

Universidad Autónoma Metropolitana Unidad Azcapotzalco
División de Ciencias Básicas e Ingeniería
Licenciatura en Ingeniería en Computación

Proyecto Tecnológico
Sistema de información web de apoyo para exámenes de recuperación.

Presentado por:
Patricia Sevilla Chalchi
208200478

Trimestre 2014 Primavera
Agosto de 2014

Rafaela Blanca Silva López
Profesor Titular
Departamento de Sistemas

Yo, 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.



Asesora.

Yo, Mónica Irene 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.



Coasesora.

Yo, Patricia Sevilla Chalchi, 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.



Alumna.

Agradecimientos:

A mi familia:

Con todo mi cariño y mi amor para las personas que hicieron todo en la vida para que yo pudiera lograr mis sueños, por motivarme y darme la mano cuando sentía que el camino se terminaba, a ustedes por siempre mi corazón y mi agradecimiento.

A mi asesora:

Por aceptarme para realizar este proyecto bajo su dirección, su confianza en mi trabajo y su capacidad para guiar mis ideas, por su paciencia y por impulsar el desarrollo de este trabajo.

A mis amigos:

Sin excluir a ninguno pero en especial a Tere por su asesoría y paciencia durante el desarrollo de este proyecto y a Dany por su apoyo incondicional.

Gracias a esas personas importantes en mi vida, que siempre estuvieron listas para brindarme toda su ayuda, ahora me toca regresar un poquito de todo lo inmenso que me han otorgado. Con todo mi cariño este proyecto se las dedico a ustedes.

Resumen

En este proyecto terminal, los alumnos de la UAM-A podrán realizar las siguientes consultas de información para la preparación de exámenes de recuperación correspondientes al trimestre en curso:

- UEA: consulta mapa curricular de Ingeniería en computación.
- Información de exámenes de recuperación programados: clave de UEA, fecha, horario, grupo, salón, observaciones (bibliografías, notas, etc.)
- Información de profesor sinodal: cubículo, horario.
- Pagina de archivos, en ella el alumno podrá consultar archivos PDF proporcionados por los profesores sinodales.

El desarrollo se realiza aplicando los beneficios proporcionados por Struts2.

Strus2 es un framework de desarrollo web, siendo su principal ventaja que la implementación es más sencilla, más rápido, y con menos complicaciones, basta con implementar la lógica propia de la aplicación y configurar el framework mediante algún mecanismo como anotaciones o archivos XML que finalmente estos se encargan de hacer la mayor parte del trabajo tedioso o repetitivo.

Índice

Agradecimientos:	iii
Resumen	iv
1. Introducción	vii
2. Antecedentes	vii
3. Justificación	viii
4. Objetivos:	ix
4.1 Objetivo General	ix
4.2 Objetivos específicos	ix
5. Marco Teórico	x
5.3 Frameworks	xiii
6. Desarrollo del Proyecto	xvii
6.1 Especificación Técnica	xxiv
7. Resultados	xxvi
7.1 Interfaz de Usuario Final	xxvi
7.1.2. Consulta UEA	xxvii
7.1.3 Consulta Examen	xxvii
7.1.4 Consulta Profesor	xxviii
8. Conclusiones	xxviii
9. Bibliografía	xxix
MANUAL DE USUARIO	xxx
10. Login Administrador	xxx
10.1 Página principal Administrador	xxx
11. Gestión Examen	xxxii
11.1 Alta Examen	xxxii
11.2 Elimina Examen	xxxiii
11.2.1 Eliminar Información	xxxiv
11.3 Actualizar Examen	xxxv
Iconos:	xxxvi

1. Introducción

Struts 2 es la nueva versión del popular framework de desarrollo web en Java Apache Struts. Es desarrollo de este proyecto está basado en el patrón MVC (Modelo-Vista-Controlador), una arquitectura que busca reducir el acoplamiento dividiendo las responsabilidades en 3 capas diferenciadas:

- El modelo, que hace referencia a los datos que maneja la aplicación y las reglas de negocio que operan sobre ellos y que se traducen en Struts 2 en las acciones.
- La vista, encargada de generar la interfaz con la que la aplicación interacciona con el usuario. En Struts2 equivale a los resultados.
- El controlador, que comunica la vista y el modelo respondiendo a eventos generados por el usuario en la vista, invocando cambios en el modelo, y devolviendo a la vista la información del modelo necesaria para que pueda generar la respuesta adecuada para el usuario. El controlador se implementa en Struts 2 mediante el filtro FilterDispatcher.[1]

Gracias a este patrón es posible tener un orden en la programación del proyecto.

Se desarrollaron dos vistas de interfaz de usuario, la razón es que cada uno tiene diferentes tareas a en el sistema, es decir, el usuario administrador tiene una interfaz adicional de gestión de información (altas, bajas, actualizaciones y finalmente consultas), podrá ingresar a la página con id de usuario y contraseña, a diferencia del usuario alumno, que únicamente realizara consultas.

2. Antecedentes

Cuando Internet surge las páginas web eran estáticas y podían ser únicamente consultadas y descargadas. Java es un lenguaje de programación orientado a objetos que ha tenido éxito en el desarrollo de las aplicaciones web, cuenta con dos componentes clave: la página JSP que combina etiquetas que manejan el contenido estático y dinámico y los Servlet. Con el paso del tiempo y el uso extendido de Java, surgen los frameworks que son bibliotecas que permiten la reutilización del código, que hacen uso de los patrones de diseño, asegurando con ello la escalabilidad del producto. Un framework puede centrarse en alguna capa del MVC (Modelo

Vista Controlador), en aspectos como definir la estructura de la aplicación, en la seguridad, entre otros. El número existente de éstas tecnologías va en aumento, pero no se dispone del análisis documentado con pruebas, que indiquen cual es el framework más adecuado de utilizar, que implique hacer pocas o nulas modificaciones al código final.

Lo que comenzó como algo bueno para la comunidad de desarrolladores, se ha vuelto un problema, debido a que los interesados en hacer uso de algún framework, pero que desconocen de estas herramientas, se le dificulta la elección de uno de ellos, corriendo el riesgo de seleccionar alguna tecnología que no cumpla con sus necesidades.

Los siguientes proyectos hacen uso de aplicaciones web, pero en ningún caso se aplican el uso de Struts2.

- Sistemas de gestión de manuales del Centro de Formación de ingeniería Industrial. [1]
- Sistema Web de seriación e información para alumnos de ingeniería [2]
- Sistemas de evaluaciones dinámicas del proceso de enseñanza aprendizaje basado en la opinión de los alumnos con doble interfaz: desktop web y mobile web. [3]
- Sistema de Evaluación Interactiva para Terminales Móviles. [4]

3. Justificación

Al realizar un análisis en las listas de aprovechamiento de exámenes de recuperación, se observa que el porcentaje de alumnos no aprobados en relación con la población de alumnos aprobados es aproximadamente del 50%.

En la tabla 1 se muestran los porcentajes de aprobación estos datos fueron tomados de una análisis de los trimestres 12-P, 12-O, 12-I.

TRIMESTRE	ALUMNOS INSCRITOS	% ALUMNOS APROBADOS	%ALUMNOS NO APROBADOS
12-P	3016	51.12%	48.88%
12-O	1430	52.90%	47.10%
12-I	1270	56.16%	43.84%

Tabla 1. Análisis de aprovechamiento en exámenes de recuperación.

En la mayoría de los casos, los alumnos desconocen los detalles que abordará el profesor sinodal en el examen de recuperación. En este proyecto terminal que el sistema implementado proporcione a los alumnos información asociada a guías, temarios, material de autoestudio, bibliografías, etc., para facilitar la preparación del examen.

El sistema de información permitirá realizar al alumno la búsqueda específica de la UEA de su interés, consultando información de material de apoyo, así como también información básica de los profesores sinodales en caso de que el alumno requiera de asesoría en forma personal.

4. Objetivos:

4.1 Objetivo General

Diseñar e implementar un sistema web que permita a los alumnos de la DCBI de la Universidad Autónoma Metropolitana obtener información de temarios, guías y recursos de autoestudio de las UEA programadas para exámenes de recuperación que atienden los profesores sinodales del departamento de sistemas.

4.2 Objetivos específicos.

Diseñar e implementar una base de datos que almacene la información necesaria para el sistema de recomendaciones de autoestudio.

- Examen
- UEA (plan de estudios).
- Profesor
- Usuario (Administrador)
- Diseñar un sistema web integrado por los siguientes módulos
 - Módulo de consulta de Examen.
 - Módulo de consulta de UEA.
 - Módulo de consulta de Profesores.
 - Módulo de gestión de administrador (altas, bajas, actualización y consulta de examen, UEA, profesor, usuario, material de apoyo).

5. Marco Teórico

Las aplicaciones Web son diferentes a los sitios Web convencionales porque éstas son capaces de crear una respuesta dinámica.

Muchos sitios Web ofrecen sólo páginas estáticas, en cambio una aplicación Web es capaz de interactuar con bases de datos y lógica de negocios con la finalidad de personalizar la respuesta a enviar.

JSP permite escritura de código Java dentro de ellos consiguiendo todas las ventajas tanto de los servlets como del código HTML. Las aplicaciones Web en general tienen tres aspectos a considerar en su desarrollo:

- El código de acceso, inserción, consulta, actualización y/o eliminación de los datos.
- El código del diseño de las páginas a mostrar.
- El código que controla el flujo de las páginas de la aplicación.

5.1 Java EE

Java EE es una plataforma creada por Sun y cuyo objetivo es el desarrollo de aplicaciones distribuidas dirigidas principalmente a la empresa. Entre sus requisitos están la fiabilidad, la facilidad de mantenimiento, la escalabilidad, etc.

Esta plataforma está basado en Java que es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystem a principios de los años 90 y cuyos principales objetivos eran:

- Uso de la metodología orientada a objetos.
- Portabilidad del mismo programa en múltiples sistemas operativos.
- Incluir por defecto soporte para trabajo en red.
- Permitir la ejecución de código en sistemas remotos de forma segura.
- Facilidad de uso tomando elementos de otros lenguajes orientados a objetos.

La portabilidad entre plataformas que proporciona este lenguaje viene dada por el hecho de que al compilar el código fuente escrito en Java se genera un código conocido como "bytecode" que es ejecutado por lo que se conoce como la máquina virtual de Java. Esta máquina virtual es un programa escrito en código nativo de la plataforma y es el encargado de interpretar y ejecutar el código, es ella la que conoce el hardware sobre el que se ejecuta la aplicación y la que actúa como intermediaria entre ambas.

Esta portabilidad de Java hace que una aplicación Java EE también se pueda instalar fácilmente en cualquier entorno que soporte dicho lenguaje.

La especificación Java EE define un modelo de capas mediante el cual la lógica de la aplicación se divide en componentes de acuerdo con su función. Cada uno de estos componentes se puede acabar instalando en una máquina diferente (aunque no necesariamente) dependiendo de la capa a la que pertenezca. Las capas en las que se divide el sistema son las siguientes: Las capas en las que se divide el sistema son las siguientes:

- Capa cliente: Se ejecuta en la máquina cliente.
- Capa web: Se ejecuta en un servidor Java EE. Es la encargada de obtener datos del cliente y de solicitar a la capa de negocio las operaciones necesarias.
- Capa de negocio: Se ejecuta en un servidor Java EE y forma el núcleo de la aplicación.
- Capa EIS (Enterprise Information System).

5.2 Patrón arquitectónico: MVC

Los patrones de diseño (patrones de diseño) proporcionan soluciones ya probadas a problemas con características similares. Proporcionan catálogos de elementos reusables en el diseño de sistemas y estandarizan la forma en que se realiza el diseño, lo que facilita el aprendizaje.

La utilización de patrones de diseño no es una práctica obligatoria en el diseño de software, pero si que es altamente recomendable ya que permiten el reducir el tiempo de desarrollo de una aplicación al evitar el tener que plantear una solución desde cero.

El patrón MVC (Modelo-Vista-Controlador) es el más ampliamente establecido desde el punto de vista arquitectónico y se caracteriza porque divide el sistema en tres partes, de forma que separa los datos de la aplicación de la interfaz de usuario y de la lógica de control.

Con esta división conseguimos, por ejemplo, que la forma de presentar unos datos sea completamente independiente de los datos en si, con lo que es sencillo el presentar los mismos datos de formas diferentes.

Las responsabilidades de cada una de las partes de este patrón son las siguientes:

Modelo: Contiene los datos con los que opera el sistema. La lógica de datos permite asegurar su integridad y facilita el derivar nuevos datos.

Vista: Representa el modelo de datos y las operaciones realizadas en la capa de negocio en un formato adecuado para que el usuario pueda interactuar (mediante la interfaz de usuario).

Vista: Representa el modelo de datos y las operaciones realizadas en la capa de negocio en un formato adecuado para que el usuario pueda interactuar (mediante la interfaz de usuario).

Controlador: Responde a los eventos, normalmente acciones del usuario, modificando el modelo y generalmente también en la vista, por lo que es encargado de la interacción entre los datos y la vista.

El flujo seguido por este patrón entre los diferentes componentes es el siguiente:

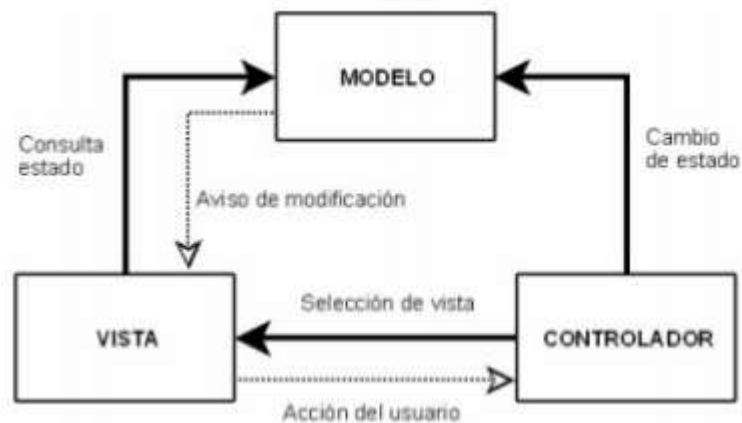


Fig.1. Patrón MVC.

- El usuario interactúa con la vista (interfaz de usuario) realizando algún tipo de acción.
- El controlador recibe desde la vista la notificación de la acción solicitada.
- El controlador accede al modelo actualizándolo conforme a la acción solicitada por el usuario.
- El controlador delega en la vista la tarea de desplegar la interfaz de usuario con la respuesta.
- La vista recupera los datos del modelo para poder completar la interfaz con las modificaciones realizadas sobre el modelo.
- El modelo no debe tener conocimiento directo de la vista, sin embargo en algunos casos puede notificar a la vista que se han producido cambios.

5.3 Frameworks

Un framework es una estructura definida a partir de la cual podemos desarrollar un proyecto de software y su intención es establecer una infraestructura que se encargue de realizar las tareas de más bajo nivel necesarias en cualquier proyecto y permitir así a los desarrolladores el poder dedicar más esfuerzo a las tareas de más alto nivel propias del negocio.

5.3.1 Apache Struts Framework.

Originalmente fue creado por Craig R. McClanahan y donado al proyecto Jakarta de Apache Software Foundation en el año 2000.

El framework provee tres componentes:

- Un manejador de petición proporcionado por el desarrollador de la aplicación, que se usa para relacionarlo a una determinada URI.
- Un manejador de respuesta que se utiliza para transferir el control a otro recurso que se encargará de completar la respuesta.
- Una biblioteca de tags que ayuda a los desarrolladores a crear aplicaciones interactivas basadas en formularios.

5.3.2 Struts2

Struts2 es un framework de aplicación web open source desarrollado por Apache y basado en el patrón MVC (Modelo-Vista-Controlador) que es utilizado ampliamente y considerado de gran solidez.

Se utiliza para construir aplicaciones web basadas en servlets JSP, pudiendo ejecutarse en cualquier contenedor de servlets, incluyendo los servidores de aplicaciones Java EE.

Utiliza internamente una serie de patrones ya definidos (Singleton, Delegate,...) y además proporciona un conjunto de etiquetas JSP personalizadas que facilitan la integración del framework con las páginas JSP. Algunas de estas etiquetas han sido utilizadas en este TFC.

El funcionamiento de Struts2 es el siguiente, cuando el usuario hace una solicitud al servidor el FilterDispatcher la captura y determina el Acción que la debe tratar.

Antes de ejecutar el Acción se le aplican a la solicitud los Interceptores que hayan sido configurados y que permiten realizar como pre-proceso una serie de tareas más o menos comunes, como pueden ser validaciones de datos.

A continuación se ejecuta la acción que realiza los accesos para recuperar o almacenar información en la base de datos y se genera la salida mediante un Resultado.

Esta salida vuelve a pasar a través de los Interceptores (en orden inverso al inicial) para realizar posibles operaciones de post-proceso y finalmente se devuelve la respuesta al usuario.

Desde el punto de vista del patrón MVC los componentes de Struts2 se distribuirían de la siguiente manera de cara a conseguir la separación de los diferentes componentes de la aplicación:

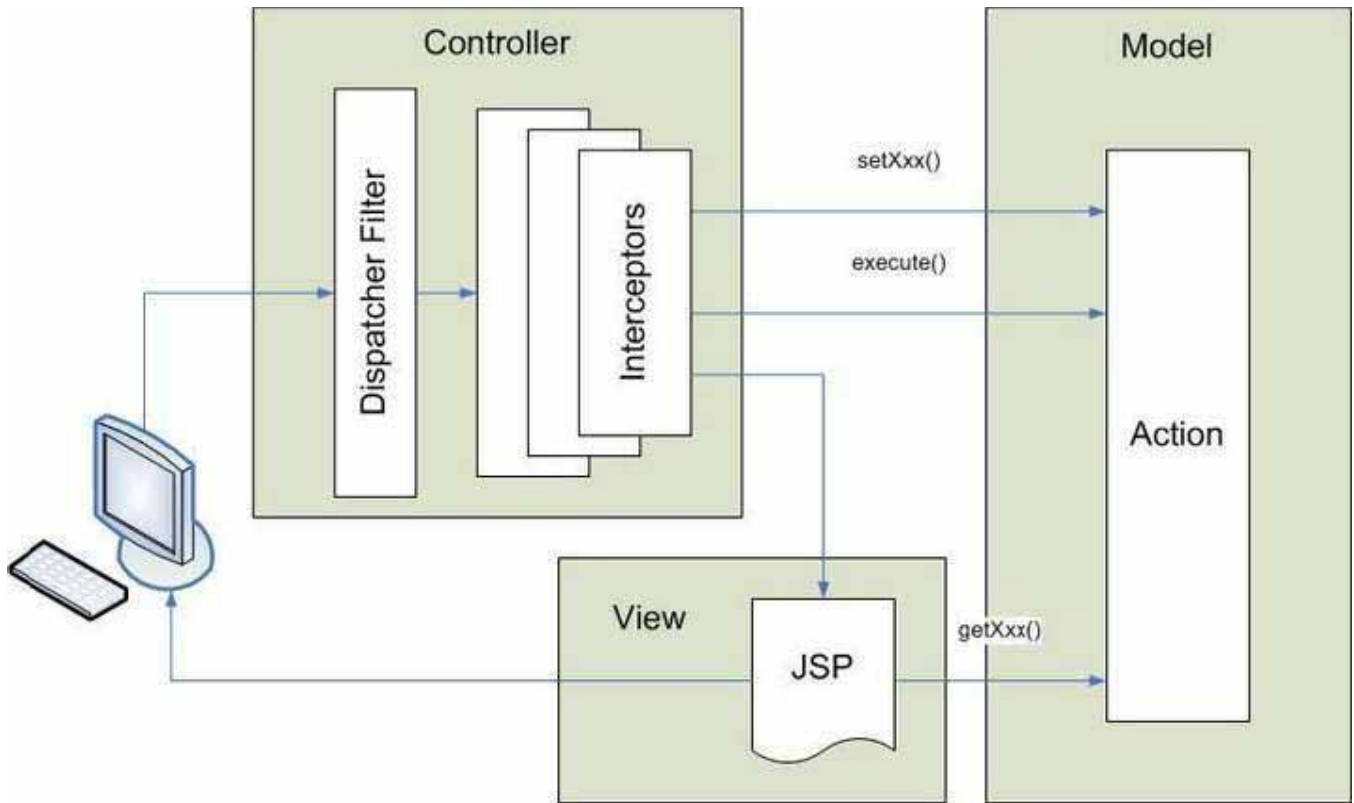


Fig.2. Struts2-MVC

5.3.3 Hibernate

Hibernate es una tecnología que simplifica el acceso a base de datos que se distribuye como una herramienta de software libre distribuida bajo los términos de la licencia GNU LGPL. Permite establecer una correspondencia entre el modelo de la Base de Datos Relacional y una serie de clases que modelan los objetos de la aplicación (mapeo objeto-relacional), es decir, relaciona los dos modelos de datos que conviven en una aplicación, el usado en la memoria del ordenador (orientación a objetos) y el usado en las bases de datos (modelo relacional).

Este mapeo hace que en la práctica se cree una base de datos orientada a objetos virtual sobre la base de datos relacional, en la que Hibernate actúa de intermediario con la aplicación que la utiliza.

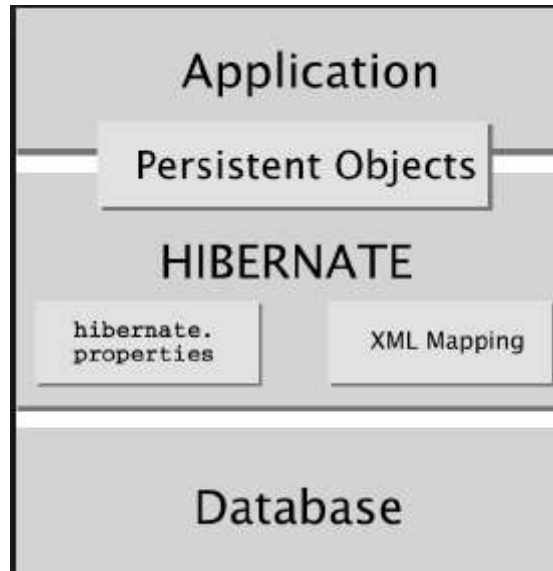


Fig.3. Esquema Hibernate.

Con esta tecnología disponemos de un sistema de acceso a bases de datos relacionales de manera transparente, únicamente se deben escribir sentencias java y de manera independiente, el código escrito con Hibernate funcionará en cualquier motor de datos al que se dé soporte, ya que Hibernate permite a la aplicación el manipular datos de la base de datos operando sobre objetos.

5.4 Base de Datos.

La última serie de producción de PostgreSQL es la 9.1. Sus características técnicas la hacen una de las bases de datos más potentes y robustos del mercado. Su desarrollo comenzó hace más de 16 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema.

- Ampliamente popular - Ideal para tecnologías Web.
- Fácil de Administrar.
- Su sintaxis SQL es estándar y fácil de aprender.

- Footprint bajo de memoria, bastante poderoso con una configuración adecuada.
- Multiplataforma.
- Capacidades de replicación de datos.

Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.

6. Desarrollo del Proyecto

Se crean los siguientes paquetes en nuestro **web-application**:

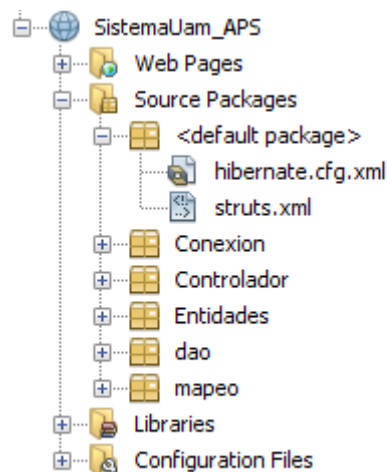


Fig.4. Paquetes

Conexión: se encuentra el código que permite el acceso a la DB cada que sea solicitado.

```

L  */
  package Conexion;

  import java.sql.Connection;
  import java.sql.DriverManager;

  public class ConexionPostgreSQL {
      public String db = "sistemaUamA";
      public String url = "jdbc:postgresql://localhost:5433/"+db;
      public String user = "postgres";
      public String pass = "0206";

      Connection link = null;

      public Connection Conectar() throws Exception
      {
          try
          {
              Class.forName("org.postgresql.Driver");
              //Creamos un enlace hacia la base de datos
              link = DriverManager.getConnection(this.url, this.user, this.pass);
          }
          catch (Exception e)
          {
              //JOptionPane.showMessageDialog(null, e);
              throw new Exception("Error en la conexión a la Base de Datos.");
          }
          return link;
      }
      public void cierraConexion() throws Exception {
          try{
              this.link.close();
          }catch(Exception e){

```

Fig. 5. Conexión.

Controlador: códigos de las clases que responde a los eventos solicitados de struts.xml.

```

68
69 public String agrego (){
70     if(examendao.agrego(examen)){
71         msg="*Se agrego un nuevo registro de Examen";
72     }
73     else{    msg="¡Ya existe!";
74     }
75     return "exito";
76 }
77 public String actualizo (){
78
79     if(examendao.actualizo(examen)){
80         msg="*Se actualizó un registro";
81     // muestra();
82     }
83     else{    msg=("¡No existe!");
84     }
85     return "exito";
86 }
87
88 public String borrar() {
89
90     if(examendao.borrar(examen)){
91         msg="*Se eliminó información";
92     }
93     else{    msg="¡Verifique la clave por favor!";
94             listaExamen=examendao.listar();}
95     return "exito";
96 }
97

```

Fig.6. Código ExamenControl.

Entidades/ mapeo: se encuentran los POJOs y .xml generados después de realizar la conexión usando Hibernate.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
3  <!-- Generated 28/08/2014 01:35:06 AM by Hibernate Tools 3.2.1.GA -->
4  <hibernate-mapping>
5  <class name="Entidades.Examen" schema="public" table="examen">
6  <id name="claveUea" type="string">
7  <column length="7" name="clave_uea"/>
8  <generator class="assigned"/>
9  </id>
10 <property name="nombre" type="string">
11 <column length="60" name="nombre"/>
12 </property>
13 <property name="profesor" type="string">
14 <column length="60" name="profesor"/>
15 </property>
16 <property name="fecha" type="string">
17 <column length="8" name="fecha"/>
18 </property>
19 <property name="hora" type="string">
20 <column length="5" name="hora"/>
21 </property>
22 <property name="grupo" type="string">
23 <column length="5" name="grupo"/>
24 </property>
25 <property name="salon" type="string">
26 <column length="4" name="salon"/>
27 </property>
28 <property name="notas" type="string">
29 <column name="notas"/>
30 </property>
31 </class>
32 </hibernate-mapping>
33

```

Fig.7. Código Examen.xml

```

9   public class Examen implements java.io.Serializable {
10
11
12     private String claveUea;
13     private String nombre;
14     private String profesor;
15     private String fecha;
16     private String hora;
17     private String grupo;
18     private String salon;
19     private String notas;
20
21     public Examen() {
22     }
23
24
25     public Examen(String claveUea) {
26         this.claveUea = claveUea;
27     }
28     public Examen(String claveUea, String nombre, String profesor, String fecha,
29         this.claveUea = claveUea;
30         this.nombre = nombre;
31         this.profesor = profesor;
32         this.fecha = fecha;
33         this.hora = hora;
34         this.grupo = grupo;
35         this.salon = salon;
36         this.notas = notas;
37     }
38
39     public String getClaveUea() {
40         return this.claveUea;
41     }
42

```

Fig.8. POJO Examen

dao: paquete con las correspondientes interface e implementación de cada entidad de nuestra DB.

```

1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5   package dao;
6
7   import Entidades.Examen;
8   import java.util.ArrayList;
9
10  /**
11   *
12   * @author ytap
13   */
14  public interface daoExamen {
15      public boolean agrego(Examen examen);
16      public boolean actualizo(Examen examen);
17      public boolean borrar(Examen examen);
18      public ArrayList<Examen> listar();
19  }
20
21

```

Fig.9. Código Interface Examen

```

19  public class implementExamen implements daoExamen{
20      private Session sesion;
21      private Transaction tx;
22
23      @Override
24      public boolean agrego(Examen examen) throws HibernateException {
25          try{
26              iniciaOperacion();
27              //sesion = HibernateUtil.getSessionFactory().getCurrentSession();
28              //tx=sesion.beginTransaction();
29              Query query = sesion.createQuery("SELECT claveUea FROM Examen WHERE claveUea='"+examen.getClaveUea()+"");
30              List<String> listaResultados = query.list();
31
32              if(listaResultados.isEmpty()){
33                  sesion.save(examen);//guarda info
34                  tx.commit();//cambios a la base
35                  return true;
36              }
37              else{
38                  return false;
39              }
40
41          }catch (HibernateException he) {
42              manejaExcepcion(he);
43              throw he;

```

Fig. 10. Implementación examen

```

276 <!-- examen -->
277
278 <action name="AltaExamen">
279   <interceptor-ref name="secureStack" />
280   <result>/AltaExamen.jsp</result>
281   <result name="login">/Login.jsp</result>
282 </action>
283 <action name="BajaExamen">
284   <interceptor-ref name="secureStack" />
285   <result>/BajaExamen.jsp</result>
286   <result name="login">/Login.jsp</result>
287 </action>
288 <action name="ActualizarExamen">
289   <interceptor-ref name="secureStack" />
290   <result>/ActualizarExamen.jsp</result>
291   <result name="login">/Login.jsp</result>
292 </action>
293

```

Fig. 11. Action Examen

En el paquete Web Pages se encuentran las carpetas:

CSS: Hojas de estilo e imágenes a usar los html.

Meta-Inf: Contiene los archivos que son reconocidos e interpretados por la Plataforma Java 2 para configurar aplicaciones, extensiones, cargadoras y servicios de clase.

Web-Inf: Se encuentran todas las vistas .jsp además del web.xml (filtros que permiten el arranque del sistema).

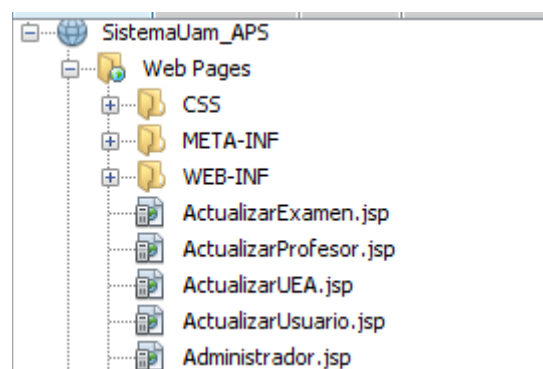


Fig. 11. Web Pages

6.1 Especificación Técnica

Los estándares y protocolos que se utilizaron son los siguientes:

SOFTWARE:

- La aplicación se desarrollo en lenguaje java JDK con el entorno de desarrollo NetBeans 7.1.2
- La base de datos se implemento en PostgreSQL 9.3.1
- El servidor de aplicaciones se monto en Apache Tomcat 6
- Los jars de Struts2.3
- Hibernate 3.1
- Lenguaje HTML (vistas) y CSS.
- Licencia Creative Commons de Reconocimineto NoComercialCompartirlgual

HARDWARE:

- Laptop HP pavillion dv5, Intel core i3, windows7, 4GB de RAM, 2.53 GHz. 600 GB HD.

6.1.1 Código para de tablas de la Base de Datos.

```
CREATE TABLE examen
(
  clave_uea character varying(7) NOT NULL,
  nombre character varying(60),
  profesor character varying(60),
  fecha character varying(8),
  hora character varying(5),
  grupo character varying(5),
  salon character varying(4),
  notas text,
  CONSTRAINT examen_pkey PRIMARY KEY (clave_uea)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE examen
  OWNER TO postgres;
```



```
CREATE TABLE profesor
(
  id_profesor character varying(12) NOT NULL,
  nombre character varying(60),
  cubiculo character varying(6),
  horario1 character varying(11),
  horario2 character varying(11),
  horario3 character varying(11),
  horario4 character varying(11),
  horario5 character varying(11),
  CONSTRAINT profesor_pkey PRIMARY KEY (id_profesor)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE profesor
  OWNER TO postgres;
```

```
CREATE TABLE uea
(
  clave_uea character varying(7) NOT NULL,
  nombre_uea character varying(60),
  creditos character varying(2),
  seriacion character varying(70),
  CONSTRAINT uea_pkey PRIMARY KEY (clave_uea)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE uea
  OWNER TO postgres;
```

```
CREATE TABLE usuario
(
  id_usuario character varying(9) NOT NULL,
  nombre character varying(60),
  contraseña character varying(12),
  rol character varying(20),
  CONSTRAINT usuario_pkey PRIMARY KEY (id_usuario)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE usuario
  OWNER TO postgres;
```

7. Resultados

7.1 Interfaz de Usuario Final.

Ingresar con la siguiente URL para visualizar la Interfaz de Usuario Final:
http://localhost:8084/SistemaUam_APS/index.html



Fig. 12. Interfaz usuario final.

7.1.2. Consulta UEA


Lleva al usuario visualizar el mapa curricular correspondiente a ingeniería en computación, obteniendo principalmente información de seriación.

http://cbi.azc.uam.mx/work/models/CBI/Documentos/Licenciaturas/IngComputacion/mapas_curriculares_Ing_Com.pdf

7.1.3 Consulta Examen.

Permite obtener información de UEA programadas para examen de recuperación.

Puede realizar búsqueda de UEA específica insertando la clave en el campo correspondiente.

Posteriormente puede general la lista completa dando clic en el icono 

Ir: llevara al usuario a la página de hipervínculos PDF proporcionados por los profesores sinodales de las UEA correspondientes.



El sistema de información web para exámenes de recuperación muestra una interfaz con un menú de navegación y una tabla de consulta de exámenes. La tabla contiene la siguiente información:

CLAVE UEA	NOMBRE UEA	PROFESOR(ES)	FECHA	HORA	GRUPO	SALON	BIBLIOGRAFIA
112033	Matemáticas Discretas	Eduin González	14-07-23	07:00	SCD0	E103	Matemáticas Discretas
113028	Programación Estructurada	Blanca Silvia Lopez	14-09-25	11:00	1902	E206	Programación Estructurada
3322712	prueba5	prueba4	14-07-23	12:00	F120	C302	prueba5
3222223	PRUEBA7	- PRUEBA	14-07-23	12:00	1902	C302	PRUEBA8
113042	Algoritmos y estructura de Datos	Hugo Pablo Leiva	14-08-29	12:00	1C12	C107	Algoritmos y estructura de Datos

Fig. 13. Consulta Examen.

7.1.4 Consulta Profesor

Permite visualizar Información de Profesor en caso de requerir asesoría personal para dudas específicas.

The screenshot shows a web browser window with the URL `localhost:8084/SistemaUam_APS/actionListarProfesor_caction`. The page title is "Sistema de Información Web Para Exámenes de Recuperación" with a date of "27 de Agosto de 2014". The main content area features a search bar with a "Buscar" button and a navigation menu with "Home", "UEA", "Exámen", and "Profesor". Below the search bar, there is a "MAY + REGISTRO" link and a "Consulta Profesor" heading. A table displays the following data:

ID PROFESOR	NOMBRE PROFESOR	CUBICULO	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES
0002	BLANCA SILVA LOPEZ	H-245	18:30-19:00	17:00-18:30	..
0004	José Luis Pantoja	H-233	16:00-17:00	18:00-19:00
0001	HUGO PABLO LEYVA	H-248	16:00-17:30	..
0005	Francisco Carvajal	H-237	16:00-17:00	11:00-14:00

Fig. 14. Consulta Profesor.

8. Conclusiones

Las aplicaciones Web son diferentes a los sitios Web convencionales porque éstas son capaces de crear una respuesta dinámica.

Muchos sitios Web ofrecen sólo páginas estáticas, en cambio una aplicación Web es capaz de interactuar con bases de datos y lógica de negocios con la finalidad de personalizar la respuesta a enviar.

La aplicación de este framework es muy extensa y de fácil uso, basta como ya se había mencionado antes aplicar lógica en el desarrollo del proyecto.

Por falta de tiempo no se logro un poco mas allá de lo propuesto, sin embargo el sistema cumple con los objetivos propuestos, esperando sea ayuda para muchos alumnos que tienen la necesidad de presentar estos exámenes de recuperación.

9. Bibliografía

[1] S.A. Galicia Alvarez. "Sistemas de gestión de manuales del Centro de Formación de Ingeniería Industrial", proyecto terminal, División de CBI, Universidad Autónoma Metropolitana Azcapotzalco, México, 2012.

[2] J.C. Zamora Trejo. "Sistema Web de seriación e información para alumnos de ingeniería", proyecto terminal, División de CBI, Universidad Autónoma Metropolitana Azcapotzalco, México, 2010.

[3] J.D. Mendoza Martínez y A.A. Santos Palacios. "Sistemas 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] Cátedra Movilidad y Educación, Universidad de Valladolid, "Proyecto EvaMovil: Sistema de Evaluación Interactiva para Terminales Móviles". Septiembre 2012. [En línea] Disponible en: <<http://ctme.uva.es/2012/09/17/proyecto-evamovil-sistemas-de-evaluacion-interactiva-para-terminales-moviles-2/>>

[5] Tecnologías Informáticas para la Web, Universidad Carlos III "Ejemplos TIW". [En línea]. Disponible en: <<http://www.lab.inf.uc3m.es/~a0080802/RAI/index.html>>

[6]"Tutoriales de Programación Java". [En línea]. Disponible en :<<http://www.javatutoriales.com/2011/06/struts-2-parte-1-configuracion.html>>

[7]"Tutorialspoint". [En línea] Disponible en: http://www.tutorialspoint.com/struts_2/

MANUAL DE USUARIO

10. Login Administrador

Para ingresar al “menú gestión” debe ingresar a la siguiente URL:

http://localhost:8084/SistemaUam_APS/Login.html

Con fines de seguridad le solicitara ingresar usuario y contraseña, estos son proporcionados por el Administrador principal, si olvida cualquiera de estos datos, deberá acudir con la persona correspondiente para solicitar esta información.

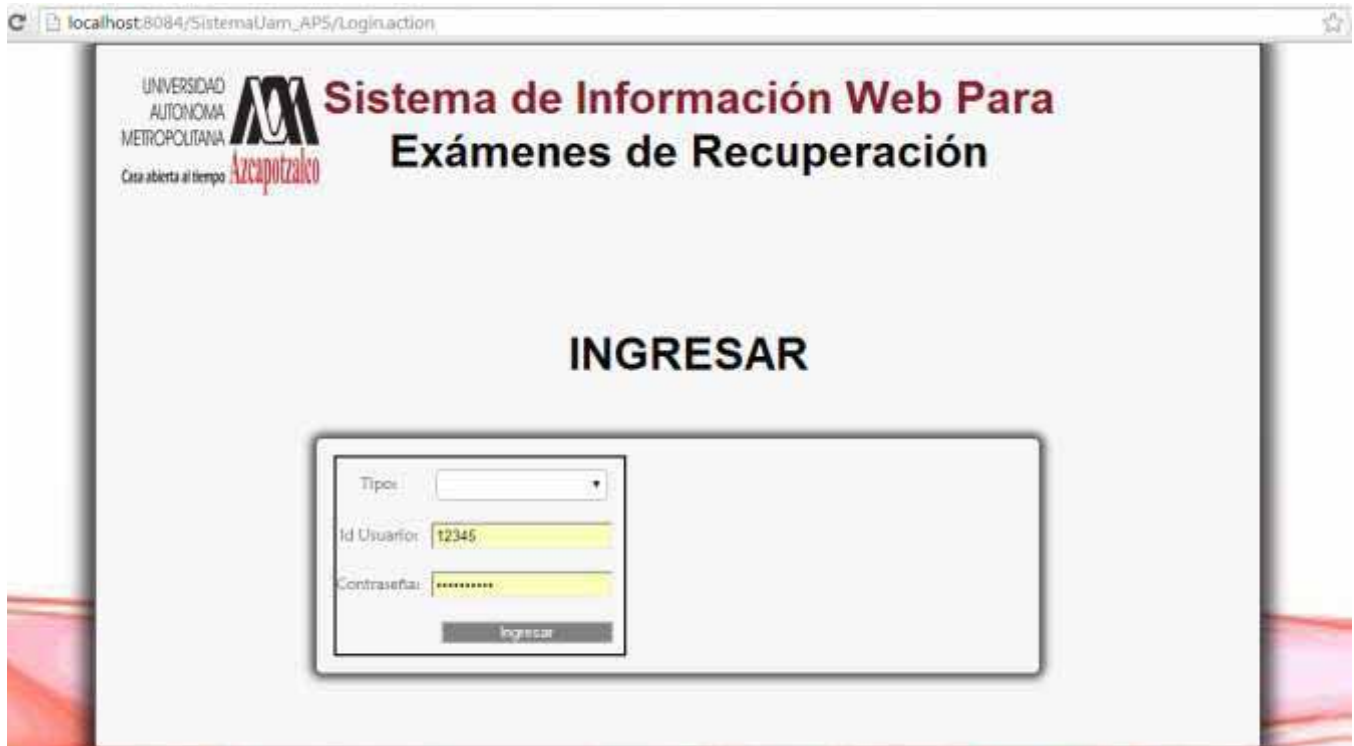


Fig. 15. Login.

10.1 Página principal Administrador

En esta página muestra al administrador las opciones de gestión: usuario, examen, profesor, archivos. La flecha verde le llevara a la interfaz “Usuario Final”



Fig. 16. Menú Administrador.

11. Gestión Examen

11.1 Alta Examen

En esta página podrá dar de alta nuevos a los exámenes programados tantos como lo desee. Las leyendas que aparecen en las cajas de textos especifican el formato a capturar la información, lo contrario significa generar errores esperados.

localhost:8084/SistemaUam_APS/AltaExamen.action

UNIVERSIDAD AUTÓNOMA METROPOLITANA **AM** *Casa abierta al tiempo* **Acapulco**

Sistema de Información Web Para Exámenes de Recuperación

27 de Agosto de 2014

Home Usuarios Examen Profesor Archivos SALIR

Alta Exámen

Clave UEA: 7 Dígitos

Nombre UEA:

Profesor:

Fecha: MM/DD/CC

Hora: HH:MM

Grupo: 0000

Salón: 000

Bibliografía:

Guardar

Fig. 17. Alta Examen.

11.2 Elimina Examen

Al dar clic sobre el icono  visualizará la siguiente pantalla:

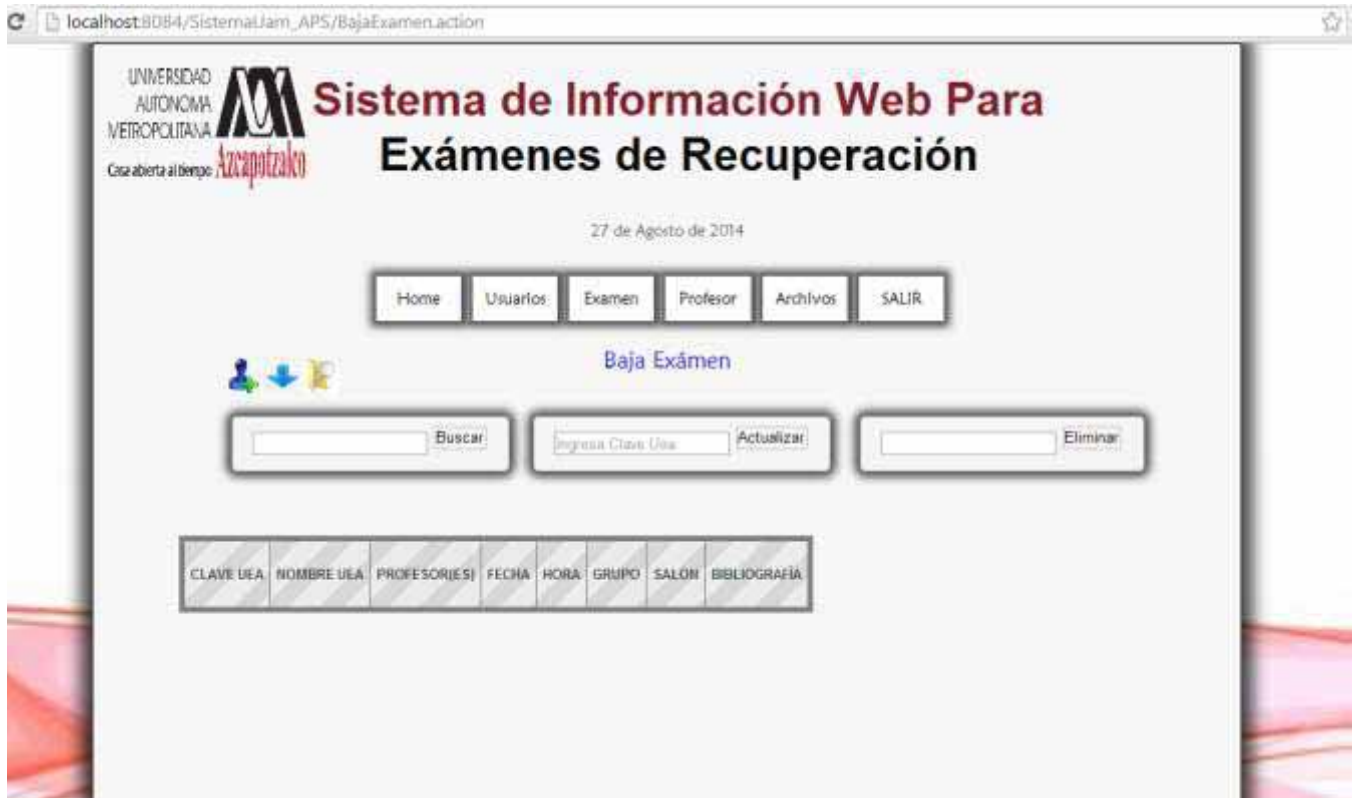



Fig. 18. Eliminar Examen.

De clic sobre el icono  para general la lista completa de registro de Examen.

11.2.1 Eliminar Información

UNIVERSIDAD AUTÓNOMA METROPOLITANA Azcapotzalco

Sistema de Información Web Para Exámenes de Recuperación

27 de Agosto de 2014

Home Usuarios Examen Profesor Archivos SALIR

Baja Exámen

Buscar Actualizar Eliminar

CLAVE UEA	NOMBRE UEA	PROFESOR(ES)	FECHA	HORA	GRUPO	SALON	BIBLIOGRAFÍA
1112033	Matemáticas Discretas	Efrén González	14-07-23	07:00	SCD2	E103	Matemáticas Discretas
1151038	Programación Estructurada	Bianca Silvia López	14-08-25	13:00	ST02	E306	Programación Estructurada
8765432	PRUEBA5	PEDRO	14-07-23	12:00	D000	E309	tf
3322112	prueba5	prueba4	14-07-23	12:00	F120	G302	prueba5
2222223	PRUEBA7	PRUEBA	14-07-23	12:00	ST02	G302	PRUEBA8
1151042	Algoritmos y estructura de Datos	Hugo Pablo Leyva	14-08-29	12:00	SC12	G307	Algoritmos y estructura de Datos

Fig. 19. Gestión examen.

Buscar

Fig. 20. Buscar.

Busca y muestra información de la Examen de la clave ingresada.

Ingresar Clave Uea Actualizar

Fig. 21. Actualizar.

Le muestra una nueva página con la información del examen de la clave insertada. En esta página puede realizar modificaciones (excepto de clave de UEA).

11.3 Actualizar Examen

El administrador podrá realizar gestión de modificación de información. Cuando considere que ha realizado los cambios necesarios dar clic en actualizar, el sistema enviara mensaje de confirmación. Puede visualizar los cambios en el menú consultas.



Fig. 22. Actualizar examen.



Fig. 23. Eliminar

Ingresar clave de UEA para eliminar de la DB.

Iconos:



"Alta": Visualizará la interfaz para dar de alta.



"Limpiar": Permite limpiar datos almacenados en los campos.



"Eliminar": Muestra la interfaz para eliminar.



"Actualizar": Muestra la Interfaz para realizar modificaciones de información.



"Consultar": Le permite visualizar la lista general almacenada en la DB.



"Generar Lista": En la Interfaz "Eliminar" le permite generar la lista general actual.



"Actualizar": Después de haber realizado una búsqueda en la Interfaz "Consulta" le permite generar nuevamente la tabla completa.