2º CFGS DAM

9 de mayo de 2025

Iván Núñez Rodríguez

Andrés Felipe González Gómez

David Fernández Fernández

Ángel Antón Gaitán

Sistema de gestión de vacantes de empleo

Reto 9

# Índice

# 1. Introducción y justificación del proyecto

1.1. Breve descripción del producto o servicio.

El "Sistema de Gestión de Vacantes de Empleo" es una plataforma integral diseñada para facilitar la conexión entre empresas que buscan talento y usuarios que buscan oportunidades laborales.

Las funcionalidades clave son:

* Para las empresas la publicación y gestión de vacantes, la recepción y filtrado de solicitudes, y la comunicación con candidatos.
* Para los usuarios la búsqueda de empleo mediante filtros, la postulación a vacantes y el seguimiento de sus solicitudes.
* Para los administradores la gestión de empresas, usuarios y categorías.

El sistema optimiza el proceso de contratación, ahorrando tiempo y recursos a las empresas y mejorando la experiencia de búsqueda de empleo para los usuarios. La plataforma centraliza la información y facilita la comunicación entre ambas partes.

1.2. Visión general de la empresa y el sector seleccionado.

El sector de la búsqueda de empleo y la contratación online está en constante crecimiento, impulsado por la digitalización y la globalización del mercado laboral.

**Público objetivo:**

* Empresas de todos los tamaños y sectores que buscan optimizar su proceso de reclutamiento.
* Usuarios de diferentes perfiles y niveles de experiencia que buscan oportunidades laborales.

**Análisis de la competencia:**

Plataformas como LinkedIn o InfoJobs ofrecen servicios similares, pero este sistema se diferencia por su enfoque integral y su interfaz intuitiva.

**Tendencias del mercado:**

El uso de inteligencia artificial para el filtrado de candidatos, la realidad virtual para entrevistas y la gamificación en los procesos de selección son tendencias emergentes que podrían integrarse en futuras versiones del sistema.

**Justificación:**

La necesidad de un sistema eficiente que conecte empresas y candidatos es evidente en el mercado laboral actual, donde la competencia por el talento es alta y la digitalización es fundamental.

**2. Objetivos generales y específicos.**

2.1. Especificar dos o tres objetivos generales.

**Optimizar el proceso de reclutamiento** para las empresas reduciendo el tiempo y los costes asociados a la contratación de personal.

**Facilitar el acceso a oportunidades laborales** para los usuarios proporcionando una plataforma intuitiva y eficiente para la búsqueda de empleo.

**Desarrollar una plataforma integral y eficiente** para la gestión de vacantes creando un sistema que cubra todas las necesidades de empresas y usuarios en el proceso de contratación.

2.2. Especificar dos o tres objetivos específicos por cada objetivo general.

En relación con la **optimización del proceso de reclutamiento**:

* Implementar una interfaz intuitiva para la publicación y gestión de vacantes desarrollando un panel de control fácil de usar para la creación, edición y eliminación de ofertas de empleo.
* Desarrollar un sistema de filtrado de candidatos eficiente utilizando algoritmos y filtros personalizables para identificar a los candidatos más adecuados para cada vacante.

En relación con **facilitar el acceso a oportunidades laborales** a usuarios:

* Crear una aplicación móvil con una experiencia de usuario óptima diseñando una interfaz atractiva y fácil de navegar para la búsqueda y postulación de empleos.
* Permitir la búsqueda de empleo mediante filtros personalizables implementando filtros por ubicación, sector, tipo de contrato, etc., para que los usuarios encuentren las ofertas que mejor se adapten a sus necesidades.

En relación con el **desarrollo de una plataforma integral y eficiente**:

* Creación de un panel de control para la gestión de los usuarios y las empresas que usen el sistema donde se puedan ver, editar, y eliminar los usuarios y las empresas.
* Desarrollo de un CRUD para la gestión de las categorías de las ofertas de empleo permitiendo al administrador crear, leer, modificar y borrar las diferentes categorías que se usarán en las ofertas de empleo.

**3. Desarrollo.**

**En este apartado se desarrolla el trabajo, desarrollando tantos subapartados como sea necesario.**

**3.1. Modelo de Base de Datos:**

* Diagrama ER (Entidad-Relación) o modelo de datos proporcionado[[1]](#footnote-1).
* Scripts de creación e inserción de datos[[2]](#footnote-2).

**3.2. Desarrollo Backend (API REST):**

En el marco del reto transversal, mi responsabilidad principal ha sido el desarrollo completo del **backend** del sistema de gestión de vacantes y solicitudes de empleo. El backend constituye el núcleo lógico de la aplicación, permitiendo la conexión entre los datos y las funcionalidades de las interfaces Android (usuarios) y Java Swing (empresas y administradores). Su construcción con Spring Boot y MySQL asegura la escalabilidad, seguridad y mantenibilidad del sistema.

**Objetivo general:**

* Desarrollar una API RESTful segura y robusta para gestionar usuarios, empresas, vacantes, categorías, solicitudes y perfiles de usuario.

**Objetivos específicos:**

* Implementar autenticación con JWT y autorización basada en roles (ADMIN, USUARIO, EMPRESA).
* Exponer endpoints seguros que gestionen el ciclo completo de CRUD de todas las entidades.
* Integrar control de acceso por anotaciones @PreAuthorize para garantizar que cada usuario accede solo a lo que le corresponde.
* Separar lógica de negocio (services), acceso a datos (repositories) y representación de datos (DTOs).

El backend ha sido implementado usando **Spring Boot 3.4.4**, con base de datos **MySQL 8**. Se ha estructurado siguiendo una arquitectura limpia de tres capas (controlador-servicio-repositorio), incorporando también DTOs para evitar acoplamiento entre las entidades JPA y la capa de presentación.[[3]](#footnote-3)

**3.3. Desarrollo Frontend (Android):**

* Tecnologías utilizadas (Android Studio, Java o Kotlin): Justificar la elección de la tecnología.
* Descripción de la interfaz de usuario y la experiencia del usuario: Mostrar capturas de pantalla y explicar el flujo de navegación.
* Implementación de la comunicación con la API: Explicar cómo se realizan las peticiones HTTP a la API.
* Funcionalidades de la app: Detallar las funcionalidades de la aplicación, búsqueda de empleo, visualización de ofertas, postulación de solicitudes, etc.

**3.4. Desarrollo Frontend (JavaFX):**

Tecnologías utilizadas:

* Explicar la elección de JavaFX y FXML para el diseño de la interfaz de usuario.
* Detallar las herramientas y bibliotecas utilizadas (Scene Builder, etc.).

Descripción de la interfaz de usuario:

* Mostrar capturas de pantalla de la interfaz diseñada con JavaFX.
* Explicar la estructura de la interfaz, los componentes utilizados y la experiencia del usuario.
* Hacer especial énfasis en la capacidad de JavaFX para crear interfaces gráficas de usuario con una apariencia más actual.

Funcionalidades para la gestión de empresas y administradores:

* Detallar las operaciones CRUD para cada entidad (empresas, administradores, categorías).
* Explicar cómo se implementan estas funcionalidades en JavaFX.

Comunicación con la base de datos:

* Explicar cómo se realizan las consultas a la base de datos desde JavaFX, haciendo uso de JDBC u otros métodos de conexión.
* Mostrar ejemplos de código relevantes.

**3.5. Gestión del Proyecto:**

* Metodología de desarrollo (Scrum, Kanban, etc.): Explicar la metodología utilizada y justificar su elección.
* Herramientas de gestión de proyectos utilizadas: Describir las herramientas utilizadas para la planificación, seguimiento y comunicación del proyecto.
* División de tareas y roles del equipo: Detallar las responsabilidades de cada miembro del equipo.
* Cronogramas y seguimientos del proyecto: Mostrar el cronograma del proyecto y explicar cómo se realizó el seguimiento del progreso.

**3.6. Seguridad:**

* Los roles y permisos se gestionan con Spring Security + JWT, lo que permite a las aplicaciones frontend (Android y FX) consumir la API de forma segura. Se ha integrado control de acceso fino mediante @PreAuthorize en cada endpoint crítico.

**3.7. Despliegue:**

* Se ha proporcionado un .jar ejecutable del backend y una guía de despliegue para facilitar las pruebas. Además, el backend está preparado para integrarse fácilmente con cualquier cliente REST.
* Explicación del despliegue del backend: Explicar cómo se ha desplegado el backend, en que servidor, como se ha configurado, etc.
* Explicación de la compilación y despliegue de las aplicaciones frontales: Explicar cómo se han compilado las aplicaciones frontales, como se han desplegado, etc.

**4. Conclusión general. Limitaciones y prospectivas.**

El desarrollo del backend ha permitido consolidar conocimientos de desarrollo web en Java con Spring Boot, así como aplicar buenas prácticas de seguridad, modularización y diseño REST. Como posibles mejoras futuras, se contempla la integración de un sistema de logs más avanzado, la incorporación de Swagger para documentación de la API, y el despliegue automatizado en un entorno cloud.

4.1. Reflexión sobre el proceso y aprendizaje obtenido.

* + - * Análisis de los desafíos superados y las habilidades desarrolladas: Describir los obstáculos encontrados y cómo se superaron, así como las habilidades técnicas y blandas adquiridas.
      * Evaluación del trabajo en equipo y la colaboración: Reflexionar sobre la dinámica del equipo y la efectividad de la colaboración.

4.2. Valoración del resultado final de la campaña y el evento.

* Evaluación del cumplimiento de los objetivos y la calidad del producto: Analizar si se alcanzaron los objetivos propuestos y evaluar la calidad del sistema desarrollado.
* Identificación de posibles mejoras y futuras implementaciones: Sugerir mejoras y funcionalidades adicionales que podrían implementarse en futuras versiones del sistema.
* Limitaciones encontradas durante el desarrollo: Describir las limitaciones técnicas o de recursos que afectaron el desarrollo del proyecto.
* Posibles mejoras futuras: Describir las posibles mejoras que se le pueden hacer al proyecto.

**5. Bibliografía y fuentes consultadas.**

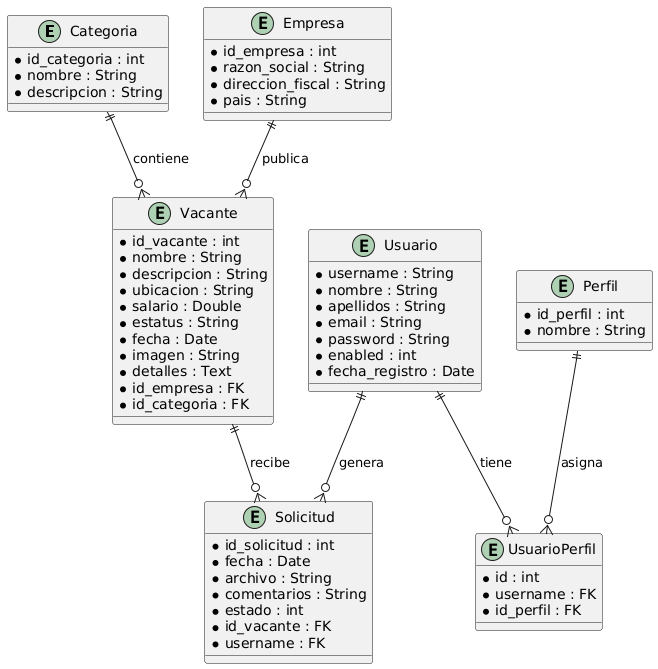
Referencias utilizadas para la investigación y desarrollo del proyecto.

Referencias utilizadas para la investigación y desarrollo del proyecto: Listar todas las fuentes utilizadas, incluyendo libros, artículos, sitios web, etc.

* Citar correctamente las fuentes según las normas APA u otro formato requerido.

## Anexo I

### Diagrama ER



### Script SQL para la creación de la BBDD con sus tablas y relaciones.

-- ESTRUCTURA DEFINITIVA ADAPTADA AL DER FINAL

CREATE DATABASE IF NOT EXISTS reto9\_db;

USE reto9\_db;

CREATE TABLE IF NOT EXISTS categorias (

id\_categoria INT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(100) NOT NULL,

descripcion TEXT

);

CREATE TABLE IF NOT EXISTS empresas (

id\_empresa INT AUTO\_INCREMENT PRIMARY KEY,

razon\_social VARCHAR(100) NOT NULL,

direccion\_fiscal VARCHAR(255),

pais VARCHAR(100)

);

CREATE TABLE IF NOT EXISTS usuarios (

username VARCHAR(50) PRIMARY KEY,

nombre VARCHAR(100),

apellidos VARCHAR(100),

email VARCHAR(100),

password VARCHAR(255),

enabled INT,

fecha\_registro DATE

);

CREATE TABLE IF NOT EXISTS perfiles (

id\_perfil INT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(50)

);

CREATE TABLE IF NOT EXISTS usuarioPerfil (

id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50),

id\_perfil INT,

FOREIGN KEY (username) REFERENCES usuarios(username),

FOREIGN KEY (id\_perfil) REFERENCES perfiles(id\_perfil)

);

CREATE TABLE IF NOT EXISTS vacantes (

id\_vacante INT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(100),

descripcion TEXT,

ubicacion VARCHAR(100),

salario DOUBLE,

estatus VARCHAR(20),

destacado TINYINT(1),

fecha DATE,

imagen VARCHAR(255),

detalles TEXT,

id\_empresa INT,

id\_categoria INT,

FOREIGN KEY (id\_empresa) REFERENCES empresas(id\_empresa),

FOREIGN KEY (id\_categoria) REFERENCES categorias(id\_categoria)

);

CREATE TABLE IF NOT EXISTS solicitudes (

id\_solicitud INT AUTO\_INCREMENT PRIMARY KEY,

fecha DATE,

archivo VARCHAR(255),

comentarios TEXT,

estado INT,

id\_vacante INT,

username VARCHAR(50),

FOREIGN KEY (id\_vacante) REFERENCES vacantes(id\_vacante),

FOREIGN KEY (username) REFERENCES usuarios(username)

);

### Script SQL para poblar las base de datos

-- Perfiles

INSERT INTO perfiles (nombre) VALUES

('ADMIN'),

('EMPRESA'),

('USUARIO');

-- Categorías

INSERT INTO categorias (nombre, descripcion) VALUES ('Desarrollo', 'Quibusdam id repudiandae optio placeat harum.');

INSERT INTO categorias (nombre, descripcion) VALUES ('Diseño', 'Sit modi eaque saepe sapiente.');

INSERT INTO categorias (nombre, descripcion) VALUES ('Ventas', 'Nemo fugit dolores. Non aut laudantium mollitia.');

INSERT INTO categorias (nombre, descripcion) VALUES ('Administración', 'Facere cupiditate assumenda aut eveniet.');

INSERT INTO categorias (nombre, descripcion) VALUES ('Atención al cliente', 'Suscipit molestiae temporibus aspernatur ea nam.');

-- Empresas

INSERT INTO empresas (razon\_social, direccion\_fiscal, pais) VALUES (

'Mancebo-Duran',

'Camino de Mario Doménech 57

Salamanca, 51280',

'Italia'

);

INSERT INTO empresas (razon\_social, direccion\_fiscal, pais) VALUES (

'Bauzà LLC',

'Urbanización de Arturo Hoz 70

Teruel, 77030',

'Kenya'

);

INSERT INTO empresas (razon\_social, direccion\_fiscal, pais) VALUES (

'Montaña-Cardona',

'Via Javier Gual 655

Lugo, 25430',

'Argelia'

);

-- Usuarios

INSERT INTO usuarios (username, nombre, apellidos, email, password, enabled, fecha\_registro)

VALUES ('user1', 'Berto', 'Maza', 'sandaliosevillano@mari.es', '$2a$10$g3n3r4d0C0ntr4', 1, '2025-02-28');

INSERT INTO usuarios (username, nombre, apellidos, email, password, enabled, fecha\_registro)

VALUES ('user2', 'Augusto', 'Blanco', 'emiliana61@arjona-caceres.es', '$2a$10$g3n3r4d0C0ntr4', 1, '2025-03-10');

INSERT INTO usuarios (username, nombre, apellidos, email, password, enabled, fecha\_registro)

VALUES ('user3', 'Flavio', 'Gordillo', 'simon36@gmail.com', '$2a$10$g3n3r4d0C0ntr4', 1, '2025-01-21');

INSERT INTO usuarios (username, nombre, apellidos, email, password, enabled, fecha\_registro)

VALUES ('user4', 'Apolinar', 'Rodríguez', 'poncio38@gmail.com', '$2a$10$g3n3r4d0C0ntr4', 1, '2025-01-01');

INSERT INTO usuarios (username, nombre, apellidos, email, password, enabled, fecha\_registro)

VALUES ('user5', 'Florencia', 'Manso', 'cirinorivera@bastida.es', '$2a$10$g3n3r4d0C0ntr4', 1, '2025-03-27');

-- UsuarioPerfil

INSERT INTO usuarioPerfil (username, id\_perfil) VALUES ('user1', 3);

INSERT INTO usuarioPerfil (username, id\_perfil) VALUES ('user2', 3);

INSERT INTO usuarioPerfil (username, id\_perfil) VALUES ('user3', 3);

INSERT INTO usuarioPerfil (username, id\_perfil) VALUES ('user4', 3);

INSERT INTO usuarioPerfil (username, id\_perfil) VALUES ('user5', 3);

-- Vacantes

INSERT INTO vacantes (nombre, descripcion, ubicacion, salario, estatus, fecha, imagen, detalles, id\_empresa, id\_categoria)

VALUES (

'Media planner', 'Nisi incidunt ipsum. Enim accusamus quidem ipsum expedita eveniet autem.', 'Córdoba',

31211, 'CREADA', '2025-03-24',

'https://placeimg.com/73/195/any', 'Dolor eius placeat porro autem minima rem quis.',

3, 1

);

INSERT INTO vacantes (nombre, descripcion, ubicacion, salario, estatus, fecha, imagen, detalles, id\_empresa, id\_categoria)

VALUES (

'Best boy', 'Voluptatibus maxime commodi totam magnam. Hic fugit optio expedita asperiores.', 'Ourense',

23088, 'CREADA', '2025-01-13',

'https://www.lorempixel.com/838/945', 'Distinctio iste nisi dignissimos. Corrupti expedita nisi a.',

3, 4

);

INSERT INTO vacantes (nombre, descripcion, ubicacion, salario, estatus, fecha, imagen, detalles, id\_empresa, id\_categoria)

VALUES (

'Scientist, research (life sciences)', 'Omnis consequatur libero ut architecto. Atque quis aut.', 'Granada',

28059, 'CREADA', '2025-01-24',

'https://www.lorempixel.com/655/242', 'Deserunt modi debitis sapiente.',

1, 4

);

INSERT INTO vacantes (nombre, descripcion, ubicacion, salario, estatus, fecha, imagen, detalles, id\_empresa, id\_categoria)

VALUES (

'Accountant, chartered certified', 'Suscipit magni hic nihil nulla. Animi dolorum doloremque mollitia alias impedit.', 'Asturias',

31899, 'CREADA', '2025-02-07',

'https://placeimg.com/379/275/any', 'Doloremque ipsam aperiam esse animi.',

1, 2

);

INSERT INTO vacantes (nombre, descripcion, ubicacion, salario, estatus, fecha, imagen, detalles, id\_empresa, id\_categoria)

VALUES (

'Copy', 'Veniam adipisci incidunt sapiente nobis. Eius id dignissimos assumenda eum.', 'Salamanca',

35434, 'CREADA', '2025-02-06',

'https://dummyimage.com/453x46', 'Dicta minima illum nihil doloribus perspiciatis.',

2, 3

);

INSERT INTO vacantes (nombre, descripcion, ubicacion, salario, estatus, fecha, imagen, detalles, id\_empresa, id\_categoria)

VALUES (

'Sports coach', 'Assumenda repudiandae consequatur nobis. Dolorum hic enim illum est excepturi.', 'Guipúzcoa',

32313, 'CREADA', '2025-03-30',

'https://dummyimage.com/770x209', 'Dolor laborum totam iure.',

2, 2

);

-- Solicitudes

INSERT INTO solicitudes (fecha, archivo, comentarios, estado, id\_vacante, username)

VALUES ('2025-01-23', 'user1\_cv.pdf', 'Accusamus at nihil ipsam.', 0, 3, 'user1');

INSERT INTO solicitudes (fecha, archivo, comentarios, estado, id\_vacante, username)

VALUES ('2025-01-29', 'user3\_cv.pdf', 'Error corporis unde cumque.', 0, 3, 'user3');

INSERT INTO solicitudes (fecha, archivo, comentarios, estado, id\_vacante, username)

VALUES ('2025-01-15', 'user1\_cv.pdf', 'Ex laudantium reiciendis eum quasi laboriosam cum.', 1, 6, 'user1');

INSERT INTO solicitudes (fecha, archivo, comentarios, estado, id\_vacante, username)

VALUES ('2025-01-06', 'user3\_cv.pdf', 'Ipsam similique in repudiandae ad ratione.', 1, 4, 'user3');

INSERT INTO solicitudes (fecha, archivo, comentarios, estado, id\_vacante, username)

VALUES ('2025-03-21', 'user5\_cv.pdf', 'Consequatur ipsum neque voluptatibus.', 1, 6, 'user5');

INSERT INTO solicitudes (fecha, archivo, comentarios, estado, id\_vacante, username)

VALUES ('2025-01-25', 'user2\_cv.pdf', 'Assumenda ipsa maxime vel nemo laboriosam.', 0, 5, 'user2');

INSERT INTO solicitudes (fecha, archivo, comentarios, estado, id\_vacante, username)

VALUES ('2025-03-13', 'user4\_cv.pdf', 'Libero voluptas nesciunt ad illum.', 1, 6, 'user4');

INSERT INTO solicitudes (fecha, archivo, comentarios, estado, id\_vacante, username)

VALUES ('2025-02-01', 'user1\_cv.pdf', 'Natus id earum vero numquam.', 0, 4, 'user1');

## Anexo II

### Documentación Técnica del Backend

**Tecnologías Utilizadas**

* Lenguaje: Java 17
* Framework: Spring Boot 3.4.4
* ORM: Spring Data JPA
* Base de Datos: MySQL 8
* Autenticación y Autorización: JWT (JSON Web Token) + Spring Security
* Control de Acceso: Roles gestionados con @PreAuthorize
* Gestor de dependencias: Maven

**Arquitectura General**

El backend está estructurado en capas siguiendo la arquitectura en capas habitual de Spring Boot:

* Controladores: Definen los endpoints REST
* Servicios: Contienen la lógica de negocio
* Repositorios: Interactúan con la base de datos
* DTOs: Separan las entidades del modelo de los datos expuestos
* Modelos: Representan las entidades de la base de datos

**Entidades Principales**

* Usuario: contiene datos personales y credenciales.
* Empresa: información de empresas registradas.
* Vacante: contiene información de ofertas laborales.
* Solicitud: solicitudes realizadas por usuarios a vacantes.
* Perfil: define los roles (ADMIN, USUARIO, EMPRESA).
* UsuarioPerfil: asocia usuarios con uno o varios perfiles.
* Categoria: categorías de empleo.

**Endpoints Destacados (ejemplos)**

**Autenticación**

* POST /api/auth/login: login y generación de token JWT
* POST /api/auth/register: registro de nuevo usuario

**Vacantes**

* GET /api/vacantes: (ADMIN) listar todas las vacantes
* GET /api/vacantes/creadas: (USUARIO) ver vacantes activas
* POST /api/vacantes: (EMPRESA) crear nueva vacante

**Solicitudes**

* POST /api/solicitudes: (USUARIO) postular a vacante
* GET /api/solicitudes/usuario/{username}: (USUARIO) ver sus solicitudes
* PUT /api/solicitudes/{id}/adjudicar: (EMPRESA) adjudicar una vacante

**Seguridad con Roles**

Se usa @PreAuthorize para limitar el acceso según el rol JWT:

@PreAuthorize("hasRole('ADMIN')")

@PreAuthorize("hasRole('USUARIO')")

@PreAuthorize("hasRole('EMPRESA')")

**Autenticación JWT**

* Se genera un JWT en el login
* El filtro JwtFilter intercepta las peticiones y valida el token
* El SecurityContextHolder se configura con los roles del usuario
* Análisis.
* Cuestionarios diseñados (formato Google Forms o similar).
* Materiales publicitarios en alta resolución.
* Cronogramas detallados y otros documentos de apoyo.

**Análisis DAFO:**

* Este análisis proporcionará una visión estratégica del proyecto, identificando sus Fortalezas, Debilidades, Oportunidades y Amenazas.
* **Fortalezas:**
  + Enumerar las ventajas internas del sistema, como su funcionalidad integral, interfaz intuitiva, tecnología de punta utilizada (Spring Boot, Android Studio), etc.
  + Destacar las habilidades y experiencia del equipo de desarrollo.
* **Debilidades:**
  + Identificar las limitaciones internas del proyecto, como posibles restricciones de tiempo, recursos limitados, o falta de experiencia en áreas específicas.
  + Ser honesto sobre las áreas que necesitan mejora.
* **Oportunidades:**
  + Explorar las tendencias del mercado y las posibles áreas de crecimiento, como la integración de inteligencia artificial, la expansión a nuevos mercados geográficos, o la colaboración con empresas asociadas.
  + Identificar nichos de mercado no atendidos.
* **Amenazas:**
  + Analizar los riesgos externos que podrían afectar el proyecto, como la competencia de otras plataformas de empleo, cambios en la legislación laboral, o fluctuaciones económicas.
  + Prever posibles problemas de seguridad y de privacidad.

**Cuestionarios diseñados (Google Forms o similar):**

* Incluir los cuestionarios utilizados para recopilar información de usuarios potenciales o empresas.
* Proporcionar un resumen de los resultados obtenidos, destacando las tendencias y conclusiones relevantes.
* Explicar cómo se utilizaron los resultados para tomar decisiones de diseño o desarrollo. Por ejemplo, si los usuarios expresaron una preferencia por una determinada funcionalidad, explicar cómo se implementó.
* Incluir gráficas o tablas que muestren los resultados de forma visual.

**Materiales publicitarios en alta resolución:**

* Incluir los diseños de logotipos, banners, folletos u otros materiales publicitarios creados para promocionar el sistema.
* Explicar la estrategia de marketing y publicidad detrás de estos materiales.
* Explicar la finalidad de cada uno de los materiales publicitarios. Por ejemplo, el logotipo representa la identidad de la marca, mientras que los banners se utilizan para atraer tráfico al sitio web.
* Incluir una explicación de los colores y la tipografía usados, y porque se han escogido.

**Cronogramas detallados y otros documentos de apoyo:**

* Incluir el cronograma del proyecto (diagrama de Gantt o similar), mostrando las fases, tareas y plazos.
* Proporcionar otros documentos relevantes, como diagramas de flujo, casos de uso, especificaciones técnicas o manuales de usuario.
* Incluir cualquier documento que pueda ayudar a comprender el desarrollo del proyecto.
* Incluir la hoja de control individual de cada uno de los participantes del proyecto, donde cada miembro del grupo evalua su participación y la de sus compañeros.

1. Ver Anexo I. [↑](#footnote-ref-1)
2. Ver Anexo I. [↑](#footnote-ref-2)
3. Ver Anexo I. Documentación Técnica del Backend. [↑](#footnote-ref-3)