 .Diagrama de Clases del Sistema de Registro y Seguimiento de Correspondencia y asignación de Aulas. [Elaboración Propia.]

### 4.1.3 Modelo Dinámico

Con lo descrito en los casos de uso , se estableció el escenario de interacción que se da entre el actor y el sistema mediante los casos de uso; el modelo dinámico representa el comportamiento de los objetos del sistema: para representar el comportamiento del sistema se emplea el diagrama de secuencia de UML para cada caso de uso y mediante el diagrama de estados se representó la secuencia de estados por los que pasa un caso de uso.

#### a) diagrama de secuencia

En la figura 4.12 y 4.13 se tienen los diagramas de secuencia del sistema.

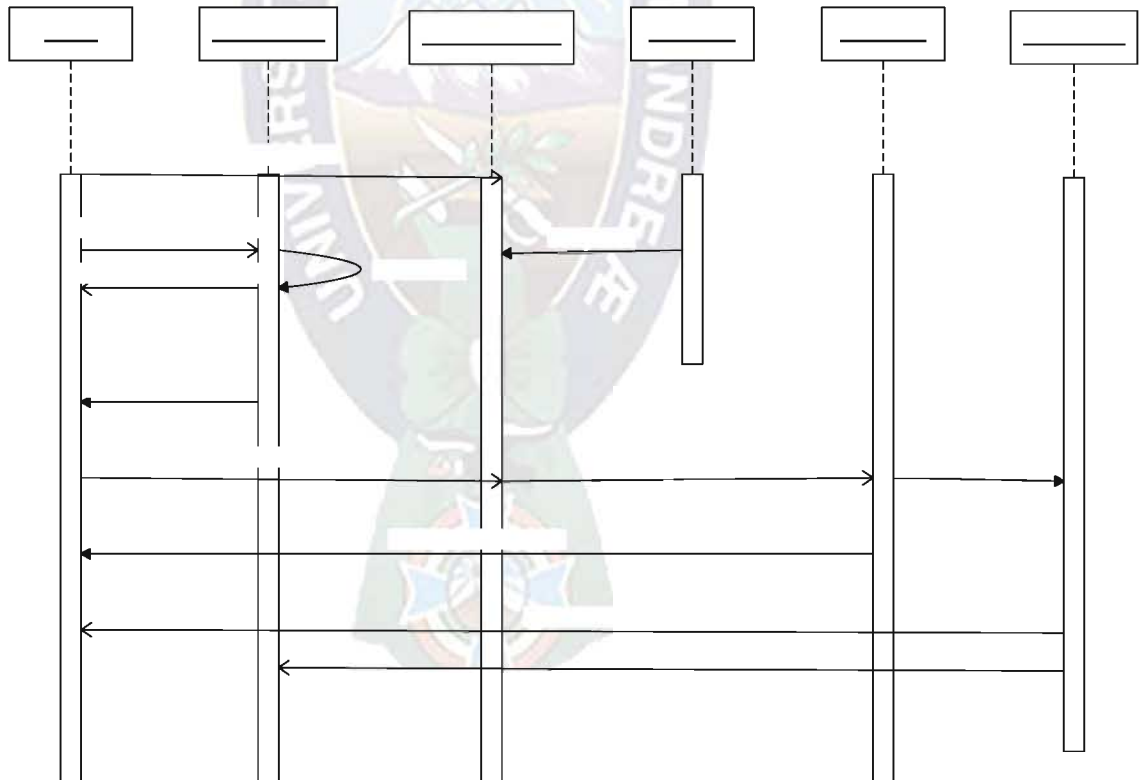
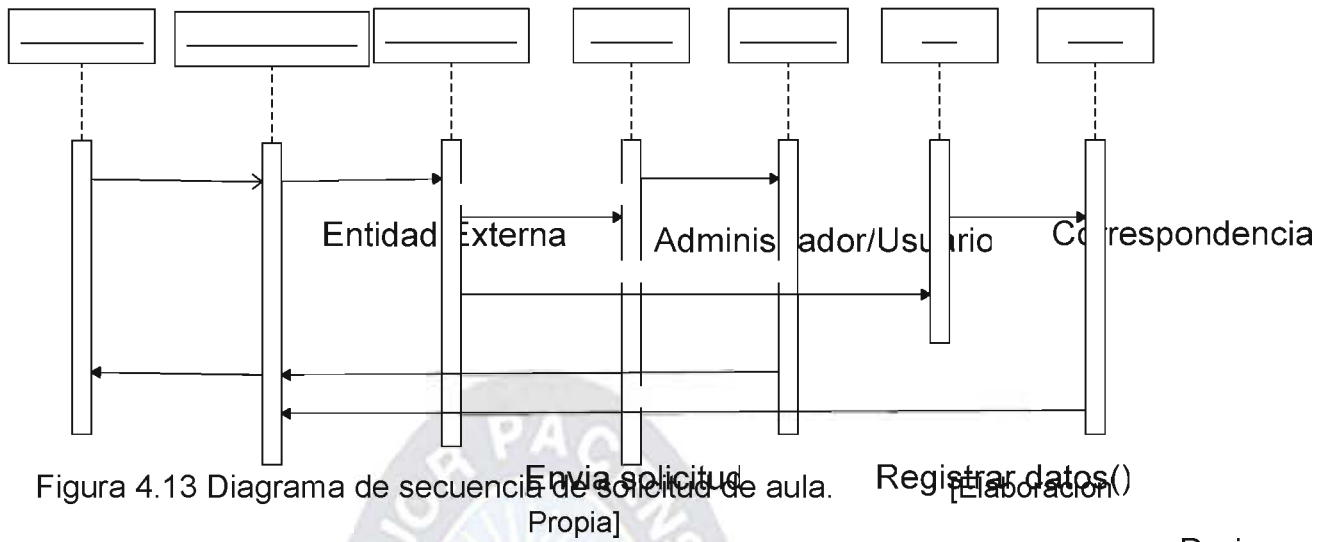


Figura 4.12 Diagrama de secuencia para el seguimiento de correspondencia. [Elaboración Propia]

Usuari

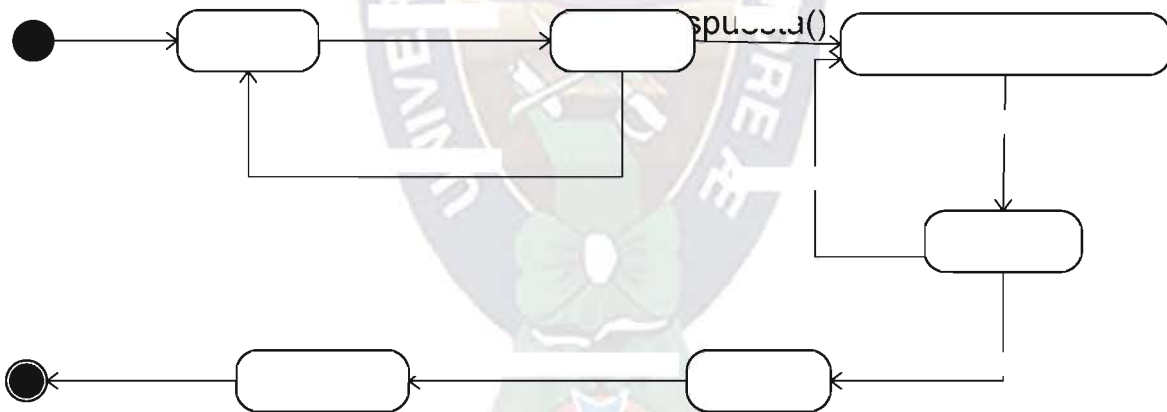
Administrador

Corres



### Diagrama de Estados

En la figura 4.14 y 4.15 se verán los diagramas de estados del sistema.



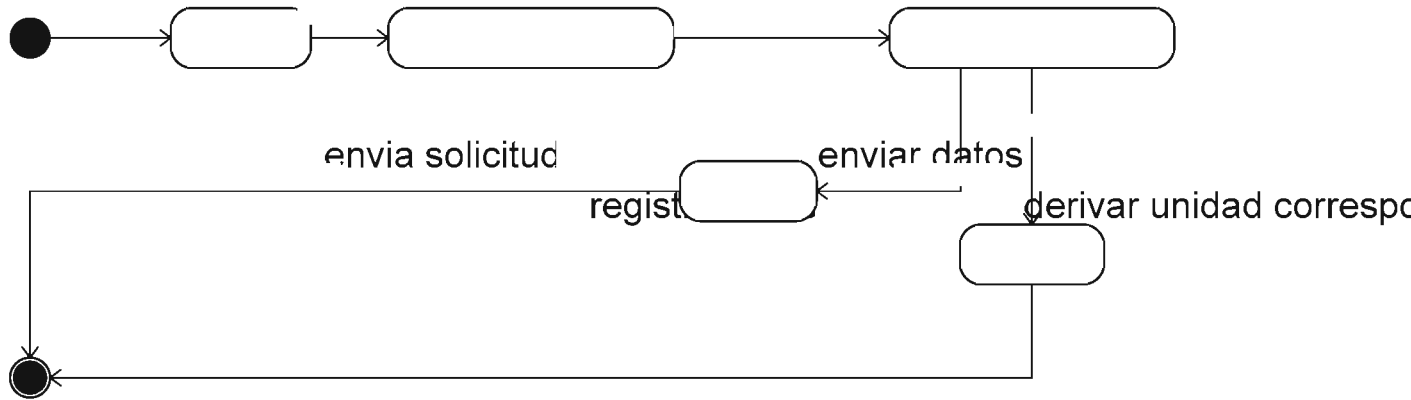


Figura 4.15 Diagrama de estado para la asignación de aula. [Elaboración Propia]

#### 4.1.4 Modelo Funcional

##### a) identificar los valores de entrada y de salida.

##### Valores de entrada

- administrador introduce clave de acceso
- numero de correspondencia ( registrar, consultar)
- datos correspondencia
- datos hoja de ruta
- solicitud de reporte
- solicitud de consulta
- solicitud de aula
- usuario solicita autenticación

emitir alarma

## Valores de salida

- sistema muestra mensaje de aceptado/ rechazado
- sistema muestra mensaje de clave incorrecto
- sistema muestra reportes
- sistema muestra consultas
- sistema muestra correspondencia, solicitudes
- sistema muestra respuesta de solicitudes

**b) Construir diagramas de flujo de datos que muestren las dependencias funcionales.**

En la figura 4.16 se observa el diagrama de contexto que corresponde al diagrama de flujo de datos de más alto nivel de las clases con interacción externa.

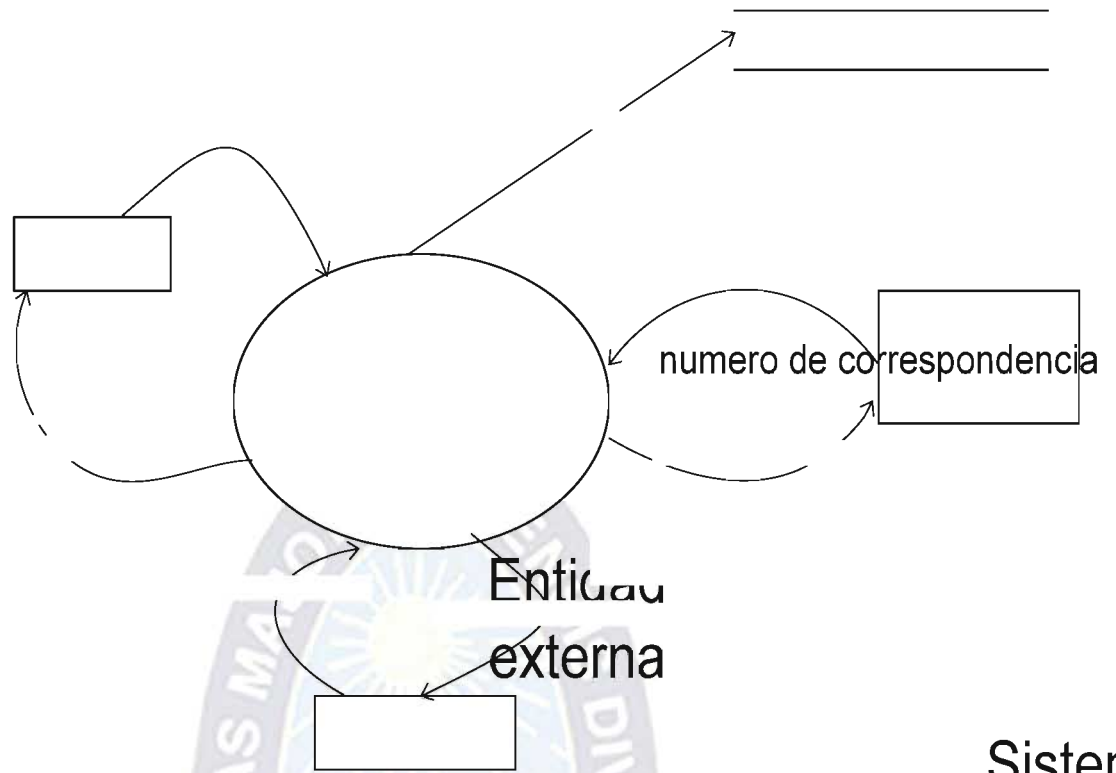


Figura 4.16 Diagrama de contexto del sistema. [Elaboración Propia]

consulta de estado de correspondencia

Sistema de  
seguimi  
correspondenci  
de a

Después de desarrollar el diagrama de contexto se detalla los diagramas de flujo de datos, un diagrama por clase, mostrando como los valores de salida se computan de los valores de entrada. En la fig. 4.17 se detalla el diagrama de flujo de datos de alto nivel.

Usua



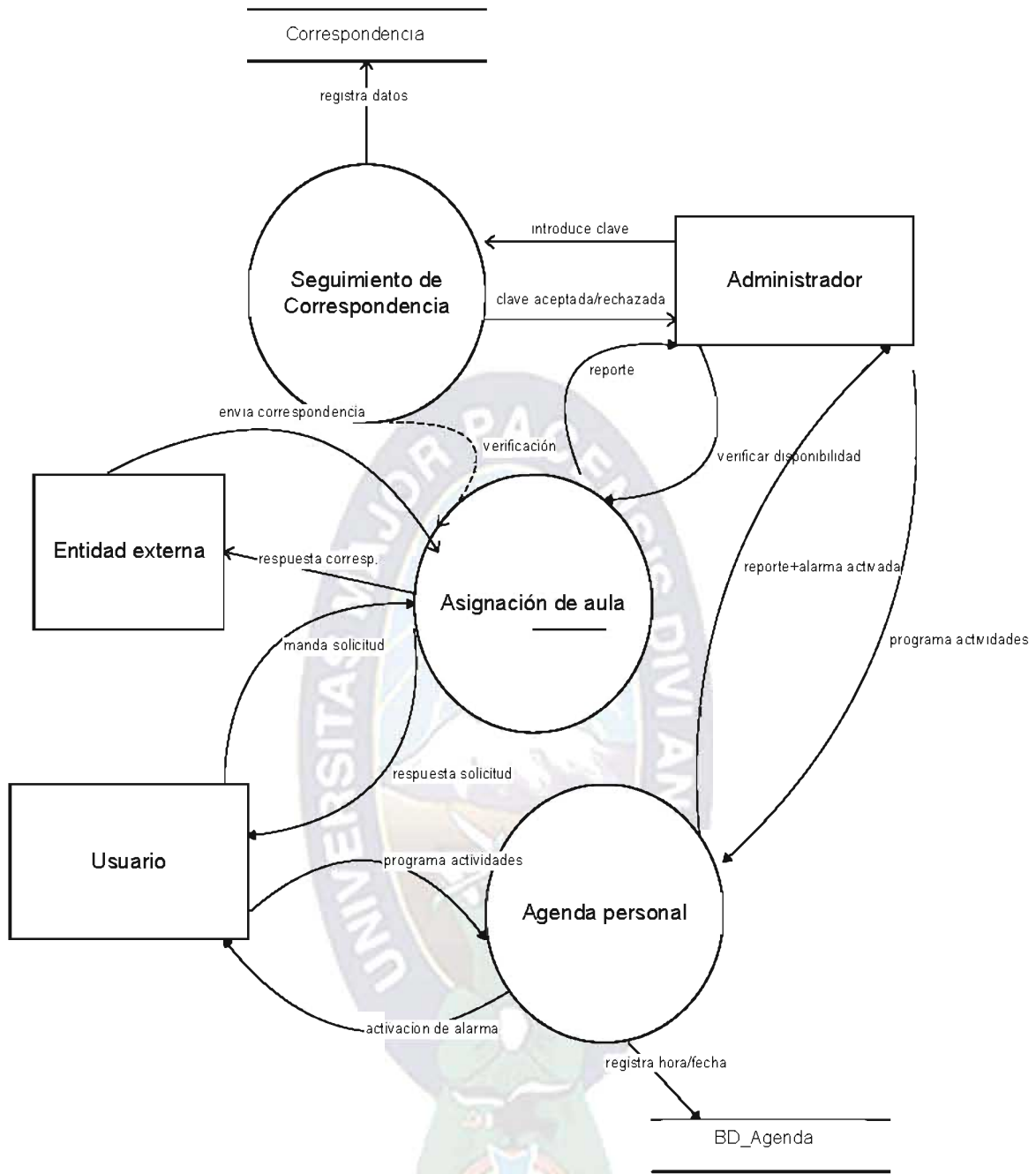


Figura 4.17. Diagrama de Flujo de Datos de alto nivel de todo el sistema. [Elaboración Propia]

## 4.2 DISEÑO

### 4.2.1 Diseño de Sistema

#### a) organizar el sistema en subsistemas.

Cada componente del sistema se agrupa según aspectos comunes. Cada subsistema identificado se encarga de procesar la informaron generad por el sistema. Estos subsistemas son los siguientes:

- Seguimiento de Correspondencia
- Registro de correspondencia
- Derivación de correspondencia
- Asignación de aula
- Agenda personal
- Generación de Reportes

#### b) Identificar la concurrencia inherente al problema.

Los procesos del sistema tienen una funcionalidad secuencial, es decir, los procesos se ejecutan paso a paso para realizar el proceso de nominación. Por esta razón los procesos del sistema no son considerados como concurrentes.

### c) **Asignar los subsistemas a los procesadores y tareas**

Todos los subsistemas son asignados a una unidad de hardware, donde se distingue lo siguiente:

Para la implementación del sistema se requiere recursos de hardware necesarios para tener un mejor rendimiento en los subsistemas.

La seguridad de acceso a los datos garantiza el control óptimo, concediendo y retirando privilegios de acceso hacia un objeto de la Base de Datos por el administrador del Sistema.

Las plataformas en las que el sistema trabaja se describen a continuación:

**Sistema Operativo:** Windows 98, XP.

**Motor de Base de Datos:** MySQL Server 4.1.9, para el almacenamiento de los datos

### d) **Seleccionar una aproximación para la administración de almacenes de datos.**

El almacenamiento de los datos en una base de datos pueden ser relacionales, jerárquicas y orientada a objetos. Cabe mencionar que los DBMS (Data Base Management Systems) Orientado a objetos aun no alcanzó una amplia difusión comercial aunque tiene un futuro prometedor.

El sistema se implanto en una base de datos relacional, teniendo en cuenta que el análisis se desarrollo en un entorno orientado a objetos es necesario considerar algunos aspectos en la implementación con la base de datos, para esto se realizo el siguiente proceso.

El modelado de objetos esta relacionado con la arquitectura en tres esquemas. En primer lugar se formula modelos de objetos para los esquemas externos y conceptual, a continuación se traducen los modelos en tablas ideales, para luego implementarlas con esquemas internos de código MySQL [Rumbaugh et al 1996]. Además toda clase de objeto se corresponde con una o más tablas.

En la figura 4.19 se observa como una clase de objetos se transforma en una tabla mediante código MySQL..

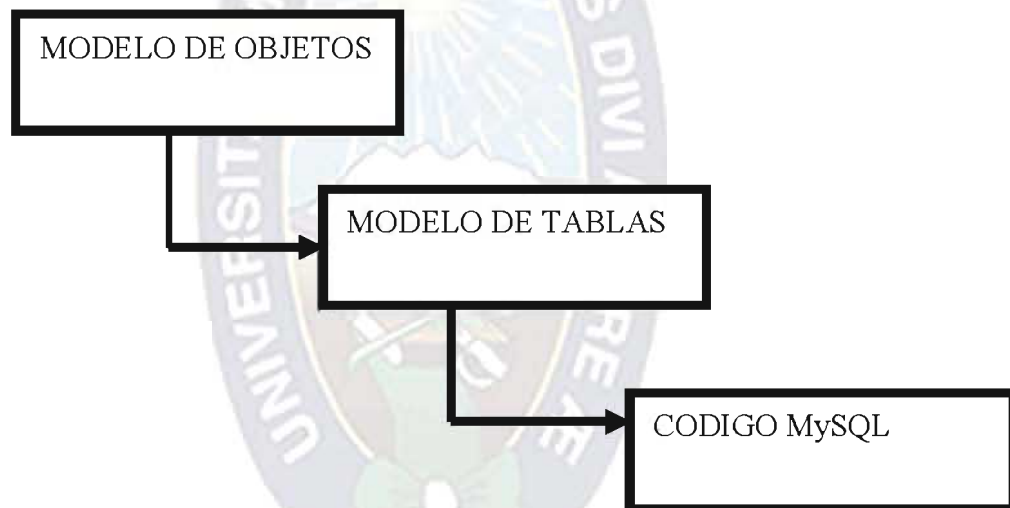


Figura 4.18 Transformación de Objeto a Código. [Elaboración Propia]

Correspondencia
-id_corr
-no_cite
-fecha
-referencia
-destinatario_reg
-rec_env
-int_ext
-tipo
-no_adj
-adjunto
-archivo
-ide_der
-id_usuario
+asig_der()
+insert()
+update()
+delete()

Alter Table 'correspondencia' in 'dbsicc'								
Field Name	Datatype	Len	Default	Collation	PK?	Binary?	Not Null?	
id_corr	int	11	0		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
no_cite	int	11			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
fecha	date				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
referencia	varchar	50		latin1_sw...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
destinatario_reg	varchar	50		latin1_sw...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
rec_env	varchar	30		latin1_sw...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
int_ext	varchar	30		latin1_sw...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
tipo	varchar	50		latin1_sw...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
no_adj	varchar	10		latin1_sw...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
adjunto	varchar	50		latin1_sw...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
• archivo	varchar	250		latin1_sw...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
id_der	int	11			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
id_usuario	int	11			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

```

CREATE TABLE `correspondencia` (
  `id_corr` int(11) NOT NULL default '0',
  `no_cite` int(11) default NULL,
  `fecha` date default NULL,
  `referencia` varchar(50) default NULL,
  `destinatario_reg` varchar(50) default NULL,
  `rec_env` varchar(30) default NULL,
  `int_ext` varchar(30) default NULL,
  `tipo` varchar(50) default NULL,
  `no_adj` varchar(10) default NULL,
  `adjunto` varchar(50) default NULL,
  `archivo` varchar(250) default NULL,
  `id_der` int(11) default NULL,
  `id_usuario` int(11) default NULL,
  PRIMARY KEY (`id_corr`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1

```

Figura 4.19 Transformación de la clase Correspondencia a la tabla correspondencia. [Elaboración Propia]

**e) Seleccionar la implementación de control en software**

El sistema estará controlado mediante el uso de procedimientos, entonces el control reside dentro del código del programa.

**f) Manejar las condiciones de contorno.**

Las condiciones de contorno o borde están implementadas en módulos de procedimientos internos del sistema.

**g) Establecer la compensación de prioridades**

Lo más importante en el sistema es la reserva de almacenamiento para los datos que son procesados en el seguimiento de la correspondencia.

**h) Arquitectura del sistema**

El sistema funciona bajo una arquitectura de administración de transacciones, su propósito es mantener la base de datos para la información y actualizarse sobre la red que consiste de un modelo de objetos.

El movimiento de la información es realizado a través de un trámite. En primer lugar se deprecia la correspondencia adicionando una Hoja de Ruta para su movimiento y derivación a otras instancias, además de poder realizar un seguimiento de la correspondencia para verificar el estado en el que se encuentra.

Una vez concluido el seguimiento (cuando su estado es archivado o despachado) en la hoja de ruta, el usuario o administrador

interactúa con el sistema registrando la última derivación con la hoja de ruta correspondiente.

Es posible generar reportes y realizar consultas, utilizando los datos disponibles almacenados en la base de datos del sistema.

#### 4.2.2 Diseño de Objetos

La transformación de la clase trámite a la tabla trámite transfiere los atributos a campos como se observa en la figura 4.19.

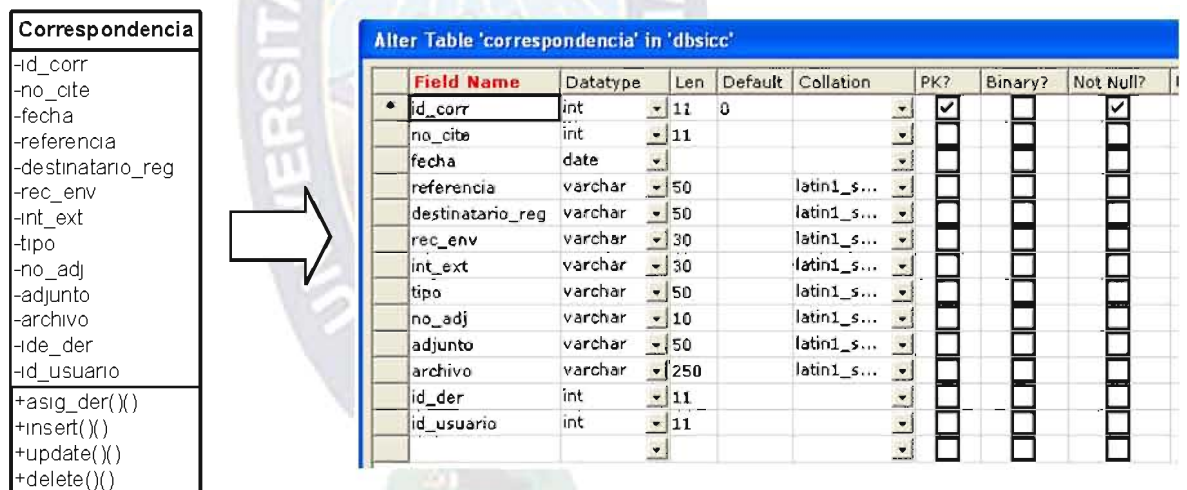


Figura 4.20 Transformación de la clase correspondencia a la tabla correspondencia. [Elaboración Propia]



Observese que la tabla correspondencia presenta un campo que se identifica como clave primaria o llave primaria, que es el id\_corr de tipo int y longitud 11, no permite valores nulos garantizando que la información sea íntegra.

Otro ejemplo de transformación de la clase Usuario a la tabla usuario que se observa en la figura 4.20.

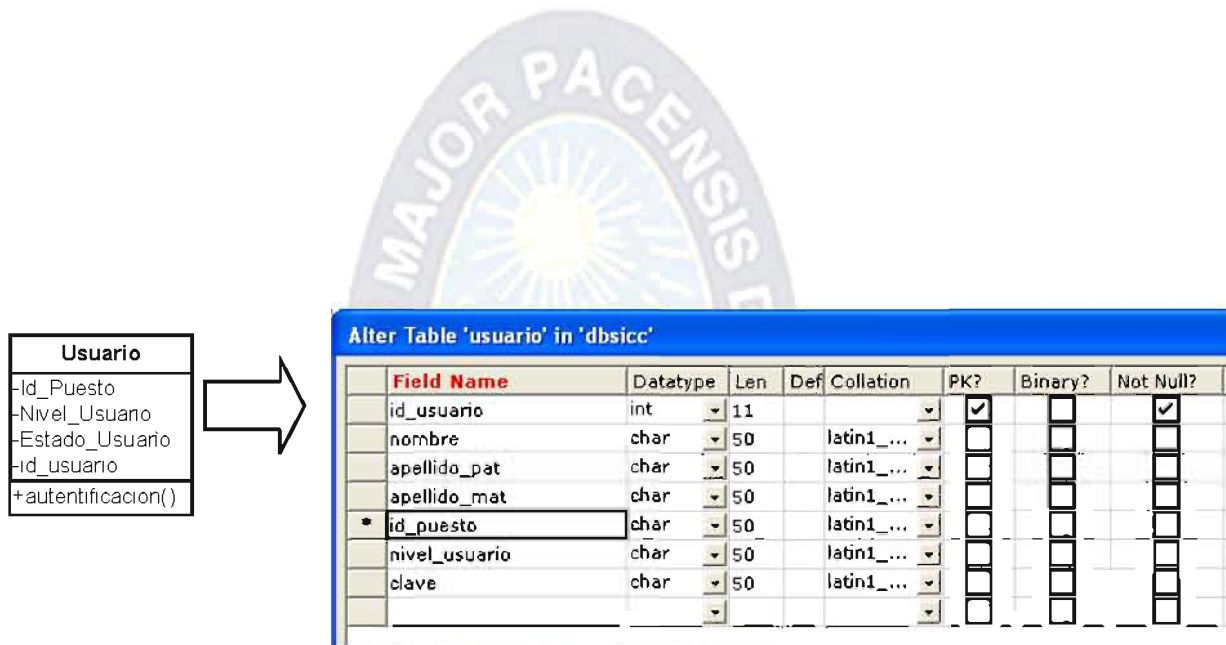


Figura 4.21 Transformación de la clase usuario a la tabla usuario.

[Elaboración Propia]

En esta tabla como en el anterior se tiene en la tabla un campo de clave primaria, que es el id \_ usuario de tipo int y longitud 11, el cual no permite valores nulos garantizando que la información sea íntegra.

### 4.3 IMPLEMENTACION

Comenzando por la elaboración de los procesos de análisis y diseño se completa la etapa de implementación.

### 4.3.1 Codificación

En la figura 4.21 se observa la codificación correspondiente a la tabla correspondencia mediante el código MySQL.

```
CREATE TABLE `correspondencia` (  
  `id_corr` int(11) NOT NULL default '0',  
  `no_cite` int(11) default NULL,  
  `fecha` date default NULL,  
  `referencia` varchar(50) default NULL,  
  `destinatario_reg` varchar(50) default NULL,  
  `rec_env` varchar(30) default NULL,  
  `int_ext` varchar(30) default NULL,  
  `tipo` varchar(50) default NULL,  
  `no_adj` varchar(10) default NULL,  
  `adjunto` varchar(50) default NULL,  
  `archivo` varchar(250) default NULL,  
  `id_der` int(11) default NULL,  
  `id_usuario` int(11) default NULL,  
  PRIMARY KEY (`id_corr`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1
```

Figura 4.22 Código MySQL para la implementación de la tabla correspondencia. [Elaboración Propia]

En la siguiente figura 4.23 se observa la codificación correspondiente a la tabla usuario mediante el uso de código MySQL.

```
CREATE TABLE `usuario` (  
  `id_usuario` int(11) NOT NULL auto_increment,  
  `nombre` char(50) default NULL,  
  `apellido_pat` char(50) default NULL,  
  `apellido_mat` char(50) default NULL,  
  `id_puesto` char(50) default NULL,  
  `nivel_usuario` char(50) default NULL,  
  `clave` char(50) default NULL,  
  PRIMARY KEY (`id_usuario`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1
```

Figura 4.23 Código MySQL para la implementación de la tabla Usuario. [Elaboración Propia]

### 4.3.2 Diseño de la Interfaz

#### ❖ Autenticación de Usuario:

- ✓ Login: Se coloca el Login del usuario o nombre de usuario.
- ✓ Password: Se coloca la clave o llave de acceso al Sistema.

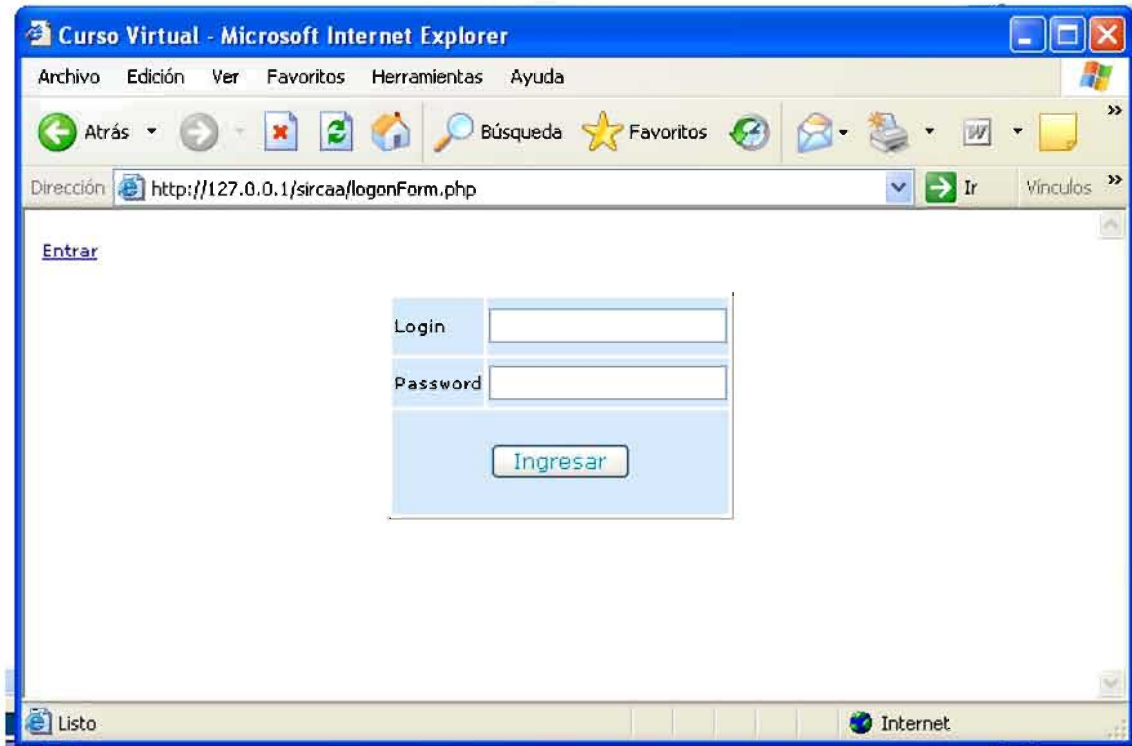


Figura 4.24 Diseño de interfaz de la autenticación de usuario. [Elaboración Propia]

#### ❖ Diseño de Menu:

- ✓ Correspondencia: Este menú tiene las opciones de Registro de Correspondencia, Listado de Correspondencia y Nueva Carta.
- ✓ Usuarios: Se registra a todos los usuarios que manejaran el sistema.

- ✓ Derivación: En este menú se envía la correspondencia que se registra al destino que corresponde.
- ✓ Agenda: En este menú se coloca las actividades de cada usuario.
- ✓ Aula: Tiene las opciones de registrar las diferentes aulas del CEMSE-CERPI, además de registrar las actividades de las diferentes aulas.
- ✓ Cerrar Sesión: Nos permite cambiar de usuario.

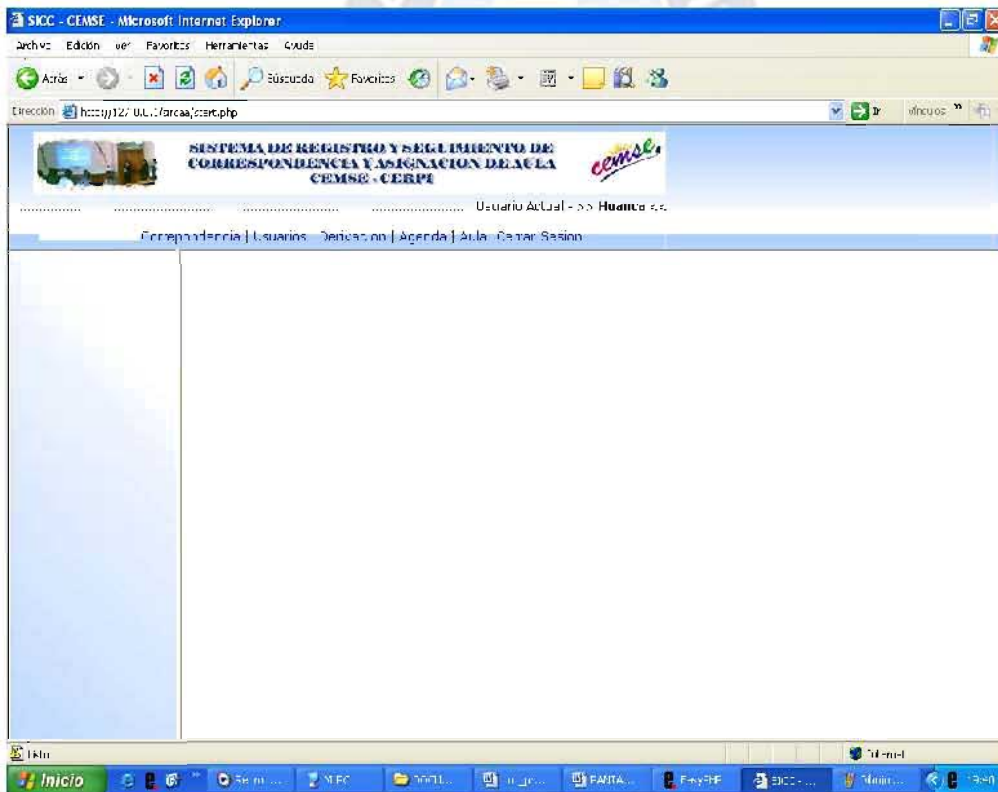


Figura 4.25 Diseño de interfaz del Menú Principal [Elaboración Propia]

❖ Menu Correspondencia:

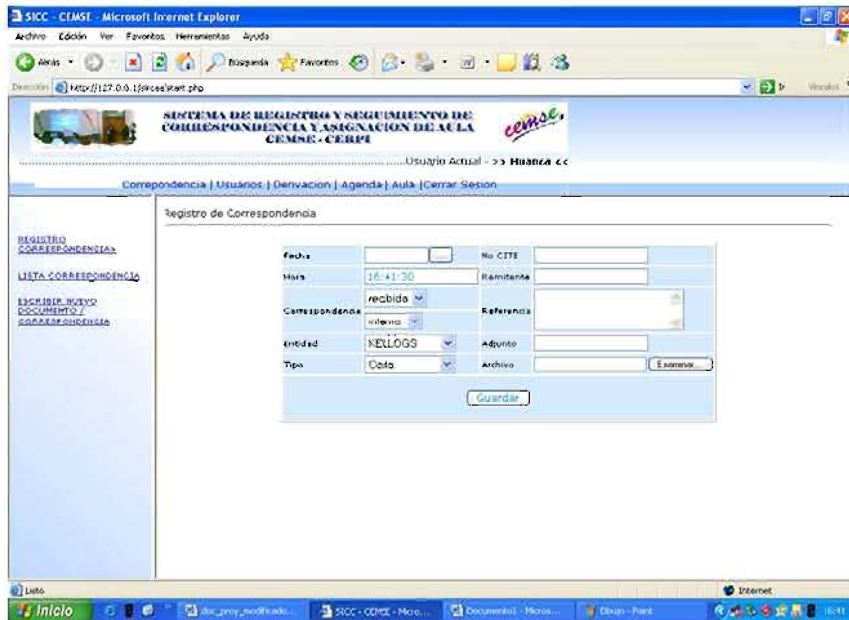


Figura 4.26 Diseño de interfaz del Menu Registro de Correspondencia.

[Elaboración Propia]

❖ Listado de Correspondencia:



Figura 4.27 Formulario de Listado de Correspondencia Registrada. [Elaboración Propia]

❖ Redacción de Correspondencia:

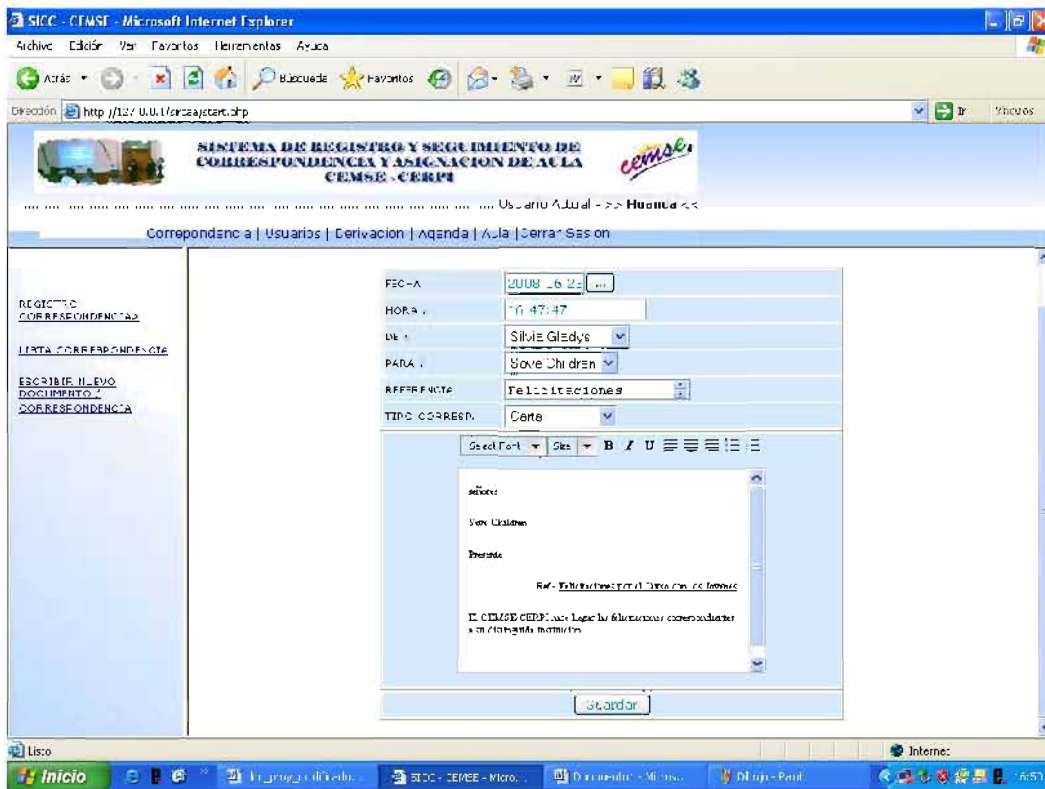


Figura 4.28 Formulario de Redacción de Correspondencia. [Elaboración Propia]

❖ Registro de Usuarios:

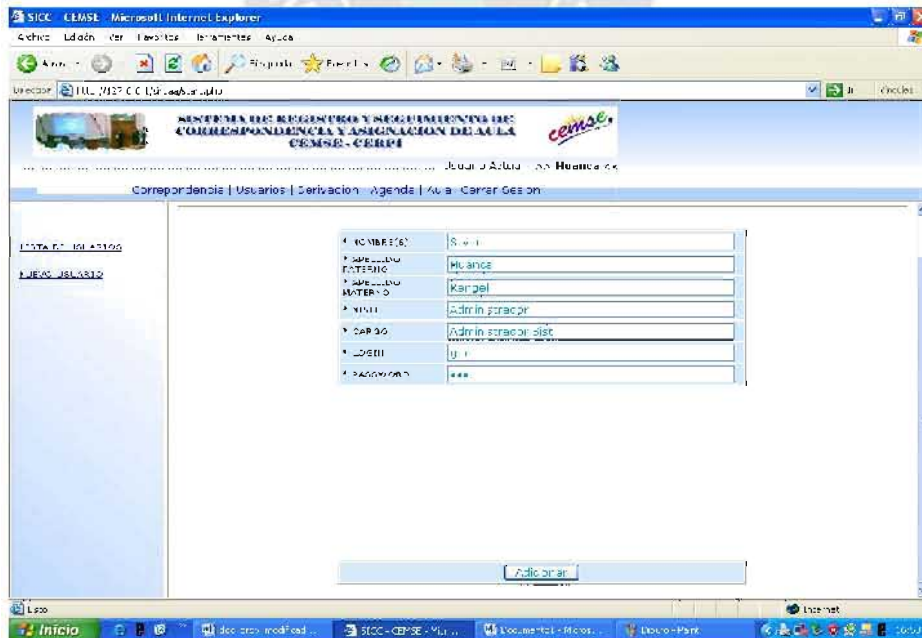


Figura 4.29 Diseño de interfaz del Menú Usuario. [Elaboración Propia]



❖ Listado de Usuarios:

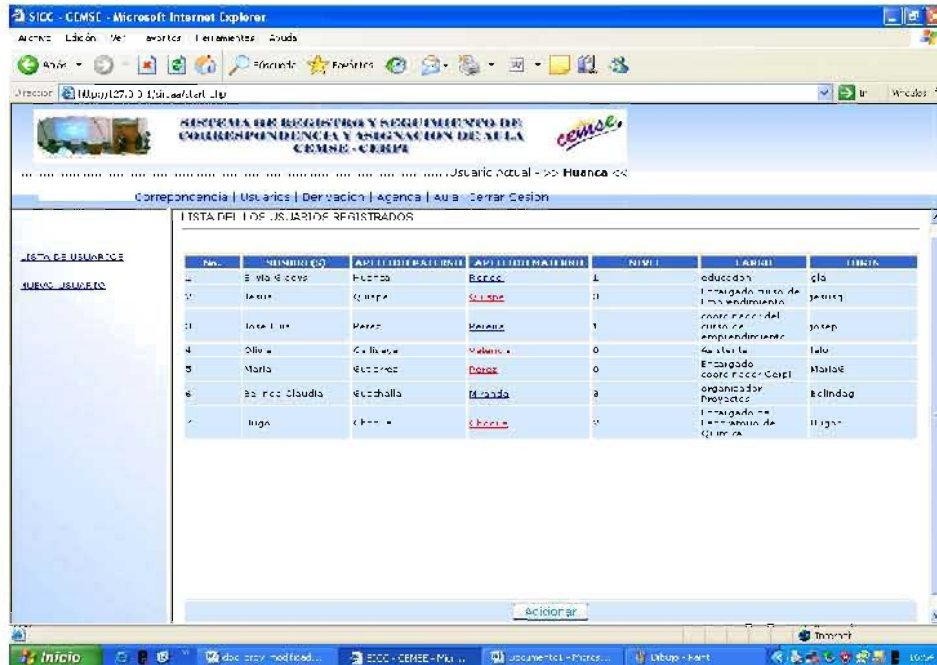


Figura 4.30 Listado de los diferentes usuarios habilitados para manejar el sistema. [Elaboración Propia]

❖ Listado de Actividades programadas para una aula:



Figura 4.31 Formulario de Vista de Horario en la semana correspondiente. [Elaboración Propia]



El diseño de las pantallas fueron desarrolladas y diseñados para que los usuarios puedan entender con facilidad, aun aquellos que no estén interiorizados con el sistema.

- ❖ **Interfaz de de Acceso al Sistema**, según Wiederhold [Wiederhold, 1998], esta interfaz permite controlar y mantener la seguridad del sistema de acuerdo a los permisos y contraseñas asignados por el administrador al usuario, el cual se puede ver esto aplicado al sistema en la **Figura 4.24**.
- ❖ **Interfaz de Tipo Menús**, los menús pueden ser ajustados para obtener entradas por el teclado o ratón. Por ello la interfaz de menús proporciona al usuario una lista en la parte superior de opciones en la pantalla de la **figura 4.25**, y al elegir una de las opciones se habilita en la parte izquierda con todas las opciones del menú elegido.
- ❖ **Interfaz de Entrada de Datos**, esta interfaz permite el registro de datos, los cuales se almacena en la base de datos. En la **figura 4.26** ... se introducen datos para registrar la correspondencia y usuarios correspondientemente.

# 5. CALIDAD DEL SOFTWARE

## 5.1 INTRODUCCION

El propósito de la ingeniería de software es construir un software de calidad. Por consiguiente en este capítulo se detalla cuantitativamente la calidad del Sistema de Registro y Seguimiento de correspondencia y Asignación de aulas a través de la aplicación de las métricas.

La medición de calidad de software se la realiza a través de métricas de control de calidad, basados en el estándar ISO 9126 que toma en cuenta los siguientes aspectos [PRES, 2000].

- ✓ Confiabilidad
- ✓ Funcionalidad
- ✓ Portabilidad
- ✓ Rendimiento
- ✓ Mantenibilidad

## 5.2 CONFIABILIDAD

La confiabilidad es un factor importante en la evaluación de la calidad general del software, para el análisis de confiabilidad de cada componente o módulo del sistema, se aplica la ley de distribución exponencial de fallas como una aplicación en la teoría de la confiabilidad; de la cual se define como sigue: "la confiabilidad  $R(t)$  de un componente durante un periodo de  $[0,t]$  o que es lo mismo a la probabilidad de que falle en un tiempo mayor que  $t$ ", es decir se tiene la ecuación :

$$R(t)=P[T>t]=1-G(t)$$

Siendo T el tiempo para fallar es una variable aleatoria exponencial, entonces se tiene:

$$R(t)=1-[1-e^{-\lambda t}]$$

$$R(t)= e^{-\lambda t}$$

Donde:

$\lambda$ : Es la tasa de constantes de fallo ( $\lambda$ =número de fallos de acceso/Numero total de acceso al sistema).

t: Período de operación en tiempo.

Una vez obtenida la función de confiabilidad, se irá analizando la confiabilidad de cada uno de los módulos del sistema.

Durante las pruebas realizadas al Sistema de Registro y Seguimiento de Correspondencia y Asignación de Aulas, de un periodo de tres días y 8 horas de trabajo por día se presento un promedio de fallas por módulo, el cual se puede apreciar en la tabla 4.1.

	<b>MODULO</b>	<b>ERROR(%)</b>
R1	Seguimiento de Correspondencia.	0.15
R2	Registro de correspondencia	0.1
R3	Derivación de correspondencia	0.1
R4	Asignación de aula	0.15
R5	Agenda personal	0.1
R6	Generación de Reportes	0.1

Tabla 5.1 Fallas del Sistema. [Elaboración Propia]

Donde, t es una variable de tiempo;; entonces t=24 horas, además la confiabilidad del sistema esta dada por;;

$$R(t) = R_1(t) R_2(t) \dots\dots\dots R_n(t)$$

Reemplazando los valores, para t=24 horas se tiene de la siguiente manera:

$$R_1(t) = e^{-(0.0015)(24)} = 0.96$$

$$R_2(t) = e^{-(0.001)(24)} = 0.98$$

$$R_3(t) = e^{-(0.001)(24)} = 0.98$$

$$R_4(t) = e^{-(0.0015)(24)} = 0.96$$

$$R_5(t) = e^{-(0.001)(24)} = 0.98$$

$$R_6(t) = e^{-(0.001)(24)} = 0.98$$

Luego empleando la ecuación se tiene:

$$R(t) = R_1(t) R_2(t) R_3(t) R_4(t) R_5(t) R_6(t) = 0.85 \approx 85\%$$

Lo que significa que el sistema funcione de forma consistente y correcta durante el periodo de tiempo determinado de tres días y 8 horas, es decir el 80% de probabilidad el sistema operó sin fallos bajo determinadas condiciones; sabiendo que la escala de confiabilidad es expresado entre 0 y 1; vale decir que el sistema tiene una medida de confiabilidad cercana a 1.

### 5.3 FUNCIONALIDAD

La funcionalidad puede representarse mediante los puntos de función que es una medida indirecta del software y del proceso por el cual se desarrolla, se centran en la funcionalidad o utilidad del programa.

Para la funcionalidad se toman en cuenta los siguientes 5 aspectos:

Parámetro de medición	Cuenta	Factor de ponderación			Cuenta Total
		Simple	Medio	Complejo	
Número de entradas de usuario	8	3	4	6	24
Número de salidas de usuario	10	4	5	7	40
Número de peticiones de usuario	7	3	4	6	21
Número de archivos	23	7	10	15	161
Número de interfaces externas	7	5	7	10	35
<b>Total</b>					<b>281</b>

Para evaluar cada factor se realiza en una escala de 0 a 5 como se observa a continuación.

Incidental	1
Moderado	2
Medio	3
Significativo	4
Esencial	5

Nº	Cuestionario	Factor
1	¿Requiere el sistema copias de seguridad y de recuperación fiable?	4
2	¿Se requiere comunicación de datos?	5
3	¿Existen funciones de procesamiento distribuido?	0
4	¿Es crítico el rendimiento?	1
5	¿Se ejecutará el sistema en un entorno operativo existente y fuertemente utilizado?	4
6	¿Requiere entrada de datos interactiva?	4
7	¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?	4
8	¿Se actualizan los archivos maestros de forma interactiva?	5
9	¿Son complejas las entradas. Las salidas, los archivos o las peticiones?	2
10	¿Es complejo el procesamiento interno?	3
11	¿Se ha diseñado el código para ser reutilizable?	3
12	¿Están incluidas en el diseño la conversión y la instalación?	2
13	¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	3
14	¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?	4
$\Sigma f_i$		44

La cuenta total obtenida debe ajustarse utilizando la ecuación del punto función (PF).

$$PF = \text{Cuenta Total} * (\text{confiabilidad} + \text{Margen de error} * \sum f_i)$$

Reemplazando se tiene el siguiente valor:

$$PF = 281 * (0.85 + 0.01 * 44) = 362.49$$

Si  $f_i = 75$  considerando el 100% reemplazando en la ecuación se tiene:

$$PF = 281 * (0.85 + 0.01 * 75) = 449.60$$

Luego la funcionalidad se calcula mediante la ecuación :

$$PF/PF' = 362.49/449.60 = 0.81$$

Esto quiere decir que la funcionalidad de los sistemas es de 81%.

#### 5.4 PORTABILIDAD

La portabilidad es la facilidad con que el software puede ser llevado de un entorno Hardware/Software a otro. Por lo cual la portabilidad del sistema se detalla a continuación:

Hardware: El sistema de Registro y Seguimiento de correspondencia y Asignación de Aulas son potables a computadoras a Pentium III y PC compatibles.

Software: Los códigos de los sistemas no son potables a otros lenguajes de desarrollo de manera directa, por poseer características y especificaciones propias de sus compiladores.

En cuanto a la plataforma el sistema esta desarrollado para sistemas operativos de la familia de Microsoft Windows, como son Windows 98, Windows 2000, Windows XP, etc.

## 5.5 RENDIMIENTO

Se mide por la velocidad de procesamiento, tiempo de respuesta consumo de recursos, rendimiento efectivo total y eficacia.

La complejidad y el tiempo de ejecución general de los sistemas es  $O(n)$  que es el peor de los casos de tiempo de ejecución.

## 5.6 MANTENIBILIDAD

La mantenibilidad se centra en el cambio que va asociado a la corrección de errores, a las adaptaciones requeridas a medida que evoluciona el software y a cambios debido a las mejoras producidas por los requerimientos cambiantes del cliente.

El estándar IEEE 982, 1-1998 [IEEE 1994], sugiere un índice de madurez del software (IMS) que proporciona una indicación de la estabilidad de un producto software. Se determina la siguiente información:

$M_t$  = número de módulos en la versión actual.

$F_c$  = número de módulos en la versión actual que se han cambiado.

$F_a$  = número de módulos en la versión actual que se han añadido.

$F_d$  = número de módulos en la versión anterior que se han borrado en la versión actual.

El índice de madurez del software se calcula de la siguiente manera:

$$IMS = [M_t - (F_c + F_a + F_d)] / M_t$$



Si el IMS se aproxima a 1.0 el producto se empieza estabilizar.

$$\text{IMS} = [10 - (4 + 1 + 0)] / 6 = 0.83$$

Por lo tanto el sistema es maduro en un 83% y estable.

## 5.7 ANALISIS COSTO-BENEFICIO DEL SISTEMA

En este punto se desarrolla el cálculo del costo que implica el Sistema De Registro y Seguimiento De Correspondencia y Asignación De Aulas, en el cual se toma los siguientes aspectos:

- ❖ Costo de análisis de programación
- ❖ Costos de capacitación del usuario
- ❖ Costos del software de desarrollo y hardware
- ❖ Costo de instalación

### 5.7.1 Costo de análisis de programación

Para desarrollar esta parte utilizaremos el modelo con el nombre COCOMO, en la cual se tiene la siguiente ecuación:

$$E = a_b \text{KLDC}^{b_b}$$

Donde:

E es el esfuerzo aplicado en personas – mes.

KLDC es el número estimado de líneas de código distribuidas para el proyecto.

El coeficiente  $a_b$  y el exponente  $b_b$  se observan a continuación:

Proyecto de Software	$a_b$	$b_b$	$c_b$	$d_b$
Orgánico	2,40	1,05	2,50	0,38
Semiacoplado	3,00	1,12	2,50	0,35
Empotrado	3,60	1,20	2,50	0,32

Reemplazando los datos en la ecuación se tiene:

$$E = 3.0(KLDC)^{1.12}$$

Las líneas de código en su totalidad se estima 2000 del cual el 30% es código reutilizable por lo que el total de LDC (líneas de código) es:

$$LDC = (\text{total LDC} - \text{LDC Reutilizable})/1000$$

$$LDC = 1.4$$

Reemplazando en la ecuación de determinación de esfuerzo se tiene:

$$E = 3.0 (1.4)^{1.12}$$

$$E = 4,37 \text{ personas} - \text{mes}$$

La estimación de duración esta dado por:  $D = c_b E^{d_b}$

Utilizando la ecuación:

$$D = 2.5 (4.37)^{0.35}$$

$$D = 4,20 \text{ meses}$$

El costo de análisis y programación es de 500 \$us por analista – programador, dando un total de costo de 9177 \$us.

### 5.7.2 Costos de capacitación del usuario

La capacitación del usuario se realizará durante un tiempo estimado de 2 horas diarias con un costo de 5\$us en un tiempo de 5 días hábiles, lo que resulta un costo total de 25 \$us.

### **5.7.3 costo de software de desarrollo y hardware**

La institución no se preocupa por las licencias de software , ya que las mismas son de distribución libre que son necesarias para desarrollar el proyecto, además de contar con los equipos necesarios para el funcionamiento de los sistemas.

### **5.7.4 costo de instalación**

La instalación de los sistemas se realizará conjuntamente con el personal del CEMSE-CERPI, motivo por el cual no existe ningún costo.



## 6. CONCLUSIONES Y RECOMENDACIONES

En este capítulo se brindan las conclusiones del desarrollo del sistema así como las recomendaciones para su mejor uso y aplicación.

### 6.1 CONCLUSIONES

Después de la culminación del análisis, diseño, implementación y pruebas, El Sistema de Registro y Seguimiento de Correspondencia y Asignación de Aulas para el CEMSE - CERPI cumple con los objetivos descritos en el capítulo 1.

Se utilizó la metodología OMT, apoyada por UML la que permite al desarrollador conocer el proceso en forma lógica. Se emplea una base de datos relacional y las interfaces implementadas ayudan a que el sistema sea más funcional para los usuarios en el momento de la introducción de datos y la recuperación de los mismos.

Al aplicar la calidad del Software al Sistema de Registro y Seguimiento de Correspondencia y Asignación de Aulas para el CEMSE-CERPI, se obtuvieron resultados aceptables proporcionándonos un sistema de información eficiente y confiable.

Se obtiene un sistema cumpliendo con los requerimientos de la institución, el cual cuenta con los siguientes módulos: Registro de Usuarios, Registro de Correspondencia, Derivación, Agenda y Aula.

Después de realizar la capacitación a los distintos usuarios del sistema y entregar el manual de usuario se cumplió con los objetivos descritos en el Capítulo 1 como ser:

- Desarrollar el subsistema de Registro de Correspondencia, que mejore la gestión de correspondencia.

- Desarrollar el subsistema de seguimiento de la correspondencia, que proporcione información del estado actual de la correspondencia recibida.
- Generar informes y reportes de la correspondencia por áreas.
- Desarrollar un sistema que proporcione información al usuario mediante consultas, que tendrá una interfaz amigable.
- Desarrollar el subsistema de solicitud de aula que proporcione información sobre la disponibilidad de las aulas con las que cuenta la institución.

Se redujo tiempo en la búsqueda de información:

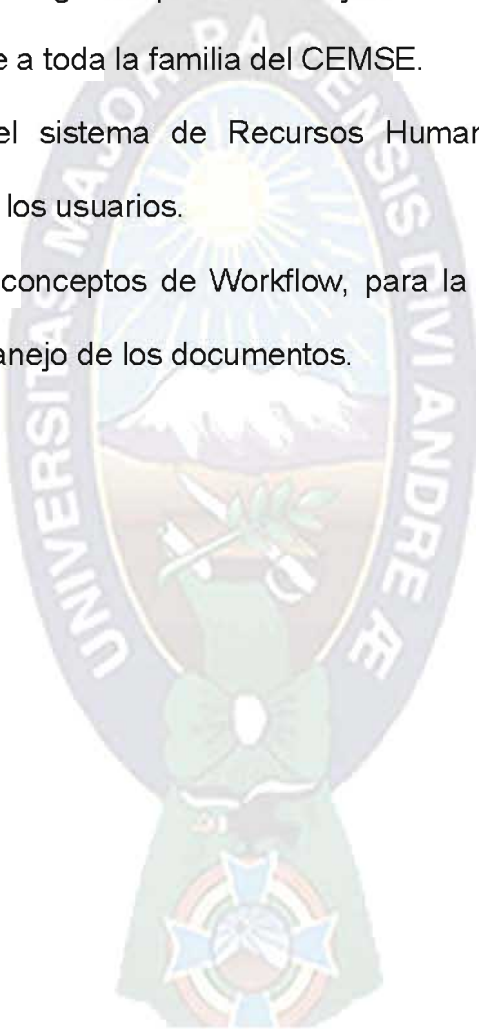
<b>Manejo manual de la Correspondencia</b>	<b>Datos de control aplicando el sistema</b>
La búsqueda manual de los documentos actuales toma más de 10 minutos dependiendo el caso.	La búsqueda toma aproximadamente 10 segundos
Es difícil , en casos imposible encontrar documentación respecto a correspondencia de gestiones pasadas.	Toda la información referente a los documentos de correspondencia que fueron ingresados está siempre disponible.
No se tiene un registro de entrada/salida de correspondencia	Se tiene información precisa de la correspondencia Recibida/Enviada por fecha y hora de ingreso/salida.

El sistema construido es para los usuarios una herramienta útil que proporciona información oportuna a todos los técnicos del CEMSE-CERPI a la hora de toma de desiciones.

## 6.2 RECOMENDACIONES

En cada institución siempre existen nuevos requerimientos dependiendo de las diversas necesidades que se tenga en el momento. Para futuros trabajos se recomienda que se tome en cuenta los siguientes puntos:

- ❖ Implementar un modulo de Digitalización de la correspondencia.
- ❖ Implementar firmas digitales para el manejo de correspondencia.
- ❖ Ampliar el alcance a toda la familia del CEMSE.
- ❖ Interactuar con el sistema de Recursos Humanos para la mejor administración de los usuarios.
- ❖ La aplicación de conceptos de Workflow, para la estandarización de procesos en el manejo de los documentos.



## **BIBLIOGRAFIA**

- [RUMB, 96] Rumbaugh J.: *Object-Oriented Modeling and Design*. ed al. Prentice-Hall, 1996.
- [PRES, 02] Pressman Roger S. : *Ingeniería del Software un enfoque practico*. Quinta Edicion. Mc graw Hill, 2002.
- [KEND,97] Kendall & Kendall, “Análisis y Diseño de Sistemas”, Prentice-Hall, 1997.
- [YOUR,93] Yourdon Edward, “Análisis estructurado moderno, Prentice-Hall, 1993.
- [OMTR,ICR] [http://pisuerga.inf.ubu.es/icruzado/tfc/OMT\\_res.pdf](http://pisuerga.inf.ubu.es/icruzado/tfc/OMT_res.pdf)
- [XFG,UML] Xavier Ferré Grau, Maria Isabel Sanchez Segura “Desarrollo Orientado a Objetos con UML”.
- [LARM, 99] Larman C.: *UML y Patrones..* Prentice Hall, 1999.
- [GEST,05] Luis Oporto Ordoñez, *Gestión Documental y Organización de Archivos Administrativos*. 2005
- [MANU,04] Cárdenas Mario, *Manual de organización de documentos archivistas*. 2004
- [GCCP,04] Gutiérrez Carmen, Proyecto de grado, “Sistema de Control y Seguimiento de Correspondencia SEDUCA , 2004.



- [ARJP,04]** Arancibia Javier, Proyecto de grado, "Sistema de Registro de Correspondencia" Compañía Boliviana de Energía Eléctrica S.A., 2004.
- [ACBP,05]** Aguilar Brígida, Proyecto de Grado, El "Sistema de Seguimiento y Control de Correspondencia para la Prefectura del Departamento de La Paz",2005 .
- [ERJP,07]** Encinas Jheymmy, Proyecto de Grado, "Sistema de Seguimiento y Control de la Documentación para la Escuela Naval Militar",2007.
- [TCMP,05]** Ticonipa Maria, Proyecto de Grado, "Sistema automatizado de Registro y Seguimiento para la Correspondencia y/o Procesos Judiciales Departamento de Asesoría Jurídica UMSA",2005.



# ANEXO A

## ESTRUCTURA DE LA BASE DE DATOS

### “DBSIRCAA”

- ❖ El nombre de la Base de Datos es “dbsircaa”, y esta compuesto por las siguientes tablas, que se encargan de almacenar la información necesaria para que el sistema funcione.
- ❖ Además se utiliza una base de datos auxiliar para generar el calendario de asignación de aulas.

#### dbsircaa\_agenda

Campo	Tipo	Collation	Atributos	Nulo	Predeterminado	Extra	Acción
<input checked="" type="checkbox"/> id_agenda	int(10)			No	0		
<input type="checkbox"/> id_usuario	int(10)			Sí	NULL		
<input type="checkbox"/> descripcion	text	latin1_swedish_ci		Sí	NULL		
<input type="checkbox"/> fecha_agenda	datetime			Sí	NULL		
<input type="checkbox"/> hora_agenda	datetime			Sí	NULL		
<input type="checkbox"/> tipo_obs	varchar(50)	latin1_swedish_ci		Sí	NULL		























































#### dbsircaa\_aula

Campo	Tipo	Collation	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/> id_aula	int(11)			No		auto_increment	
<input type="checkbox"/> nombre_aula	varchar(100)	latin1_swedish_ci		Sí	NULL		
<input type="checkbox"/> capacidad	int(11)			Sí	NULL		
<input type="checkbox"/> descripcion	text	latin1_swedish_ci		Sí	NULL		
<input type="checkbox"/> encargado	int(11)			Sí	NULL		

























#### dbsircaa\_carta

Campo	Tipo	Collation	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/> id_carta	int(11)			No		auto_increment	
<input type="checkbox"/> num_corr	int(11)			No	0		
<input type="checkbox"/> contenido	text	latin1_swedish_ci		No			





















































































## dbsircaa\_derivacion

Campo	Tipo	Collation	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/> id_der	int(11)			No		auto_increment	     
<input type="checkbox"/> id_usuario	int(11)			No	0		     
<input type="checkbox"/> num_corr	int(10)			Si	NULL		     
<input type="checkbox"/> nohoja_adj	int(11)			Si	NULL		     
<input type="checkbox"/> fecha_der	datetime			Si	NULL		     
<input type="checkbox"/> hora_der	datetime			Si	NULL		     
<input type="checkbox"/> prox_destino	varchar(100)	latin1_swedish_ci		Si	NULL		     
<input type="checkbox"/> estado_der	varchar(100)	latin1_swedish_ci		Si	NULL		     
<input type="checkbox"/> observacion	varchar(200)	latin1_swedish_ci		Si	NULL		     





































## dbsircaa\_entidad

Campo	Tipo	Collation	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/> id_entidad	int(11)			No	0		     
<input type="checkbox"/> nombre_razon	varchar(150)	latin1_swedish_ci		Si	NULL		     
<input type="checkbox"/> direccion	varchar(150)	latin1_swedish_ci		Si	NULL		     
<input type="checkbox"/> fono	varchar(50)	latin1_swedish_ci		Si	NULL		     













## dbsircaa\_registro

Campo	Tipo	Collation	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/> id_corr	int(11)			No	0		     
<input type="checkbox"/> num_corr	int(11)			No		auto_increment	     
<input type="checkbox"/> no_cite	int(11)			Si	NULL		     
<input type="checkbox"/> remitente	varchar(50)	latin1_swedish_ci		Si	NULL		     
<input type="checkbox"/> fecha	date			Si	NULL		     
<input type="checkbox"/> referencia	varchar(50)	latin1_swedish_ci		Si	NULL		     
<input type="checkbox"/> rec_env	varchar(30)	latin1_swedish_ci		Si	NULL		     
<input type="checkbox"/> int_ext	varchar(30)	latin1_swedish_ci		Si	NULL		     
<input type="checkbox"/> id_tipo	varchar(50)	latin1_swedish_ci		Si	NULL		     
<input type="checkbox"/> adjunto	varchar(50)	latin1_swedish_ci		Si	NULL		     
<input type="checkbox"/> archivo	text	latin1_swedish_ci		Si	NULL		     
<input type="checkbox"/> id_usuario	int(11)			Si	NULL		     
<input type="checkbox"/> id_entidad	int(11)			Si	NULL		     
<input type="checkbox"/> hora	time			Si	NULL		     
















































### dbsircaa\_solicita

Campo	Tipo	Collation	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/> id_corr	int(11)			Sí	NULL		     
<input type="checkbox"/> id_aula	int(11)			Sí	NULL		     
<input type="checkbox"/> fecha_sol	datetime			Sí	NULL		     
<input type="checkbox"/> hora_sol	datetime			Sí	NULL		     
<input type="checkbox"/> nom_resp	varchar(100)	latin1_swedish_ci		Sí	NULL		     
<input type="checkbox"/> ap_resp	varchar(100)	latin1_swedish_ci		Sí	NULL		     

### dbsircaa\_tipo

Campo	Tipo	Collation	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/> id_tipo	int(11)			No	0		     
<input type="checkbox"/> descripcion	varchar(30)	latin1_swedish_ci		Sí	NULL		     

### dbsircaa\_usuario

Campo	Tipo	Collation	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/> id_usuario	int(11)			No		auto_increment	     
<input type="checkbox"/> nombre	char(100)	latin1_swedish_ci		Sí	NULL		     
<input type="checkbox"/> ap_pat	char(100)	latin1_swedish_ci		Sí	NULL		     
<input type="checkbox"/> ap_mat	char(100)	latin1_swedish_ci		Sí	NULL		     
<input type="checkbox"/> nivel	int(11)			Sí	NULL		     
<input type="checkbox"/> cargo	char(50)	latin1_swedish_ci		Sí	NULL		     
<input type="checkbox"/> login	char(100)	latin1_swedish_ci		Sí	NULL		     
<input type="checkbox"/> password	char(100)	latin1_swedish_ci		Sí	NULL		     

# ANEXO B

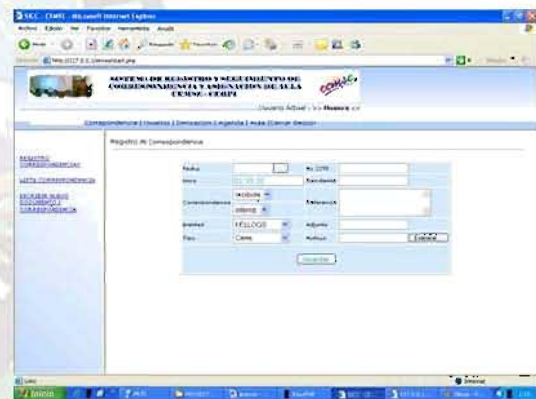
## PANTALLAS PRINCIPALES DEL SISTEMA DE REGISTRO Y SEGUIMIENTO DE CORRESPONDENCIA Y ASIGNACION DE AULAS CEMSE-CERPI

Login:   
Password:

**Autenticación de Usuario:**  
Nos permite el ingreso de los usuarios registrados y autorizados.



**Menú Principal:**  
Nos muestra en la parte superior el usuario actual que esta utilizando el sistema.



**Registro de Correspondencia:**  
Nos permite registrar los datos de una correspondencia Recibida/Enviada, interna/externa.

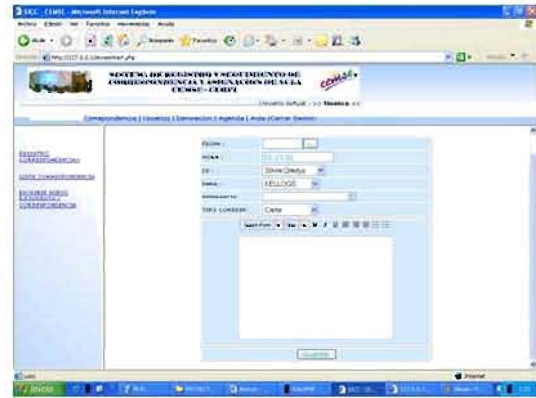
Id	Fecha	Asunto	Destinatario	Remitente	Estado	Tipo	Asignado a	Asignado a	Asignado a	Asignado a
1	2017-08-08	INFORMACIÓN	Administración	Administración	Activo	Interna	Administración	Administración	Administración	Administración
2	2017-08-08	INFORMACIÓN	Administración	Administración	Activo	Interna	Administración	Administración	Administración	Administración
3	2017-08-08	INFORMACIÓN	Administración	Administración	Activo	Interna	Administración	Administración	Administración	Administración
4	2017-08-08	INFORMACIÓN	Administración	Administración	Activo	Interna	Administración	Administración	Administración	Administración
5	2017-08-08	INFORMACIÓN	Administración	Administración	Activo	Interna	Administración	Administración	Administración	Administración
6	2017-08-08	INFORMACIÓN	Administración	Administración	Activo	Interna	Administración	Administración	Administración	Administración
7	2017-08-08	INFORMACIÓN	Administración	Administración	Activo	Interna	Administración	Administración	Administración	Administración
8	2017-08-08	INFORMACIÓN	Administración	Administración	Activo	Interna	Administración	Administración	Administración	Administración
9	2017-08-08	INFORMACIÓN	Administración	Administración	Activo	Interna	Administración	Administración	Administración	Administración
10	2017-08-08	INFORMACIÓN	Administración	Administración	Activo	Interna	Administración	Administración	Administración	Administración
11	2017-08-08	INFORMACIÓN	Administración	Administración	Activo	Interna	Administración	Administración	Administración	Administración
12	2017-08-08	INFORMACIÓN	Administración	Administración	Activo	Interna	Administración	Administración	Administración	Administración
13	2017-08-08	INFORMACIÓN	Administración	Administración	Activo	Interna	Administración	Administración	Administración	Administración
14	2017-08-08	INFORMACIÓN	Administración	Administración	Activo	Interna	Administración	Administración	Administración	Administración
15	2017-08-08	INFORMACIÓN	Administración	Administración	Activo	Interna	Administración	Administración	Administración	Administración

**Listado de Correspondencia:**  
Nos permite ver toda la correspondencia registrada. También nos permite Derivar, editar y borrar la correspondencia.



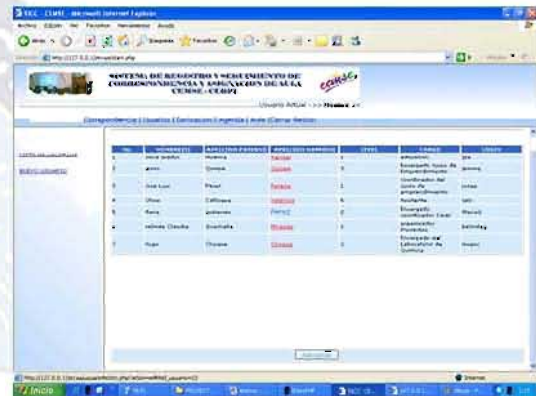
### **Escribir Correspondencia:**

Nos permite escribir una carta o documento pequeño.



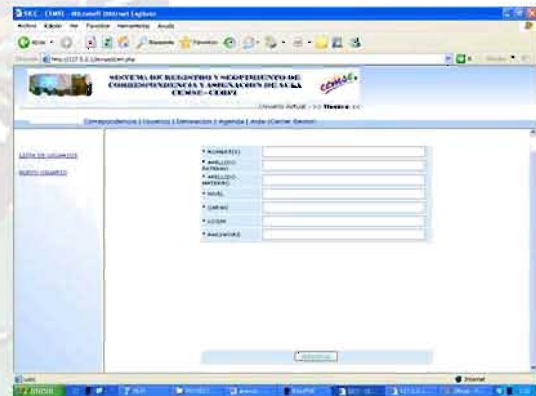
### **Listado de Usuarios:**

Nos permite listar todos los usuarios permitidos por el sistema.



### **Registro de nuevo Usuario:**

Nos permite registrar a nuevos usuarios.



### **Registro de actividad en agenda:**

Nos permite registrar las actividades de los usuarios.

