

**Universidad Autónoma Metropolitana  
Unidad Azcapotzalco**

**División de Ciencias Básicas e Ingeniería**

**Licenciatura en Ingeniería en Computación**

**Proyecto terminal II:**

**“Sistema de Gestión Digital de Documentos”**

**Alumno:**

**Nombre:** Juan Asdrúbal Pastrana Cedeño  
**Matricula:** 205204407

**Trimestre:** 12-P

**Asesora:**

**Nombre:** M. en C. Rafaela Blanca Silva López  
**Titular B**  
**Departamento:** Sistemas

**Fecha de entrega:** 28/Agosto/2012

## Contenido

|  |    |
|--|----|
| Objetivos Generales.....                             | 6  |
| Objetivos específicos: .....                         | 6  |
| Introducción.....                                    | 6  |
| Justificación.....                                   | 6  |
| Antecedentes.....                                    | 8  |
| Descripción técnica .....                            | 9  |
| Especificación técnica .....                         | 12 |
| Diseño lógico del sistema .....                      | 13 |
| Diagrama Entidad - Relación.....                     | 13 |
| Esquema Físico de la Base de datos .....             | 14 |
| Diccionario de Datos.....                            | 17 |
| Tabla area .....                                     | 17 |
| Tabla cargo.....                                     | 17 |
| Tabla puesto .....                                   | 17 |
| Tabla responsable.....                               | 18 |
| Tabla tipo_usuario .....                             | 19 |
| Tabla tipo_documento .....                           | 19 |
| Tabla usuario.....                                   | 20 |
| Tabla recibidos.....                                 | 21 |
| Tabla elaborados.....                                | 22 |
| Tabla profesor.....                                  | 23 |
| Tabla administrativo .....                           | 24 |
| Diagrama de Clases.....                              | 25 |
| Diagrama de clases modulo validar de usuarios.....   | 26 |
| Diagrama de clases modulo documentos recibidos ..... | 27 |
| .....  | 27 |
| Diagrama de clases modulo documentos elaborado ..... | 27 |
| .....  | 27 |
| Diagrama de clases modulo documentos recibidos ..... | 28 |
| Modelo de Casos de uso .....                         | 29 |

|  |    |
|--|----|
| Diagrama general de casos de uso .....                           | 29 |
| Diagrama de caso de uso validar usuario .....                    | 29 |
| Diagrama de caso de uso estructura organizacional .....          | 30 |
| Diagrama de caso de uso documentos Elaborados .....              | 30 |
| Diagrama de caso de uso Gestión de documentos Recibidos .....    | 31 |
| Diagrama de Actividad Gestión Organizacional.....                | 35 |
| Diagrama de Actividad Documentos Recibidos .....                 | 36 |
| Diagrama de Actividad Documentos Elaborados .....                | 37 |
| Documentación de casos de uso .....                              | 38 |
| Caso de uso Validar Usuario .....                                | 38 |
| Caso de uso Documentos recibidos .....                           | 39 |
| Caso de uso Visualizar Documentos Recibidos.....                 | 43 |
| Caso de uso documentos elaborados.....                           | 45 |
| Caso de uso Gestion Organizacional.....                          | 50 |
| Agregar Areas .....  | 50 |
| Agregar Cargos.....  | 51 |
| Nuevo Responsable de la Gestión Organizacional (Directorio)..... | 52 |
| Administración de Responsables.....                              | 54 |
| Modificar información del remitente .....                        | 55 |
| Caso de uso salir del sistema .....                              | 57 |
| Código fuente .....  | 58 |
| mx.uam.azc.cbi.sistemas.gestion.usuarios.actions .....           | 58 |
| LoginAction .....  | 58 |
| mx.uam.azc.cbi.sistemas.gestion.comunicacion.actions .....       | 59 |
| PrincipalAction .....  | 59 |
| ResponsableAction .....  | 59 |
| ElaboradosAction.....  | 64 |
| ElaboradosAction.....  | 64 |
| RecibidosAction .....  | 70 |
| FileUploadAction.....  | 74 |
| mx.uam.azc.cbi.sistemas.gestion.comunicacion.bussines .....      | 77 |
| AdministradorArea .....  | 77 |

|  |     |
|--|-----|
| AdministradorCargo.....  | 78  |
| AdministradorElaborados.....                                     | 78  |
| AdministradorRecibidos.....                                      | 78  |
| AdministradorResponsables.....                                   | 79  |
| mx.uam.azc.cbi.sistemas.gestion.comunicacion.bussines.local..... | 79  |
| AdministradorAreaImpl.....                                       | 79  |
| AdministradorCargoImpl.....                                      | 80  |
| AdministradorElaboradosImpl.....                                 | 81  |
| AdministradorRecibidosImpl.....                                  | 82  |
| AdministradorResponsablesImpl.....                               | 84  |
| mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data.....           | 85  |
| AreaDTO.....   | 85  |
| CargoDTO.....  | 86  |
| ElaboradosDTO.....   | 86  |
| ElaboradosJoinDTO.....   | 89  |
| RecibidosDTO.....  | 91  |
| RecibidosJoinDTO.....  | 93  |
| ResponsableDTO.....  | 95  |
| ResponsableJoinDTO.....  | 97  |
| mx.uam.azc.cbi.sistemas.gestion.comunicacion.interceptors.....   | 100 |
| UsuariosInterceptor.....   | 100 |
| mx.uam.azc.cbi.sistemas.gestion.usuarios.bussines.....           | 100 |
| AdministradorUsuarios.....                                       | 100 |
| mx.uam.azc.cbi.sistemas.gestion.usuarios.bussines.local.....     | 101 |
| AdministradorUsuariosImpl.....                                   | 101 |
| mx.uam.azc.cbi.sistemas.gestion.usuarios.Data.....               | 102 |
| AdministrativoDTO.....   | 102 |
| ProfesorDTO.....   | 103 |
| PuestoDTO.....   | 105 |
| UsuarioDTO.....  | 105 |
| Struts.xml.....  | 108 |
| base.xml.....  | 110 |



|   |     |
|---|-----|
| spring-dev.xml .....  | 110 |
| spring-services.xml .....   | 111 |
| querys.hbm.xml .....  | 112 |
| Web.xml.....  | 113 |
| Manual del usuario .....  | 115 |
| Ingresar al sistema .....   | 115 |
| Generar u nuevo documento recibido .....                              | 116 |
| Visualizar Documentos Recibidos .....                                 | 119 |
| Caso de uso documentos elaborados .....                               | 122 |
| Gestion Organizacional (Directorio) .....                             | 127 |
| Agregar Areas .....   | 127 |
| Agregar Cargos.....   | 128 |
| Nuevo Responsable de la Gestión Organizacional (Directorio).....      | 128 |
| Salir del sistema .....   | 130 |
| Manual de Administrador del sistema.....                              | 131 |
| Administración de Responsables .....                                  | 131 |
| Modificar información del remitente .....                             | 132 |
| Salir del sistema .....   | 133 |
| Manual de instalación.....  | 134 |
| Prerequisitos .....   | 134 |
| Instalación del sistema del sistema de gestión web Comunicación ..... | 134 |
| Bibliografía .....  | 135 |

## Objetivos Generales

Desarrollar un sistema de información web que permita gestionar de manera electrónica los documentos del personal del departamento de sistemas.

## Objetivos específicos:

- Diseñar una base de datos relacional para almacenar la información necesaria para el “Sistema de Gestión Digital de Documentos”.
- Diseñar e implementar un módulo de autenticación de usuarios del sistema (administradores y usuarios finales).
- Diseñar e implementar un módulo que permita clasificar y publicar los documentos del personal del departamento de sistemas, en el sistema
- Diseñar e implementar un módulo que permita realizar búsquedas de los documentos publicados del personal del departamento de sistemas, utilizando metadatos como patrones de búsqueda.
- Diseñar e implementar un módulo que permita consultar y modificar información de los documentos publicados.
- Diseñar e implementar un módulo que permita registrar y consultar la solicitud de años sabáticos y días económicos.

## Introducción

El objetivo de este proyecto es diseñar e implementar una aplicación web que permita gestionar los documentos digitales para el Departamento de sistemas de la Universidad Autónoma Metropolitana Unidad Azcapotzalco además de registrar eventos como el año sabático y días económicos.

## Justificación

La documentación que se maneja dentro del departamento de sistemas actualmente se almacena en carpetas y se manipula manualmente, esto implica contar con espacio necesario para almacenar al menos la documentación de los últimos 5 años.

El sistema que se desarrollará como parte de este proyecto contempla brindar servicio a través de internet, lo que permitirá tener acceso a la información desde cualquier lugar y en cualquier momento siempre y cuando tenga los privilegios necesarios.

La razón por la cual un Ingeniero en Computación puede realizar este proyecto es porque se requieren conocimientos sobre diseño de algoritmos, diseño y manipulación de bases de datos, diseño de aplicaciones web dinámicas, métodos y procesos de Ingeniería de Software y condiciones para la calidad de un producto de software. El cursar el plan de estudios de la Licenciatura en Ingeniería en Computación implica la adquisición de los conocimientos anteriormente mencionados.

Existen productos enfocados a gestión de documentos digitales, entre los productos de software mencionados en los antecedentes la principal desventaja es el costo por el pago de licencias comerciales, en la Universidad existe el proyecto “Implementación de Gestor de Documentos”, la diferencia funcional con el “Sistema de Gestión de Documentos Digitales” es el manejo de metadatos personalizable, los privilegios de usuario, la forma en la que se clasifican los documentos y las búsquedas en base a metadatos. Tanto los productos comerciales como el interno no tienen la funcionalidad de registrar eventos como los días económicos y permisos de año sabático. Tecnológicamente la diferencia principal es el uso de frameworks, Gestor de Base de datos y la plataforma del sistema operativo con licencia libre ya que los anteriores están diseñados para trabajar en Microsoft Windows que requiere el pago de una licencia comercial más el costo del sistema como se muestra en al Tabla 1.

| Sistema ---<br>-----<br>Funciones y IT | Sistema de Gestión de Documentos Digitales   | Document Mangement Software  | Inforouter   | Implementación de Gestor de Documentos   |
|--|--|--|--|--|
| <b>Tipo de Clasificación</b>           | Rubros en BdD  | File System OS   | File System OS   | File System OS   |
| <b>Privilegios de Usuarios</b>         | <ul style="list-style-type: none"> <li>• Administrador.</li> <li>• Usuario.</li> <li>• Publicador.</li> <li>• Asistente</li> </ul> | <ul style="list-style-type: none"> <li>• Administrador.</li> <li>• Usuario.</li> </ul> | <ul style="list-style-type: none"> <li>• Administrador.</li> <li>• Usuario.</li> </ul> | <ul style="list-style-type: none"> <li>• Administrador.</li> <li>• Usuario.</li> </ul> |
| <b>Permite búsquedas</b>               | SI   | SI   | SI   | SI   |
| <b>Tipo de Búsqueda</b>                | Clasificación y Metadatos  | Metadatos y contenido  | Indexado automático por todo el contenido  | Tipo de documento  |
| <b>Plataforma</b>                      | Linux  | Windows Server   | Windows Server   | Windows  |
| <b>Base de Datos</b>                   | PostgreSQL   | SQL Server   | SQL Server   | MySQL  |
| <b>Licencia</b>                        | Creative Commons   | Comercial  | Comercial  | Creative Commons   |
| <b>Lenguaje de Programación</b>        | Java EE: Spring, Struts y Hibernate  | Visual Studio  | Visual Studio  | Java SE: JSP y Servlets  |

## Antecedentes

- Referencias internas.

### Implementación de un Gestor de Documentos [2]

La administración de la información por medio de un sistema web, es capaz de ordenar la información por carpetas y proyectos restringidos a diferentes tipos de usuarios.

Poder administrar la información en un solo punto en el que se pueda acceder desde cualquier lugar, tiene muchas ventajas que se pueden aprovechar para diferentes aplicaciones como proyectos escolares que se realizan en grupo, proyectos de oficina en el que intervienen diferentes usuarios, para compartir fotos y archivos en familia, entre muchas otras utilidades.

Una desventaja del proyecto es que al ser una primera versión no es tan robusto, debido que no se realizaron pruebas de campo, pero a futuro se pueden realizar de tal forma que el producto pueda ser accesible a diferentes usuarios, como una herramienta libre. También deberían realizarse pruebas automáticas que verificaran todos los casos del sistema. Con estas pruebas y verificaciones mejorarlo y que sea de satisfacción para las necesidades del usuario.

A este proyecto se le pueden agregar algunos frameworks como Struts, Spring, Hibérnate, o utilizar algunas técnicas como Ajax para darle algunas animaciones al sistema.

- Referencias Externas

### **InfoRouter.**

InfoRouter[3] es un software de gestión de documentos electrónicos, que incluye gestión de activos digitales, imágenes de documentos, digitalización de documentos, flujo de trabajo, gestión de registros, archivado de documentos, almacenamiento de documentos, formularios electrónicos, control de documentos ISO. El sistema requiere una plataforma Windows Server.

Una desventaja del producto es el costo ya que se tiene que pagar la licencia del sistema, sistema Operativo Windows Server y Manejador de la Base de Datos ya que utiliza toda la tecnología de Microsoft

### **Document Mangement**

El Sistema de Administración de Documentos (Document Mangement Software)[4] karomi ENTERPRISE es una solución integral que permite gestionar el ciclo de vida de los documentos impresos y electrónicos mediante una aplicación Web.

Una desventaja del producto es el costo ya que se tiene que pagar la licencia del sistema, Sistema Operativo Windows Server y Manejador de la Base de Datos ya que utiliza toda la tecnología de Microsoft

## Descripción técnica

El proyecto consiste en un sistema de información que permita publicar documentos y clasificarlos de acuerdo a los siguientes rubros:

- Plazas
- Requisiciones
- Oficios
- Informes
- Planes
- Ordenes de mantenimiento
- Licencias
- Horarios
- Proyectos terminales
- Encuestas

El sistema contará con los siguientes módulos

**Módulo 1.** Diseño e implementación de la autenticación de usuarios del sistema a través de perfiles. El sistema considera los perfiles: administrador, asistente administrativo, publicador y usuario. El administrador tendrá acceso a todos los módulos que integran el sistema. El Publicador podrá publicar documentos, modificar y hacer búsquedas dentro de las clasificaciones a las que tenga acceso. El asistente administrativo adicional al publicador puede gestionar los rubros para clasificar los documentos.

**Módulo 2.** Diseño e implementación de gestión de clasificaciones, donde el administrador podrá hacer registro y actualización de los datos de clasificaciones. En éste módulo el administrador debe definir las clasificaciones principales en las que el departamento organiza los documentos. Debe tener la capacidad de crear sub clasificaciones.

**Ejemplo:**

- Oficios
  - Enviados
    - Jefatura
    - Asistente Administrativo
    - Horarios
    - Otros
  - Recibidos
    - Rectoría general
    - Rectoría unidad
    - División
    - Departamentos

**Módulo 3.** Diseño e implementación del registro y actualización de los datos del usuario. En éste módulo se debe ingresar la información que permita dar de alta al usuario, consultar y actualizar sus datos. Solo el administrador asignara a modificación el perfil del usuario.

**Módulo 4.** Diseño e implementación de publicación de documentos.

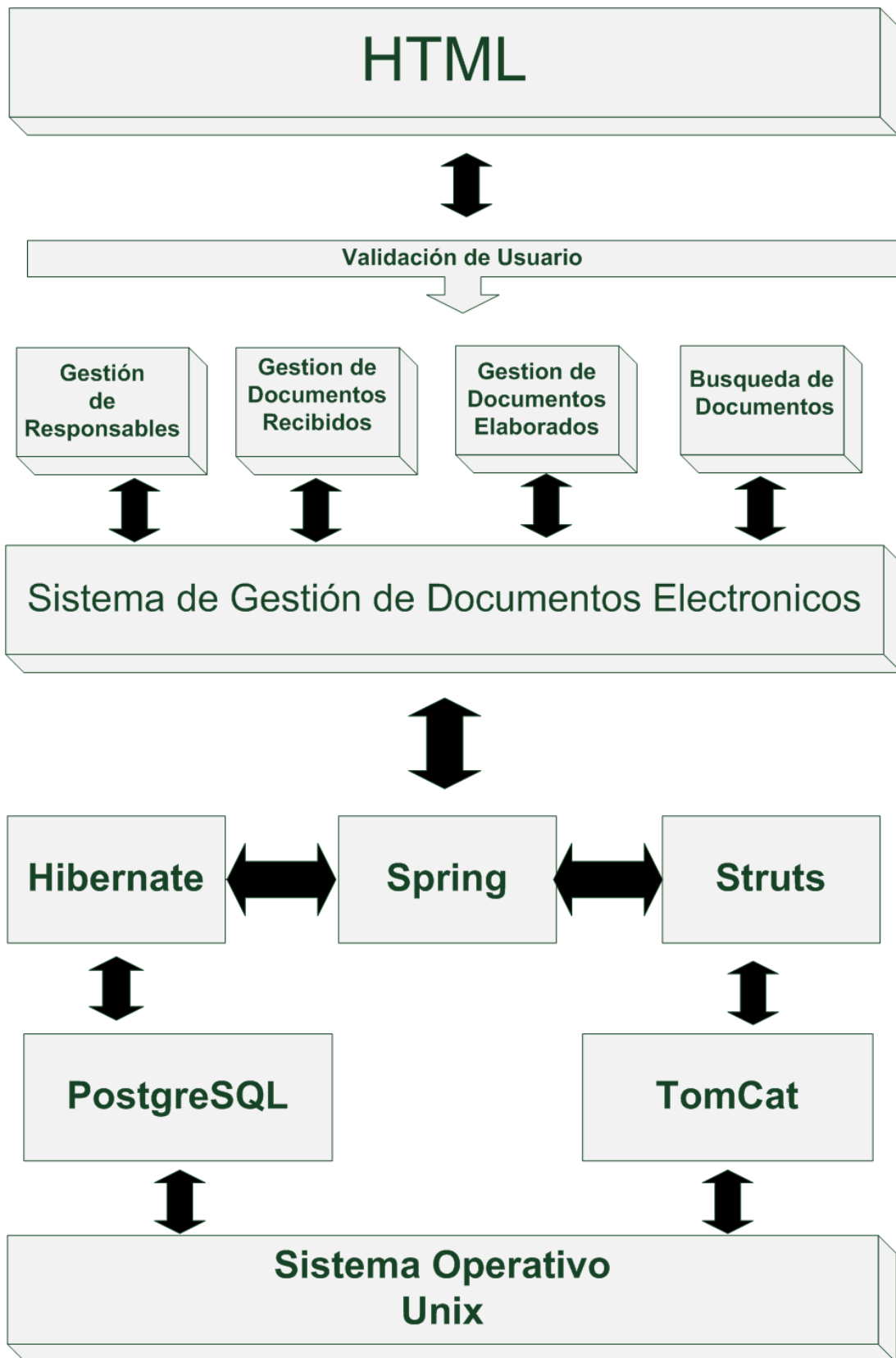
Permite al usuario que tenga privilegios publicar en el sistema un documento nuevo. El documento adjunto incluye metadatos que permitirán realizar la búsqueda del documento, por ejemplo: título, propietario, fecha de publicación, clasificación y sub clasificación.

**Módulo 5.** Diseño e implementación de Modificación de documentos.

Permite al usuario al usuario que tiene privilegios publicar en el sistema una nueva versión del documento, además se permite actualizar información relevante (metadatos) como fecha de publicación, clasificación y sub clasificación.

**Módulo 6.** Diseño e implementación de búsqueda de documentos.

Permite al usuario que tenga privilegios ingresar información relevante (metadatos), que fungirán como filtros: título, propietario, fecha del documento, clasificación, sub clasificación. El módulo debe generar una búsqueda en la base de datos y mostrar los vínculos para acceder a los documentos siempre y cuando el usuario tenga privilegios para acceder a él.



**Figura 2.** Diagrama de bloques

## Especificación técnica

El sistema se desarrollará en una plataforma abierta, utilizando como IDE<sup>2</sup>: Eclipse<sup>3</sup> [5] sobre sistema operativo Linux. Para que el sistema sea multiplataforma se desarrollará con el lenguaje de programación Java. Basado en la arquitectura Modelo Vista Controlador.

El sistema integra servidores de aplicaciones abiertos como Tomcat<sup>4</sup> [6], así como bases de datos relacionales abiertas como Postgres<sup>5</sup> [7].

Para agilizar el desarrollo se utilizarán frameworks<sup>6</sup> como Spring<sup>7</sup> [8], Hibernate<sup>8</sup> [9] y Struts<sup>9</sup> [10]. Se plantea utilizar el estándar J2EE 1.4 como base, con el fin de poder ser montado a futuro sobre diferentes servidores de aplicaciones.

La figura 2 muestra las partes generales que componen el sistema: El usuario se conecta al sistema mediante un navegador web utilizando HTML. Hace uso de los módulos ya mencionados para consultar y generar su información, el sistema realiza las peticiones dadas.

Tras la programación básica, será necesaria la implementación en *struts* para la plataforma web y el servidor Tomcat.

Utilizaremos Spring para la parte de servidores e Hibernate como intermediario en la parte de programación con la base de datos, la cual estará administrada con PostgreSQL, todo sobre un sistema operativo basado en Unix.

### Licencia de Software Libre-Creative Commons.



- Reconocimiento (Attribution): En cualquier explotación de la obra autorizada por la licencia hará falta reconocer la autoría.
- Compartir Igual (Share alike): La explotación autorizada incluye la creación de obras derivadas siempre que mantengan la misma licencia al ser divulgadas.
- Reconocimiento - Compartir-Igual (by-sa): Se permite la distribución de las cuales se debe hacer una licencia igual a la que regula la obra original.

---

<sup>2</sup> **IDE**.- (Integrated Development Environment - Entorno integrado de desarrollo). Aplicación compuesta por un conjunto de herramientas útiles para un programador.

<sup>3</sup> **Eclipse**.- Es una plataforma de desarrollo integrado que puede ser usada para crear diversas aplicaciones como sitios de internet.

<sup>4</sup> **Tomcat**.- Es un servidor web con soporte de servlets y JSPs.

<sup>5</sup> **Postgres**.- Sistema de gestión de base de datos relacional orientada a objetos y libre.

<sup>6</sup> **Framework**.- Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado, típicamente puede incluir soporte de programas, bibliotecas y un lenguaje.

<sup>7</sup> **Spring**.- Es un framework de código abierto de desarrollo de aplicaciones para la plataforma Java.

<sup>8</sup> **Hibernate**.- Es una herramienta de mapeo objeto-relacional para la plataforma Java.

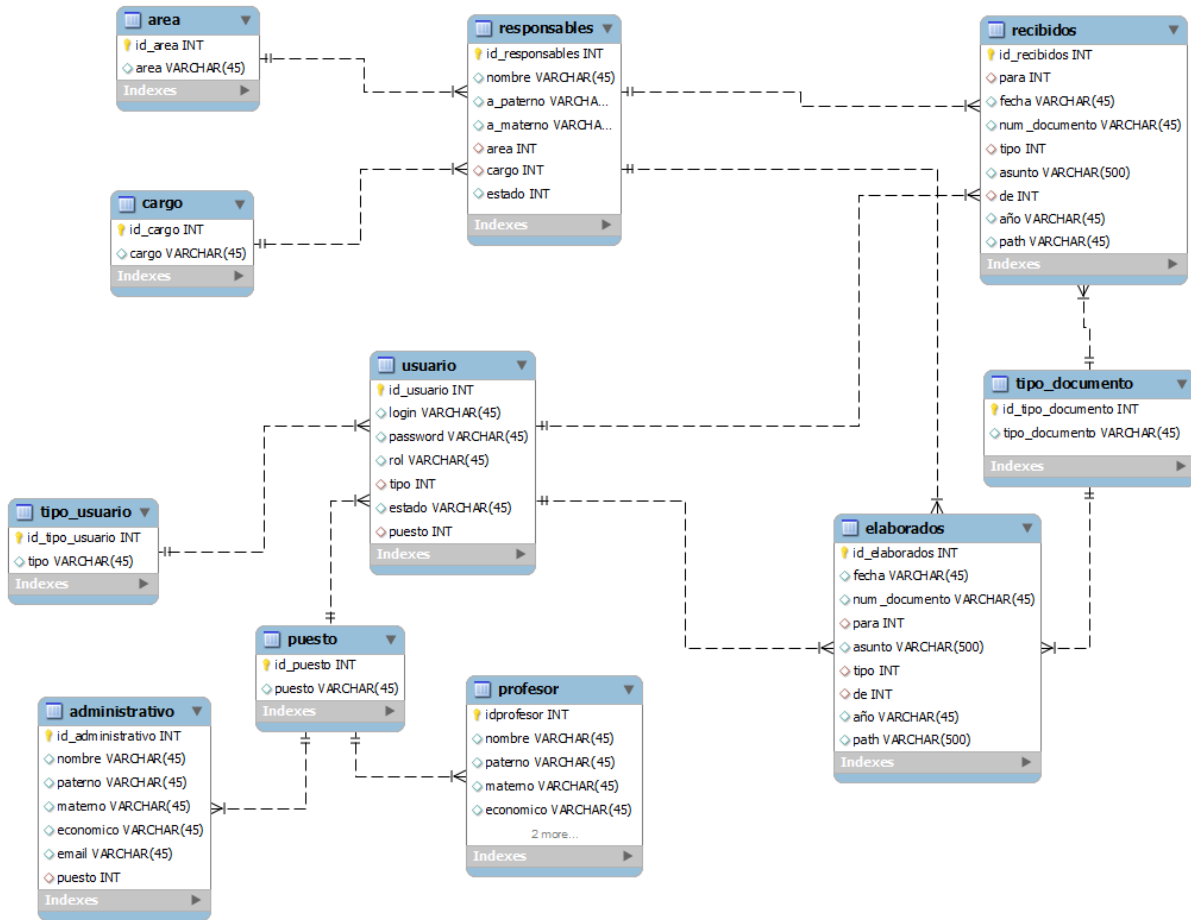
<sup>9</sup> **Struts**.- Es una herramienta de soporte para el desarrollo de aplicaciones Web bajo el patrón MVC bajo la plataforma J2EE.

<sup>10</sup> **E-Learning Knowledge**.- Proyecto creado con la finalidad de crear entornos colaborativos de aprendizaje.



# Diseño lógico del sistema

## Diagrama Entidad - Relación



## Esquema Físico de la Base de datos

```
CREATE TABLE area (
  id_area bigserial NOT NULL ,
  area VARCHAR(45) ,
  PRIMARY KEY (id_area) )
;

CREATE TABLE cargo (
  id_cargo bigserial NOT NULL ,
  cargo VARCHAR(45) ,
  PRIMARY KEY (id_cargo) )
;

CREATE TABLE puesto (
  id_puesto bigserial NOT NULL ,
  puesto VARCHAR(45) ,
  PRIMARY KEY (id_puesto) )
;

CREATE TABLE responsables (
  id_responsables bigserial NOT NULL ,
  nombre VARCHAR(45) ,
  a_paterno VARCHAR(45) ,
  a_materno VARCHAR(45) ,
  email VARCHAR(45) ,
  telefono VARCHAR(45) ,
  ubicacion VARCHAR(45) ,
  area INT ,
  cargo INT ,
  estado INT ,
  PRIMARY KEY (id_responsables) ,
  CONSTRAINT area
    FOREIGN KEY (area )
    REFERENCES area (id_area ) ,
  FOREIGN KEY (cargo )
    REFERENCES cargo (id_cargo ))
;

CREATE TABLE tipo_usuario (
  id_tipo_usuario bigserial NOT NULL ,
  tipo VARCHAR(45) ,
  PRIMARY KEY (id_tipo_usuario) )
;

CREATE TABLE usuario (
  id_usuario INT NOT NULL ,
  login VARCHAR(45) ,
  password VARCHAR(45) ,
  rol VARCHAR(45) ,
```

```

tipo INT ,
nombre VARCHAR(45) ,
paterno VARCHAR(45) ,
materno VARCHAR(45) ,
numero_economico VARCHAR(45) ,
estado VARCHAR(45) ,
puesto INT,
PRIMARY KEY (id_usuario) ,
    FOREIGN KEY (puesto )
    REFERENCES puesto (id_puesto ),
    FOREIGN KEY (tipo )
    REFERENCES tipo_usuario (id_tipo_usuario ))
;

CREATE TABLE tipo_documento (
    id_tipo_documento bigserial NOT NULL ,
    tipo_documento VARCHAR(45) ,
    PRIMARY KEY (id_tipo_documento) )
;

CREATE TABLE recibidos (
    id_recibidos bigserial NOT NULL ,
    para INT ,
    fecha VARCHAR(45) ,
    num_documento VARCHAR(45) ,
    tipo INT ,
    asunto character varying NULL ,
    de INT ,
    anio VARCHAR(45) ,
    directorio character varying ,
    documento character varying ,
    PRIMARY KEY (id_recibidos) ,
    FOREIGN KEY (de )
    REFERENCES responsables (id_responsables ),
    FOREIGN KEY (para )
    REFERENCES usuario (id_usuario ),
    FOREIGN KEY (tipo )
    REFERENCES tipo_documento (id_tipo_documento ))
;

CREATE TABLE elaborados (
    id_elaborados bigserial NOT NULL ,
    fecha VARCHAR(45) ,
    num_documento VARCHAR(45) ,
    para INT ,
    asunto character varying ,
    contenido character varying ,
    tipo INT ,
    de INT ,
    anio VARCHAR(45) ,
    directorio character varying ,
    documento character varying ,
    PRIMARY KEY (id_elaborados) ,

```

```

FOREIGN KEY (de )
REFERENCES usuario (id_usuario ),
FOREIGN KEY (para )
REFERENCES responsables (id_responsables ),
FOREIGN KEY (tipo )
REFERENCES tipo_documento (id_tipo_documento ))
;

```

```

CREATE TABLE profesor (
  id_profesor INT NOT NULL ,
  nombre VARCHAR(45) ,
  paterno VARCHAR(45) ,
  materno VARCHAR(45) ,
  economico VARCHAR(45) ,
  email VARCHAR(45) ,
  puesto INT,
  PRIMARY KEY (id_profesor),
  FOREIGN KEY (puesto )
    REFERENCES puesto (id_puesto ))
;

```

```

CREATE TABLE administrativo (
  id_administrativo INT NOT NULL ,
  nombre VARCHAR(45) ,
  paterno VARCHAR(45) ,
  materno VARCHAR(45) ,
  economico VARCHAR(45) ,
  email VARCHAR(45) ,
  puesto INT,
  PRIMARY KEY (id_administrativo),
  FOREIGN KEY (puesto )
    REFERENCES puesto (id_puesto ))
;

```

## Diccionario de Datos

### Tabla area

| Nombre de la Tabla   | Descripción   |                |             |         | Nombre de Clase |              | Diseño       |                       |           |
|----------------------|---|----------------|-------------|---------|-----------------|--------------|--------------|-----------------------|-----------|
| area                 | Tabla para almacenar información relacionada con las areas. |                |             |         | AreaDTO         |              |              |                       |           |
| <b>Base de Datos</b> |   |                |             |         | <b>Objetos</b>  |              | <b>Forma</b> |                       |           |
| Nombre               | Descripción   | Tipo de Dato   | Restricción | Formato | Atributo        | Tipo de Dato | Etiqueta     | Nombre del componente | Pagina    |
| id_area              | Clave de area   | bigserial      | PK          |         | idarea          | long         | none         | area.area             | Areas.jsp |
| area                 | area  | VARCHAR ( 45 ) |             |         | area            | string       | Area         | Área.area             | Areas.jsp |

### Tabla cargo

| Nombre de la Tabla   | Descripción                                      |                |             |         | Nombre de Clase |              | Diseño       |                       |            |
|----------------------|--|----------------|-------------|---------|-----------------|--------------|--------------|-----------------------|------------|
| cargo                | Tabla para almacenar información relacionada con |                |             |         | CargoDTO        |              |              |                       |            |
| <b>Base de Datos</b> |  |                |             |         | <b>Objetos</b>  |              | <b>Forma</b> |                       |            |
| Nombre               | Descripción                                      | Tipo de Dato   | Restricción | Formato | Atributo        | Tipo de Dato | Etiqueta     | Nombre del componente | Pagina     |
| id_cargo             | Clave del cargo                                  | bigserial      | PK          |         | idcargo         | long         | none         | Cargo.cargo           | Cargos.jsp |
| cargo                | cargo  | VARCHAR ( 45 ) |             |         | cargo           | string       | Cargo        | Cargo.cargo           | Cargos.jsp |

### Tabla puesto

| Nombre de la Tabla   | Descripción   |                |             |         | Nombre de Clase |              | Diseño       |                       |        |
|----------------------|---|----------------|-------------|---------|-----------------|--------------|--------------|-----------------------|--------|
| Puesto               | Tabla para almacenar información relacionada con los puestos. |                |             |         | PuestoDTO       |              | CATALOGO     |                       |        |
| <b>Base de Datos</b> |   |                |             |         | <b>Objetos</b>  |              | <b>Forma</b> |                       |        |
| Nombre               | Descripción   | Tipo de Dato   | Restricción | Formato | Atributo        | Tipo de Dato | Etiqueta     | Nombre del componente | Pagina |
| id_puesto            | Clave del puesto  | bigserial      | PK          |         | idpuesto        | long         | none         |                       |        |
| puesto               | puesto  | VARCHAR ( 45 ) |             |         | puesto          | string       | Puesto       |                       |        |

## Tabla responsable

| Nombre de la Tabla    | Descripción  |              |             |         | Nombre de Clase      |              | Diseño                  |                           |                     |
|-----------------------|--|--------------|-------------|---------|----------------------|--------------|-------------------------|---------------------------|---------------------|
| responsable           | Tabla para almacenar información relacionada con los responsables. |              |             |         | ResponsableDTO       |              | JSP                     |                           |                     |
| Base de Datos         |  |              |             |         | Objetos              |              | Forma                   |                           |                     |
| Nombre                | Descripción  | Tipo de Dato | Restricción | Formato | Atributo             | Tipo de Dato | Etiqueta                | Nombre del componente     | Página              |
| <b>id_responsable</b> | Clave del responsable  | bigserial    | PK          |         | <b>idresponsable</b> | long         | None                    | <b>Resp.idresponsable</b> | AddResponsables.jsp |
| <b>nombre</b>         | Nombre del responsable   | VARCHAR(45)  |             |         | <b>nombre</b>        | String       | <b>nombre</b>           | <b>Resp.nombre</b>        | AddResponsables.jsp |
| <b>a_paterno</b>      | Apellido paterno responsable                                       | VARCHAR(45)  |             |         | <b>apaterno</b>      | String       | <b>Apellido paterno</b> | <b>Resp.apaterno</b>      | AddResponsables.jsp |
| <b>a_materno</b>      | Apellido materno responsable                                       | VARCHAR(45)  |             |         | <b>amaterno</b>      | String       | <b>Apellido materno</b> | <b>Resp.amaterno</b>      | AddResponsables.jsp |
| <b>email</b>          | Email responsable  | VARCHAR(45)  |             |         | <b>email</b>         | String       | <b>email</b>            | <b>Resp.email</b>         | AddResponsables.jsp |
| <b>telefono</b>       | Teléfono responsable   | VARCHAR(45)  |             |         | <b>telefono</b>      | String       | <b>telefono</b>         | <b>Resp.telefono</b>      | AddResponsables.jsp |
| <b>ubicacion</b>      | Ubicación del responsable  | VARCHAR(45)  |             |         | <b>ubicacion</b>     | String       | <b>ubicacion</b>        | <b>Resp.ubicacion</b>     | AddResponsables.jsp |
| <b>area</b>           | Area del responsable   | INT          | NO NULL     |         | <b>area</b>          | String       | <b>area</b>             | <b>Resp.area</b>          | AddResponsables.jsp |
| <b>cargo</b>          | Cargo del responsable  | INT          | NO NULL     |         | <b>cargo</b>         | String       | <b>cargo</b>            | <b>Resp.Cargo</b>         | AddResponsables.jsp |
| <b>estado</b>         | Estado del responsable   | INT          | NO NULL     |         | <b>estado</b>        | String       | <b>Estado</b>           | <b>Resp.estado</b>        | AddResponsables.jsp |

## Tabla tipo\_usuario

| Nombre de la Tabla | Descripción   |              |             |         | Nombre de Clase | Diseño       |          |                       |        |
|--------------------|---|--------------|-------------|---------|-----------------|--------------|----------|-----------------------|--------|
| tipo de usuario    | Tabla para almacenar información relacionada con los tipos de usuarios. |              |             |         | -               | CATALOGO     |          |                       |        |
| Base de Datos      |   |              |             |         | Objetos         |              | Forma    |                       |        |
| Nombre             | Descripción   | Tipo de Dato | Restricción | Formato | Atributo        | Tipo de Dato | Etiqueta | Nombre del componente | Página |
| id_tipo_usuario    | Clave de tipo de usuario  | bigserial    | PK          |         | idtipousuario   | long         | none     |                       |        |
| tipo               | Tipo de usuario   | VARCHAR(45)  |             |         | tipo            | String       | tipo     |                       |        |

## Tabla tipo\_documento

| Nombre de la Tabla | Descripción   | Nombre de Clase    | Diseño   |
|--------------------|---|--------------------|----------|
| tipo de documento  | Tabla para almacenar información relacionada con los tipos de documentos. | TipodedocumentoDTO | CATALOGO |

| Base de Datos     |                    |              |             |         | Objetos         |              | Forma    |                       |        |
|-------------------|--------------------|--------------|-------------|---------|-----------------|--------------|----------|-----------------------|--------|
| Nombre            | Descripción        | Tipo de Dato | Restricción | Formato | Atributo        | Tipo de Dato | Etiqueta | Nombre del componente | Página |
| id_tipo_documento | Clave de documento | bigserial    | PK          |         | idtipodocumento | long         |          |                       |        |
| tipo_documento    | Tipo de documento  | VARCHAR(45)  |             |         | tipodocumento   | String       |          |                       |        |

## Tabla usuario

| Nombre de la Tabla   | Descripción  |              |             |         | Nombre de Clase |              | Diseño           |                       |           |
|----------------------|--|--------------|-------------|---------|-----------------|--------------|------------------|-----------------------|-----------|
| usuario              | Tabla para almacenar información relacionada con los usuarios. |              |             |         | UsuarioDTO      |              | JSP              |                       |           |
| <b>Base de Datos</b> |  |              |             |         | <b>Objetos</b>  |              | <b>Forma</b>     |                       |           |
| Nombre               | Descripción  | Tipo de Dato | Restricción | Formato | Atributo        | Tipo de Dato | Etiqueta         | Nombre del componente | Página    |
| id_usuario           | Clave del usuario  | INT          | PK          |         | idusuario       | INT          | none             | Usr.idusuario         | Login.jsp |
| login                | Login del usuario  | VARCHAR(45)  |             |         | login           | String       | login            | Usr.login             | Login.jsp |
| password             | Password del usuario   | VARCHAR(45)  |             |         | password        | String       | password         | Usr.password          | Login.jsp |
| Rol                  | Rol del usuario  | VARCHAR(45)  |             |         | rol             | String       | rol              | Usr.rol               | Login.jsp |
| tipo                 | Tipo de usuario  | INT          |             |         | tipo            | INT          | tipo             | Usr.tipo              | Login.jsp |
| nombre               | Nombre del usuario   | VARCHAR(45)  |             |         | nombre          | String       | nombre           | Usr.nombre            | Login.jsp |
| paterno              | Apellido paternos del usuario                                  | VARCHAR(45)  |             |         | paterno         | String       | paterno          | Usr.paterno           | Login.jsp |
| materno              | Apellido materno del usuario                                   | VARCHAR(45)  |             |         | materno         | String       | materno          | Usr.materno           | Login.jsp |
| numero_economico     | Numero económico del usuario                                   | VARCHAR(45)  |             |         | numeroeconomico | String       | Numero Economico | Usr.numero_economico  | Login.jsp |
| estado               | Estado del usuario   | VARCHAR(45)  |             |         | estado          | String       | estado           | Usr.estado            | Login.jsp |



## Tabla recibidos

| Nombre de la Tabla   | Descripción   |                   |             |         | Nombre de Clase |              | Diseño        |                       |                  |
|----------------------|---|-------------------|-------------|---------|-----------------|--------------|---------------|-----------------------|------------------|
| recibidos            | Tabla para almacenar información relacionada con los recibidos. |                   |             |         | Recibidos DTO   |              |               |                       |                  |
| <i>Base de Datos</i> |   |                   |             |         | <i>Objetos</i>  |              | <i>Forma</i>  |                       |                  |
| Nombre               | Descripción   | Tipo de Dato      | Restricción | Formato | Atributo        | Tipo de Dato | Etiqueta      | Nombre del componente | Página           |
| id_recibidos         | Clave de recibidos  | bigserial         | PK          |         | idrecibidos     | long         | none          |                       |                  |
| para                 | Para  | INT               | NO NULL     |         | para            | INT          | para          | Rec.para              | AddRecibidos.jsp |
| de                   | de  | INT               | NO NULL     |         | de              | INT          | para          | Rec.de                | AddRecibidos.jsp |
| fecha                | Fecha de recibido   | VARCHAR(45)       |             |         | fecha           | String       | fecha         | Rec.fecha             | AddRecibidos.jsp |
| num_documento        | Numero de documento recibido                                    | VARCHAR(45)       |             |         | numdocumento    | String       | num_documento | Rec.numdocumento      | AddRecibidos.jsp |
| asunto               | Asunto recibido   | INT               |             |         | asunto          | INT          | asunto        | Rec.asunto            | AddRecibidos.jsp |
| anio                 | Anio que se recibio   | VARCHAR(45)       |             |         | anio            | String       | anio          | Rec.anio              | AddRecibidos.jsp |
| directorio           | Directorio  | character varying |             |         | directorio      | String       | directorio    | Rec.directorio        | AddRecibidos.jsp |
| documento            | Documento recibido  | character varying |             |         | documento       | String       | documento     | Rec.documento         | AddRecibidos.jsp |

## Tabla elaborados

| Nombre de la Tabla   | Descripción  |                   |             |         | Nombre de Clase |              | Diseño        |                       |                   |
|----------------------|--|-------------------|-------------|---------|-----------------|--------------|---------------|-----------------------|-------------------|
| elaborados           | Tabla para almacenar información relacionada con los elaborados. |                   |             |         | ElaboradosDTO   |              | JSP           |                       |                   |
| <i>Base de Datos</i> |  |                   |             |         | <i>Objetos</i>  |              | <i>Forma</i>  |                       |                   |
| Nombre               | Descripción  | Tipo de Dato      | Restricción | Formato | Atributo        | Tipo de Dato | Etiqueta      | Nombre del componente | Página            |
| id_elaborados        | Clave de elaborados  | bigserial         | PK          |         | idelaborados    | long         | none          | Elab.idelaborados     | AddElaborados.jsp |
| fecha                | Fecha de elaborados  | VARCHAR(45)       |             |         | fecha           | String       | fecha         | Elab.fecha            | AddElaborados.jsp |
| num_documento        | Numero de documento elaborado                                    | VARCHAR(45)       |             |         | numdocumento    | String       | num_documento | Elab.numdocumento     | AddElaborados.jsp |
| para                 | Para   | INT               |             |         | para            | INT          | para          | Elab.Para             | AddElaborados.jsp |
| asunto               | Asunto   | character varying |             |         | asunto          | String       | asunto        | Elab.asunto           | AddElaborados.jsp |
| contenido            | Contenido de elaborados  | character varying |             |         | contenido       | String       | contenido     | Elab.contenido        | AddElaborados.jsp |
| tipo                 | Tipo de elaborados   | INT               | NO NULL     |         | tipo            | INT          | tipo          | Elab.tipo             | AddElaborados.jsp |
| de                   | De   | INT               | NO NULL     |         | de              | INT          | de            | Elab.de               | AddElaborados.jsp |
| anio                 | Anio elaborado   | VARCHAR(45)       |             |         | anio            | String       | anio          | Elab.anio             | AddElaborados.jsp |
| directorio           | Directorio   | character varying |             |         | directorio      | String       | directorio    | Elab.directorio       | AddElaborados.jsp |
| documento            | documento  | character varying |             |         | documento       | String       | documento     | Elab.documento        | AddElaborados.jsp |

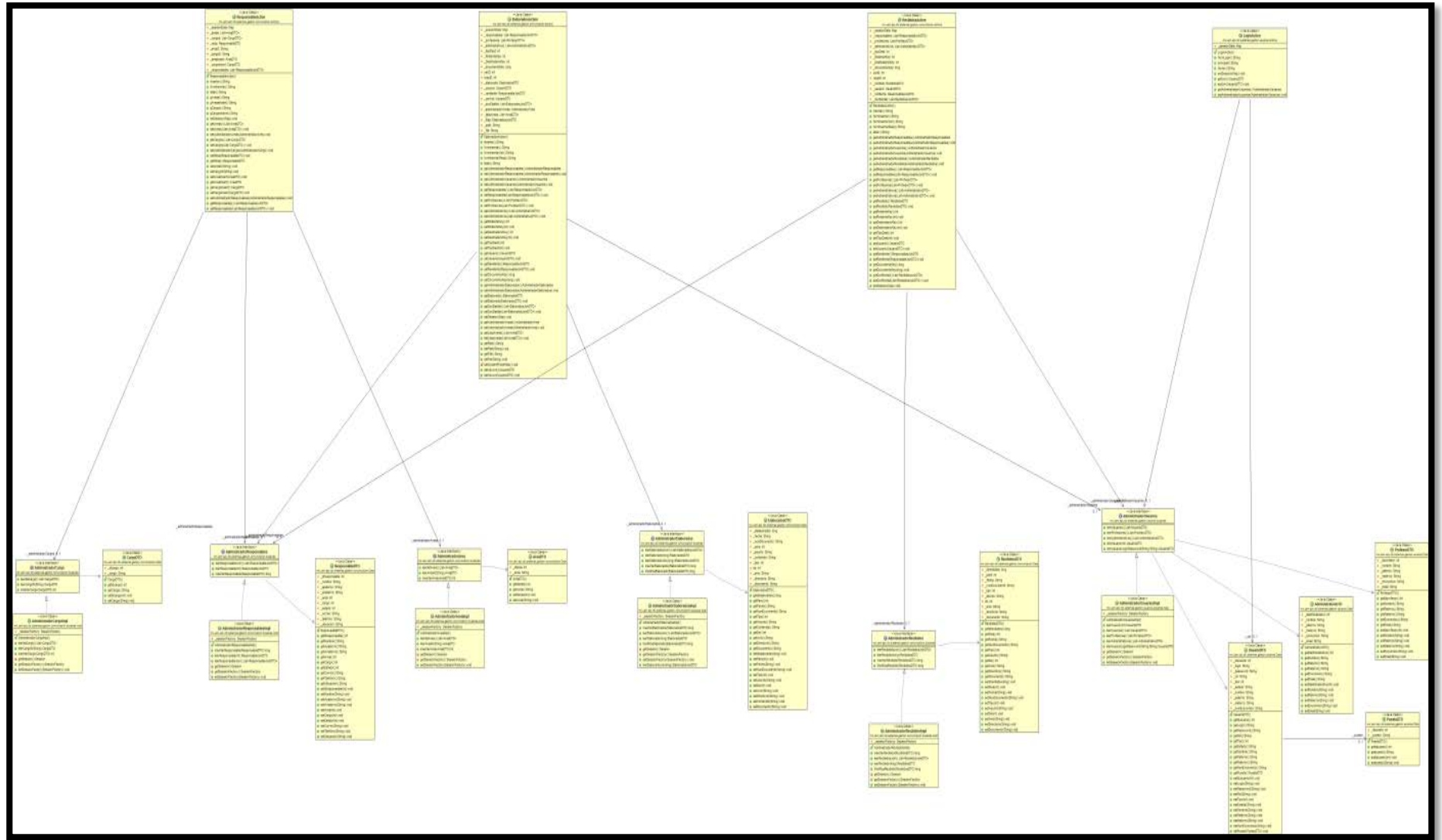
## Tabla profesor

| Nombre de la Tabla   | Descripción  |              |             |         | Nombre de Clase |              | Diseño       |                       |        |
|----------------------|--|--------------|-------------|---------|-----------------|--------------|--------------|-----------------------|--------|
| profesor             | Tabla para almacenar información relacionada con los profesores. |              |             |         | ProfesorDTO     |              | CATALOGO     |                       |        |
| <b>Base de Datos</b> |  |              |             |         | <b>Objetos</b>  |              | <b>Forma</b> |                       |        |
| Nombre               | Descripción  | Tipo de Dato | Restricción | Formato | Atributo        | Tipo de Dato | Etiqueta     | Nombre del componente | Pagina |
| id_profesor          | Clave del profesor   | INT          | PK          |         | idprofesor      | INT          | none         |                       |        |
| nombre               | Nombre del profesor  | VARCHAR(45)  |             |         | nombre          | String       | nombre       |                       |        |
| paterno              | Apellido paterno del profesor                                    | VARCHAR(45)  |             |         | paterno         | String       | paterno      |                       |        |
| materno              | Apellido materno del profesor                                    | VARCHAR(45)  |             |         | materno         | String       | materno      |                       |        |
| economico            | Numero económico del profesor                                    | VARCHAR(45)  |             |         | economico       | String       | economico    |                       |        |
| email                | Email del profesor   | VARCHAR(45)  |             |         | email           | String       | email        |                       |        |
| puesto               | Clave del Puesto   | VARCHAR(45)  |             |         | puesto          | String       | puesto       |                       |        |

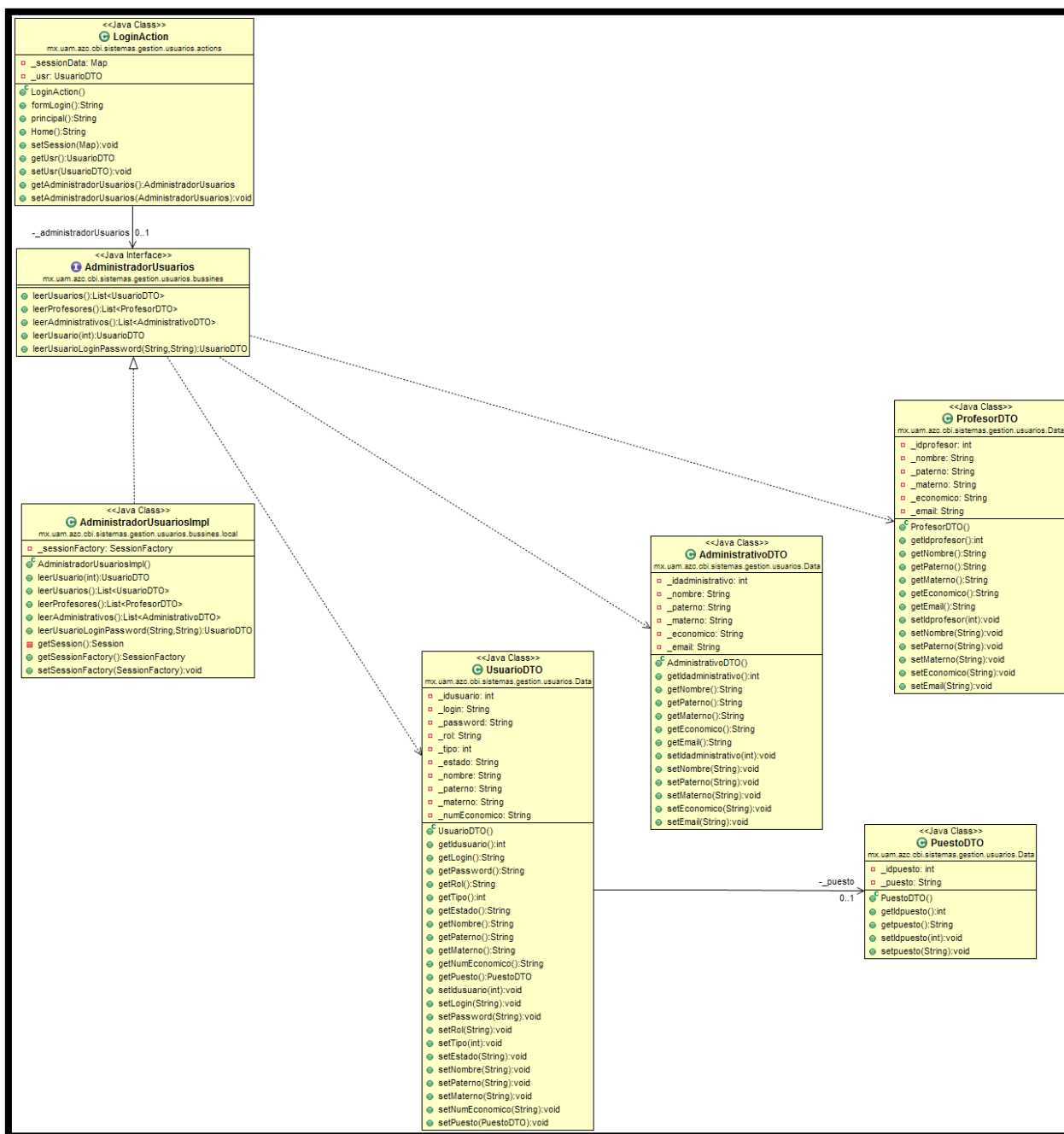
## Tabla administrativa

| Nombre de la Tabla | Descripción   |              |             |         | Nombre de Clase   | Diseño       |           |                       |        |
|--------------------|---|--------------|-------------|---------|-------------------|--------------|-----------|-----------------------|--------|
| administrativo     | Tabla para almacenar información relacionada con los administrativos. |              |             |         | AdministrativoDTO | CATALOGO     |           |                       |        |
| Base de Datos      |   |              |             |         | Objetos           |              | Forma     |                       |        |
| Nombre             | Descripción   | Tipo de Dato | Restricción | Formato | Atributo          | Tipo de Dato | Etiqueta  | Nombre del componente | Página |
| id_administrativo  | Clave de administrativo   | INT          | PK          |         | idadministrativo  | INT          | none      |                       |        |
| nombre             | Nombre del administrativo   | VARCHAR(45)  |             |         | nombre            | String       | nombre    |                       |        |
| paterno            | Apellido paterno del administrativo                                   | VARCHAR(45)  |             |         | paterno           | String       | paterno   |                       |        |
| materno            | Apellido materno del administrativo                                   | VARCHAR(45)  |             |         | materno           | String       | materno   |                       |        |
| economico          | Numero económico del administrativo                                   | VARCHAR(45)  |             |         | economico         | String       | economico |                       |        |
| email              | Email del administrativo  | VARCHAR(45)  |             |         | email             | String       | email     |                       |        |
| puesto             | Clave del puesto administrativo                                       | INT          |             |         | puesto            | String       | puesto    |                       |        |

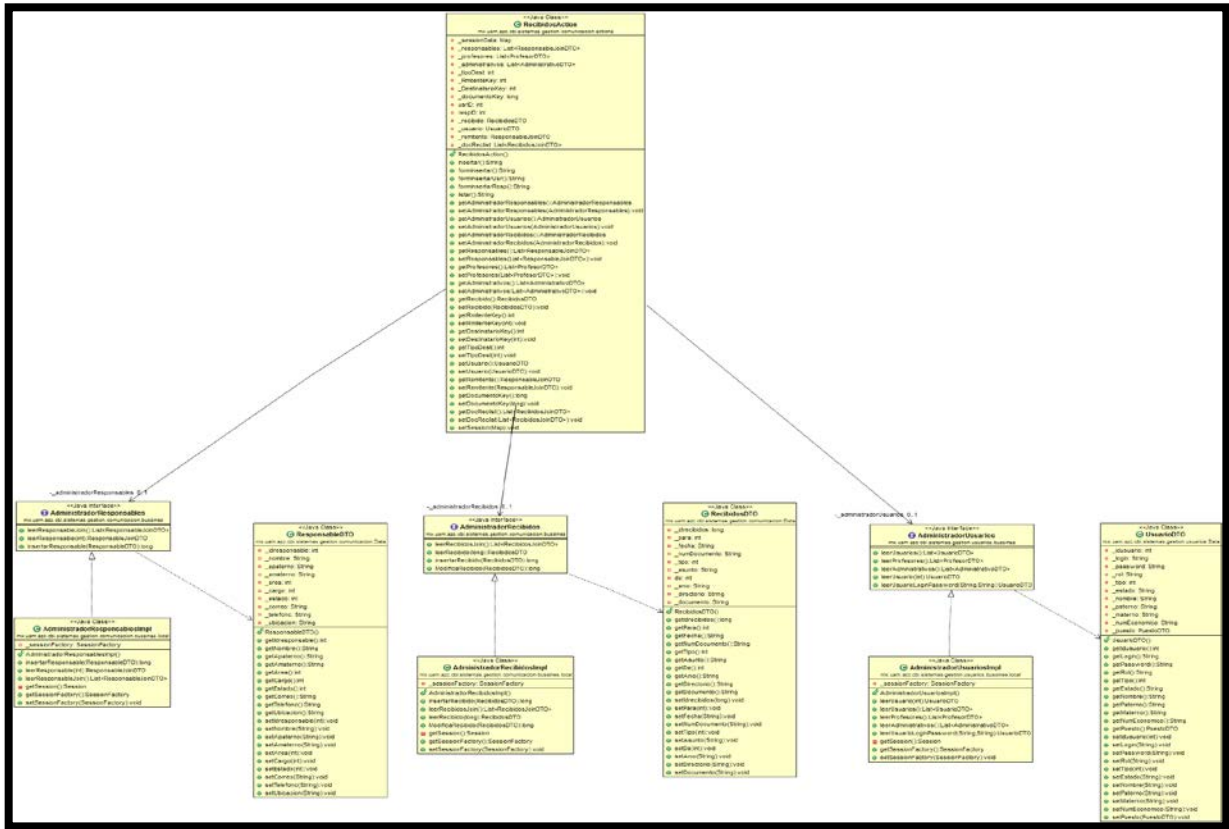
# Diagrama de Clases



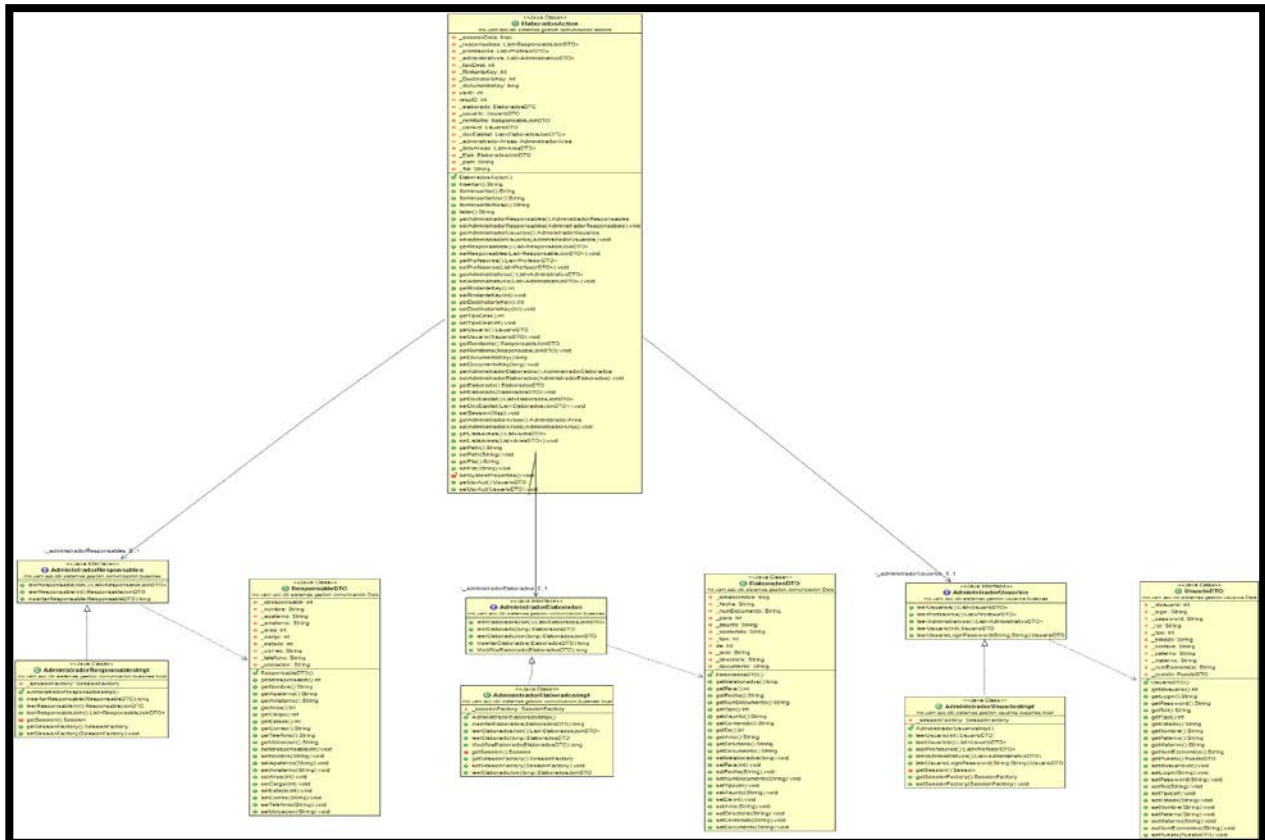
## Diagrama de clases modulo validar de usuarios



# Diagrama de clases modulo documentos recibidos



# Diagrama de clases modulo documentos elaborado







# Modelo de Casos de uso

## Diagrama general de casos de uso

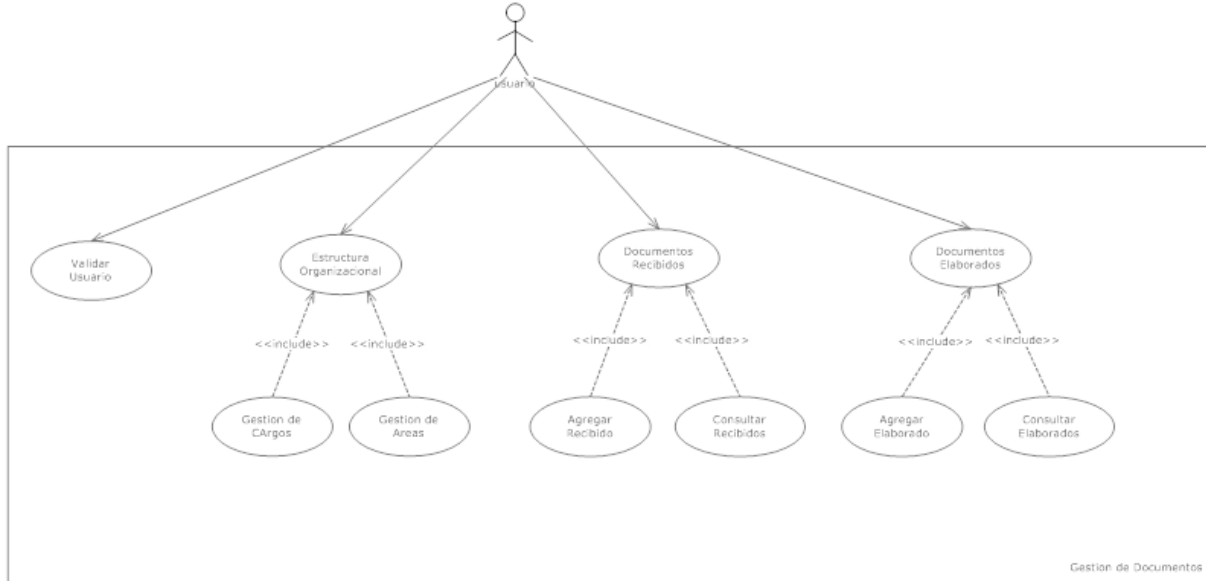
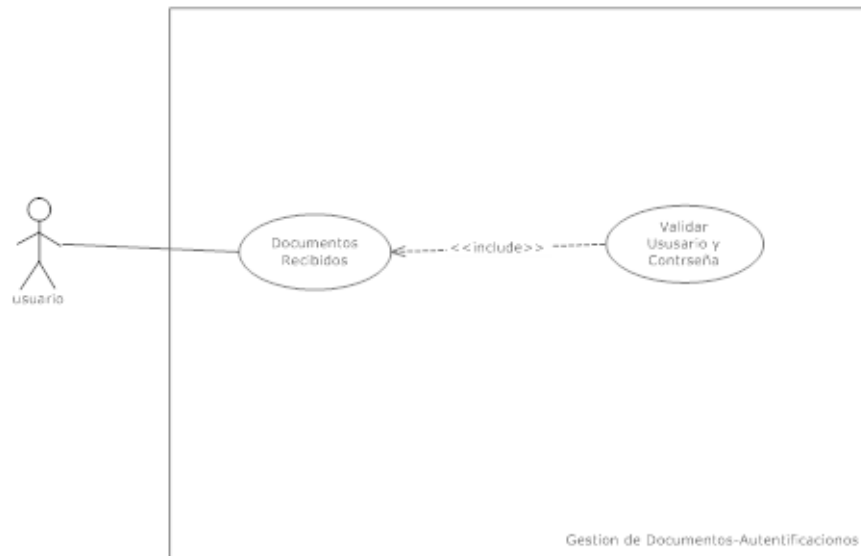


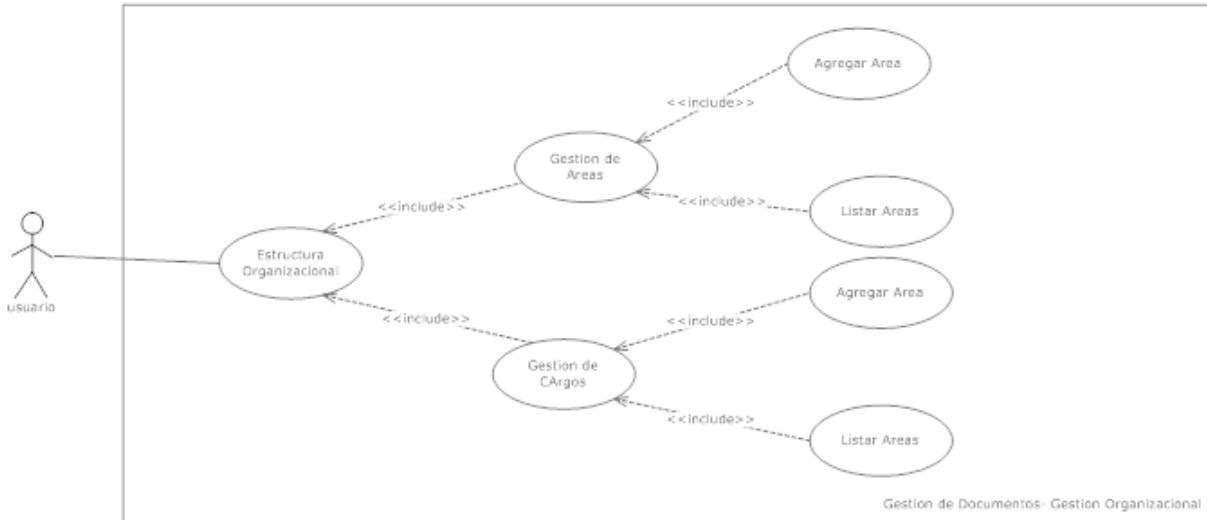
Diagrama general de casos de uso

## Diagrama de caso de uso validar usuario



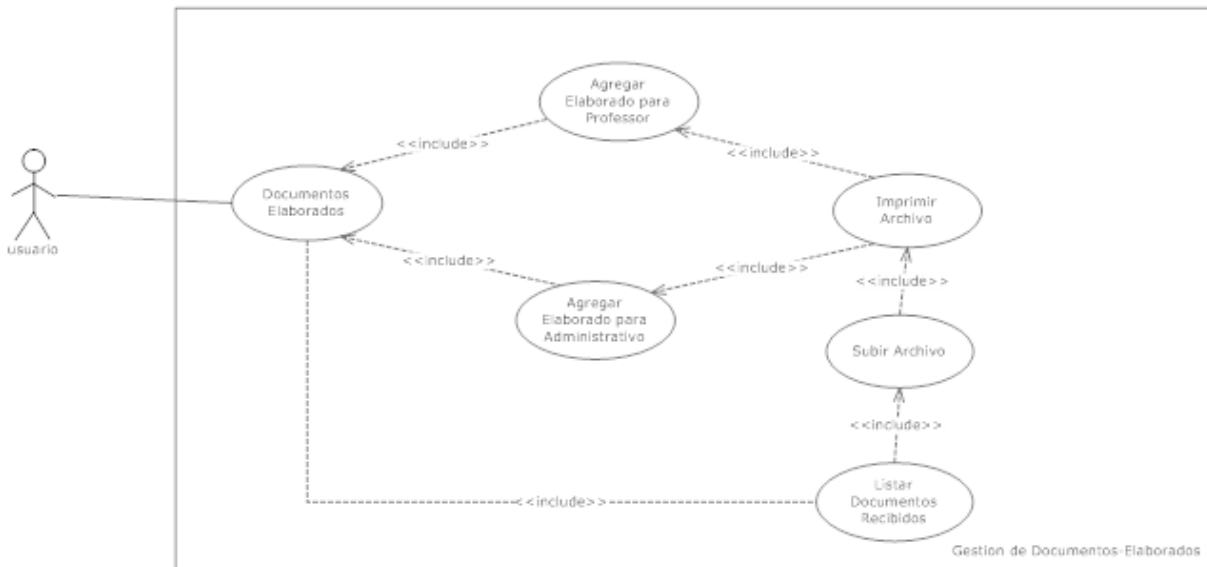
Caso de uso Validar usuario

## Diagrama de caso de uso estructura organizacional



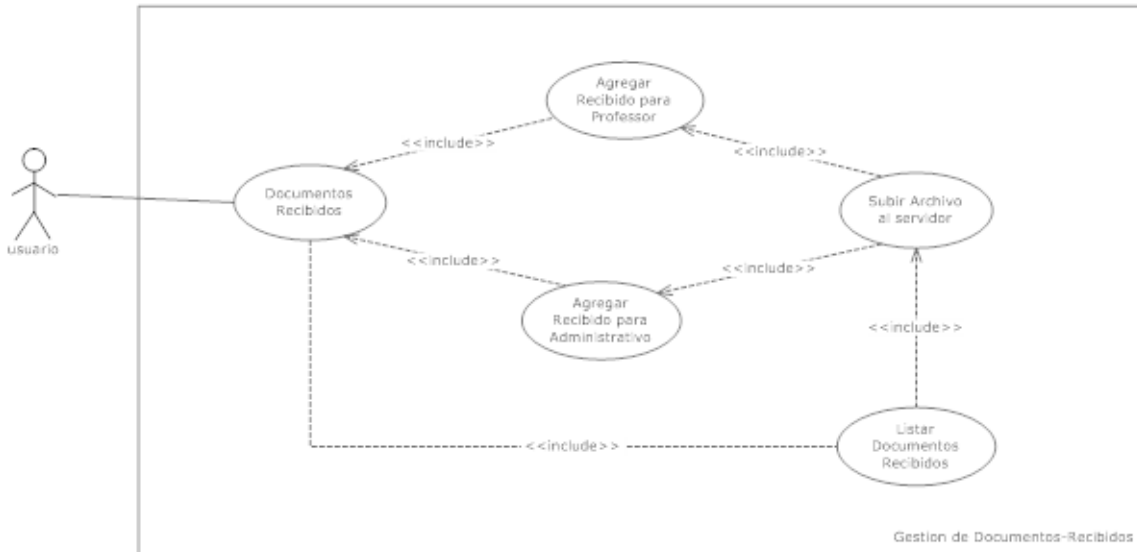
Caso de uso Gestión de Estructura Organizacional

## Diagrama de caso de uso documentos Elaborados



Caso de uso Gestión de documentos Elaborados

## Diagrama de caso de uso Gestión de documentos Recibidos



Caso de uso Gestión de documentos Recibidos

**Descripción de las actividades del Proceso de Estructura Organizacional**

| <b>N° Actividad</b> | <b>Descripción</b>   | <b>Responsable</b>                       |
|---------------------|--|--|
| 1.0                 | El Administrador del sistema recibe la petición de agregar a una persona al directorio del departamento  | Administrador                            |
| 1.1                 | Ingresa al sistema para registrarlo en el sistema  | Administrador                            |
| 2.0                 | Se recopilan los datos de que área pertenece y que puesto ocupa  | Administrador                            |
| 2.1                 | Se dan de alta en el modulo de Área y Puesto   | Administrador                            |
| 3.0                 | Se registra el responsable con los siguientes datos: <ul style="list-style-type: none"> <li>• Nombre</li> <li>• Apellido Paterno</li> <li>• Apellido Materno</li> <li>• Correo electrónico</li> <li>• Teléfono</li> <li>• ubicación</li> <li>• de quien área es</li> <li>• que puesto tiene</li> </ul> | Asistente administrativo                 |
| 4.0                 | Una vez agregado el Responsable el Administrador y el asistente administrativo puede visualizar los que se han agregado.   | Administrador y Asistente administrativo |
|                     | FIN  |  |

\*Responsable persona que ocupa un puesto dentro de un área de la Organización Estructural

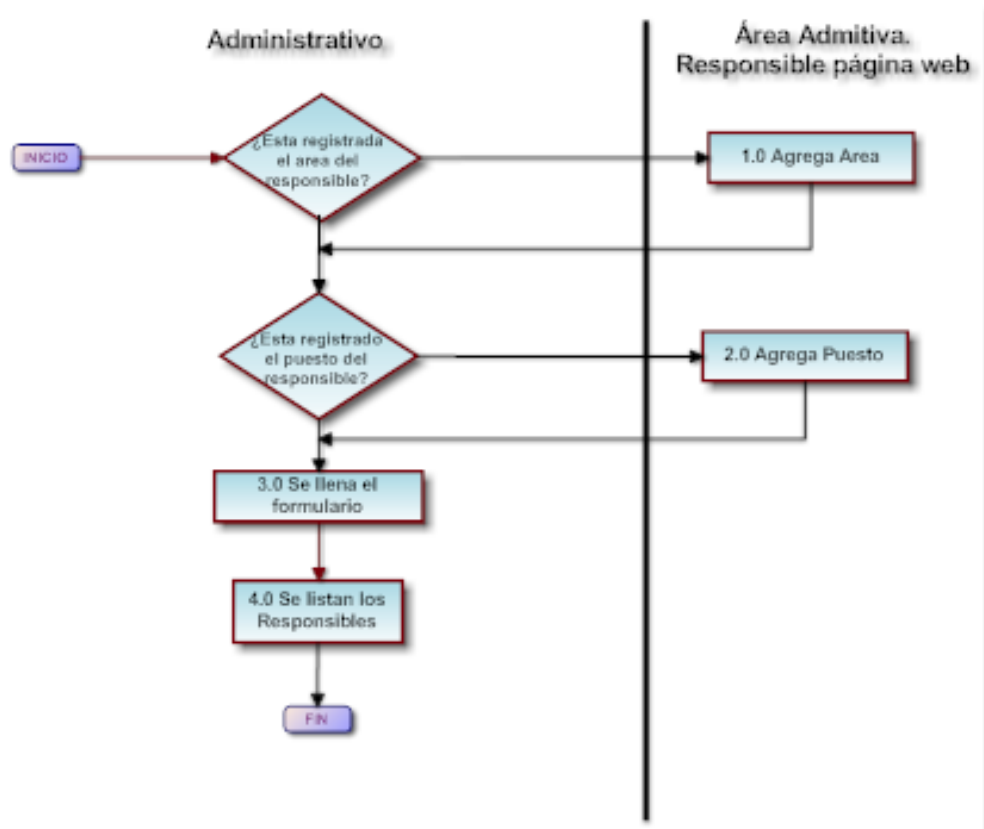
### Descripción de las actividades del Proceso de Recepción de Oficios

| N°<br>Actividad | Descripción  | Responsable              |
|-----------------|--|--------------------------|
| 1.0             | El Asistente administrativo del departamento de sistemas reciben alguna Invitación, Aviso, Informe o minuta de algún departamento de la UAM o alguna institución externa al departamento de sistemas.  | Asistente administrativo |
| 1.1             | Se realiza el acuse y se entrega en la división en donde ponen el sello con la fecha y hora en que fue recibido.   | Asistente administrativo |
| 2.0             | Se digitaliza el documento y se registra un nuevo documento recibido.  | Asistente administrativo |
| 2.3             | Se archiva el acuse tanto en carpeta física como en archivo digitalizado   | Personal Administrativo  |
| 2.4             | Si el Remitente no se encuentra en el sistema este debe solicitar al administrador que lo agregue a la Organización Estructural o Directorio del departamento  | Asistente administrativo |
| 3.0             | <p>Se registra el documento recibido con los siguientes datos:</p> <ul style="list-style-type: none"> <li>• fecha</li> <li>• numero del documento</li> <li>• tipo</li> <li>• asunto</li> <li>• de quien proviene</li> <li>• Propio del expediente</li> </ul> <p>Se adjunta el archivo digitalizado ya sea en formato PDF o en caso de que sea un conjunto de archivos que acompañan el documento en un archivo comprimido ZIP.</p> | Asistente administrativo |
| 4.0             | Si es propio del expediente de algún particular este se deberá seleccionar a que tipo de usuario es si es un administrativo o un profesor para ser seleccionado y generar una copia del archivo en su expediente.  | Asistente administrativo |
| 4.1             | Una vez agregado el documento el usuario puede visualizar los documentos que se han agregado.  | Asistente administrativo |
|                 | FIN  |                          |

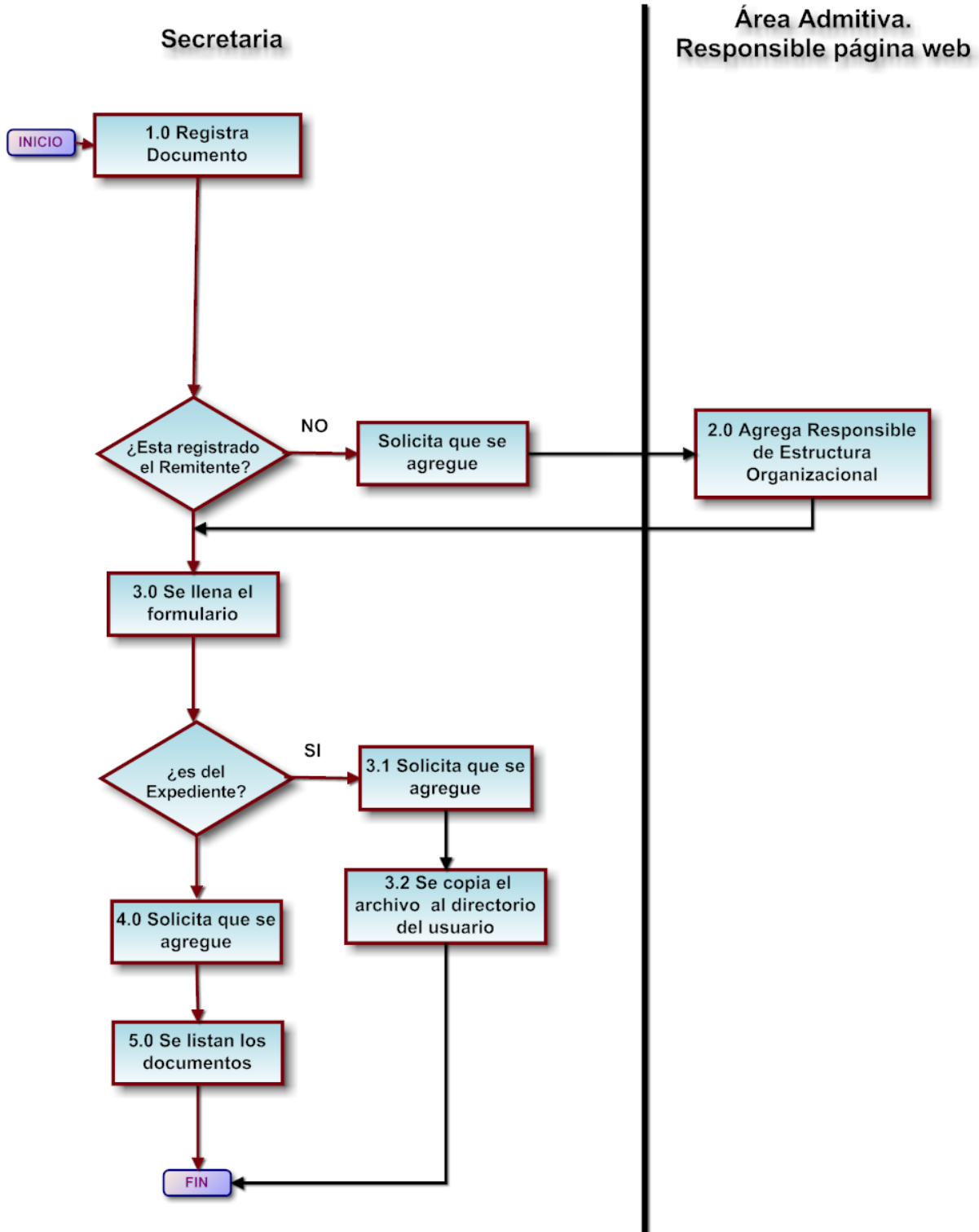
### Descripción de las actividades del Proceso de Elaboración de Oficios

| N°<br>Actividad | Descripción  | Responsable              |
|-----------------|--|--------------------------|
| 1.0             | El Asistente administrativo recibe la solicitud de elaboración de oficio para algún departamento de la UAM o de alguna institución externa.  | Asistente administrativo |
| 2.0             | Solicita a la jefatura de sistemas que se asigne el número de documento.   | Asistente administrativo |
| 2.1             | Ingresa al sistema para generar el documento   | Asistente administrativo |
| 2.2             | <p>Se genera el documento con los siguientes datos:</p> <ul style="list-style-type: none"> <li>• de</li> <li>• fecha</li> <li>• numero del documento</li> <li>• Para quien esta dirigido</li> <li>• tipo</li> <li>• asunto</li> </ul> <p>Se genera el archivo en formato PDF para su impresión</p> | Asistente administrativo |
| 3.0             | Se imprime y se recopilan las firmas y sellos necesarios para su entrega.  | Asistente administrativo |
| 3.1             | Se digitaliza el acuse de recibo y se registra el documento elaborado.   | Personal Administrativo  |
| 3.2             | Se archiva el acuse de recibo tanto en carpeta física como en archivo digitalizado.  | Asistente administrativo |
| 4.1             | Una vez agregado el documento el usuario puede visualizar los documentos que se han agregado.  | Asistente administrativo |
|                 | FIN  |                          |

## Diagrama de Actividad Gestión Organizacional

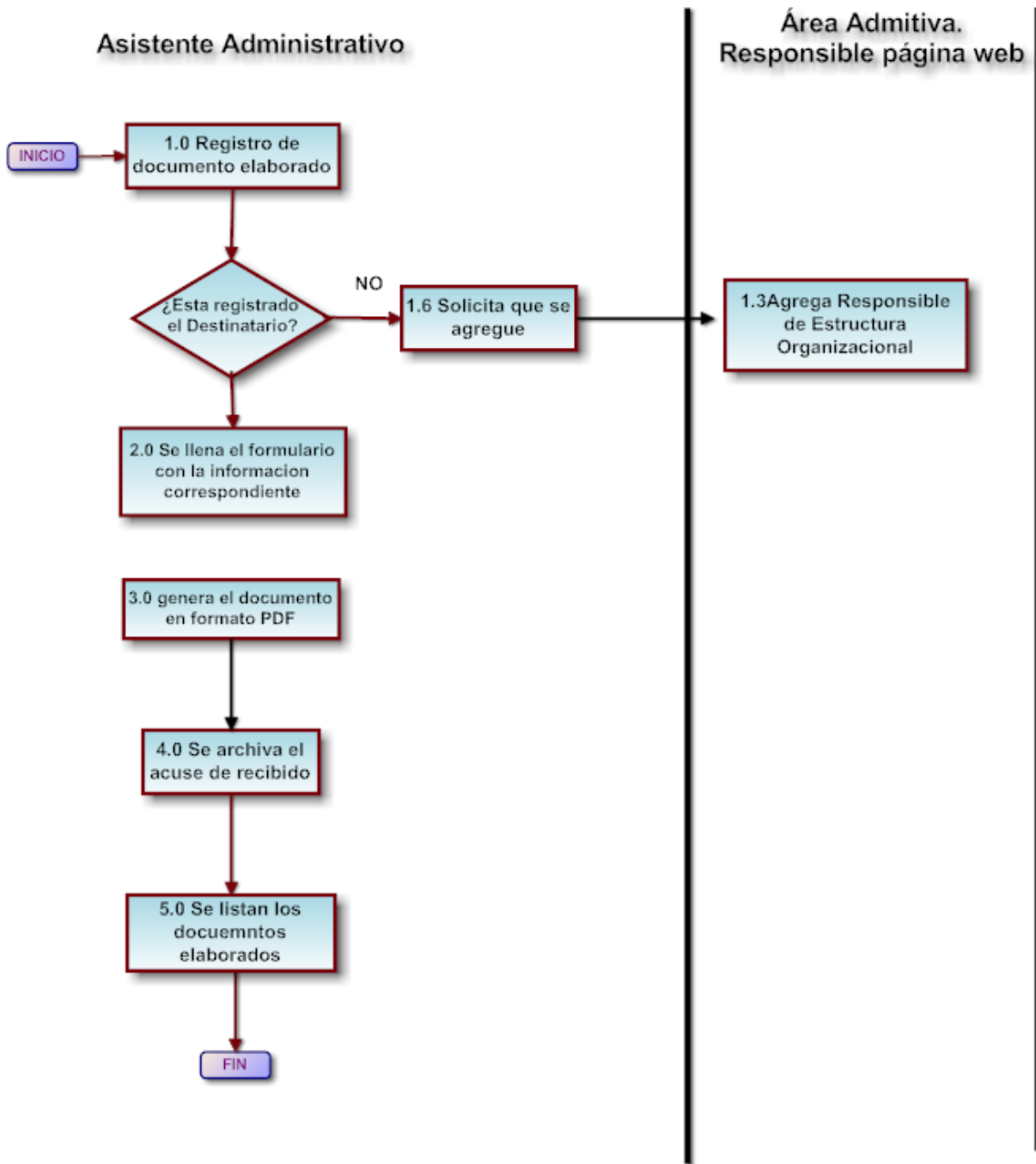


## Diagrama de Actividad Documentos Recibidos





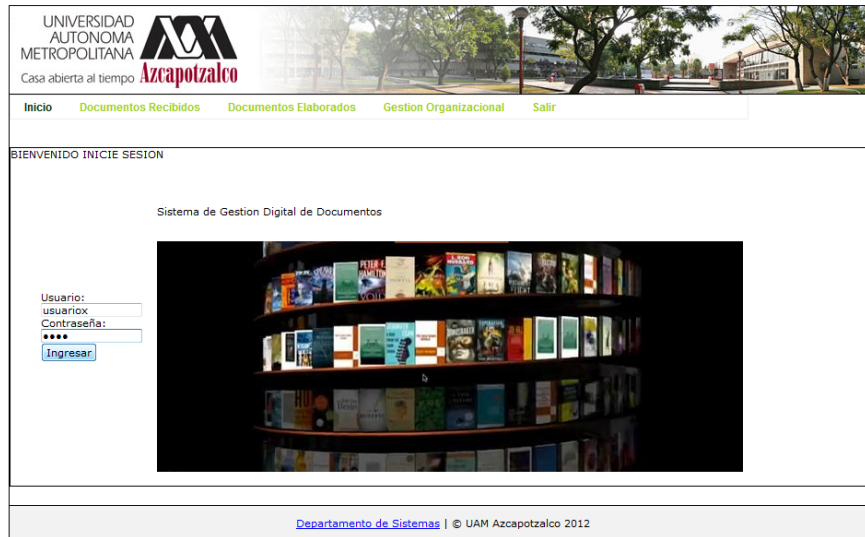
## Diagrama de Actividad Documentos Elaborados



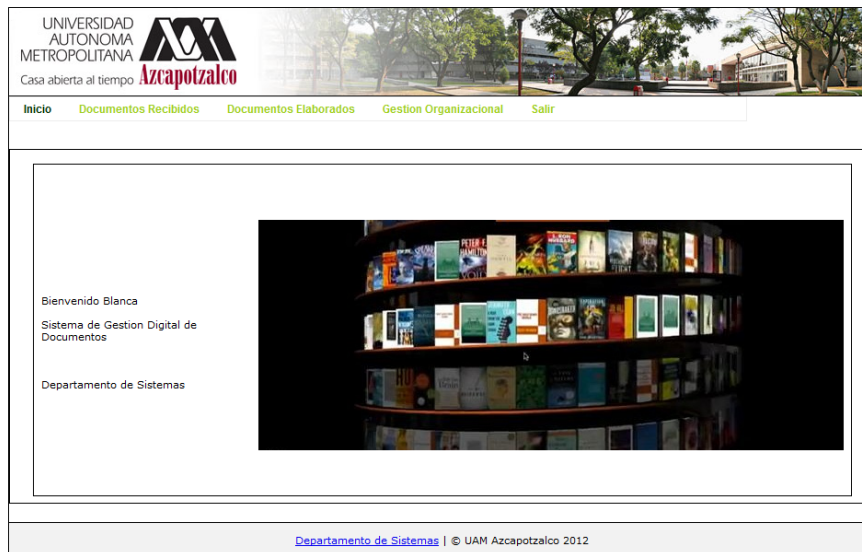
# Documentación de casos de uso

## Caso de uso Validar Usuario

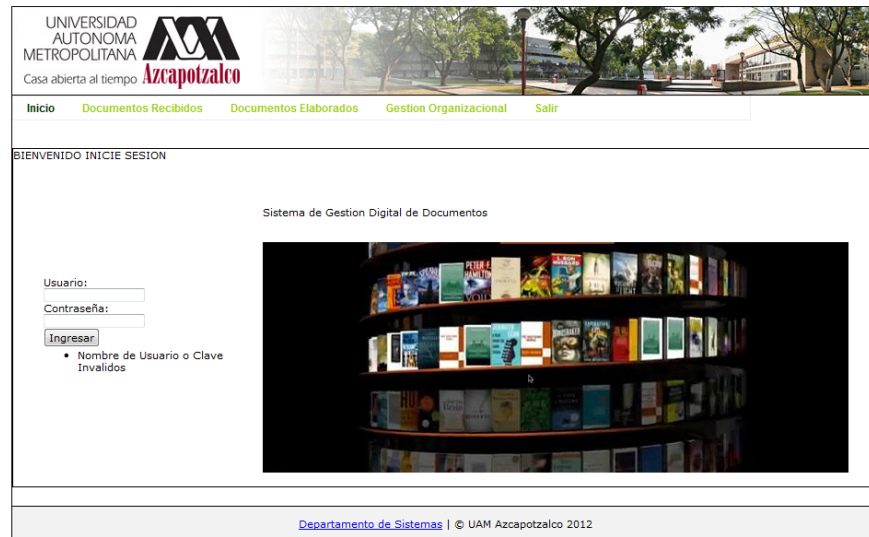
El usuario introduce su clave de usuario y su password



El usuario y password se envían al controlador, se consulta en la base de datos, si los datos son válidos se muestra la siguiente pantalla.

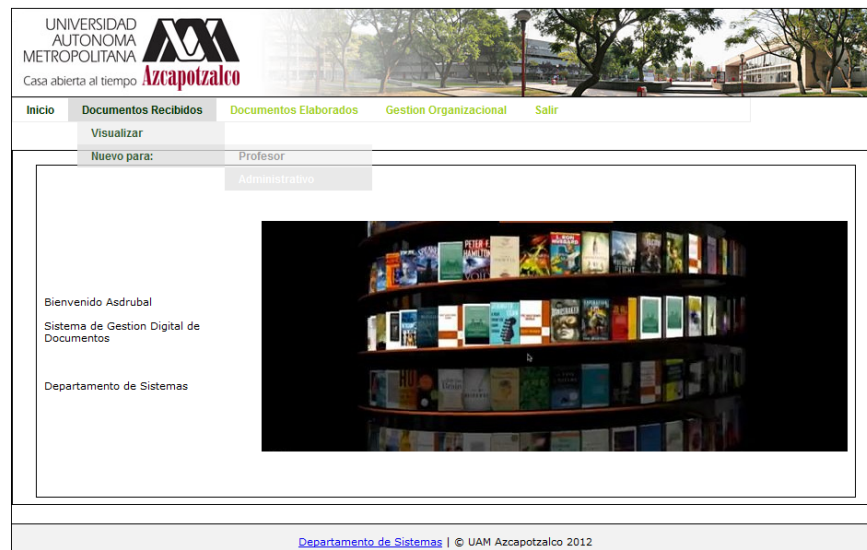


Si los datos no corresponden con los que se encuentran en la base de datos se re direcciona a la pantalla de login y se informa que los datos no son validos.



## Caso de uso Documentos recibidos

Se selecciona del menú la opción **Documentos Recibidos**, del menú desplegable se elige la opción **Nuevo para:** y se escoge la opción de **Profesores** o **Administrativos** ya que los documento pueden llegar para algún profesor o un administrativo del departamento de sistemas



Se selecciona al remitente del área administrativa que es el que envía el documento, si este no se encuentra en la lista se debe agregar en el modulo de **Gestión Organizacional** y dar aviso al administrador de la pagina para que corrobore datos y se active para su uso.

UNIVERSIDAD AUTONOMA METROPOLITANA Azcapotzalco

Inicio Documentos Recibidos Documentos Elaborados Gestion Organizacional Salir

Nuevo Documento Recibido

De:

Seleccione a quien va dirigido el documento

| Nombre:   | Cargo:                         |
|---|--------------------------------|
| <a href="#">Secretaria 01 Paterno Materno</a>         | <a href="#">Administrativo</a> |
| <a href="#">Secretaria 02 A Paterno A Materno</a>     | <a href="#">Administrativo</a> |
| <a href="#">Elena Cruz Miguel</a>                     | <a href="#">Administrativo</a> |
| <a href="#">Administrativo 02 A Paterno A Materno</a> | <a href="#">Administrativo</a> |

Para:

Seleccione quien envia el documento

| Nombre:                                | Area:                            | Cargo:                               |
|--|----------------------------------|--------------------------------------|
| <a href="#">Jazmin Pastrana</a>        | <a href="#">CYAD</a>             | <a href="#">Secretario Academico</a> |
| <a href="#">Juan Asdrubal Pastrana</a> | <a href="#">CBI</a>              | <a href="#">Rector</a>               |
| <a href="#">Ricardo Flores</a>         | <a href="#">CBI</a>              | <a href="#">Jefe de Departamento</a> |
| <a href="#">Gabriel Soto</a>           | <a href="#">Rectoria General</a> | <a href="#">Rector</a>               |

Web/recibidos formInsertarUsr.action?DestinatarioKey=3001

Si se selecciona del menú principal la opción **Nuevo para:** se elige **profesor** se desplegara la misma pantalla pero con una lista de profesores del área de sistemas para seleccionar al usuario que se le envía el documento.

UNIVERSIDAD AUTONOMA METROPOLITANA Azcapotzalco

Inicio Documentos Recibidos Documentos Elaborados Gestion Organizacional Salir

Nuevo Documento Recibido

De:

Nuevo para: **Profesor**

Administrativo

Seleccione a quien va dirigido el documento

| Nombre:                                     | Cargo:                   |
|---|--------------------------|
| <a href="#">Blanca Silva Lopez</a>          | <a href="#">Profesor</a> |
| <a href="#">Hugo Pablo Levva</a>            | <a href="#">Profesor</a> |
| <a href="#">Risto Fermin Banquel Kuoppa</a> | <a href="#">Profesor</a> |
| <a href="#">Francisco Zaragoza Martinez</a> | <a href="#">Profesor</a> |

Para:

Seleccione quien envia el documento

| Nombre:                                | Area:                            | Cargo:                               |
|--|----------------------------------|--------------------------------------|
| <a href="#">Jazmin Pastrana</a>        | <a href="#">CYAD</a>             | <a href="#">Secretario Academico</a> |
| <a href="#">Juan Asdrubal Pastrana</a> | <a href="#">CBI</a>              | <a href="#">Rector</a>               |
| <a href="#">Ricardo Flores</a>         | <a href="#">CBI</a>              | <a href="#">Jefe de Departamento</a> |
| <a href="#">Gabriel Soto</a>           | <a href="#">Rectoria General</a> | <a href="#">Rector</a>               |

nWeb/recibidos formInsertar.action?tipoDest=1

Se selecciona al usuario del departamento de sistemas al que esta dirigido el documento

UNIVERSIDAD AUTONOMA METROPOLITANA Azcapotzalco

Inicio Documentos Recibidos Documentos Elaborados Gestion Organizacional Salir

Nuevo Documento Recibido

De:  
Elena Cruz Miguel Asistente Administrativo

Para:  
Seleccione quien envia el documento

| Nombre:                                | Area:                            | Cargo:                               |
|--|----------------------------------|--------------------------------------|
| <a href="#">Jazmin Pastrana</a>        | <a href="#">CYAD</a>             | <a href="#">Secretario Academico</a> |
| <a href="#">Juan Asdrubal Pastrana</a> | <a href="#">CBI</a>              | <a href="#">Rector</a>               |
| <a href="#">Ricardo Flores</a>         | <a href="#">CBI</a>              | <a href="#">Jefe de Departamento</a> |
| <a href="#">Gabriel Soto</a>           | <a href="#">Rectoria General</a> | <a href="#">Rector</a>               |

Fecha:

Numero del documento:

Tipo:

[Web/recibidos\\_formInsertarResp.action?RmitenteKey=7](#)

Se llena el formulario con la información correspondiente al documento recibido y se presiona el botón **Siguiente**.

UNIVERSIDAD AUTONOMA METROPOLITANA Azcapotzalco

Inicio Documentos Recibidos Documentos Elaborados Gestion Organizacional Salir

Nuevo Documento Recibido

De:  
Elena Cruz Miguel Asistente Administrativo

Para:  
Jazmin Pastrana Avila Secretario Academico CYAD

Fecha:

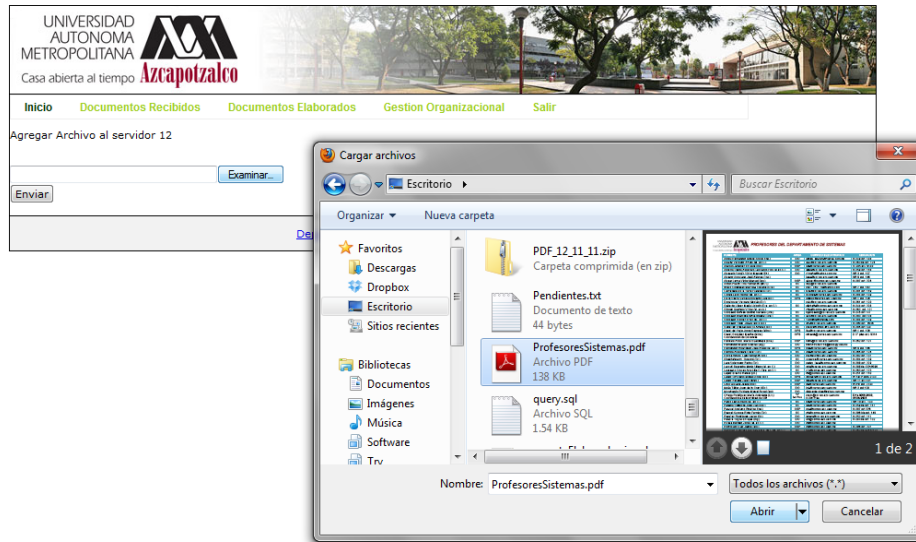
Numero del documento:

Tipo:

Asunto:

[Departamento de Sistemas](#) | © UAM Azcapotzalco 2012

El sistema genera el registro del documento en la base de datos y se despliega la pantalla para buscar el documento recibido digitalizado. Se selecciona el botón **Examinar** y se busca el documento en el sistema de archivos del usuario, una vez seleccionado se presiona el botón de **Abrir**.



Se selecciona la opción enviar para que el sistema envíe el archivo al servidor.

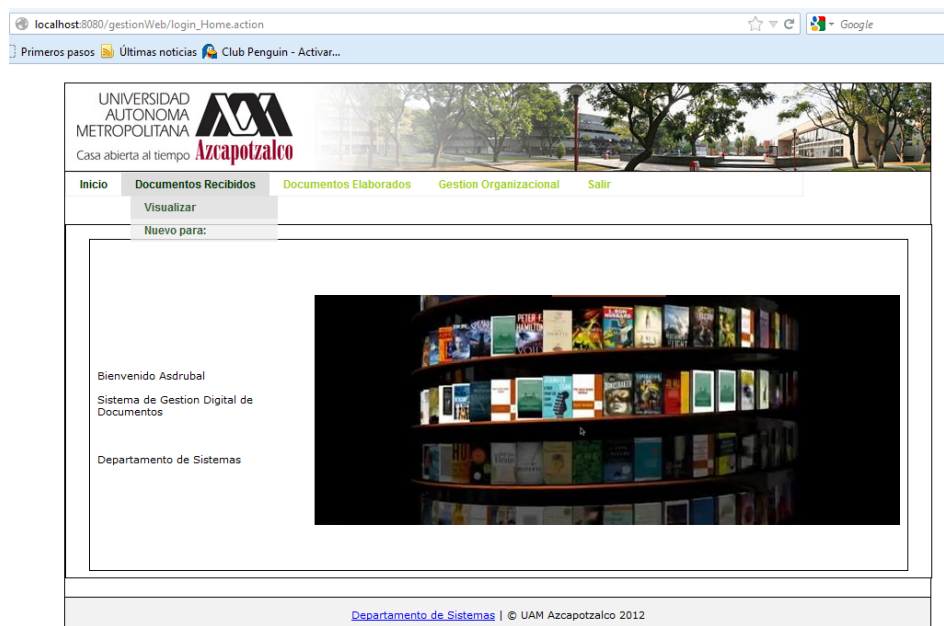


Una vez enviado el archivo el sistema lo coloca en el directorio que le corresponde en el servidor y se re direcciona a la pantalla documentos recibidos en la que se muestran todos los documentos enviados al servidor

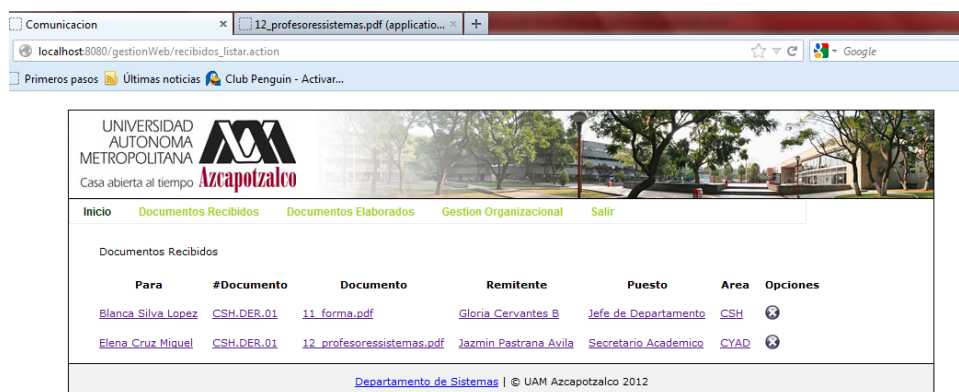
| Para               | #Documento | Documento                | Remitente             | Puesto               | Area | Opciones |
|--------------------|------------|--------------------------|-----------------------|----------------------|------|----------|
| Blanca Silva Lopez | CSH.DER.01 | 11_forma.pdf             | Gloria Cervantes B    | Jefe de Departamento | CSH  |          |
| Elena Cruz Miquel  | CSH.DER.01 | 12_profesoresistemas.pdf | Jazmin Pastrana Avila | Secretario Academico | CYAD |          |

## Caso de uso Visualizar Documentos Recibidos

Se selecciona del menú principal la opción **documentos Recibidos** y del menú desplegable se selecciona la opción visualizar

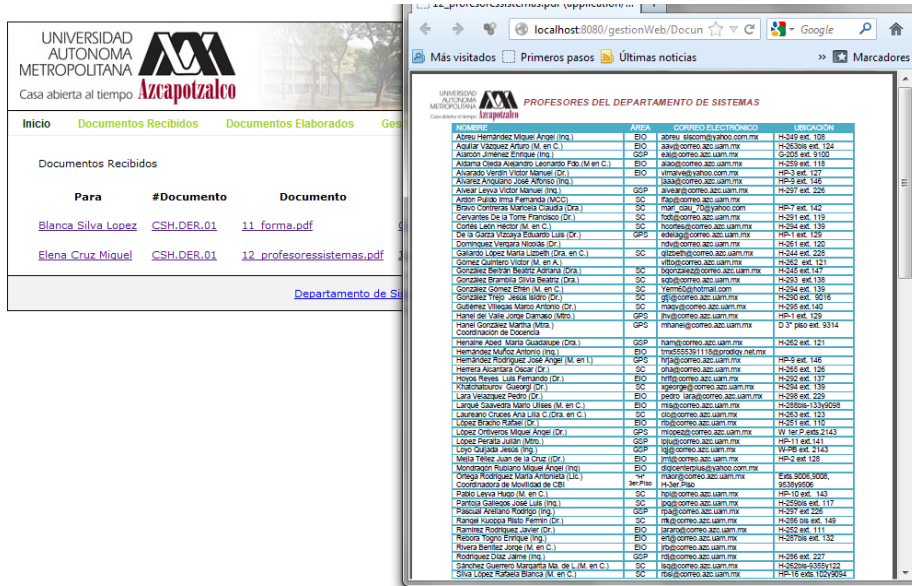


Se muestra la pantalla en donde se listan los documentos recibidos y se puede seleccionar alguno de ellos para descargarlo





Se desplegara una ventana con el documento si el navegador cuenta con el visualizador del archivo como de PDF.



Si se desea eliminar un documento recibido en la lista de documentos se encuentra un encabezado llamado **Opciones** el cual proporciona un botón para eliminar.



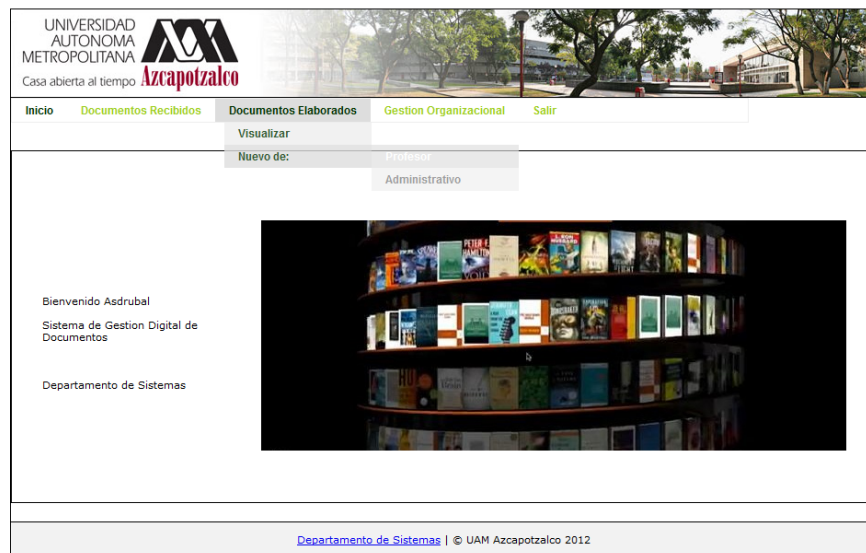
El sistema te re direcciona a la misma pantalla mostrando el la lista de los documentos recibidos sin el documento previamente eliminado.





## Caso de uso documentos elaborados

Se selecciona del menú la opción **Documentos Elaborados**, del menú desplegable se elige la opción **Nuevo para:** y se escoge la opción de **Profesores** o **Administrativos** ya que pueden llegar solicitudes de elaborar documentos de algún profesor o un administrativo del departamento de sistemas



**Seleccionar al usuario del departamento de sistemas que envía el documento**

UNIVERSIDAD AUTONOMA METROPOLITANA Azcapotzalco

Inicio Documentos Recibidos Documentos Elaborados Gestion Organizacional Salir

Nuevo Documento Elaborados

De:

Seleccione a quien envia el documento

| Nombre:                                     | Cargo:                   |
|---|--------------------------|
| <a href="#">Blanca Silva Lopez</a>          | <a href="#">Profesor</a> |
| <a href="#">Hugo Pablo Levva</a>            | <a href="#">Profesor</a> |
| <a href="#">Risto Fermin Rangel Kuoppa</a>  | <a href="#">Profesor</a> |
| <a href="#">Francisco Zaragoza Martinez</a> | <a href="#">Profesor</a> |

Para:

Seleccione quien envia el documento

| Nombre:                                | Area:                            | Cargo:                               |
|--|----------------------------------|--------------------------------------|
| <a href="#">Jazmin Pastrana</a>        | <a href="#">CYAD</a>             | <a href="#">Secretario Academico</a> |
| <a href="#">Juan Asdrubal Pastrana</a> | <a href="#">CBI</a>              | <a href="#">Rector</a>               |
| <a href="#">Ricardo Flores</a>         | <a href="#">CBI</a>              | <a href="#">Jefe de Departamento</a> |
| <a href="#">Gabriel Soto</a>           | <a href="#">Rectoria General</a> | <a href="#">Rector</a>               |

stionWeb/elaborados formInsertarUsr.action?DestinatarioKey=201

Se selecciona al destinatario del directorio que es al que se envía el documento, si este no se encuentra en la lista se debe agregar en el modulo de **Gestión Organizacional** y dar aviso al administrador de la pagina para que corrobore datos y se active para su uso.

UNIVERSIDAD AUTONOMA METROPOLITANA Azcapotzalco

Inicio Documentos Recibidos Documentos Elaborados Gestion Organizacional Salir

Nuevo Documento Elaborados

De:

Risto Fermin Rangel Kuoppa Profesor Investigador

Para:

Seleccione quien envia el documento

| Nombre:                                | Area:                            | Cargo:                               |
|--|----------------------------------|--------------------------------------|
| <a href="#">Jazmin Pastrana</a>        | <a href="#">CYAD</a>             | <a href="#">Secretario Academico</a> |
| <a href="#">Juan Asdrubal Pastrana</a> | <a href="#">CBI</a>              | <a href="#">Rector</a>               |
| <a href="#">Ricardo Flores</a>         | <a href="#">CBI</a>              | <a href="#">Jefe de Departamento</a> |
| <a href="#">Gabriel Soto</a>           | <a href="#">Rectoria General</a> | <a href="#">Rector</a>               |

Fecha:

Numero del documento:

Asunto:

elaborados formInsertarResp.action?RmitenteKey=7

Se llena el formulario con la información correspondiente al documento elaborado y se presiona el botón **Siguiente**.

UNIVERSIDAD AUTONOMA METROPOLITANA Azcapotzalco

Inicio Documentos Recibidos Documentos Elaborados Gestion Organizacional Salir

Nuevo Documento Elaborados

De:  
Risto Fermin Rangel Kuoppa Profesor Investigador

Para:  
Jazmin Pastrana Avila Secretario Academico CYAD

Fecha: 27 de Agosto del 2012

Numero del documento: CBI.SIS/2012/02

Asunto: Aviso

Contenido:  
Por medio de la presente se comunica a todo el personal que labora en el departamento de sistemas y al personal de CYAD que se encuentra trabajando en nuestras instalaciones que va a quedar cerrado el laboratorio de base de datos por remodelacion y cambio de infraestructura de red

[Siguiente](#)

[Departamento de Sistemas](#) | © UAM Azcapotzalco 2012

El sistema genera un archivo temporal en formato PDF

UNIVERSIDAD AUTONOMA METROPOLITANA Azcapotzalco

Inicio Documentos Recibidos Documentos Elaborados Gestion Organizacional Salir

Descargue el documento para su impresion: 49

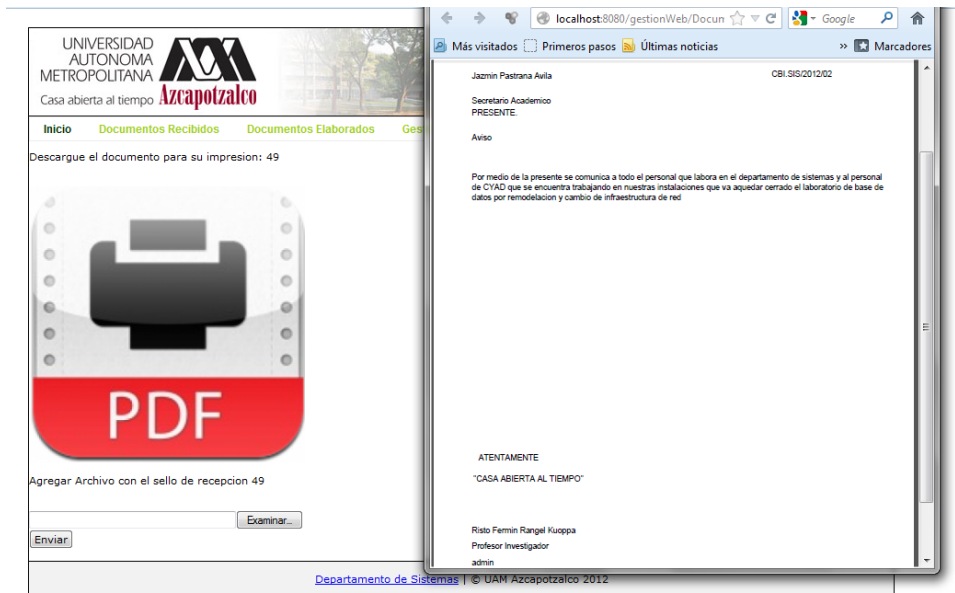


Agregar Archivo con el sello de recepcion 49

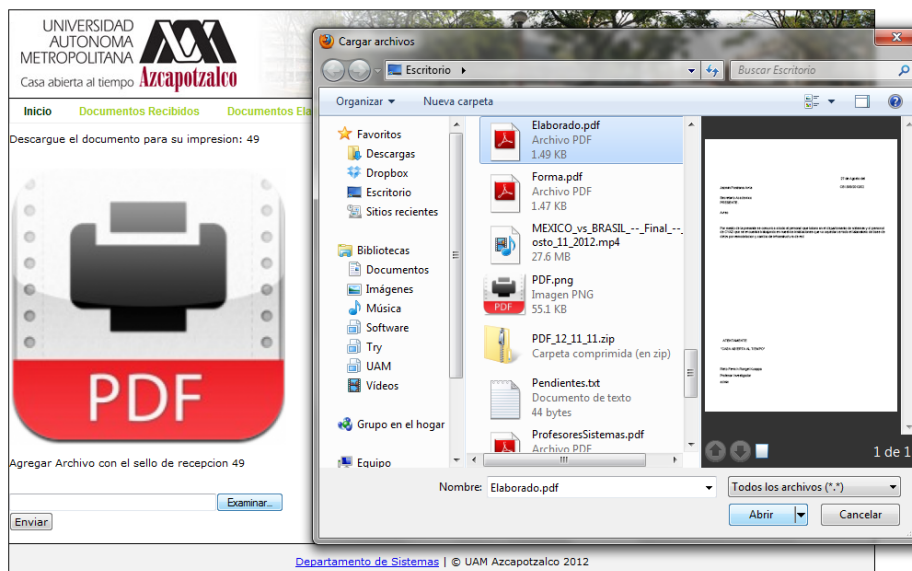
[Enviar](#) [Examinar...](#)

[Departamento de Sistemas](#) | © UAM Azcapotzalco 2012

Se selecciona la imagen de la impresora para descargar el archivo e imprimirlo en hoja membretada del departamento de sistemas.



Una vez impreso el documento y entregado se digitaliza el acuse de recibo presiona el boton exminar para buscar el archivo en el sistema de archivos del usuario, una vez seleccionado se preciona el boton aceptar.



Se selecciona la opción de enviar para subir el archivo al servidor y se almacene en el directorio correspondiente.



Se re direcciona a la pantalla **Documentos Elaborados** para poder visualizar los documentos que se han elaborado

| #Documento                      | Para                                  | Cargo                                 | Area                             | Fecha                                 | Opciones |
|---------------------------------|---------------------------------------|---------------------------------------|----------------------------------|---------------------------------------|----------|
| <a href="#">CBL.SIS/2012/01</a> | <a href="#">Mario A. Rocha Perez</a>  | <a href="#">Coordinador de DYSCOM</a> | <a href="#">Rectoria General</a> | <a href="#">25 de Agosto del 2012</a> |          |
| <a href="#">CBL.SIS/2012/02</a> | <a href="#">Jazmin Pastrana Avila</a> | <a href="#">Secretario Academico</a>  | <a href="#">CYAD</a>             | <a href="#">27 de Agosto del 2012</a> |          |
| <a href="#">CBL.SIS/2012/02</a> | <a href="#">Jazmin Pastrana Avila</a> | <a href="#">Secretario Academico</a>  | <a href="#">CYAD</a>             | <a href="#">27 de Agosto del 2012</a> |          |

**Si se desea se puede consultar el acuse de recibo o eliminarlo en caso de que se halla seleccionado mal el archivo.**

UNIVERSIDAD AUTONOMA METROPOLITANA Azcapotzalco

Casa abierta al tiempo

Inicio Documentos Recibidos Documentos Elaborados **Gestion Organizacional**

Documentos Elaborados

| #Documento                      | Para                                  | Cargo                                 | Area                         |
|---------------------------------|---------------------------------------|---------------------------------------|------------------------------|
| <a href="#">CBI.SIS/2012/01</a> | <a href="#">Mario A. Rocha Perez</a>  | <a href="#">Coordinador de DYSCOM</a> | <a href="#">Rectoria Gen</a> |
| <a href="#">CBI.SIS/2012/02</a> | <a href="#">Jazmin Pastrana Avila</a> | <a href="#">Secretario Academico</a>  | <a href="#">CYAD</a>         |

[Departamento de Sistemas](#) | © UAM

27 de Agosto del 2012  
CBI.SIS/2012/02

Jazmin Pastrana Avila  
Secretario Academico  
PRESENTE.

Aviso

Por medio de la presente se comunica a todo el personal que labora en el departamento de sistemas y al personal de CYAD que se encuentra trabajando en nuestras instalaciones que va a quedar cerrado el laboratorio de base de datos por remodelacion y cambio de infraestructura de red

ATENTAMENTE

"CASA ABIERTA AL TIEMPO"

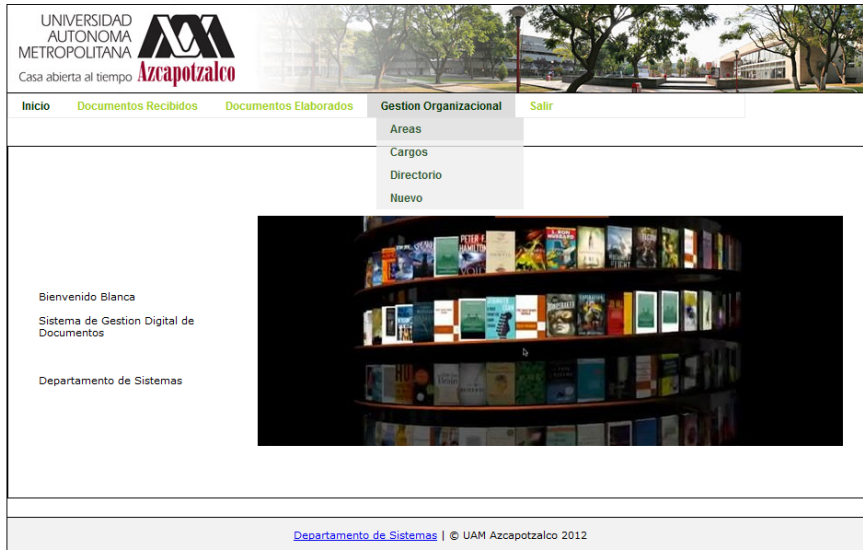
Risto Fermin Rangell Kuoppa  
Profesor Investigador  
admin

documentos/Elaborados/49\_eLaborado.pdf

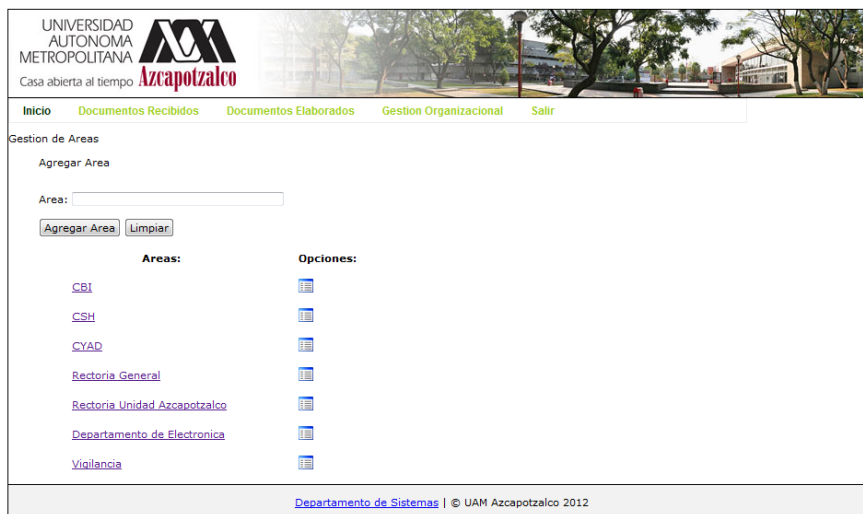
## Caso de uso Gestion Organizacional

### Agregar Areas

Se selecciona del menu principal la opcion **Gestion Organizacional** del menu desplegable se selecciona Areas.

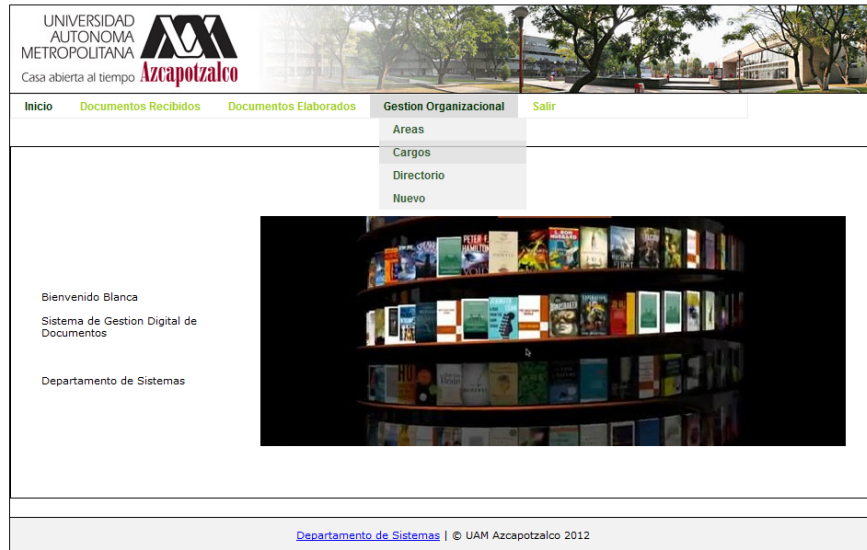


Se despliega la pantalla de gestión de Áreas donde se llena el formulario con el nombre del área para agregarla se selecciona el botón **Agregar Area**, se re direcciona a esta misma pantalla mostrando el área agregada.

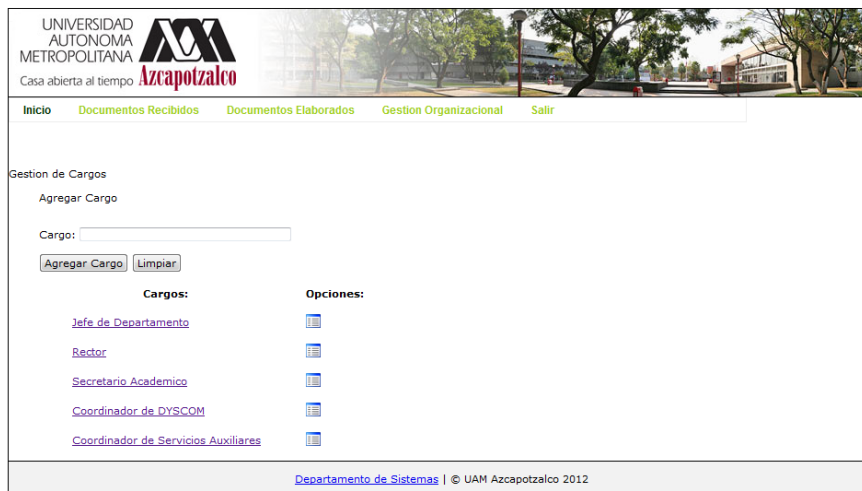


## Agregar Cargos

Se selecciona del menu principal la opcion **Gestion Organizacional** del menu desplegable se selecciona Cargos.



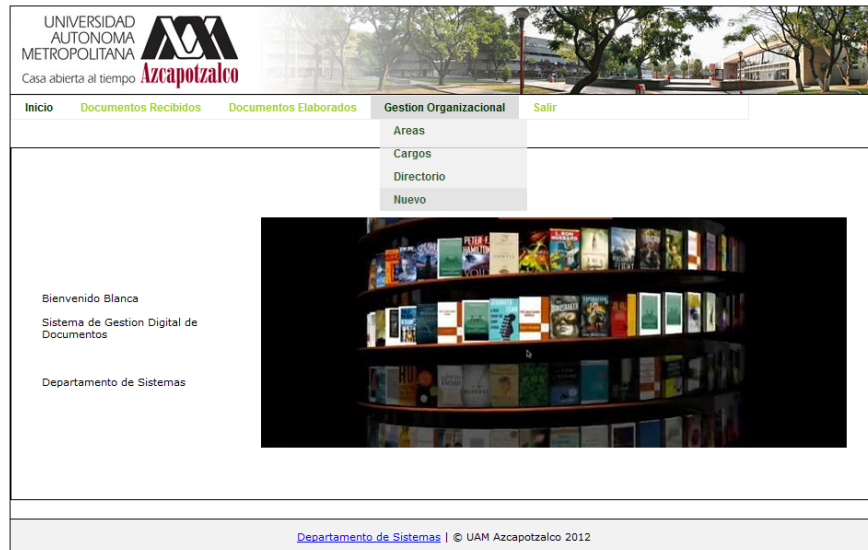
Se despliega la pantalla de gestión de Cargos donde se llena el formulario con el nombre del cargo para agregarlo se selecciona el botón **Agregar Cargo**, se re direcciona a esta misma pantalla mostrando el cargo agregado.



## Nuevo Responsable de la Gestión Organizacional (Directorio)

Se selecciona del menu principal la opcion **Gestion Organizacional** del menu desplegable se selecciona Nuevo





Se muestra la pantalla con un formulario donde se llenan los datos personales del responsable de, para agregarlo se selecciona la opción **Agregar Remitente**.


Se listan los Responsables Activos re direcciona a la pantalla siguiente con la lista de responsables . El administrador del sistema tiene que activar el responsable agregado para que este aparesca en la lista del directorio.

|   |  |                                     |           |                                  |                                      |           |
|--|--|-------------------------------------|-----------|----------------------------------|--------------------------------------|-----------|
| <a href="#">Inicio</a> <a href="#">Documentos Recibidos</a> <a href="#">Documentos Elaborados</a> <a href="#">Gestion Organizacional</a> <a href="#">Salir</a> |  |                                     |           |                                  |                                      |           |
| Nombre:  | E-Mail:  | Ubicacion:                          | Telefono: | Area:                            | Cargo:                               | Opciones: |
| <a href="#">Jazmin Pastrana Avila</a>  | <a href="mailto:mimin2003@gmail.com">mimin2003@gmail.com</a>         | <a href="#">Edif T 2do Piso #23</a> | 98765432  | <a href="#">CYAD</a>             | <a href="#">Secretario Academico</a> | 2         |
| <a href="#">Juan Asdrubal Pastrana Cede?</a>   | <a href="mailto:juan.pastrana@gmail.com">juan.pastrana@gmail.com</a> | <a href="#">Kza</a>                 | 43754248  | <a href="#">CBI</a>              | <a href="#">Rector</a>               | 2         |
| <a href="#">Ricardo Flores D</a>   | <a href="mailto:id@gmail.com">id@gmail.com</a>                       | <a href="#">Av. San Pablo</a>       | 98765432  | <a href="#">CBI</a>              | <a href="#">Jefe de Departamento</a> | 2         |
| <a href="#">Gabriel Soto Materno</a>   | <a href="mailto:login@hotmail.com">login@hotmail.com</a>             | <a href="#">HP 2do</a>              | 55223344  | <a href="#">Rectoria General</a> | <a href="#">Rector</a>               | 2         |
| <a href="#">Erika Avila Alonso</a>   | <a href="mailto:erikus@hotmail.com">erikus@hotmail.com</a>           | <a href="#">kza</a>                 | 36181410  | <a href="#">CYAD</a>             | <a href="#">Jefe de Departamento</a> | 2         |

Departamento de Sistemas | © UAM Azcapotzalco 2012

## Administración de Responsables

Se selecciona del menu principal la opcion **Gestion Organizacional** del menu desplegable se selecciona Directorio, se re despliega la pantalla siguiente con la lista de responsables . En la cual se proporciona en un encabezado las opciones Editar, Activar y desactivar responsable.

|   |  |   |           |                                  |                                       |           |
|--|--|---|-----------|----------------------------------|---------------------------------------|-----------|
| <a href="#">Inicio</a> <a href="#">Documentos Recibidos</a> <a href="#">Documentos Elaborados</a> <a href="#">Gestion Organizacional</a> <a href="#">Salir</a> |  |   |           |                                  |                                       |           |
| Nombre:  | E-Mail:  | Ubicacion:                              | Telefono: | Area:                            | Cargo:                                | Opciones: |
| <a href="#">Jazmin Pastrana Avila</a>  | <a href="mailto:mimin2003@gmail.com">mimin2003@gmail.com</a>               | <a href="#">Edif T 2do Piso #23</a>     | 98765432  | <a href="#">CYAD</a>             | <a href="#">Secretario Academico</a>  | 2         |
| <a href="#">Gloria Cervantes B</a>   | <a href="mailto:username@correo.azc.uam.mx">username@correo.azc.uam.mx</a> | <a href="#">Edif. C 2do piso</a>        | 44332211  | <a href="#">CSH</a>              | <a href="#">Jefe de Departamento</a>  | 0         |
| <a href="#">Mario A. Rocha Perez</a>   | <a href="mailto:live@hotmail.com">live@hotmail.com</a>                     | <a href="#">Uam Azc</a>                 | 12345678  | <a href="#">Rectoria General</a> | <a href="#">Coordinador de DYSCOM</a> | 0         |
| <a href="#">Juan Asdrubal Pastrana Cede?</a>   | <a href="mailto:juan.pastrana@gmail.com">juan.pastrana@gmail.com</a>       | <a href="#">Kza</a>                     | 43754248  | <a href="#">CBI</a>              | <a href="#">Rector</a>                | 2         |
| <a href="#">Ricardo Flores D</a>   | <a href="mailto:id@gmail.com">id@gmail.com</a>                             | <a href="#">Av. San Pablo</a>           | 98765432  | <a href="#">CBI</a>              | <a href="#">Jefe de Departamento</a>  | 2         |
| <a href="#">Dr. Luis E. Nore? Franco</a>   | <a href="mailto:dircbi@correo.azc.uam.mx">dircbi@correo.azc.uam.mx</a>     | <a href="#">Edificio H-P, 1er. Piso</a> | 5318 9001 | <a href="#">CBI</a>              | <a href="#">Jefe de Departamento</a>  | 0         |
| <a href="#">Gabriel Soto Materno</a>   | <a href="mailto:login@hotmail.com">login@hotmail.com</a>                   | <a href="#">HP 2do</a>                  | 55223344  | <a href="#">Rectoria General</a> | <a href="#">Rector</a>                | 2         |
| <a href="#">Erika Avila Alonso</a>   | <a href="mailto:erikus@hotmail.com">erikus@hotmail.com</a>                 | <a href="#">kza</a>                     | 36181410  | <a href="#">CYAD</a>             | <a href="#">Jefe de Departamento</a>  | 2         |

## Modificar información del remitente

El administrador puede cambiar los datos de los responsables así como el área y cargo en la que se encuentran una vez que los datos son modificados en el formulario si se desea guardar los cambios se selecciona el botón **Modificar Responsables**.

UNIVERSIDAD AUTONOMA METROPOLITANA Azcapotzalco

Casa abierta al tiempo

Inicio Documentos Recibidos Documentos Elaborados Gestion Organizacional Salir

Nombre:  Antes: Dr. Luis E.

Apellido Paterno:  Antes: Nore?

Apellido Materno:  Antes: Franco

Email:  Antes: dircbi@correo.azc.uam.mx

Teléfono:  Antes: 5318 9001

Ubicación:  Antes: Edificio H-P, 1er. Piso

Area:  Antes: CBI

Cargo:  Antes: Jefe de Departamento

Departamento de Sistemas | © UAM Azcapotzalco 2012

Se despliega la pantalla con el directorio y se muestran los cambios que se realizaron a el responsable.

UNIVERSIDAD AUTONOMA METROPOLITANA Azcapotzalco

Casa abierta al tiempo



Inicio Documentos Recibidos Documentos Elaborados Gestion Organizacional Salir

| Nombre:                      | E-Mail:                  | Ubicación:              | Telefono: | Area:            | Cargo:               | Opciones: |
|------------------------------|--------------------------|-------------------------|-----------|------------------|----------------------|-----------|
| Jazmin Pastrana Avila        | mimin2003@gmail.com      | Edif T 2do Piso #23     | 98765432  | CYAD             | Secretario Academico | 2         |
| Juan Asdrubal Pastrana Cede? | juan.pastrana@gmail.com  | Kza                     | 43754248  | CBI              | Rector               | 2         |
| Ricardo Flores D             | id@gmail.com             | Av. San Pablo           | 98765432  | CBI              | Jefe de Departamento | 2         |
| Dr. Luis E. Nore? Franco     | dircbi@correo.azc.uam.mx | Edificio H-P, 1er. Piso | 5318 9001 | CBI              | Jefe de Departamento | 2         |
| Gabriel Soto Materno         | lojin@hotmail.com        | HP 2do                  | 55223344  | Rectoria General | Rector               | 2         |
| Erika Avila Alonso           | erikus@hotmail.com       | kza                     | 36181410  | CYAD             | Jefe de Departamento | 2         |

Departamento de Sistemas | © UAM Azcapotzalco 2012

























El administrador tiene que activar al responsable una vez que este fue dado de alta o se modifico su información personal con el botón que tiene un circulo color verde con una paloma blanca (opción de en medio).

- Desactivar: 
- Activar: 
- Modificar: 

METROPOLITANA    
 Casa abierta al tiempo

[Inicio](#) [Documentos Recibidos](#) [Documentos Elaborados](#) [Gestion Organizacional](#) [Salir](#)

Administración de Responsables

| Nombre:                                      | E-Mail:  | Ubicacion:                              | Telefono: | Area:                            | Cargo:                                | Opciones:  |
|--|--|---|-----------|----------------------------------|---------------------------------------|--|
| <a href="#">Jazmin Pastrana Avila</a>        | <a href="mailto:mimin2003@gmail.com">mimin2003@gmail.com</a>               | <a href="#">Edif T 2do Piso #23</a>     | 98765432  | <a href="#">CYAD</a>             | <a href="#">Secretario Academico</a>  |   <br>2       |
| <a href="#">Gloria Cervantes B</a>           | <a href="mailto:username@correo.azc.uam.mx">username@correo.azc.uam.mx</a> | <a href="#">Edif. C 2do piso</a>        | 44332211  | <a href="#">CSH</a>              | <a href="#">Jefe de Departamento</a>  |   <br>0       |
| <a href="#">Mario A. Rocha Perez</a>         | <a href="mailto:live@hotmail.com">live@hotmail.com</a>                     | <a href="#">Uam Azc</a>                 | 12345678  | <a href="#">Rectoria General</a> | <a href="#">Coordinador de DYSCOM</a> |   <br>0       |
| <a href="#">Juan Asdrubal Pastrana Cede?</a> | <a href="mailto:juan.pastrana@gmail.com">juan.pastrana@gmail.com</a>       | <a href="#">Kza</a>                     | 43754248  | <a href="#">CBI</a>              | <a href="#">Rector</a>                |   <br>2 |
| <a href="#">Ricardo Flores D</a>             | <a href="mailto:rf@gmail.com">rf@gmail.com</a>                             | <a href="#">Av. San Pablo</a>           | 98765432  | <a href="#">CBI</a>              | <a href="#">Jefe de Departamento</a>  |   <br>2 |
| <a href="#">Dr. Luis E. Nere? Franco</a>     | <a href="mailto:dircbi@correo.azc.uam.mx">dircbi@correo.azc.uam.mx</a>     | <a href="#">Edificio H-P, 1er. Piso</a> | 5318 9001 | <a href="#">CBI</a>              | <a href="#">Jefe de Departamento</a>  |   <br>0 |
| <a href="#">Gabriel Soto Materno</a>         | <a href="mailto:login@hotmail.com">login@hotmail.com</a>                   | <a href="#">HP 2do</a>                  | 55223344  | <a href="#">Rectoria General</a> | <a href="#">Rector</a>                |   <br>2 |
| <a href="#">Erika Avila Alonso</a>           | <a href="mailto:erikus@hotmail.com">erikus@hotmail.com</a>                 | <a href="#">kza</a>                     | 36181410  | <a href="#">CYAD</a>             | <a href="#">Jefe de Departamento</a>  |   <br>2 |

[Departamento de Sistemas](#) | © UAM Azcapotzalco 2012

Una vez que el administrador active al responsable el usuario lo puede visualizar y utilizar en su entorno de trabajo.

## Caso de uso salir del sistema

Se tiene que seleccionar la opción **Salir** del menú principal y la sesión se finaliza, y se re direcciona a la pantalla de login



The screenshot shows the login interface for the 'Sistema de Gestion Digital de Documentos'. At the top left, the logo of 'UNIVERSIDAD AUTONOMA METROPOLITANA Azcapotzalco' is displayed. A navigation menu includes 'Inicio', 'Documentos Recibidos', 'Documentos Elaborados', 'Gestion Organizacional', and 'Salir'. The main content area features a login form with fields for 'Usuario:' and 'Contraseña:', and an 'Ingresar' button. A background image of a curved bookshelf is visible. The footer contains the text 'Departamento de Sistemas | © UAM Azcapotzalco 2012'.

## Código fuente

### mx.uam.azc.cbi.sistemas.gestion.usuarios.actions

#### LoginAction

```
package mx.uam.azc.cbi.sistemas.gestion.usuarios.actions;

import mx.uam.azc.cbi.sistemas.gestion.usuarios.Data.UsuarioDTO;
import mx.uam.azc.cbi.sistemas.gestion.usuarios.bussines.AdministradorUsuarios;
import java.util.*;
import com.opensymphony.xwork2.ActionSupport;
import org.apache.struts2.interceptor.SessionAware;

public class LoginAction extends ActionSupport implements SessionAware
{
    private Map _sessionData;
    private UsuarioDTO _usr;
    private AdministradorUsuarios _administradorUsuarios;

    public String formLogin()
    {
        _sessionData.remove( "usuario" );
        return "login";
    }

    public String principal()
    {
        String tmp = "principal";

        _usr = _administradorUsuarios.leerUsuarioLoginPassword( _usr.getLogin(), _usr.getPassword() );
        if ( _usr == null )
        {
            addActionMessage( "Nombre de Usuario o Clave Invalidos" );
            tmp = "login";
            return tmp;
        }
        else
        {
            _sessionData.put( "usuario", new UsuarioDTO() );
            _sessionData.put( "usuario", _usr );
        }

        return tmp;
    }

    public String Home(){
        _usr = (UsuarioDTO)_sessionData.get( "usuario" );

        return "home";
    }

    public void setSession( Map sessionData )
    {
        _sessionData = sessionData;
    }

    public UsuarioDTO getUusr()
    {
        return _usr;
    }
}
```

```

}

public void setUsr( UsuarioDTO usr )
{
    _usr = usr;
}

public AdministradorUsuarios getAdministradorUsuarios()
{
    return _administradorUsuarios;
}

public void setAdministradorUsuarios(
    AdministradorUsuarios administradorUsuarios )
{
    _administradorUsuarios = administradorUsuarios;
}
}

```

#### mx.uam.azc.cbi.sistemas.gestion.comunicacion.actions

##### PrincipalAction

```

package mx.uam.azc.cbi.sistemas.gestion.comunicacion.actions;

import mx.uam.azc.cbi.sistemas.gestion.usuarios.Data.UsuarioDTO;
import com.opensymphony.xwork2.ActionSupport;
import java.util.Map;

public class PrincipalAction extends ActionSupport
{
    UsuarioDTO _usr = new UsuarioDTO();

    private Map _sessionData;
    public String Home(){
        _usr = (UsuarioDTO)_sessionData.get( "usuario");
        return "home";
    }

    public UsuarioDTO getUsr()
    {
        return _usr;
    }

    public void setUsr( UsuarioDTO usr )
    {
        _usr = usr;
    }

    public void setSession( Map sessionData )
    {
        _sessionData = sessionData;
    }
}

```

##### ResponsibleAction

```

package mx.uam.azc.cbi.sistemas.gestion.comunicacion.actions;

import mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data.*;
import mx.uam.azc.cbi.sistemas.gestion.comunicacion.bussines.*;
import mx.uam.azc.cbi.sistemas.gestion.usuarios.Data.UsuarioDTO;
import org.apache.struts2.interceptor.SessionAware;
import com.opensymphony.xwork2.ActionSupport;
import java.util.*;

```

```

/**
 * @author Asdrubal
 */
public class ResponsableAction extends ActionSupport implements SessionAware
{
    private Map _sessionData;
    private AdministradorArea _administradorAreas;
    private List<AreaDTO> _areas = new ArrayList<AreaDTO>();
    private AdministradorCargo _administradorCargos;
    private List<CargoDTO> _cargos = new ArrayList<CargoDTO>();
    private AdministradorResponsables _administradorResponsables;
    private ResponsableDTO _resp;
    private ResponsableDTO _respCh;
    private ResponsableJoinDTO _responsableOld;
    private String _areaS;
    private String _cargoS;
    private int _Key=0;
    private int _tipo=0;
    private AreaDTO _areaInsert;
    private CargoDTO _cargoInsert;
    private UsuarioDTO _usr;
    private List<ResponsableJoinDTO> _responsables = new ArrayList<ResponsableJoinDTO>();

    public String Insertar()
    {
        AreaDTO _artmp = new AreaDTO();
        _artmp = _administradorAreas.leerAreaN( _areaS );
        _resp.setArea(_artmp.getIdarea() );

        CargoDTO _cartmp = new CargoDTO();
        _cartmp = _administradorCargos.leerCargoN( _cargoS );
        _resp.setCargo( _cartmp.getIdcargo() );

        _administradorResponsables.insertarResponsable( _resp );
        _responsables = _administradorResponsables.leerResponsableJoin();
        return "insertar";
    }

    public String Activar(){
        _usr =(UsuarioDTO) _sessionData.get( "usuario" );
        _tipo= _usr.getTipo();

        if(_tipo==2){
            _respCh= _administradorResponsables.leerResponsableNN( _Key );
            _respCh.setEstado( 2 );
            _administradorResponsables.ModificaResponsable( _respCh );
            _responsables= _administradorResponsables.leerResponsableJoin();
            return "listarAdmin";
        }
        else{
            _responsables= _administradorResponsables.leerResponsableJoinAct();
            return "listar";
        }
    }

    public String Desactivar(){
        _usr =(UsuarioDTO) _sessionData.get( "usuario" );
        _tipo= _usr.getTipo();

        if(_tipo==2){
            _respCh= _administradorResponsables.leerResponsableNN( _Key );
            _respCh.setEstado( 1 );
            _administradorResponsables.ModificaResponsable( _respCh );

            _responsables= _administradorResponsables.leerResponsableJoin();
            return "listarAdmin";
        }
        else{
            _responsables= _administradorResponsables.leerResponsableJoinAct();
            return "listar";
        }
    }
}

```



```

    }
}

public String ModificarFrm(){
    _usr =(UsuarioDTO) _sessionData.get( "usuario" );
    _tipo= _usr.getTipo();

    if(_tipo==2){
        _areas = _administradorAreas.leerlistArea();
        _cargos = _administradorCargos.leerlistCargo();

        _responsableOld = _administradorResponsables.leerResponsable( _Key );
        return "modificar";
    }
    else {
        _responsables= _administradorResponsables.leerResponsableJoinAct();
        return "listar";
    }
}

public String Modificar()
{
    AreaDTO _artmp = new AreaDTO();
    _artmp = _administradorAreas.leerAreaN( _areas );
    _resp.setArea(_artmp.getIdarea() );

    CargoDTO _cartmp = new CargoDTO();
    _cartmp = _administradorCargos.leerCargoN( _cargos );
    _resp.setCargo( _cartmp.getIdcargo() );
    _resp.setIdresponsable( _Key );
    _administradorResponsables.ModificaResponsable( _resp );

    _responsables = _administradorResponsables.leerResponsableJoin();
    return "listarAdmin";
}

public String formInsertar()
{
    _areas = _administradorAreas.leerlistArea();
    _cargos = _administradorCargos.leerlistCargo();
    return "formInsertar";
}

public String listar()
{
    _usr =(UsuarioDTO) _sessionData.get( "usuario" );
    _tipo= _usr.getTipo();

    if(_tipo==2){
        _responsables= _administradorResponsables.leerResponsableJoin();
        return "listarAdmin";
    }
    else {
        _responsables= _administradorResponsables.leerResponsableJoinAct();
        return "listar";
    }
}

//Gestion Areas

public String gAreas()
{
    _areas = _administradorAreas.leerlistArea();
    return "gAreas";
}

```

```

public String gAreasInsert()
{
    _administradorAreas.insertarArea( _areaInsert );
    _areas = _administradorAreas.leerlistArea();
    return "gAreasInsert";
}

//Gestion Cargos

public String gCargos()
{
    _cargos= _administradorCargos.leerlistCargo();
    return "gCargos";
}

public String gCargosInsert()
{
    //System.out.println(""+ _cargoInsert.getCargo());
    _administradorCargos.insertarCargo( _cargoInsert );
    _cargos= _administradorCargos.leerlistCargo();

    return "gCargosInsert";
}

//Metodos get y sets
public void setSession( Map sessionData )
{
    _sessionData = sessionData;
}

public List<AreaDTO> getAreas()
{
    return _areas;
}

public void setAreas( List<AreaDTO> areas )
{
    _areas = areas;
}

public void setAdministradorAreas( AdministradorArea administradorAreas )
{
    _administradorAreas = administradorAreas;
}

public List<CargoDTO> getCargos()
{
    return _cargos;
}

public void setCargos( List<CargoDTO> cargos )
{
    _cargos = cargos;
}

public void setAdministradorCargos( AdministradorCargo administradorCargos )
{
    _administradorCargos = administradorCargos;
}

public void setResp( ResponsableDTO resp )
{
    _resp = resp;
}

public ResponsableDTO getResp()
{
    return _resp;
}

public void setAreaS( String areaS )

```

```

{
    _areaS = areaS;
}

public void setCargoS( String cargoS )
{
    _cargoS = cargoS;
}

public void setAreaInsert( AreaDTO areaInsert )
{
    _areaInsert = areaInsert;
}

public AreaDTO getAreaInsert()
{
    return _areaInsert;
}

public CargoDTO getCargoInsert()
{
    return _cargoInsert;
}

public void setCargoInsert( CargoDTO cargoInsert )
{
    _cargoInsert = cargoInsert;
}

public void setAdministradorResponsables(
    AdministradorResponsables administradorResponsables )
{
    _administradorResponsables = administradorResponsables;
}

public List<ResponsableJoinDTO> getResponsables()
{
    return _responsables;
}

public void setResponsables( List<ResponsableJoinDTO> responsables )
{
    _responsables = responsables;
}

public ResponsableDTO getRespCh()
{
    return _respCh;
}

public void setRespCh( ResponsableDTO respCh )
{
    _respCh = respCh;
}

public int getKey()
{
    return _Key;
}

public void setKey( int key )
{
    _Key = key;
}

public ResponsableJoinDTO getResponsableOld()
{
    return _responsableOld;
}

public void setResponsableOld( ResponsableJoinDTO responsableOld )
{

```

```

    _responsableOld = responsableOld;
}

public AdministradorArea getAdministradorAreas()
{
    return _administradorAreas;
}

public AdministradorCargo getAdministradorCargos()
{
    return _administradorCargos;
}

public AdministradorResponsables getAdministradorResponsables()
{
    return _administradorResponsables;
}
}

```

## ElaboradosAction

```

package mx.uam.azc.cbi.sistemas.gestion.comunicacion.actions;

import mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data.*;
import mx.uam.azc.cbi.sistemas.gestion.usuarios.Data.*;
import mx.uam.azc.cbi.sistemas.gestion.usuarios.bussines.AdministradorUsuarios;
import mx.uam.azc.cbi.sistemas.gestion.comunicacion.bussines.*;
import mx.uam.azc.cbi.sistemas.gestion.comunicacion.reportes.create.CreateReportElaborados;
import net.sf.jasperreports.engine.*;
import net.sf.jasperreports.engine.data.JRBeanCollectionDataSource;
import net.sf.jasperreports.engine.design.JasperDesign;
import net.sf.jasperreports.engine.export.JRPdfExporter;
import net.sf.jasperreports.engine.xml.JRXmlLoader;
import org.apache.struts2.interceptor.SessionAware;
import com.opensymphony.xwork2.ActionSupport;
import java.io.InputStream;
import java.util.*;
import org.springframework.web.context.ContextLoader;

```

## ElaboradosAction

```

public class ElaboradosAction extends ActionSupport implements SessionAware
{
    private Map _sessionData;

    private AdministradorResponsables _administradorResponsables;
    private AdministradorUsuarios _administradorUsuarios;
    private AdministradorElaborados _administradorElaborados;
    private List<ResponsableJoinDTO> _responsables = new ArrayList<ResponsableJoinDTO>();
    private List<ProfesorDTO> _profesores = new ArrayList<ProfesorDTO>();
    private List<AdministrativoDTO> _administrativos = new ArrayList<AdministrativoDTO>();
    private int _tipoDest;
    private int _RmitenteKey;
    private int _DestinatarioKey;
    private long _documentoKey;
    private int usrID=0;
    private int respID=0;
    private int _Key;
    private ElaboradosDTO _elaborado = new ElaboradosDTO();
    private ElaboradosDTO _ElaboradoDel = new ElaboradosDTO();
    private UsuarioDTO _usuario = new UsuarioDTO();
    private ResponsableJoinDTO _remitente = new ResponsableJoinDTO();
    private UsuarioDTO _usrAct = new UsuarioDTO();
    private List<ElaboradosJoinDTO> _docElablist = new ArrayList<ElaboradosJoinDTO>();
    private AdministradorArea _administradorAreas;
}

```

```

private List<AreaDTO> _listaAreas = new ArrayList<AreaDTO>();
private ElaboradosJoinDTO _Elab = new ElaboradosJoinDTO();
private String _path;
private String _file;

public String Borrar()
{
    _ElaboradoDel = _administradorElaborados.leerElaborado( _Key );
    _documentoKey = _administradorElaborados.EliminaElaborado( _ElaboradoDel );
    _docElablist = _administradorElaborados.leerElaboradosJoin();

    return "listar";
}

public String Insertar() throws JRException
{
    int tmpDe = (Integer) _sessionData.get( "deId" );
    _elaborado.setDe( tmpDe );

    int tmpPara = (Integer) _sessionData.get( "paraId" );
    _elaborado.setPara( tmpPara );

    _documentoKey = _administradorElaborados.insertarElaborados( _elaborado );
    // System.out.println("Documento:"+ _documentoKey);

    //Generacion de Reporte
    _Elab= _administradorElaborados.leerElaboradoJoin( _documentoKey );
    _usrAct= (UsuarioDTO) _sessionData.get( "usuario" );
    _path =
ContextLoader.getCurrentWebApplicationContext().getServletContext().getRealPath("/Documentos/Elaborados
/Print/");
    System.out.println(_path);
    _listaAreas = _administradorAreas.leerlistArea();
    String _Date = _Elab.getFecha();
    String _ClaveDoc = _Elab.getNumDocumento();
    String _Destinatario = _Elab.getPara().getNombre() + " " + _Elab.getPara().getApaterno() + " " +
_Elab.getPara().getAmaterno();
    String _cargo = _Elab.getPara().getCargo().getCargo();
    String _asunto = _Elab.getAsunto();
    String _contenido = _Elab.getContenido();
    String _de = _Elab.getDe().getNombre() + " " + _Elab.getDe().getPaterno() + " " +
_Elab.getDe().getMaterno();
    String _puesto = _Elab.getDe().getPuesto().getpuesto();
    String _elabora = _usrAct.getLogin();
    _file = "Forma.pdf";
    setSystemProperties();
    ClassLoader loader = CreateReportElaborados.class.getClassLoader();
    InputStream istream = loader.getResourceAsStream(
"mx/uam/azc/cbi/sistemas/gestion/comunicacion/reportes/reporteElaborados.jrxml" );
    JasperDesign design = JRXmlLoader.Load( istream );
    JasperReport report = JasperCompileManager.compileReport( design );

    Map params = new HashMap();
    params.put( "cargo", cargo);
    params.put( "asunto", asunto);
    params.put( "anio", Date );
    params.put( "NoDoc", ClaveDoc );
    params.put( "contenido", contenido);
    params.put( "para", Destinatario);
    params.put( "de", de);
    params.put( "puesto", puesto);
    params.put( "elabora", elabora);

    JasperPrint jasperPrint = JasperFillManager.fillReport(report, params, new
JRBeanCollectionDataSource(_listaAreas));
    JRExporter exporter = new JRPdfExporter();
    exporter.setParameter(JRExporterParameter.JASPER_PRINT, jasperPrint);
    exporter.setParameter(JRExporterParameter.OUTPUT_FILE, new java.io.File(_path + "/" + _file));
    exporter.exportReport();
    return "insertar";
}

```

```

public String formInsertar()
{
    _responsables = _administradorResponsables.leerResponsableJoinAct();
    _profesores = _administradorUsuarios.leerProfesores();
    _administrativos = _administradorUsuarios.leerAdministrativos();
    return "formInsertar";
}

public String formInsertarUsr()
{
    // insertar y obtener lista de la sesion
    if(_DestinatarioKey>0){
        _sessionData.put( "deId", _DestinatarioKey );
        usrID = (Integer)_sessionData.get( "deId" );
    }

    if(_RmitenteKey>0){
        _sessionData.put( "paraId", _RmitenteKey );
        respID = (Integer)_sessionData.get( "paraId" );
    }

    _responsables = _administradorResponsables.leerResponsableJoinAct();
    _profesores = _administradorUsuarios.leerProfesores();
    _administrativos = _administradorUsuarios.leerAdministrativos();

    System.out.println(usrID);
    if(usrID >0){
        _usuario = _administradorUsuarios.leerUsuario( usrID );
    }

    if(respID >0){
        _remitente = _administradorResponsables.leerResponsable( respID );
    }
    //System.out.println(_usuario.getLogin());
    return "formInsertar";
}

public String formInsertarResp()
{
    usrID = (Integer)_sessionData.get( "deId" );

    if(_RmitenteKey>0){
        _sessionData.put( "paraId", _RmitenteKey );
        respID = (Integer)_sessionData.get( "paraId" );
    }

    _responsables = _administradorResponsables.leerResponsableJoinAct();
    _profesores = _administradorUsuarios.leerProfesores();
    _administrativos = _administradorUsuarios.leerAdministrativos();
    if(usrID >0){
        _usuario = _administradorUsuarios.leerUsuario( usrID );
    }
    if(respID >0){
        _remitente = _administradorResponsables.leerResponsable( respID );
    }
    return "formInsertar";
}

public String listar()
{
    _docElablist = _administradorElaborados.leerElaboradosJoin();
    return "listar";
}

// gets y sets
public AdministradorResponsables getAdministradorResponsables()
{
    return _administradorResponsables;
}

```

```

public void setAdministradorResponsables(
    AdministradorResponsables administradorResponsables )
{
    _administradorResponsables = administradorResponsables;
}

public AdministradorUsuarios getAdministradorUsuarios()
{
    return _administradorUsuarios;
}

public void setAdministradorUsuarios(
    AdministradorUsuarios administradorUsuarios )
{
    _administradorUsuarios = administradorUsuarios;
}

public List<ResponsableJoinDTO> getResponsables()
{
    return _responsables;
}

public void setResponsables( List<ResponsableJoinDTO> responsables )
{
    _responsables = responsables;
}

public List<ProfesorDTO> getProfesores()
{
    return _profesores;
}

public void setProfesores( List<ProfesorDTO> profesores )
{
    _profesores = profesores;
}

public List<AdministrativoDTO> getAdministrativos()
{
    return _administrativos;
}

public void setAdministrativos( List<AdministrativoDTO> administrativos )
{
    _administrativos = administrativos;
}

public int getRmitenteKey()
{
    return _RmitenteKey;
}

public void setRmitenteKey( int rmitenteKey )
{
    _RmitenteKey = rmitenteKey;
}

public int getDestinatarioKey()
{
    return _DestinatarioKey;
}

public void setDestinatarioKey( int destinatarioKey )
{
    _DestinatarioKey = destinatarioKey;
}

public int getTipoDest()
{
    return _tipoDest;
}

```

```

public void setTipoDest( int tipoDest )
{
    _tipoDest = tipoDest;
}

public UsuarioDTO getUsuario()
{
    return _usuario;
}

public void setUsuario( UsuarioDTO usuario )
{
    _usuario = usuario;
}

public ResponsableJoinDTO getRemitente()
{
    return _remitente;
}

public void setRemitente( ResponsableJoinDTO remitente )
{
    _remitente = remitente;
}

public long getDocumentoKey()
{
    return _documentoKey;
}

public void setDocumentoKey( long documentoKey )
{
    _documentoKey = documentoKey;
}

public AdministradorElaborados getAdministradorElaborados()
{
    return _administradorElaborados;
}

public void setAdministradorElaborados(
    AdministradorElaborados administradorElaborados )
{
    _administradorElaborados = administradorElaborados;
}

public ElaboradosDTO getElaborado()
{
    return _elaborado;
}

public void setElaborado( ElaboradosDTO elaborado )
{
    _elaborado = elaborado;
}

public List<ElaboradosJoinDTO> getDocElablist()
{
    return _docElablist;
}

public void setDocElablist( List<ElaboradosJoinDTO> docElablist )
{
    _docElablist = docElablist;
}

public void setSession( Map sessionData )
{
    _sessionData = sessionData;
}

```



```

}

public AdministradorArea getAdministradorAreas()
{
    return _administradorAreas;
}

public void setAdministradorAreas( AdministradorArea administradorAreas )
{
    _administradorAreas = administradorAreas;
}

public List<AreaDTO> getListaAreas()
{
    return _listaAreas;
}

public void setListaAreas( List<AreaDTO> listaAreas )
{
    _listaAreas = listaAreas;
}

public String getPath()
{
    return _path;
}

public void setPath( String path )
{
    _path = path;
}

public String getFile()
{
    return _file;
}

public void setFile( String file )
{
    _file = file;
}

private static void setSystemProperties()
{
    System.setProperty( "jasper.reports.compile.keep.java.file", "false" );
}

public UsuarioDTO getUsrAct()
{
    return _usrAct;
}

public void setUsrAct( UsuarioDTO usrAct )
{
    _usrAct = usrAct;
}

public int getKey()
{
    return _Key;
}

public void setKey( int key )
{
    _Key = key;
}

public ElaboradosDTO getElaboradoDel()
{
    return _ElaboradoDel;
}

```

```

}

public void setElaboradoDel( ElaboradosDTO elaboradoDel )
{
    _ElaboradoDel = elaboradoDel;
}
}

```

## RecibidosAction

```

package mx.uam.azc.cbi.sistemas.gestion.comunicacion.actions;

import mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data.*;
import mx.uam.azc.cbi.sistemas.gestion.usuarios.Data.*;
import mx.uam.azc.cbi.sistemas.gestion.usuarios.bussines.AdministradorUsuarios;
import mx.uam.azc.cbi.sistemas.gestion.comunicacion.bussines.*;
import org.apache.struts2.interceptor.SessionAware;
import com.opensymphony.xwork2.ActionSupport;
import java.util.*;

public class RecibidosAction extends ActionSupport implements SessionAware
{
    private Map _sessionData;
    private AdministradorResponsables _administradorResponsables;
    private AdministradorUsuarios _administradorUsuarios;
    private AdministradorRecibidos _administradorRecibidos;
    private List<ResponsableJoinDTO> _responsables = new ArrayList<ResponsableJoinDTO>();
    private List<ProfesorDTO> _profesores = new ArrayList<ProfesorDTO>();
    private List<AdministrativoDTO> _administrativos = new ArrayList<AdministrativoDTO>();
    private int _tipoDest;
    private int _RmitenteKey;
    private int _DestinatarioKey;
    private long _documentoKey;
    private int usrID=0;
    private int respID=0;
    private int _Key=0;
    private RecibidosDTO _recibido = new RecibidosDTO();
    private RecibidosDTO _recibidoDel = new RecibidosDTO();
    private UsuarioDTO _usuario = new UsuarioDTO();
    private ResponsableJoinDTO _remitente = new ResponsableJoinDTO();
    //private List<RecibidosJoinDTO> _recibidoslist = new ArrayList<RecibidosJoinDTO>();
    private List<RecibidosJoinDTO> _docReclist = new ArrayList<RecibidosJoinDTO>();

    public String Insertar()
    {
        int tmpDe = (Integer) _sessionData.get( "remitenteId" );
        _recibido.setDe( tmpDe );

        int tmpPara = (Integer) _sessionData.get( "destinatarioId" );
        _recibido.setPara( tmpPara );
        _documentoKey = _administradorRecibidos.insertarRecibido( _recibido );
        System.out.println("Documento:"+ _documentoKey);
        return "insertar";
    }

    public String Borrar()
    {
        _recibidoDel = _administradorRecibidos.leerRecibido( _Key );
        _documentoKey = _administradorRecibidos.EliminaRecibido( _recibidoDel );
        _docReclist = _administradorRecibidos.leerRecibidosJoin();

        return "listar";
    }

    public String formInsertar()
    {
        _responsables = _administradorResponsables.leerResponsableJoinAct();
        _profesores = _administradorUsuarios.leerProfesores();
        _administrativos = _administradorUsuarios.leerAdministrativos();
    }
}

```

```

        return "formInsertar";
    }

    public String formInsertarUsr()
    {
        // insertar y obtener lista de la sesion
        if(_DestinatarioKey>0){
            _sessionData.put( "destinatarioId", DestinatarioKey );
            usrID = (Integer)_sessionData.get( "destinatarioId" );
        }

        if(_RmitenteKey>0){
            _sessionData.put( "remiteId", RmitenteKey );
            respID = (Integer)_sessionData.get( "remiteId" );
        }

        _responsables = _administradorResponsables.leerResponsableJoinAct();
        _profesores = _administradorUsuarios.leerProfesores();
        _administrativos = _administradorUsuarios.leerAdministrativos();

        if(usrID >0){
            _usuario = _administradorUsuarios.leerUsuario( usrID );
        }

        if(respID >0){
            //_usuario = _administradorUsuarios.leerUsuario( usrID );
            _remite = _administradorResponsables.leerResponsable( respID );
        }
        return "formInsertar";
    }

    public String formInsertarResp()
    {
        usrID = (Integer)_sessionData.get( "destinatarioId" );

        if(_RmitenteKey>0){
            _sessionData.put( "remiteId", RmitenteKey );
            respID = (Integer)_sessionData.get( "remiteId" );
        }

        _responsables = _administradorResponsables.leerResponsableJoinAct();
        _profesores = _administradorUsuarios.leerProfesores();
        _administrativos = _administradorUsuarios.leerAdministrativos();

        if(usrID >0){
            _usuario = _administradorUsuarios.leerUsuario( usrID );
        }
        if(respID >0){
            //_usuario = _administradorUsuarios.leerUsuario( usrID );
            _remite = _administradorResponsables.leerResponsable( respID );
        }

        return "formInsertar";
    }

    public String listar()
    {
        _docReclist = _administradorRecibidos.leerRecibidosJoin();
        return "listar";
    }

    // gets y sets

    public AdministradorResponsables getAdministradorResponsables()
    {
        return _administradorResponsables;
    }

```

```

public void setAdministradorResponsables(
    AdministradorResponsables administradorResponsables )
{
    _administradorResponsables = administradorResponsables;
}

public AdministradorUsuarios getAdministradorUsuarios()
{
    return _administradorUsuarios;
}

public void setAdministradorUsuarios(
    AdministradorUsuarios administradorUsuarios )
{
    _administradorUsuarios = administradorUsuarios;
}

public AdministradorRecibidos getAdministradorRecibidos()
{
    return _administradorRecibidos;
}

public void setAdministradorRecibidos(
    AdministradorRecibidos administradorRecibidos )
{
    _administradorRecibidos = administradorRecibidos;
}

public List<ResponsableJoinDTO> getResponsables()
{
    return _responsables;
}

public void setResponsables( List<ResponsableJoinDTO> responsables )
{
    _responsables = responsables;
}

public List<ProfesorDTO> getProfesores()
{
    return _profesores;
}

public void setProfesores( List<ProfesorDTO> profesores )
{
    _profesores = profesores;
}

public List<AdministrativoDTO> getAdministrativos()
{
    return _administrativos;
}

public void setAdministrativos( List<AdministrativoDTO> administrativos )
{
    _administrativos = administrativos;
}

public RecibidosDTO getRecibido()
{

```

```

    return _recibido;
}

public void setRecibido( RecibidosDTO recibido )
{
    _recibido = recibido;
}

public int getRmitenteKey()
{
    return _RmitenteKey;
}

public void setRmitenteKey( int rmitenteKey )
{
    _RmitenteKey = rmitenteKey;
}

public int getDestinatarioKey()
{
    return _DestinatarioKey;
}

public void setDestinatarioKey( int destinatarioKey )
{
    _DestinatarioKey = destinatarioKey;
}

public int getTipoDest()
{
    return _tipoDest;
}

public void setTipoDest( int tipoDest )
{
    _tipoDest = tipoDest;
}

public UsuarioDTO getUsuario()
{
    return _usuario;
}

public void setUsuario( UsuarioDTO usuario )
{
    _usuario = usuario;
}

public ResponsableJoinDTO getRemitente()
{
    return _remitente;
}

public void setRemitente( ResponsableJoinDTO remitente )
{
    _remitente = remitente;
}

public long getDocumentoKey()
{
    return _documentoKey;
}

```

```

public void setDocumentoKey( long documentoKey )
{
    _documentoKey = documentoKey;
}

public List<RecibidosJoinDTO> getDocReclist()
{
    return _docReclist;
}

public void setDocReclist( List<RecibidosJoinDTO> docReclist )
{
    _docReclist = docReclist;
}

public int getKey()
{
    return _Key;
}

public void setKey( int key )
{
    _Key = key;
}

public RecibidosDTO getRecibidoDel()
{
    return _recibidoDel;
}

public void setRecibidoDel( RecibidosDTO recibidoDel )
{
    _recibidoDel = recibidoDel;
}

public void setSession( Map sessionData )
{
    _sessionData = sessionData;
}
}

```

## FileUploadAction

```

package mx.uam.azc.cbi.sistemas.gestion.comunicacion.actions;

```

```

import mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data.*;
import mx.uam.azc.cbi.sistemas.gestion.comunicacion.bussines.AdministradorElaborados;
import mx.uam.azc.cbi.sistemas.gestion.comunicacion.bussines.AdministradorRecibidos;
import java.io.File;
import java.io.IOException;
import java.util.*;
import com.opensymphony.xwork2.ActionSupport;
import org.apache.commons.io.FileUtils;
import org.apache.struts2.interceptor.SessionAware;
import org.springframework.web.context.ContextLoader;

```

```

public class FileUploadAction extends ActionSupport implements SessionAware
{
    private Map _sessionData;
    private File fileUpload;
    private String fileUploadContentType;
    private String fileUploadFileName;
}

```

```

private static final String UPLOAD_FOLDER = "upload";
private long _Key;
private RecibidosDTO _recibido = new RecibidosDTO();
private List<RecibidosJoinDTO> _docRecllist = new ArrayList<RecibidosJoinDTO>();
private AdministradorRecibidos _administradorRecibidos;
private ElaboradosDTO _Elaborado = new ElaboradosDTO();
private List<ElaboradosJoinDTO> _docElablist = new ArrayList<ElaboradosJoinDTO>();
private AdministradorElaborados _administradorElaborados;

public String subir()
{
    String _Path =
ContextLoader.getCurrentWebApplicationContext().getServletContext().getRealPath("/Documentos/Recibidos"
);
    File theFile = new File(_Path + "/" + _Key+ "_" + fileUploadFileName);
    try
    {
        FileUtils.copyFile(fileUpload, theFile);
    }
    catch ( IOException e )
    {
        e.printStackTrace();
    }
    String directorio = theFile.getAbsolutePath();
    String documento = theFile.getName();

    _recibido = _administradorRecibidos.leerRecibido( _Key );
    _recibido.setDocumento( documento );
    _recibido.setDirectorio( directorio );
    _administradorRecibidos.ModificaRecibido( _recibido );
    _docRecllist = _administradorRecibidos.leerRecibidosJoin();

    return "uploadResult";
}

public String subirElaborado()
{
    String _Path =
ContextLoader.getCurrentWebApplicationContext().getServletContext().getRealPath("/Documentos/Elaborados
");
    File theFile = new File( _Path + "/" + _Key+ "_" + fileUploadFileName);
    try
    {
        FileUtils.copyFile(fileUpload, theFile);
    }
    catch ( IOException e )
    {
        e.printStackTrace();
    }
    String directorio = theFile.getAbsolutePath();
    String documento = theFile.getName();
    _Elaborado = _administradorElaborados.leerElaborado( _Key );
    _Elaborado.setDirectorio( directorio );
    _Elaborado.setDocumento( documento );
    _administradorElaborados.ModificaElaborado( _Elaborado );
    _docElablist = _administradorElaborados.leerElaboradosJoin();

    return "uploadResult2";
}

public String formUpload()
{
    return "fileupload";
}

//gets y sets

public File getFileUpload()
{

```

```

        return fileUpload;
    }

    public void setFileUpload( File fileUpload )
    {
        this.fileUpload = fileUpload;
    }

    public String getFileUploadContentType()
    {
        return fileUploadContentType;
    }

    public void setFileUploadContentType( String fileUploadContentType )
    {
        this.fileUploadContentType = fileUploadContentType;
    }

    public String getFileUploadFileName()
    {
        return fileUploadFileName;
    }

    public void setFileUploadFileName( String fileUploadFileName )
    {
        this.fileUploadFileName = fileUploadFileName;
    }

    public long getKey()
    {
        return _key;
    }

    public void setKey( long key )
    {
        _key = key;
    }

    public RecibidosDTO getRecibido()
    {
        return _recibido;
    }

    public void setRecibido( RecibidosDTO recibido )
    {
        _recibido = recibido;
    }

    public AdministradorRecibidos getAdministradorRecibidos()
    {
        return _administradorRecibidos;
    }

    public void setAdministradorRecibidos(
        AdministradorRecibidos administradorRecibidos )
    {
        _administradorRecibidos = administradorRecibidos;
    }

    public List<RecibidosJoinDTO> getDocRecllist()
    {
        return _docRecllist;
    }

    public void setDocRecllist( List<RecibidosJoinDTO> docRecllist )
    {
        _docRecllist = docRecllist;
    }

```



```

public ElaboradosDTO getElaborado()
{
    return _Elaborado;
}

public void setElaborado( ElaboradosDTO elaborado )
{
    _Elaborado = elaborado;
}

public List<ElaboradosJoinDTO> getDocElablist()
{
    return _docElablist;
}

public void setDocElablist( List<ElaboradosJoinDTO> docElablist )
{
    _docElablist = docElablist;
}

public AdministradorElaborados getAdministradorElaborados()
{
    return _administradorElaborados;
}

public void setAdministradorElaborados(
    AdministradorElaborados administradorElaborados )
{
    _administradorElaborados = administradorElaborados;
}

public void setSession( Map sessionData )
{
    _sessionData = sessionData;
}
}

```

## mx.uam.azc.cbi.sistemas.gestion.comunicacion.bussines

### AdministradorArea

```

package mx.uam.azc.cbi.sistemas.gestion.comunicacion.bussines;

```

```

import mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data.AreaDTO;
import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;
import java.util.List;

```

```

@Transactional(rollbackFor = Exception.class)

```

```

public interface AdministradorArea
{

```

```

    @Transactional(propagation = Propagation.SUPPORTS, readOnly = true)
    public List<AreaDTO> leerlistArea();

```

```

    @Transactional( propagation=Propagation.SUPPORTS, readOnly = true )
    public AreaDTO leerAreaN(String area);

```

```

    public int insertarArea( AreaDTO area );

```

```

}

```

## AdministradorCargo

```
package mx.uam.azc.cbi.sistemas.gestion.comunicacion.bussines;

import mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data.*;
import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;
import java.util.List;
import java.util.Map;

@Transactional(rollbackFor = Exception.class)
public interface AdministradorCargo
{

    @Transactional(propagation = Propagation.SUPPORTS, readOnly = true)
    public List<CargoDTO> leerlistCargo();

    @Transactional( propagation=Propagation.SUPPORTS, readOnly = true )
    public CargoDTO leerCargoN(String cargo);

    public int insertarCargo( CargoDTO cargo );
}

```

```
package mx.uam.azc.cbi.sistemas.gestion.comunicacion.bussines;

import mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data.ElaboradosDTO;
import mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data.ElaboradosJoinDTO;
import org.springframework.transaction.annotation.Propagation;

```

## AdministradorElaborados

```
import org.springframework.transaction.annotation.Transactional;
import java.util.List;
import java.util.Map;

@Transactional(rollbackFor = Exception.class)
public interface AdministradorElaborados
{
    @Transactional(propagation = Propagation.SUPPORTS, readOnly = true)
    public List<ElaboradosJoinDTO> leerElaboradosJoin();

    @Transactional(propagation = Propagation.SUPPORTS, readOnly = true)
    public ElaboradosDTO leerElaborado(long idelaborado);

    @Transactional(propagation = Propagation.SUPPORTS, readOnly = true)
    public ElaboradosJoinDTO leerElaboradoJoin(long idelaborado);

    public long insertarElaborados( ElaboradosDTO elaborado );

    public long ModificaElaborado( ElaboradosDTO elaboradoCh );

    public long EliminaElaborado( ElaboradosDTO elaboradoDel );
}

```

## AdministradorRecibidos

```
package mx.uam.azc.cbi.sistemas.gestion.comunicacion.bussines;

import mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data.*;
import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;
import java.util.List;

```

```

import java.util.Map;

@Transactional(rollbackFor = Exception.class)
public interface AdministradorRecibidos
{
    @Transactional(propagation = Propagation.SUPPORTS, readOnly = true)
    public List<RecibidosJoinDTO> leerRecibidosJoin();

    @Transactional(propagation = Propagation.SUPPORTS, readOnly = true)
    public RecibidosDTO leerRecibido(long idrecibidos);

    public long insertarRecibido( RecibidosDTO recibido );

    public long ModificaRecibido( RecibidosDTO recibidoCh );

    public long EliminaRecibido( RecibidosDTO recibidoDel );
}

```

## AdministradorResponsables

```

package mx.uam.azc.cbi.sistemas.gestion.comunicacion.bussines;

import mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data.*;
import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;
import java.util.List;
import java.util.Map;

@Transactional(rollbackFor = Exception.class)
public interface AdministradorResponsables
{
    @Transactional(propagation = Propagation.SUPPORTS, readOnly = true)
    public List<ResponsableJoinDTO> leerResponsableJoin();

    @Transactional(propagation = Propagation.SUPPORTS, readOnly = true)
    public ResponsableJoinDTO leerResponsable( int id );

    @Transactional(propagation = Propagation.SUPPORTS, readOnly = true)
    public ResponsableDTO leerResponsableNN( int id );

    @Transactional(propagation = Propagation.SUPPORTS, readOnly = true)
    public List<ResponsableJoinDTO> leerResponsableJoinAct();

    public long insertarResponsable( ResponsableDTO responsable );

    public int ModificaResponsable( ResponsableDTO responsableCh );
}

```

## mx.uam.azc.cbi.sistemas.gestion.comunicacion.bussines.local

### AdministradorAreaImpl

```

package mx.uam.azc.cbi.sistemas.gestion.comunicacion.bussines.local;

import mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data.AreaDTO;
import mx.uam.azc.cbi.sistemas.gestion.comunicacion.bussines.AdministradorArea;

import java.util.List;
import java.util.Map;
import org.hibernate.*;
import java.util.*;

```

```

public class AdministradorAreaImpl implements AdministradorArea
{
    private SessionFactory _sessionFactory;

    public List<AreaDTO> leerlistArea()
    {
        Session session = getSession();
        Query query = session.getNamedQuery( "leer_Areas" );
        return query.list();
    }

    public AreaDTO leerAreaN( String area )
    {
        Session session = _sessionFactory.getCurrentSession();
        Query query = session.getNamedQuery( "leer_area" );
        query.setString( 0, area );
        List<AreaDTO> lista = query.list();
        return lista.isEmpty() ? null : lista.get( 0 );
    }

    public int insertarArea( AreaDTO area )
    {
        Session session = getSession();
        session.save( area );
        return area.getIdarea();
    }

    private Session getSession()
    {
        return _sessionFactory.getCurrentSession();
    }

    public SessionFactory getSessionFactory()
    {
        return _sessionFactory;
    }

    public void setSessionFactory( SessionFactory sessionFactory )
    {
        _sessionFactory = sessionFactory;
    }
}

```

## AdministradorCargoImpl

```

package mx.uam.azc.cbi.sistemas.gestion.comunicacion.bussines.local;

import mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data.AreaDTO;
import mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data.CargoDTO;
import mx.uam.azc.cbi.sistemas.gestion.comunicacion.bussines.AdministradorCargo;
import java.util.List;
import java.util.Map;
import org.hibernate.*;
import java.util.*;

public class AdministradorCargoImpl implements AdministradorCargo
{
    private SessionFactory _sessionFactory;

    public List<CargoDTO> leerlistCargo()
    {
        Session session = getSession();
        Query query = session.getNamedQuery( "leer_Cargos" );
    }
}

```

```

        return query.list();
    }

    public CargoDTO leerCargoN( String cargo )
    {
        Session session = _sessionFactory.getCurrentSession();
        Query query = session.getNamedQuery( "leer_cargo" );
        query.setString( 0, cargo );
        List<CargoDTO> lista = query.list();
        return lista.isEmpty() ? null : lista.get( 0 );
    }

    public int insertarCargo( CargoDTO cargo )
    {
        Session session = getSession();
        session.save( cargo );
        return cargo.getIdcargo();
    }

    private Session getSession()
    {
        return _sessionFactory.getCurrentSession();
    }

    public SessionFactory getSessionFactory()
    {
        return _sessionFactory;
    }

    public void setSessionFactory( SessionFactory sessionFactory )
    {
        _sessionFactory = sessionFactory;
    }
}

```

## AdministradorElaboradosImpl

```

package mx.uam.azc.cbi.sistemas.gestion.comunicacion.bussines.local;

import mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data.*;
import mx.uam.azc.cbi.sistemas.gestion.comunicacion.bussines.AdministradorElaborados;
import java.util.List;
import java.util.Map;
import org.hibernate.*;
import java.util.*;

public class AdministradorElaboradosImpl implements AdministradorElaborados
{
    private SessionFactory _sessionFactory;

    public long insertarElaborados( ElaboradosDTO elaborado )
    {
        Session session = getSession();
        session.save( elaborado );
        return elaborado.getIdelaborados();
    }
}

```

```

public List<ElaboradosJoinDTO> leerElaboradosJoin()
{
    Session session = _sessionFactory.getCurrentSession();
    Query query = session.getNamedQuery( "leer_Elaborados" );
    return query.list();
}

public ElaboradosDTO leerElaborado(long idelaborado)
{
    Session session = getSession();
    return ( ElaboradosDTO )session.get( ElaboradosDTO.class, idelaborado );
}

public long ModificaElaborado( ElaboradosDTO elaboradoCh )
{
    Session session = getSession();
    session.update( elaboradoCh );
    return elaboradoCh.getIdelaborados();
}

public long EliminaElaborado( ElaboradosDTO elaboradoDel )
{
    Session session = getSession();
    session.delete( elaboradoDel );
    return elaboradoDel.getIdelaborados();
}

private Session getSession()
{
    return _sessionFactory.getCurrentSession();
}

public SessionFactory getSessionFactory()
{
    return _sessionFactory;
}

public void setSessionFactory( SessionFactory sessionFactory )
{
    _sessionFactory = sessionFactory;
}

public ElaboradosJoinDTO leerElaboradoJoin(long idelaborado)
{
    Session session = getSession();
    return ( ElaboradosJoinDTO )session.get( ElaboradosJoinDTO.class, idelaborado );
}
}

```

## AdministradorRecibidosImpl

```

package mx.uam.azc.cbi.sistemas.gestion.comunicacion.bussines.local;

import mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data.*;
import mx.uam.azc.cbi.sistemas.gestion.comunicacion.bussines.AdministradorRecibidos;

```

```

import mx.uam.azc.cbi.sistemas.gestion.usuarios.Data.UsuarioDTO;
import java.util.List;
import java.util.Map;
import org.hibernate.*;
import java.util.*;

public class AdministradorRecibidosImpl implements AdministradorRecibidos
{
    private SessionFactory _sessionFactory;

    public List<RecibidosJoinDTO> leerRecibidosF(String filter)
    {
        Session session = getSession();
        Query query = session.getNamedQuery( "leer_Recibidos_filtro" );
        query.setString( "nombre", "%" + filter + "%" );
        return query.list();
    }

    public long insertarRecibido( RecibidosDTO recibido )
    {
        Session session = getSession();
        session.save( recibido );
        return recibido.getIdrecibidos();
    }

    public List<RecibidosJoinDTO> leerRecibidosJoin()
    {
        Session session = _sessionFactory.getCurrentSession();
        Query query = session.getNamedQuery( "leer_Recibidos" );
        return query.list();
    }

    public RecibidosDTO leerRecibido(long idrecibidos)
    {
        Session session = getSession();
        return ( RecibidosDTO )session.get( RecibidosDTO.class, idrecibidos );
    }

    public long ModificaRecibido( RecibidosDTO recibidoCh )
    {
        Session session = getSession();
        session.update( recibidoCh );
        return recibidoCh.getIdrecibidos();
    }

    public long EliminaRecibido( RecibidosDTO recibidoDel )
    {
        Session session = getSession();
        session.delete( recibidoDel );
        return recibidoDel.getIdrecibidos();
    }

    private Session getSession()
    {
        return _sessionFactory.getCurrentSession();
    }

    public SessionFactory getSessionFactory()
    {
        return _sessionFactory;
    }

    public void setSessionFactory( SessionFactory sessionFactory )
    {

```

```

    _sessionFactory = sessionFactory;
}
}

```

## AdministradorResponsablesImpl

```

package mx.uam.azc.cbi.sistemas.gestion.comunicacion.bussines.local;

import mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data.*;
import mx.uam.azc.cbi.sistemas.gestion.comunicacion.bussines.AdministradorResponsables;
import java.util.List;
import java.util.Map;
import org.hibernate.*;
import java.util.*;

public class AdministradorResponsablesImpl implements AdministradorResponsables
{
    private SessionFactory _sessionFactory;

    public List<ResponsableJoinDTO> leerResponsableJoinAct()
    {
        int lra = 2; // lista de los responsables que estan activos

        Session session = _sessionFactory.getCurrentSession();
        Query query = session.getNamedQuery( "leer_Responsable_Activos" );
        query.setInteger( 0, lra );
        List<ResponsableJoinDTO> lista = query.list();
        return lista.isEmpty() ? null : lista;
    }

    public long insertarResponsable( ResponsableDTO responsable )
    {
        Session session = getSession();
        session.save( responsable );
        return responsable.getIdresponsable();
    }

    public ResponsableJoinDTO leerResponsable( int id )
    {
        Session session = getSession();
        return ( ResponsableJoinDTO )session.get( ResponsableJoinDTO.class, id );
    }

    public List<ResponsableJoinDTO> leerResponsableJoin()
    {
        Session session = _sessionFactory.getCurrentSession();
        Query query = session.getNamedQuery( "leer_Reponsables" );
        return query.list();
    }

    public ResponsableDTO leerResponsableNN( int id )
    {
        Session session = getSession();
        return ( ResponsableDTO )session.get( ResponsableDTO.class, id );
    }
}

```



```

public int ModificaResponsable( ResponsableDTO responsableCh )
{
    Session session = getSession();
    session.update( responsableCh );
    return responsableCh.getIdresponsable();
}

private Session getSession()
{
    return _sessionFactory.getCurrentSession();
}

public SessionFactory getSessionFactory()
{
    return _sessionFactory;
}

public void setSessionFactory( SessionFactory sessionFactory )
{
    _sessionFactory = sessionFactory;
}
}

```

## mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data

### AreaDTO

```

package mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data;

import java.io.Serializable;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.*;

@Entity
@Table(name = "area")
public class AreaDTO implements Serializable
{

    private int _idarea;
    private String _area;

    @Id @Column( name="id_area" ) @GeneratedValue( strategy=IDENTITY )
    public int getIdarea()
    {
        return _idarea;
    }

    @Column( name="area" )
    public String getArea()
    {
        return _area;
    }

    public void setIdarea( int idarea )
    {
        _idarea = idarea;
    }
}

```

```

    public void setArea( String area )
    {
        _area = area;
    }
}

```

## CargoDTO

```

package mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data;

```

```

import java.io.Serializable;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.*;

```

```

@Entity
@Table(name = "cargo")
public class CargoDTO implements Serializable
{
    private int _idcargo;
    private String _cargo;

    @Id @Column( name="id_cargo" ) @GeneratedValue( strategy=IDENTITY )
    public int getIdcargo()
    {
        return _idcargo;
    }

    @Column( name="cargo" )
    public String getCargo()
    {
        return _cargo;
    }

    public void setIdcargo( int idcargo )
    {
        _idcargo = idcargo;
    }

    public void setCargo( String cargo )
    {
        _cargo = cargo;
    }
}

```

## ElaboradosDTO

```

package mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data;

```

```

import java.io.Serializable;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.*;

@Entity
@Table(name = "elaborados")
public class ElaboradosDTO implements Serializable
{

    private long _idelaborados;
    private String _fecha;
    private String _numDocumento;
    private int _para;
    private String _asunto;
    private String _contenido;
    private int _tipo;
    private int de;
    private String _anio;
    private String _directorio;
    private String _documento;

    @Id
    @Column(name = "id_elaborados")
    @GeneratedValue(strategy = IDENTITY)
    public long getIdelaborados()
    {
        return _idelaborados;
    }

    @Column(name = "para")
    public int getPara()
    {
        return _para;
    }

    @Column(name = "fecha")
    public String getFecha()
    {
        return _fecha;
    }

    @Column(name = "num_documento")
    public String getNumDocumento()
    {
        return _numDocumento;
    }

    @Column(name = "tipo")
    public int getTipo()
    {
        return _tipo;
    }

    @Column(name = "asunto")
    public String getAsunto()
    {
        return _asunto;
    }

    @Column(name = "contenido")
    public String getContenido()
    {
        return _contenido;
    }

    @Column(name = "de")
    public int getDe()
    {

```

```

    return de;
}

@Column(name = "anio")
public String getAnio()
{
    return _anio;
}

@Column(name = "directorio")
public String getDirectorio()
{
    return _directorio;
}

@Column(name = "documento")
public String getDocumento()
{
    return _documento;
}

public void setIdelaborados( long idelaborados )
{
    _idelaborados = idelaborados;
}

public void setPara( int para )
{
    _para = para;
}

public void setFecha( String fecha )
{
    _fecha = fecha;
}

public void setNumDocumento( String numDocumento )
{
    _numDocumento = numDocumento;
}

public void setTipo( int tipo )
{
    _tipo = tipo;
}

public void setAsunto( String asunto )
{
    _asunto = asunto;
}

public void setDe( int de )
{
    this.de = de;
}

public void setAnio( String anio )
{
    _anio = anio;
}

public void setDirectorio( String directorio )
{
    _directorio = directorio;
}

public void setContenido( String contenido )
{
    _contenido = contenido;
}

```

```

public void setDocumento( String documento )
{
    _documento = documento;
}
}

```

## ElaboradosJoinDTO

```

package mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data;

import mx.uam.azc.cbi.sistemas.gestion.usuarios.Data.UsuarioDTO;
import java.io.Serializable;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.*;

@Entity
@Table(name = "elaborados")
public class ElaboradosJoinDTO implements Serializable
{

    private long _idelaborados;
    private String _fecha;
    private String _numDocumento;
    private ResponsableJoinDTO _para;
    private String _asunto;
    private String _contenido;
    private int _tipo;
    private UsuarioDTO de;
    private String _anio;
    private String _directorio;
    private String _documento;

    @Id
    @Column(name = "id_elaborados")
    @GeneratedValue(strategy = IDENTITY)
    public long getIdelaborados()
    {
        return _idelaborados;
    }

    @OneToOne
    @JoinColumn(name="para")
    public ResponsableJoinDTO getPara()
    {
        return _para;
    }

    @Column(name = "fecha")
    public String getFecha()
    {
        return _fecha;
    }

    @Column(name = "num_documento")
    public String getNumDocumento()
    {
        return _numDocumento;
    }

    @Column(name = "tipo")
    public int getTipo()

```

```

{
    return _tipo;
}

@Column(name = "asunto")
public String getAsunto()
{
    return _asunto;
}

@Column(name = "contenido")
public String getContenido()
{
    return _contenido;
}

@OneToOne
@JoinColumn(name="de")
public UsuarioDTO getDe()
{
    return de;
}

@Column(name = "anio")
public String getAnio()
{
    return _anio;
}

@Column(name = "directorio")
public String getDirectorio()
{
    return _directorio;
}

@Column(name = "documento")
public String getDocumento()
{
    return _documento;
}

public void setIdelaborados( long idelaborados )
{
    _idelaborados = idelaborados;
}

public void setPara( ResponsableJoinDTO para )
{
    _para = para;
}

public void setFecha( String fecha )
{
    _fecha = fecha;
}

public void setNumDocumento( String numDocumento )
{
    _numDocumento = numDocumento;
}

public void setTipo( int tipo )
{
    _tipo = tipo;
}

public void setAsunto( String asunto )
{
    _asunto = asunto;
}

```

```

public void setDe( UsuarioDTO de )
{
    this.de = de;
}

public void setAnio( String anio )
{
    _anio = anio;
}

public void setDirectorio( String directorio )
{
    _directorio = directorio;
}

public void setContenido( String contenido )
{
    _contenido = contenido;
}

public void setDocumento( String documento )
{
    _documento = documento;
}
}

```

## RecibidosDTO

```

package mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data;

```

```

import java.io.Serializable;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.*;

```

```

@Entity
@Table(name = "recibidos")
public class RecibidosDTO implements Serializable
{

```

```

    private long _idrecibidos;
    private int _para;
    private String _fecha;
    private String _numDocumento;
    private int _tipo;
    private String _asunto;
    private int de;
    private String _anio;
    private String _directorio;
    private String _documento;

```

```

@Id
@Column(name = "id_recibidos")
@GeneratedValue(strategy = IDENTITY)
public long getIdrecibidos()
{
    return _idrecibidos;
}

```

```

@Column(name = "para")
public int getPara()
{
    return _para;
}

```

```

@Column(name = "fecha")
public String getFecha()
{
    return _fecha;
}

@Column(name = "num_documento")
public String getNumDocumento()
{
    return _numDocumento;
}

@Column(name = "tipo")
public int getTipo()
{
    return _tipo;
}

@Column(name = "asunto")
public String getAsunto()
{
    return _asunto;
}

@Column(name = "de")
public int getDe()
{
    return de;
}

@Column(name = "anio")
public String getAnio()
{
    return _anio;
}

@Column(name = "directorio")
public String getDirectorio()
{
    return _directorio;
}

@Column(name = "documento")
public String getDocumento()
{
    return _documento;
}

public void setIdrecibidos( long idrecibidos )
{
    _idrecibidos = idrecibidos;
}

public void setPara( int para )
{
    _para = para;
}

public void setFecha( String fecha )
{
    _fecha = fecha;
}

public void setNumDocumento( String numDocumento )
{
    _numDocumento = numDocumento;
}

public void setTipo( int tipo )
{
    _tipo = tipo;
}

```



```

}

public void setAsunto( String asunto )
{
    _asunto = asunto;
}

public void setDe( int de )
{
    this.de = de;
}

public void setAnio( String anio )
{
    _anio = anio;
}

public void setDirectorio( String directorio )
{
    _directorio = directorio;
}

public void setDocumento( String documento )
{
    _documento = documento;
}
}

```

## RecibidosJoinDTO

```

package mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data;

import mx.uam.azc.cbi.sistemas.gestion.usuarios.Data.UsuarioDTO;
import java.io.Serializable;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.*;

@Entity
@Table(name = "recibidos")
public class RecibidosJoinDTO implements Serializable
{

    private long _idrecibidos;
    private UsuarioDTO _para;
    private String _fecha;
    private String _numDocumento;
    private int _tipo;
    private String _asunto;
    private ResponsableJoinDTO de;
    private String _anio;
    private String _directorio;
    private String _documento;

    @Id
    @Column(name = "id_recibidos")
    @GeneratedValue(strategy = IDENTITY)
    public long getIdrecibidos()
    {

```

```

    return _idrecibidos;
}

@OneToOne
@JoinColumn(name="para")
public UsuarioDTO getPara()
{
    return _para;
}

@Column(name = "fecha")
public String getFecha()
{
    return _fecha;
}

@Column(name = "num_documento")
public String getNumDocumento()
{
    return _numDocumento;
}

@Column(name = "tipo")
public int getTipo()
{
    return _tipo;
}

@Column(name = "asunto")
public String getAsunto()
{
    return _asunto;
}

@OneToOne
@JoinColumn(name="de")
public ResponsableJoinDTO getDe()
{
    return de;
}

@Column(name = "anio")
public String getAnio()
{
    return _anio;
}

@Column(name = "directorio")
public String getDirectorio()
{
    return _directorio;
}

@Column(name = "documento")
public String getDocumento()
{
    return _documento;
}

public void setIdrecibidos( long idrecibidos )
{
    _idrecibidos = idrecibidos;
}

public void setFecha( String fecha )
{
    _fecha = fecha;
}

```

```

public void setNumDocumento( String numDocumento )
{
    _numDocumento = numDocumento;
}

public void setTipo( int tipo )
{
    _tipo = tipo;
}

public void setAsunto( String asunto )
{
    _asunto = asunto;
}

public void setAnio( String anio )
{
    _anio = anio;
}

public void setDirectorio( String directorio )
{
    _directorio = directorio;
}

public void setDocumento( String documento )
{
    _documento = documento;
}

public void setPara( UsuarioDTO para )
{
    _para = para;
}

public void setDe( ResponsableJoinDTO de )
{
    this.de = de;
}
}

```

## ResponsableDTO

```

package mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data;

```

```

import java.io.Serializable;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.*;

```

```

@Entity
@Table(name = "responsables")
public class ResponsableDTO implements Serializable
{
    private int _idresponsable;

    private String _nombre;

    private String _apaterno;

    private String _amaterno;

    private int _area;

    private int _cargo;
}

```

```

private int _estado;

private String _correo;

private String _telefono;

private String _ubicacion;

@Id
@Column(name = "id_responsables")
@GeneratedValue(strategy = IDENTITY)
public int getIdresponsable()
{
    return _idresponsable;
}

@Column(name = "nombre")
public String getNombre()
{
    return _nombre;
}

@Column(name = "a_paterno")
public String getApaterno()
{
    return _apaterno;
}

@Column(name = "a_materno")
public String getAmaterno()
{
    return _amaterno;
}

@Column(name = "area")
public int getArea()
{
    return _area;
}

@Column(name = "cargo")
public int getCargo()
{
    return _cargo;
}

@Column(name = "estado")
public int getEstado()
{
    return _estado;
}

@Column(name = "email")
public String getCorreo()
{
    return _correo;
}

@Column(name = "telefono")
public String getTelefono()
{
    return _telefono;
}

@Column(name = "ubicacion")
public String getUbicacion()
{
    return _ubicacion;
}

public void setIdresponsable( int idresponsable )
{
    _idresponsable = idresponsable;
}

```

```

public void setNombre( String nombre )
{
    _nombre = nombre;
}

public void setApaterno( String apaterno )
{
    _apaterno = apaterno;
}

public void setAmaterno( String amaterno )
{
    _amaterno = amaterno;
}

public void setArea( int area )
{
    _area = area;
}

public void setCargo( int cargo )
{
    _cargo = cargo;
}

public void setEstado( int estado )
{
    _estado = estado;
}

public void setCorreo( String correo )
{
    _correo = correo;
}

public void setTelefono( String telefono )
{
    _telefono = telefono;
}

public void setUbicacion( String ubicacion )
{
    _ubicacion = ubicacion;
}
}

```

## ResponsableJoinDTO

```

package mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data;

import java.io.Serializable;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.*;

@Entity
@Table(name = "responsables")
public class ResponsableJoinDTO implements Serializable
{
    private int _idresponsable;

    private String _nombre;

    private String _apaterno;

```

```

private String _amaterno;

private AreaDTO _area;

private CargoDTO _cargo;

private int _estado;

private String _correo;

private String _telefono;

private String _ubicacion;

@Id
@Column(name = "id_responsables")
@GeneratedValue(strategy = IDENTITY)
public int getIdresponsable()
{
    return _idresponsable;
}

@OneToOne
@JoinColumn(name="area")
public AreaDTO getArea()
{
    return _area;
}

@OneToOne
@JoinColumn(name="cargo")
public CargoDTO getCargo()
{
    return _cargo;
}

@Column(name = "nombre")
public String getNombre()
{
    return _nombre;
}

@Column(name = "a_paterno")
public String getApaterno()
{
    return _apaterno;
}

@Column(name = "a_materno")
public String getAmaterno()
{
    return _amaterno;
}

@Column(name = "estado")
public int getEstado()
{
    return _estado;
}

@Column(name = "email")
public String getCorreo()
{
    return _correo;
}

@Column(name = "telefono")
public String getTelefono()
{
    return _telefono;
}

@Column(name = "ubicacion")

```

```

public String getUbicacion()
{
    return _ubicacion;
}

public void setIdresponsable( int idresponsable )
{
    _idresponsable = idresponsable;
}

public void setNombre( String nombre )
{
    _nombre = nombre;
}

public void setApaterno( String apaterno )
{
    _apaterno = apaterno;
}

public void setAmaterno( String amaterno )
{
    _amaterno = amaterno;
}

public void setArea( AreaDTO area )
{
    _area = area;
}

public void setCargo( CargoDTO cargo )
{
    _cargo = cargo;
}

public void setEstado( int estado )
{
    _estado = estado;
}

public void setCorreo( String correo )
{
    _correo = correo;
}

public void setTelefono( String telefono )
{
    _telefono = telefono;
}

public void setUbicacion( String ubicacion )
{
    _ubicacion = ubicacion;
}

}

```

## mx.uam.azc.cbi.sistemas.gestion.comunicacion.interceptors

### UsuariosInterceptor

```
package mx.uam.azc.cbi.sistemas.gestion.comunicacion.interceptors;

import mx.uam.azc.cbi.sistemas.gestion.usuarios.Data.UsuarioDTO;
import com.opensymphony.xwork2.interceptor.Interceptor;
import com.opensymphony.xwork2.ActionInvocation;
import java.util.Map;

public class UsuariosInterceptor implements Interceptor
{
    public void destroy()
    {
        System.out.println("Destruyendo UsuariosInterceptor...");
    }
    public void init()
    {
        System.out.println("Inicializando UsuariosInterceptor...");
    }

    public String intercept( ActionInvocation invocation ) throws Exception
    {
        // System.out.println("administracion:protegida");
        // return invocation.invoke();
        Map<?, ?> session = invocation.getInvocationContext().getSession();

        UsuarioDTO usuario = ( UsuarioDTO )session.get( "usuario" );

        if( usuario != null)
        {
            if (usuario.getTipo() == 1 || usuario.getTipo() == 2)
            {
                return invocation.invoke();
            }
            else return "login";
        }

        else
            return "login";
    }
}
```

## mx.uam.azc.cbi.sistemas.gestion.usuarios.bussines

### AdministradorUsuarios

```
package mx.uam.azc.cbi.sistemas.gestion.usuarios.bussines;

import mx.uam.azc.cbi.sistemas.gestion.usuarios.Data.UsuarioDTO;
import mx.uam.azc.cbi.sistemas.gestion.usuarios.Data.ProfesorDTO;
import mx.uam.azc.cbi.sistemas.gestion.usuarios.Data.AdministrativoDTO;
import org.springframework.transaction.annotation.Propagation;
import org.springframework.transaction.annotation.Transactional;
import java.util.List;
import java.util.Map;

@Transactional(rollbackFor = Exception.class)
public interface AdministradorUsuarios
```



```

{
    @Transactional(propagation = Propagation.SUPPORTS, readOnly = true)
    public List<UsuarioDTO> leerUsuarios();

    @Transactional(propagation = Propagation.SUPPORTS, readOnly = true)
    public List<ProfesorDTO> leerProfesores();

    @Transactional(propagation = Propagation.SUPPORTS, readOnly = true)
    public List<AdministrativoDTO> leerAdministrativos();

    @Transactional(propagation = Propagation.SUPPORTS, readOnly = true)
    public UsuarioDTO leerUsuario( int id );

    @Transactional( propagation=Propagation.SUPPORTS, readOnly = true )
    public UsuarioDTO leerUsuarioLoginPassword(String login, String pasw);
}

```

## mx.uam.azc.cbi.sistemas.gestion.usuarios.bussines.local

### AdministradorUsuariosImpl

```

package mx.uam.azc.cbi.sistemas.gestion.usuarios.bussines.local;

import mx.uam.azc.cbi.sistemas.gestion.usuarios.Data.*;
import mx.uam.azc.cbi.sistemas.gestion.usuarios.bussines.AdministradorUsuarios;

import java.util.List;
import java.util.Map;
import org.hibernate.*;
import java.util.*;

public class AdministradorUsuariosImpl implements AdministradorUsuarios
{
    private SessionFactory _sessionFactory;

    public UsuarioDTO leerUsuario( int id )
    {
        Session session = getSession();
        return ( UsuarioDTO )session.get( UsuarioDTO.class, id );
    }

    public List<UsuarioDTO> leerUsuarios()
    {
        Session session = _sessionFactory.getCurrentSession();
        Query query = session.getNamedQuery( "leer_Usuarios" );
        return query.list();
    }

    public List<ProfesorDTO> leerProfesores()
    {
        Session session = _sessionFactory.getCurrentSession();
        Query query = session.getNamedQuery( "leer_Profesores" );
        return query.list();
    }
}

```

```

}

public List<AdministrativoDTO> leerAdministrativos()
{
    Session session = _sessionFactory.getCurrentSession();
    Query query = session.getNamedQuery( "leer_Administrativos" );
    return query.list();
}

public UsuarioDTO leerUsuarioLoginPassword( String login, String pasw )
{
    Session session = _sessionFactory.getCurrentSession();
    Query query = session.getNamedQuery( "leer_usuario_login_pasw" );
    query.setString( 0, login );
    query.setString( 1, pasw );
    List<UsuarioDTO> lista = query.list();
    return lista.isEmpty() ? null : lista.get( 0 );
}

private Session getSession()
{
    return _sessionFactory.getCurrentSession();
}

public SessionFactory getSessionFactory()
{
    return _sessionFactory;
}

public void setSessionFactory( SessionFactory sessionFactory )
{
    _sessionFactory = sessionFactory;
}
}

```

## mx.uam.azc.cbi.sistemas.gestion.usuarios.Data

### AdministrativoDTO

```
package mx.uam.azc.cbi.sistemas.gestion.usuarios.Data;
```

```
import java.io.Serializable;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.*;
```

```
@Entity
@Table(name = "administrativo")
public class AdministrativoDTO implements Serializable
{
    private int _idadministrativo;
    private String _nombre;
    private String _paterno;
    private String _materno;
    private String _economico;
    private String _email;

    @Id @Column( name="id_administrativo" ) @GeneratedValue( strategy=IDENTITY )
    public int getIdadministrativo()
    {

```

```

        return _idadministrativo;
    }

    @Column( name="nombre" )
    public String getNombre()
    {
        return _nombre;
    }

    @Column( name="paterno" )
    public String getPaterno()
    {
        return _paterno;
    }

    @Column( name="materno" )
    public String getMaterno()
    {
        return _materno;
    }

    @Column( name="economico" )
    public String getEconomico()
    {
        return _economico;
    }

    @Column( name="email" )
    public String getEmail()
    {
        return _email;
    }

    public void setIdadministrativo( int idadministrativo )
    {
        _idadministrativo = idadministrativo;
    }

    public void setNombre( String nombre )
    {
        _nombre = nombre;
    }

    public void setPaterno( String paterno )
    {
        _paterno = paterno;
    }

    public void setMaterno( String materno )
    {
        _materno = materno;
    }

    public void setEconomico( String economico )
    {
        _economico = economico;
    }

    public void setEmail( String email )
    {
        _email = email;
    }
}

```

## ProfesorDTO

```
package mx.uam.azc.cbi.sistemas.gestion.usuarios.Data;
```

```

import java.io.Serializable;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.*;

@Entity
@Table(name = "profesor")
public class ProfesorDTO implements Serializable
{
    private int _idprofesor;
    private String _nombre;
    private String _paterno;
    private String _materno;
    private String _economico;
    private String _email;

    @Id @Column( name="id_profesor" ) @GeneratedValue( strategy=IDENTITY )
    public int getIdprofesor()
    {
        return _idprofesor;
    }

    @Column( name="nombre" )
    public String getNombre()
    {
        return _nombre;
    }

    @Column( name="paterno" )
    public String getPaterno()
    {
        return _paterno;
    }

    @Column( name="materno" )
    public String getMaterno()
    {
        return _materno;
    }

    @Column( name="economico" )
    public String getEconomico()
    {
        return _economico;
    }

    @Column( name="email" )
    public String getEmail()
    {
        return _email;
    }

    public void setIdprofesor( int idprofesor )
    {
        _idprofesor = idprofesor;
    }

    public void setNombre( String nombre )
    {
        _nombre = nombre;
    }

    public void setPaterno( String paterno )
    {
        _paterno = paterno;
    }

    public void setMaterno( String materno )
    {

```

```

    _materno = materno;
}

public void setEconomico( String economico )
{
    _economico = economico;
}

public void setEmail( String email )
{
    _email = email;
}

}

```

## PuestoDTO

```
package mx.uam.azc.cbi.sistemas.gestion.usuarios.Data;
```

```
import java.io.Serializable;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.*;
```

```
@Entity
@Table(name = "puesto")
public class PuestoDTO implements Serializable
{
    private int _idpuesto;
    private String _puesto;
```

```
@Id @Column( name="id_puesto" ) @GeneratedValue( strategy=IDENTITY )
public int getIdpuesto()
{
    return _idpuesto;
}
```

```
@Column( name="puesto" )
public String getpuesto()
{
    return _puesto;
}
```

```
public void setIdpuesto( int idpuesto )
{
    _idpuesto = idpuesto;
}
```

```
public void setpuesto( String puesto )
{
    _puesto = puesto;
}
```

```
}
```

## UsuarioDTO

```
package mx.uam.azc.cbi.sistemas.gestion.usuarios.Data;
```

```

import java.io.Serializable;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.*;

@Entity
@Table(name = "usuario")
public class UsuarioDTO implements Serializable
{
    private int _idusuario;
    private String _login;
    private String _password;
    private String _rol;
    private int _tipo;
    private String _estado;
    private String _nombre;
    private String _paterno;
    private String _materno;
    private String _numEconomico;
    private PuestoDTO _puesto;

    @Id @Column( name="id_usuario" ) @GeneratedValue( strategy=IDENTITY )
    public int getIdusuario()
    {
        return _idusuario;
    }

    @Column( name="login" )
    public String getLogin()
    {
        return _login;
    }

    @Column( name="password" )
    public String getPassword()
    {
        return _password;
    }

    @Column( name="rol" )
    public String getRol()
    {
        return _rol;
    }

    @Column( name="tipo" )
    public int getTipo()
    {
        return _tipo;
    }

    @Column( name="estado" )
    public String getEstado()
    {
        return _estado;
    }

    @Column( name="nombre" )
    public String getNombre()
    {
        return _nombre;
    }
}

```

```

@Column( name="paterno" )
public String getPaterno()
{
    return _paterno;
}

@Column( name="materno" )
public String getMaterno()
{
    return _materno;
}

@Column( name="numero_economico" )
public String getNumEconomico()
{
    return _numEconomico;
}

@OneToOne
@JoinColumn(name="puesto")
public PuestoDTO getPuesto()
{
    return _puesto;
}

public void setIdusuario( int idusuario )
{
    _idusuario = idusuario;
}

public void setLogin( String login )
{
    _login = login;
}

public void setPassword( String password )
{
    _password = password;
}

public void setRol( String rol )
{
    _rol = rol;
}

public void setTipo( int tipo )
{
    _tipo = tipo;
}

public void setEstado( String estado )
{
    _estado = estado;
}

public void setNombre( String nombre )
{
    _nombre = nombre;
}

public void setPaterno( String paterno )
{
    _paterno = paterno;
}

public void setMaterno( String materno )
{
    _materno = materno;
}

public void setNumEconomico( String numEconomico )

```

```

{
    _numEconomico = numEconomico;
}

public void setPuesto( PuestoDTO puesto )
{
    _puesto = puesto;
}
}

```

## Struts.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC
"-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
"http://struts.apache.org/dtds/struts-2.0.dtd">
<struts>
    <constant name="struts.devMode" value="true"/>
    <constant name="struts.objectFactory"
        value="org.apache.struts2.spring.StrutsSpringObjectFactory"/>
    <constant name="struts.ui.theme" value="simple"/>
    <include file="base.xml"/>
    <package name="gestionWeb" extends="base">

        <action name="responsable_*"
            class="mx.uam.azc.cbi.sistemas.gestion.comunicacion.actions.ResponsableAction" method="{1}">
            <interceptor-ref name="usuarioStack"/>
            <result name="formInsertar">/WEB-INF/pages/esOrganizacional/AddResponsables.jsp</result>
            <result name="insertar">/WEB-INF/pages/esOrganizacional/listar.jsp</result>
            <result name="gAreas">/WEB-INF/pages/esOrganizacional/Areas.jsp</result>
            <result name="gAreasInsert">/WEB-INF/pages/esOrganizacional/Areas.jsp</result>
            <result name="gCargos">/WEB-INF/pages/esOrganizacional/Cargos.jsp</result>
            <result name="gCargosInsert">/WEB-INF/pages/esOrganizacional/Cargos.jsp</result>
            <result name="listar">/WEB-INF/pages/esOrganizacional/listar.jsp</result>
            <result name="listarAdmin">/WEB-INF/pages/esOrganizacional/listarAdmin.jsp</result>
            <result name="modificar">/WEB-INF/pages/esOrganizacional/ModificaResponsables.jsp</result>
            <result name="Login">/WEB-INF/pages/login/login.jsp</result>

```



```

</action>

<action name="recibidos_*"
class="mx.uam.azc.cbi.sistemas.gestion.comunicacion.actions.RecibidosAction" method="{1}">
  <interceptor-ref name="usuarioStack"/>
  <result name="formInsertar">/WEB-INF/pages/docRecibidos/AddRecibidos.jsp</result>
  <result name="insertar">/WEB-INF/pages/fileUpload/fileupload.jsp</result>
  <result name="listar">/WEB-INF/pages/docRecibidos/listar.jsp</result>
  <result name="login">/WEB-INF/pages/login/login.jsp</result>
</action>

<action name="elaborados_*"
class="mx.uam.azc.cbi.sistemas.gestion.comunicacion.actions.ElaboradosAction" method="{1}">
  <interceptor-ref name="usuarioStack"/>
  <result name="formInsertar">/WEB-INF/pages/docElaborados/AddElaborados.jsp</result>
  <result name="insertar">/WEB-INF/pages/fileUpload/fileuploadElab.jsp</result>
  <result name="listar">/WEB-INF/pages/docElaborados/listar.jsp</result>
  <result name="login">/WEB-INF/pages/login/login.jsp</result>
</action>

<action name="Login_*" class="mx.uam.azc.cbi.sistemas.gestion.usuarios.actions.LoginAction"
method="{1}">
  <result name="login">/WEB-INF/pages/login/login.jsp</result>
  <result name="principal">/WEB-INF/pages/login/principal.jsp</result>
  <result name="home">/WEB-INF/pages/login/principal.jsp</result>
</action>

  <action name="fileUpload_*"
class="mx.uam.azc.cbi.sistemas.gestion.comunicacion.actions.FileUploadAction"
method="{1}">

  <result name="fileupload">/WEB-INF/pages/fileUpload/fileupload.jsp</result>
  </action>
  <action name="result_*" class="mx.uam.azc.cbi.sistemas.gestion.comunicacion.actions.FileUploadAction"
method="{1}">
    <interceptor-ref name="exception"/>
<!--      <interceptor-ref name="i18n"/> -->
    <interceptor-ref name="fileUpload">
      <param name="maximumSize">985140000</param>
    </interceptor-ref>
    <interceptor-ref name="params">
      <param name="excludeParams">dojo\..*,^struts\..*</param>
    </interceptor-ref>
    <interceptor-ref name="validation">
      <param name="excludeMethods">input,back, cancel, browse</param>
    </interceptor-ref>
    <interceptor-ref name="workflow">
      <param name="excludeMethods">input,back, cancel, browse</param>
    </interceptor-ref>

    <result name="uploadResult">/WEB-INF/pages/docRecibidos/listar.jsp</result>
    <result name="uploadResult2">/WEB-INF/pages/docElaborados/listar.jsp</result>
  </action>

</package>
</struts>

```

## base.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE struts PUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
    "http://struts.apache.org/dtds/struts-2.0.dtd">

<struts>
  <package name="base" extends="struts-default" abstract="true">

    <interceptors>
      <interceptor name="usuarios_interceptor"
        class="mx.uam.azc.cbi.sistemas.gestion.comunicacion.interceptors.UsuariosInterceptor"/>

      <!-- <interceptor name="map_to_request" -->

      <interceptor-stack name="customStack">
        <interceptor-ref name="exception"/>
      <!-- <interceptor-ref name="map_to_request"/> -->
        <interceptor-ref name="servletConfig"/>
        <interceptor-ref name="prepare"/>
        <interceptor-ref name="i18n"/>
        <interceptor-ref name="debugging"/>
        <interceptor-ref name="profiling"/>
        <interceptor-ref name="checkbox"/>
        <interceptor-ref name="params">
          <param name="excludeParams">dojo\..*\</param>
        </interceptor-ref>
        <interceptor-ref name="conversionError"/>
        <interceptor-ref name="validation">
          <param name="excludeMethods">input,back, cancel, browse</param>
        </interceptor-ref>
        <interceptor-ref name="workflow">
          <param name="excludeMethods">input,back, cancel, browse</param>
        </interceptor-ref>
      </interceptor-stack>
      <interceptor-stack name="usuarioStack">
        <interceptor-ref name="usuarios_interceptor"/>
        <interceptor-ref name="customStack"/>
      </interceptor-stack>

    </interceptors>

    <default-interceptor-ref name="customStack"/>

  </package>
</struts>
```

---

## spring-dev.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans
  xmlns="http://www.springframework.org/schema/beans"
  xmlns:tx="http://www.springframework.org/schema/tx"
  xmlns:context="http://www.springframework.org/schema/context"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
```

<http://www.springframework.org/schema/tx>  
<http://www.springframework.org/schema/tx/spring-tx-2.5.xsd>  
<http://www.springframework.org/schema/context>  
<http://www.springframework.org/schema/context/spring-context-2.5.xsd>  
 default-autowire="byName" default-lazy-init="true">

```

<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource"
  destroy-method="close">
  <property name="driverClassName">
    <value>org.postgresql.Driver</value>
  </property>
  <property name="url">
    <value>jdbc:postgresql://localhost:5432/gestionWeb2?autoReconnect=true
    </value>
  </property>
  <property name="maxWait" value="10000"/>
  <property name="username" value="postgres"/>
  <property name="password" value="AdmCompnt01"/>
</bean>
<bean id="transactionManager"
  class="org.springframework.orm.hibernate3.HibernateTransactionManager"/>
<tx:annotation-driven/>
<bean id="sessionFactory"
  class="org.springframework.orm.hibernate3.annotation.AnnotationSessionFactoryBean">
  <property name="annotatedClasses">
    <list>
      <value>mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data.AreaDTO</value>
      <value>mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data.ResponsableDTO</value>
      <value>mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data.CargoDTO</value>
      <value>mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data.ResponsableJoinDTO</value>
      <value>mx.uam.azc.cbi.sistemas.gestion.usuarios.Data.UsuarioDTO</value>
      <value>mx.uam.azc.cbi.sistemas.gestion.usuarios.Data.ProfesorDTO</value>
      <value>mx.uam.azc.cbi.sistemas.gestion.usuarios.Data.AdministrativoDTO</value>
      <value>mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data.RecibidosDTO</value>
      <value>mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data.RecibidosJoinDTO</value>
      <value>mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data.ElaboradosDTO</value>
      <value>mx.uam.azc.cbi.sistemas.gestion.comunicacion.Data.ElaboradosJoinDTO</value>
      <value>mx.uam.azc.cbi.sistemas.gestion.usuarios.Data.PuestoDTO</value>
    </list>
  </property>
  <property name="hibernateProperties">
    <props>
      <prop key="hibernate.dialect">
        org.hibernate.dialect.PostgreSQLDialect
      </prop>
    </props>
  </property>
  <property name="mappingResources">
    <list>
      <value>queries.hbm.xml</value>
    </list>
  </property>
</bean>
<import resource="spring-services.xml"/>
</beans>

```

---

## spring-services.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:tx="http://www.springframework.org/schema/tx"
  xmlns:context="http://www.springframework.org/schema/context"

```

```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-2.5.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-2.5.xsd"
default-autowire="byName" default-lazy-init="true">

<bean id="administradorAreas"
class="mx.uam.azc.cbi.sistemas.gestion.comunicacion.bussines.Local.AdministradorAreaImpl"/>
<bean id="administradorCargos"
class="mx.uam.azc.cbi.sistemas.gestion.comunicacion.bussines.Local.AdministradorCargoImpl"/>
<bean id="administradorResponsables"
class="mx.uam.azc.cbi.sistemas.gestion.comunicacion.bussines.Local.AdministradorResponsablesImpl"/>
<bean id="administradorUsuarios"
class="mx.uam.azc.cbi.sistemas.gestion.usuarios.bussines.Local.AdministradorUsuariosImpl"/>
<bean id="administradorRecibidos"
class="mx.uam.azc.cbi.sistemas.gestion.comunicacion.bussines.Local.AdministradorRecibidosImpl"/>
<bean id="administradorELaborados"
class="mx.uam.azc.cbi.sistemas.gestion.comunicacion.bussines.Local.AdministradorELaboradosImpl"/>
</beans>

```

---

## querys.hbm.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping PUBLIC
"-//Hibernate/Hibernate Mapping DTD//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>

<query name="Leer_Areas">
  SELECT a FROM AreaDTO a
</query>

<query name="Leer_Cargos">
  SELECT c FROM CargoDTO c
</query>

<query name="Leer_area">
FROM AreaDTO a WHERE
(
(a.area = ? )
)
</query>

<query name="Leer_cargo">
FROM CargoDTO c WHERE
(
(c.cargo = ? )
)
</query>

<query name="Leer_Reponsables">
  FROM ResponsableJoinDTO r
</query>

<query name="Leer_Responsable">
  FROM ResponsableJoinDTO R WHERE
(
(R.idresponsable = ? )
)

```

```

)
</query>

<query name="Leer_Usuario">
FROM UsuarioDTO U WHERE
(
(U.idusuario = ? )
)
</query>

<query name="Leer_Usuarios">
SELECT u FROM UsuarioDTO u
</query>

<query name="Leer_Profesores">
SELECT P FROM ProfesorDTO P
</query>

<query name="Leer_Administrativos">
SELECT A FROM AdministrativoDTO A
</query>

<query name="Leer_Recibidos">
FROM RecibidosJoinDTO Rs
</query>

<query name="Leer_Recibido">
FROM RecibidosDTO R WHERE
(
(R.idrecibidos = ? )
)
</query>

<query name="Leer_Elaborados">
FROM ElaboradosJoinDTO E
</query>
<query name="Leer_usuario_Login_pasw">
FROM UsuarioDTO u WHERE
(
(u.login = ? )AND
(u.password = ?)
)
</query>
<query name="Leer_Responsable_Activos">
FROM ResponsableJoinDTO rja WHERE
(
(rja.estado = ? )
)
</query>
<query name="Leer_Recibidos_filtro">
FROM RecibidosJoinDTO r WHERE(r.anio LIKE :nombre )
</query>

</hibernate-mapping>

```

---

## Web.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"

```

```

xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" id="WebApp_ID" version="2.5">
  <display-name>gestionWeb</display-name>
  <context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>classpath:spring-dev.xml</param-value>
  </context-param>
  <listener>
    <listener-class>
      org.springframework.web.context.ContextLoaderListener
    </listener-class>
  </listener>
  <filter>
    <filter-name>struts-cleanup</filter-name>
    <filter-class>
      org.apache.struts2.dispatcher.ActionContextCleanUp
    </filter-class>
  </filter>
  <filter-mapping>
    <filter-name>struts-cleanup</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
  <filter>
    <display-name>SiteMeshFilter</display-name>
    <filter-name>SiteMeshFilter</filter-name>
    <filter-class>com.opensymphony.module.sitemesh.filter.PageFilter</filter-
class>
  </filter>
  <filter-mapping>
    <filter-name>SiteMeshFilter</filter-name>
    <url-pattern>*.action</url-pattern>
  </filter-mapping>
  <filter>
    <filter-name>struts2</filter-name>
    <filter-class>org.apache.struts2.dispatcher.FilterDispatcher</filter-class>
  </filter>
  <filter-mapping>
    <filter-name>struts2</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
</web-app>

```

# Manual del usuario

## Ingresar al sistema

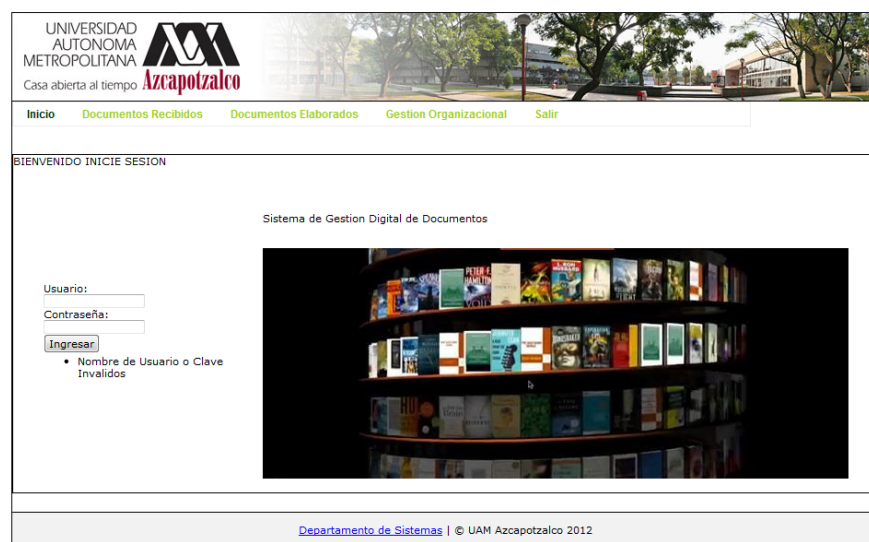
Introduce su clave de usuario y su password

The screenshot shows the login interface of the 'Sistema de Gestion Digital de Documentos'. At the top, there is a header with the university logo and name: 'UNIVERSIDAD AUTONOMA METROPOLITANA Azcapotzalco' and the slogan 'Casa abierta al tiempo'. Below the header is a navigation menu with links: 'Inicio', 'Documentos Recibidos', 'Documentos Elaborados', 'Gestion Organizacional', and 'Salir'. The main content area is titled 'BIENVENIDO INICIE SESION' and contains the text 'Sistema de Gestion Digital de Documentos'. On the left side, there are input fields for 'Usuario:' (containing 'usuarioux') and 'Contraseña:' (containing '\*\*\*\*'), followed by an 'Ingresar' button. On the right side, there is a large image of a curved bookshelf filled with books. At the bottom of the page, there is a footer with the text 'Departamento de Sistemas | © UAM Azcapotzalco 2012'.

El usuario y password se validan y si los datos son válidos se muestra la siguiente pantalla.

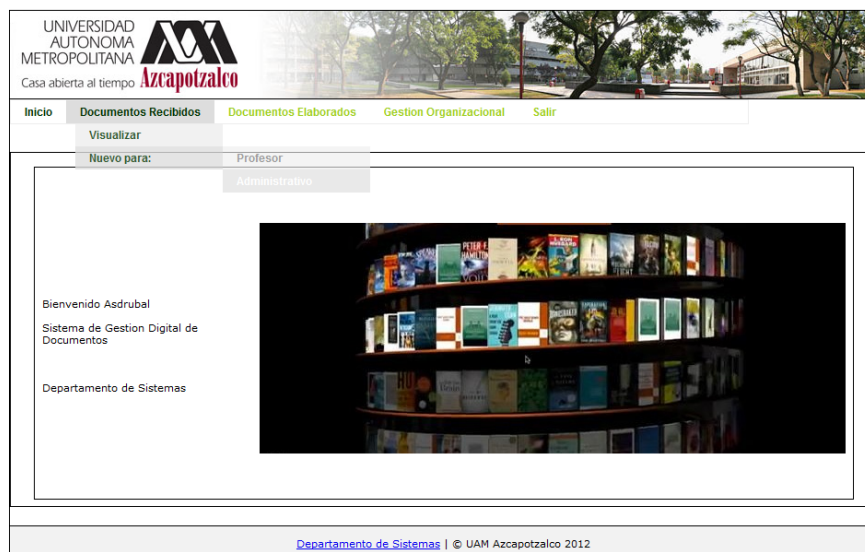
The screenshot shows the main dashboard of the 'Sistema de Gestion Digital de Documentos' after a successful login. The header and navigation menu are identical to the login page. The main content area is titled 'BIENVENIDO Blanca' and contains the text 'Sistema de Gestion Digital de Documentos' and 'Departamento de Sistemas'. On the right side, there is a large image of a curved bookshelf filled with books. At the bottom of the page, there is a footer with the text 'Departamento de Sistemas | © UAM Azcapotzalco 2012'.

Si los datos no corresponden con los que se encuentran en la base de datos se re direcciona a la pantalla de login y se informa que los datos no son validos.



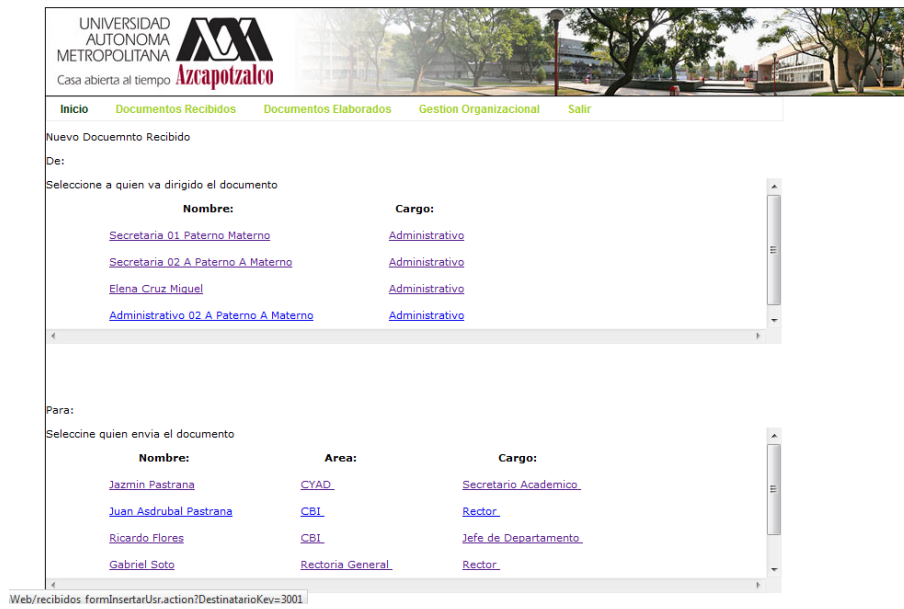
## Generar u nuevo documento recibido

Se selecciona del menú la opción **Documentos Recibidos**, del menú desplegable se elige la opción **Nuevo para:** y se escoge la opción de **Profesores** o **Administrativos** ya que los documento pueden llegar para algún profesor o un administrativo del departamento de sistemas

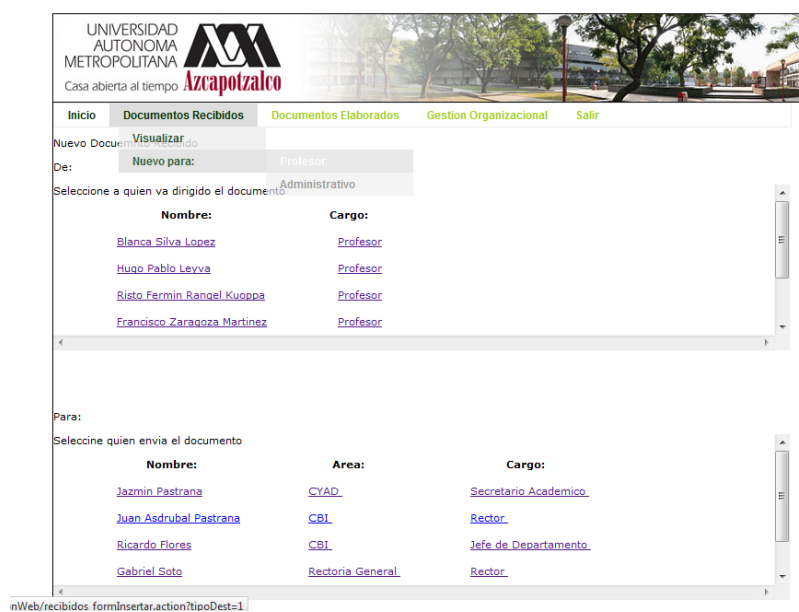


Se selecciona al remitente del área administrativa que es el que envía el documento, si este no se encuentra en la lista se debe agregar en el modulo de **Gestión Organizacional** y dar aviso al administrador de la pagina para que corrobore datos y se active para su uso.





Si se selecciona del menú principal la opción **Nuevo para:** se elige **profesor** se desplegara la misma pantalla pero con una lista de profesores del área de sistemas para seleccionar al usuario que se le envía el documento.



Se selecciona al usuario del departamento de sistemas al que esta dirigido el documento

UNIVERSIDAD AUTONOMA METROPOLITANA Azcapotzalco

Inicio Documentos Recibidos Documentos Elaborados Gestion Organizacional Salir

Nuevo Documento Recibido

De:  
Elena Cruz Miguel Asistente Administrativo

Para:  
Seleccione quien envia el documento

| Nombre:                                | Area:                            | Cargo:                               |
|--|----------------------------------|--------------------------------------|
| <a href="#">Jazmin Pastrana</a>        | <a href="#">CYAD</a>             | <a href="#">Secretario Academico</a> |
| <a href="#">Juan Asdrubal Pastrana</a> | <a href="#">CBI</a>              | <a href="#">Rector</a>               |
| <a href="#">Ricardo Flores</a>         | <a href="#">CBI</a>              | <a href="#">Jefe de Departamento</a> |
| <a href="#">Gabriel Soto</a>           | <a href="#">Rectoria General</a> | <a href="#">Rector</a>               |

Fecha:

Numero del documento:

Tipo:

[Web/recibidos\\_forminsertarResp.action?RmitenteKey=7](#)

Se llena el formulario con la información correspondiente al documento recibido y se presiona el botón **Siguiente**.

UNIVERSIDAD AUTONOMA METROPOLITANA Azcapotzalco

Inicio Documentos Recibidos Documentos Elaborados Gestion Organizacional Salir

Nuevo Documento Recibido

De:  
Elena Cruz Miguel Asistente Administrativo

Para:  
Jazmin Pastrana Avila Secretario Academico CYAD

Fecha:

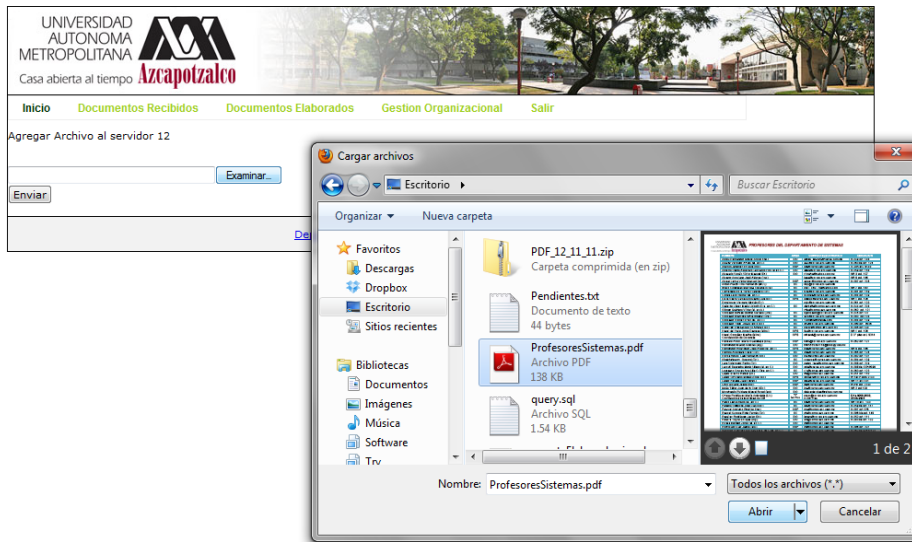
Numero del documento:

Tipo:

Asunto:

[Departamento de Sistemas](#) | © UAM Azcapotzalco 2012

El sistema genera el registro del documento en la base de datos y se despliega la pantalla para buscar el documento recibido digitalizado. Se selecciona el botón **Examinar** y se busca el documento en el sistema de archivos del usuario, una vez seleccionado se presiona el botón de **Abrir**.



Se selecciona la opción enviar para que el sistema envíe el archivo al servidor.

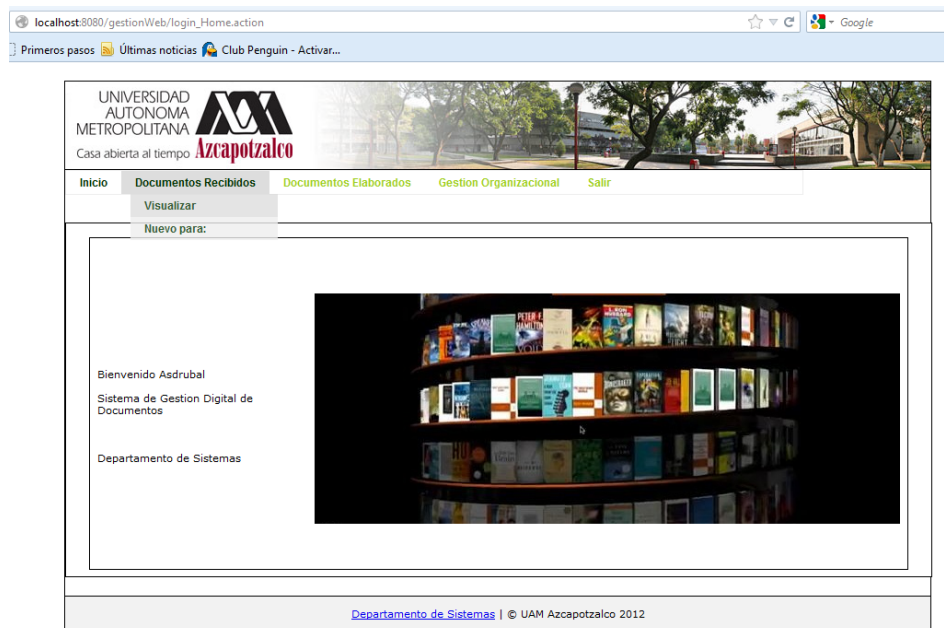


Una vez enviado el archivo el sistema lo coloca en el directorio que le corresponde en el servidor y se direcciona a la pantalla documentos recibidos en la que se muestran todos los documentos enviados al servidor



## Visualizar Documentos Recibidos

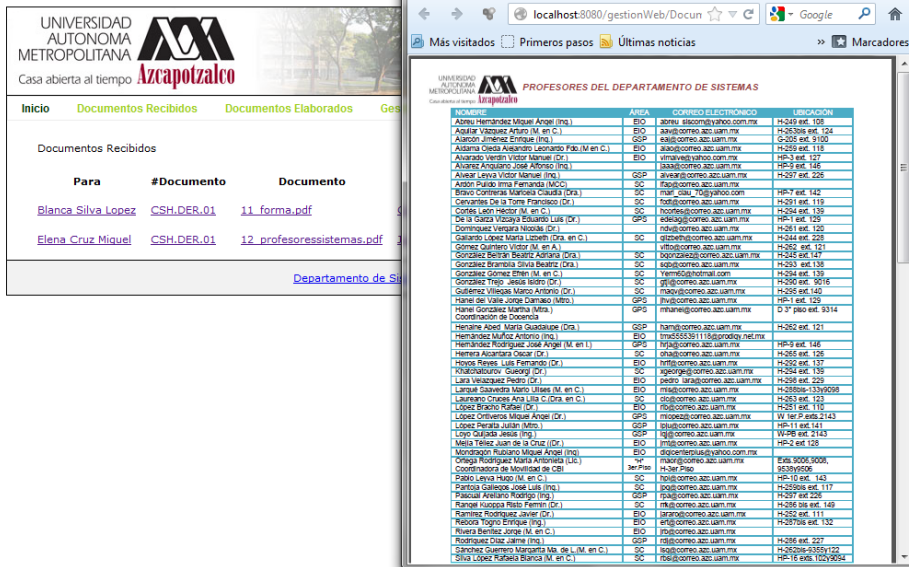
Se selecciona del menú principal la opción **documentos Recibidos** y del menú desplegable se selecciona la opción visualizar



Se muestra la pantalla en donde se listan los documentos recibidos y se puede seleccionar alguno de ellos para descargarlo



Se desplegara una ventana con el documento si el navegador cuenta con el visualizador del archivo como de PDF.



Si se desea eliminar un documento recibido en la lista de documentos se encuentra un encabezado llamado **Opciones** el cual proporciona un botón para eliminar.

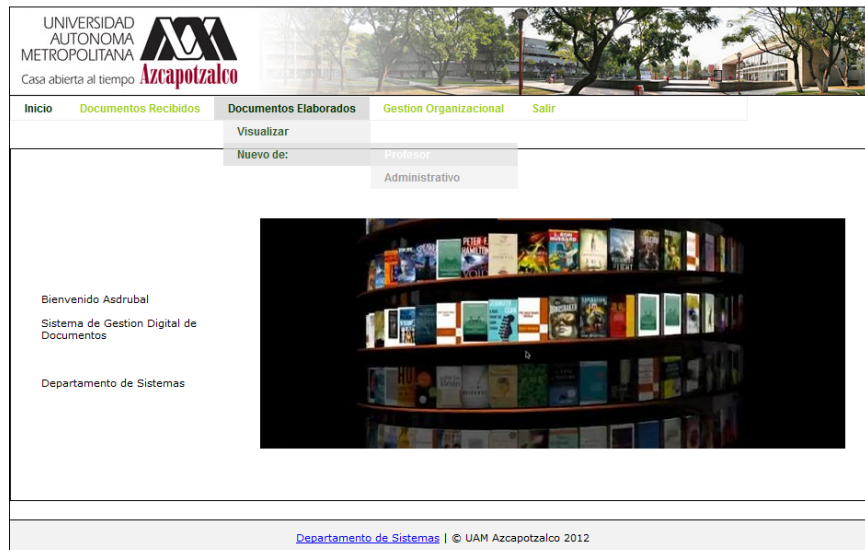


El sistema te re direcciona a la misma pantalla mostrando el la lista de los documentos recibidos sin el documento previamente eliminado.

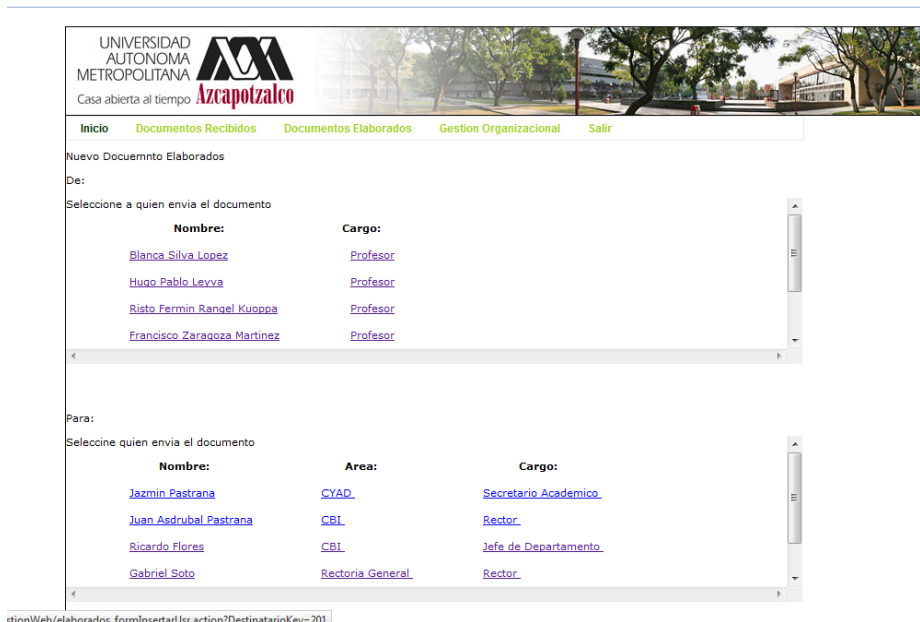


## Caso de uso documentos elaborados

Se selecciona del menú la opción **Documentos Elaborados**, del menú desplegable se elige la opción **Nuevo para:** y se escoge la opción de **Profesores** o **Administrativos** ya que pueden llegar solicitudes de elaborar documentos de algún profesor o un administrativo del departamento de sistemas



Seleccionar al usuario del departamento de sistemas que envía el documento



Se selecciona al destinatario del directorio que es al que se envía el documento, si este no se encuentra en la lista se debe agregar en el modulo de **Gestión Organizacional** y dar aviso al administrador de la pagina para que corrobore datos y se active para su uso.



UNIVERSIDAD AUTONOMA METROPOLITANA **Azcapotzalco**  
 Casa abierta al tiempo

[Inicio](#) [Documentos Recibidos](#) [Documentos Elaborados](#) [Gestion Organizacional](#) [Salir](#)

Nuevo Documento Elaborados

De:  
Risto Fermin Rangel Kuoppa Profesor Investigador

Para:  
Seleccione quien envia el documento

| Nombre:                                | Area:                            | Cargo:                               |
|--|----------------------------------|--------------------------------------|
| <a href="#">Jazmin Pastrana</a>        | <a href="#">CYAD</a>             | <a href="#">Secretario Academico</a> |
| <a href="#">Juan Asdrubal Pastrana</a> | <a href="#">CBI</a>              | <a href="#">Rector</a>               |
| <a href="#">Ricardo Flores</a>         | <a href="#">CBI</a>              | <a href="#">Jefe de Departamento</a> |
| <a href="#">Gabriel Soto</a>           | <a href="#">Rectoria General</a> | <a href="#">Rector</a>               |

Fecha:

Numero del documento:

Asunto:

[elaborados?forminsertarResp.action?RmitenteKey=7](#)

Se llena el formulario con la información correspondiente al documento elaborado y se presiona el botón **Siguiente**.

UNIVERSIDAD AUTONOMA METROPOLITANA **Azcapotzalco**  
 Casa abierta al tiempo

[Inicio](#) [Documentos Recibidos](#) [Documentos Elaborados](#) [Gestion Organizacional](#) [Salir](#)

Nuevo Documento Elaborados

De:  
Risto Fermin Rangel Kuoppa Profesor Investigador

Para:  
Jazmin Pastrana Avila Secretario Academico CYAD

Fecha:

Numero del documento:

Asunto:

Contenido:

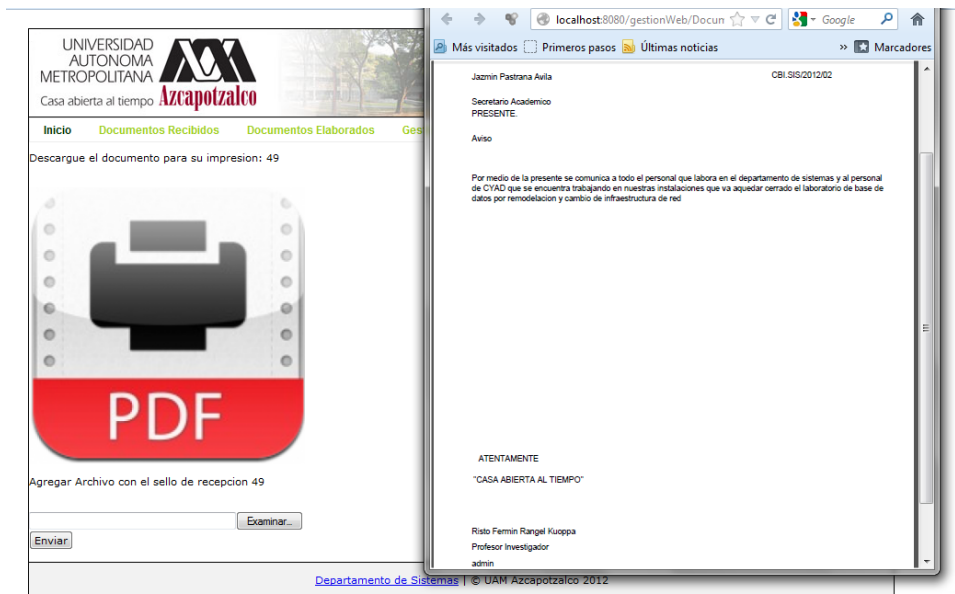
[Siguiente](#)

[Departamento de Sistemas](#) | © UAM Azcapotzalco 2012

El sistema genera un archivo temporal en formato PDF

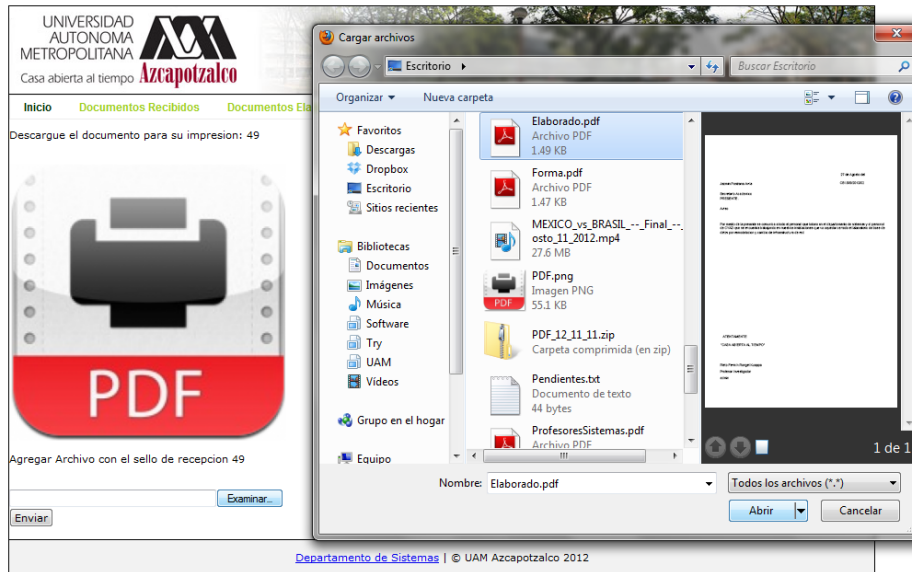


Se selecciona la imagen en de la impresora para descargar el archivo e imprimirlo en hoja membretada del departamento de sistemas.



Una vez impreso el documento y entregado se digitaliza el acuse de recibo presiona el boton exminar para buscar el archivo en el sistema de archivos del usuario, una vez seleccionado se preciona el boton aceptar.





Se selecciona la opción de enviar para subir el archivo al servidor y se almacene en el directorio correspondiente.



Se re direcciona a la pantalla **Documentos Elaborados** para poder visualizar los documentos que se han elaborado

UNIVERSIDAD AUTONOMA METROPOLITANA Azcapotzalco  
Casa abierta al tiempo

Inicio Documentos Recibidos Documentos Elaborados Gestion Organizacional Salir

Documentos Elaborados

| #Documento                      | Para                                  | Cargo                                 | Area                             | Fecha                                 | Opciones |
|---------------------------------|---------------------------------------|---------------------------------------|----------------------------------|---------------------------------------|----------|
| <a href="#">CBI.SIS/2012/01</a> | <a href="#">Mario A. Rocha Perez</a>  | <a href="#">Coordinador de DYSCOM</a> | <a href="#">Rectoria General</a> | <a href="#">25 de Agosto del 2012</a> |          |
| <a href="#">CBI.SIS/2012/02</a> | <a href="#">Jazmin Pastrana Avila</a> | <a href="#">Secretario Academico</a>  | <a href="#">CYAD</a>             | <a href="#">27 de Agosto del 2012</a> |          |
| <a href="#">CBI.SIS/2012/02</a> | <a href="#">Jazmin Pastrana Avila</a> | <a href="#">Secretario Academico</a>  | <a href="#">CYAD</a>             | <a href="#">27 de Agosto del 2012</a> |          |

[Departamento de Sistemas](#) | © UAM Azcapotzalco 2012

Si se desea se puede consultar el acuse de recibo o eliminarlo en caso de que se halla seleccionado mal el archivo.

UNIVERSIDAD AUTONOMA METROPOLITANA Azcapotzalco  
Casa abierta al tiempo

Inicio Documentos Recibidos Documentos Elaborados Gestion Organizacional

Documentos Elaborados

| #Documento                      | Para                                  | Cargo                                 | Area                         |
|---------------------------------|---------------------------------------|---------------------------------------|------------------------------|
| <a href="#">CBI.SIS/2012/01</a> | <a href="#">Mario A. Rocha Perez</a>  | <a href="#">Coordinador de DYSCOM</a> | <a href="#">Rectoria Gen</a> |
| <a href="#">CBI.SIS/2012/02</a> | <a href="#">Jazmin Pastrana Avila</a> | <a href="#">Secretario Academico</a>  | <a href="#">CYAD</a>         |

[Departamento de Sistemas](#) | © UAM

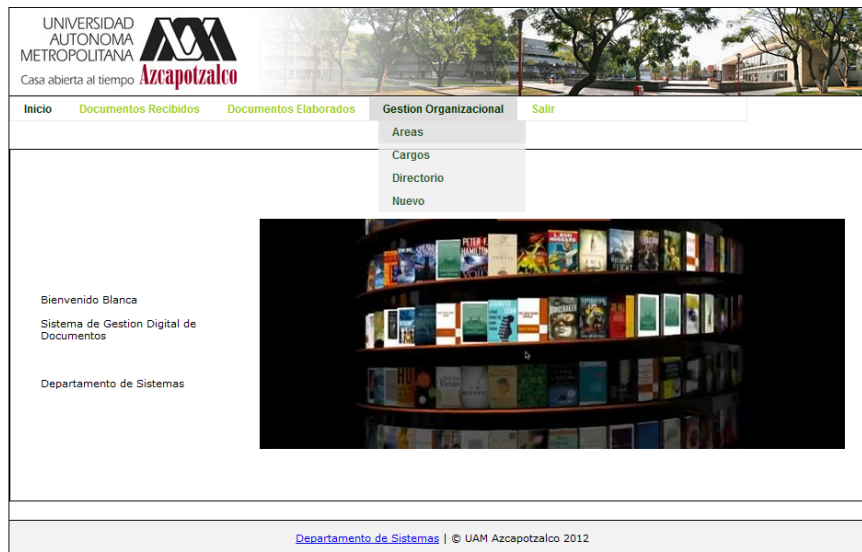
27 de Agosto del  
Jazmin Pastrana Avila  
CBI.SIS/2012/02  
Secretario Academico  
PRESENTE  
Aviso  
Por medio de la presente se comunica a todo el personal que labora en el departamento de sistemas y al personal de CYAD que se encuentra trabajando en nuestras instalaciones que va a quedar cerrado el laboratorio de base de datos por remodelación y cambio de infraestructura de red  
ATENTAMENTE  
"CASA ABIERTA AL TIEMPO"  
Risto Fermin Rangel Kuoppa  
Profesor Investigador  
admin

documentos/Elaborados/49 #laborado.pdf

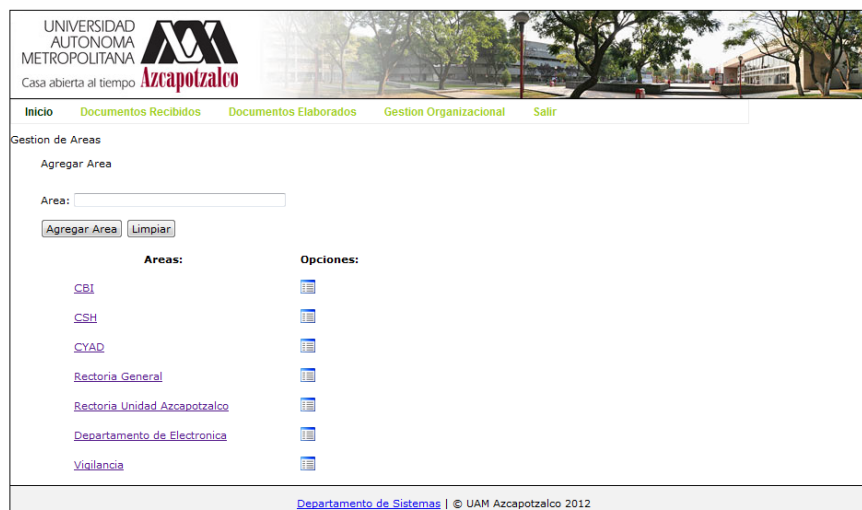
## Gestion Organizacional (Directorio)

### Agregar Areas

Se selecciona del menu principal la opcion **Gestion Organizacional** del menu desplegable se selecciona Areas.

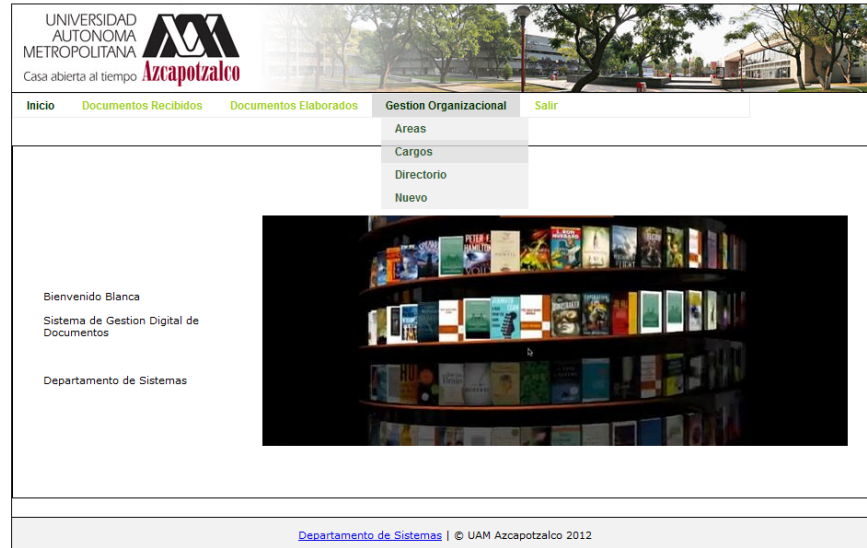


Se despliega la pantalla de gestión de Áreas donde se llena el formulario con el nombre del área para agregarla se selecciona el botón **Agregar Area**, se re direcciona a esta misma pantalla mostrando el área agregada.

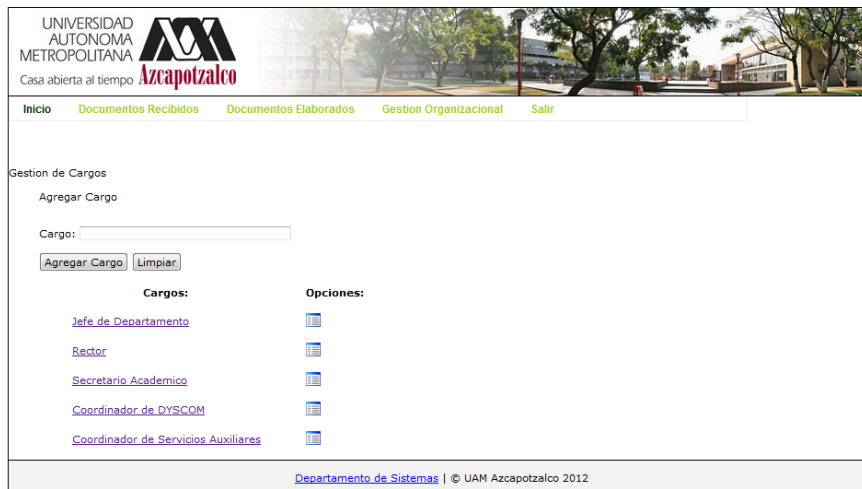


## Agregar Cargos

Se selecciona del menu principal la opcion **Gestion Organizacional** del menu desplegable se selecciona Cargos.

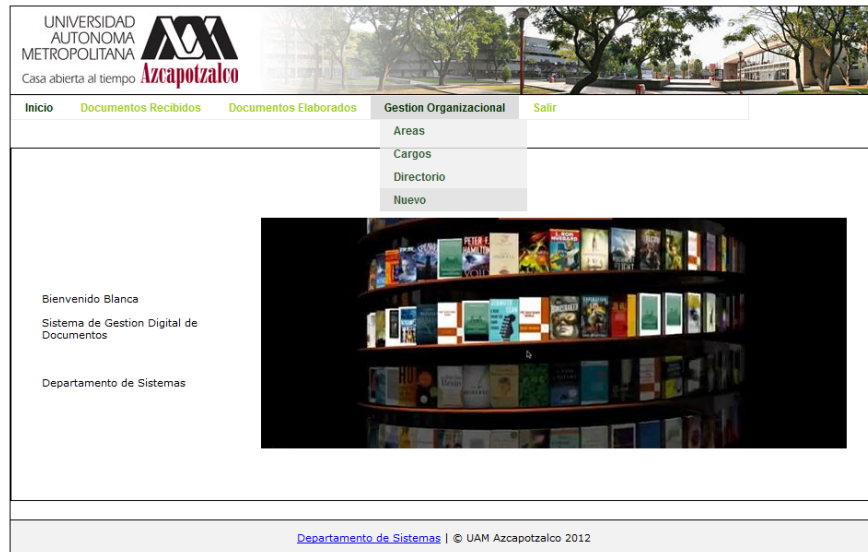


Se despliega la pantalla de gestión de Cargos donde se llena el formulario con el nombre del cargo para agregarlo se selecciona el botón **Agregar Cargo**, se re direcciona a esta misma pantalla mostrando el cargo agregado.



## Nuevo Responsable de la Gestión Organizacional (Directorio)

Se selecciona del menu principal la opcion **Gestion Organizacional** del menu desplegable se selecciona Nuevo



Se muestra la pantalla con un formulario donde se llenan los datos personales del responsable de, para agregarlo se selecciona la opción **Agregar Remitente**.

UNIVERSIDAD AUTONOMA METROPOLITANA Azcapotzalco

Casa abierta al tiempo

Inicio Documentos Recibidos Documentos Elaborados Gestion Organizacional Salir

Nombre:

Apellido Paterno:

Apellido Materno :

Email:

Teléfono:

Ubicacion:

Area:

Cargo:

Departamento de Sistemas | © UAM Azcapotzalco 2012


Se listan los Responsables Activos re direcciona a la pantalla siguiente con la lista de responsables . El adminsitrador del sistema tiene que activar el responsable agregado para que este aparesca en la lista del directorio.

| Nombre:                                      | E-Mail:  | Ubicacion:                          | Telefono: | Area:                            | Cargo:                               | Opciones: |
|--|--|-------------------------------------|-----------|----------------------------------|--------------------------------------|-----------|
| <a href="#">Jazmin Pastrana Avila</a>        | <a href="mailto:mimin2003@gmail.com">mimin2003@gmail.com</a>         | <a href="#">Edif T 2do Piso #23</a> | 98765432  | CYAD                             | <a href="#">Secretario Academico</a> | 2         |
| <a href="#">Juan Asdrubal Pastrana Cede?</a> | <a href="mailto:juan.pastrana@gmail.com">juan.pastrana@gmail.com</a> | <a href="#">Kza</a>                 | 43754248  | CBI                              | <a href="#">Rector</a>               | 2         |
| <a href="#">Ricardo Flores D</a>             | <a href="mailto:rd@gmail.com">rd@gmail.com</a>                       | <a href="#">Av. San Pablo</a>       | 98765432  | CBI                              | <a href="#">Jefe de Departamento</a> | 2         |
| <a href="#">Gabriel Soto Materno</a>         | <a href="mailto:login@hotmail.com">login@hotmail.com</a>             | <a href="#">HP 2do</a>              | 55223344  | <a href="#">Rectoria General</a> | <a href="#">Rector</a>               | 2         |
| <a href="#">Erika Avila Alonso</a>           | <a href="mailto:erikus@hotmail.com">erikus@hotmail.com</a>           | <a href="#">kza</a>                 | 36181410  | CYAD                             | <a href="#">Jefe de Departamento</a> | 2         |

[Departamento de Sistemas](#) | © UAM Azcapotzalco 2012

## Salir del sistema

Se tiene que seleccionar la opción **Salir** del menú principal y la sesión se finaliza. y se re direcciona a la pantalla de login

| Inicio   | Documentos Recibidos | Documentos Elaborados | Gestion Organizacional | Salir |
|--|----------------------|-----------------------|------------------------|-------|
| BIENVENIDO INICIE SESION   |                      |                       |                        |       |
| Sistema de Gestion Digital de Documentos   |                      |                       |                        |       |
|                              |                      |                       |                        |       |
| Usuario: <input type="text"/><br>Contraseña: <input type="password"/><br><input type="button" value="Ingresar"/> |                      |                       |                        |       |
| <a href="#">Departamento de Sistemas</a>   © UAM Azcapotzalco 2012   |                      |                       |                        |       |

## Manual de Administrador del sistema

### Administración de Responsables

Se selecciona del menu principal la opcion **Gestion Organizacional** del menu desplegable se selecciona Directorio, se re despliega la pantalla siguiente con la lista de responsables . En la cual se proporciona en un encabezado las opciones Editar, Activar y desactivar responsable.

| UNIVERSIDAD AUTONOMA METROPOLITANA Azcapotzalco                                |  |   |           |                  |                       |           |
|--|--|---|-----------|------------------|-----------------------|-----------|
| Inicio Documentos Recibidos Documentos Elaborados Gestion Organizacional Salir |  |   |           |                  |                       |           |
| Administración de Responsables   |  |   |           |                  |                       |           |
| Nombre:  | E-Mail:  | Ubicacion:                              | Telefono: | Area:            | Cargo:                | Opciones: |
| <a href="#">Jazmin Pastrana Avila</a>  | <a href="mailto:mimin2003@gmail.com">mimin2003@gmail.com</a>               | <a href="#">Edif. T. 2do. Piso #23</a>  | 98765432  | CYAD             | Secretario Academico  |           |
| <a href="#">Gloria Cervantes B</a>   | <a href="mailto:username@correo.azc.uam.mx">username@correo.azc.uam.mx</a> | <a href="#">Edif. C. 2do piso</a>       | 44332211  | CSH              | Jefe de Departamento  |           |
| <a href="#">Mario A. Rocha Peresz</a>  | <a href="mailto:live@hotmail.com">live@hotmail.com</a>                     | <a href="#">Uam Azc</a>                 | 12345678  | Rectoria General | Coordinador de DYSCOM |           |
| <a href="#">Juan Asdrubal Pastrana Cede?</a>                                   | <a href="mailto:juan.pastrana@gmail.com">juan.pastrana@gmail.com</a>       | <a href="#">Kza</a>                     | 43754248  | CBI              | Rector                |           |
| <a href="#">Ricardo Flores D</a>   | <a href="mailto:id@gmail.com">id@gmail.com</a>                             | <a href="#">Av. San Pablo</a>           | 98765432  | CBI              | Jefe de Departamento  |           |
| <a href="#">Dr. Luis E. Nore? Franco</a>                                       | <a href="mailto:dircbi@correo.azc.uam.mx">dircbi@correo.azc.uam.mx</a>     | <a href="#">Edificio H-P, 1er. Piso</a> | 5318 9001 | CBI              | Jefe de Departamento  |           |
| <a href="#">Gabriel Soto Materno</a>   | <a href="mailto:login@hotmail.com">login@hotmail.com</a>                   | <a href="#">HP 2do</a>                  | 55223344  | Rectoria General | Rector                |           |
| <a href="#">Erika Avila Alonso</a>   | <a href="mailto:erikus@hotmail.com">erikus@hotmail.com</a>                 | <a href="#">kza</a>                     | 36181410  | CYAD             | Jefe de Departamento  |           |



## Modificar información del remitente

El administrador puede cambiar los datos de los responsables así como el área y cargo en la que se encuentran una vez que los datos son modificados en el formulario si se desea guardar los cambios se selecciona el botón **Modificar Responsables**.

UNIVERSIDAD AUTÓNOMA METROPOLITANA Azcapotzalco

Casa abierta al tiempo

Inicio Documentos Recibidos Documentos Elaborados Gestion Organizacional Salir

Nombre:  Antes: Dr. Luis E.

Apellido Paterno:  Antes: Nore?

Apellido Materno:  Antes: Franco

Email:  Antes: dircbi@correo.azc.uam.mx

Teléfono:  Antes: 5318 9001

Ubicación:  Antes: Edificio H-P, 1er. Piso

Area:  Antes: CBI

Cargo:  Antes: Jefe de Departamento

Departamento de Sistemas | © UAM Azcapotzalco 2012

Se despliega la pantalla con el directorio y se muestran los cambios que se realizaron a el responsable.

UNIVERSIDAD AUTÓNOMA METROPOLITANA Azcapotzalco

Casa abierta al tiempo

Inicio Documentos Recibidos Documentos Elaborados Gestion Organizacional Salir


| Nombre:                                      | E-Mail:  | Ubicación:                              | Telefono:                             | Area:                            | Cargo:                               | Opciones: |
|--|--|---|---------------------------------------|----------------------------------|--------------------------------------|-----------|
| <a href="#">Jazmin Pastrana Avila</a>        | <a href="mailto:mimin2003@gmail.com">mimin2003@gmail.com</a>           | <a href="#">Edif T 2do Piso #23</a>     | <a href="tel:98765432">98765432</a>   | <a href="#">CYAD</a>             | <a href="#">Secretario Academico</a> | 2         |
| <a href="#">Juan Asdrubal Pastrana Cede?</a> | <a href="mailto:juan.pastrana@gmail.com">juan.pastrana@gmail.com</a>   | <a href="#">Kza</a>                     | <a href="tel:43754248">43754248</a>   | <a href="#">CBI</a>              | <a href="#">Rector</a>               | 2         |
| <a href="#">Ricardo Flores D</a>             | <a href="mailto:id@gmail.com">id@gmail.com</a>                         | <a href="#">Av. San Pablo</a>           | <a href="tel:98765432">98765432</a>   | <a href="#">CBI</a>              | <a href="#">Jefe de Departamento</a> | 2         |
| <a href="#">Dr. Luis E. Nore? Franco</a>     | <a href="mailto:dircbi@correo.azc.uam.mx">dircbi@correo.azc.uam.mx</a> | <a href="#">Edificio H-P, 1er. Piso</a> | <a href="tel:5318 9001">5318 9001</a> | <a href="#">CBI</a>              | <a href="#">Jefe de Departamento</a> | 2         |
| <a href="#">Gabriel Soto Materno</a>         | <a href="mailto:login@hotmail.com">login@hotmail.com</a>               | <a href="#">HP 2do</a>                  | <a href="tel:55223344">55223344</a>   | <a href="#">Rectoria General</a> | <a href="#">Rector</a>               | 2         |
| <a href="#">Erika Avila Alonso</a>           | <a href="mailto:erikus@hotmail.com">erikus@hotmail.com</a>             | <a href="#">kza</a>                     | <a href="tel:36181410">36181410</a>   | <a href="#">CYAD</a>             | <a href="#">Jefe de Departamento</a> | 2         |

Departamento de Sistemas | © UAM Azcapotzalco 2012

El administrador tiene que activar al responsable una vez que este fue dado de alta o se modifico su información personal con el botón que tiene un circulo color verde con una paloma blanca (opción de en medio).

- Desactivar:
- Activar:
- Modificar:




| METROPOLITANA  <b>Azacapotzalco</b><br>Casa abierta al tiempo |  |   |                        |                  |                       |           |
|--|--|---|------------------------|------------------|-----------------------|-----------|
| Inicio   | Documentos Recibidos   | Documentos Elaborados                   | Gestion Organizacional | Salir            |                       |           |
| Administración de Responsables   |  |   |                        |                  |                       |           |
| Nombre:  | E-Mail:  | Ubicacion:                              | Telefono:              | Area:            | Cargo:                | Opciones: |
| <a href="#">Jazmin Pastrana Avila</a>  | <a href="mailto:mimin2003@gmail.com">mimin2003@gmail.com</a>               | <a href="#">Edif T 2do Piso #23</a>     | 98765432               | CYAD             | Secretario Academico  | <br>2     |
| <a href="#">Gloria Cervantes B</a>   | <a href="mailto:username@correo.azc.uam.mx">username@correo.azc.uam.mx</a> | <a href="#">Edif. C 2do piso</a>        | 44332211               | CSH              | Jefe de Departamento  | <br>0     |
| <a href="#">Mario A. Rocha Perez</a>   | <a href="mailto:live@hotmail.com">live@hotmail.com</a>                     | <a href="#">Uam Azc</a>                 | 12345678               | Rectoria General | Coordinador de DYSCOM | <br>0     |
| <a href="#">Juan Asdrubal Pastrana Cede?</a>   | <a href="mailto:juan.pastrana@gmail.com">juan.pastrana@gmail.com</a>       | <a href="#">Kza</a>                     | 43754248               | CBI              | Rector                | <br>2     |
| <a href="#">Ricardo Flores D</a>   | <a href="mailto:rd@gmail.com">rd@gmail.com</a>                             | <a href="#">Av. San Pablo</a>           | 98765432               | CBI              | Jefe de Departamento  | <br>2     |
| <a href="#">Dr. Luis E. Nore? Franco</a>   | <a href="mailto:dircbi@correo.azc.uam.mx">dircbi@correo.azc.uam.mx</a>     | <a href="#">Edificio H-P, 1er. Piso</a> | 5318 9001              | CBI              | Jefe de Departamento  | <br>0     |
| <a href="#">Gabriel Soto Materno</a>   | <a href="mailto:login@hotmail.com">login@hotmail.com</a>                   | <a href="#">HP 2do</a>                  | 55223344               | Rectoria General | Rector                | <br>2     |
| <a href="#">Erika Avila Alonso</a>   | <a href="mailto:erikus@hotmail.com">erikus@hotmail.com</a>                 | <a href="#">kza</a>                     | 36181410               | CYAD             | Jefe de Departamento  | <br>2     |
| <a href="#">Departamento de Sistemas</a>   © UAM Azcapotzalco 2012   |  |   |                        |                  |                       |           |

Una vez que el administrador active al responsable el usuario lo puede visualizar y utilizar en su entorno de trabajo.

## Salir del sistema

Se tiene que seleccionar la opción **Salir** del menú principal y la sesión se finaliza, y se re direcciona a la pantalla de login

| UNIVERSIDAD AUTONOMA METROPOLITANA  <b>Azacapotzalco</b><br>Casa abierta al tiempo |                      |  |                        |       |  |  |
|---|----------------------|--|------------------------|-------|--|--|
| Inicio  | Documentos Recibidos | Documentos Elaborados  | Gestion Organizacional | Salir |  |  |
| BIENVENIDO INICIE SESION  |                      |  |                        |       |  |  |
| Sistema de Gestion Digital de Documentos  |                      |  |                        |       |  |  |
| Usuario: <input type="text"/><br>Contraseña: <input type="password"/><br><input type="button" value="Ingresar"/>  |                      |  |                        |       |  |  |
| <a href="#">Departamento de Sistemas</a>   © UAM Azcapotzalco 2012  |                      |  |                        |       |  |  |

## Manual de instalación

### Prerequisitos

- Gestor de base de datos PostgreSQL
- Máquina virtual de java JRE
- Servidor de aplicación web Apache Tomcat

### Instalación del sistema del sistema de gestión web Comunicación

Se debe ejecutar el scrip de la base de datos en postgresQL

**gestion.sql**

Copiar el archivo **gestionWeb.war** al servidor Tomcat en el directorio:

**apache-tomcat-7.0.26/webapps/ gestionWeb.war**

**iniciar el servidor de aplicaciones con el comando**

**./apache-tomcat-7.0.26/webapps/ startup.sh**

**Entrar al siguiente link dentro del servidor para verificar que el servidor este levantado correctamente**

**El sistema tiene configurada la clave del usuario postgres en el archivo spring -dev.xml**

**El usuario con figurado es: postgres**

**Y la contraseña: AdmCompnt01**

## Bibliografía

- [1] *Proyectos Terminales Ing. Computación*, [en línea], [www.cbi.azc.uam.mx](http://www.cbi.azc.uam.mx), (Fecha de consulta: 19/JUN/2011).
- [2] C. A. Flores Casillas, “Implementación de un Gestor de Documentos”, Proyecto Terminal Ingeniería en Computación Universidad Autónoma Metropolitana Unidad Azcapotzalco, Distrito Federal, México, 2011.
- [3] *Inforouter*, [en línea], <http://www.inforouter.com/>, (Fecha de consulta: 28/NOV/2011).
- [4] Document Management, [en línea], <http://www.estrasol.com.mx/document-management.php>, (Fecha de consulta: 28/NOV/2011).
- [5] *Eclipse*, [en línea], <http://eclipse.org>, (Fecha de consulta: 28/NOV/2011).
- [6] *Tomcat*, [en línea], <http://tomcat.apache.org/> (Fecha de consulta: 28/NOV/2011).
- [7] *Postgresql*, [en línea], <http://www.postgresql.org/>, (Fecha de consulta: 28/NOV/2011).
- [8] *Spring*, [en línea], <http://www.springsource.org/>, (Fecha de consulta: 28/NOV/2011).
- [9] *Hibernate*, [en línea], <http://www.hibernate.org/>, (Fecha de consulta: 28/NOV/2011).
- [10] *Struts*, [en línea], <http://struts.apache.org/>, (Fecha de consulta: 28/NOV/2011).