

## APLICACIÓN GESTIÓN RRHH\_ JAVA

### 1- INTRODUCCIÓN

Una de las tareas primordiales para una empresa consiste en seleccionar personal para su plantilla, que facilite el desarrollo de sus actividades de la forma más competente posible. Normalmente se busca personal para un puesto de trabajo específico y se requieren unas capacidades o habilidades concretas. Es trabajo del departamento de recursos humanos, en adelante RRHH, estudiar cada currículum que les entregan y seleccionar los mejores candidatos para cada puesto en función de sus habilidades y experiencias.

A menudo estos currículums se amontonan y se deben leer hojas y hojas de candidatos que no interesan o se dejan de leer currículums de otros que podrían estar muy cualificados para la empresa.

En otras ocasiones, se puede llegar a entrevistar a un candidato en más de una ocasión, sin disponer de los datos de otras entrevistas al ser realizadas por evaluadores diferentes.

Una búsqueda de un candidato entre 50 posibles, implica leer un mínimo de 100 hojas de currículums, hacer entrevistas a los que puedan interesar y después seleccionar los más aptos. Mientras que un sistema de información permitiría hacer una búsqueda con los parámetros requeridos, obteniendo un resultado de los candidatos aptos ordenados por sus habilidades, agilizando el proceso de selección, ahorrando tiempo, recursos y, por lo tanto, dinero.

Otro asunto en los procesos de selección es que se requiere un trabajo colaborativo inter-departamental, ya que pueden entrar en juego competencias de distintos perfiles. Con esta aplicación se pretende que el trabajo del departamento de RRHH sea mucho

más eficaz tanto en el tiempo empleado en la búsqueda de personal, como en los resultados obtenidos en la selección de candidatos.

- Herramientas para desarrollar la aplicación a elegir por el grupo (tecnología de desarrollo en java)

## 2- OBJETIVOS

El objetivo principal es el desarrollo del sistema de información “Aplicación web para la gestión curricular de la empresa CIC Consulting informático”, aportando valor añadido al departamento de RRHH de la empresa.

Para la realización de este proyecto, se definen de manera más específica los siguientes sub-objetivos:

1. Almacenamiento de toda la información relevante de los candidatos y sus currículums.
2. Guardar los datos de los candidatos presentados a cada proceso de selección y la información generada en sus entrevistas.
3. Implementación de un potente buscador de candidatos por habilidades, titulaciones y experiencia.
4. Construcción de una interfaz de la aplicación accesible y funcional desde distintos dispositivos.
5. Conectar con el directorio activo de la empresa para autenticar a los usuarios del sistema de información.
6. Funcionalidad para mostrar estadísticas con la información recibida en la Aplicación

### 3- ANÁLISIS DEL SISTEMA

Para ello, primero hay que llevar a cabo una identificación de los diferentes escenarios que se puedan dar, así como del personal que hará uso del sistema.

A continuación, se debe llevar una especificación de casos de uso, a partir de la cual hay que obtener los requisitos tanto funcionales como no funcionales que debe cumplir la aplicación.

#### - 3.1. ESPECIFICACIÓN DE CASOS DE USO

En ingeniería de software, un caso de uso proporciona un posible uso de la aplicación en un momento determinado, dentro de un contexto determinado, llevado a cabo por un usuario determinado. Generalmente, se utiliza para definir el comportamiento de la aplicación con un diagrama fácil de interpretar para una persona no técnica. Con que suele utilizarse para definir la funcionalidad con el cliente.

Para simplificar el diagrama se ha utilizado un patrón de casos de uso que llamaremos CRUD. Este patrón nos permite especificar una o varias de las acciones Create (Crear), Read (Leer), Update (Modificar) y Delete (Borrar) definiéndolo en un único caso de uso. Estos casos de uso estarán definidos por las iniciales de las acciones CRUD pudiendo tener una o varias de ellas, seguida por la entidad sobre la que actúan.

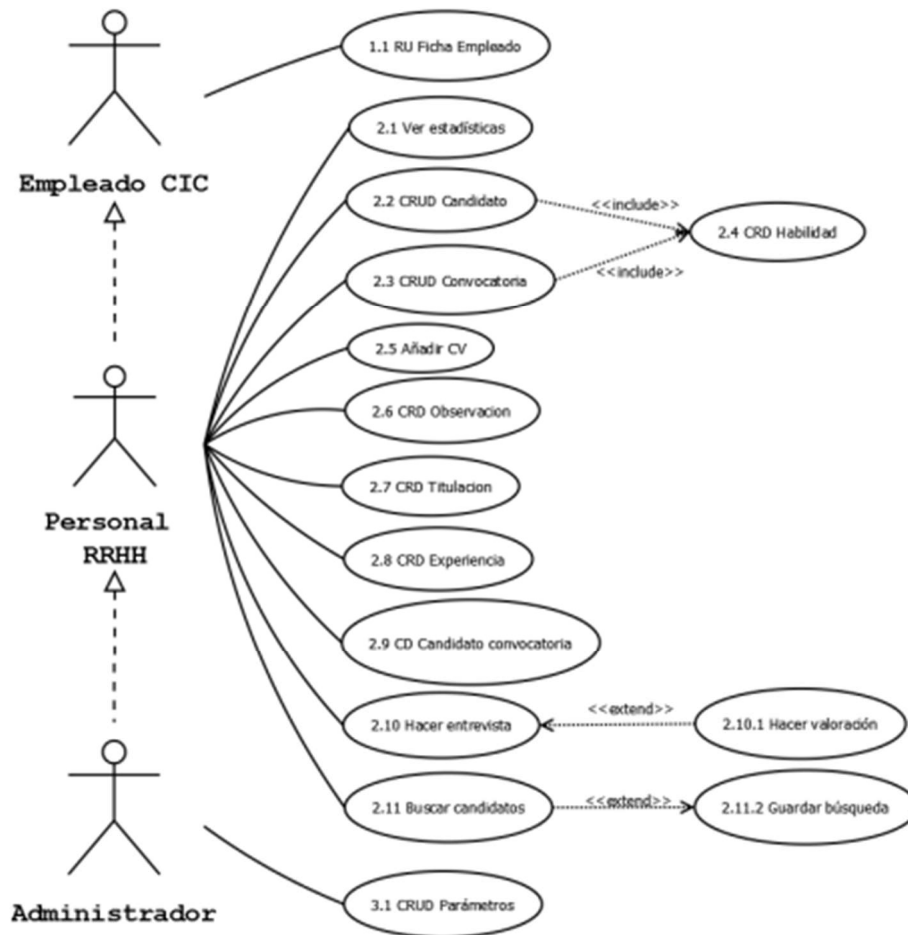


Ilustración 26. Diagrama de Casos de Uso del Sistema

### - 3.1.2 ESPECIFICACIÓN DE REQUISITOS

El objeto de la especificación es definir de manera clara y precisa todas las funcionalidades y restricciones del sistema que se desea construir. Los requisitos deben ser medibles, comprobables y sin ambigüedades o contradicciones.

A continuación, se especifican los requisitos que debe cumplir el sistema. Estos se dividen en dos grandes bloques:

- Funcionales, aquellos requisitos que determinan la funcionalidad del software, esto es, todas las acciones que pueden realizar los diferentes roles de la aplicación.
- No funcionales, aquellos requisitos que determinan las restricciones o condiciones de otros aspectos como la calidad, estándares, costes, estabilidad y portabilidad.

#### - 4 ARQUITECTURA

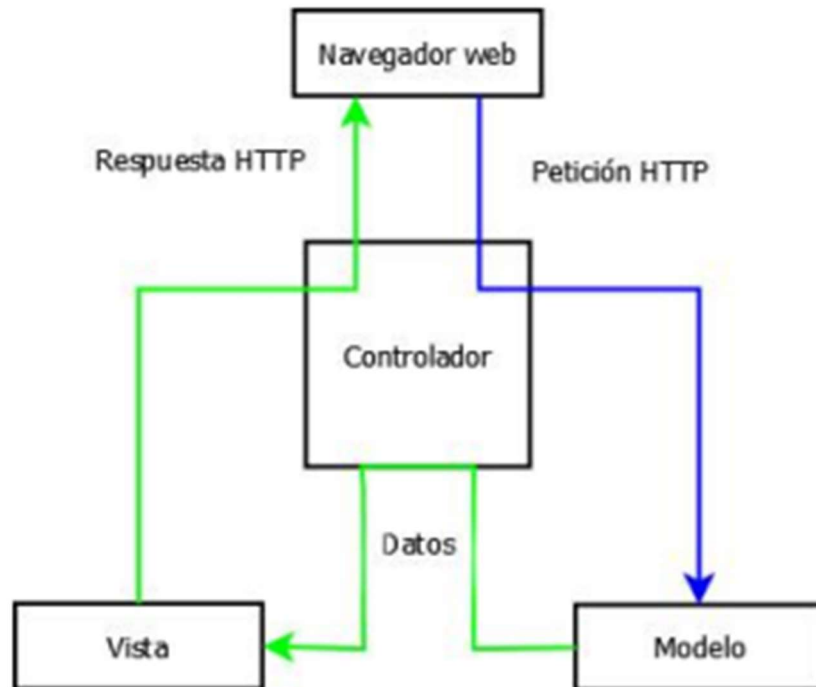
El modelo vista controlador es un patrón de arquitectura software que separa la aplicación en distintas capas para hacerla más modular, “testable” y segura, entre otras cosas.

Las capas de las que consta son las siguientes:

- Modelo: Gestiona los accesos que se realizan a la base de datos que contiene la información que utiliza la aplicación.
- Vista: Esta es la capa con la que la aplicación interactúa con el usuario. Esta es la capa con la que interactúa el usuario (normalmente una interfaz gráfica).
- Controlador: Contiene toda la lógica de negocio, recogiendo los datos necesarios del modelo y pasando la información necesaria a las vistas.

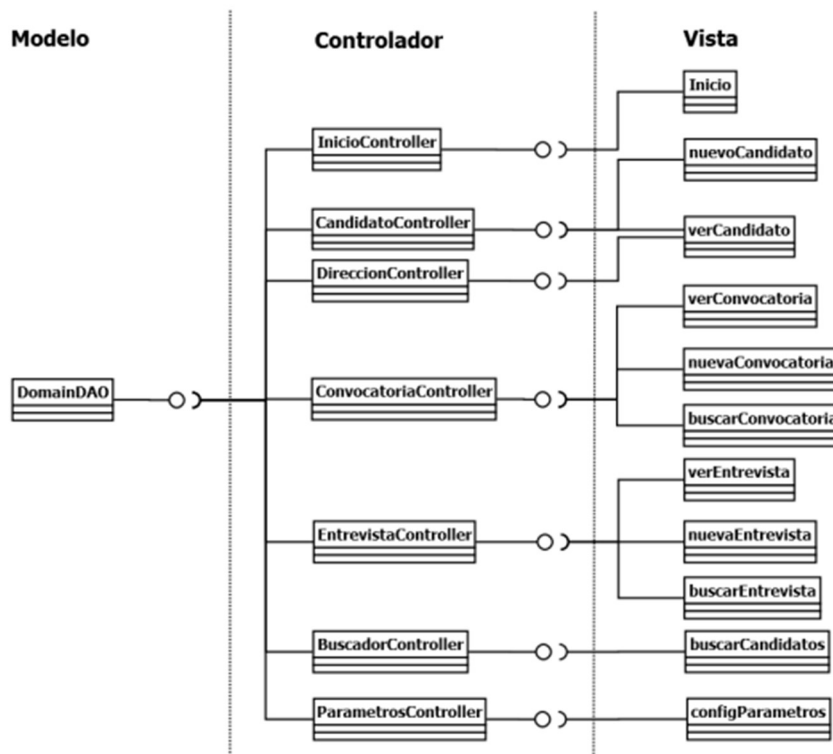
Para cada implementación pueden diferir algunos detalles en la implementación y en cómo se comunica cada capa. En el caso de esta aplicación, el flujo de control que debe seguir es el siguiente:

1. El usuario interactúa con la interfaz de usuario desde su navegador web. (por ejemplo, el usuario envía un formulario).
2. El controlador recibe la notificación de la acción solicitada por el usuario. Éste gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
3. El controlador realiza las operaciones pertinentes con los datos y accede al modelo actualizando la base de datos con la nueva información.
4. El controlador genera la vista de acuerdo con la petición del usuario, que es enviada a su navegador web.
5. La interfaz de usuario espera nuevas interacciones, comenzando el ciclo de nuevo.



*Ilustración 29. Esquema arquitectura MVC*

## 4- DIAGRAMA DE LA APLICACIÓN



## 6- COMO IMPLEMENTAR LA APLICACIÓN

- En primer lugar, se implementa el modelo, ya que con éste definimos los datos necesarios a almacenar y a mostrar posteriormente.
- Después, se hay que tomar un requisito cada vez, escribiendo sus pruebas correspondientes.
- Por último, se debe implementar el controlador y la vista necesarios para dotarlo de la funcionalidad prevista.

## 7- MAPA WEB DE LA APLICACIÓN

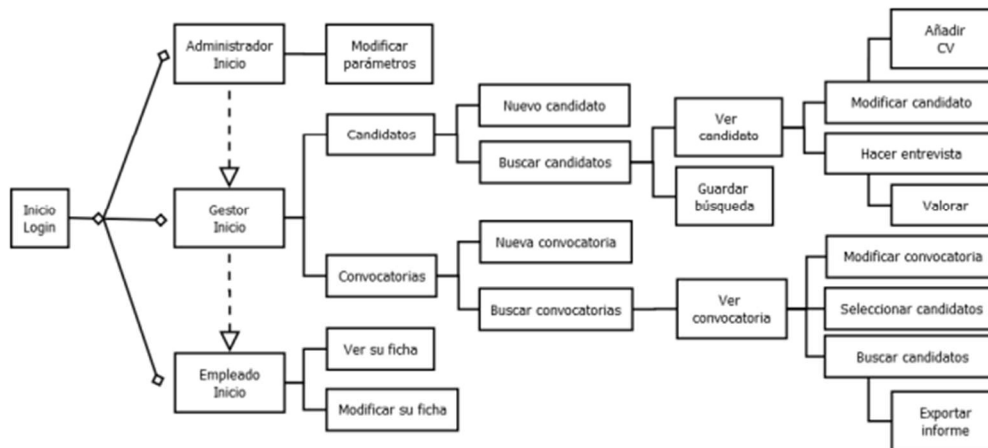


Ilustración 33. Mapa Web de la Aplicación