

# DESARROLLO DE APLICACIONES WEB

Proyecto: FCT Hatore

Diseñado por:

Alejandro Oses Armentas

Sergio Lucena Fernández

## **FCT Hatore – Índice**

### **ÍNDICE:**

**INTRODUCCIÓN Y JUSTIFICACIÓN. *Página 4.***

**OBJETIVOS. *Página 5.***

**ANÁLISIS DEL CONTEXTO. *Páginas 6-7.***

**DESARROLLO DEL CONTENIDO. *Páginas 8-22.***

#### **LENGUAJES DE PROGRAMACIÓN:**

- JAVA. *Páginas 9-11.*

#### **LENGUAJES DE HOJAS DE ESTILOS:**

- CSS 3. *Página 11.*

#### **LENGUAJES DE MARCAS:**

- HTML 5. *Página 11.*
- XML (Configuraciones). *Páginas 11-12.*

#### **LENGUAJES DE SCRIPTING:**

- JAVASCRIPT Y AJAX. *Página 12.*

#### **LENGUAJE DE ACCESO A BASE DE DATOS:**

- SQL. *Páginas 12-13.*

#### **ENTORNOS DE DESARROLLO (IDE):**

- ECLIPSE. *Página 13.*
- MYSQL WORKBENCH. *Página 13.*

#### **SERVIDOR:**

- TOMCAT 7. *Página 14.*

#### **GESTOR DE DEPENDENCIAS**

- MAVEN. *Páginas 14-15*

#### **GENERADOR DE MAPEOS:**

- HIBERNATE. *Páginas 15-17.*

#### **FRAMEWORKS:**

- BOOTSTRAP. *Página 17.*
- JSF (JAVA SERVER FACES). *Página 18.*

**BIBLIOTECAS:**

- JQUERY. *Página 18.*
- JAVAX MAIL. *Página 18.*
- FULLPAGE. *Página 19.*
- JAVA.SECURITY. *Página 19.*
- JDBC (JAVA DATABASE CONNECTIVITY). *Páginas 19-20*
- JAVAX SERVLET. *Página 20.*

**TECNOLOGÍAS:**

- JDK 8 (JAVA DEVELOPMENT KIT V. 8). *Página 20.*
- J2SE (JRE, JAVA RUNTIME ENVIRONMENT). *Página 20.*
- JSP (JAVA SERVER PAGES). *Página 21.*

**APLICACIÓN DE CONTROL DE VERSIONES:**

GITHUB; Usuarios: SergioLucenaFdz, Xelka Equipo: Codimasters. *Página 22.*

**EXPLICACIÓN DETALLADA DE LAS PARTES QUE  
COMPRENDEN EL PROYECTO. *PÁGINAS 22-24.***

**CONCLUSIÓN Y VALORACIÓN PERSONAL. *Páginas 25-26.***

## INTRODUCCION Y JUSTIFICACIÓN

Nuestro proyecto consiste en una aplicación web para la gestión de la Formación en Centros de Trabajo, en los cuales intervienen empresas que se darán de alta, y se les asignarán alumnos de los distintos centros para la culminación de su formación académica. Para dicha gestión, se asignan profesores a alumnos, y tutores de la empresa que quieran participar en la FCT.

Este proyecto será útil tanto para alumnos, como profesores, como empresas, debido a que la antigua interfaz está muy desactualizada y no es muy intuitiva para los usuarios. Además, favorece la integridad de los datos debido a la unificación de los mismos en una Base de Datos. También evita pérdida de información, ya que todo está dentro de un sistema bien organizado.

Cuenta con un uso más cómodo y dinámico para cualquier tipo de usuario, mediante una “responsive interface”, en conjunción con unas de las tecnologías más punteras disponibles, referente al ámbito programático.

Los usuarios contarán con una gran cantidad de información que de otra forma sería mas problemática de obtener, ya que cuenta con una base de datos muy bien relacionada.

La aplicación dará la oportunidad al administrador de gestionar cualquier tipo de dato, los directores podrán asignar nuevos profesores a sus centros o eliminarlos, los profesores podrán asignarse sus alumnos o eliminarlos de su asignación, etc. Es decir, existe una jerarquía piramidal de permisos en la aplicación, en la cual un usuario manda sobre otro.

## OBJETIVOS

Los objetivos que se desean conseguir con la aplicación son:

1. Fomentar el buen funcionamiento de los centros educativos, y favorecer las relaciones entre ellos mismos y las empresas de la zona:

La interfaz permitirá una vista sencilla a la vez que una funcionalidad completa, es decir, al usuario no le costará trabajo habituarse a la navegación de la aplicación y podrá hacer un manejo rápido y eficaz de los datos relacionados entre centro y empresa.

2. Mejorar la gestión de las FCT:

Se podrá llevar un control sobre los datos de las prácticas de cada alumno, además contará con una interfaz rápida y “user-friendly”.

3. Agilizar el proceso de asignación de alumnos a sus respectivos centros de trabajo:

El administrador, director y profesor podrá asignar o eliminar a alumnos de los tutores de la empresa.

4. Conseguir una buena respuesta por parte de los usuarios que la controlen.

Al ser una aplicación sencilla y con una barra de navegación óptima, el acceso a todos los elementos será muy rápido y fluido.

5. Mejorar la calidad de la antigua aplicación web.

6. Facilitar todo el proceso de la FCT

Durante todo el tiempo que dure la FCT se podrán acceder a diversos datos que nos indiquen donde se encuentra cada alumno.

7. Mantener el máximo control posible sobre la información que le corresponde a cada usuario.

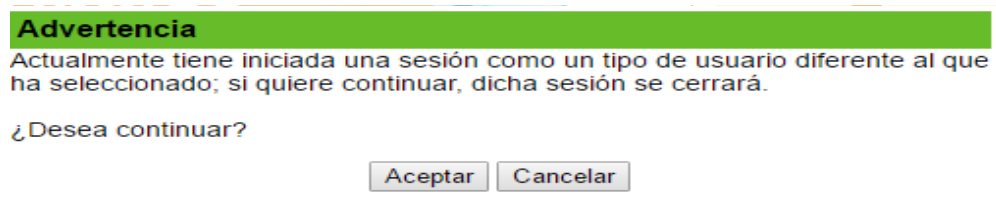
Los usuarios, podrán customizar a su gusto la gran mayoría de la interfaz y editar sus propios datos y los datos de los usuarios que estén bajo su responsabilidad.

## ANÁLISIS DEL CONTEXTO

Para empezar, nuestra principal premisa es la comodidad para el usuario, por lo que nos esforzamos en realizar una página en la que no haya que buscar que tipo de usuario eres como en la anterior página de “fpnet”.



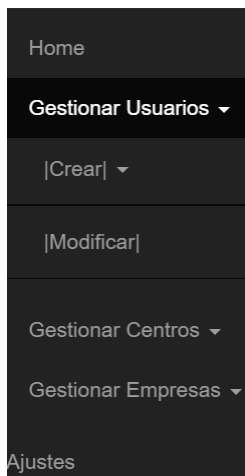
Unos de los principales problemas de dicha página, consiste en que todo tipo de usuario, tiene visibilidad sobre todos los paneles, lo que hace de una experiencia poco agradable e intuitiva para el usuario.



Toda la navegación es pausada y con “pop-ups”, haciendo que sea poco fluida.

Por lo que un diseño en el que cada usuario tendrá un panel propio, facilitará mucho la interactividad de la página, lo que ahorra tiempo para el usuario, además de aumentar la satisfacción del mismo. Con esto conseguimos un buen “feedback” a la vez que facilitamos los trámites y las gestiones dentro de la misma aplicación.

A demás, contamos con una barra de navegación con “dropdowns” (Menús desplegables), por ejemplo, nuestro panel de administrador tiene la parte de usuarios, la parte de centros, y la parte de empresas. En cada uno de ellos, tendremos diferentes opciones administrativas.



Éste es nuestro menú desplegado en tamaño de pantalla adaptado a dispositivos pequeños. Como veréis contaremos con un enlace a la página principal, además de diferentes menús desplegables. Como podéis observar, el panel de gestión de usuarios, está desplegado. En él contaremos con las opciones de crear cualquier tipo de usuario y de modificar a cualquier usuario de la base de datos.

Referente a la estética, en la aplicación de “fpnet”, no hay forma de editarla, sin embargo, en nuestra aplicación hemos incluido un apartado exclusivamente diseñado para que cada usuario pueda elegir entre una amplia variedad de temas.



*Como podemos apreciar, las secciones de la pagina son las mismas, pero con diferentes temas a gusto del usuario.*

A diferencia de “fpnet”, nuestra página, tendrá en la mayoría de sus secciones, una única carga principal de todos los contenidos para poder navegar entre ellas de la forma más fluida y rápida posible, por lo que el usuario, no notará apenas que está cambiando de una página a otra o que está cambiando de una funcionalidad a otra totalmente distinta.

## **DESARROLLO DEL CONTENIDO**

A continuación, hablaremos de las apis, herramientas, y bibliotecas, lenguajes de marcas, lenguajes de programación, lenguajes de scripting y lenguajes de gestores de bases de datos empleados para el desarrollo del proyecto:

### **LENGUAJES DE PROGRAMACIÓN:**

- JAVA

### **LENGUAJES DE HOJAS DE ESTILOS:**

- CSS 3

### **LENGUAJES DE MARCAS:**

- HTML 5
- XML (Configuraciones)

### **LENGUAJES DE SCRIPTING:**

- JAVASCRIPT Y AJAX

### **LENGUAJE DE ACCESO A BASE DE DATOS:**

- SQL

### **ENTORNOS DE DESARROLLO (IDE):**

- ECLIPSE
- MYSQL WORKBENCH

### **SERVIDOR:**

- TOMCAT 7

### **GESTOR DE DEPENDENCIAS**

- MAVEN

### **GENERADOR DE MAPEOS:**

- HIBERNATE



FRAMEWORKS:

- BOOTSTRAP
- JSF (JAVA SERVER FACES)

BIBLIOTECAS:

- JQUERY
- JAVAX MAIL
- FULLPAGE
- JAVA.SECURITY
- JDBC (JAVA DATABASE CONNECTIVITY)
- JAVAX SERVLET

TECNOLOGÍAS:

- JDK 8 (JAVA DEVELOPMENT KIT V. 8)
- J2SE (JRE, JAVA RUNTIME ENVIRONMENT)
- JSP (JAVA SERVER PAGES)

APLICACIÓN DE CONTROL DE VERSIONES:

- GITHUB; Usuarios: SergioLucenaFdz, Xelka Equipo: Codimasters

## JAVA

Java es un lenguaje de programación orientado a objetos y de propósito general (Esto quiere decir que puede emplearse tanto para desarrollo de aplicaciones web, como para desarrollo de aplicaciones de escritorio), que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. La intención principal de éste lenguaje es que los programas en el mismo se escriban solamente una única vez, y que se puedan ejecutar en cualquier dispositivo (Multi-Plataforma).

El lenguaje de programación Java fue originalmente desarrollado por James Gosling de la empresa Sun Microsystems, que posteriormente fue adquirida por Oracle Corporation. Su sintaxis deriva en gran medida de C y C++, pero con menos utilidades a bajo nivel que ellos. Las aplicaciones de Java son compiladas en bytecode, para ser ejecutado en cualquier JVM (Máquina virtual de Java).

Hemos empleado Java en nuestro proyecto para la creación de las clases que serán utilizadas y compiladas dentro del mismo.

Las entidades de la base de datos que no implican una tabla de relación n:m serán convertidas a clases dentro de nuestra aplicación, a demás, hemos creado clases para los diferentes controladores de cada entidad. Los controladores que hemos usado han sido OperacionesBD, que tendrán todos los métodos necesarios para las consultas de MySQL, como por ejemplo el Login, o el Registro de usuarios. Los usuarios se almacenan en objetos tras las funcionalidades del Login, que previamente han sido insertados mediante otro método de la misma clase en el Registro a la base de datos.

Para ello hemos tenido que importar los siguientes paquetes:

- packageConexion: Una clase creada por nosotros que será la encargada de realizar la conexión con la base de datos.
- Java.sql: En el que más adelante hablaremos de el.
- Md5: Una clase que usaremos para pasar las contraseñas a md5.
- Email: Hablaremos más adelante de el.
- Random: Usado para generar contraseñas aleatorias para los responsables de empresa que más adelante se explicará el propósito de esta función.
- Entities: Paquete generado por hibernate, en su apartado hablaremos de este apartado.

Cada objeto que creamos en el login cuenta con alrededor de 11 objetos en su interior, debido a que el objeto usuario, que sera el generado, contiene en su interior otros objetos de otras clases que a su vez contienen más objetos debido a las relaciones de la base de datos

## **CSS 3**

El Lenguaje de Estilo en Cascada se emplea en la creación y definición de presentaciones de documentos HTML o XML. El W3C es el encargado de formular los estándares de especificación de las hojas de estilo.

El CSS3 en nuestra aplicación ha sido utilizado en toda empleando la mayor parte de su funcionalidad extendida, es decir, aplicar estilos a elementos html por medio de pseudo-clases con llamada a elementos tanto block como in-line, es decir, llamando a contenedores o aplicando el estilo directamente al elemento.

## **HTML5**

Es un lenguaje de marcas usado para la elaboración de páginas web. Está constituido en un diseño en forma de árbol esto quiere decir que las etiquetas van unas dentro de otra, formando así contenedores.

Estamos empleando HTML en su versión más reciente, HTML versión 5, haciendo uso de etiquetas incluidas en esta versión para su estructura, como por ejemplo, las etiquetas “nav”, ”footer”, “section”, etc.

## **XML**

Es un lenguaje de marcas utilizado para almacenar datos de forma legible. En principio, su estructura es igual que la de html, es decir una estructura en forma de árbol.

En nuestra aplicación hemos usado lenguaje xml para editar configuraciones de diferentes archivos que utiliza java o el servidor, como por ejemplo web.xml, reveng.xml, etc.

## **JAVASCRIPT Y AJAX**

JavaScript es un lenguaje de programación orientado a objetos usado para generar eventos dinámicos en el frontend( parte visual de la página web), debido a que todo el código se ejecutará en el lado del cliente. Javascript se utiliza para interactuar dinámicamente con el DOM del documento.

Ajax es una técnica de desarrollo usado en Javascript para crear aplicaciones interactivas. Estas aplicaciones realizan peticiones al servidor sin tener que recargar la página, por lo que éstas serán mucho mas rápidas, cómodas y fluidas.

En nuestro proyecto, hemos usado Javascript para declarar eventos onClick( cuando se clickea en un elemento) y onKeyUp(cuando se levanta la tecla del teclado) para activar funciones que interactuarán mediante Ajax para hacer las consultas en la base de datos pertinentes para la creación de filtros, eliminación de datos de la base de datos y para hacer formularios interactivos para el registro. Una vez pasado el proceso de Ajax, se devolverán los datos a Javascript y se modificará el dom, borrando elemntos del html, incluyendo nuevos, etc, todo esto sin recargar la página.

## **SQL**

SQL es un lenguaje declarativo de acceso a bases de datos relacionales. Permite especificar distintos tipos de operaciones entre ellas. Se caracteriza por su manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar, de forma sencilla, información de bases de datos, así como realizar cambios en ellas.

En nuestro proyecto hemos empleado SQL tanto de forma indirecta, para crear las tablas de la base de datos, a través de la herramienta de desarrollo MySQL Workbench, de la que hablaremos más adelante, como de forma directa para las consultas a la base de datos.

## **ECLIPSE**

Eclipse es un software compuesto por un conjunto de herramientas de programación “Open-Source” multiplataforma. Ésta plataforma es típicamente utilizada para el desarrollo de IDEs, como el IDE de Java denominado JDT (Java Development Toolkit) y el compilador ECJ, a parte de ser también una comunidad de usuarios.

Eclipse fué desarrollado originalmente por IBM como el sucesor de su familia de herramientas VisualAge, que más tarde fue a pasar a manos de la Fundación Eclipse (Eclipse Foundation).

Nuestro proyecto ha sido desarrollado completamente en Eclipse, haciendo uso de sus amplias configuraciones y herramientas integradas.

## **MySQL Workbench**

MySQL Workbench es una herramienta visual unificada para arquitectos de bases de datos y desarrolladores. MySQL Workbench ofrece modelado de datos, desarrollo de SQL y un conjunto de herramientas comprensivas para configuración de servidores, administración de usuarios, copias de seguridad, etc. MySQL Workbench está disponible en Windows, Linux y Mac OS X.

La base de datos está realizada en su totalidad empleando ésta herramienta, haciendo conexión a un servidor externo en el que está hospedada la misma.

## **TOMCAT**

Tomcat es un contenedor web con soporte de servlets y JSPs, Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de tomcat a menudo se presenta en combinación con el servidor web Apache.

Hemos usado Tomcat en su versión 7 para el despliegue de la aplicación.

## **MAVEN**

Maven es una herramienta open-source con el objetivo de simplificar los procesos de build (compilar y generar ejecutables a partir del código fuente).

Por otra parte, con Maven la gestión de dependencias entre módulos y distintas versiones de librerías se hace muy sencilla. En este caso, solo tenemos que indicar los módulos que componen el proyecto, o qué librerías utiliza el software que estamos desarrollando en un fichero de configuración de Maven del proyecto llamado POM.

A demás, en el caso de las librerías, no tienes ni que descargarlas a mano. Maven posee un repositorio remoto donde se encuentran la mayoría de librerías que se utilizan en los desarrollos de software y que la propia herramienta se descarga cuando sea necesario

Podríamos decir que Maven es una herramienta capaz de gestionar un proyecto software completo desde la etapa en la que se comprueba que el código es correcto, hasta que se despliega la aplicación, pasando por la ejecución de pruebas y generación de informes y documentación.

Maven establece una estructura común de directorios para todos los proyectos. Por ejemplo, el código está en \$(raíz del proyecto)/src/main/java, los recursos en \$(raíz del proyecto)/src/main/resources. Los tests están en \$(raíz del proyecto)/src/test.

En Maven se definen tres ciclos de build del software con una serie de etapas diferenciadas.

Para instalar Maven en Eclipse, es tan sencillo como acudir a la página de Maven for Eclipse (m2e), copiar la url de descarga apta para administrar paquetería de Eclipse, una vez en Eclipse ir a help->install new software, y en el cuadro de búsqueda introducir la url: <http://download.eclipse.org/technology/m2e/releases>, esperar a que Eclipse detecte los paquetes, ticarlos y darle a la pestaña de siguiente, para que el gestor instale los paquetes seleccionados.

Para instalar Maven en Eclipse, es tan sencillo como acudir a la página de Maven for Eclipse (m2e), copiar la url de descarga apta para administrar paquetería de Eclipse, una vez en Eclipse ir a help->install new software, y en el cuadro de búsqueda introducir la url: <http://download.eclipse.org/technology/m2e/releases>, esperar a que Eclipse detecte los paquetes, ticarlos y darle a la pestaña de siguiente, para que el gestor instale los paquetes seleccionados.

Hemos usado Maven para generar toda la estructura de archivos del proyecto además de la gestión de las dependencias que necesitamos en el proyecto.

## **HIBERNATE**

Hibernate es una herramienta de mapeo, objeto-relacional para la plataforma JAVA, que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación mediante archivos declarativos

(xml) o anotaciones en los beans, de las entidades que permiten establecer estas relaciones.

Hibernate busca solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: El usado en la memoria de la computadora (orientado a objetos) y el usado en las bases de datos (modelo relacional). Para lograr esto, permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y que forma tienen. Con ésta información Hibernate permite a la aplicación manipular los datos en la base de datos operando sobre objetos. Hibernate convertirá los datos entre los tipos utilizados por Java y los definidos por SQL. Hibernate genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de BBDD con un ligero incremento en el tiempo de ejecución.

Hibernate está diseñado para ser flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente. También tiene la funcionalidad de crear la base de datos a partir de la funcionalidad disponible. Hibernate ofrece también un lenguaje de consulta de datos llamado HQL, al mismo tiempo que una api para construir las consultas programáticamente, conocida como Criteria.

Hibernate para Java puede ser utilizado en aplicaciones Java independientes o en aplicaciones Java EE, mediante el componente Hibernate Annotations que implementa el estándar JPA, que es parte de ésta plataforma.



Para instalar el plugin Hibernate y usarlo como dependencia en un proyecto Maven, es tan sencillo como abrir el archivo pom.xml del proyecto y en la etiqueta dependencies, abrir una etiqueta dependency, haciendo referencia al plugin Hibernate de la siguiente manera:

```
<dependency>  
  <groupId>org.hibernate</groupId>  
  <artifactId>hibernate-core</artifactId>  
  <version>5.1.0.Final</version>  
</dependency>
```

En nuestra aplicación hemos usado Hibernate para crear las clases de las entidades de la base de datos, estas se crean con sus métodos get y set para cada atributo de la base de datos.

## **BOOTSTRAP**

Bootstrap es un framework o conjunto de herramientas de código abierto para el diseño de sitios y aplicaciones web. Incluye plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación, etc. Es el proyecto más popular de GitHub y es usado por la NASA y la MSNBC junto a las demás organizaciones.

Hemos empleado Bootstrap en nuestro proyecto como base de nuestra interfaz de usuario. Haciendo uso de una pequeña parte de sus funcionalidades, y de sus sistema de rejillas para la posición de los elementos.

## **JSF**

JavaServer Faces es una tecnología y framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuarios en aplicaciones Java EE.

JSF incluye un conjunto de APIs para representar los distintos componentes de una interfaz de usuario y administrar su estado, manejar distintos eventos y validar entradas y definir un esquema de navegación de páginas.

Hemos empleado JSF para la correcta funcionalidad de la interfaz del usuario, a demás hemos configurado su xml para definir la ruta en la cual el servidor Tomcat hace uso de la misma tecnología, asignando /faces/\* como la misma ruta.

## **JQUERY Y BOOTSTRAP VALIDATOR**

Es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Jquery es software libre y de código abierto.

Hemos empleado Jquery en nuestro proyecto en el plugin de Bootstrap Validator, para las validaciones de los formularios.

## **Javax Mail**

JavaMail es una API que brinda un framework para crear emails, y aplicaciones de mensajería. La API de JavaMail está incluida de forma opcional para su uso en la plataforma Java SE, y también está incluida en la plataforma Java EE.

Hemos empleado ésta API para enviar mails con las contraseñas y los usuarios de distintos usuarios que necesiten de un tratamiento más específico, como lo son los Responsables de Empresa, Directores, etc.

## **FULLPAGE**

FullPage.js es un plugin basado en JQuery que permite crear páginas de scrolling dentro de un mismo documento html. Creado por el desarrollador web Álvaro Trigo. Además, este plugin ofrece una documentación bien organizada con muchos ejemplos hechos a mano.

Hemos empleado este plugin para las navegaciones entre las distintas secciones de las páginas, para darles muy buenos efectos visuales, y fluidez extrema al usuario, además de que al importar los html en las diferentes secciones, nos ahorramos la carga de estas.

## **JAVA SECURITY**

Brinda un número de características diseñadas para mejorar la seguridad de las aplicaciones Java. Incluye restricciones durante la ejecución de la máquina virtual de java.

Hemos usado java security para implementar la codificación en MD5 de las contraseñas, de modo que cuando un usuario se registra, su contraseña pasa a ser en la base de datos una clave encriptada y cuando este se logea, comparamos la clave de la base de datos con la contraseña que introduce en el formulario del login previamente encriptada también a MD5.

## **JDBC**

Es una API de java que permite la ejecución de operaciones sobre bases de datos desde este lenguaje, independientemente del sistema operativo donde se ejecute o la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice.

Esta API, se presenta como una colección de interfaces Java y métodos de gestión de manejadores de conexión hacia cada modelo específico de base de datos.

La API JDBC, ha sido empleada en todas las características del “backend” encargadas de la inserción, actualización, eliminación y selección de datos dentro de la base de datos para luego mostrar un resultado en el “frontend”, mediante deferentes procesos como pueden ser “submits” normales de formularios con recarga hasta AJAX , en el que hemos usado el método GET, que aunque sea menos seguro, no tiene importancia, debido a que no se está pasando ningún dato peligroso, como podría ser una contraseña.

### **JAVAX SERVLET**

Es una clase de Java, utilizada para ampliar las capacidades de un servidor. Aunque los servlets pueden responder a cualquier tipo de solicitudes, éstos son utilizados comunmente para extender las aplicaciones alojadas por servidores web, de tal manera que pueden ser vistos como applets( pequeños programas que se ejecutan en el contexto de un navegador web) de java que se ejecutan en servidores en vez de navegadores web.

Es necesario para correr en Tomcat nuestra aplicación importar el paquete servlet-api.jar.

### **JDK**

Java Development Kit, es un software que provee herramientas de desarrollo para la creación de programas en Java.

Usamos JDK en su versión 8 para poder desarrollar nuestro proyecto.

### **J2SE**

Es una colección de APIs de Java. En esta api vienen incluidas APIs tales como java.security y java.sql(de sql JDBC) mencionadas anteriormente.

## **JAVASERVER PAGES**

Es una tecnología que ayuda a los desarrolladores de software a crear páginas web dinámicas basadas en HTML, XML, entre otros tipos de documentos. JSP es similar a PHP pero en Java.

Para poder desplegar JSP se requiere un servidor compatible con contenedores servlet como Tomcat.

La principal ventaja de JSP frente a otros lenguajes es que el lenguaje Java es un lenguaje de propósito general que excede el mundo web. Esto permite separar en niveles las aplicaciones web, dejando la parte encargada de generar el documento HTML en el archivo JSP.

Otra de las ventajas de la portabilidad de Java. Es común que los desarrolladores trabajen en una plataforma y que la aplicación termine siendo ejecutada en otra.

Hemos usado JSP para la creación de las páginas web dinámicas de nuestra página, además de para instanciar los objetos de las clases creadas en java y llamar a sus diversos métodos y asignar variables de sesión.

Centrandos en el último uso, podemos decir que el login generará un objeto de la clase usuario que contendrá en su interior más objetos con toda la información del usuario que se está logeando y este objeto que nos devuelve el método que controla el login, lo instanciaremos como una variable de sesión para evitar realizar consultas a la base de datos, debido a que ya tendremos toda la información necesaria, como los enlaces a fotos, el nombre del usuario logeado, etc.

## **GITHUB**

Es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git.

Hemos usado GitHub para la subida del código fuente que hemos programado para llevar a cabo un control de versiones, además de demostrar que ambos compañeros hemos trabajado en el código.

### **Explicación detallada de las partes que comprenden el proyecto:**

Principalmente, hemos creado un paquete conexión, que contendrá una clase encargada de efectuar todas las conexiones con la base de datos y las desconexiones.

Una vez terminado éste proceso, hemos hecho un paquete controladoresConexión, que alberga la clase OperacionesBd, en la que incluimos diferentes métodos relacionados con el registro y el login de la app.

El registro se dividirá en diferentes métodos estáticos para cada tipo de usuario. Cada método comprobará si el nombre de usuario existe ya en la base de datos, y si no existe, insertará los datos en sus respectivas tablas.

Fijándonos en el frontend, el registro contará con distintos formularios, algunos de ellos generados mediante AJAX, para obtener los datos necesarios para la inserción del tipo de usuario pertinente. En otros casos, el registro del usuario solamente lo podrá hacer el Admin, como por ejemplo, el tipo de usuario Responsable de Empresa. A éste usuario, se le generará una contraseña automática, y con un email la recibirá y podrá acceder con ella a su panel, donde podrá editar todos sus datos personales, incluyendo el nick y la contraseña.

Al principio del login hemos declarado todas las variables y objetos necesarios para la creación de un objeto final de tipo Usuario, que albergará en su interior objetos como por ejemplo: El objeto de tipo Alumno, o el objeto de tipo Profesor. Todos estos objetos y variables, estarán inicializadas con un valor por defecto para evitar que hayan atributos nulos, evitando así errores de compilación.

Para el desarrollo del login, hemos evaluado los datos que se necesitarán de forma general para todos los usuarios y divididos mediante condiciones las consultas específicas para cada tipo. El último paso realizado en el login ha sido asignarle mediante el método set todos los atributos con los datos recopilados a cada objeto y luego, generar el objeto usuario que contendrá como argumentos los objetos anteriores y retornar el objeto final para asignárselo a una variable de sesión.

Ahora, fijándonos en el frontend, el login simplemente será un formulario con un campo usuario, y un campo contraseña, y mediante el método POST se enviarán los datos pasados como argumento al método de OperacionesBd encargado del login.

Una vez comprobado que el usuario existe, se le redirigirá a su panel correspondiente, que contendrá todas las funcionalidades que necesita en función de su rol dentro de la app.

En el panel, contaremos con una portada en la vista principal, además de una barra de navegación para que el usuario navegue hacia las diferentes secciones del panel.

Para el Alumno, las opciones con las que contará serán las de editarse a sí mismo, darse de baja, observar su tutor, su profesor y la información referente al centro, como en que centro está registrado o la familia profesional en la que está.

Para el profesor, las opciones con las que contará serán las de editarse a sí mismo, darse de baja, observar a los tutores con los que mantiene contacto, la información relevante de sus alumnos y de su centro.

Para el responsable de la empresa, las opciones con las que contará serán las de editarse a sí mismo y a su empresa, darse de baja a el y a su empresa, observar sus datos, los de su empresa, y observar información relevante sobre sus tutores así como asignar alumnos libres a sus tutores.

Para el tutor, las opciones con las que contará serán las de editarse a sí mismo, darse de baja, observar su información, la de su empresa y la de los alumnos que tiene a cargo.

Para el director, las opciones con las que contará, serán las de editarse a sí mismo, darse de baja, editar su centro, registrar profesores, asignar profesores a los alumnos y eliminar profesores.

En los anexos, se mostrarán fotos del diagrama entidad relación así como de los paneles de los usuarios.



## **CONCLUSIÓN Y VALORACIÓN PERSONAL**

Haber realizado el proyecto con Java, ha sido una experiencia muy enriquecedora gracias a la oportunidad de aprender éste lenguaje debido a la gran portabilidad y potencia que ofrece. Gracias a esto nos hemos dado cuenta de que somos capaces de aprender un nuevo lenguaje por nuestra cuenta. Con Java simplemente hemos tardado al rededor de un mes en realizar el proyecto, aunque nos hubiera gustado implementar muchas mas funcionalidades y haber evaluado de una forma más óptima las funcionalidades implementadas, pero debido a las diversas dificultades por las que hemos pasado como la baja de nuestro tutor, aprender un lenguaje y herramientas completamente desconocidas para nosotros hasta la fecha, nos ha resultado imposible, aun así, pensamos que para el tiempo que hemos tenido, el proyecto ha resultado ser bastante competente.

El hecho de haber usado AJAX nos ha hecho ver desde otro punto de vista las páginas web dinámicas, haciéndolas mucho mas vistosas y enriquecedoras para el usuario. El resultado final del proyecto, no está lo completo que nos hubiera gustado debido a que es a nuestro parecer, el proyecto más largo que se ha asignado, teniendo un número de hasta 6 tipos de usuario, cada uno con su panel correspondiente, con lo que eso conlleva a la creación de diferentes plantillas para cada uno, a demás de distintas consultas para la base de datos.

El uso de clases y objetos, gracias al último consejo que nos dió Javier Morillas Zafra, nuestro tutor, nos dió una visión mucho mas elaborada sobre la potencia de las clases y objetos, gracias a ese pequeño consejo, pudimos implementar en el login la variable de sesión con el objeto dentro, por el que nos resultó muy util para el desarrollo de la aplicación.

Entre lo más satisfactorio que nos llevamos de la experiencia, son la creación del login, y enviar emails a través de Java, algo que nos llevó un buen rato descubrirlo, y con el que nos divertimos mucho una vez conseguido el resultado esperado, ya que hicimos el testeo de la api enviándole a algunos de nuestros compañeros hasta 8 mensajes consecutivos debido a que usamos un bucle para enviar mensajes.

El compañerismo ha sido una de las principales experiencias más gratificantes, debido a que eso nos ha dado la oportunidad de poder trabajar hasta 18 horas en un mismo día (Contando tanto las horas de FCT como las trabajadas en el proyecto), que de no haber sido por la compañía y la diversión que eso conlleva hubiera sido prácticamente imposible, a demás, ha existido una excelente compenetración debido a que a Sergio Lucena le encanta la parte del frontend, y a Alejandro Oses le gusta la parte de backend, por lo que la experiencia ha sido fluida, a demás, cuando parecía que no había final y se nos echaba el tiempo encima, nos apoyábamos mutuamente para hacerlo todo más llevadero y para hacer el relevo de turnos.

Ante todo la experiencia ha sido enriquecedora, y cualquiera de las dos partes la volvería a repetir, esperando que la próxima vez bajo mejores circunstancias.

## BIBLIOGRAFÍAS

Libro consultado:

La biblia MySQL, proporcionado por Javier Morillas zafra.

Java:

[https://www.google.es/urlsa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwjnbOQvP3MAhWByRoKHdLWDsEQFggcMAA&url=https%3A%2F%2Fes.wikipedia.org%2Fwiki%2FJava\\_\(lenguaje\\_de\\_programaci%25C3%25B3n\)&usg=AFQjCNHqaKaw4haLDXbuHzFL-Aj2BHUNfg](https://www.google.es/urlsa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwjnbOQvP3MAhWByRoKHdLWDsEQFggcMAA&url=https%3A%2F%2Fes.wikipedia.org%2Fwiki%2FJava_(lenguaje_de_programaci%25C3%25B3n)&usg=AFQjCNHqaKaw4haLDXbuHzFL-Aj2BHUNfg)

CSS:

[https://es.wikipedia.org/wiki/Hoja\\_de\\_estilos\\_en\\_cascada](https://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada)

HTML:

<https://es.wikipedia.org/wiki/HTML>

XML:

[https://es.wikipedia.org/wiki/Extensible\\_Markup\\_Language](https://es.wikipedia.org/wiki/Extensible_Markup_Language)

JAVASCRIPT Y AJAX:

<https://es.wikipedia.org/wiki/JavaScript>

<https://es.wikipedia.org/wiki/AJAX>

SQL:

<https://es.wikipedia.org/wiki/SQL>

ECLIPSE:

[https://www.google.es/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=0ahUKEwic9bvMvf3MAhXFORoKHZNBC4cQFggmMAE&url=https%3A%2F%2Fes.wikipedia.org%2Fwiki%2FEclipse\\_\(software\)&usg=AFQjCNF24tFu53y6bxflnVX34qGsTXsWRA](https://www.google.es/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=0ahUKEwic9bvMvf3MAhXFORoKHZNBC4cQFggmMAE&url=https%3A%2F%2Fes.wikipedia.org%2Fwiki%2FEclipse_(software)&usg=AFQjCNF24tFu53y6bxflnVX34qGsTXsWRA)

MYSQL WORKBENCH:

[https://es.wikipedia.org/wiki/MySQL\\_Workbench](https://es.wikipedia.org/wiki/MySQL_Workbench)

TOMCAT 7:

<https://es.wikipedia.org/wiki/Tomcat>

MAVEN:

[www.javiergarzas.com/2014/06/maven-en-10-min.html](http://www.javiergarzas.com/2014/06/maven-en-10-min.html)

HIBERNATE:

<https://es.wikipedia.org/wiki/Hibernate>

<https://blogdeaitor.wordpress.com/2012/09/07/uso-de-hibernate-con-oracle-y-mysql>

BOOTSTRAP:

[https://es.wikipedia.org/wiki/Twitter\\_Bootstrap](https://es.wikipedia.org/wiki/Twitter_Bootstrap)

JSF (JAVA SERVER FACES):

[https://es.wikipedia.org/wiki/JavaServer\\_Faces](https://es.wikipedia.org/wiki/JavaServer_Faces)

JQUERY:

<https://es.wikipedia.org/wiki/JQuery>

JAVAX MAIL

<https://es.wikipedia.org/wiki/JavaMail>

<http://www.oracle.com/technetwork/java/javamail/index.html>

FULLPAGE:

<http://webdesign.tutsplus.com/articles/vertical-and-horizontal-scrolling-with-fullpagejs--cms-24215>

JAVA.SECURITY:

[https://en.wikipedia.org/wiki/Java\\_security](https://en.wikipedia.org/wiki/Java_security)

JDBC (JAVA DATABASE CONNECTIVITY):

[https://es.wikipedia.org/wiki/Java\\_Database\\_Connectivity](https://es.wikipedia.org/wiki/Java_Database_Connectivity)

JAVAX SERVLET:

[https://es.wikipedia.org/wiki/Java\\_Servlet](https://es.wikipedia.org/wiki/Java_Servlet)

JDK 8 (JAVA DEVELOPMENT KIT V. 8):

[https://es.wikipedia.org/wiki/Java\\_Development\\_Kit](https://es.wikipedia.org/wiki/Java_Development_Kit)

J2SE (JRE, JAVA RUNTIME ENVIRONMENT):

[https://es.wikipedia.org/wiki/Java\\_SE](https://es.wikipedia.org/wiki/Java_SE)

JSP (JAVA SERVER PAGES):

[https://es.wikipedia.org/wiki/JavaServer\\_Pages](https://es.wikipedia.org/wiki/JavaServer_Pages)

GITHUB:

<https://es.wikipedia.org/wiki/GitHub>

**Enlaces de ayuda:**

<http://stackoverflow.com/>

<http://www.w3schools.com/>

<http://www.simplecodestuffs.com/sending-email-through-gmail-server/>

**FECHA TERMINACIÓN:**

Por terminar.

**CONTRAPORTADA:**

Proyecto final de curso del grado superior de desarrollo de aplicaciones web desarrollado por Sergio Lucena Fernández y Alejandro Oses Armentas basado en el desarrollo de una nueva página web para la administración de la FCT.

Consiste en un proyecto en el que su lenguaje de programación principal ha sido java .

Contiene una bonita y cómoda interfaz y diversas funcionalidades administrativas que se presentarán brevemente en estas páginas y podrán apreciarse mucho mejor al entrar en la aplicación.