

**Universidad Autónoma Metropolitana
Unidad Azcapotzalco**

**División de Ciencias Básicas e Ingeniería
Licenciatura en Ingeniería en Computación**

Reporte de Proyecto Terminal

**Sistema de retroalimentación de unidades de enseñanza aprendizaje (UEA) por parte
del profesor.**

Alumno: Villalba Gallardo Miriam
Matrícula: 206241321



Firma del alumno

Trimestre 14-I
Fecha de entrega: 10 de abril del 2014

Asesora
M.C. Rafaela Blanca Silva López
Titular "B"
Departamento de Sistemas



Firma del asesor

Yo, M.C. Rafaela Blanca Silva López , declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

A handwritten signature in black ink, appearing to read 'Rafaela S.L.', written over a horizontal line.

Firma del asesor

Yo, Villalba Gallardo Miriam, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

A handwritten signature in black ink, appearing to read 'Miriam', written over a horizontal line.

Firma del alumno

Tabla de contenido

Resumen	4
Introducción	4
Justificación.....	4
Antecedentes	5
Referencias Internas.....	5
Referencias Externas.....	5
Objetivo	7
Objetivo General.....	7
Objetivos Específicos	7
Marco teórico	8
Análisis.....	8
Desarrollo del Proyecto.....	8
Descripción Técnica	8
Especificación Técnica	11
Entorno de desarrollo	11
Entregables.....	12
Diagrama de Entidad Relación	12
Esquema de la Base de Datos	13
Diccionario de datos	18
Diagrama de actividades.....	20
Diagrama de navegación.....	22
Código de Ejemplo de cómo esta constituido el funcionamiento de la gestión de Usuarios en el sistema.....	26
Clase Usuario	26
Clase Action Usuario	29
Implementacion de Usuarios.....	31
Forma Usuario (jsp)	32
Manual de Instalación y Configuración.....	38
Descargar Java.....	38
Instalación de Eclipse.....	40
Descargar Struts2	41
Descargar Apache Tomcat	42
<i>Levantar apache tomcat.....</i>	<i>44</i>
Descargar Hibernate.....	45
Manual de Usuario	48
Resultados	52
Conclusión	53
Bibliografía.....	53

Resumen

El siguiente documento redacta de manera detallada como está constituido el **“Sistema de retroalimentación e unidades de enseñanza y aprendizaje (UEA) por parte del profesor”**, que pretende agilizar y facilitar la evaluación de las UEA para así poder detectar problemas que mejoren la calidad de la enseñanza y por consiguiente en este documento se muestran los módulos de cómo fue planeado y desarrollado el sistema, detallando elementos de su constitución y funcionamiento.

Introducción

Actualmente el profesor envía comentarios de las UEA que impartió a través de una hoja que se entrega con su acta de la UEA, sin embargo su análisis y detección de problemas para darle solución es complicado, por lo que en este proyecto se propone implementar un sistema web que permita a los profesores de la División de Ciencias Básicas e Ingeniería de la UAM-A realizar evaluaciones del entorno educativo como es el material didáctico, aula, equipo y recursos educativos, lo que permitirá tener información inmediata y poder resolver los problemas recurrentes que sean detectados.

Justificación

El sistema de información propuesto pretende agilizar y facilitar la evaluación de unidades de enseñanza-aprendizaje (UEA), que actualmente se realiza llenando hojas de evaluación al concluir el trimestre, lo cual hace el proceso poco eficiente, ya que toda la información se encuentra registrada en papel y es complicado analizarla y generar reportes rápidamente que permitan la toma de decisiones para actualizar equipo de computo, mejorar la iluminación de la sala o salón de clases, contar con aire acondicionado y elaborar material didáctico u otros recursos que sean necesarios para mejorar la calidad del proceso de enseñanza-aprendizaje del alumno.

Una vez implementado en su totalidad, los profesores y los jefes de departamento tendrán a su disposición una herramienta que les facilite y permita llevar a cabo la evaluación de UEA en las diferentes asignaturas de un profesor.

El diseño y la implementación de este sistema requieren de conocimiento en programación, análisis, bases de datos y diseño de sistemas, como también de la habilidad de investigación y autoaprendizaje de herramientas para realizar una serie de actividades. Los conocimientos y habilidades se adquieren a lo largo de la carrera de Ingeniería en Computación, por lo que este proyecto no podría ser concluido satisfactoriamente por una persona que no tuviera este perfil.

Antecedentes

Referencias Internas

Existen proyectos internos orientados a la evaluación tales como:

1. “Evaluación de programas fuente en lenguaje C con entradas y salidas descritas mediante una cadena de formato” [1] . El proyecto fue realizado en el trimestre 07-P , y está orientada a la evaluación de programas fuentes en lenguaje C, pero el proyecto no cuenta con instrumentos de evaluación.
2. “Análisis estadístico de evaluaciones docentes utilizando modelos de ecuaciones estructurales”[2] . El proyecto fue realizado en el trimestre 10-I ,y está dirigido a realizar evaluaciones al personal docente e instalaciones, pero no cuenta con los instrumentos de evaluación definidos por expertos.
3. “Sistema de evaluaciones dinámicas del proceso de enseñanza aprendizaje basado en la opinión de los alumnos con doble interfaz: desktop web y mobile web” [3] . Este proyecto fue realizado en el trimestre 10-O. Lleva a cabo una evaluación dinámica del proceso de enseñanza-aprendizaje basándose en la opinión del alumno, pero no está orientado a evaluaciones realizadas por profesores.
4. “Sistema de gestión de calificaciones para los cursos impartidos por un profesor”
5. [4] . Este proyecto fue realizado en el trimestre 11-P. En, a través de una serie de opciones, un profesor pueda llevar el control de calificaciones de sus grupos. Sin embargo, este sistema está dirigido a los profesores, pero no realiza evaluaciones de las UEA que los profesores imparten en el trimestre, como tampoco realiza evaluaciones de su entorno y recursos.

Referencias Externas

Actualmente existen Software que cumplen con el funcionamiento de evaluación, los cuales son:

MOODLE [5]: Es un sistema de gestión de cursos de código abierto que cuenta con herramientas específicas para evaluar el proceso de enseñanza-aprendizaje, pero no está orientado a la evaluación de UEA.

MENTOR[6]: Es una herramienta comercial que permite realizar evaluaciones al personal de una empresa , instalaciones o equipo, pero no está orientado a la evaluación de UEA, que se incluye en esta propuesta.

QTRAINING[7]: Es una herramienta comercial que ayuda a los altos mandos a saber en qué áreas se requiere mayor capacitación, ya que está enfocada a la evaluación del personal de una empresa, pero no orientada al proceso de enseñanza-aprendizaje.

Aunque existen proyectos enfocados a la evaluación, como los anteriores y éstos no cubren la evaluación de UEA por parte de los profesores al concluir el trimestre. A continuación se muestra una tabla comparativa con los distintos proyectos y Software Vs esta propuesta (*Tabla 1*). Lo que se plantea en esta propuesta es la evaluación de UEA aplicando instrumentos de evaluación definidos por expertos y la generación de reportes que facilitan el análisis y permiten la toma de decisiones para mejorar la calidad educativa.

Tabla 1. Tabla comparativa de características de los distintos proyectos y Software Vs esta propuesta.

Antecedentes Internos					
Título	Enfocados a la Evaluación	Enfocado al proceso de enseñanza-aprendizaje	Genera reportes	Enfocado a la evaluación de UEA	El profesor evalúa el espacio y el entorno educativo
“Evaluación de programas fuente en lenguaje C con entradas y salidas descritas mediante una cadena de formato”. [1]	Sí	No	No	No	No
“Análisis estadístico de evaluaciones docentes utilizando modelos de ecuaciones estructurales”.[2]	Sí	Sí	Sí	No	No
“Sistema de evaluaciones dinámicas del proceso de enseñanza aprendizaje basado en la opinión de los alumnos con doble interfaz: desktop web y mobile web”. [3]	Sí	Sí	No	No	No
“Sistema de gestión de calificaciones para los cursos impartidos por un profesor”. [4]	Sí	Sí	Sí	No	No

Antecedentes Externos					
MOODLE [5]	Sí	Sí	Sí	No	No
MENTOR [6]	Sí	Sí	Sí	No	No
QTRAINING [7]	Sí	No	Sí	No	No
Sistema de Información propuesto					
“Sistema de retroalimentación de unidades de enseñanza aprendizaje (UEA) por parte del profesor.”	Sí	Sí	Sí	Sí	Sí

Objetivo

Este Proyecto forma parte del Sistema de Información para la Evaluación de la Calidad Educativa.

Objetivo General

Diseñar e implementar un Sistema de Información Web que permita a los profesores de la División de Ciencias Básicas e Ingeniería de la UAM-Azcapotzalco evaluar el contexto del proceso de enseñanza-aprendizaje de las UEA que imparten en el trimestre, aplicando instrumentos de evaluación definidos por expertos, que permitan evaluar:
El espacio donde se imparte la clase, el material educativo y recursos adicionales utilizados para llevar a cabo el proceso de enseñanza-aprendizaje, el equipo de computo en caso de usarlo o plataforma educativa en caso de usarse.

Objetivos Específicos

- Diseñar e implementar la Base de Datos Relacional del sistema de evaluación educativa.
- Diseñar e implementar el módulo de evaluación del proceso de enseñanza-aprendizaje por parte del profesor permitiendo evaluar el espacio o entorno educativo (aula, equipo y material didáctico).
- Diseñar e implementar el módulo de evaluación del proceso de enseñanza-aprendizaje que permita a los profesores evaluar los recursos de enseñanza (recursos educativos e instrumentos de evaluación).
- Diseñar e implementar el módulo de generación de reportes que permita al jefe de departamento realizar reportes de las evaluaciones por UEA, profesor y profesor-UEA.

Marco teórico

Se requiere de conocimientos en el manejo de SQL, Struts , Java, Hibernate y HTML para el desarrollo del sistema y de realizar entrevistas al jefe de Departamento de Sistemas para poder establecer los criterios necesarios para el diseño de reportes generados por el sistema que le faciliten analizar y poder detectar problemas en el entorno y los recursos.

Análisis

Para la realización de este proyecto su elaboración se planeo en el desarrollo de tres módulos, que fueron desarrollados de forma separada para posteriormente realizar pruebas de funcionamiento y finalmente integrarlos en un solo proyecto.

Desarrollo del Proyecto

Descripción Técnica

El “Sistema de retroalimentación de unidades de enseñanza aprendizaje (UEA) por parte del profesor” será utilizado por medio de una interfaz Web, por lo que el usuario podrá acesar al sistema desde un navegador y obtener la información requerida rápidamente.

El diagrama de casos de uso del sistema se muestra en la **Figura 1**.

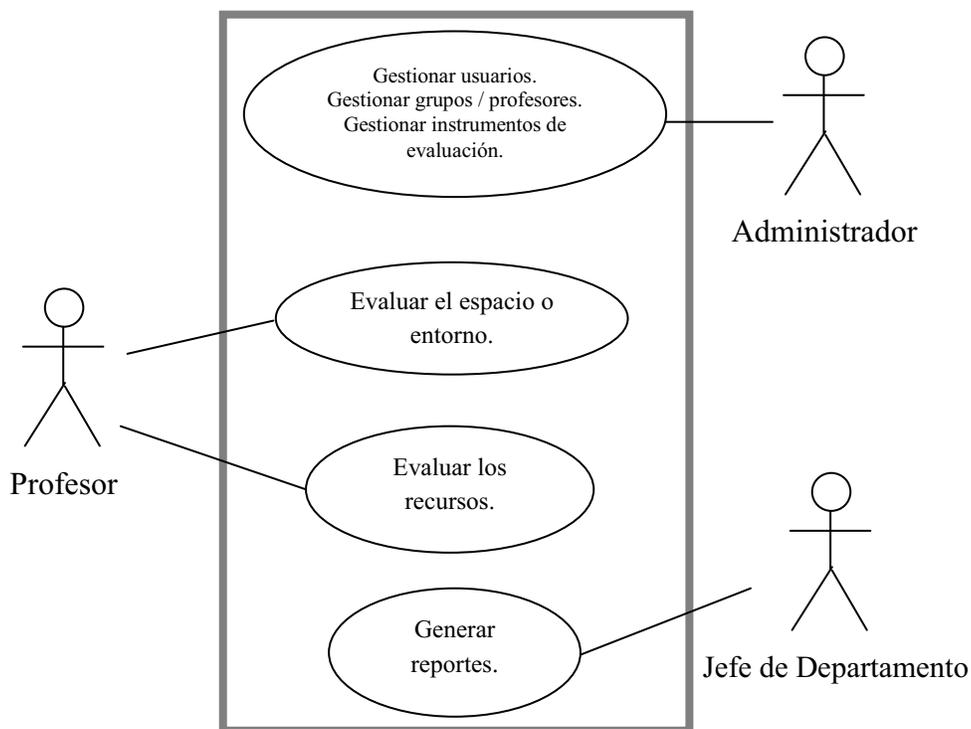


Figura1 : Diagrama General del sistema

Los módulos que constituyen el proyecto son:

- **Módulo de Base de Datos Relacional del sistema de evaluación.**

Permite la interconexión entre los datos que se encuentran en las tablas diseñadas con los atributos necesarios para la evaluación, lo cual permite realizar la creación de tablas, las consultas y la captura de datos para las tablas.

- **Módulo de Evaluación del entorno.**

Este módulo permitirá al profesor evaluar su entorno de trabajo, como es el caso del aula, el equipo y el material didáctico, a través de un instrumento de evaluación constituido por preguntas definidas por expertos. Éstas ayudarán a determinar qué es lo que hace falta en el entorno para así mejorar la calidad educativa, ya que no contar con un entorno adecuado afectará la calidad del alumno.

- **Módulo de Evaluación de los recursos.**

En este módulo se llevará a cabo la evaluación de los recursos educativos e instrumentos de evaluación, permitiendo al profesor evaluar los recursos con los que cuenta para impartir su clase.



Figura 2 : Caso de uso de la operaciones realizadas por el Profesor

- **Módulo de Generación de reportes.**

En este módulo se generarán reportes de las evaluaciones registradas en el sistema, ya sea por UEA, profesor y profesor-UEA, que pueden ser generados por trimestre o año y, así, facilitar su análisis para la toma de decisiones que mejore el proceso de enseñanza-aprendizaje. La generación de reportes puede realizarse como se muestra en la **Figura 3**.

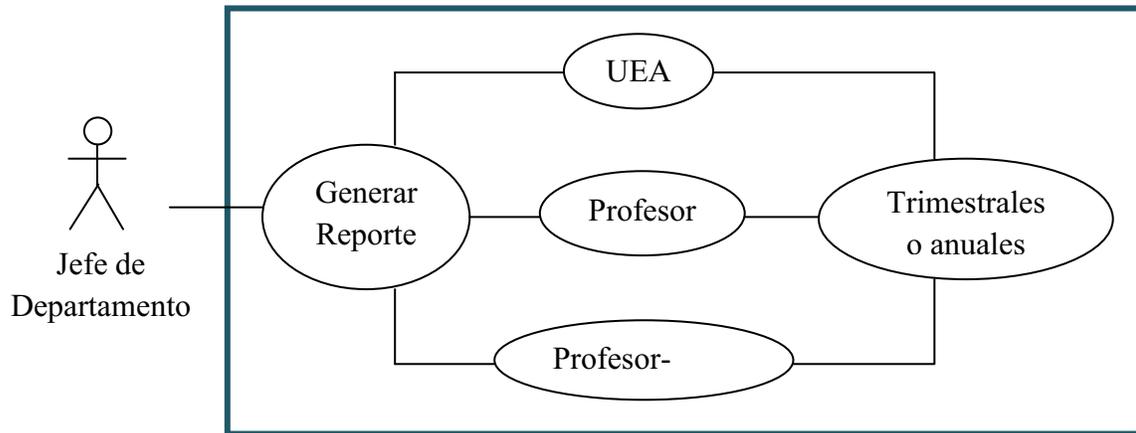


Figura 3. Caso de uso de la generación de reportes.

Para mostrar cómo se lleva a cabo el proceso de evaluación en el sistema se muestra la **Figura 3**.

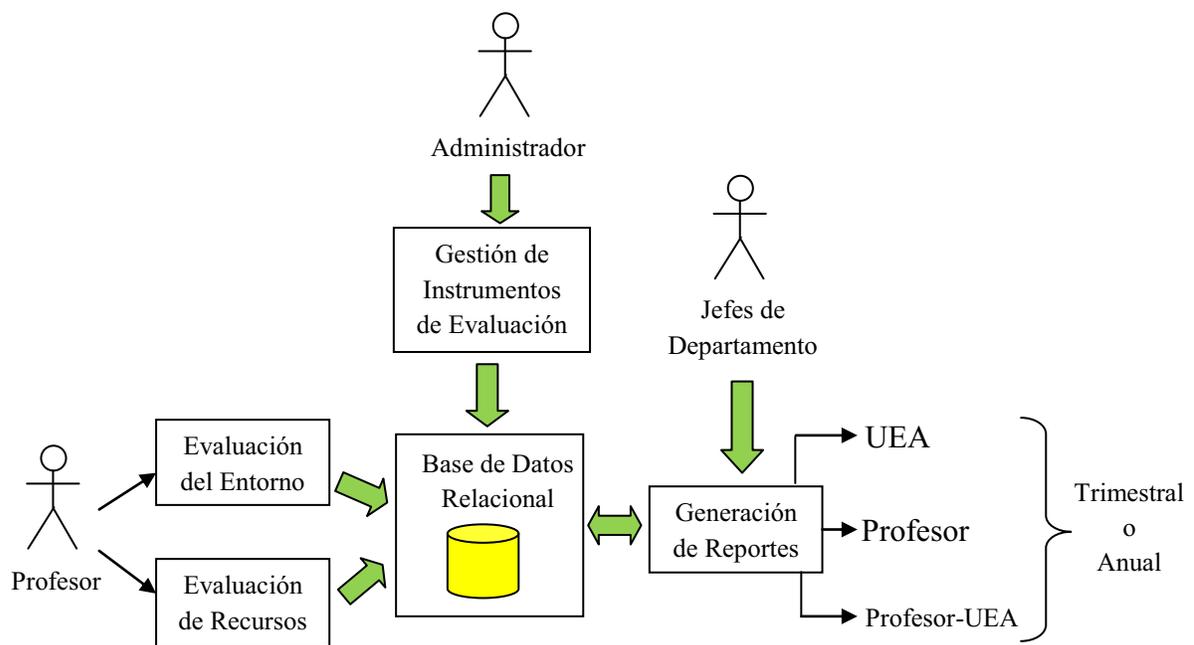


Figura 4. Diagrama del proceso de evaluación.

Especificación Técnica

Entorno de desarrollo

Para lograr el desarrollo del sistema y el cumplimiento de los objetivos descritos anteriormente, el sistema utilizará el lenguaje de programación *Java*¹ así como con las herramientas de *Apache Struts*² y *Hibernate*³, para obtener la compatibilidad del sistema con otros. El desarrollo será sobre el sistema operativo *Windows*, y la base de datos será manejada con lenguaje *SQL*⁴ en el entorno de *MySQL*.⁵

Se hará uso del siguiente Software:

- NetBeans6 IDE 6.0.1 para el desarrollo de aplicaciones.
- Apache-tomcat⁷ 7.0 para sistemas de 32bits.

El proyecto se dará por terminado cuando se entregue la implementación de los módulos anteriormente mencionados y se realicen las siguientes pruebas:

- Realizar una alta, baja y cambio de una materia.
- Realizar una alta, baja y cambio de un profesor.
- Realizar una evaluación del entorno.
- Realizar una evaluación del los recursos.
- Generar un reporte.

¹ *Java*: Es un lenguaje de programación orientado a objetos.

² *Apache Struts*: Es una herramienta de soporte para el desarrollo de aplicaciones Web.

³ *Hibernate*: Es una herramienta de mapeo objeto-relacional (ORM) de atributos entre una base de datos relacional y el modelo de objetos de una aplicación.

⁴ *SQL*: Es un lenguaje de consulta estructurado de acceso a base de datos relacionales.

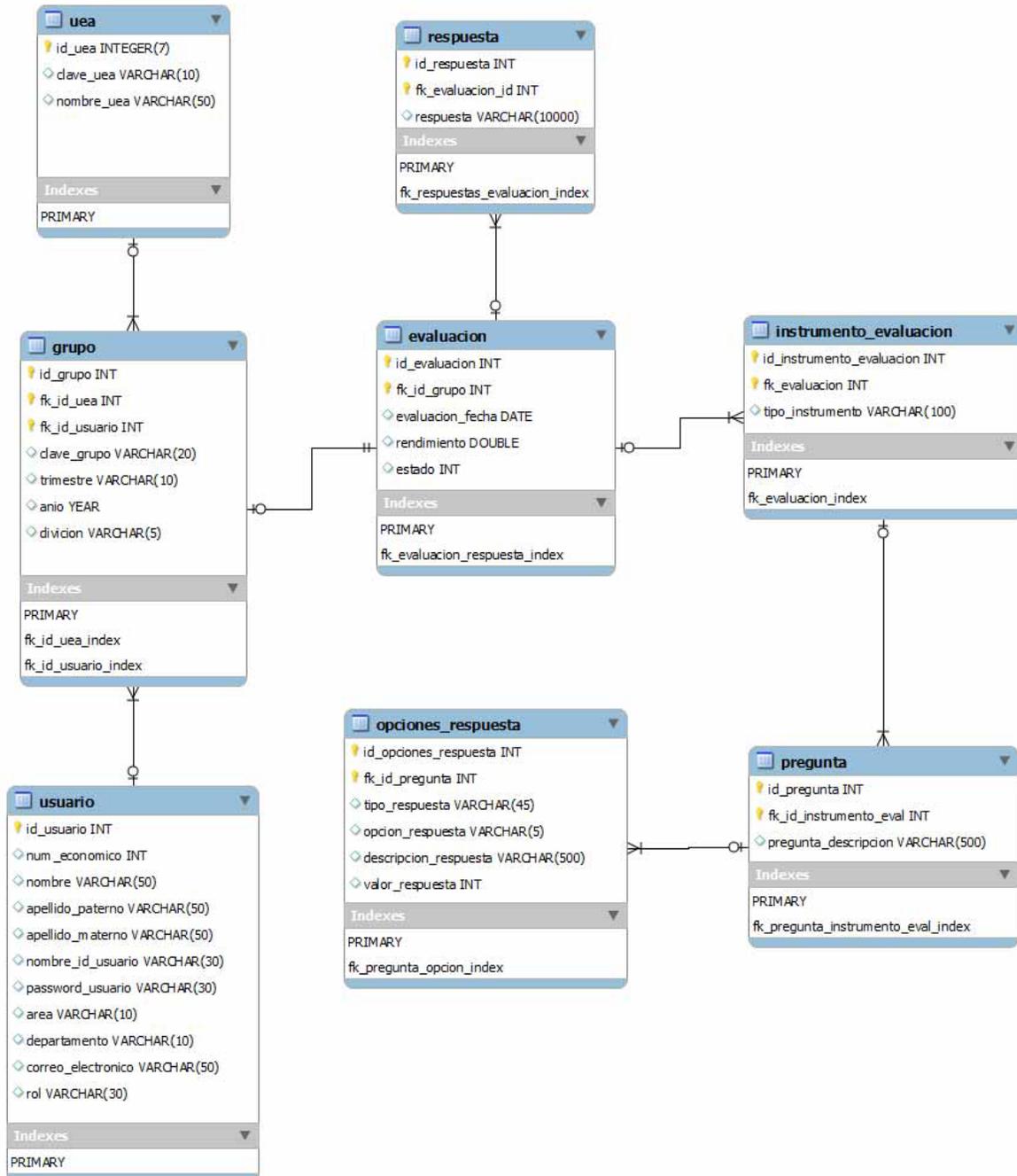
⁵ *MySQL*: Es un sistema de gestión de bases de datos relacionales.

⁶ *NetBeans*: Es un entorno de desarrollo integrado libre.

⁷ *Apache Tomcat*: Es un servidor Web con soporte de *Servlets*(que son programas que se ejecutan en un servidor).

Entregables

Diagrama de Entidad Relación



Esquema de la Base de Datos

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';
```

```
DROP SCHEMA IF EXISTS `22sep13` ;
CREATE SCHEMA IF NOT EXISTS `22sep13` DEFAULT CHARACTER SET latin1
COLLATE latin1_spanish_ci ;
USE `22sep13` ;
```

```
-----
-- Table `22sep13`.`usuario`
-----
```

```
DROP TABLE IF EXISTS `22sep13`.`usuario` ;
```

```
CREATE TABLE IF NOT EXISTS `22sep13`.`usuario` (
  `id_usuario` INT NOT NULL AUTO_INCREMENT,
  `num_economico` INT NULL,
  `nombre` VARCHAR(50) CHARACTER SET 'utf8' COLLATE 'utf8_general_ci' NULL,
  `apellido_paterno` VARCHAR(50) CHARACTER SET 'utf8' COLLATE
'utf8_general_ci' NULL,
  `apellido_materno` VARCHAR(50) CHARACTER SET 'utf8' COLLATE
'utf8_general_ci' NULL,
  `nombre_id_usuario` VARCHAR(30) CHARACTER SET 'utf8' COLLATE
'utf8_general_ci' NULL,
  `password_usuario` VARCHAR(30) CHARACTER SET 'utf8' COLLATE
'utf8_general_ci' NULL,
  `area` VARCHAR(10) CHARACTER SET 'utf8' COLLATE 'utf8_general_ci' NULL,
  `departamento` VARCHAR(10) CHARACTER SET 'utf8' COLLATE 'utf8_general_ci'
NULL,
  `correo_electronico` VARCHAR(50) CHARACTER SET 'utf8' COLLATE
'utf8_general_ci' NULL,
  `rol` VARCHAR(30) CHARACTER SET 'utf8' COLLATE 'utf8_general_ci' NULL,
  PRIMARY KEY (`id_usuario`))
ENGINE = InnoDB;
```

-- Table `22sep13`.`uea`

DROP TABLE IF EXISTS `22sep13`.`uea` ;

```
CREATE TABLE IF NOT EXISTS `22sep13`.`uea` (  
  `id_uea` INT(7) NOT NULL AUTO_INCREMENT,  
  `clave_uea` VARCHAR(10) CHARACTER SET 'utf8' COLLATE 'utf8_general_ci'  
  NULL COMMENT 'Clave uea',  
  `nombre_uea` VARCHAR(50) CHARACTER SET 'utf8' COLLATE 'utf8_general_ci'  
  NULL,  
  PRIMARY KEY (`id_uea`))  
ENGINE = InnoDB;
```

-- Table `22sep13`.`grupo`

DROP TABLE IF EXISTS `22sep13`.`grupo` ;

```
CREATE TABLE IF NOT EXISTS `22sep13`.`grupo` (  
  `id_grupo` INT NOT NULL AUTO_INCREMENT,  
  `fk_id_uea` INT NULL,  
  `fk_id_usuario` INT NULL,  
  `clave_grupo` VARCHAR(20) CHARACTER SET 'utf8' COLLATE 'utf8_general_ci'  
  NULL,  
  `trimestre` VARCHAR(10) CHARACTER SET 'utf8' COLLATE 'utf8_general_ci'  
  NULL,  
  `anio` YEAR NULL,  
  `divicion` VARCHAR(5) CHARACTER SET 'utf8' COLLATE 'utf8_general_ci' NULL,  
  PRIMARY KEY (`id_grupo`, `fk_id_usuario`, `fk_id_uea`),  
  CONSTRAINT `fk_id_uea`  
  FOREIGN KEY (`fk_id_uea`)  
  REFERENCES `22sep13`.`uea` (`id_uea`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION,  
  CONSTRAINT `fk_id_usuario`  
  FOREIGN KEY (`fk_id_usuario`)  
  REFERENCES `22sep13`.`usuario` (`id_usuario`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
CREATE INDEX `fk_id_uea_index` USING BTREE ON `22sep13`.`grupo` (`fk_id_uea`  
ASC);
```

```
CREATE INDEX `fk_id_usuario_index` ON `22sep13`.`grupo` (`fk_id_usuario` ASC);
```

-- Table `22sep13`.`evaluacion`

DROP TABLE IF EXISTS `22sep13`.`evaluacion` ;

```
CREATE TABLE IF NOT EXISTS `22sep13`.`evaluacion` (  
  `id_evaluacion` INT NOT NULL AUTO_INCREMENT,  
  `fk_id_grupo` INT NULL,  
  `evaluacion_fecha` DATE NULL,  
  `rendimiento` DOUBLE NULL DEFAULT 0,  
  `estado` INT NULL,  
  PRIMARY KEY (`id_evaluacion`, `fk_id_grupo`),  
  CONSTRAINT `fk_evaluacion_respuesta`  
    FOREIGN KEY (`fk_id_grupo`)  
    REFERENCES `22sep13`.`grupo` (`id_grupo`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
CREATE INDEX `fk_evaluacion_respuesta_index` ON `22sep13`.`evaluacion`  
(`fk_id_grupo` ASC);
```

-- Table `22sep13`.`instrumento_evaluacion`

DROP TABLE IF EXISTS `22sep13`.`instrumento_evaluacion` ;

```
CREATE TABLE IF NOT EXISTS `22sep13`.`instrumento_evaluacion` (  
  `id_instrumento_evaluacion` INT NOT NULL AUTO_INCREMENT,  
  `fk_evaluacion` INT NULL,  
  `tipo_instrumento` VARCHAR(100) CHARACTER SET 'utf8' COLLATE  
'utf8_general_ci' NULL,  
  PRIMARY KEY (`id_instrumento_evaluacion`, `fk_evaluacion`),  
  CONSTRAINT `fk_evaluacion`  
    FOREIGN KEY (`fk_evaluacion`)  
    REFERENCES `22sep13`.`evaluacion` (`id_evaluacion`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
CREATE INDEX `fk_evaluacion_index` ON `22sep13`.`instrumento_evaluacion`  
(`fk_evaluacion` ASC);
```

-- Table `22sep13`.`pregunta`

DROP TABLE IF EXISTS `22sep13`.`pregunta` ;

```
CREATE TABLE IF NOT EXISTS `22sep13`.`pregunta` (  
  `id_pregunta` INT NOT NULL AUTO_INCREMENT,  
  `fk_id_instrumento_eval` INT NULL,  
  `pregunta_descripcion` VARCHAR(500) CHARACTER SET 'utf8' COLLATE  
'utf8_general_ci' NULL,  
  PRIMARY KEY (`id_pregunta`, `fk_id_instrumento_eval`),  
  CONSTRAINT `fk_pregunta_instrumento_eval`  
  FOREIGN KEY (`fk_id_instrumento_eval`)  
  REFERENCES `22sep13`.`instrumento_evaluacion` (`id_instrumento_evaluacion`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
CREATE INDEX `fk_pregunta_instrumento_eval_index` ON `22sep13`.`pregunta`  
(`fk_id_instrumento_eval` ASC);
```

-- Table `22sep13`.`opciones_respuesta`

DROP TABLE IF EXISTS `22sep13`.`opciones_respuesta` ;

```
CREATE TABLE IF NOT EXISTS `22sep13`.`opciones_respuesta` (  
  `id_opciones_respuesta` INT NOT NULL AUTO_INCREMENT,  
  `fk_id_pregunta` INT NULL,  
  `tipo_respuesta` VARCHAR(45) CHARACTER SET 'utf8' COLLATE 'utf8_general_ci'  
  NULL,  
  `opcion_respuesta` VARCHAR(5) CHARACTER SET 'utf8' COLLATE 'utf8_general_ci'  
  NULL,  
  `descripcion_respuesta` VARCHAR(500) CHARACTER SET 'utf8' COLLATE  
'utf8_general_ci' NULL,  
  `valor_respuesta` INT NULL,  
  PRIMARY KEY (`id_opciones_respuesta`, `fk_id_pregunta`),  
  CONSTRAINT `fk_pregunta_opcion`  
  FOREIGN KEY (`fk_id_pregunta`)  
  REFERENCES `22sep13`.`pregunta` (`id_pregunta`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
CREATE INDEX `fk_pregunta_opcion_index` ON `22sep13`.`opciones_respuesta`  
(`fk_id_pregunta` ASC);
```

-- Table `22sep13`.`respuesta`

DROP TABLE IF EXISTS `22sep13`.`respuesta` ;

```
CREATE TABLE IF NOT EXISTS `22sep13`.`respuesta` (  
  `id_respuesta` INT NOT NULL AUTO_INCREMENT,  
  `fk_evaluacion_id` INT NULL,  
  `respuesta` VARCHAR(10000) CHARACTER SET 'utf8' COLLATE 'utf8_general_ci'  
  NULL,  
  PRIMARY KEY (`id_respuesta`, `fk_evaluacion_id`),  
  CONSTRAINT `fk_respuestas_evaluacio`  
  FOREIGN KEY (`fk_evaluacion_id`)  
  REFERENCES `22sep13`.`evaluacion` (`id_evaluacion`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
CREATE INDEX `fk_respuestas_evaluacion_index` ON `22sep13`.`respuesta`  
(`fk_evaluacion_id` ASC);
```

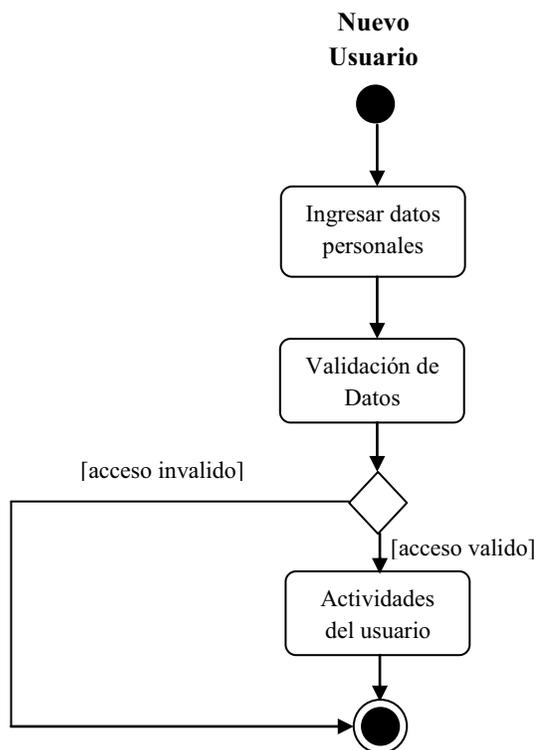
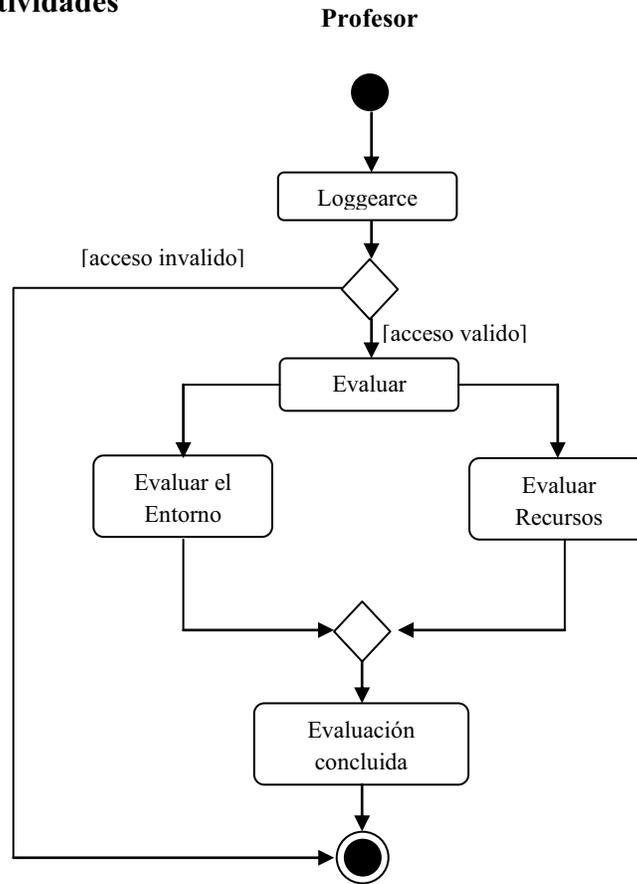
```
SET SQL_MODE=@OLD_SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;  
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Diccionario de datos

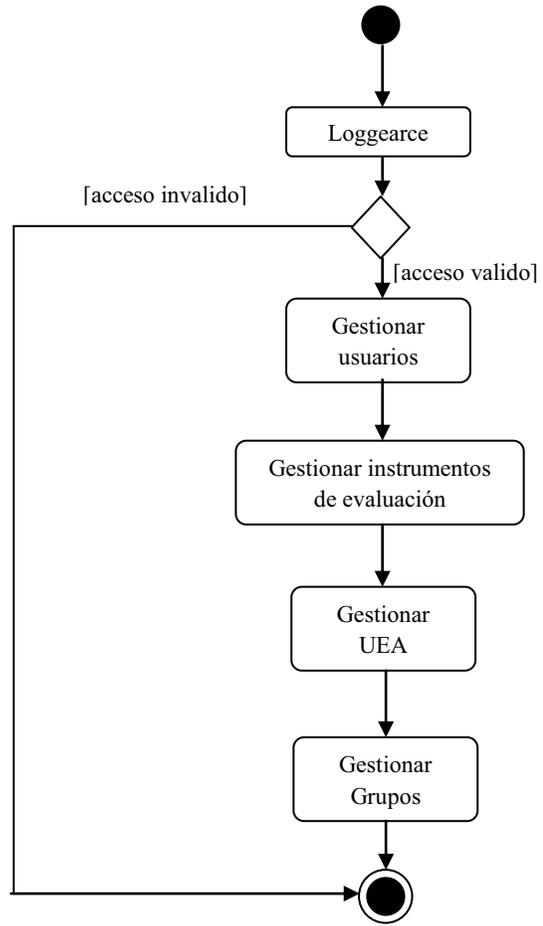
columna	tabla	tipo	not null	default	llave	Descripción
id_uea	uea	int	not null		primary key, foreign key	identificador de uea
clave_uea	uea	varchar(10)	not null			clave para identificar uea
nombre_uea	uea	varchar(50)				descripción de la uea
id_grupo	grupo	int	not null		primary key, foreign key	identificador de grupo
fk_id_uea	grupo	int	not null		foreign key	es llave primaria de la tabla uea
fk_id_usuario	grupo	int	not null		foreign key	es la llave primaria de la tabla usuario
clave_grupo	grupo	varchar(20)				descripción de la clave del grupo
trimestre	grupo	varchar (10)				
anio	grupo	year				indica el año en que se dio de alta el grupo
división	grupo	varchar(5)				indicador de la división a la que pertenece el grupo
id_usuario	usuario	int	not null		primary key, foreign key	indicador de usuario
num_economico	usuario	int	not null			numero económico correspondiente a cada usuario
nombre	usuario	varchar(50)	not null			nombre de pila del usuario
apellido_paterno	usuario	varchar(50)	not null			apellido paterno del usuario
apellido_materno	usuario	varchar(50)				apellido materno del usuario
nombre_id_usuario	usuario	varchar(30)	not null			nombre de usuario identificador del sistema
password_usuario	usuario	varchar(30)	not null			Password del usuario
area	usuario	varchar(10)				área a la que pertenece el usuario
departamento	usuario	varchar(50)				departamento al que pertenece el usuario
correo_electronico	usuario	varchar(50)				Correo electrónico
rol	usuario	varchar(30)	not null			rol que desempeña el usuario en el sistema, con indicador de valor: 1 :administrador 2:profesor 3: jefe de departamento

columna	tabla	tipo	not null	default	llave	Descripción
id_evaluacion	evaluacion	int	not null		primary key, foreign key	identificador de evaluación
fk_id_grupo	evaluacion	int	not null		foreign key	es llave primaria de la tabla grupo
evaluación_fecha	evaluacion	date				indicador de la fecha en que se realizo una evaluación
rendimiento	evaluacion	double		0		indicador del rendimiento obtenido en una evaluación
estado	evaluacion	int				indicador con valor: 1: evaluación no contestada 2: evaluación realizada
id_respuesta	respuesta	int	not null		primary key, foreign key	identificador de respuesta
fk_evaluacion_id	respuesta	int	not null		foreign key	es llave primaria de la tabla evaluación
respuesta	respuesta	varchar(1000)				
id_instrumento_evaluacion	instrumento_evaluacion	int	not null		primary key, foreign key	identificador de instrumento_evaluacion
fk_id_evaluacion	instrumento_evaluacion	int	not null		foreign key	es llave primaria de la tabla evaluacion
tipo_instrumento	instrumento_evaluacion	varchar(100)				
id_pregunta	pregunta	int	not null		primary key, foreign key	identificador de pregunta
fk_id_instrumento_eval	pregunta	int	not null		foreign key	es llave primaria de la tabla instrumento_evaluacion
pregunta_descripcion	pregunta	varchar(500)				
id_opciones_respuesta	opciones_respuesta	int	not null		primary key	identificador de opciones_respuesta
fk_id_pregunta	opciones_respuesta	int	not null		foreign key	es llave primaria de la tabla pregunta
tipo_respuesta	opciones_respuesta	varchar(45)				indicador si la respuesta es : 1: numérica 2: abierta 3: si/no 4: opción múltiple
opción_respuesta	opciones_respuesta	varchar(5)				
descripción_respuesta	opciones_respuesta	varchar(500)				descripción de las respuesta
valor_respuesta	opciones_respuesta	int				peso numérico que se le asigna al resultado de una pregunta

Diagrama de actividades



Administrador



Jefe de Departamento

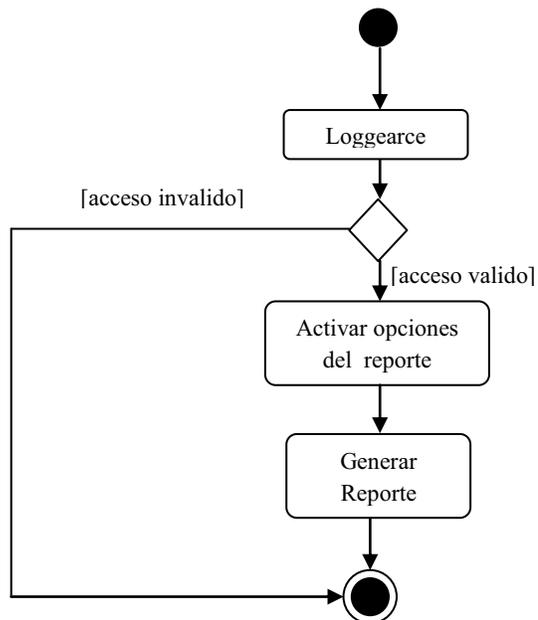
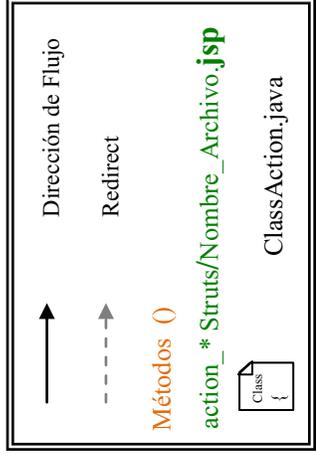
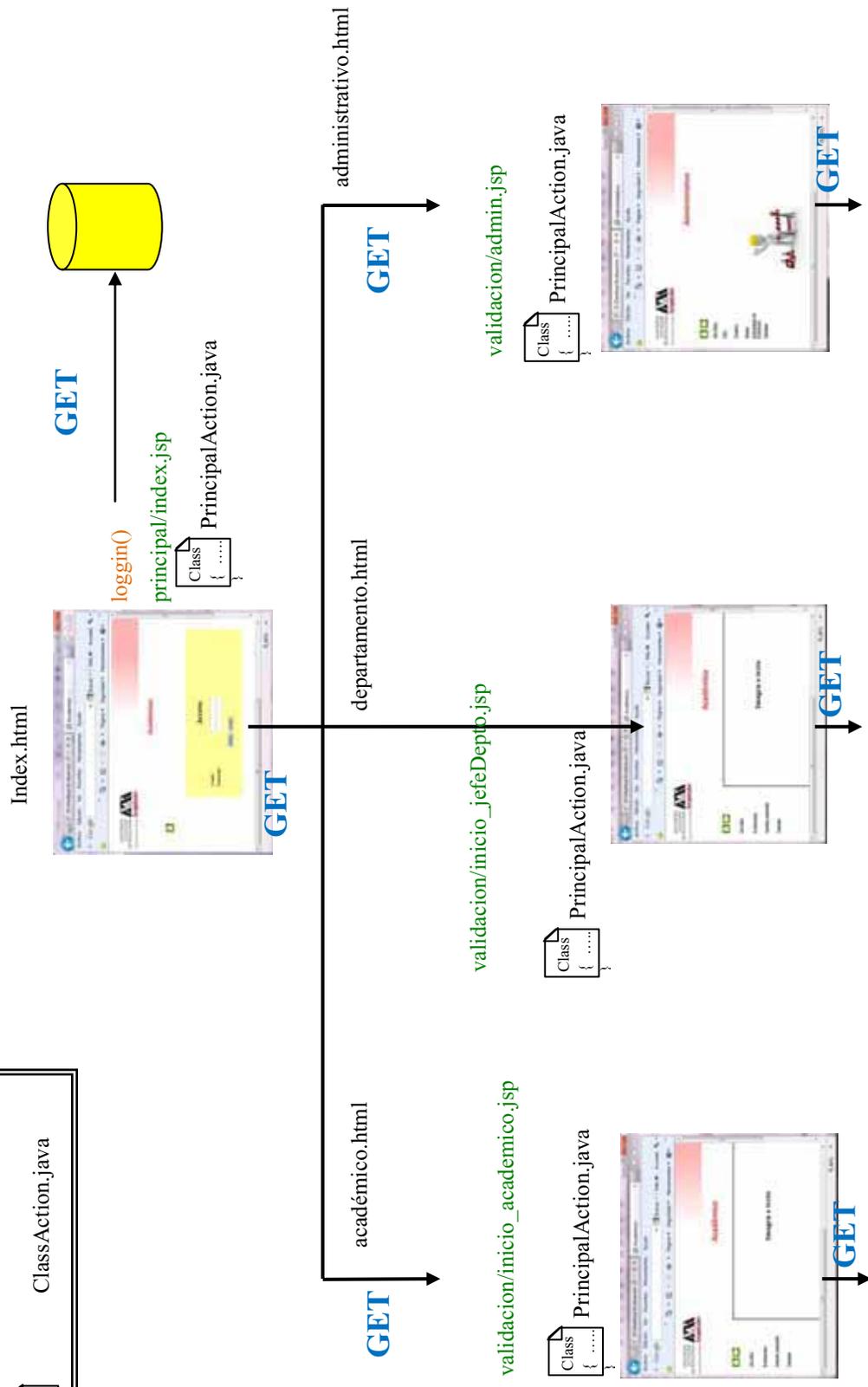


Diagrama de navegación

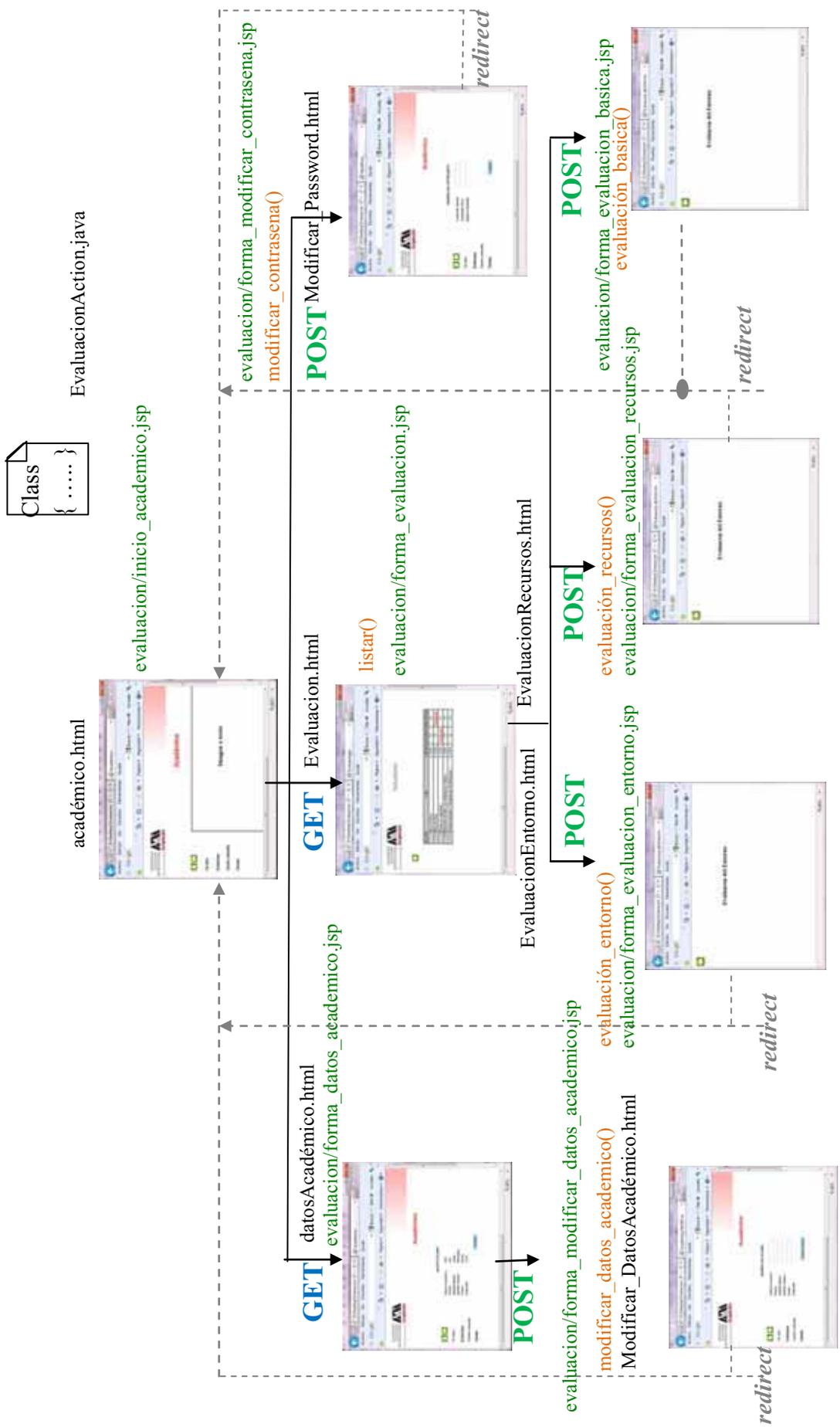


El sistema consta de tres módulos , Administrativo, Académico y Departamento (jefe de departamento)

FLUJO DE PANTALLA



MODULO ACADÉMICO



MODULO DEPARTAMENTO (Jefe de Departamento)

```
Class
{
  ...
}
```

ReportesAction.java

Index.html

validacion/inicio_jefeDepto.jsp
departamento.html



PrincipalAction.java

```
Class
{
  ...
}
```

redirect

redirect

GET

DatosDepto.html



reportes/forma_datos_depto.jsp

Forma_datos_depto()

```
Class
{
  ...
}
```

ReportesAction..java

POST

reportes/forma_modificar_datos_depto.jsp

ModificarDatosDepto.html



Forma_modificar_datos_depto()

```
Class
{
  ...
}
```

ReportesAction..java

redirect

GET

GenerarReporteDepto.html



reportes/forma_datos_reporte.jsp

Forma_reporte()

```
Class
{
  ...
}
```

ReportesAction..java

POST

Reporte.html



ReportesAction..java

reportes/ver_reporte.jsp

Ver_reportes()

```
Class
{
  ...
}
```

ReportesAction..java

redirect

POST



reportes/forma_modificar_contraseña.jsp

Forma_modificar_contraseña()

```
Class
{
  ...
}
```

ReportesAction..java

Código de Ejemplo de cómo está constituido el funcionamiento de la gestión de Usuarios en el sistema

Clase Usuario

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package mx.uam.azc.sistemas.evaluacion.modelo;

import java.io.Serializable;
//import java.sql.Date;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="usuario")

public class Usuario implements Serializable {

    private static final long serialVersionUID = -8767337896773261247L;

    private int id_usuario;
    private int no_economico;
    private String nombre;
    private String ap_paterno;
    private String ap_materno;
    private String nombre_id_usuario;
    private String password;
    private String area;
    private String departamento;
    private String correo_electronico;
    private String rol_usuario;

    @Id
    @GeneratedValue
    @Column(name="id_usuario")
    public int getId_usuario() {
        return id_usuario;
    }

    public void setId_usuario(int id_usuario) {
        this.id_usuario = id_usuario;
    }

    @Column(name="num_economico")
    public int getNo_economico() {
        return no_economico;
    }

    public void setNo_economico(int no_economico) {
```

```

        this.no_economico = no_economico;
    }
    @Column(name="nombre")
    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
    @Column(name="apellido_paterno")
    public String getAp_paterno() {
        return ap_paterno;
    }

    public void setAp_paterno(String ap_paterno) {
        this.ap_paterno = ap_paterno;
    }
    @Column(name="apellido_materno")
    public String getAp_materno() {
        return ap_materno;
    }

    public void setAp_materno(String ap_materno) {
        this.ap_materno = ap_materno;
    }
    @Column(name="nombre_id_usuario")
    public String getNombre_id_usuario() {
        return nombre_id_usuario;
    }

    public void setNombre_id_usuario(String nombre_id_usuario) {
        this.nombre_id_usuario = nombre_id_usuario;
    }
    @Column(name="password_usuario")
    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }
    @Column(name="area")
    public String getArea() {
        return area;
    }

    public void setArea(String area) {
        this.area = area;
    }
    @Column(name="departamento")
    public String getDepartamento() {
        return departamento;
    }

    public void setDepartamento(String departamento) {
        this.departamento = departamento;
    }

```

```
}
@Column(name="correo_electronico")
public String getCorreo_electronico() {
    return correo_electronico;
}

public void setCorreo_electronico(String correo_electronico) {
    this.correo_electronico = correo_electronico;
}
@Column(name="rol")
public String getRol_usuario() {
    return rol_usuario;
}

public void setRol_usuario(String rol_usuario) {
    this.rol_usuario = rol_usuario;
}

}
```

Clase Action Usuario

```
/**
 *
 */
package mx.uam.azc.sistemas.evaluacion.actions;

import com.opensymphony.xwork2.ActionSupport;

import mx.uam.azc.sistemas.evaluacion.controller.AdminUsuario_Impl;
import mx.uam.azc.sistemas.evaluacion.modelo.Usuario;
import java.util.List;

/**
 * @author Miriam
 *
 */

public class UsuariosAction extends ActionSupport
{

    private Usuario usuario;
    private List<Usuario> usuarioList;
    private int id;
    private AdminUsuario_Impl linkController;

    // ***** GET & SET
    private static final long serialVersionUID = 9149826260758390091L;

    public Usuario getUsuario()
    {
        return this.usuario;
    }

    public void setUsuario( Usuario usuario )
    {
        this.usuario = usuario;
    }

    public List<Usuario> getUsuarioList()
    {
        return this.usuarioList;
    }

    public void setUsuarioList( List<Usuario> usuarioList )
    {
        this.usuarioList = usuarioList;
    }

    public int getId()
    {
        return this.id;
    }

    public void setId( int id )
    {

```

```

        this.id = id;
    }

    public AdminUsuario_Impl getLinkController()
    {
        return this.linkController;
    }

    public void setLinkController( AdminUsuario_Impl linkController )
    {
        this.linkController = linkController; }
    //***** CONSTRUCTOR
    public UsuariosAction() {
        linkController =new AdminUsuario_Impl();
    }
    //**** METODOS ****

    public String execute() {
        this.usuarioList = linkController.list();
        return SUCCESS;
    }

    public String inicio_admin()
    {
        return "inicio_admin";
    }
    public String forma_usuarios()
    {
        return "forma_usuarios";
    }
    public String addUsuario()
    {
        System.out.println(getUsuario());
        try {
            linkController.addUsuario(getUsuario());
        } catch (Exception e) {
            e.printStackTrace();
        }

        this.usuarioList = linkController.list();
        return SUCCESS;
        //return "forma_ueas_redirect";
    }
    public String deleteUsuario()
    {
        linkController.deleteUsuario( getId() );
        return SUCCESS;
    }
    public String updateUsuario()
    {
        return "forma_usuarios_redirect";
    }
}

```

Implementacion de Usuarios

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package mx.uam.azc.sistemas.evaluacion.controller;

import java.util.List;

import org.hibernate.classic.Session;

import mx.uam.azc.sistemas.evaluacion.modelo.Usuario;
import net.viralpatel.contact.util.HibernateUtil;
import org.hibernate.HibernateException;

public class AdminUsuario_Impl extends HibernateUtil {

//public class ContactManager extends HibernateUtil {
    public Usuario addUsuario(Usuario usuario) {
        Session session =
HibernateUtil.getSessionFactory().getCurrentSession();
        session.beginTransaction();
        session.save(usuario);
        session.getTransaction().commit();
        return usuario;
    }

    public Usuario deleteUsuario(int id) {
        Session session =
HibernateUtil.getSessionFactory().getCurrentSession();
        session.beginTransaction();
        Usuario usuario = (Usuario) session.load(Usuario.class, id);
        if (null != usuario) {
            session.delete(usuario);
        }
        session.getTransaction().commit();
        return usuario;
    }

    public List<Usuario> list() {

        Session session =
HibernateUtil.getSessionFactory().getCurrentSession();
        session.beginTransaction();
        List<Usuario> usuario = null;
        try {
            usuario = (List<Usuario>) session.createQuery("from
Usuario").list();

```

```

//      usuarios = (List<UsuarioDTO>) session.createQuery("from
c_usuario").list();

    } catch (HibernateException e) {
        e.printStackTrace();
        session.getTransaction().rollback();
    }
    session.getTransaction().commit();
    return usuario;
}

//}
}

```

Forma Usuario (jsp)

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>

<%@ taglib prefix="s" uri="/struts-tags" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Usuarios</title>

<link rel="stylesheet" type="text/css"
href="css/pestanas/ddcolortabs.css" />
<link rel="stylesheet" type="text/css" href="css/style.css" />

<style type="text/css">
<!--
.style1 {color: #EBEFF2}
.style3 {color: #990000}
.style2 {color: #666666}
.style5 {color: #666666; font-weight: bold; }
-->
body {
    margin: 0px;
    padding: 0px;
    height: 0px;
    width: 0px;
    border-top-width: 0px;
    border-top-style: 0;
    top: 0px;
    clip: rect(0px,auto,auto,auto);
    margin-top: 0px;

```

```

}
-->
</style>

</head>
<body>
<table width="1226" height="765" border="0" align="top" cellpadding="0"
cellspacing="0">
  <tr style="height:85">
    <td height="26" colspan="4"></td>
    <td colspan="8" ><h2></h2>      </td>
  </tr>
  <tr style="height:10px">
    <td width="15">&nbsp;</td>
    <td width="164" height="32"><table width="99" height="30" border="0"
cellpadding="0" cellspacing="0">
      <tr>
        <th width="47" height="30" scope="col"><a href="index"></a></th>
        <th width="51" scope="col" height="30"><a
href="validacion_inicio_admin"></a></th>
      </tr>
    </table></td>
    <td width="19">&nbsp;</td>
    <td width="43">&nbsp;</td>
    <td width="218">&nbsp;</td>
    <td colspan="2" align="left"><h4 class="style1">USUARIO</h4></td>
    <td width="85">&nbsp;</td>
    <td width="53">&nbsp;</td>
    <td width="231">&nbsp;</td>
    <td width="37">&nbsp;</td>
    <td width="120">&nbsp;</td>
  </tr>
  <tr>
    <td height="6" colspan="12"><div class="tabcontainer">
      <div class="ddcolortabsline"></div>
    </div>      </td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td align="center"><span class="style3">Administrativo</span></td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td align="left" ><span class="style5">ALTA USUARIO</span></td>
    <td colspan="2">&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>

<s:form method="post" action="addUsuario" theme="simple">

```

```

<tr>
  <td rowspan="2">&nbsp;</td>
  <td rowspan="6" valign="top"><jsp:include
page="../../menu/menuAdmin.jsp"/>&nbsp;</td>
  <td align="right">&nbsp;</td>
  <td colspan="2">Nombre*:
    <s:textfield name="usuario.nombre" />
  </td>
  <td colspan="2">Area:
    <s:textfield name="usuario.area" />
  </td>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
</tr>

<tr>
  <td height="19">&nbsp;</td>
  <td colspan="2">Ap Paterno*:
    <s:textfield name="usuario.ap_paterno" />
  </td>
  <td colspan="2">Departamento:
    <s:textfield name="usuario.departamento" />
  </td>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
</tr>

<tr>
  <td height="19">&nbsp;</td>
  <td>&nbsp;</td>
  <td colspan="2">Ap Materno*:
    <s:textfield name="usuario.ap_materno" />
  </td>
  <td colspan="2">Correo:
    <s:textfield name="usuario.correo_electronico" />
  </td>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
  <td>&nbsp;</td>
</tr>

<tr>
  <td height="19">&nbsp;</td>
  <td>&nbsp;</td>
  <td colspan="2">Usuario*:
    <s:textfield name="usuario.nombre_id_usuario" />
  </td>
  <td colspan="2">Rol*:

```

```

    <% /*
    <s:select list="selecci" name="usuario.rol_usuario" headerKey="-
1">
        <option value="1">Administrador</option>
        <option value="2">Departamento</option>
        <option value="3" selected="selected">Profesor</option>
    </s:select>
    */%>

    <s:select label="Rol"
        name="usuario.rol_usuario"
        headerKey="-1" headerValue="Select Rol"
        list="#{'01':'Administrador',
'02':'Departamento', '03':'Profesor'}"
        value="rol_usuario"
        required="true"
    />

```

```

    </td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
</tr>

<tr>
    <td height="30" >&nbsp;</td>
    <td>&nbsp;</td>
    <td colspan="2">(max 10 digitos)<br />
        Password* :
        <s:textfield name="usuario.password" />
    </td>
    <td colspan="2">N Economico:
    <s:textfield name="usuario.no_economico" />
    </td>
    <td>
    <s:submit value="Insertar" name="insertar" id="insertar" />
    </td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
</tr>

</s:form>

```

```

<tr>
    <td height="24">&nbsp;</td>
    <td>&nbsp;</td>
    <td colspan="7"><table width="878" border="1">

```

```

        <tr>
            <td width="26" bgcolor="#CCCCCC"><div
align="center"><strong>ID</strong></div></td>
            <td width="261" bgcolor="#CCCCCC"><div
align="center"><strong>Nombre</strong></div></td>
            <td width="59" bgcolor="#CCCCCC"><div
align="center"><strong>Rol</strong></div></td>
            <td width="106" bgcolor="#CCCCCC"><div
align="center"><strong>Usuario</strong></div></td>
            <td width="65" bgcolor="#CCCCCC"><div
align="center"><strong>Password</strong></div></td>
            <td width="96" bgcolor="#CCCCCC"><div
align="center"><strong>Departamento</strong></div></td>
            <td width="162" bgcolor="#CCCCCC"><div
align="center"><strong>Correo</strong></div></td>
            <td width="96" bgcolor="#CCCCCC"><div align="center"><strong>No
Economico</strong></div></td>
            <td width="22" bgcolor="#CCCCCC">&nbsp;</td>
            <td width="22" bgcolor="#CCCCCC">&nbsp;</td>
        </tr>

        <s:iterator value="usuarioList" var="usuario">
        <tr>
            <td><s:property value="id_usuario"/></td>
            <td><s:property value="nombre"/>,<s:property
value="ap_paterno"/>,<s:property value="ap_materno"/>
            </td>
            <td><s:property value="rol_usuario"/></td>
            <td><s:property value="nombre_id_usuario"/></td>
            <td><s:property value="password"/></td>
            <td><s:property value="departamento"/></td>
            <td><s:property value="correo_electronico"/></td>
            <td><s:property value="no_economico"/></td>
            <td>
                <a href="deleteUsuario?id=<s:property value="id_usuario"/>">
                
                </a>
            </td>
            <td></td>
        </tr>
        </s:iterator>

    </table></td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
</tr>
<tr>
    <td height="31">&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td valign="top">&nbsp;</td>
    <td width="158">&nbsp;</td>
    <td width="90">&nbsp;</td>
    <td>&nbsp;</td>

```

```

        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td height="32">&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td valign="top">&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td height="36">&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td valign="top">&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td height="309"><pre>&nbsp;</pre></td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td valign="top">&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
</table>
</body>
</html>

```

Manual de Instalación y Configuración

Descargar Java

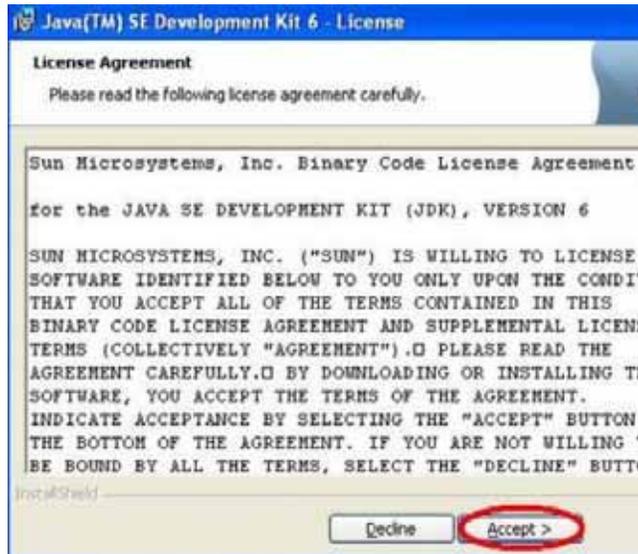
La versión de Java utilizada en el desarrollo del sistema , es la versión jdk_6u1-wundows.exe

1.- Descargar Java del siguiente link

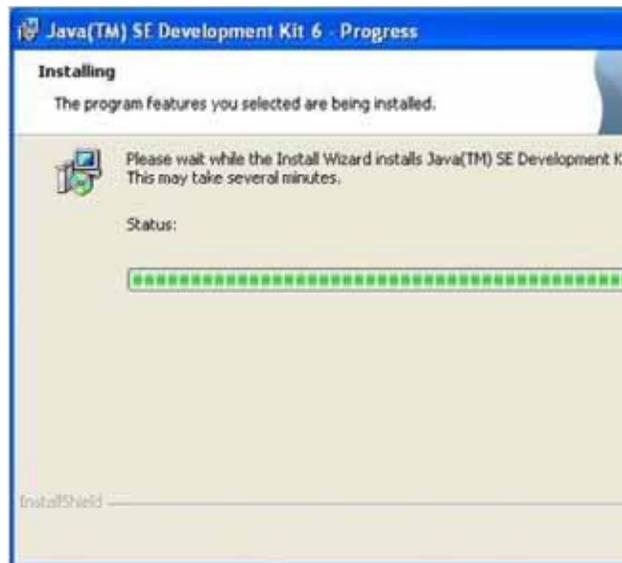
<http://www.oracle.com/technetwork/es/java/javase/downloads/index.html>



2.- Ejecute el archivo descargado y presione “Aceptar” , para ejecutar los términos de la licencia



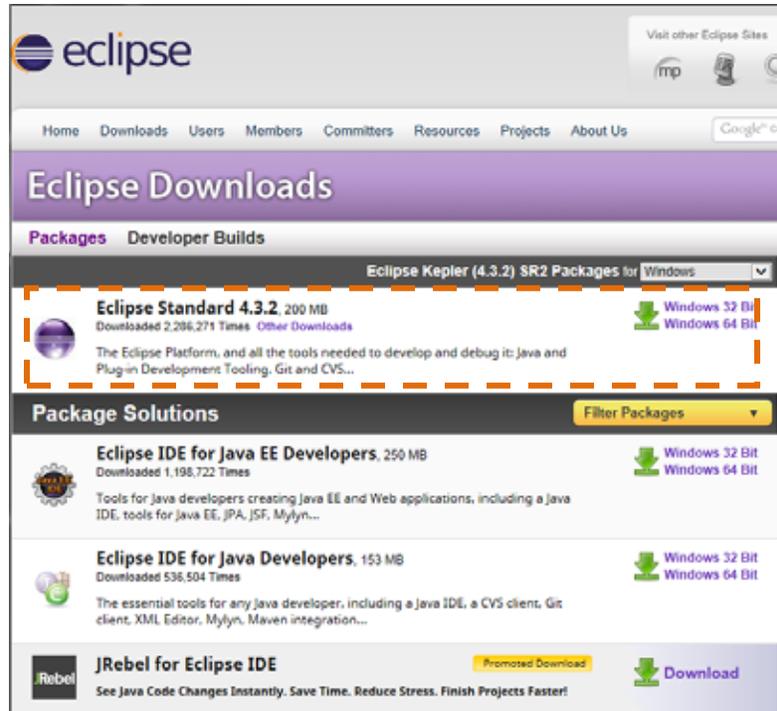
3.-Espere mientras avanza el proceso de instalación



4.- Cuando termine el proceso, aparecerá una ventana emergente donde presionaremos “Finish”

Instalación de Eclipse

1.- De la página oficial de Eclipse descargamos



2.-Se Descomprime la carpeta en C:/

3.- Se ingresa en la carpeta d Eclipse y se ejecuta el archivo eclipse.exe para poder abrir eclipse

Descargar Struts2

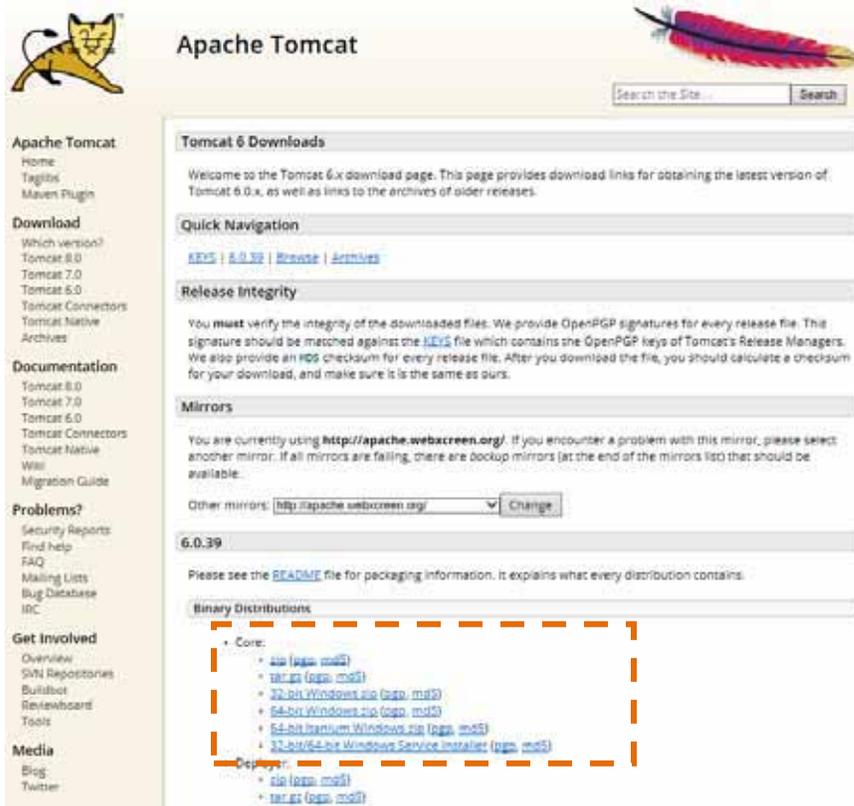
1.- Descarga Struts2 del siguiente link <http://struts.apache.org/development/2.x/>



The screenshot shows the Apache Struts 2 website. At the top left is the Apache Software Foundation logo with the URL <http://www.apache.org/>. Below it, it says "Last Published: 2013-11-05 | Version: 2.3.16-SNAPSHOT". The main header reads "About Apache Struts 2". The central content area features the Struts 2 logo, which includes the "WebWork" logo and the text "Struts 2". To the right of the logo, a text block describes Struts 2 as an elegant, extensible framework for creating enterprise-ready Java web applications. Below this, a paragraph explains that Struts 2 was originally known as WebWork 2 and that the WebWork and Struts communities joined forces to create Struts 2. At the bottom of the main content area, there are three buttons: "Download Now" (with a version indicator "version 2.3.16-SNAPSHOT"), "Get Started" (with a gear icon), and "Plugin Registry" (with a puzzle piece icon). On the left side, there is a navigation menu with categories like "STRUTS 2", "PARENT PROJECT", "DOCUMENTATION", "SUPPORT", "REFERENCE", and "DEVELOPMENT". The "STRUTS 2" category is expanded, showing links for "Welcome", "Releases", "Thanks!", "Sponsorship", "Apache Struts", "Getting Started", "Tutorials", "FAQs", "Guides", "User Mailing List", "Issue Tracker (JIRA)", "Wiki Pages", "XWork API", "Struts Core API", "Plugin APIs", "Tag Reference", "Plugin Registry", and "Struts 2.x Draft Docs".

Descargar Apache Tomcat

1.- Descargar Apache Tomcat del siguiente link , <http://tomcat.apache.org/download-60.cgi>

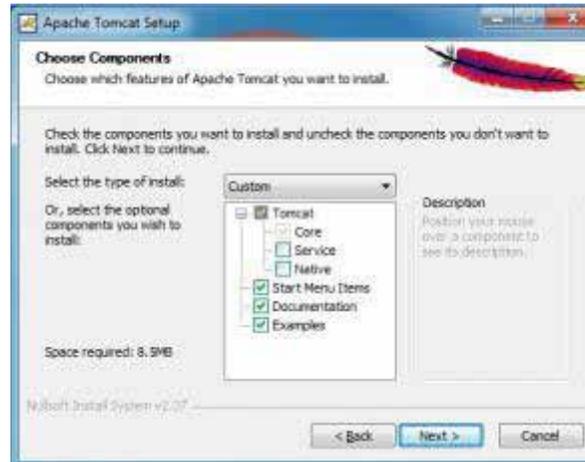


2.- Ejecutamos el archivo descargado , y nos aparece una ventana emergente dándonos la bienvenida , y presionamos el botón “Next”

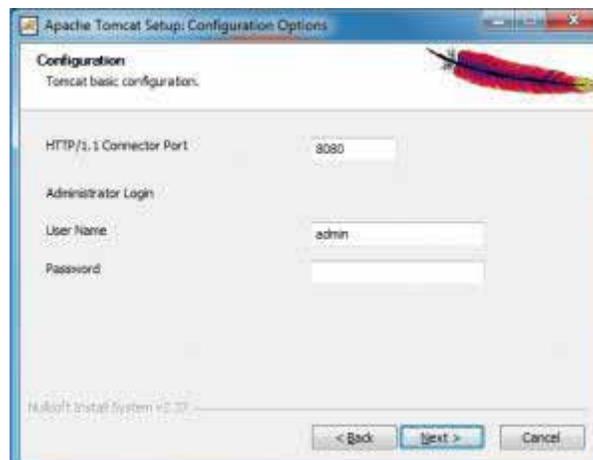
3.- Aparecerá una pantalla de los acuerdos de licencia de uso para Apache Tomcat y seleccionamos el botón de “I Agree”



4.- Seleccionamos los complementos que se desean instalar.



5.- indicamos la configuración del puerto con el valor de 8080 , y le indicamos el nombre de usuario y password, y presionamos next.

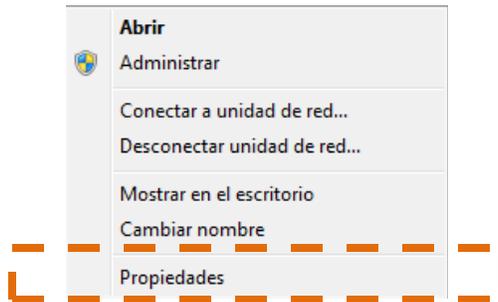


6.- Finalizamos la instalación

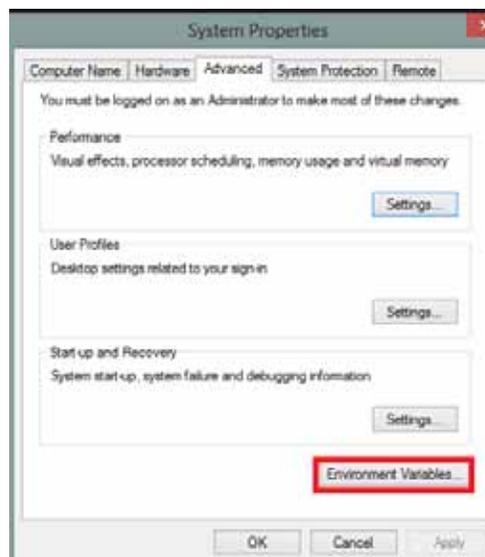


Levantar apache tomcat

1.- Boton inicio -> Equipo y desplegamos un menú contextual , y seleccionamos propiedades



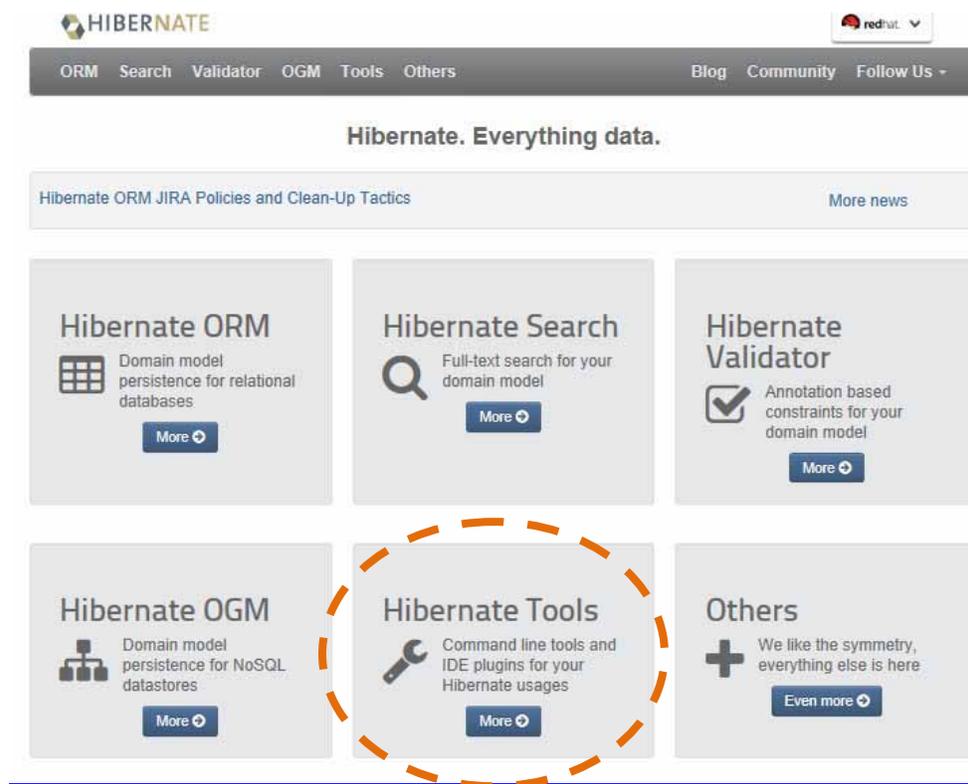
2.- En el menú del lado izquierdo se selecciona “configuración avanzada del sistema”, y se selecciona la opción de “variables de entorno” y en la variable CLASSPATH se le indica la ruta del fichero **servlet-api.jar** que se encuentra en el **tomcat_home\common\lib**



3.- Para verificar que Tomcat funciona bien, una vez levantado, abrir una sesión del navegador y teclear <http://localhost:8080>

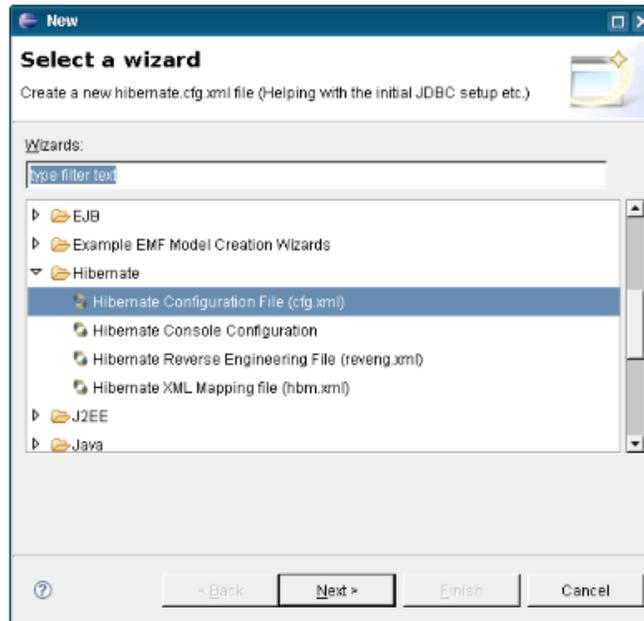
Descargar Hibernate

1.- Descargar Hibernate del siguiente link, <http://hibernate.org/>

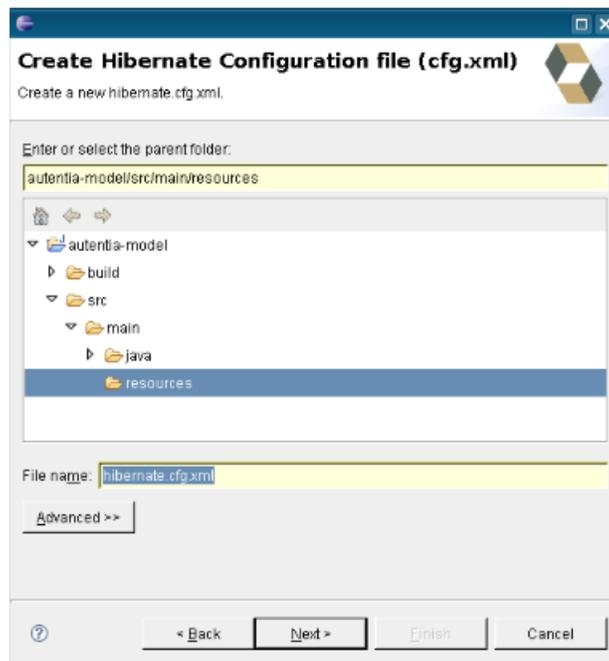


2.- Una vez descargado basta con descomprimirlo en el directorio de Eclipse , que es el mismo directorio donde se encuentra el ejecutable de Eclipse.

3.- Para configurar Hibernate , se selecciona la opción de Configuration File



4.- se le indica donde debe de guardarse el archivo cfg.xml



5.- Ahora se le indica el dialecto que debe de usar Hibernate, Tambien le indicamos la clase del driver de acceso a la base de datos , la URL de conexión , el usuario y el password.

Hibernate Configuration File (cfg.xml)
This wizard creates a new configuration file to use with Hibernate.

Container: /autentia-model/src/main/resources
File name: hibernate.cfg.xml
Session factory name:
Database dialect: MySQLInnoDB
Driver class: com.mysql.jdbc.Driver
Connection URL: jdbc:mysql://localhost:3306/
Default Schema:
Default Catalog:
Username:
Password:
 Create a console configuration

< Back Next > Finish Cancel

Al presionar Finalizar, se genera un fichero con la siguiente información de nuestra configuración.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
"-//Hibernate/Hibernate Configuration DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
  <session-factory>
    <property
name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
    <property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/Evaluaciones<
/property>
    <property
name="hibernate.connection.username">root</property>
```

```
<property name="hibernate.connection.password">root</property>
<property
name="hibernate.dialect">org.hibernate.dialect.MySQLInnoDBDialect</proper
ty>
</session-factory>
</hibernate-configuration>
```

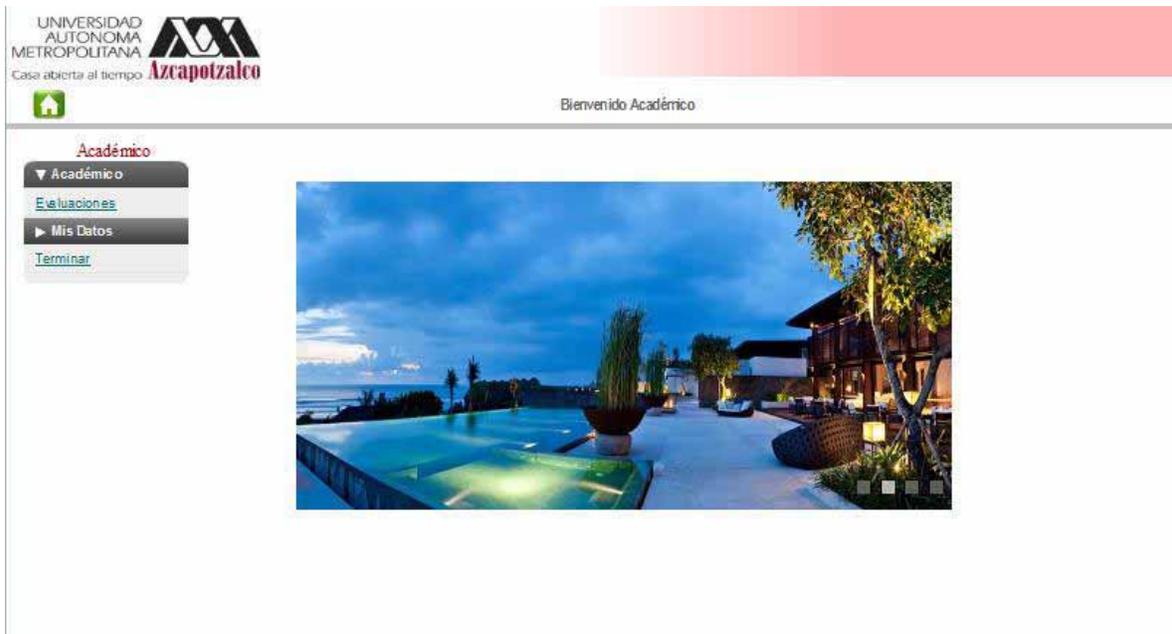
Manual de Usuario

La pagina principal que se despliega, el usuario debe realizar el Loggin con un usuario y password previamente solicitados al Administrador del sistema.



Académico

El modulo de Académico muestra una ventana de Bienvenida



En el menú se encuentra la opción de Evaluaciones y al presionar el botón de Evaluación , nos despliega las evaluaciones que debemos realizar



ID	Grupo	UEA	Trimestre	Básica	Instrumentos	
1	1	BASE DE DATOS	O	evaluacion		
2	2	DISEÑO DE ALGORITMOS	O	evaluacion		
3	3	SOFTWARE DE BASE	P	evaluacion		
4	4	COMPILADORES	O	evaluacion		

Se genera La evaluación y al finalizar se presiona el botón enviar Evaluación



Academico

INSTRUMENTOS DE EVALUACION

TIPO 1

RESPECTO AL CURSO

1.- Porcentaje cubierto del programa:

2.- Justificación

3.- Porcentaje de puntualidad del profesor %:

4.- Porcentaje de asistencia del profesor %:

5.- Presento puntualmente a los alumnos el programa de UEA:

6.- Presento criterios de evaluación de la UEA:

7.- Horas a la semana asignada por usted a asesorías:

8.- Bibliografía Utilizada

9.- Numero de alumnos en la 1a lista:

10.- Numero de alumnos en la 2a lista:

11.- Numero de NA:

12.- Numero de S:

13.- Numero de B:

14.- Numero de MB:

15.- Numero de alumnos que aparecen en el acta con NA y no presentaron examen global (desertores) :

EVALUE EL PROCESO DE ENSEÑANZA APRENDIZAJE EN CUANTO

Administrador

Se despliega una pantalla de Bienvenida Administrador



En el menú Izquierdo se pueden realizar Gestiones de UEA, Grupo, Usuarios.

ID	Nombre	Rol	Usuario	Password	Departamento	Correo	No Económico	
1	miri,luna ,luna	administrador	lo do	123	CEI	miri.l1@123.com	300	✗
2	lo la,lima,lima	profesor	mie l	456	CEI	lola.l1@123.com	305	✗
3	luis,lara ,mendez	administrador	luna	789	CEI	luis.lm@123.com	310	✗
4	klaus,phoma ,lila	profesor	lima	120	CEI	klaus.p.t@123.com	315	✗

Resultados

El resultado final, es un sistema que realiza, Altas, Bajas y Cambios.

UNIVERSIDAD AUTÓNOMA METROPOLITANA Azcapotzalco

INSTRUMENTOS DE EVALUACION

Administrativo

ALTA USUARIO

Nombre*: Área:

Ap Paterno*: Departamento:

Ap Materno*: Correo:

Usuario*: Rol*:

(max 10 dígitos)

Password*: N Económico:

ID	Nombre	Rol	Usuario	Password	Departamento	Correo	No Económico		
1	miri.lira_lira	administrador	lodo	123	CEI	miri.l1@123.com	300	X	!
2	lola.lira_lira	profesor	miel	456	CEI	lola.l1@123.com	305	X	!
3	luis.lira_gonzalez	administrador	lura	789	CEI	luis.lm@123.com	310	X	!
4	klaus.phara_lila	profesor	lira	120	CEI	klaus.p.l@123.com	315	X	!

El sistema permite realizar evaluaciones a las unidades de enseñanza aprendizaje UEA y por consiguiente la generación de reportes que permitirán realizar un análisis de la información para la detección de problemas y mejorar la calidad de la enseñanza.

UNIVERSIDAD AUTÓNOMA METROPOLITANA Azcapotzalco

INSTRUMENTOS DE EVALUACION

Academico

TIPO 1

RESPECTO AL CURSO

1.- Porcentaje cubierto del programa:

2.- Justificación

3.- Porcentaje de puntualidad del profesor %:

4.- Porcentaje de asistencia del profesor %:

5.- Presento puntualmente a los alumnos el programa de UEA:

6.- Presento criterios de evaluación de la UEA:

7.- Horas a la semana asignada por usted a asesorías:

8.- Bibliografía Utilizada

9.- Numero de alumnos en la 1a lista:

10.- Numero de alumnos en la 2a lista:

11.- Numero de NA:

12.- Numero de S:

13.- Numero de B:

14.- Numero de MB:

15.- Numero de alumnos que aparecen en el acta con NA y no presentaron examen global (desertores):

EVALUE EL PROCESO DE ENSEÑANZA APRENDIZAJE EN CUANTO

Conclusión

Concluyo que con la implementación del sistema, la realización de la evaluación a la unidad de enseñanza aprendizaje (UEA) se realizaría de manera fácil y permitiría agilizar el análisis, la detección de problemas y generar reportes , y estos permitan la toma de decisiones para la mejora del entorno y los recursos.

Bibliografía

- [1] R. A. Castro, “Evaluador de programas fuente en lenguaje C con entradas y salidas descritas mediante una cadena de formato”, Proyecto Terminal, División de CBI, Universidad Autónoma Metropolitana Azcapotzalco, México, 2007.
- [2] V. V. Arévalo, “Análisis estadístico de evaluaciones docentes utilizando modelos de ecuaciones estructurales”, Proyecto Terminal, División de CBI, Universidad Autónoma Metropolitana Azcapotzalco, México, 2007.
- [3] J. D. Mendoza y A. A. Santos, “Sistema de evaluaciones dinámicas del proceso de enseñanza aprendizaje basado en la opinión de los alumnos con doble interfaz: desktop web y mobile web”, Proyecto Terminal, División de CBI, Universidad Autónoma Metropolitana Azcapotzalco, México, 2010.
- [4] I. Y. Olmos, “Sistema de gestión de calificaciones para los cursos impartidos por un profesor”, Proyecto Terminal, División de CBI, Universidad Autónoma Metropolitana Azcapotzalco, México, 2011.
- [5] Moodle (5 de marzo de 2012). *Calificaciones MoodleDocs* [En línea]. Disponible: <http://docs.moodle.org/es/Calificaciones>
- [6] Mentor (5 de marzo de 2012). *Aula Mentor* [En línea]. Disponible: <http://www.mentor.educacion.es/>
- [7] Qtraining (5 de marzo de 2012). *Desarrollo del personal* [En línea]. Disponible: <http://qtraining.org/>