

Documentación de la aplicación web Work Market:

Para desarrollar una aplicación web, es importante contar con una documentación adecuada que describa todos los aspectos relevantes del proyecto.

Presentación del Proyecto de Work Market:

Work Market es una plataforma web diseñada para facilitar la contratación de personal y la búsqueda de empleo, proporcionando un espacio en línea donde los empleadores pueden publicar puestos de trabajo y los candidatos interesados pueden postularse.

Nuestra plataforma tiene como objetivo conectar a empleadores y candidatos de manera efectiva, proporcionando una solución simplificada y transparente para la contratación y la búsqueda de empleo. A través de sus características y funcionalidades, Work Market busca mejorar la eficiencia del proceso de contratación y fomentar una relación laboral más satisfactoria, sencilla y confiable para ambas partes.

Las personas podrán buscar y acceder a los puestos de trabajo que deseen, y explorar diversas opciones de empleo, desde trabajos temporales hasta empleos a tiempo completo. Esto les permite adaptar su búsqueda a sus preferencias y necesidades específicas, utilizando las herramientas necesarias para filtrar los trabajos por ubicación, categoría, nivel de experiencia, estudios, y otros criterios relevantes. Una vez que encuentran un trabajo de su interés, pueden postularse directamente a través de la plataforma a través de la creación de perfiles detallados donde proporcionen su información, resalten sus habilidades, experiencia y logros, y adjuntando su currículum u otros documentos relevantes que los ayuden a aumentar sus posibilidades de ser seleccionados para los puestos deseados.

El alcance de Work Market abarca una amplia gama de sectores y tipos de empleo, desde trabajos temporales hasta empleos a tiempo completo. No solo podrá ser utilizado por personas con una carrera universitaria, sino que también abarcará oficios.

La plataforma permite la publicación de trabajos en diversos campos, como tecnología, marketing, diseño, finanzas, servicios profesionales, entre otros, con el objetivo de cubrir las diferentes necesidades laborales.

Nuestro propósito principal es conectar a empresas y profesionales en un entorno virtual eficiente y transparente, fomentando la contratación ágil y efectiva.

Los objetivos principales que pretendemos alcanzar con Work Market son:

- **Simplificar el proceso de contratación:** Work Market busca agilizar y simplificar el proceso de contratación, eliminando las barreras tradicionales y reduciendo los plazos para encontrar y seleccionar candidatos, brindando una mayor información a las personas interesadas en conseguir un puesto.
Los empleadores pueden crear fácilmente perfiles de trabajo detallados y recibir solicitudes de personas interesadas, lo que les permite encontrar rápidamente lo que buscan.
- **Facilitar la búsqueda de empleo:** La plataforma brinda a los candidatos una forma sencilla de explorar y postularse a una amplia variedad de oportunidades laborales. Las personas pueden crear perfiles que resalten sus habilidades y experiencia, y utilizar herramientas de búsqueda avanzada para encontrar trabajos que se ajusten a sus preferencias y capacidades.
- **Promover la transparencia y la confianza:** Esta página tiene como finalidad promover la transparencia y la confianza en el proceso de contratación. Los empleadores pueden revisar los perfiles de los candidatos, incluyendo sus calificaciones y comentarios de trabajos anteriores, lo que les ayuda a tomar decisiones informadas. Del mismo modo, los candidatos también pueden investigar y evaluar a los empleadores antes de postularse, lo que fomenta una relación laboral más equitativa y satisfactoria.
- **Impulsar la flexibilidad laboral:** Work Market apoya la creciente demanda de flexibilidad laboral al ofrecer oportunidades de empleo temporal o permanente. Los profesionales pueden encontrar proyectos a corto o largo plazo que se ajusten a sus horarios y preferencias, lo que les brinda la posibilidad de diversificar su experiencia laboral y adaptarse a diferentes necesidades.

Herramientas utilizadas:

- Lenguajes Java, HTML, JSP, SQL, CCS
- IDE NetBeans
- Servidor web Apache tomcat
- Servidor de base de datos MySql con Workbench
- Patrón MVC

Java: Java es un lenguaje de programación ampliamente utilizado en el desarrollo de aplicaciones web. Es conocido por su portabilidad, ya que puede ejecutarse en diferentes sistemas operativos sin necesidad de modificaciones adicionales. Java ofrece una amplia gama de bibliotecas y frameworks que facilitan el desarrollo de aplicaciones robustas y escalables. Java fue utilizado para desarrollar la lógica de negocio de la aplicación, como la gestión de usuarios, la creación y publicación de puestos de trabajo, y el procesamiento de las solicitudes de los candidatos.

HTML (HyperText Markup Language): es el lenguaje de marcado estándar utilizado para crear la estructura y el contenido de las páginas web. Se utiliza para definir la disposición de los elementos y organizar el contenido mediante etiquetas y atributos. HTML se complementa con CSS y JavaScript para mejorar la apariencia y la funcionalidad de las páginas web. HTML fue empleado para crear la estructura y el contenido de las páginas web, permitiendo definir la apariencia y la disposición de los elementos visuales.

JSP (JavaServer Pages): JSP es una tecnología basada en Java que permite la generación dinámica de contenido web. Permite combinar código Java y HTML dentro de un mismo archivo, lo que facilita la creación de páginas web dinámicas y la integración con lógica de negocio compleja. JSP se ejecuta en el servidor y genera HTML que se envía al cliente. JSP permitió la generación dinámica de contenido web al combinar código Java con HTML, facilitando la presentación de información personalizada y la interacción con los usuarios.

SQL (Structured Query Language): es un lenguaje de programación utilizado para interactuar con bases de datos relacionales. Permite realizar consultas y manipular datos almacenados en tablas, lo que resulta fundamental para el almacenamiento y la recuperación de información en una aplicación web. SQL se utiliza para crear, modificar y consultar la base de datos. MySql se utilizó como el sistema de gestión de base de datos para almacenar y recuperar información relacionada con los puestos de trabajo, los candidatos, los empleadores, entre otros datos relevantes. MySql Workbench nos permitió diseñar y administrar la base de datos, crear tablas, definir relaciones y realizar consultas SQL.

IDE NetBeans: es un Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés) utilizado para desarrollar aplicaciones en Java. Proporciona herramientas y características que facilitan la escritura de código, la depuración, el diseño de interfaces de usuario y la gestión de proyectos. NetBeans es ampliamente utilizado y cuenta con una comunidad

activa de desarrolladores. Nos proporcionó un entorno de desarrollo integrado para escribir y depurar el código Java, facilitando la programación eficiente y la gestión del proyecto.

Servidor web Apache Tomcat: es un servidor web y un contenedor de servlets de código abierto que se utiliza para ejecutar aplicaciones web basadas en Java. Proporciona un entorno de ejecución seguro y eficiente para desplegar y ejecutar aplicaciones web Java. Tomcat es ampliamente utilizado debido a su estabilidad y rendimiento. Apache Tomcat actuó como el servidor web y el contenedor de servlets Java para ejecutar Work Market, gestionando las solicitudes y respuestas HTTP, así como la ejecución de los componentes Java.

Servidor de base de datos MySQL con Workbench: MySQL es un sistema de gestión de bases de datos relacional de código abierto. Es ampliamente utilizado en el desarrollo web debido a su rendimiento, confiabilidad y facilidad de uso. MySQL Workbench es una herramienta gráfica que permite administrar y diseñar bases de datos MySQL. Permite crear, modificar y gestionar tablas, consultas y otros aspectos relacionados con la base de datos.

En conjunto, estas herramientas y lenguajes permitieron la creación de Work Market como una aplicación web funcional. Java, HTML y JSP se utilizaron para desarrollar la lógica y la interfaz de usuario, NetBeans nos proporcionó un entorno de desarrollo integrado, Apache Tomcat actuó como servidor web, y a MySQL con Workbench lo utilizamos para gestionar la base de datos.

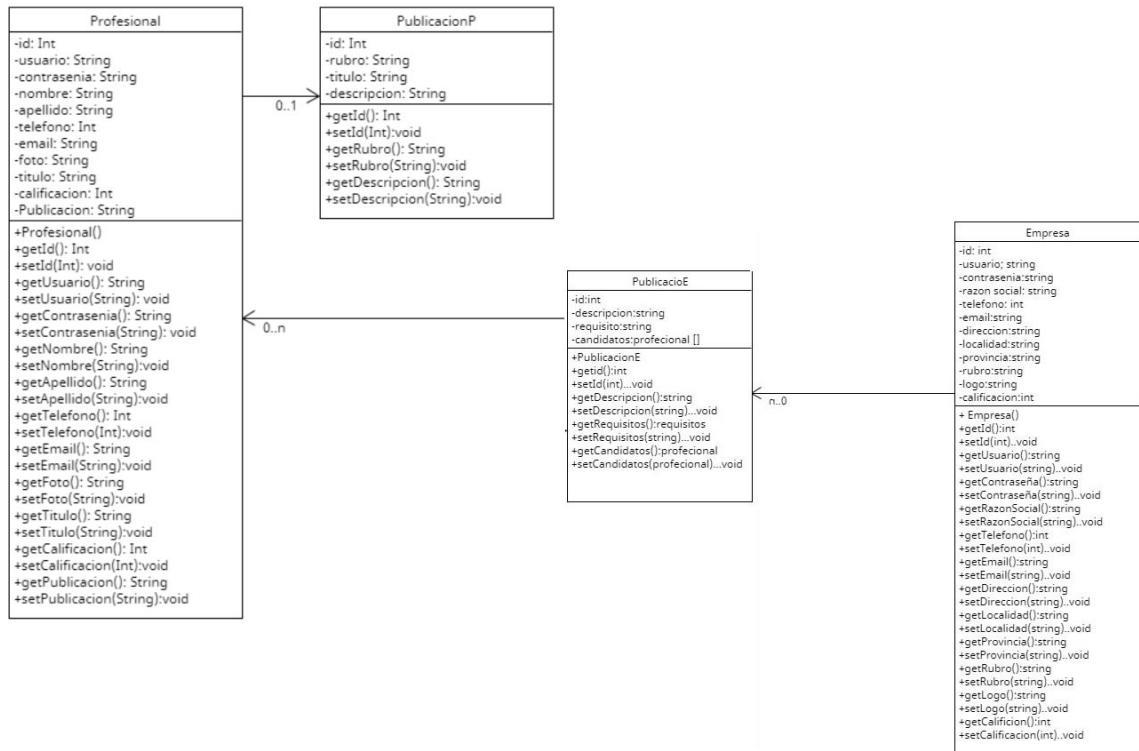
Modelo, vista, controlador: Para responder al Patrón MVC, definimos en nuestro proyecto un paquete modelo en el que se encuentran las clases de dominio, las vistas las definimos en la carpeta web que son todos los archivos HTML y JSP. Y para conectar ambas partes utilizamos los servlet de java los cuales permiten asumir el rol de controladores, haciendo de nexo entre la vista y el modelo de dominio.

Modelo adaptativo: definimos 5 zonas dentro de cada página: cabecera, barra de navegación lateral izquierda, barra lateral derecha, cuerpo principal y pie de página.

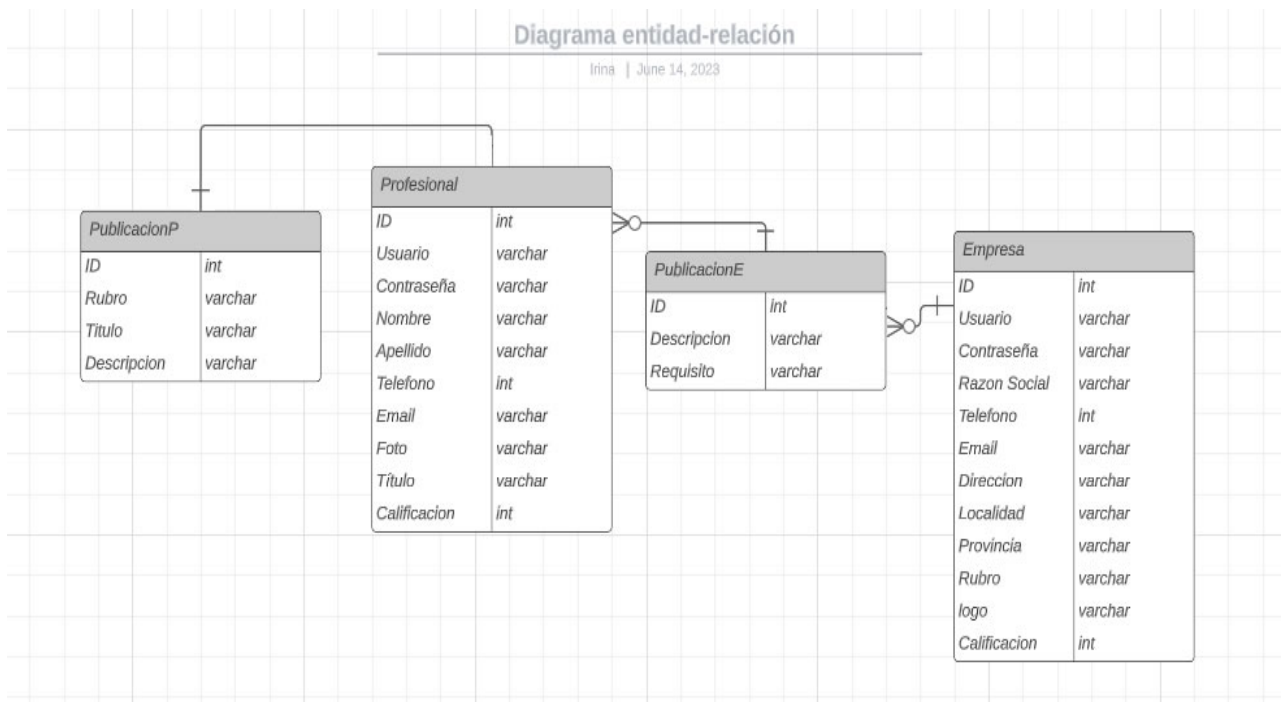
Para que nuestro modelo sea adaptable a diferentes tamaños de pantallas utilizamos:

- Grid ccs: que nos permiten crear rápidamente cuadrículas flexibles de forma prácticamente instantánea, proporcionándonos muchas propiedades CSS que son de gran utilidad.
- Media query: que nos permite definir una regla o conjunto de reglas CSS que se ejecutan para unas condiciones específicas de la pantalla, logrando que el sitio sea adaptable a las diferentes resoluciones de pantalla.

Diagramas de clases:



Modelo entidad relación:



Casos de uso:

Nombre del caso de uso: Registrar Profesional

Actores: Usuario

Descripción: Este caso de uso describe el proceso de registro de un usuario en la aplicación web.

Precondiciones: El usuario no debe estar registrado en la aplicación web.

Postcondiciones: El usuario queda registrado en la base de datos, desde este momento puedo loguearse y comenzar a utilizar el sistema.

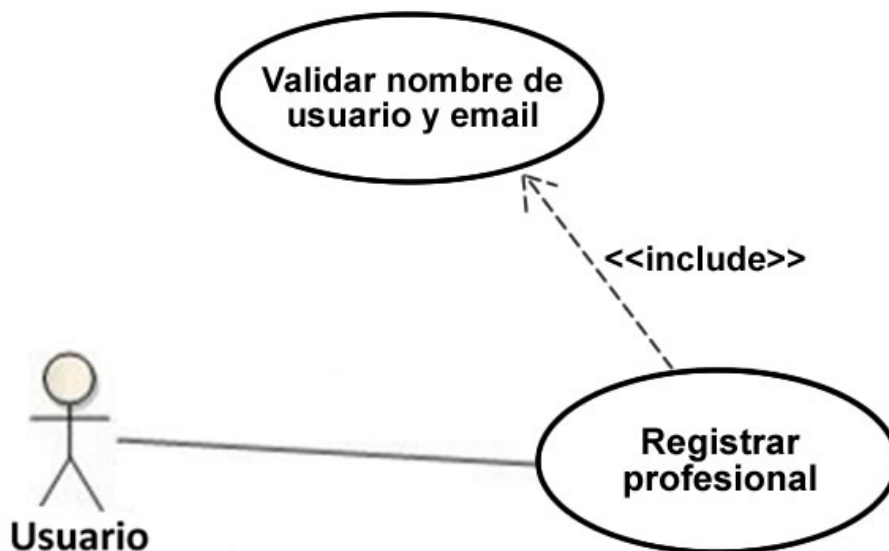
Flujo básico:

1. El usuario llena el formulario para registrarse.
2. El sistema controla que los campos del formulario que son obligatorios estén completos.
3. El usuario presiona el botón de registro.
4. El sistema comprueba que el nombre de usuario y el email no estén registrados en la base de datos.
5. En este caso realiza la inserción del profesional en la base de datos y le informa al usuario que ya está registrado.

Flujo alternativo:

- 4.1. Si en nombre de usuario y el email están registrados en la base de datos, la inserción no se realiza y se le informa al usuario.

Diagrama del caso de uso:



Nombre del caso de uso: Registrar Empresa

Actores: Usuario

Descripción: Este caso de uso describe el proceso de registro de una empresa en la aplicación web.

Precondiciones: El usuario no debe estar registrado en la aplicación web.

Postcondiciones: El usuario queda registrado en la base de datos, desde este momento puedo loguearse y comenzar a utilizar el sistema.

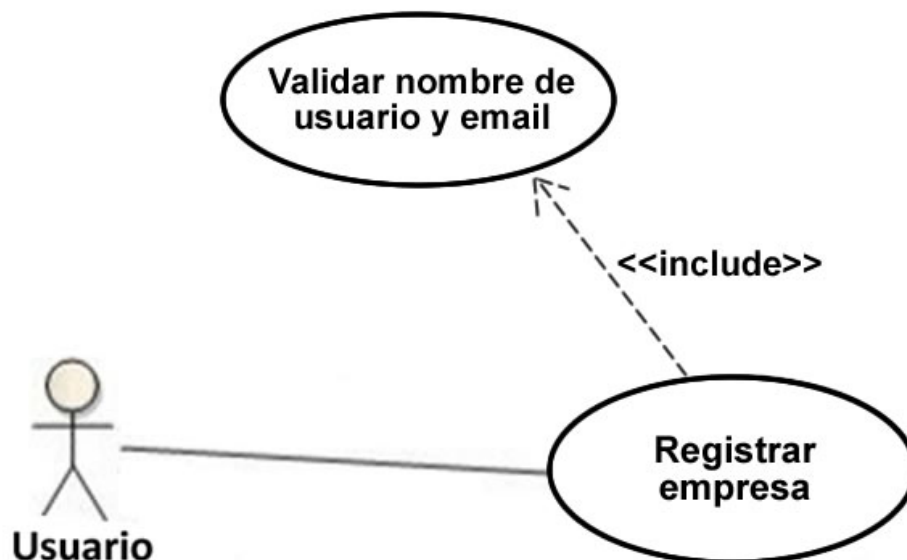
Flujo básico:

6. El usuario llena el formulario para registrarse.
7. El sistema controla que los campos del formulario que son obligatorios estén completos.
8. El usuario presiona el botón de registro.
9. El sistema comprueba que el nombre de usuario y el email no estén registrados en la base de datos.
10. En este caso realiza la inserción de la empresa en la base de datos y le informa al usuario que ya está registrado.

Flujo alternativo:

- 4.2. Si en nombre de usuario y el email están registrados en la base de datos, la inserción no se realiza y se le informa al usuario.

Diagrama del caso de uso:



Nombre del caso de uso: Iniciar Sesión

Actores: profesional, empresa.

Descripción: Este caso de uso describe el proceso de inicio de sesión de un usuario registrado en la aplicación web (puede ser un profesional o una empresa).

Precondiciones: El usuario debe estar registrado en la aplicación web.

Postcondiciones: El usuario inicia sesión en el sistema y accede a las funciones del panel de control.

Flujo básico:

11. El usuario llena el formulario de inicio de sesión.
12. El sistema controla que los campos del formulario que son obligatorios estén completos.
13. El usuario presiona el botón de inicio de sesión.
14. El sistema comprueba que el nombre de usuario y la contraseña estén registrados en la base de datos.
15. En este caso realiza el inicio de sesión y muestra el panel del control al usuario para que comience a utilizar sus funciones.

Flujo alternativo:

- 4.3. Si en nombre de usuario y la contraseña no coinciden con los registrados en la base de datos, el inicio de sesión no se realiza y se le informa al usuario.

Diagrama del caso de uso:

