

PFC: Selección personal



Universidad de La Coruña

Proyecto Fin de Carrera Ingeniería Técnica de Gestión Selección Personal

Autor: Antonio Pena Santiso
Tutor: Ramón Miñones Crespo

PFC: Selección personal

Índice de contenido

Resumen.....	5
Palabras clave.....	5
Hardware y software utilizado.....	5
Introducción.....	6
Objetivos.....	6
Descripción.....	6
Conceptos Previos.....	7
Sistemas de Gestión de Bases de Datos.....	7
SQL.....	8
MySQL.....	8
HTML.....	9
CSS.....	10
Java.....	10
Java Servlets.....	13
JSP.....	13
JavaScript.....	14
NetBeans.....	15
Estudio de alternativas existentes.....	16
Ejemplo de alternativa: El portal Infojobs.....	16
Planificación.....	19
Modelo de Desarrollo.....	19
Riesgos.....	20
Diagrama de Gantt.....	20
Evaluación de Costes.....	22
Horas Necesarias.....	22
Coste de personal.....	22
Análisis.....	23
Actores.....	23
Requisitos funcionales.....	24
Casos de Uso.....	28
Diagrama de Casos de Uso.....	28
Caso de Uso: Registro de usuario.....	29
Caso de Uso: Añadir Usuario.....	30
Caso de Uso: Inicio de sesión.....	31
Caso de Uso: Mostrar Ofertas.....	32
Caso de Uso: Buscar Oferta.....	32
Caso de Uso: Añadir Oferta.....	33
Caso de Uso: Inscribirse en oferta.....	33
Caso de Uso: Subir Currículo PDF.....	34
Caso de Uso: Eliminar Usuario.....	35
Caso de Uso: Modificar Estado de la Inscripción.....	36
Caso de Uso: Crear Cuestionario.....	37
Caso de Uso: Añadir Pregunta.....	38
Caso de Uso: Mostrar Respuestas Cuestionario.....	39
Caso de Uso: Añadir Comentario.....	40
Diseño.....	41

PFC: Selección personal

Diagrama de Clases.....	41
Diagramas de Secuencia.....	42
Iniciar Sesión.....	42
Registro de Usuario.....	43
Inscribirse en Oferta.....	44
Diagrama Entidad Relación.....	45
Arquitectura.....	46
El Patrón MVC.....	46
El Patrón Singleton.....	47
Pruebas.....	48
Pruebas de Persistencia.....	48
Pruebas de Integración.....	48
Pruebas sobre la Interfaz.....	48
Conclusiones.....	49
Mejoras de la aplicación.....	49
Impresiones.....	50
Bibliografía.....	51
Anexo 1: Manual del Usuario.....	52
Introducción.....	52
Requisitos.....	52
Estructura de ficheros.....	53
Preparar el servidor de Aplicaciones.....	56
Preparar Glassfish.....	56
Preparar Apache.....	57
Despliegue de la aplicación.....	58
Sentencias de creación de la base de datos.....	63
Guía.....	66
Inicio.....	66
Guía para usuarios.....	68
Registro.....	68
Ofertas.....	69
Intereses.....	69
Detalles de Oferta.....	70
Perfil.....	71
Guía para administradores.....	72
Ofertas.....	72
Nueva Oferta.....	73
Editar Cuestionario.....	74
Interesados.....	75
Usuarios.....	75
Editar Usuario.....	76

Índice de ilustraciones

Ilustración 1: Página Principal Infojobs.....	17
Ilustración 2: Ejemplo Oferta Infojobs.....	18
Ilustración 3: Tabla de Planificación.....	20
Ilustración 4: Diagrama de Gantt.....	21
Ilustración 5: Diagrama de Casos de Uso.....	28
Ilustración 6: Diagrama de Clases.....	41
Ilustración 7: Diagrama de Secuencia, Iniciar Sesión.....	42
Ilustración 8: Diagrama de secuencia, Registro de Usuario.....	43
Ilustración 9: Diagrama de Secuencia, Inscribirse en oferta.....	44
Illustration 10: Diagrama Entidad Relacion.....	45
Illustration 11: Patrón Singleton.....	47
Ilustración 12: Ficheros Java.....	53
Ilustración 13: Ficheros de la web.....	54
Ilustración 14: Despliegue de Aplicaciones Glassfish.....	58
Ilustración 15: Creación de Pool de conexiones.....	59
Ilustración 16: Creación de pool de conexiones 2.....	60
Ilustración 17: Creación de Recurso JDBC.....	61
Ilustración 18: Página Principal.....	66
Ilustración 19: Menú para Administradores.....	67
Ilustración 20: Menú para Usuarios registrados.....	67
Ilustración 21: Menú para Invitados.....	67
Ilustración 22: Registro de Usuarios.....	68
Ilustración 23: Listado de Ofertas.....	69
Ilustración 24: Listado de Ofertas Inscritas.....	69
Ilustración 25: Detalle Oferta.....	70
Ilustración 26: Perfil.....	71
Ilustración 27: Ofertas como Administrador.....	72
Ilustración 28: Nueva Oferta.....	73
Ilustración 29: Editar Cuestionario.....	74
Ilustración 30: Interesados como Administrador.....	75
Ilustración 31: Editar Usuario como Administrador.....	76

Resumen

El objetivo de este proyecto es el diseño y desarrollo de una aplicación para gestionar el proceso de reclutamiento y selección de personal en una empresa. El proyecto consiste en una aplicación web utilizando arquitectura MVC y tecnología JAVA EE. Se ha diseñado de forma general, con intención de que pudiera ser añadida a la web de una empresa cualquiera.

Los objetivos de este proyecto son:

1. Desarrollo de una base de datos que almacene información de los usuarios interesados en trabajar para la empresa e información de los puestos ofertados por la propia empresa.
2. Desarrollo de una web que sirva como interfaz de la base de datos y permita consultar y modificar sus datos.
3. Aumentar y desarrollar los conocimientos de la tecnología usada.

Palabras clave

JavaEE, MVC, JSP, MySQL, JDBC, Web, Junit, Netbeans, Glassfish

Hardware y software utilizado

- IDE: Netbeans 7.2.1
- Servidor Web: Apache Http Server 2.4
- Servidor de aplicaciones: GlassFish Server Open Source Edition 3.1.2.2
- Base de Datos: MySQL 5.5.29

Introducción

Objetivos

El principal objetivo a la hora de realizar este proyecto es la aplicación práctica de los conocimientos adquiridos a lo largo de la carrera, combinándolos para realizar un propósito concreto, en este caso una aplicación web.

Por un lado se busca experimentar el ciclo de desarrollo de una aplicación desde cero, y por otro es un intento de familiarizarse con Java EE y la programación web en general.

El proyecto intenta brindar una aplicación sencilla para empresas que no tengan medios para desarrollar un sistema a medida para organizar los datos de las plazas vacantes de que dispongan y para recibir currículos de candidatos a esas plazas.

Descripción

Este proyecto consiste en una aplicación web que permite a la empresa de cuya web forme parte publicar ofertas de puestos vacantes que quieran cubrir. Los usuarios que accedan a la web pueden ver información de estas ofertas, y pueden registrarse para depositar sus currículos en la base de datos e inscribirse como interesados para los puestos que les interesen. El usuario administrador, puede entonces acceder a los datos de los usuarios y gestionar las inscripciones en las ofertas de modo que el usuario pueda ver el estado de su solicitud (pendiente, descartado, etc).

La aplicación permite además distinguir a los usuarios que son empleados de la empresa de los que no, y permite publicar ofertas solo visibles para empleados, con el fin de reclutar dentro de la propia empresa. Además, se pueden crear cuestionarios con preguntas que el candidato que se inscriba en una oferta debe responder. Los usuarios administradores pueden después ver las respuestas que un usuario ha dado a estos tests.

El sistema permite además que los administradores escriban notas sobre los usuarios, visibles por otros administradores.

Conceptos Previos

Reclutamiento de Personal

El proceso de reclutamiento consiste en la creación de una reserva de candidatos para un puesto de trabajo disponible. La empresa anuncia al mercado la disponibilidad del puesto y trata de atraer candidatos cualificados interesados. La reserva de candidatos puede estar compuesta de empleados de la propia empresa que aspiren a un puesto distinto, candidatos externos o una mezcla de ambos.

El reclutamiento busca atraer candidatos cualificados. Atraer a candidatos que no cumplan con los requisitos supone un coste innecesario de tiempo y por tanto debe ofrecerse información clara de cuales son las características que un trabajador debe cumplir para ocupar un puesto. Para lograr los mejores resultados debe realizarse un análisis del puesto de trabajo previamente a anunciar que se encuentra vacante. Esto consiste en recopilar y organizar sistemáticamente información sobre el puesto para identificar las tareas y responsabilidades que un empleado tendrá en caso de ocupar ese puesto.

Un análisis del puesto bien hecho significa que la oferta de trabajo anunciada contendrá requisitos específicos sobre los estudios necesarios y las tareas a realizar y eso provocará que una parte de los candidatos se autoexcluyan al considerar que no cumplen los requisitos. De esta forma aumenta la adecuación de los candidatos al puesto solicitado.

Los portales de empleo online han facilitado en gran medida el proceso de reclutamiento para un gran número de empresas, sobretudo en lo que respecta a puestos administrativos o informáticos. Se puede reunir una reserva grande y diversa de candidatos de forma muy rápida, y los candidatos pueden utilizar filtros automáticos según sus preferencias para ver exclusivamente las ofertas que les interesen.

Selección de Personal

Habiendo reunido una reserva de candidatos durante el reclutamiento, comienza el proceso de selección. En esta fase se evalúa la calidad de las solicitudes del puesto y se realizan pruebas destinadas a separar a la persona que mejor se ajuste a las necesidades de la empresa de entre los candidatos posibles.

En el proceso de selección se incluyen la revisión de los currículos y referencias por parte del responsable de recursos humanos de la empresa, las entrevistas presenciales y pruebas como tests estandarizados de conocimientos, de personalidad o de otro tipo. El objetivo del proceso es no sólo descubrir si un candidato sería capaz de cumplir las tareas requeridas, si no también saber si se adaptará adecuadamente al entorno de la empresa, a sus compañeros, si tiene motivación para dar un rendimiento adecuado, etc.

Sistemas de Gestión de Bases de Datos

Los sistemas de gestión de bases de datos son un tipo de software que permite el acceso, manipulación y almacenamiento de datos en una base de datos. Estos sistemas manejan peticiones de usuarios o de otros programas para que estos no tengan que saber donde están los datos físicamente en los dispositivos de almacenamiento. Estos sistemas aseguran la integridad de los datos (que sean accesibles y estén debidamente organizados) y su seguridad (de modo que solo los usuarios con permiso tengan acceso a ellos).

Los SGBS más comunes son los sistemas relacionales, que organizan los datos en tablas definidas formalmente y que permiten que se accedan a sus datos y se ordenen de distintas formas según las consultas que se realicen sin tener que alterar la organización de las propias tablas. A pesar de que estos sistemas nunca han llegado a cumplir por completo los principios relacionales y de que tienen competencia de otras tecnologías (los sistemas orientados a objetos por ejemplo), la gran mayoría de bases de datos en uso en la actualidad utilizan este tipo organización.

SQL

El lenguaje de consulta estructurado o SQL es un lenguaje declarativo de acceso a bases de datos relacionales que permite realizar diversas operaciones sobre éstas. Actualmente es uno de los lenguajes de bases de datos más usado, tanto es así que SQL es el estándar que deben seguir los sistemas gestores de bases de datos relacionales. El lenguaje SQL esta gobernado por el estándar ANSI/ISO, que asegura la portabilidad del lenguaje a distintas plataformas. La última versión del estándar SQL es SQL:2011.



El lenguaje SQL consta de un lenguaje de definición de datos y un lenguaje de manipulación de datos. El primero es el utilizado para crear las estructuras de la base de datos e incluye sentencias como CREATE o DROP que permiten definir las tablas e índices que forman el sistema de la base de datos. El segundo es el que se usa para introducir, manipular y consultar los datos y consta de sentencias como SELECT, INSERT o UPDATE.

MySQL

MySQL es un sistema de gestión de bases de datos relacionales muy utilizado en la actualidad. Este sistema es la opción obvia para la mayoría de aplicaciones web, dado que es fácil de usar y ofrece soporte para las tecnologías más actuales, como Ajax, Ruby o PHP. Además al ser software libre, los costes para utilizarlo son bajos o nulos, por lo que es una opción popular para presupuestos reducidos, al contrario que otros sistemas como Oracle o SQLServer.



Además, debido a su sencillez, es un sistema muy rápido, sin características adicionales que ralenticen su funcionamiento, de modo que puede consultar un gran número de datos rápidamente, y en caso de que se requiera alguna funcionalidad adicional no incluida en MySQL básico, el hecho de que el sistema esté tan extendido significa que lo más probable es que alguien haya creado una distribución que incluya las funciones requeridas.

HTML

HTML son las siglas de HyperText Markup Language (Lenguaje de Marcas de Hipertexto), utilizado para la construcción de páginas web.

Su uso está destinado a describir la estructura y contenido de las sitios web en forma de texto, así como para complementar el texto con objetos tales como imágenes.



Este lenguaje se construye mediante etiquetas que delimitan las secciones de la web y alojan la información. Los navegadores web leen los ficheros HTML e interpretan las etiquetas para componer las páginas web visibles al usuario. En general hoy en día se utiliza junto con ficheros CSS en los que se especifica por separado la presentación de las páginas,

HTML puede además incluir scripts (JavaScript, JSP) que afectan al comportamiento de navegadores web y otros procesadores de HTML.

Puede ser creado y editado por cualquier editor de textos básico, como Gedit en Linux o Notepad en Windows, simplemente hay que tener en cuenta que la extensión del documento ha de ser *.htm o *.html.

En la actualidad, la versión más extendida es HTML 4.01, pero HTML5 se encuentra en desarrollo. Esta nueva versión del lenguaje pretende servir como nuevo estándar, sustituyendo tanto a HTML4 como a XHTML (una alternativa a HTML4 basada en la sintaxis de XML) y tiene como principal objetivo ofrecer soporte mejorado a funciones multimedia, de modo que pueda evitar depender de plugins externos. HTML5 ha sido diseñado además pensando en la compatibilidad con dispositivos móviles e incluye importantes mejoras para el uso de aplicaciones web.

En este proyecto se usa HTML4, ya que no se considera necesaria la nueva funcionalidad de HTML5 y muchos navegadores web antiguos que no son compatibles con las nuevas funciones siguen siendo usados extensivamente. No obstante, la adaptación de la aplicación a HTML5 es trivial, el nuevo lenguaje reconoce los elementos obsoletos de HTML4 y muchos nuevos elementos de HTML5 son interpretados como elementos similares de HTML4 cuando los navegadores no los soportan; así que aunque no se consideró necesario usar el nuevo lenguaje, en el futuro la aplicación podría adaptarse a él si se quisiera.

CSS

CSS (Cascade Style Sheets o Hojas de Estilo en Cascada) es un lenguaje utilizado para definir la presentación de un documento escrito en HTML o XML. CSS sirve para hacer independiente la estructura de una página web de su presentación, de modo que puedan modificarse sin que se afecten mutuamente.



Cuando se utiliza CSS, las etiquetas no deben proporcionar información acerca de la visualización del documento, sino marcar la estructura. La información del estilo se encuentra preferiblemente en un fichero independiente en el que se asocian formatos de letra, alineaciones o imágenes de fondo a las etiquetas que se usarán en el documento HTML.

Java

Java es una tecnología de programación que consta de un lenguaje de programación orientado a objetos y de la máquina virtual Java, que permite ejecutar los programas escritos en ese lenguaje. Es actualmente uno de los lenguajes de programación más utilizados, especialmente en lo que respecta a aplicaciones web.

Java fue creado a principios de los 90 por Sun Microsystems tomando gran parte de la sintaxis de C y C++ y eliminando herramientas de bajo nivel como la manipulación de memoria y punteros.



Java fue diseñado desde un principio para ser un lenguaje orientado a objetos. Al contrario que en la programación estructurada tradicional, en la que se examina el problema a resolver y se implementa una solución mediante una secuencia de funciones, en la programación orientada a objetos se considera que lo importante es identificar los objetos que participan en el problema.

PFC: Selección personal

Si por ejemplo programáramos un sistema para gestionar un restaurante, en programación estructurada identificaríamos las tareas a realizar: Cocinar un plato, servir una mesa, realizar el cobro, etc, y crearíamos los procedimientos para realizarlas. En orientación a objetos, en cambio, identificaríamos los objetos que forman el sistema: El cocinero, el camarero, el plato, la mesa, la factura, etc, y asignaríamos funcionalidades a cada objeto y estudiaríamos cómo se comunican esos objetos entre sí. Un camarero tendría la función de transportar un plato, que tendría una mesa asignada, etc.

La orientación a objetos cuenta con algunas características que la hacen mas eficiente que la programación estructurada clásica, fundamentalmente la herencia, el polimorfismo y el ocultamiento de información. En el ejemplo del restaurante, puede haber un objeto Plato básico, y después cada plato distinto (una sopa, una ensalada,etc) podría sobrescribir el Plato general con sus propias características. Esta es la llamada herencia, por medio de la cual se aprovecha un objeto ya existente para crear otro objeto nuevo similar con funcionalidad añadida. Por otro lado, el polimorfismo consistiría en que el camarero no necesitaría tener un método distinto para servir cada Plato, si no que tendría un único método “servirPlato” que haría cosas distintas según cada tipo de plato. Por último, el ocultamiento de información significa que una clase no tiene porqué saber la implementación de otra. De esta forma, para el Plato es indiferente cómo el camarero lo sirva, sólo que lo haga correctamente.

El código orientado a objetos se considera altamente reutilizable. En programación estándar, cada problema es un todo y es difícil extraer partes del sistema para usarlas en otros sistemas. Sería difícil aprovechar el código del restaurante para otro restaurante con distintos tipos de platos, mientras que con los objetos, puede usarse la clase Plato para derivar clases nuevas que se ajusten al nuevo problema. De esta forma, al reutilizar el código usado con éxito anteriormente, se acelera el desarrollo y se reducen los errores de programación.

Cuando se compila un programa Java, se crea un código intermediario llamado bytecode. Este código es interpretado por la máquina virtual para convertir el código Java en código que el sistema operativo sobre el que funcione la máquina virtual pueda entender, de modo que cualquier sistema operativo puede ejecutar un programa Java si hay una maquina virtual apropiada instalada.

PFC: Selección personal

Como el código se interpreta en cada ejecución, Java tiende a ser más lento que los lenguajes que generan código nativo al sistema operativo como C, sin embargo Java compensa este defecto con una versatilidad sin límites, siendo capaz de implementar cualquier tipo de función en cualquier arquitectura.

Otra característica importante del lenguaje Java es el recolector de basura. El recolector es un proceso que se ejecuta en la máquina virtual de Java, y que se encarga de liberar memoria asignada a objetos que sólo tienen referencias nulas. Esta significa que el programador no tiene que preocuparse de liberar manualmente la memoria que ya no necesita, como sí tendría que hacer en lenguajes como C.

Existen especificaciones distintas de Java, pensadas para adaptarse a varios rangos de necesidades:

Java Standard Edition(JavaSE): La edición básica de Java, actualmente en la versión 7.0. Incluye la especificación del lenguaje y de la máquina virtual de Java, y forma el núcleo de Java. En esta plataforma se incluye también la API JDBC para conexión con bases de datos.

Java Micro Edition(JavaME): Una plataforma de Java especialmente pensada para desarrollar aplicaciones para sistemas con escasos recursos, como teléfonos móviles. Aunque nuevas tecnologías móviles como Android han reducido su popularidad, sigue siendo usada ampliamente.

Java Enterprise Edition(JavaEE): Esta plataforma incluye todas las especificaciones de JavaSE y además un gran número de utilidades pensadas para el uso de redes y aplicaciones web. Esta edición está pensada para funcionar junto con un servidor de aplicaciones e incluye especificaciones para correo electrónico, XML, o servicios web.

Es esta última plataforma la que se utiliza en este proyecto y de las especificaciones adicionales que incluye son especialmente importantes aquí dos de ellas: Los Servlets de Java y JSP.

Java Servlets

Los servlets son programas de Java que se ejecutan en el lado del servidor y pueden recibir y responder a peticiones de clientes. Permiten conectar páginas con aplicaciones Java alojadas en un servidor.

Los servlets pueden realizar gran variedad de funciones. Pueden procesar información enviada a través de un formulario HTML, pueden sincronizar varios clientes ya que pueden manejar varias peticiones a la vez, o pueden redirigir peticiones a otras páginas o incluso a otros servlets.

En una aplicación web basada en el patrón MVC, los servlets constituyen la capa de Controlador. De esta forma, la capa de Vista con la que interactúa el usuario nunca tiene contacto con la lógica del programa, si no que son los servlets los que le proporcionan los datos que solicite.

JSP

Java Server Pages o JSP es una tecnología basada en Java que permite crear páginas web dinámicas. JSP permite introducir código Java en el mismo fichero que HTML, de modo que cuando un servidor de aplicaciones interpreta el fichero, utiliza las partes de JSP para generar servlets Java que crean la página final que se mostrará al usuario.



La tecnología JSP comparte todas las ventajas de Java. Es multiplataforma y permite separar claramente las capas de una aplicación web, dejando que los ficheros .jsp generen el código HTML pero pudiéndose comunicar fácilmente con el resto de la aplicación.

Las JSPs se comportan de forma idéntica a un servlet. En tiempo de ejecución, el código JSP es compilado como un servlet por el servidor de aplicaciones y permanece compilado hasta que se detecte que ha cambiado.

JavaScript

JavaScript es un lenguaje basado en objetos y guiado por eventos, utilizado para acceder a objetos en aplicaciones. Se utiliza integrado en un navegador web permitiendo el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas.

The logo for JavaScript, with 'Java' in blue and 'Script' in orange.

Es un lenguaje influenciado por múltiples lenguajes y fue diseñado con el objetivo de tener una sintaxis similar a la de Java, aunque más sencilla para facilitar el uso a programadores principiantes. Todos los navegadores actuales interpretan el código de JavaScript integrado dentro de las páginas web.

A pesar del nombre y de que JavaScript copia muchas convenciones de Java, los dos lenguajes no están relacionados. JavaScript es un lenguaje de tipado débil y dinámico en el que los objetos se basan en prototipos en vez de clases. Además, JavaScript no es cargado como bytecode por una máquina virtual, sino como código fuente legible.

En este proyecto, JavaScript se usa fundamentalmente para validar que los datos introducidos en los campos de los formularios sean correctos sin tener que enviar peticiones al servidor y cargar nuevamente las páginas web. Incluso si deben efectuarse comprobaciones en el lado del servidor de todas formas, cuando los errores se capturan en el cliente se libera carga del servidor.

NetBeans

NetBeans es un entorno de desarrollo integrado (en inglés, IDE) de código abierto especializado en Java. Fue creado por un estudiante checo en 1996 y pertenece en la actualidad a Oracle.



Un IDE es un entorno de programación empaquetado como una aplicación, que incluye un editor de código, un compilador, un depurador y un constructor de interfaces gráficas. Una IDE proporciona al programador un entorno de trabajo agradable que facilita el desarrollo de aplicaciones y reduce la posibilidad de errores.

NetBeans está diseñado para trabajar principalmente con Java, pero ofrece soporte para otros lenguajes como C, HTML o JavaScript. Actualmente se encuentra en la versión 7.3, si bien este proyecto ha utilizado la versión anterior 7.2.1.

La IDE ofrece gran cantidad de funcionalidades para apoyar el desarrollo de aplicaciones, algunas de las más útiles son:

- Compilación en tiempo real que permite detectar errores en el código sin requerir el proceso normal de compilación.
- Generación automática de código, útil para crear métodos get y set y similares funciones rutinarias.
- Organización en directorios del proyecto de forma sencilla.
- Funciones de refactorización automática para Java.
- Ejecución del programa con solo pulsar un botón

En este proyecto, NetBeans ha permitido realizar en un entorno común la programación del Modelo y el Controlador en Java, la programación de la vista en HTML, JSP, JavaScript y CSS, la creación y configuración de la base de datos y las funciones del servidor de aplicaciones Glassfish, que se encuentra integrado como parte de la propia IDE.

Estudio de alternativas existentes

Existen muchos portales de empleo en la web actualmente, como Infojobs o Tecnoempleo, que sirven para reunir empresas con trabajadores interesados. Muchas empresas tienen también sistemas integrados en sus páginas web para que aquellos que busquen trabajo en la empresa puedan hacer llegar sus datos a ésta. Este proyecto pretende crear una alternativa a estos sistemas integrados en la empresa que pueda servir para aquellos negocios que no quieran o no puedan crear un sistema a medida.

Existe software comercial como PeopleNet de Meta4, los módulos de selección de personal de A3 Software o la herramienta e-preselec de Infoempleo. Se comentarán también algunas de estas alternativas.

Ejemplo de alternativa: El portal Infojobs

Infojobs es un portal de empleo online español fundado en 1998 que y actualmente constituye la mayor bolsa de empleo de Europa. En este portal, las empresas pueden dejar información de plazas vacantes

y aquellos que busquen trabajo pueden aspirar a estas plazas aportando sus currículos y permitiendo si quieren que cualquier empresa acceda a ellos.

The logo for InfoJobs, featuring the word "InfoJobs" in a bold, dark blue, sans-serif font. The "Info" part is slightly smaller and positioned to the left of "Jobs".

Aunque esta es su función básica, Infojobs incluye también ayudas para la orientación laboral (anuncios de cursos, másters), soporte para trabajadores freelance gran cantidad de artículos y análisis de datos para apoyar el desarrollo de una carrera profesional.

Para este proyecto, se ha investigado la función principal del portal, las funciones de reclutamiento de candidatos.

PFC: Selección personal

Infojobs intenta hacer que encontrar un puesto que interese al candidato entre la enorme cantidad de ofertas sea sencillo. Cuenta con un sistema de búsqueda con gran cantidad de parámetros y opciones para filtrar dinámicamente los resultados obtenidos una vez buscados. Permite además recibir notificaciones automáticas de ofertas que se ajusten a unos parámetros que el usuario puede definir y guarda búsquedas anteriores para mostrar automáticamente nuevas ofertas que se ajusten a los intereses del usuario.

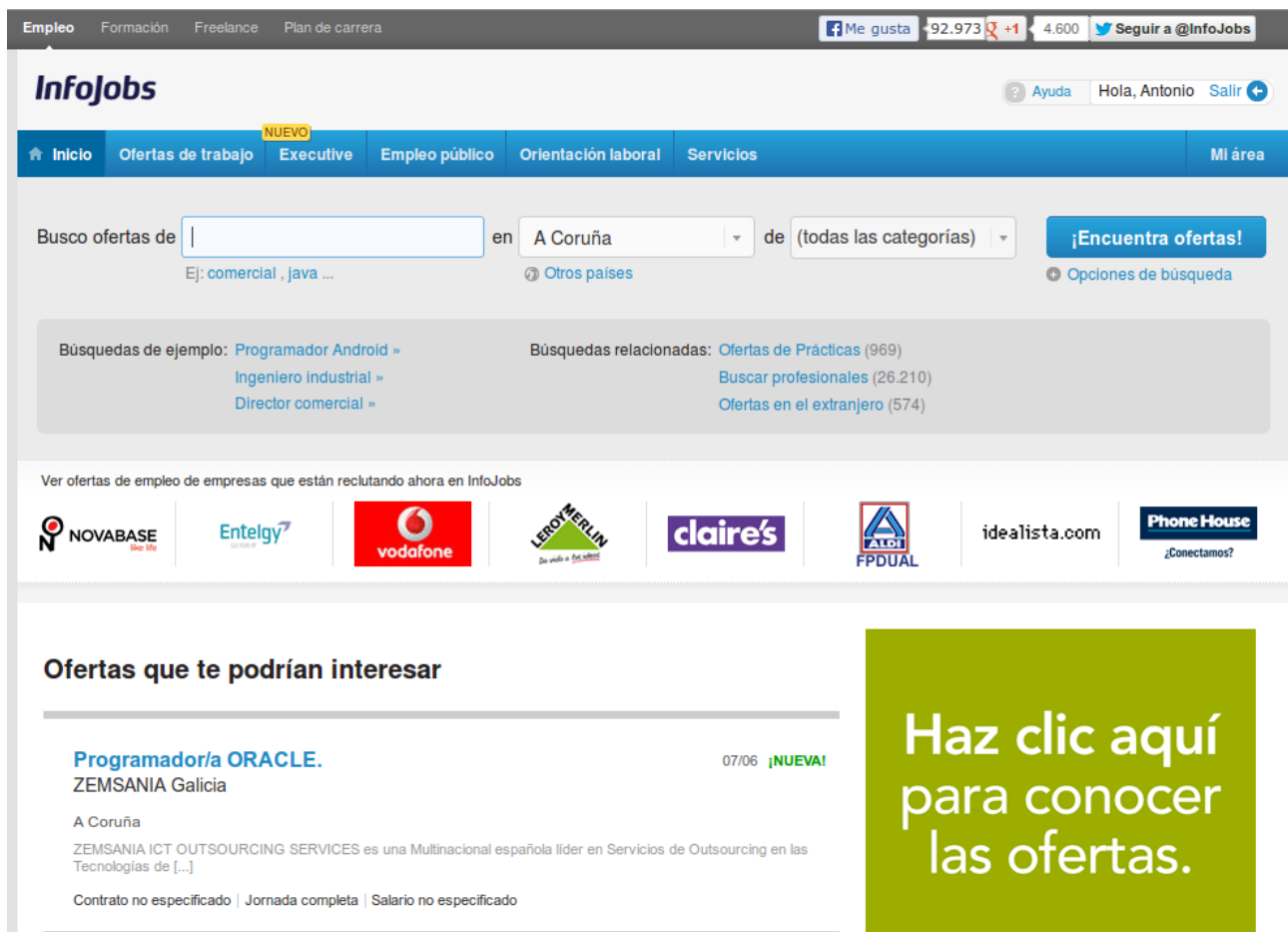


Ilustración 1: Página Principal Infojobs

Cuando un usuario se interesa por un puesto puede inscribirse como candidato en él. A partir de ese momento recibe notificaciones del estado de su aplicación, si la empresa ha leído su currículum y cuando lo ha hecho, si el proceso progresa o si es interrumpido y descartado.

PFC: Selección personal

En este proyecto se crea una aplicación similar a la herramienta de búsqueda de empleo de Infojobs, con la diferencia de que se pretende crear una aplicación integrada en la propia empresa. Se ha utilizado este portal como inspiración para tomar algunas decisiones con respecto a algunas funciones y a la interfaz del proyecto.

Requisitos

Estudios mínimos: Ingeniero Técnico - Técnico en Informática de Gestión

Experiencia mínima: Al menos 2 años

Conocimientos necesarios: Calidad del software Hibernate J2EE Java JEE5 JSF (JavaServer Faces) PMD SONAR Spring SQA

Imprescindible residente en: Provincia Puesto Vacante

Requisitos mínimos:

- Al menos 2-3 años en análisis, diseño y desarrollo de proyectos Java/J2EE/JEE5
- Al menos 2-3 años con análisis estático SONAR y/o PMD, pruebas de sistema, revisión de código fuente,...
- Capacidad de realizar análisis del diseño de una aplicación (modelo de datos, clases, etc).
- Lenguaje de modelado: UML.
- Herramientas de análisis estático: PMD, Sonar,...
- Base de Datos: Oracle, SQL,...

Requisitos deseados:

- Valorable conocimientos en Mainframe, DB2.
- Valorable certificación ISTQB.
- Valorable conocimientos y/o experiencia de herramientas de monitorización, pruebas de carga: Introscope, Dynatrace, LoadRunner,...
- Valorable conocimientos y/o experiencia en y frameworks SPRING/HIBERNATE/JSF.

Contrato

Tipo de contrato: Indefinido

Duración: INDEFINIDO

Jornada laboral: Completa

Así va el proceso

1 personas inscritas en esta oferta de empleo

La empresa estuvo gestionando CVs por última vez **hoy**

Nuestro consejo: no dudes en inscribirte necesariamente todos los otros inscritos

[Quiero inscribirme en la oferta](#)

☒ Recibe ofertas como ésta por e-mail | Comparte esta oferta: [f](#) [t](#) [e](#) [p](#)

Ilustración 2: Ejemplo Oferta Infojobs

PFC: Selección personal

Otras alternativas

People Net

People Net es la herramienta de gestión de personal de Meta4, incluye módulos variados para, por ejemplo, gestionar nóminas, trazar planes de carrera o evaluar el rendimiento de empleados. Incluye también un módulo de selección de personal, que sigue las tareas de publicación de vacantes y reclutamiento, seguimiento de la selección e integración posterior del nuevo empleado.

Abajo puede verse la lista de funciones con las que cuenta el módulo.



Además, Meta4 ofrece como parte del módulo un portal web llamado **Job Site** que permite a los usuarios ver información sobre las plazas vacantes, enviar sus currículos y rellenar plantillas corporativas con los datos que sean relevantes.

PFC: Selección personal

E-preselec

Esta herramienta ha sido creada por el portal infoempleo.com y está orientada fundamentalmente a realizar los procesos de selección de personal de la forma más ágil e intuitiva posible.



E-preselec trabaja junto a otros portales y aplicaciones de personal, para poder almacenar en una sola base de datos centralizada currículos recibidos de múltiples fuentes. Permite buscar rápidamente candidatos que se ajusten a un perfil, marcar su proceso de selección y monitorizar su progreso. La aplicación cuenta también con herramientas estadísticas que permiten hacer un seguimiento y comparación de todos los candidatos que tenga almacenados.

Este programa es además de fácil integración en páginas web de empresas y es compatible con las herramientas de gestión de personal usados actualmente (como la anteriormente mencionada PeopleNet) y con las grandes redes sociales como LinkedIn o Facebook. Funciona también con aplicaciones de tests psicotécnicos, pruebas de idiomas, etc.

Planificación

Modelo de Desarrollo

Durante el desarrollo del proyecto se ha intentado seguir el enfoque adecuado a una empresa de software. Importa no sólo el resultado final, si no también haber definido el proceso de forma clara y tener la documentación adecuada.

Se ha utilizado un desarrollo en cascada típico que ha constado de las siguientes etapas:

1. Definición y planificación del proyecto
2. Análisis
3. Diseño
4. Implementación
5. Pruebas
6. Documentación

Además durante la realización del proyecto y en paralelo a las etapas anteriormente descritas, se ha avanzado en el estudio de la arquitectura Java EE y el diseño web.

PFC: Selección personal

Riesgos

Se detectan los siguientes riesgos que pueden interferir con la planificación:

1. Falta de experiencia con las tecnologías usadas.
2. Cambios en los requisitos.
3. Imprevistos de carácter laboral.
4. Imprevistos de carácter familiar.

Diagrama de Gantt

Se ha realizado un diagrama de Gantt para mostrar la planificación del proyecto. Este diagrama ha sido realizado con Open Office Calc.

Nombre de la tarea	Fecha de Inicio	Duración
Inicio	05/11/12	0
Estudio Preliminar	05/11/12	7
Fin de la fase de Análisis	12/11/12	0
Diseño de la Web	12/11/12	14
Diseño de la Base de Datos	26/11/12	7
Fin de la fase de Diseño	03/12/12	0
Implementación de la Base de Datos	03/12/12	18
Implementación de la aplicación	21/12/12	80
Fin de la fase de Implementación	11/03/13	0
Memoria del proyecto	11/03/13	30
Entrega del proyecto	10/04/13	1

Ilustración 3: Tabla de Planificación

PFC: Selección personal

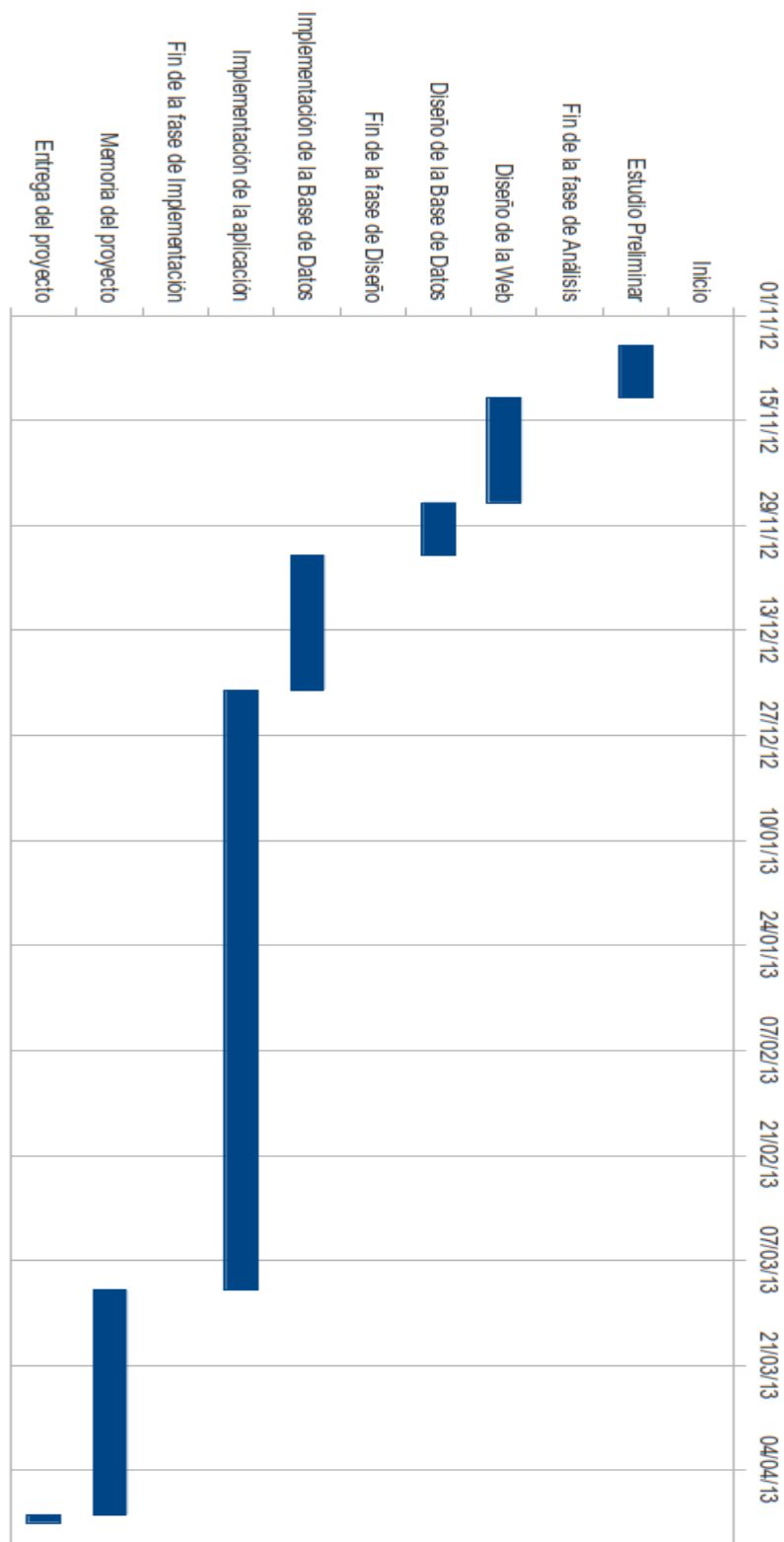


Ilustración 4: Diagrama de Gantt

PFC: Selección personal

Evaluación de Costes

Aquí estimamos el presupuesto del proyecto, examinando posibles gastos.

Horas Necesarias

Basándose en el anterior diagrama de Gantt, se pueden aproximar las horas necesarias, asumiendo 2 horas de trabajo cada día.

Análisis: 7 días x 2 horas/día = 14 horas

Diseño: 21 días x 2 horas/día = 42 horas

Implementación: 98 días x 2 horas/día = 196 horas

Memoria: 30 días x 2 horas/día = 60 horas

El proyecto se desarrolla por tanto en un total de 312 horas.

Coste de personal

Puesto	Horas	Coste(€)/Hora	Coste(€)
Analista	14	12	168
Diseñador	42	9	378
Programador	196	9	1764
Responsable Documentación	60	7	420
Total			2730

Se obtiene que los costes de las horas de trabajo ascienden a 2730€.

Análisis

A continuación se describe el análisis del problema propuesto y se describen los actores y casos de uso.

Actores

Los actores en Selección Personal son:

1. Invitado: Representa a un usuario cualquiera de Internet, que pueda acceder a la página para buscar en las ofertas o solo para curiosar. Accede sin registrarse ni figurar en la base de datos.

Este usuario puede acceder a la lista de ofertas, buscar según los criterios disponibles(Nombre del Puesto, Descripción del Puesto, Localización del Puesto) y ver toda la información si le interesa alguna oferta.

En todo momento, este usuario puede optar por registrarse usando la opción para ello visible en todo momento. De esta forma, entregando sus datos al sistema queda registrado y puede iniciar sesión como Candidato.

2. Candidato: Un usuario que se ha registrado en el sistema para poder hacer llegar sus datos a la empresa y poder inscribirse en las ofertas de trabajo que le interesen.

Puede completar o modificar sus datos y cargar su Currículo en la base de datos. Puede también ver el estado actual de sus inscripciones.

3. Administrador: Un usuario con permisos amplios, perteneciente a la empresa, que puede crear nuevas ofertas y cuestionarios y acceder a los datos subidos por los usuarios. Puede también modificar los datos de los usuarios, eliminar cuentas de usuario o modificar sus permisos. Puede además añadir comentarios a los perfiles de los usuarios.

Requisitos funcionales

1. **Control de Acceso:** Un invitado no necesita registrarse para utilizar las funciones que para él están disponibles. Un candidato y un administrador en cambio, sí deben estar registrados para acceder a determinadas partes de la aplicación.

Para acceder a esas funciones deben iniciar su sesión. Se considera la sesión iniciada cuando se introduce correctamente el email y la contraseña de un usuario registrado.

Un usuario puede iniciar la sesión desde cualquier parte de la aplicación y de igual manera puede cerrar su sesión en cualquier momento después de iniciada.

2. **Consultas:** Cualquier usuario puede consultar la tabla de Ofertas. Un usuario administrador puede consultar libremente la tabla de Usuarios, Cuestionarios, Preguntas y Comentarios.

3. **Mantenimiento** (Altas, bajas y modificaciones de tablas): Un Candidato da de alta una entrada en la tabla de Usuarios al registrarse, y puede modificar o dar de baja esa entrada en cualquier momento.

Un administrador puede realizar altas, bajas y modificaciones en toda la tabla de Usuarios y también en la tabla de Ofertas.

Puede también crear el cuestionario de una oferta y preguntas para rellenarlo, que se guardan en las tablas Cuestionarios y Preguntas respectivamente.

Un administrador puede además crear, modificar y eliminar comentarios de la tabla Comentarios.

PFC: Selección personal

El contenido de estas tablas es el siguiente:

Tabla Usuarios:

<i>Atributo</i>	<i>Tipo</i>	<i>Descripción</i>
Nombre	String, Obligatorio	El nombre de pila del usuario.
Apellidos	String, Obligatorio	Los apellidos del usuario.
Email	String, Obligatorio	El correo electrónico del usuario.
Contraseña	String, Obligatorio	La contraseña para iniciar la sesión del usuario.
Teléfono	String, Obligatorio	El teléfono de contacto del usuario.
Tipo	String, Obligatorio	Indica los permisos del usuario. Modificable por un administrador
Currículo	Texto, Opcional	El currículum en texto del usuario.
Foto	Bytes, Opcional	Un fichero con la foto del usuario.
CVFichero	Bytes, Opcional	Un fichero con el currículum del usuario.

Tabla Ofertas:

<i>Atributo</i>	<i>Tipo</i>	<i>Descripción</i>
ID	Entero, Obligatorio	Clave de la oferta.
Fecha	Fecha, Obligatorio	Fecha de publicación de la oferta
Puesto	String, Obligatorio	Nombre del puesto que se oferta.
Descripción	String, Obligatorio	Descripción del puesto que se oferta.
Destinatarios	String, Obligatorio	Indica qué usuarios tienen permiso para ver la oferta.
Ubicación	String, Obligatorio	El teléfono de contacto del usuario.
Sueldo	String, Opcional	El sueldo ofrecido.
Estudios	String, Opcional	El nivel de estudios requerido para el puesto.
Experiencia	String, Opcional	La experiencia laboral previa requerida.
Requisitos	String, Opcional	Otros requisitos que pueda tener el puesto.
Duración	String, Opcional	Duración del contrato.
Tipo_Contrato	String, Opcional	Indica si el contrato es fijo, temporal, en prácticas, etc.

PFC: Selección personal

Tabla Cuestionarios:

<i>Atributo</i>	<i>Tipo</i>	<i>Descripción</i>
ID	Entero, Obligatorio	Clave del cuestionario
Nombre	String, Obligatorio	Nombre dado al cuestionario.
Oferta	Entero, Obligatorio	Clave de la oferta asociada.

Tabla Preguntas:

<i>Atributo</i>	<i>Tipo</i>	<i>Descripción</i>
ID	Entero, Obligatorio	Clave de la pregunta.
Enunciado	Texto, Opcional	El contenido de la pregunta.
Cuestionario	Entero, Obligatorio	Clave del cuestionario al que pertenece.

Tabla Comentarios:

<i>Atributo</i>	<i>Tipo</i>	<i>Descripción</i>
ID	Entero, Obligatorio	Clave del comentario
Fecha	Date, Obligatorio	Fecha en la que se hizo el comentario.
Usuario	String, Obligatorio	Email del usuario cuyo perfil se comenta.
Autor	String, Obligatorio	Email del administrador que escribió el comentario.
Texto	Texto, Opcional	Contenido del comentario.

PFC: Selección personal

4. **Asignaciones:** Un usuario candidato puede inscribirse en una oferta y de esta forma asociar su cuenta con la oferta en cuestión creando una entrada en la tabla de Interesados. Puede así mismo eliminar esa entrada posteriormente. Si la oferta tiene un cuestionario, entonces el usuario debe contestar las preguntas y creará una entrada en la tabla Respuestas para cada una. Un usuario administrador puede modificar el campo estado de las inscripciones.

Estas dos tablas constan de lo siguiente:

1. Interesados:

<i>Atributo</i>	<i>Tipo</i>	<i>Descripción</i>
Email	String, Obligatorio	Email que identifica al usuario.
Oferta	Entero, Obligatorio	La clave que identifica la oferta.
Estado	String, Obligatorio	El estado actual de esta inscripción.
Fecha	Date, Obligatorio	Fecha en la que se realizó la inscripción.

2. Respuestas:

<i>Atributo</i>	<i>Tipo</i>	<i>Descripción</i>
Usuario	String, Obligatorio	Email que identifica al usuario.
Pregunta	Entero, Obligatorio	La clave que identifica la pregunta.
Contenido	Texto, Obligatorio	La respuesta introducida por el usuario.

Requisitos no funcionales

1. Cumplimiento de Estándares: Se debe intentar que la aplicación cumpla con los estándares para que la visualización en los distintos navegadores sea correcta.
Deben poder usarse Internet Explorer, Google Chrome y Firefox. Puede haber diferencias de visualización pero ello no debe suponer un funcionamiento incorrecto de la aplicación.
2. Seguridad: La base de datos estará protegida con contraseña para que sólo pueda acceder a ella el administrador de la aplicación.
3. Respuesta a errores: Los errores que pueda cometer el usuario serán evitados en el cliente, no permitiendo campos vacíos o datos de tipo erróneo. Se informará al usuario correctamente de los errores y se evitará en lo posible requerir que se repita la introducción de datos.
4. Eficiencia y facilidad: La aplicación web debe estar bien estructurada, que la navegación sea clara e intuitiva, ofreciendo al usuario lo que necesite. La aplicación desde un principio ha de ofrecer al usuario lo que busca, como poder efectuar el login una vez accede a la aplicación. Una vez dentro de la aplicación, todas las funciones que nos ofrece deben ser fáciles de utilizar. A la vez todo debe estar bien estructurado y bien indicado para hacer la aplicación lo más amigable posible al usuario que la utilice.

PFC: Selección personal

Casos de Uso

Diagrama de Casos de Uso

Se identifican los siguientes casos de uso para el proyecto.

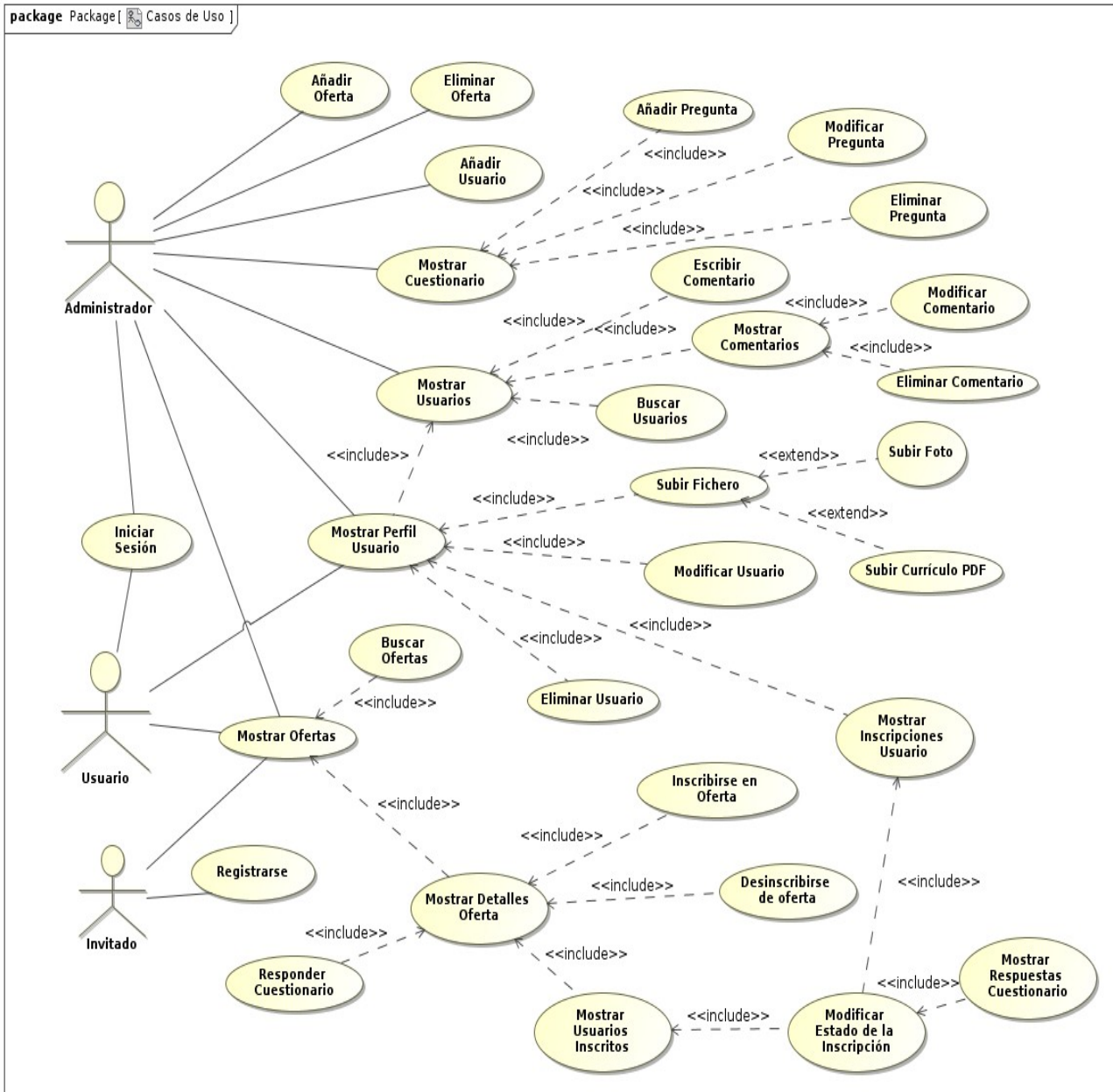


Ilustración 5: Diagrama de Casos de Uso

PFC: Selección personal

Caso de Uso: Registro de usuario

Actores: Invitado

Precondiciones: Ninguna

Descripción: Registra al usuario en el sistema, añadiendo sus datos a la base de datos y permitiéndole iniciar sesiones después.

Secuencia:

1. El Usuario solicita iniciar el registro.
2. El sistema muestra la interfaz de registro de usuarios.
3. El Usuario introduce sus datos y los envía.
4. El sistema introduce los datos en una nueva entrada en la base de datos identificada por el email introducido.
5. El sistema inicia la sesión con el nuevo usuario y muestra la página principal.

Excepciones:

- Paso 3: El usuario introduce incorrectamente algún dato
 - El sistema indica el error y continúa en la interfaz de registro.
- Paso 3: Ya existe un usuario asociado a la dirección de email
 - El sistema informa del error después de comprobar la base de datos y vuelve a la interfaz de registro.

Postcondiciones: El Usuario queda registrado en la base de datos y con sesión iniciada.

PFC: Selección personal

Caso de Uso: Añadir Usuario

Actores: Administrador

Precondiciones: Haber iniciado la sesión como administrador.

Descripción: Similar al caso Registro de usuario, pero en este caso es un administrador quien registra a un usuario y puede asignarle distintos niveles de acceso.

Secuencia:

1. El Administrador accede a la sección de usuarios y escoge registrar un nuevo usuario.
2. El sistema muestra la interfaz de registro de usuarios.
3. El administrador introduce los datos y permisos del nuevo usuario y los envía.
4. El sistema introduce los datos en una nueva entrada en la base de datos identificada por el email introducido.
5. El sistema regresa a la sección de usuarios.

Excepciones:

- Paso 3: El administrador introduce incorrectamente algún dato
 - El sistema indica el error y continúa en la interfaz de registro.
- Paso 3: Ya existe un usuario asociado a la dirección de email
 - El sistema informa del error después de comprobar la base de datos y vuelve a la interfaz de registro.

Postcondiciones: El nuevo usuario queda registrado en la base de datos.

PFC: Selección personal

Caso de Uso: Inicio de sesión

Actores: Usuario, Administrador

Precondiciones: Haber realizado con anterioridad un registro de usuario. Que no haya una sesión iniciada en este momento.

Descripción: Inicia una sesión del usuario para poder acceder a las funciones que lo requieran.

Secuencia:

1. El Usuario introduce su email y su contraseña en el formulario de logueo y acepta.
2. El sistema comprueba que el usuario exista y la contraseña sea correcta.
3. El sistema inicia la sesión y devuelve a la página principal.

Excepciones:

- Paso 2: El email introducido no está en la base de datos
 - El sistema muestra un mensaje de error.
- Paso 2: La contraseña introducida es incorrecta.
 - El sistema muestra un mensaje de error.

Postcondiciones: Queda iniciada la sesión del usuario.

PFC: Selección personal

Caso de Uso: Mostrar Ofertas

Actores: Invitado, Usuario, Administrador

Precondiciones: Ninguna

Descripción: Permite ver una lista de los puestos ofertados.

Secuencia:

1. El usuario accede a la sección de ofertas de la aplicación.
2. El sistema muestra la lista de ofertas disponibles.

Postcondiciones: Ninguna.

Caso de Uso: Buscar Oferta

Actores: Invitado, Usuario, Administradores

Precondiciones: Ninguna,

Descripción: Permite buscar ofertas que se ajusten a los criterios pedidos.

Secuencia:

1. El usuario accede a la sección de ofertas de la aplicación.
2. El usuario rellena los campos de búsqueda que le interesan (Puesto, Descripción, País).
3. El sistema devuelve una lista de ofertas cuyos campos incluyen lo introducido en la búsqueda.

Flujos alternativos:

- Paso 3: El sistema no encuentra ofertas que coincidan con los parámetros de búsqueda y muestra un mensaje.

Postcondiciones: Ninguna

PFC: Selección personal

Caso de Uso: Añadir Oferta

Actores: Administrador

Precondiciones: Haber iniciado sesión.

Descripción: Permite al administrador añadir una oferta a la base de datos.

Secuencia:

1. El administrador accede a la sección de ofertas y escoge nueva oferta.
2. El sistema muestra la interfaz de creación de ofertas.
3. El administrador rellena los datos del formulario y los envía.
4. El sistema crea una nueva entrada en la base de datos y le asigna un identificador entero.
5. El sistema devuelve al administrador a la pagina principal de ofertas

Postcondiciones: La nueva oferta queda registrada en la base de datos.

Caso de Uso: Inscribirse en oferta

Actores: Usuario

Precondiciones: Haber iniciado sesión. El usuario no puede haberse escrito con anterioridad en esa oferta.

Descripción: Permite al usuario inscribirse en el registro de interesados en una oferta.

Secuencia:

1. El usuario accede a la zona de ofertas.
2. El usuario accede a los detalles de una de las ofertas.
3. El usuario selecciona la opción de inscribirse.
4. El usuario responde las preguntas del cuestionario de haberlo.
5. El sistema añade una entrada en la base de datos que relaciona el usuario con la oferta escogida.
6. El sistema devuelve al usuario a la página de la oferta.

Postcondiciones: El usuario interesado queda registrado en la base de datos.

PFC: Selección personal

Caso de Uso: Subir Currículo PDF

Actores: Usuario

Precondiciones: Haber iniciado sesión.

Descripción: Permite al usuario subir a la base de datos un fichero en PDF de su currículo.

Secuencia:

3. El Usuario accede a su perfil de usuario.
4. El sistema muestra los datos del usuario.
5. El Usuario escribe el path del fichero en el espacio correspondiente o explora el sistema de ficheros para seleccionarlo, y acto seguido sube el fichero.
6. El sistema almacena el contenido del fichero en la base de datos.
7. El sistema vuelve a la página de perfil.

Excepciones:

- Paso 4: El sistema detecta que el fichero subido no es del tipo correcto.
 - El sistema muestra un mensaje de error y vuelve a la página del perfil.

Postcondiciones: El fichero queda subido en la base de datos.

PFC: Selección personal

Caso de Uso: Eliminar Usuario

Actores: Administrador

Precondiciones: Haber iniciado sesión como administrador. Debe existir un usuario registrado.

Descripción: Permite al administrador eliminar a un usuario de la base de datos.

Secuencia:

1. El administrador accede a la sección de usuarios.
2. El administrador accede a los detalles de uno de los usuarios.
3. El administrador escoge la opción de eliminar el usuario.
4. El sistema elimina el usuario de la base de datos.
5. El sistema vuelve a la sección de usuarios.

Postcondiciones: El usuario queda borrado de la base de datos y también las inscripciones que haya podido hacer en ofertas.

PFC: Selección personal

Caso de Uso: Modificar Estado de la Inscripción

Actores: Administrador

Precondiciones: Haber iniciado sesión como administrador. Debe existir un usuario y una oferta, y el usuario ha de haberse inscrito en esa oferta.

Descripción: Permite al administrador modificar el estado del proceso de selección de un usuario.

Secuencia:

1. El administrador accede a la sección de ofertas.
2. El administrador accede a los detalles de una de las ofertas.
3. El administrador elige ver la lista de usuarios inscritos en esa oferta.
4. El sistema muestra el estado actual de los usuarios respecto a la oferta
5. El administrador escoge el nuevo estado de la inscripción y confirma.
6. Se actualiza el estado en la base de datos.
7. El sistema muestra la lista de usuarios inscritos en la oferta con el estado actualizado.

Postcondiciones: El estado de la inscripción queda modificado.

PFC: Selección personal

Caso de Uso: Crear Cuestionario

Actores: Administrador

Precondiciones: Haber iniciado sesión como administrador. Debe existir una oferta.

Descripción: Permite al administrador crear un cuestionario de preguntas para la oferta.

Secuencia:

1. El administrador accede a la sección de ofertas.
2. El administrador accede a los detalles de una de las ofertas.
3. El administrador elige editar el cuestionario de la oferta.
4. El administrador puede introducir un nombre para el cuestionario.
5. El administrador puede añadir nuevas preguntas(**Caso Añadir Pregunta**)
6. El cuestionario se guarda en el sistema.

Postcondiciones: El cuestionario queda creado.

PFC: Selección personal

Caso de Uso: Añadir Pregunta

Actores: Administrador

Precondiciones: Haber iniciado sesión como administrador. Debe existir una oferta y debe tener un cuestionario.

Descripción: Permite al administrador añadir preguntas al cuestionario asignado a una oferta.

Secuencia:

1. El administrador accede a la sección de ofertas.
2. El administrador accede a los detalles de una de las ofertas.
3. El administrador elige editar el cuestionario de la oferta.
4. El administrador escribe el enunciado de la pregunta y escoge añadirla.
5. La pregunta queda asignada al cuestionario y se almacena en el sistema.
6. El sistema muestra la página del cuestionario.

Postcondiciones: La pregunta queda añadida al cuestionario.

PFC: Selección personal

Caso de Uso: Mostrar Respuestas Cuestionario

Actores: Administrador

Precondiciones: Haber iniciado sesión como administrador. Debe existir una oferta y debe tener un cuestionario. Un usuario debe haberse inscrito en la oferta y respondido al cuestionario.

Descripción: Permite al administrado ver lo que ha respondido un usuario en un cuestionario.

Secuencia:

1. El administrador accede a la sección de ofertas.
2. El administrador accede a los detalles de una de las ofertas.
3. El administrador elige mostrar los usuarios inscritos.
4. El administrador elige ver las respuestas de uno de los usuarios.
5. El sistema muestra el cuestionario de la oferta con las respuestas que el usuario introdujo anteriormente.

Postcondiciones: Ninguna.

PFC: Selección personal

Caso de Uso: Añadir Comentario

Actores: Administrador

Precondiciones: Haber iniciado sesión como administrador. Debe existir un usuario registrado.

Descripción: Permite al administrador escribir un comentario en el perfil de un usuario que será visible a otros administradores.

Secuencia:

1. El administrador accede a la sección de usuarios.
2. El administrador accede al perfil de un usuario.
3. El administrador escribe un comentario en el cuadro adecuado y lo envía.
4. El sistema guarda el comentario.
5. El sistema muestra la pantalla de comentarios de ese usuario, con todos los comentarios que se han hecho hasta ese momento.

Postcondiciones: El comentario queda almacenado en el sistema.

Diseño

Acabado el análisis, se procede a la fase de diseño del proyecto, en la que se presentan los diagramas de clase y de secuencia que definen la aplicación. Se expone también el diseño de la base de datos y se explica la arquitectura empleada.

Diagrama de Clases

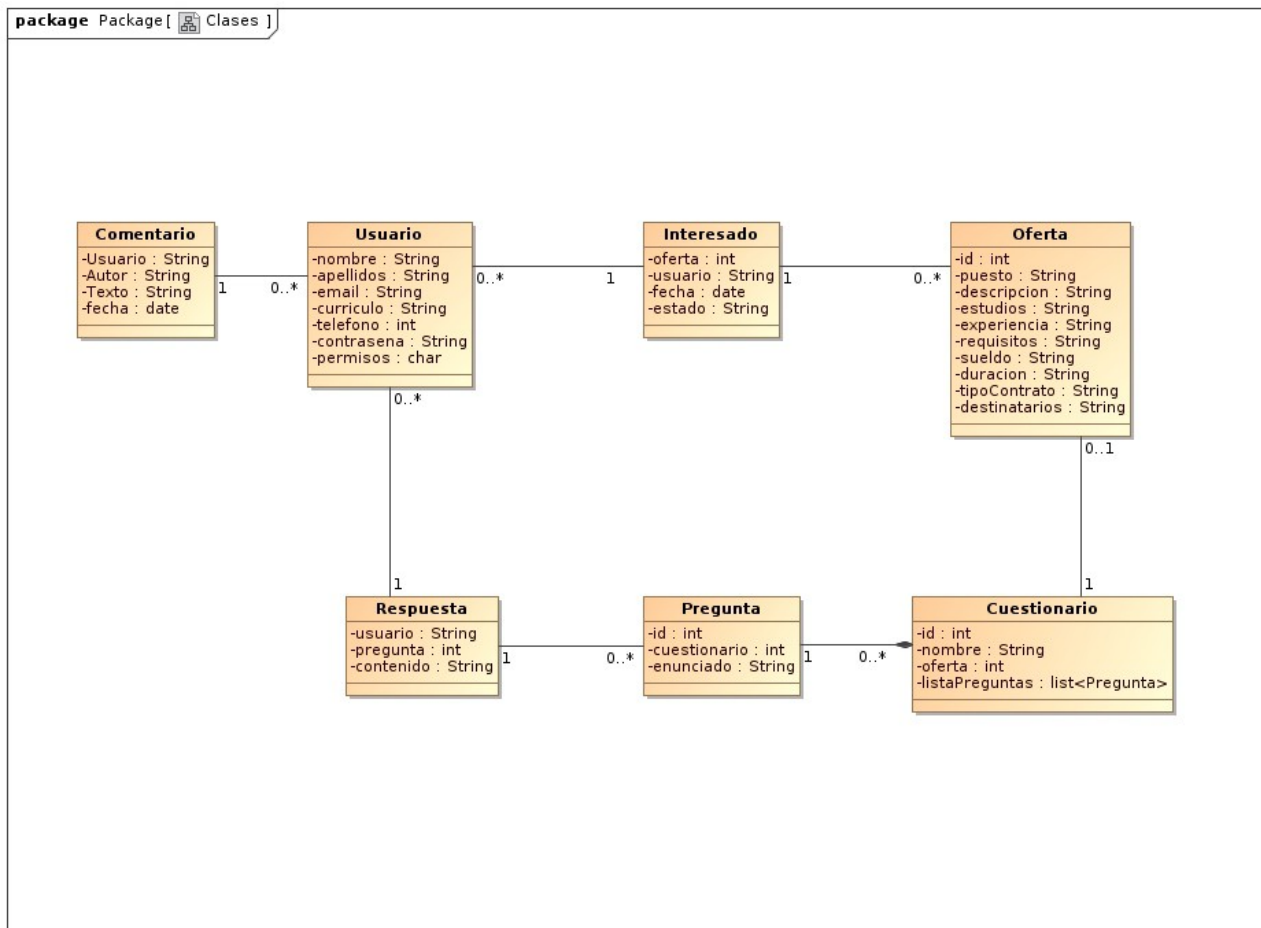


Ilustración 6: Diagrama de Clases

PFC: Selección personal

Diagramas de Secuencia

A continuación se ofrecen diagramas de secuencia para algunas de los procesos más importantes.

Iniciar Sesión

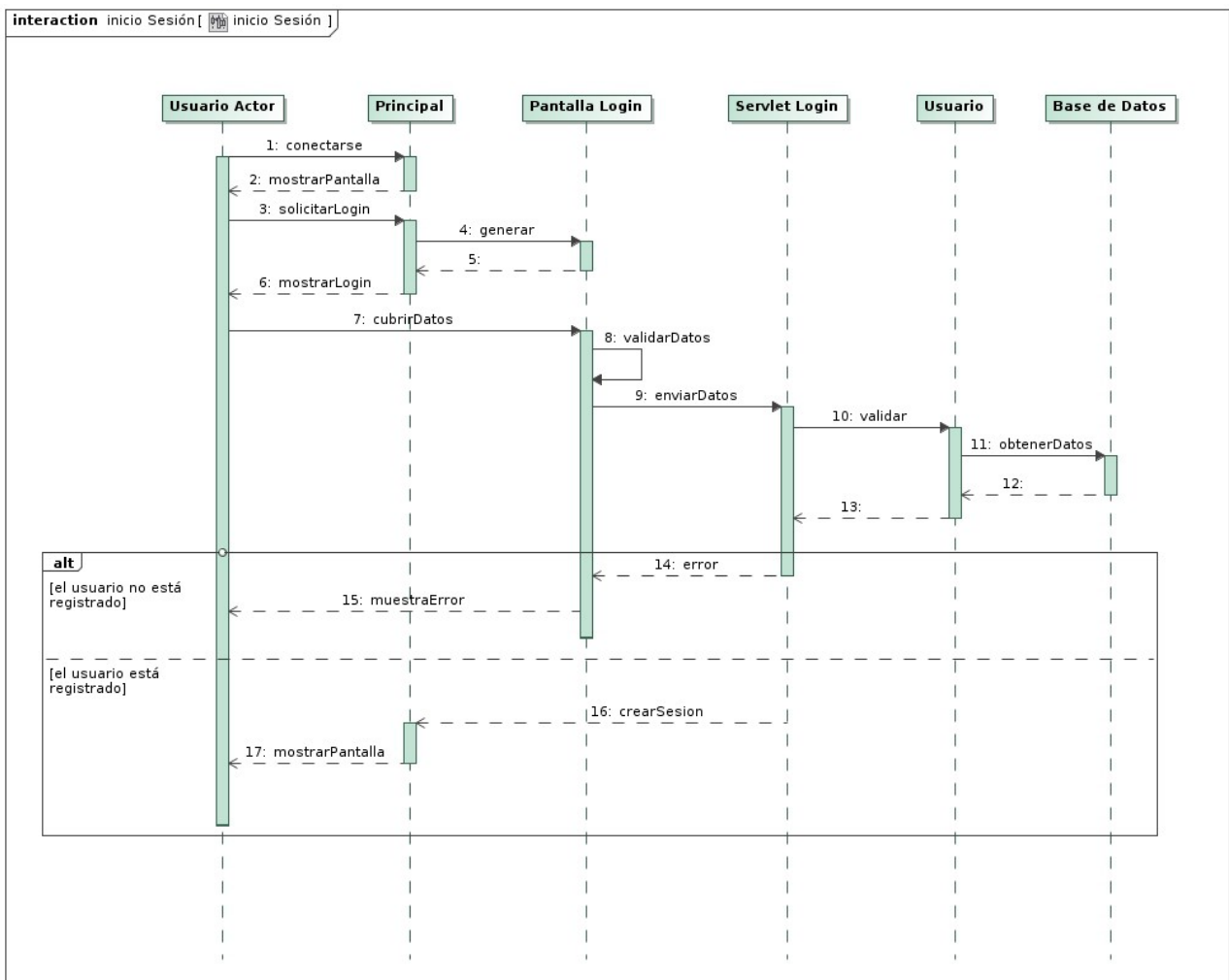


Ilustración 7: Diagrama de Secuencia, Iniciar Sesión

PFC: Selección personal

Registro de Usuario

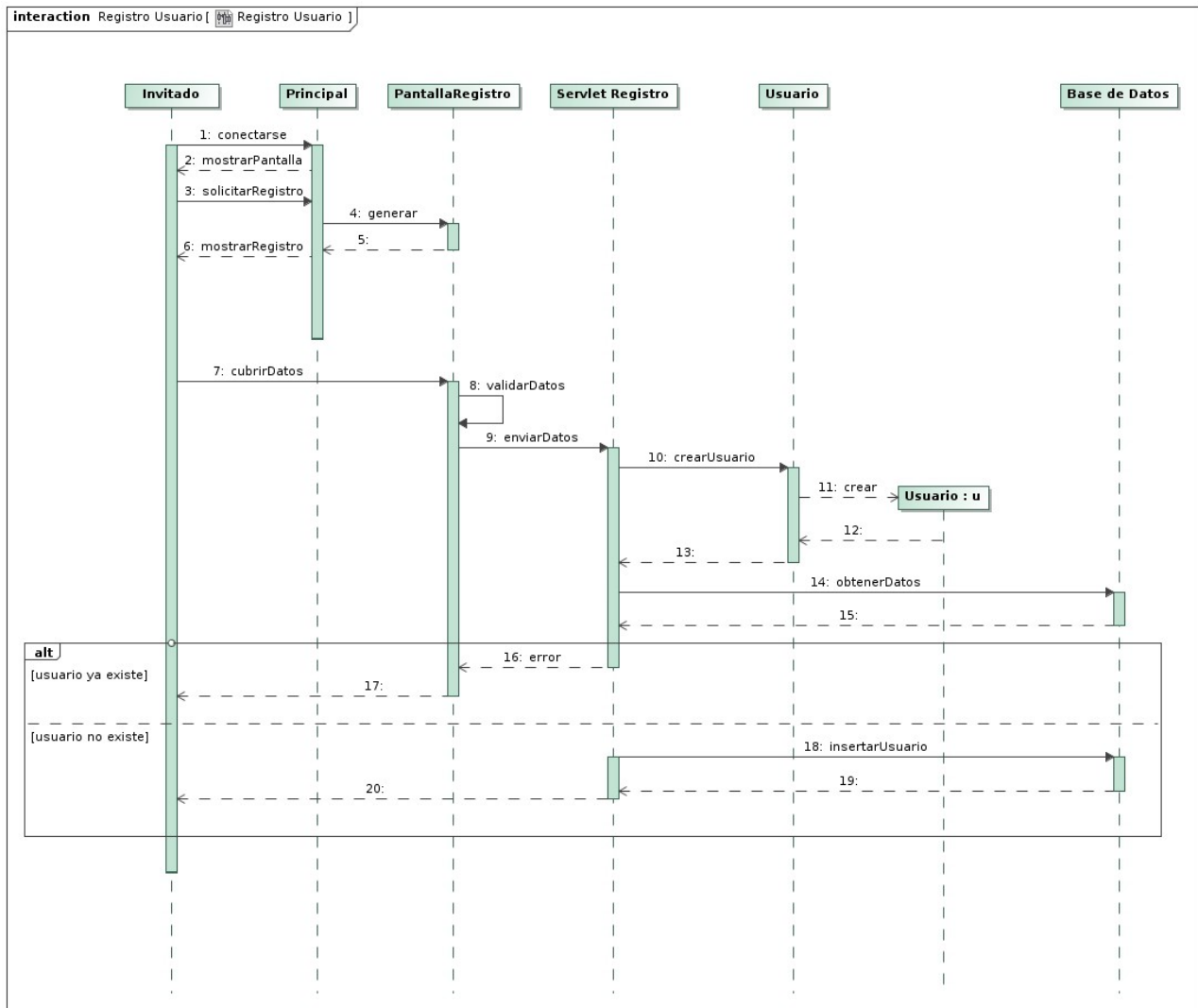


Ilustración 8: Diagrama de secuencia, Registro de Usuario

PFC: Selección personal

Inscribirse en Oferta

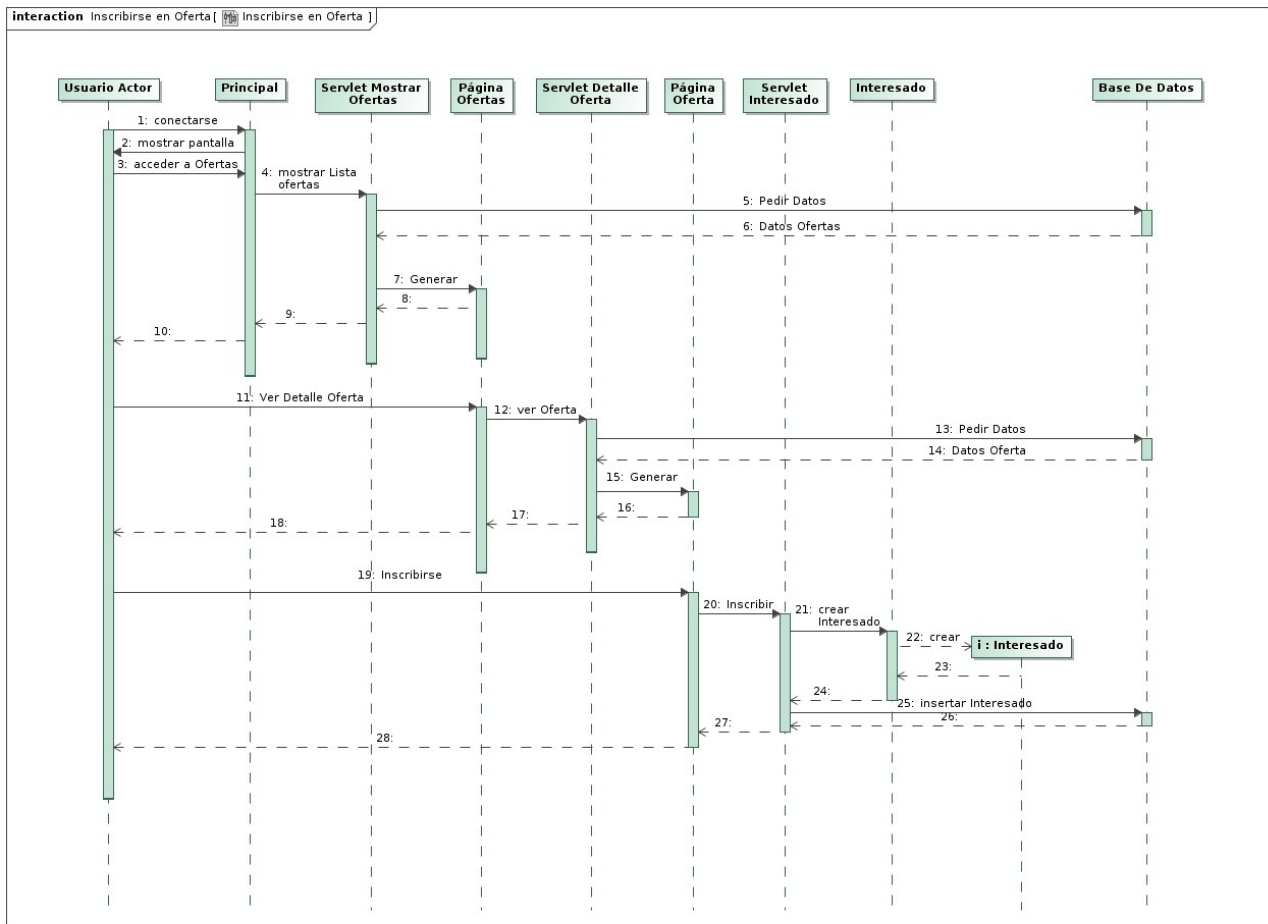


Ilustración 9: Diagrama de Secuencia, Inscribirse en oferta

PFC: Selección personal

Diagrama Entidad Relación

Aquí se presenta un esquema de la estructura de la base de datos creada:

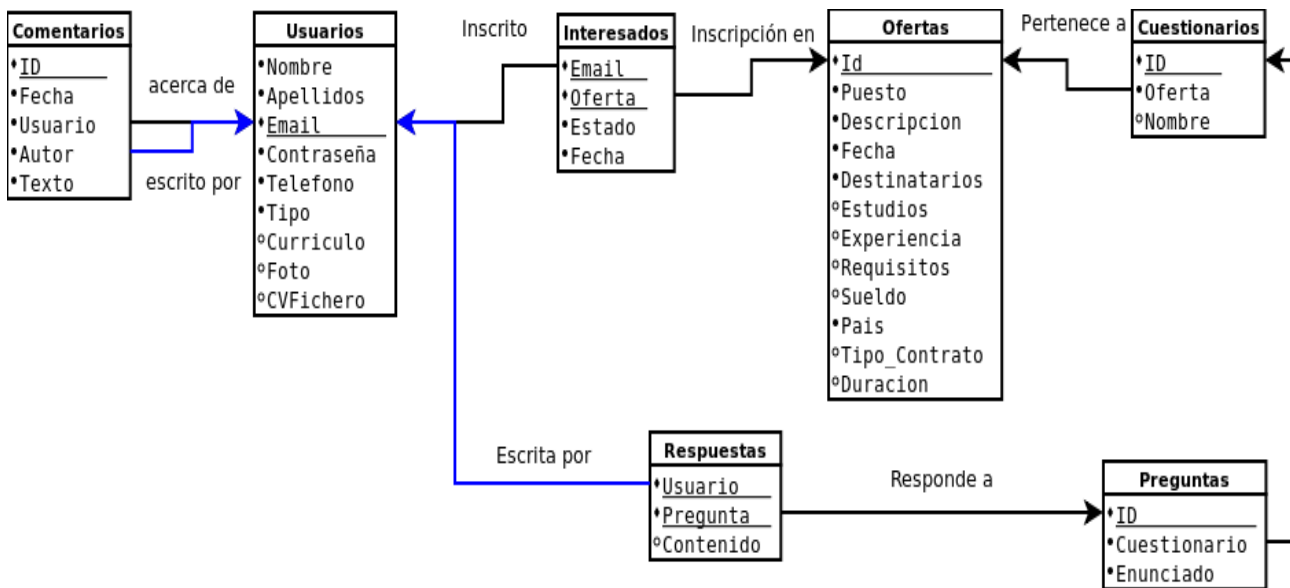


Illustration 10: Diagrama Entidad Relación

Arquitectura

El Patrón MVC

La aplicación está estructurada según el patrón Modelo-Vista-Controlador, que separa la interfaz del usuario de la lógica del programa. Según esto, las capas de la aplicación son las siguientes:

Cliente: Se conecta a Internet y accede a la aplicación utilizando un navegador. En ésta aplicación, el cliente es ligero, y se limita a enviar peticiones por medio de formularios y enlaces al servidor, donde está situada la lógica del programa.

Vista: La interfaz que se muestra al usuario. Está implementada en ficheros de Java Server Pages, que permiten crear páginas web dinámicas. Esta capa se limita a mostrar de forma adecuada los datos que recibe del Controlador, y no contiene nada de la lógica del programa.

Controlador: Está implementado como una serie de Servlets Java que responden a peticiones que el usuario haga sobre la Vista e invocan al Modelo.

Modelo: Contiene la lógica de la aplicación, recibe las peticiones del Controlador. Está compuesto de Beans de Java.

Persistencia: Capa que implementa la conexión con la base de datos.

Este tipo de arquitectura dividida en capas ofrece una serie de ventajas:

1. Ofrece facilidad de mantenimiento de la aplicación, porque puede modificarse una de las capas sin tener que alterar el resto.
2. El modelo puede ser reutilizado fácilmente al estar aislado del resto del programa.
3. El sistema es fácilmente escalable, puede ampliarse, reducirse o modificarse.

PFC: Selección personal

4. Los usuarios pueden utilizar la aplicación independientemente del sistema operativo que utilicen y en caso de ser actualizado no tienen que descargar software adicional.

El Patrón Singleton

Este patrón permite guardar instancias únicas de clases que permanecerán inalteradas en las distintas ejecuciones de la aplicación.

Una clase definida según el patrón Singleton contiene como atributo una instancia de la propia clase. Otras clases pueden obtener acceso a esta instancia llamando a un método estático `getInstancia`, que si y sólo si no hay ninguna instancia creada llama al constructor privado de la clase y de lo contrario devuelve la instancia ya existente.

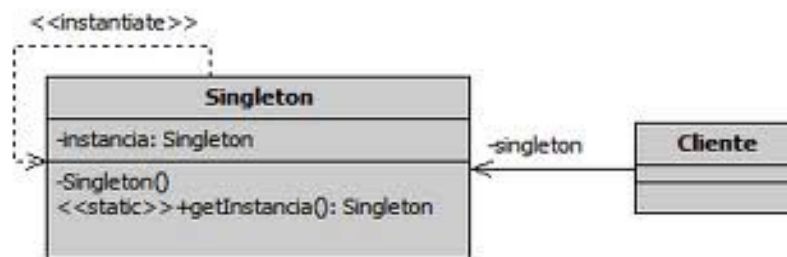


Illustration 11: Patrón Singleton

En este proyecto se utiliza este patrón para mantener una única instancia de la clase que manipula la base de datos.

Pruebas

Pruebas de Persistencia

Se han realizado pruebas automatizadas de la base de datos con ayuda de DBUnit para comprobar que los datos son introducidos y leídos de forma correcta. Se crearon unas tablas nuevas con datos predeterminados que se exportan a un fichero XML antes de empezar los test, y se restauran después de cada prueba.

Pruebas de Integración

Se han creado pruebas para comprobar el correcto funcionamiento de los elementos del sistema en unión utilizando JunitEE. Se ha utilizado Mockito para simular las peticiones a los servlets para comprobar que se comporten de la forma prevista.

Pruebas sobre la Interfaz

Se ha utilizado la herramienta Selenium para crear pruebas automatizadas de la vista de la aplicación. En estas pruebas se recorren todas las partes de la aplicación, se ejecutan todas las funciones y se comprueba que la salida al usuario sea la esperada.

Conclusiones

Dados los objetivos del proyecto se consideran, de uno en uno:

- En cuanto al desarrollo de la base de datos que permita almacenar los datos de usuarios y de los puestos de trabajo, se considera el objetivo cumplido con el sistema desarrollado, que permite guardar datos y ficheros de los usuarios, datos de las ofertas y asociaciones entre ambos.
- En cuanto al desarrollo de una aplicación web que permita acceder y manipular la anteriormente mencionada base de datos de forma sencilla, se considera que la web creada funciona correctamente y cuenta con una interfaz intuitiva que cumple el objetivo.
- En cuanto al objetivo de aumentar los conocimientos de las tecnologías usadas, se considera que se ha cumplido con creces, habiendo expandido drásticamente las competencias en todas las herramientas utilizadas y en la integración de estas en un único sistema.

Mejoras de la aplicación

La aplicación sirve de base para distintas mejoras y ampliaciones, algunas posibilidades podrían ser:

- Se podría hacer uso de Ajax para ejecutar ciertos aspectos de la interfaz en el cliente y reducir las cargas de páginas, aumentando la velocidad y la usabilidad de la aplicación.
- Se podría aumentar el soporte a la evaluación de candidatos introduciendo funcionalidad para comparar candidatos mediante un sistema de puntuaciones y rankings.

PFC: Selección personal

Impresiones

La realización de este proyecto ha sido una experiencia enriquecedora de la que se han sacado valiosas lecciones. Se ha podido comprobar de primera mano la importancia de una planificación bien diseñada y los problemas de ser demasiado optimista con respecto a los plazos.

En cuanto a las herramientas usadas, destaca la gran cantidad de opciones que se presentaron al optar por Java. La enorme selección de frameworks, componentes y entornos de desarrollo puede abrumar y algunas decisiones que se han tenido que tomar en fases tempranas del desarrollo pueden no parecer tan buena idea más tarde.

Por ejemplo, a la hora de implementar la conexión con la base de datos se ha utilizado simple JDBC, pero quizás hubiera sido mejor utilizar Hibernate para contar con las características de mapeo a objetos.

También a destacar el amplio soporte a patrones que ofrece a Java, que ha facilitado la implementación de una aplicación según el patrón MVC. Como en el caso de la base de datos, se llega a la conclusión de que en proyectos futuros sería adecuado el uso de frameworks que favorezcan el diseño de la aplicación, en este caso Spring ofrecería un soporte más que adecuado y facilitaría la fase de testeo en proyectos que hagan uso del patrón MVC.

Como detalle, se ha probado el rendimiento del servidor Glassfish y su capacidad para trabajar con un servidor Apache. Los resultados son satisfactorios y la mayor facilidad para administrar el servidor hace que Glassfish se convierta probablemente en el servidor de aplicaciones preferido para futuros proyectos.

Bibliografía

ALUR, Deepak, CRUPI, John y MALKS Dan. **Core J2EE patterns: best practices and design strategies**. Upper Saddle River, New Jersey: Prentice Hall, 2001.

BOOCH, Grady, RUMBAUGH, James y JACOBSON, Ivar. **The unified modeling language: user guide**. Reading, Massachusetts: Addison-Wesley, 1999.

GÓMEZ-MEJIA, Luis, BALKIN, David y CARDY, Robert. **Gestión de Recursos Humanos**. Madrid: Prentice Hall, 1999.

HUNTER, Jason. **Java Servlet Programming**, Sebastopol, California : O'Reilly, 2001

REESE, George. **Database programming with JDBC and Java**. 2nd ed. Sebastopol, California: O'Reilly, 2000.

SCHAFER, Steven, **HTML, xHTML and CSS bible**. Indianapolis, Wiley 2010.

VASILIEV, Yuli, **Beginning database-driven application development in Java EE using Glassfish**, Berkeley: Apress, 2008.

W3Schools Online Web Tutorials: www.w3schools.com

Anexo 1: Manual del Usuario

Introducción

Este es el manual de uso para la aplicación Selección Personal, desarrollada como proyecto fin de carrera en JAVA EE. En estas páginas se explicará como preparar y utilizar la aplicación.

Requisitos

- Administrador:
 - Servidor Web como Apache Web Server.
 - Servidor de aplicaciones compatible con JAVA y MySQL como Apache Tomcat o Glassfish.
 - Base de Datos MySQL.
 - JAVA EE 6.
 - Navegador Web (Mozilla, Google Chrome, Internet Explorer...)
- Usuario:
 - Navegador Web (Mozilla, Google Chrome, Internet Explorer...)

PFC: Selección personal

Estructura de ficheros

Aquí se verá la lista de ficheros y como están distribuidos en directorios.

Primero se muestran imágenes de la estructura:

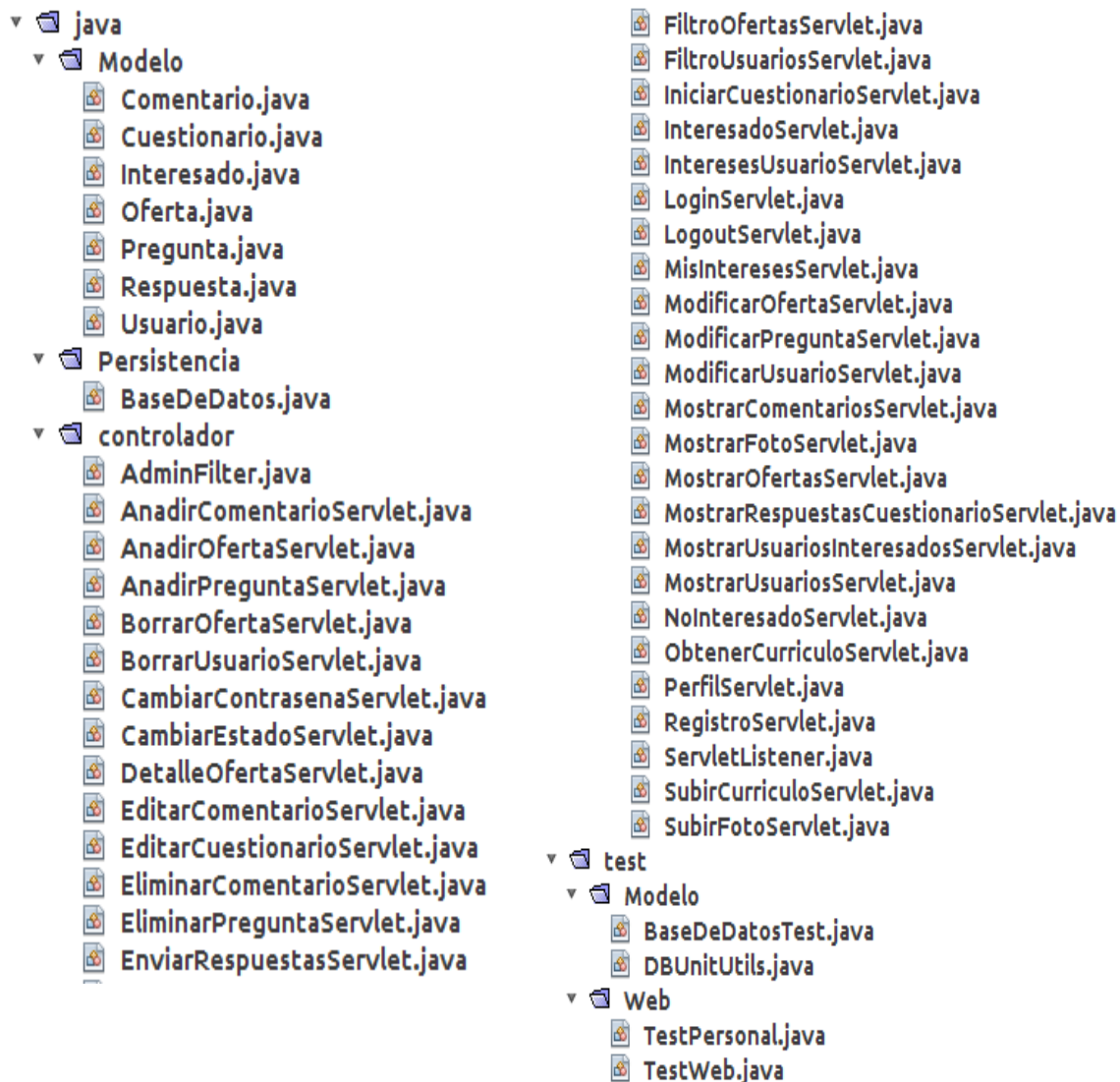


Ilustración 12: Ficheros Java

PFC: Selección personal

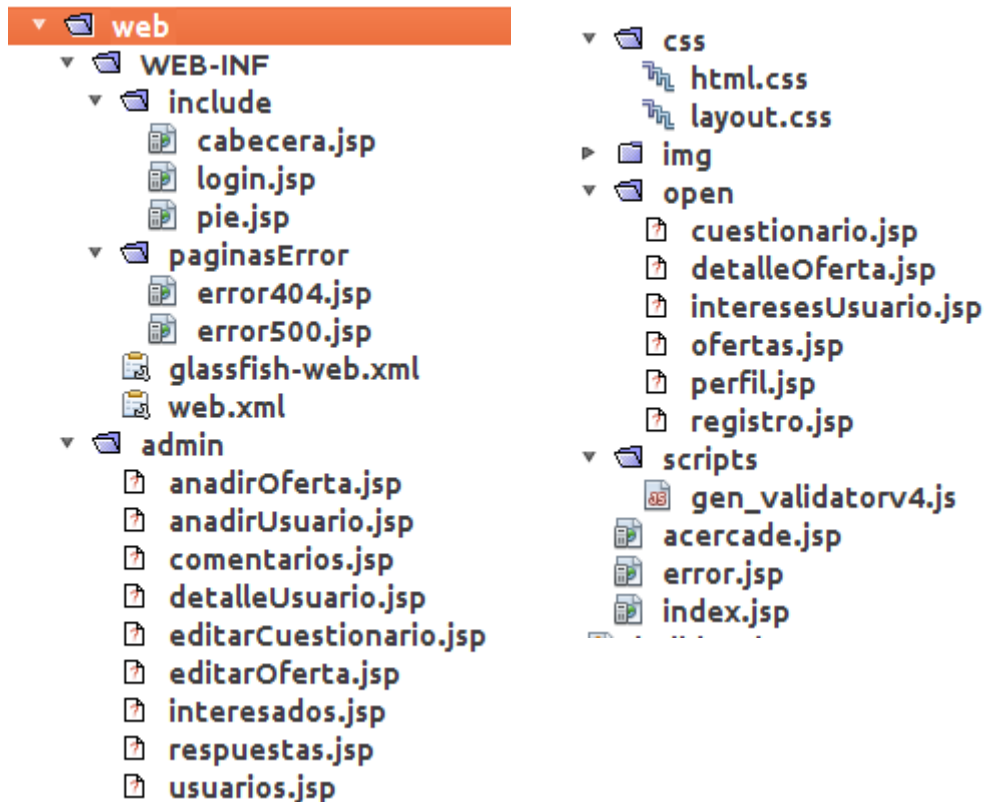


Ilustración 13: Ficheros de la web

La estructura de los ficheros Java es sencilla, en primer lugar vemos la carpeta Modelo, que contiene los Java Beans que forman la base de la aplicación.

Le sigue la carpeta Persistencia, que contiene la clase java con los métodos para conectar con la base de datos del sistema.

Después se encuentra la carpeta Controlador, donde están todos los servlets y filtros que se utilizan para realizar las distintas funciones de la aplicación.

A continuación se ve la carpeta de tests, que incluye las carpetas Modelo y Web, donde se incluyen las clases java que implementan los tests de unidad y los test de integración e interfaz respectivamente.

PFC: Selección personal

En la siguiente imagen se ve la estructura de la vista de la aplicación. Vemos primero la carpeta WEB-INF, donde se incluyen los ficheros JSP a los que no se puede acceder directamente con una petición HTTP, si no que deben ser incluidos en el interior de otras páginas. Así vemos la cabecera, el login y el pie que están incluidos en todas las páginas de la aplicación. Aquí también están las paginas de error a las que dirige el servidor de aplicaciones en caso de errores.

Después podemos ver la carpeta admin, en la que se encuentran aquellas páginas a las que solo se puede acceder con permisos de administrador. De asegurar que así sea se encarga un filtro del controlador situado en la carpeta Controlador anteriormente vista.

A continuación vemos la carpeta css, donde se encuentran las hojas de estilo que definen el aspecto de la vista. El fichero html.css define el formato de letra y aspecto de los distintos elementos html que forman las páginas, mientras que layout.css define la forma de distribución de los elementos en el marco de la página.

La carpeta img contiene todas las imágenes utilizadas en la vista, que son utilizadas según las requiere el css.

En open se encuentran las páginas jsp que no requieren permisos especiales para acceder a ellas. Son las funciones disponibles a invitados y usuarios normales.

Posteriormente se encuentra la carpeta scripts, que guarda los ficheros de JavaScript se han usado, en este caso contiene el fichero gen_validatorv4.js, que contiene scripts de validación de datos para formularios.

Por último están las paginas no contenidas en un directorio, la página principal, la de error y la página con información sobre la aplicación.

Preparar el servidor de Aplicaciones

En este proyecto se utilizó juntos el servidor web Apache y el servidor de aplicaciones Glassfish incluido en la IDE Netbeans. A continuación se explica cómo conectar los servidores para que las peticiones al servidor lleguen hasta Glassfish.

Preparar Glassfish

Se instala primero el servidor Glassfish descargando el paquete o simplemente mediante apt-get.

A continuación se arranca el servidor con:

```
$GLASSFISH_HOME/glassfish/bin/asadmin start-domain
```

Esto inicia el dominio por defecto domain1, con esto se puede acceder al servidor desde el puerto 8080 y a la consola desde el puerto 4848.

A continuación se añade un escuchador HTTP en el puerto 8009 que escuche conexiones AJP. Desde aquí, Apache enviará peticiones al puerto 80 hacia Glassfish.

Creamos el escuchador:

```
$GLASSFISH_HOME/glassfish/bin/asadmin --user admin --host localhost --port 4848 create-http-listener --listeneraddress 0.0.0.0 --listenerport 8009 --defaultvs server jk-connector-8009
```

Lo activamos:

```
$GLASSFISH_HOME/glassfish/bin/asadmin --user admin --host localhost --port 4848 set configs.config.server-config.network-config.network-listeners.network-listener.jk-connector-8009.jk-enabled=true
```

Y reiniciamos Glassfish:

```
$GLASSFISH_HOME/glassfish/bin/asadmin stop-domain  
$GLASSFISH_HOME/glassfish/bin/asadmin start-domain
```

PFC: Selección personal

Preparar Apache

Descargamos el conector `jk_mod` de Tomcat y colocamos el fichero `.so` en el directorio de módulos (`/etc/httpd/modules` por defecto).

Creamos un fichero llamado `workers.properties` con el siguiente contenido:

```
worker.list=worker1
worker.worker1.type=ajp13
worker.worker1.host=localhost
worker.worker1.port=8009
```

A continuación añadimos en el fichero `httpd.conf` de Apache lo siguiente:

```
JkWorkersFile /path/to/workers.properties
JkLogFile /var/log/httpd/mod_jk.log
JkLogLevel debug
JkLogStampFormat "[%a %b %d %H:%M:%S %Y]"
JkOptions +ForwardKeySize +ForwardURISCompat
-ForwardDirectories
JkRequestLogFormat "%w %V %T"
```

Sustituyendo el `path` de nuestro fichero `workers.properties`. Además se añade en la sección de `VirtualHost` de `httpd.conf` lo siguiente:

```
JkMount /aplicacion/* worker1
```

Sustituyendo aplicación por la carpeta que contenga la aplicación desplegada en Glassfish. A continuación se reinicia Apache para que tenga efecto la nueva configuración.

Así, todas las peticiones que lleguen a Apache para ir a <http://dominio.com/aplicacion/> son atendidas por `worker1`, que conecta con el escuchador del puerto 8009 en `localhost` y muestra los contenidos de Glassfish. De esta forma queda lista la aplicación Glassfish accesible desde el puerto 80.

PFC: Selección personal

Despliegue de la aplicación:

Para poder usar la aplicación debe desplegarse primero en un servidor de aplicaciones. En su desarrollo se ha utilizado un servidor Glassfish, si tenemos uno instalado podemos desplegar la aplicación en él de varias formas:

1. Podemos copiar la carpeta de la aplicación directamente en la carpeta “autodeploy” dentro del directorio del servidor Glassfish.
2. Podemos acceder a la consola del servidor a través de un navegador, introduciendo la dirección de la máquina y el puerto 4848 por defecto (ej. localhost:4848). En la pantalla que aparece podemos escoger “Desplegar una aplicación” para que salga una pantalla como la siguiente:

The screenshot shows the 'Desplegar Aplicaciones o Módulos' (Deploy Applications or Modules) page in the Glassfish Administration Console. At the top right are 'Aceptar' (Accept) and 'Cancelar' (Cancel) buttons. Below the title is a note: 'Especifique la ubicación de la aplicación o del módulo que desea desplegar. Una aplicación puede estar en un archivo empaquetado o se puede especificar como directorio.' (Specify the location of the application or module you want to deploy. An application can be in a packaged file or specified as a directory.) and a red asterisk note: '* Indica que es un campo obligatorio' (Indicates that it is a required field).

The 'Ubicación:' (Location) section has two radio button options:

- ☒ **Archivo Empaquetado que Cargar en el Servidor** (Packaged File to Load on the Server). This option has a text input field and an 'Examinar...' (Browse...) button.
- ☐ **Archivo empaquetado local o directorio accesible desde GlassFish Server** (Locally packaged file or directory accessible from GlassFish Server). This option has two buttons: 'Examinar Archivos...' (Browse Files...) and 'Examinar Carpetas...' (Browse Folders...).

The 'Tipo:' (Type) dropdown menu is set to 'Aplicación Web' (Web Application).

The 'Raíz de Contexto:' (Context Root) field is empty, with a hint: 'Ruta de acceso relativa a la URL base del servidor.' (Relative access path to the server's base URL).

The 'Nombre de Aplicación:' (Application Name) field is empty, marked with a red asterisk.

The 'Servidores Virtuales:' (Virtual Servers) section shows a list with one entry, 'server', which is selected. A hint below says: 'Asocia un nombre de dominio de Internet a un servidor físico.' (Associate an Internet domain name with a physical server).

The 'Estado:' (Status) section has a checkbox labeled 'Activada' (Activated) which is checked. A hint says: 'Permite a los usuarios acceder a la aplicación.' (Allows users to access the application).

The 'Precompilar JSP:' (Precompile JSP) checkbox is unchecked. A hint says: 'Precompila páginas JSP durante el despliegue.' (Precompiles JSP pages during deployment).

The 'Ejecutar Verificador:' (Run Validator) checkbox is unchecked. A hint says: 'Verifica la sintaxis y semántica del descriptor de despliegue. Los paquetes de verificador deben estar instalados.' (Checks the syntax and semantics of the deployment descriptor. Validator packages must be installed).

The 'Forzar Nuevo Despliegue:' (Force New Deployment) checkbox is unchecked. A hint says: 'Fuerza el nuevo despliegue incluso aunque esta aplicación ya se haya desplegado o ya exista.' (Forces the new deployment even though this application has already been deployed or already exists).

The 'Mantener Estado:' (Maintain State) checkbox is unchecked. A hint says: 'Mantiene sesiones web, instancias SFSB y temporizadores EJB creados de forma persistente entre nuevos despliegues.' (Maintains web sessions, SFSB instances, and EJB timers created persistently between new deployments).

The 'Bibliotecas:' (Libraries) field is empty. A hint says: 'Lista separada por comas de los archivos JAR de la biblioteca. Especifique los archivos JAR de biblioteca a través de sus rutas de acceso absolutas o relativas. Especifique las rutas de acceso relativas a raiz-instancia /lib/applibs. Las bibliotecas están disponibles para la aplicación en el orden en que se han especificado.' (Comma-separated list of JAR files from the library. Specify JAR files through their absolute or relative access paths. Specify relative access paths relative to raiz-instancia /lib/applibs. Libraries are available for the application in the order they were specified).

The 'Descripción:' (Description) field is empty.

Ilustración 14: Despliegue de Aplicaciones Glassfish

Desde aquí cargamos fácilmente la aplicación.

PFC: Selección personal

Una vez que la aplicación esta desplegada, hay que crear la conexión con la base de datos. Para ello podemos usar la consola del servidor también.

Crearemos un pool de conexiones JDBC, y después un recurso JDBC para que la aplicación pueda conectarse a la base de datos.

Primero hay que crear el Pool de conexiones JDBC. Haciendo click en la parte izquierda de la consola en Recursos → JDBC → Pools de Conexiones JDBC. Nos sale una lista de pools existentes, podemos darle a Nuevo... para que aparezca esto:

Ilustración 15: Creación de Pool de conexiones

PFC: Selección personal

Rellenando esto y dándole a siguiente nos sale la siguiente pantalla:

The image shows a web application interface with two main sections: 'Configuración General' and 'Configuración de Pool'.

Configuración General

- Nombre de Pool:** EjemploPool
- Tipo de Recurso:** javax.sql.DataSource
- Proveedor de Controladores de la Base de Datos:** MySql
- Nombre de Clase de Origen de Datos:** com.mysql.jdbc.jdbc2.optional.MysqlDataSource (selected from a dropdown menu). Below the input field, there is a note: "Seleccione o introduzca un nombre de clase específico del proveedor que implante las API DataSource y/o XADataSource."
- Nombre de Clase de Controlador:** (empty dropdown menu). Below the input field, there is a note: "Seleccione o introduzca un nombre de clase específico del proveedor que implante la interfaz java.sql.Driver."
- Ping:** ☐ **Activada**. Below the checkbox, there is a note: "Si se activa, se hace ping en el pool durante la creación o la nueva configuración a fin de identificar y advertir si hay valores erróneos para sus atributos"
- Descripción:** (empty text input field)

Configuración de Pool

- Tamaño de Pool Inicial y Mínimo:** 8 **Conexiones**. Below the input field, there is a note: "Número mínimo e inicial de conexiones mantenidas en el pool"
- Tamaño de Pool Máximo:** 32 **Conexiones**. Below the input field, there is a note: "Número máximo de conexiones que se pueden crear para responder a las solicitudes del cliente"
- Cantidad de Cambio de Tamaño del Pool:** 2 **Conexiones**. Below the input field, there is a note: "Número de conexiones que se van a eliminar cuando caduque el timeout de inactividad del pool"
- Timeout de Inactividad:** 300 **Segundos**. Below the input field, there is a note: "Tiempo máximo que una conexión puede permanecer inactiva en el pool"
- Tiempo de Espera Máximo:** 60000 **Milisegundos**. Below the input field, there is a note: "Tiempo que espera el emisor de la llamada antes de que se envíe un mensaje de timeout de conexión"

Transacción


- Conexiones No Transaccionales:** ☐ **Activada**. Below the checkbox, there is a note: "Devuelve conexiones que no son transaccionales"
- Aislamiento de Transacción:** read-committed (selected from a dropdown menu). Below the dropdown, there is a note: "Si no se especifica, utiliza el nivel por defecto para el controlador JDBC"
- Nivel de Aislamiento:** ☒ **Garantizado**. Below the checkbox, there is a note: "Todas las conexiones utilizan el mismo nivel de aislamiento: necesita aislamiento de transacción"

Ilustración 16: Creación de pool de conexiones 2

Aquí podemos dejarlo como está, pero conviene seleccionar read-commited en aislamiento de transacción para ahorrarnos algunos problemas de concurrencia.

PFC: Selección personal

Así creamos el pool de conexiones, a continuación creamos un recurso JDBC clickando en Recursos → JDBC → Recursos JDBC y tras seleccionar nuevo nos encontramos esto:



Nuevo Recurso JDBC Aceptar Cancelar

Especifique un nombre JNDI único que identifique el recurso JDBC que desea crear. El nombre sólo debe contener caracteres alfanuméricos, de subrayado, guiones y puntos.

Nombre JNDI: *

Nombre de Pool:

Use la página [Pools de Conexiones JDBC](#) para crear pools nuevos

Descripción:

Estado: ☒ **Activada**

Propiedades Adicionales (0)

Nombre	Valor	Descripción:
No se han encontrado elementos.		

Ilustración 17: Creación de Recurso JDBC

Poniendo nombre al recurso y seleccionando el pool creado anteriormente, le damos a aceptar y ya tenemos el recurso.

A continuación, hay que editar los ficheros de configuración de la aplicación para que acceda a los recursos que acabamos de crear. Debemos editar el fichero web.xml en el directorio de la aplicación para que incluya las siguientes líneas:

```
<context-param>
  <param-name>direccionBase</param-name>
  <param-value>jdbc/Ejemplo</param-value>
</context-param>
<context-param>
  <param-name>nombreBase</param-name>
  <param-value>ejemplo</param-value>
</context-param>
```

Donde jdbc/Ejemplo es sustituido por el nombre que se la haya dado al recurso JDBC y ejemplo es sustituido por el nombre de la base de datos.

PFC: Selección personal

Incluimos en el mismo fichero también lo siguiente:

```
<resource-ref>
  <description>Base de datos para seleccion de personal</description>
  <res-ref-name>jdbc/ejemplo</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
  <res-sharing-scope>Shareable</res-sharing-scope>
</resource-ref>
```

Sustituyendo el nombre del recurso JDBC.

Debemos modificar otro fichero además, el fichero de definiciones del servidor de aplicaciones, que para Glassfish es glassfish-web.xml para que contenga lo siguiente:

```
<resource-ref>
  <res-ref-name>jdbc/ejemplo</res-ref-name>
  <jndi-name>jdbc/ejemplo</jndi-name>
</resource-ref>
```

Hecho esto, la aplicación está lista para ser lanzada accediendo a la dirección del servidor web desde un navegador.

PFC: Selección personal

Sentencias de creación de la base de datos

Para la creación de las tablas SQL se utilizan las siguientes sentencias:

```
DROP DATABASE IF EXISTS personal;
```

```
CREATE DATABASE personal;
```

```
CREATE TABLE personal.Usuarios(  
    Nombre VARCHAR(50) NOT NULL,  
    Apellidos VARCHAR(50) NOT NULL,  
    Email VARCHAR(50) NOT NULL,  
    Telefono VARCHAR(20) NOT NULL,  
    Contraseña VARCHAR(20) NOT NULL,  
    Tipo VARCHAR(1) NOT NULL,  
    CV LONGTEXT,  
    Foto LONGBLOB,  
    CVFichero LONGBLOB,  
    Primary Key(Email)  
);
```

```
CREATE TABLE personal.Ofertas(  
    ID SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,  
    Puesto VARCHAR(50),  
    Descripcion LONGTEXT,  
    Pais VARCHAR(50),  
    Ubicacion VARCHAR(50),
```

PFC: Selección personal

```
Fecha DATE,  
Estudios VARCHAR(100),  
Experiencia VARCHAR(100),  
Requisitos VARCHAR(100),  
Tipo_Contrato VARCHAR(50),  
Sueldo VARCHAR(20),  
Duracion VARCHAR(30),  
Destinatarios VARCHAR(20) DEFAULT 'Abierta' NOT NULL,  
Primary Key(ID)  
);
```

```
CREATE TABLE personal.Interesados(  
    Oferta SMALLINT UNSIGNED NOT NULL,  
    Puesto VARCHAR(50) NOT NULL,  
    Email VARCHAR(50) NOT NULL,  
    Estado VARCHAR(20) DEFAULT 'Apuntado' NOT NULL,  
    Fecha DATE,  
    Primary Key (Oferta,Email)  
    Foreign Key(Oferta) References personal.Ofertas(ID) ON DELETE CASCADE  
    Foreign Key(Usuario) References personal.Usuarios(Email) ON DELETE  
CASCADE,  
);
```

```
CREATE TABLE personal.Cuestionarios(  
    ID SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,  
    Nombre VARCHAR(30) NOT NULL,  
    Oferta SMALLINT UNSIGNED UNIQUE NOT NULL,  
    Primary KEY(ID),  
    Foreign Key(Oferta) References personal.Ofertas(ID) ON DELETE CASCADE  
);
```

PFC: Selección personal

```
CREATE TABLE personal.Preguntas(  
    ID SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,  
    Enunciado LONGTEXT,  
    Cuestionario SMALLINT UNSIGNED NOT NULL,  
    Primary Key(ID),  
    Foreign Key(Cuestionario) References personal.Cuestionarios(ID) ON DELETE  
    CASCADE  
);
```

```
CREATE TABLE personal.Respuestas(  
    Usuario VARCHAR(50) NOT NULL,  
    Pregunta SMALLINT UNSIGNED NOT NULL,  
    Contenido LONGTEXT,  
    FOREIGN KEY(Usuario) REFERENCES personal.Usuarios(Email) ON  
    DELETE CASCADE,  
    FOREIGN KEY(Pregunta) REFERENCES personal.Preguntas(ID) ON  
    DELETE CASCADE  
);
```

```
CREATE TABLE personal.Comentarios(  
    ID SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,  
    Fecha DATETIME,  
    Usuario VARCHAR(50) NOT NULL,  
    Texto LONGTEXT,  
    Autor VARCHAR(50) NOT NULL,  
    PRIMARY KEY(ID),  
    FOREIGN KEY(Usuario) REFERENCES personal.Usuarios(Email) ON  
    DELETE CASCADE  
);
```

PFC: Selección personal

Guía

Inicio:

Al acceder a la aplicación se muestra lo siguiente:

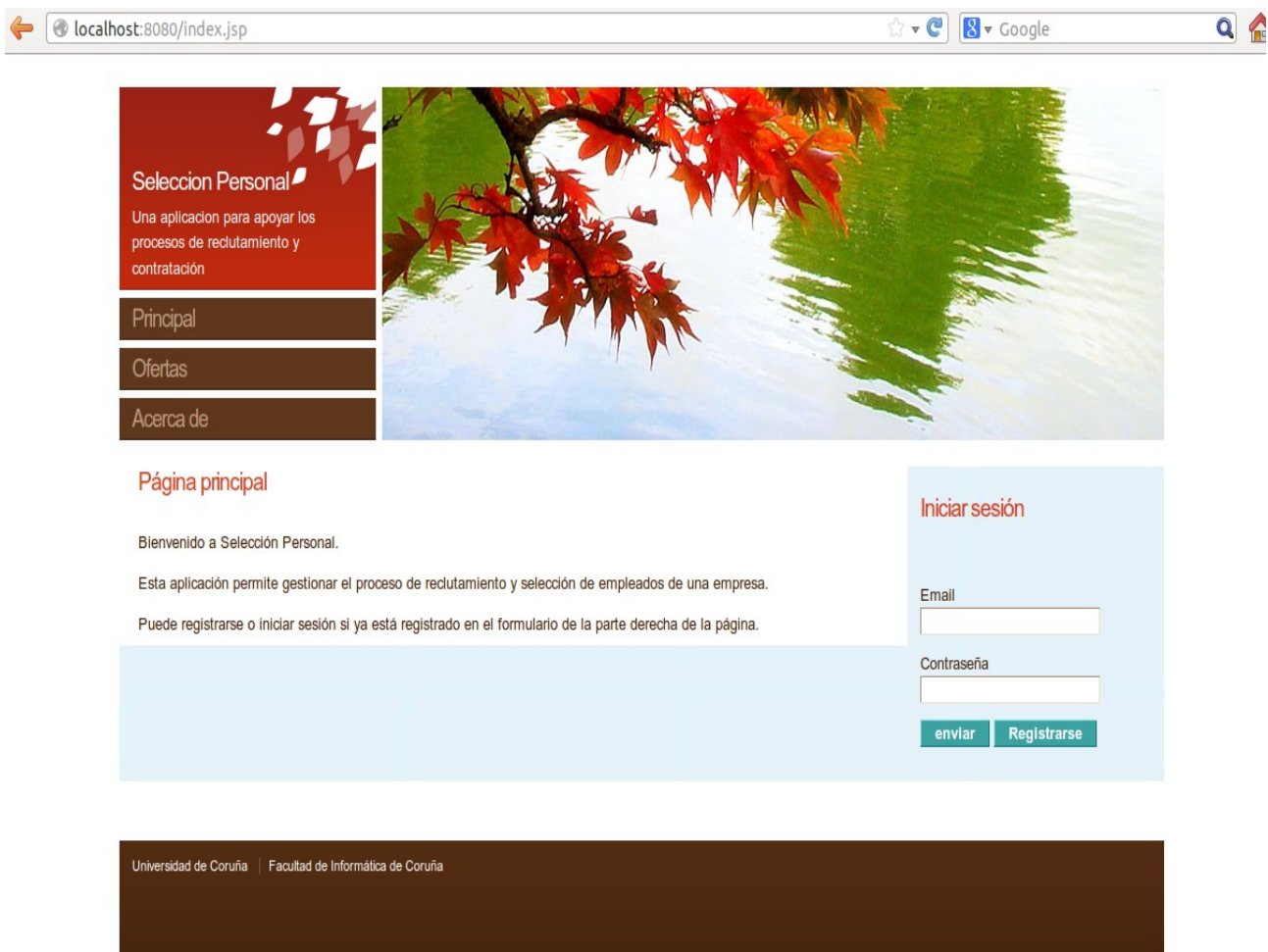


Ilustración 18: Página Principal

Aquí podemos ver el menú principal en la parte superior derecha de la pantalla y que está presente en todo momento en la aplicación.

A la derecha está la zona para iniciar una sesión o registrarse si no se tiene una cuenta.

PFC: Selección personal

El menú superior principal permite acceder a cada sección de la web, y cambia según los permisos del usuario:



Ilustración 19: Menú para Administradores

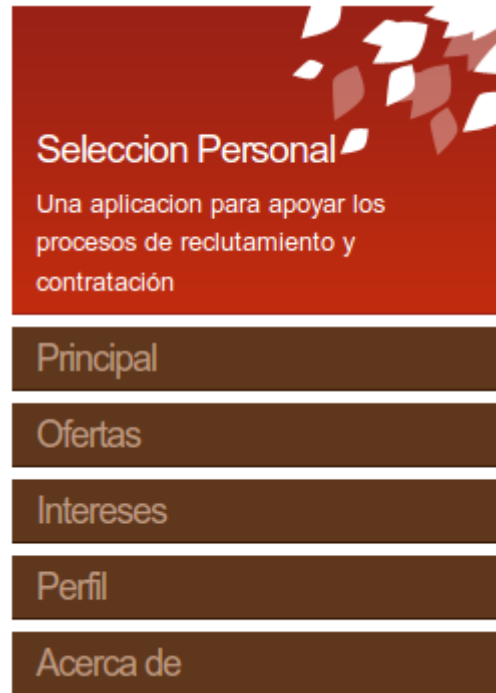


Ilustración 20: Menú para Usuarios registrados



Ilustración 21: Menú para Invitados

PFC: Selección personal

Guía para usuarios

Registro

Haciendo click en la opción de registrarse de la parte de login situada a la derecha, se pasa a la siguiente pantalla:

Registro de usuarios

Nombre	<input type="text" value="Nombre"/>
Apellidos	<input type="text" value="Apellidos"/>
Email	<input type="text" value="email@email.com"/>
Contraseña	<input type="password" value="*****"/>
Repetir contraseña	<input type="password" value="*****"/>
Telefono	<input type="text" value="123456789"/>
CV	<div><div>Currículo de ejemplo</div></div>

enviar

Ilustración 22: Registro de Usuarios

El email será el nombre que el usuario deberá usar para iniciar la sesión. El campo de CV permite escribir un currículum en texto simple, que puede ser filtrado después por la aplicación para encontrar usuarios que se ajusten a un perfil. Una vez creada una cuenta, se puede acceder a las funciones de la aplicación.

PFC: Selección personal

Ofertas:

Desde la pantalla de ofertas se puede ver la lista de las ofertas disponibles, se puede filtrar para mostrar solo las ofertas que se ajusten a ciertos criterios y se puede acceder a la información de cada oferta disponible. Esta sección esta abierta incluso para invitados.

Listado de Ofertas

Mis Intereses

Filtrar

Puesto

Descripción

País

Ilustración 23: Listado de Ofertas

Intereses:

La pantalla de intereses muestra una tabla con las ofertas en las que el usuario se ha inscrito y el estado actual de estas.

Listado de Ofertas en las que está inscrito

Puesto	Estado	Fecha de Inscripción
Analista de Sistemas	Apuntado	03/04/2013
Diseñador Web	Apuntado	03/04/2013

Ilustración 24: Listado de Ofertas Inscritas

PFC: Selección personal

Detalles de Oferta:

Selección Personal

Una aplicación para apoyar los procesos de reclutamiento y contratación


Principal

Ofertas

Intereses

Perfil

Acerca de



Analista de Sistemas

Fecha: 2013-04-03

LUGAR

País: España

Ubicación: Huesca

DESCRIPCIÓN

Puesto: Analista de Sistemas

Descripción: Descripción de un trabajo de analista

Sueldo: 1200 € al mes

Duración del contrato: 6 meses

Tipo de Contrato: Tiempo Completo

REQUISITOS

Estudios mínimos: Algunos

Experiencia mínima: Toda

Requisitos mínimos: Muchos

Inscribirse

SESION INICIADA

Usuario: jackjack@gmail.com

Cerrar sesión

Ilustración 25: Detalle Oferta

PFC: Selección personal

Perfil:

Tu Perfil

Nombre	<input type="text" value="Nombre"/>
Apellidos	<input type="text" value="Apellidos"/>
Email	<input type="text" value="email@email.com"/>
Telefono	<input type="text" value="123456789"/>
Curriculum Vitae	<div>Currículo de ejemplo</div>

Modificar

Contraseña actual	<input type="password"/>
Nueva Contraseña	<input type="password"/>
Repetir Contraseña	<input type="password"/>

Cambiar Contraseña

Foto	<input type="text"/>	Examinar...
------	----------------------	-------------

Cambiar foto

Curriculo pdf	<input type="text"/>	Examinar...
---------------	----------------------	-------------

Subir CV Descargar CV

Darse de Baja

Ilustración 26: Perfil

Desde aquí, el usuario puede modificar sus datos, cambiar su contraseña y subir una foto o un fichero con su currículum. También tiene la opción si lo desea de dar de baja su cuenta y eliminar todos sus datos del sistema.

PFC: Selección personal

Guía para administradores

El usuario que cuenta con permisos de administrador dispone de una serie de opciones adicionales sobre un usuario corriente.

Ofertas

La sección de ofertas permite a un administrador introducir nuevas ofertas en el sistema, suprimir ofertas ya existentes o ver una lista de los usuarios que están inscritos en una oferta.

Listado de Ofertas

[Nueva oferta](#)

[Filtrar](#)

Puesto

Descripción

País

Ilustración 27: Ofertas como Administrador

PFC: Selección personal

Nueva Oferta

En este formulario se rellenan los datos de la oferta a añadir y se deciden los destinatarios de la oferta. Puede ser una oferta abierta que cualquiera puede ver, una destinada a empleados que solo usuarios que sean empleados de la empresa verán o bien una oferta cerrada que visible solo a administradores hasta que se modifique su estado.

Nueva oferta


Puesto	<input type="text"/>
Descripción	<div><div></div></div>
País	<input type="text"/>
Ubicación	<input type="text"/>
Estudios mínimos	<input type="text"/>
Requisitos mínimos	<input type="text"/>
Experiencia mínima	<input type="text"/>
Tipo de contrato	<input type="text"/>
Sueldo	<input type="text"/>
Duración del contrato	<input type="text"/>
Destinatarios	<div>Abierta </div>
<div>Publicar</div>	

Ilustración 28: Nueva Oferta

PFC: Selección personal

Editar Cuestionario

Se puede añadir un cuestionario a una oferta para que los usuarios que se inscriban contesten a determinadas preguntas. Para ello se puede hacer click en el botón de editar cuestionario una vez creada una oferta.

Editar cuestionario de la oferta Pocero Ejecutivo

Nombre [Modificar](#)

Pregunta 1:

¿De qué color es el caballo blanco de Santiago?

[Modificar](#) [Eliminar](#)

Añadir nueva Pregunta:

[Añadir Pregunta](#)

Ilustración 29: Editar Cuestionario

Aquí se puede poner un nombre para el cuestionario y ver, editar y eliminar las preguntas que contenga, o añadir una pregunta nueva.

PFC: Selección personal

Interesados

Desde la sección de ofertas, un administrador puede acceder a una lista de usuarios inscritos en ella, acceder a sus perfiles para ver sus datos y modificar el estado de su solicitud, que al inscribirse es Apuntado por defecto.

Interesados en el puesto de Diseñador Web

Nombre	Apellidos	Email	Estado		
Jack	McJackington	jackjack@gmail.com	Apuntado	Cambiar Estado	Detalle
Jacobo	Rodríguez	jacobo.rodriguez@ya.com	Descartado	Cambiar Estado	Detalle

Ilustración 30: Interesados como Administrador

Usuarios

Un administrador tiene acceso también a la lista completa de usuarios registrados:

Listado de Usuarios

[Nuevo Usuario](#)

[Filtrar](#)

Apellidos

Clave Currículo

Nombre	Apellidos	Email	Telefono	Permisos		
Jack	McJackington	jackjack@gmail.com	555678467	Candidato	Detalle	Inscripciones
Jacobo	Rodríguez	jacobo.rodriguez@ya.com	564565667	Candidato	Detalle	Inscripciones
Pepote	Ramirez	pepram@gmail.com	555467634	Empleado	Detalle	Inscripciones

Desde aquí puede acceder a los perfiles de los usuarios para ver sus datos y descargar sus currículos, puede acceder a la lista de ofertas en las que está inscrito un usuario y puede añadir un nuevo usuario al sistema.

PFC: Selección personal

Editar Usuario

Un administrador que accede al perfil de un usuario obtiene una pantalla ligeramente distinta que la que obtiene el propio usuario al acceder a él.

Editar Usuario

Nombre	<input type="text" value="Jacobo"/>
Apellidos	<input type="text" value="Rodríguez"/>
Email	<input type="text" value="jacobro.rodriuez@ya.com"/>
Telefono	<input type="text" value="564565667"/>
CV	<div>Estudios de Alpinista Adjunto en la Universidad Complutense de Nepal del Toto</div>
Permisos	<div>Candidato</div> <div>Modificar</div>
Foto	<div><input type="text"/> Examinar...</div> <div>Cambiar foto</div>
Curriculo pdf	<div><input type="text"/> Examinar...</div> <div>Subir CV Descargar CV</div>
<div>Eliminar Usuario</div>	

Ilustración 31: Editar Usuario como Administrador

Aquí, además de modificar los datos o eliminar el usuario, el administrador puede cambiar los permisos de éste para identificarlo como un empleado de la empresa (y por tanto darle acceso a las ofertas para reclutamiento interno) o darle permisos de administrador si es apropiado.