



JOBS X DEVS:: PLATAFORMA DE EMPLEO PARA DESARROLLADORES

Memoria del Proyecto

Autor: Jesús David Ramos Cardona

Tutores: Luis Garcia, Fortunato Yekue, Manuel Sanchez

Valencia, 30 de Mayo de 2025

Agredecimientos

Quiero agradecer en primer lugar a Dios, por darme la fuerza y la sabiduría para completar este proyecto.

A mi esposa, por su esfuerzo constante, por estar a mi lado en los momentos difíciles y por soportar con paciencia estos años de formación. A mi familia, por su apoyo incondicional durante todo el proceso.

También agradezco sinceramente al centro de estudios Grupo Studio, y especialmente a los profesores Luis García, Fortu (mi tutor) y demás docentes que han sido parte importante del aprendizaje y del acompañamiento.

Finalmente, a mis compañeros de clase, por su ayuda, por compartir sus conocimientos y por formar parte de este camino profesional.

Tabla de contenido

1. Presentación del proyecto	4
1.1 Introducción	4
1.2 Objetivos	4
1.3 Tecnologías y herramientas empleadas	4
1.4 Planificación temporal	5
2. Estudio de viabilidad	5
2.1 Establecimiento del alcance del sistema	5
2.2 Estudio de la situación actual	6
2.2.1 La aplicación	6
2.3 Definición de los requisitos del sistema	6
2.4 Estudio de la solución adoptada	7
2.5 Costes	7
2.6 Análisis de riesgos	8
3. Análisis	8
3.1 Requisitos funcionales	8
3.2 Requisitos no funcionales	8
3.3 Definición de interfaces de usuario	9
4. Diseño del sistema	15
4.1 Arquitectura	15
4.1.1 Identificación de subsistemas	16
4.2 Diseño de la base de datos	16
4.3 Manejo de errores	18
4.3.1 Manejo de errores en el backend	18
4.3.2 Manejo de errores en el frontend	18
4.4 Elección de alternativas de componentes y licencias más adecuadas	19
4.5 Especificaciones de desarrollo	19
4.5.1 Necesidades de equipos	19
4.5.2 Necesidades software	20
4.6 Especificación del plan de pruebas	21
5. Desarrollo	21
5.1 Desarrollo en el lado del cliente	21
5.2 Desarrollo del lado del servidor	22
6. Conclusiones	23
6.1 Nivel de satisfacción general	23
6.2 Propuestas de mejora	23
7. Bibliografía	24
8. Enlaces	24

1. Presentación del proyecto

1.1 Introducción

Este proyecto consiste en el desarrollo de Jobs x Devs, una plataforma web especializada en conectar a desarrolladores con ofertas de empleo que se ajusten a su stack tecnológico. El campo de actuación es el ámbito del empleo tecnológico, con un enfoque particular en resolver uno de los problemas más habituales en las bolsas de empleo genéricas: la falta de precisión y detalle sobre las tecnologías que realmente se utilizan en los puestos ofertados.

Jobs x Devs pretende aportar una solución clara, intuitiva y adaptada tanto para desarrolladores como para empresas, optimizando la búsqueda de talento y mejorando la visibilidad de los perfiles técnicos.

1.2 Objetivos

El objetivo principal de este proyecto es crear una aplicación web centrada exclusivamente en el empleo para desarrolladores, optimizando la coincidencia entre tecnologías requeridas y tecnologías dominadas.

Objetivos específicos:

- Permitir a las empresas publicar ofertas especificando claramente su stack tecnológico.
- Facilitar a los desarrolladores encontrar empleos afines a sus habilidades.
- Diseñar un sistema de registro y autenticación diferenciado por tipo de usuario.
- Crear una interfaz limpia, moderna y accesible.

1.3 Tecnologías y herramientas empleadas

A continuación se presenta un resumen de las herramientas y componentes software utilizados para este proyecto:

- Sistema operativo: macOS
- Frontend: HTML5, CSS3, JavaScript, Vue 3 (Composition API)
- Backend: Node.js con Express
- Base de datos: MySQL
- Control de versiones: Git + GitHub
- Gestor de paquetes: npm
- Entorno de desarrollo: Visual Studio Code
- Diseño UI/UX: Figma
- Gestión del proyecto y documentación: Notion y Google Docs

1.4 Planificación temporal

El coste temporal estimado del proyecto es de 85 horas, distribuidas en las diferentes fases necesarias para su desarrollo. La finalización del proyecto está prevista para la primera semana de junio de 2025.

FASE	FECHA INICIO	FECHA FIN	HORAS ESTIMADAS
Estudio de viabilidad	22/04/2025	25/04/2025	5 h
Análisis	26/04/2025	01/05/2025	10 h
Diseño	02/05/2025	08/05/2025	15 h
Desarrollo	09/05/2025	31/05/2025	45 h
Pruebas y mejoras	01/06/2025	07/06/2025	10 h
Total	-	-	85 h

2. Estudio de viabilidad

2.1 Establecimiento del alcance del sistema

Al tratarse de una plataforma web construida con tecnologías libres y ampliamente soportadas, el coste del proyecto se mantiene bajo, especialmente al no depender de licencias comerciales. La aplicación será accesible desde cualquier navegador moderno, sin necesidad de instalación en el dispositivo del usuario.

El sistema está orientado tanto a desarrolladores como a empresas tecnológicas. Permitirá la creación de perfiles personalizados, publicación de ofertas de empleo técnicas y búsqueda avanzada. Se prevé que la solución sea escalable y fácilmente ampliable.

2.2 Estudio de la situación actual

Actualmente no existe ninguna solución que resuelva específicamente el problema que me encontré al buscar empleo como desarrollador: muchas ofertas no especifican bien las tecnologías usadas, o están publicadas en portales genéricos donde se mezclan todo tipo de perfiles. Esto genera pérdida de tiempo tanto para candidatos como para empresas. Por eso, *Jobs x Devs* nace como una plataforma diferente, creada desde cero, pensada específicamente para este sector.

2.2.1 La aplicación

Jobs x Devs es una plataforma web que conecta desarrolladores con empresas tecnológicas, permitiendo que ambas partes encuentren lo que realmente buscan. Las empresas pueden crear una cuenta, configurar su perfil y publicar ofertas donde detallan el stack tecnológico exacto de sus proyectos.

Por su parte, los desarrolladores pueden registrarse, completar su perfil profesional con tecnologías dominadas y experiencias previas, y aplicar solo a ofertas compatibles con su stack. La plataforma ofrece filtros por tipo de contrato, modalidad (remoto, híbrido, presencial), o localización, lo que mejora la precisión de las búsquedas y ahorra tiempo.

Todo el sistema está diseñado para ser rápido, claro y fácil de usar, tanto para quienes publican como para quienes buscan. Además, incluye una interfaz adaptada según el tipo de usuario (empresa o desarrollador), lo cual mejora aún más la experiencia.

2.3 Definición de los requisitos del sistema

Técnicos:

- La aplicación debe ser accesible desde cualquier navegador moderno.
- El almacenamiento de la información se realizará en una base de datos relacional (MySQL).
- El sistema debe seguir estándares web actuales (HTML5, CSS3, JavaScript).
- La plataforma debe seguir buenas prácticas de seguridad, como el cifrado de contraseñas y validación de datos.

Operativos:

- Permitir a los usuarios acceder a la plataforma sin necesidad de instalar nada.
- El mantenimiento debe poder realizarse fácilmente gracias a un código claro y documentado.
- La interfaz debe ser rápida, clara y adaptable a distintos dispositivos (responsive).

Legales:

- Toda la información personal se tratará cumpliendo con la normativa vigente sobre protección de datos (RGPD).
- Las tecnologías empleadas deben utilizar licencias libres o abiertas.

Económicos:

- Se debe evitar el uso de herramientas que requieran pago por licencia.
- El sistema debe ser sostenible sin coste económico adicional gracias al uso de servicios gratuitos para el desarrollo y despliegue inicial.

2.4 Estudio de la solución adoptada

Se ha optado por desarrollar la aplicación a medida con tecnologías web modernas y libres. No existe actualmente una solución en el mercado que cumpla los requisitos específicos de Jobs x Devs, como la diferenciación clara entre usuarios empresa y desarrollador.

2.5 Costes

El desarrollo de Jobs x Devs se ha realizado íntegramente sin costes económicos gracias al uso de tecnologías y plataformas gratuitas. A continuación se detallan los costes aproximados si se contemplara su uso en producción:

- Dominio y alojamiento web (anual): 50 €
- Base de datos gestionada en producción: 0 € con Railway plan gratuito (hasta ciertos límites)
- Frontend (Vercel): 0 € en plan gratuito
- Licencias de software: 0 € (todas las herramientas utilizadas son de código abierto o gratuitas)
- Diseño con Figma: versión gratuita

Coste total estimado: 0 € durante el desarrollo y entorno local. En un entorno profesional se podrían estimar unos 50–100 € anuales por costes de producción básicos.

2.6 Análisis de riesgos

Durante el desarrollo se han considerado los siguientes riesgos potenciales:

- Pérdida de información: mitigado con uso de Git y control de versiones en GitHub.
- Problemas de seguridad: mitigados mediante validaciones de formularios, cifrado de contraseñas, autenticación con JWT y control de roles.
- Errores en producción: evitado con pruebas manuales y separación clara de roles y rutas.
- Dependencia de servicios gratuitos: Vercel y Railway tienen limitaciones, por lo que en caso de crecimiento real se debería considerar migrar a servicios profesionales.

Todos los riesgos se han abordado de manera preventiva, asegurando la estabilidad del sistema tanto en desarrollo como en su futura puesta en marcha.

3. Análisis

3.1 Requisitos funcionales

- Registro y login de desarrolladores y empresas con formularios independientes.
- Redirección automática a la vista correspondiente según el tipo de usuario.
- Panel de control con información personal editable.
- Gestión de ofertas: creación, edición, visualización y eliminación.
- Postulación a ofertas por parte de los desarrolladores.
- Listado de postulaciones para ambos roles.
- Vista para administrador con control total sobre usuarios y ofertas.

3.2 Requisitos no funcionales

- Interfaz responsive adaptada a dispositivos móviles.
- Navegación protegida mediante rutas seguras (JWT).
- Buen rendimiento en carga inicial y navegación entre vistas.
- Validación de formularios tanto en frontend como backend.
- Diseño accesible y uso de iconografía clara.
- Arquitectura modular y mantenimiento sencillo.

3.3 Definición de interfaces de usuario

La aplicación web Jobs x Devs cuenta con un diseño responsive y una interfaz dividida en roles, donde cada tipo de usuario accede a vistas y funcionalidades específicas. Todas las rutas privadas están protegidas, y los usuarios deben iniciar sesión para acceder a su contenido. A continuación, se describen las principales vistas:

3.3.1 Home

Es la pantalla de inicio de la plataforma. Desde aquí se puede acceder a los formularios de login y registro, tanto para desarrolladores como para empresas.

Elementos clave:

- Logo y nombre de la aplicación
- Accesos a login y registro por rol
- Vista informativa y navegación clara

3.3.2 Registro y autenticación

Cada rol (desarrollador o empresa) tiene su propio formulario de registro y login. Una vez registrado, el sistema muestra una alerta de confirmación y redirige automáticamente al login correspondiente. Tras autenticarse, el usuario es enviado a su perfil.

Elementos clave:

- Formularios separados para empresa y desarrollador
- Alertas de confirmación
- Redirección tras registro o login

3.3.3 Perfil de desarrollador

El desarrollador accede a un panel privado donde puede:

- Ver y editar su perfil
- Cambiar su contraseña
- Acceder al listado de ofertas disponibles en la ruta /offers
- Aplicar a ofertas
- Visualizar las ofertas a las que ya ha aplicado y eliminarlas

3.3.4 Perfil de empresa

Las empresas acceden a su panel desde el cual pueden:

- Ver y editar sus datos
- Cambiar su contraseña
- Publicar nuevas ofertas
- Ver un listado de sus ofertas (editar o eliminar)
- Consultar los desarrolladores que han aplicado a sus ofertas

3.3.5 Perfil de administrador

El administrador inicia sesión desde un login exclusivo (no público) ya que este usuario se crea directamente en base de datos. Desde su panel tiene acceso a:

- Listado de todas las ofertas publicadas por empresas
- Listado de todos los desarrolladores registrados
- Posibilidad de editar o eliminar registros

3.3.6 Ruta /offers

Ruta accesible públicamente donde se listan todas las ofertas disponibles. Si el usuario está autenticado como desarrollador, puede aplicar directamente desde esta vista. Si no ha iniciado sesión, podrá ver la información general de la oferta, pero no postularse.

3.3.7 Ruta /developers

Ruta pública desde la que se puede visualizar un listado de desarrolladores y acceder a su perfil público para contactar con ellos.

3.3.8 Página 404

Pantalla informativa que se muestra cuando el usuario accede a una ruta inexistente dentro del sistema.

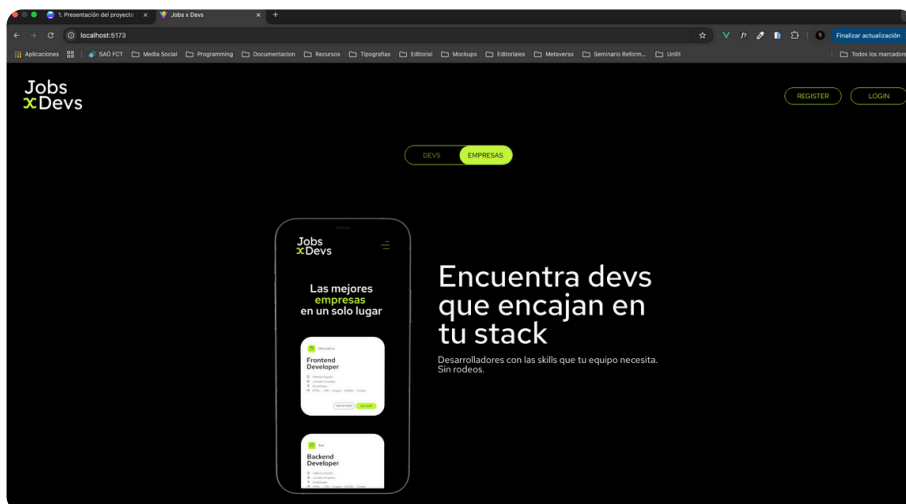
Elementos clave:

- Mensaje de error
- Enlace para volver al inicio

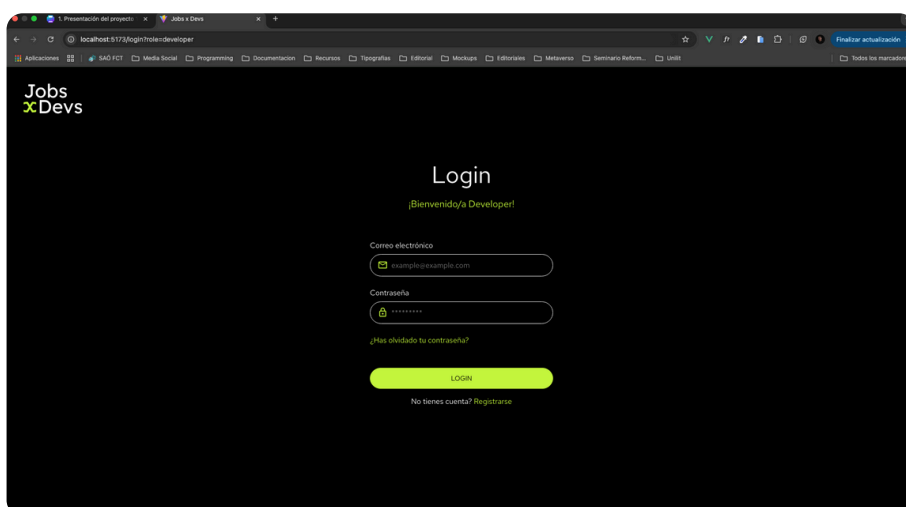
3.3.9 Capturas de las interfaces



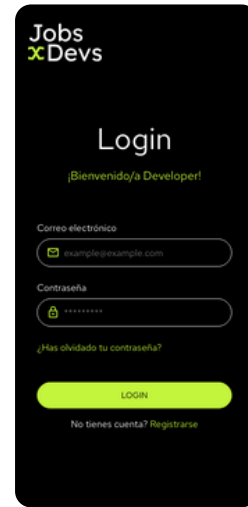
*Vista de Home como Developer

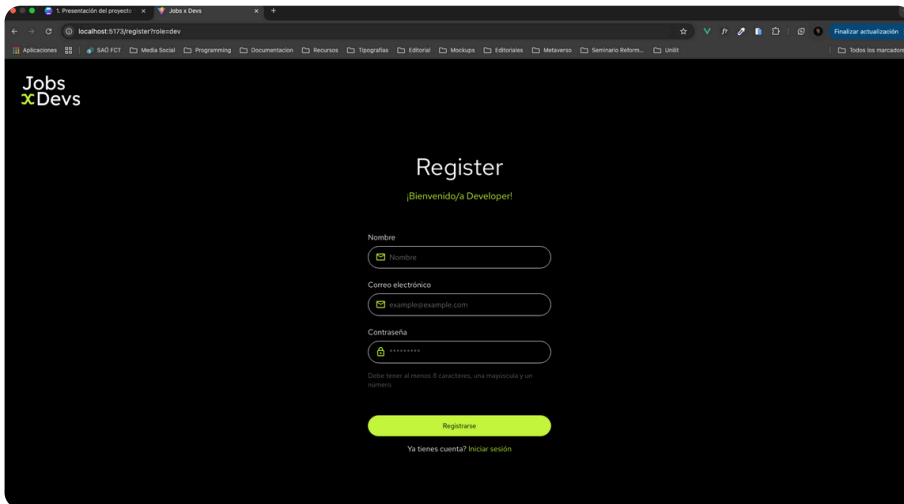


*Vista de Home como Company

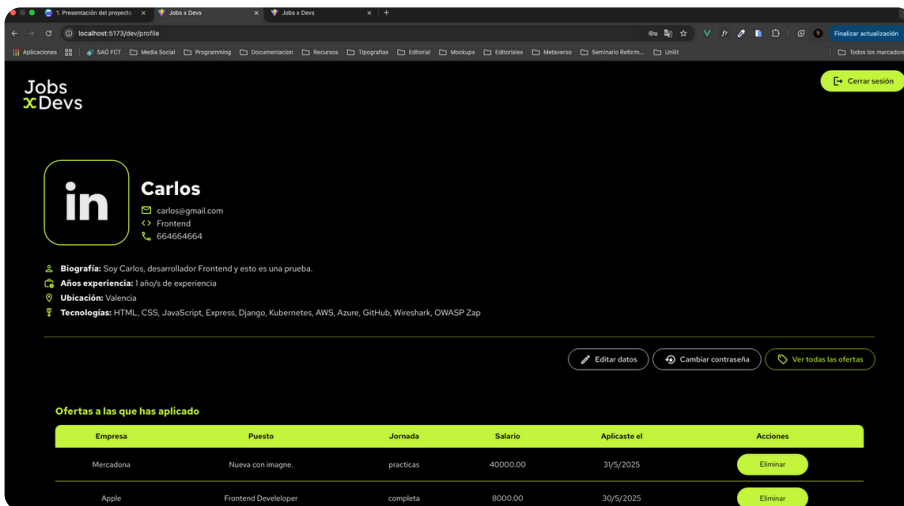
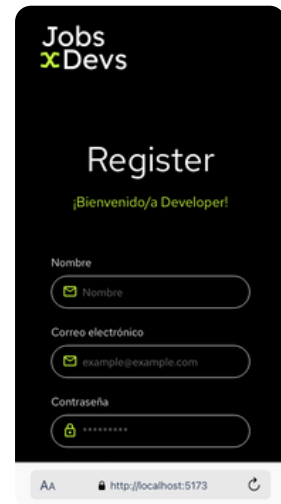


*Vista de Login

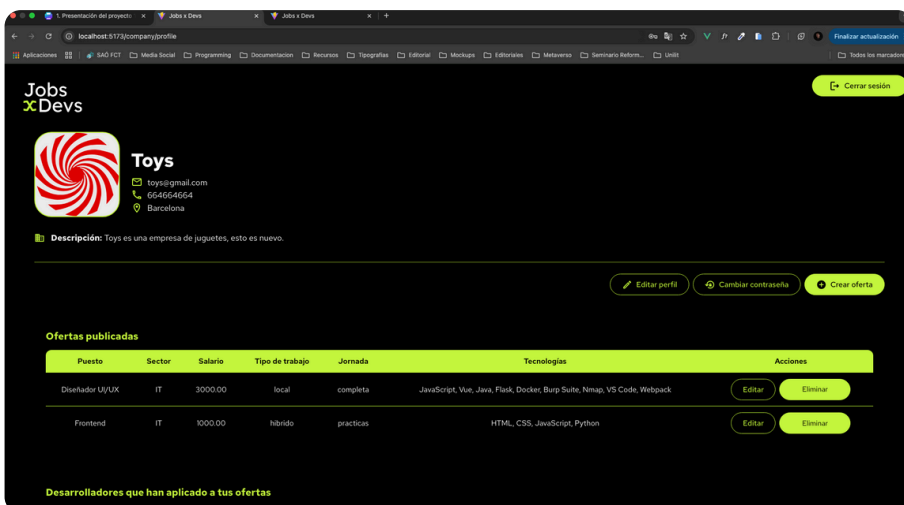




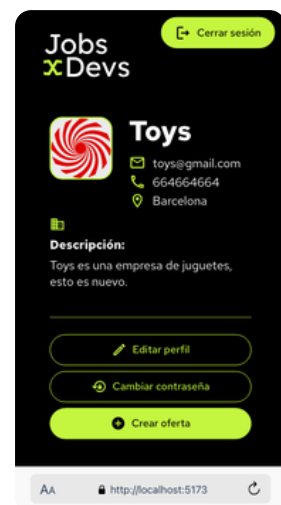
*Vista de Register

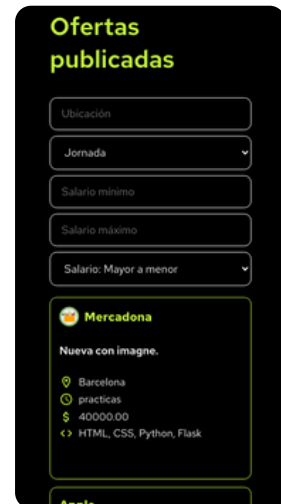
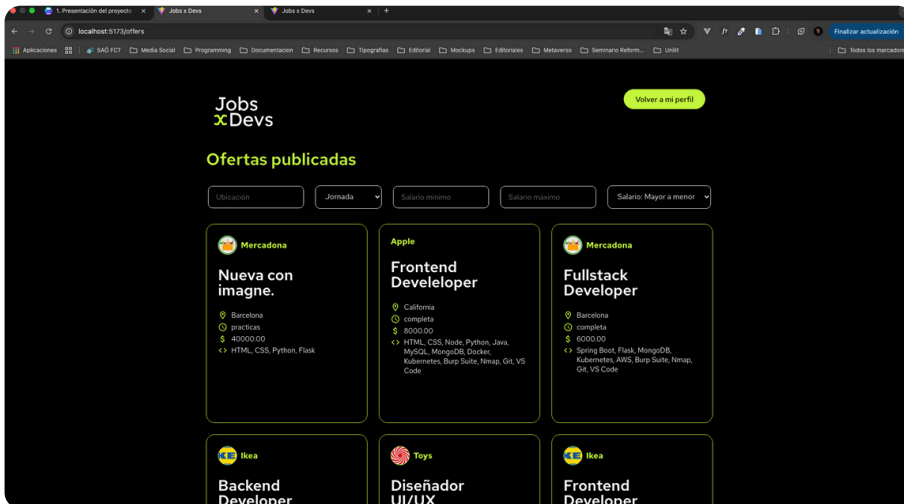


*Vista de Perfil como Developer

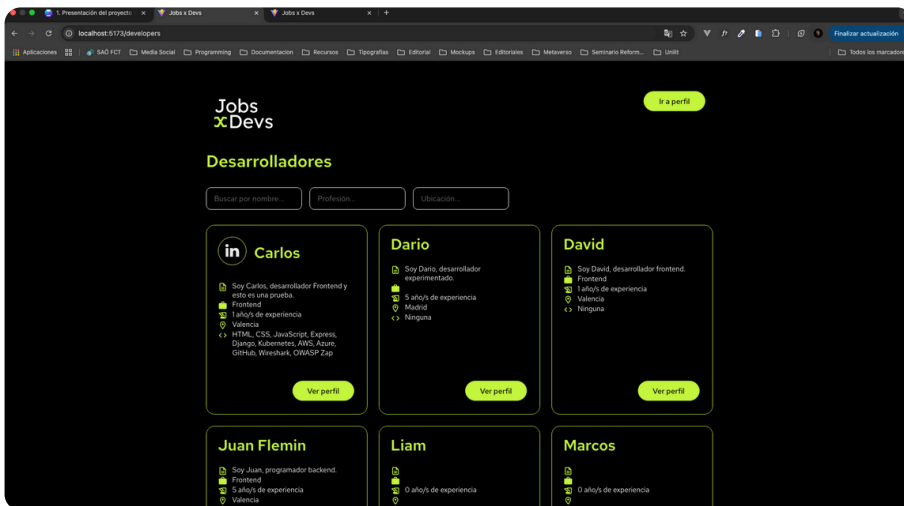


*Vista de Perfil como Company

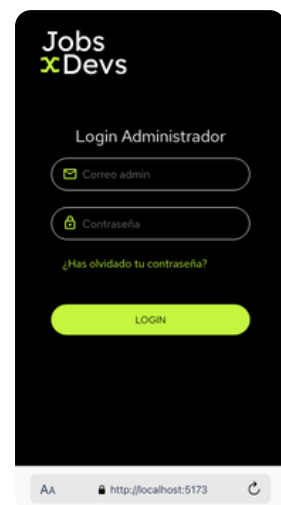
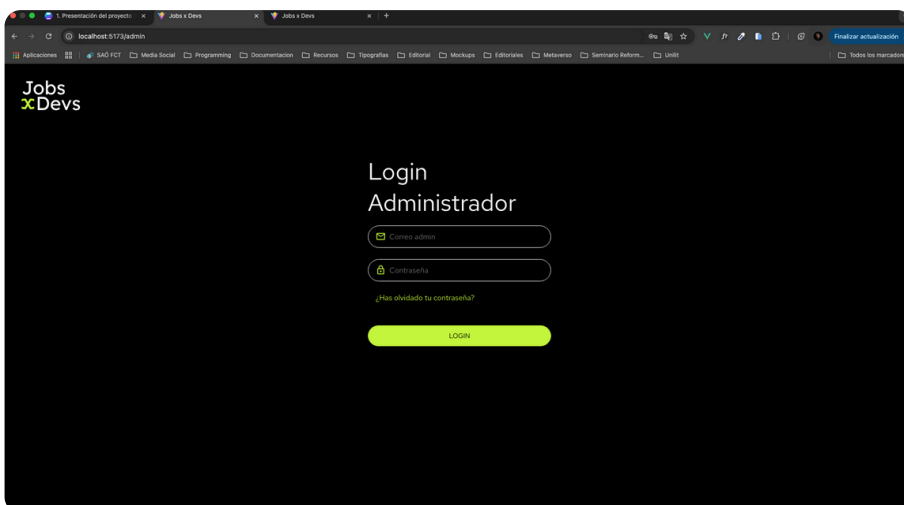




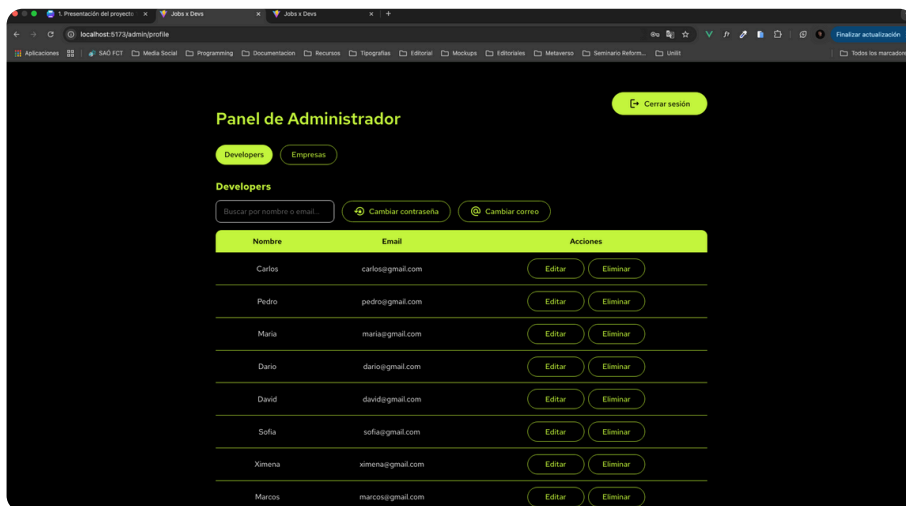
*Vista de Offers



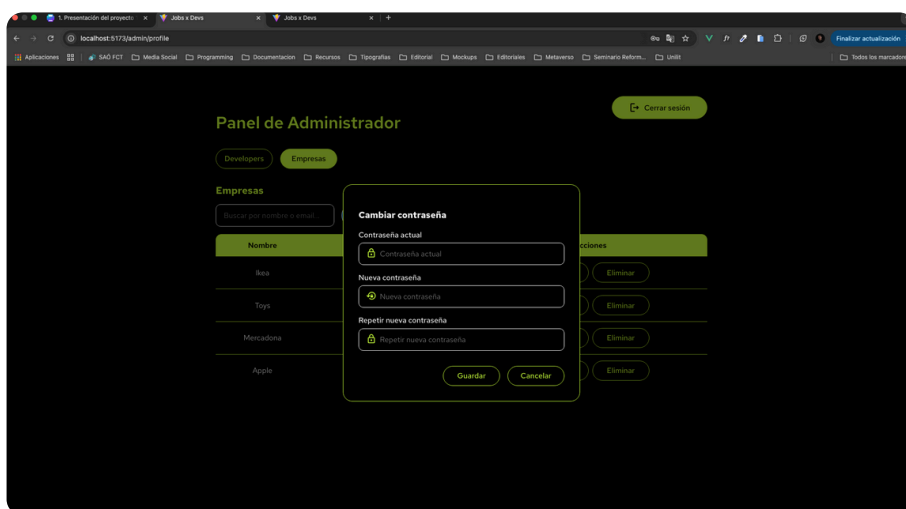
*Vista de Developers



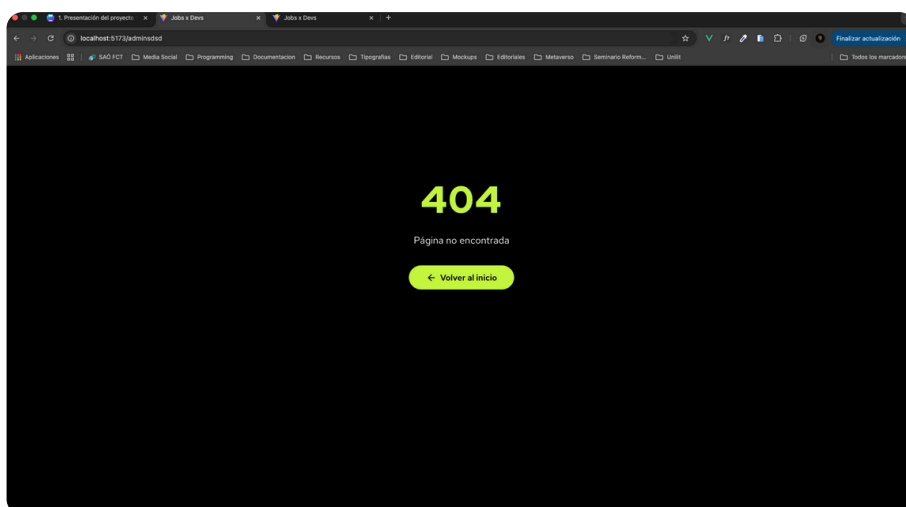
*Vista de Login Administrador



*Vista de Perfil Administrador

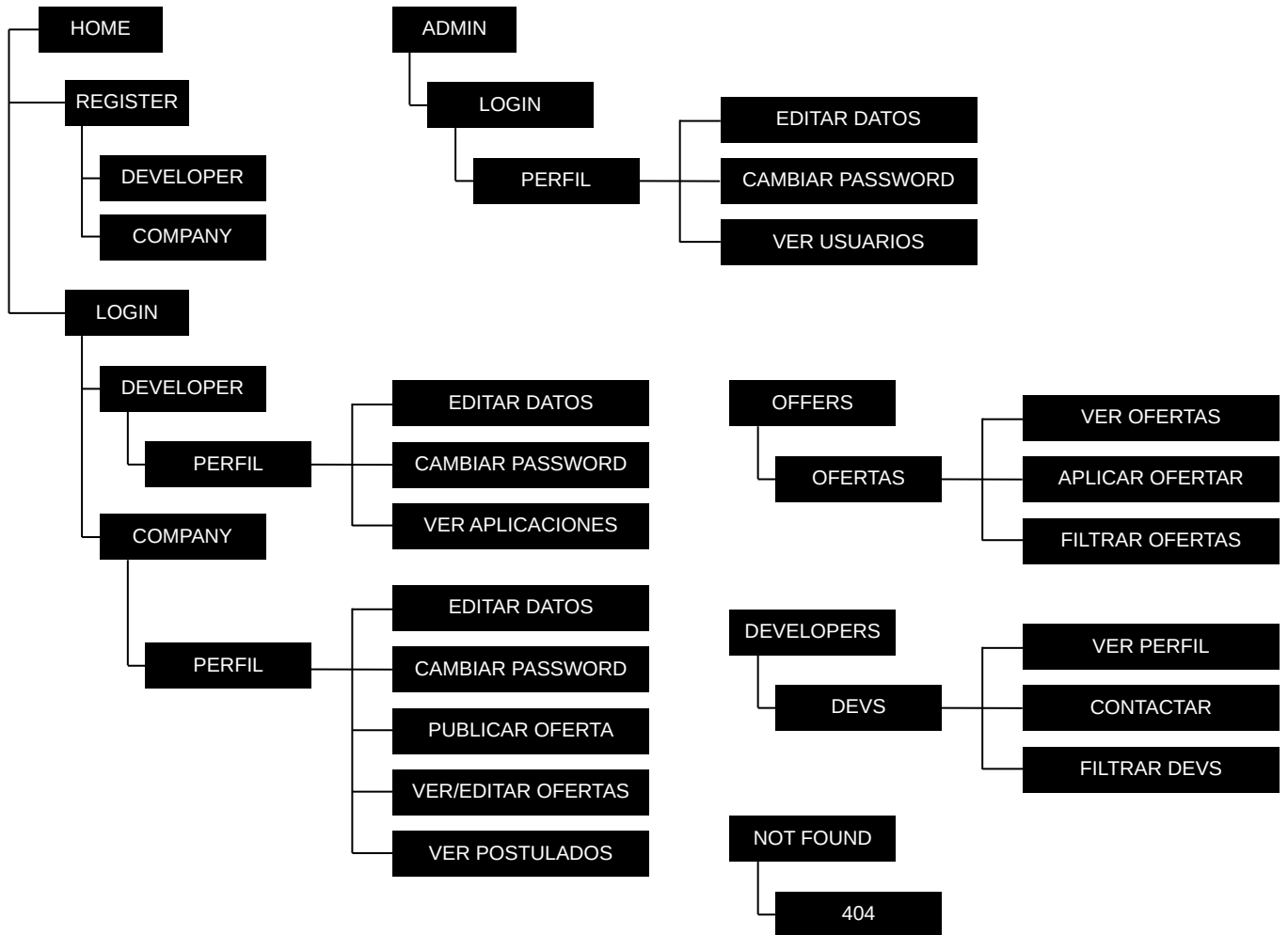


*Vista de Modal para cambiar contraseña



*Vista de Página 404

3.3.10 Mapa de navegación



4. Diseño del sistema

4.1 Arquitectura

La arquitectura de Jobs x Devs está basada en el modelo cliente-servidor, dividido en tres capas principales:

- Capa de presentación (frontend): desarrollada con Vue 3 utilizando Composition API. Esta capa se encarga de la interfaz de usuario y la interacción con los distintos componentes visuales y formularios.
- Capa de lógica de negocio (servidor): desarrollada con Node.js y Express, gestiona las rutas, controladores, validaciones y operaciones relacionadas con usuarios, autenticación y ofertas de empleo.
- Capa de datos: utiliza MySQL como sistema de gestión de bases de datos relacional. Aquí se almacenan los datos de usuarios, ofertas, aplicaciones y roles de acceso.

La comunicación entre el frontend y el backend se realiza mediante peticiones HTTP a endpoints definidos en la API REST. La arquitectura permite una separación clara de responsabilidades, facilita el mantenimiento y permite su futura escalabilidad.

4.1.1 Identificación de subsistemas

A continuación se identifican los principales subsistemas en los que se estructura la aplicación y su relación dentro del sistema general:

- Subsistema de autenticación y gestión de sesiones: controla el acceso de los usuarios según su rol, protege las rutas privadas y permite el inicio/cierre de sesión.
- Subsistema de gestión de usuarios: permite el registro, edición y eliminación de cuentas de desarrolladores y empresas. El administrador puede gestionar todos los usuarios desde su panel.
- Subsistema de gestión de ofertas: permite a las empresas publicar, editar y eliminar ofertas, y a los desarrolladores aplicar a ellas y gestionar sus postulaciones.
- Subsistema de visualización pública: se encarga de mostrar el home y la ruta /developers con la posibilidad de ver y contactar con los desarrolladores.
- Subsistema de administración: exclusivo del rol administrador, donde puede visualizar y editar los datos de todas las ofertas y desarrolladores registrados.

Estos subsistemas colaboran entre sí a través de rutas protegidas y operaciones coordinadas desde la API RESTful del backend.

4.2 Diseño de la base de datos

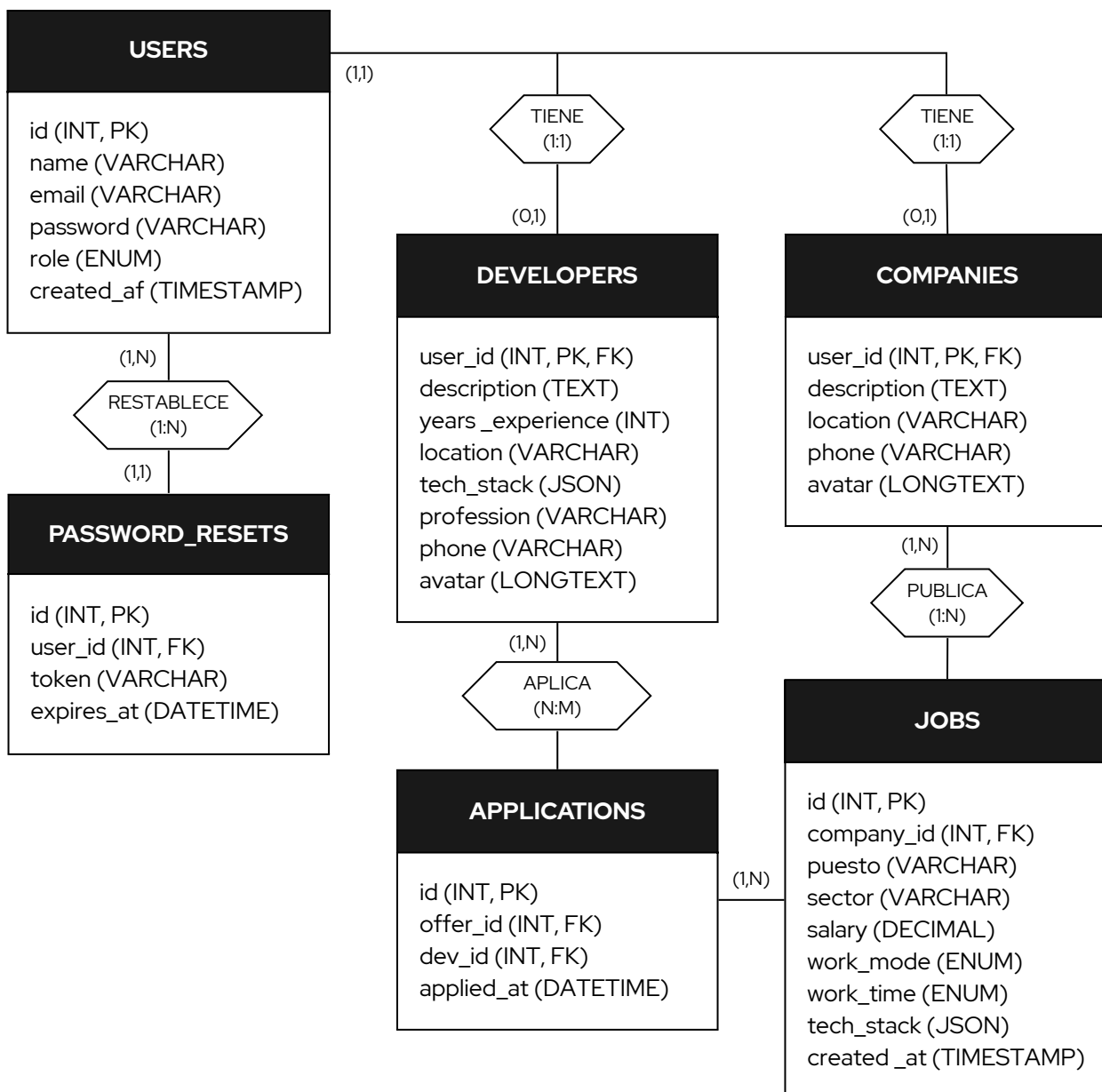
La base de datos del sistema Jobs x Devs está diseñada con un modelo relacional utilizando MySQL. A continuación se describen las principales tablas y relaciones:

- users: tabla principal de usuarios con los campos comunes (nombre, email, contraseña, rol). El campo role distingue entre 'dev', 'company' y 'admin'.
- developers: tabla que extiende a los usuarios con rol 'dev'. Contiene descripción, años de experiencia, localización, stack tecnológico (en JSON), profesión, teléfono y avatar.
- companies: tabla que extiende a los usuarios con rol 'company'. Contiene descripción, localización, teléfono y avatar.
- jobs: tabla de ofertas de empleo. Cada oferta pertenece a una empresa (vía company_id) y contiene información sobre el puesto, sector, modalidad, jornada, salario y tecnologías (campo JSON).
- applications: representa la relación muchos a muchos entre desarrolladores y ofertas. Cada aplicación guarda la fecha de postulación.
- password_resets: tabla auxiliar para gestionar el restablecimiento de contraseñas.

Relaciones clave:

- Un usuario puede ser desarrollador, empresa o administrador.
- Un desarrollador puede aplicar a múltiples ofertas y cada oferta puede recibir múltiples aplicaciones.
- Un administrador tiene visibilidad y control sobre todos los datos.

El modelo es escalable, claro y flexible gracias al uso de claves foráneas con eliminación en cascada (ON DELETE CASCADE), lo cual facilita la integridad referencial. A continuación, se muestra un diagrama relacional con las tablas y relaciones principales del sistema:



4.3 Manejo de errores

En el desarrollo de JobsXDevs, se ha prestado especial atención a la gestión de errores tanto en el lado del servidor (backend) como en el lado del cliente (frontend), implementando estrategias y herramientas que permiten detectar, informar y resolver incidencias de manera eficiente y segura.

4.3.1 Manejo de errores en el backend

En el backend, la gestión de errores se realiza de manera estructurada y centralizada. Cada controlador y ruta está diseñado para capturar posibles excepciones mediante bloques try/catch, lo que permite identificar y manejar tanto errores esperados (por ejemplo, datos inválidos o intentos de acceso no autorizado) como errores inesperados (fallos en la base de datos, problemas de red, etc.).

Para la validación de datos de entrada, se utiliza la librería express-validator, que permite definir reglas de validación y sanitización para cada endpoint. Si los datos recibidos no cumplen con los requisitos establecidos, el middleware de validación intercepta la petición y responde automáticamente con un mensaje de error detallado y un código de estado HTTP adecuado (generalmente 400 Bad Request). Esto previene que datos incorrectos o maliciosos lleguen a la lógica de negocio o a la base de datos.

La autenticación y autorización también están protegidas mediante middlewares específicos. Si un usuario intenta acceder a recursos restringidos sin los permisos necesarios, el sistema responde con códigos de error, según corresponda. Además, se emplea un middleware global para capturar cualquier error no gestionado en las rutas, devolviendo siempre una respuesta en formato JSON que incluye un mensaje claro y nunca expone información sensible del sistema.

4.3.2 Manejo de errores en el frontend

En el frontend, desarrollado con Vue.js, el objetivo principal del manejo de errores es ofrecer una experiencia de usuario clara, informativa y sin frustraciones. Todas las interacciones con la API del backend están envueltas en bloques try/catch, de modo que cualquier error en la comunicación (por ejemplo, fallos de red, respuestas con error del servidor, etc.) puede ser capturado y gestionado de forma adecuada.

Para la gestión del estado de error, se emplean variables reactivas que permiten actualizar la interfaz en tiempo real según la aparición o resolución de errores. Además, se utilizan componentes de notificación (toasts o alertas) para informar al usuario de errores generales, como problemas de conexión o fallos inesperados.

4.4 Elección de alternativas de componentes y licencias más adecuadas

Para el desarrollo de Jobs x Devs se han utilizado tecnologías de software libre o de uso gratuito, seleccionadas por su estabilidad, comunidad activa y adecuación al proyecto. La elección de componentes ha tenido en cuenta tanto criterios técnicos como legales para asegurar la sostenibilidad y compatibilidad de licencias.

COMPONENTE	TECNOLOGÍA	LICENCIA
Backend	Node.js + Express	MIT
Frontend	JavaScript + Vue 3	MIT
Base de datos	MySQL	GPL
Estilo y maquetación	HTML5, CSS3	Abierto
Entorno de desarrollo	Visual Studio Code	MIT
Control de versiones	Git / GitHub	GPL / SaaS
Diseño UI/UX	Figma (versión pro)	Propietario

La combinación de estas tecnologías ha permitido mantener el proyecto completamente funcional y accesible sin incurrir en costes por licencias. Todo el código desarrollado es original y puede licenciarse bajo una licencia abierta como MIT o GPL si se desea en el futuro.

4.5 Especificaciones de desarrollo

4.5.1 Necesidades de equipos

Equipo cliente:

- Navegador moderno (Chrome, Firefox, Edge)
- Conexión a internet
- No requiere instalación

Equipo de desarrollo:

- MacBook Pro con macOS
- Chip Apple M1
- 16 GB de RAM
- Conexión a internet estable

Servidor (desarrollo local):

- Servidor Express ejecutado desde entorno local para pruebas

4.5.2 Necesidades software

Utilidades de desarrollo:

- Sistema operativo: macOS
- Editor de código: Visual Studio Code
- Git para control de versiones
- Gestor de paquetes: npm

Aplicaciones del sistema:

- Backend: Node.js + Express
 - Librerías: bcryptjs, dotenv, express-validator, jsonwebtoken, cors, nodemailer, mysql2, morgan, cookie-parser
 - Entorno de desarrollo: nodemon
- Frontend: Vue 3 + Vite
 - Librerías: axios, vue-router, vue-toastification, jwt-decode, material-icons
 - Plugins: @vitejs/plugin-vue
 - Preprocesador: sass-embedded

Documentación y diseño:

- Figma para el prototipado UI
- Google Docs y Notion para la planificación y redacción de la memoria del proyecto

4.6 Especificación del plan de pruebas

Durante el desarrollo de Jobs x Devs se realizaron pruebas manuales en entorno local, enfocadas en validar que todas las funcionalidades principales estuvieran operativas y protegidas adecuadamente.

Pruebas realizadas:

- Comprobación de que todas las rutas privadas requirieran un token JWT válido.
- Verificación del flujo completo de registro, login y redirección por rol.
- Validación del funcionamiento correcto de cada ruta según el tipo de usuario (empresa, desarrollador, administrador).
- Pruebas de envío de correos para recuperación de contraseña mediante el módulo nodemailer.

Estas pruebas permitieron detectar errores puntuales durante el desarrollo, que fueron corregidos antes de dar por finalizada la versión funcional del proyecto.

5. Desarrollo

5.1 Desarrollo en el lado del cliente

En este proyecto, el frontend está construido con Vue.js, un framework progresivo de JavaScript para construir interfaces de usuario modernas y reactivas.

Características principales del frontend:

- Estructura SPA (Single Page Application): La aplicación se comporta como una SPA, lo que significa que la navegación entre páginas es rápida y sin recargas completas del navegador.
- Componentización: El código está organizado en componentes reutilizables (por ejemplo, formularios de login, perfiles, cards de ofertas, etc.), lo que facilita el mantenimiento y la escalabilidad.
- Gestión de rutas: Se utiliza Vue Router para gestionar la navegación entre las diferentes vistas (login, registro, perfil de empresa, perfil de desarrollador, ofertas, administración, etc.).
- Consumo de API: El frontend se comunica con el backend a través de peticiones HTTP (usando Axios o fetch), enviando y recibiendo datos en formato JSON.

- Autenticación y roles: El frontend gestiona la autenticación de usuarios y muestra diferentes interfaces según el rol (desarrollador, empresa, administrador).
- Responsive Design: Se han implementado estilos responsivos para asegurar que la aplicación sea usable tanto en escritorio como en dispositivos móviles y tablets.
- Gestión de estado y formularios: Se utilizan variables reactivas y formularios controlados para manejar el estado de la aplicación y la validación de datos del usuario.
- Feedback al usuario: Se muestran mensajes de éxito o error según las acciones realizadas (por ejemplo, al iniciar sesión, editar el perfil, aplicar a una oferta, etc.).

5.2 Desarrollo del lado del servidor

En este proyecto, el backend está desarrollado con Node.js y Express.js.

Características principales del backend:

- API RESTful: El backend expone una API RESTful que permite al frontend realizar operaciones CRUD (crear, leer, actualizar, eliminar) sobre los recursos principales: usuarios, empresas, desarrolladores, ofertas y aplicaciones.
- Gestión de autenticación y autorización: Se implementa autenticación basada en JWT (JSON Web Tokens) y cookies HttpOnly para proteger las rutas y asegurar que solo los usuarios autenticados puedan acceder a ciertas funcionalidades. Además, se gestionan diferentes roles (dev, company, admin) y permisos asociados.
- Estructura modular: El código está organizado en controladores, rutas, middlewares y modelos, siguiendo buenas prácticas de separación de responsabilidades.
- Conexión a base de datos MySQL: Se utiliza MySQL como sistema de gestión de base de datos relacional, accediendo a través del paquete mysql2. Se gestionan tablas para usuarios, empresas, desarrolladores, ofertas, aplicaciones, etc.
- Validación y seguridad: Se emplean middlewares para validar los datos de entrada (por ejemplo, usando express-validator), proteger contra ataques comunes (rate limiting, CORS, etc.) y asegurar la integridad de la información.
- Gestión de emails: El backend puede enviar correos electrónicos para funcionalidades como recuperación de contraseña o notificaciones, usando nodemailer.
- Logs y manejo de errores: Se utiliza morgan para el logging de peticiones y se implementa un manejo centralizado de errores para facilitar el mantenimiento y la depuración.
- Desarrollo y despliegue: El backend puede ejecutarse en modo desarrollo con nodemon y en producción con Node.js estándar.

6. Conclusiones

6.1 Nivel de satisfacción general

El desarrollo de Jobs x Devs ha supuesto un reto enriquecedor a nivel técnico y organizativo. Me ha permitido aplicar conocimientos de frontend y backend en un proyecto completo, estructurado y con utilidad práctica. La gestión de rutas protegidas, autenticación por roles y manejo de la base de datos han sido puntos clave en el aprendizaje.

Estoy satisfecho con el resultado alcanzado, ya que he logrado construir una plataforma funcional, adaptable y que responde a una necesidad real del entorno profesional de los desarrolladores.

6.2 Propuestas de mejora

Aunque el MVP del proyecto es completamente funcional, existen varias líneas de mejora que permitirían ampliar significativamente las capacidades de la plataforma y ofrecer una experiencia más completa a los usuarios:

- Mejorar los filtros de búsqueda: permitir filtrado por tecnologías específicas, nivel de experiencia, modalidad de trabajo y salario estimado.
- Implementar un sistema de coincidencia inteligente: cruzar automáticamente los stacks tecnológicos de las ofertas con los del perfil del desarrollador para mostrar un índice de compatibilidad (match).
- Añadir mensajería interna simple: permitir comunicación directa entre empresas y desarrolladores sin necesidad de salir de la plataforma.
- Sistema de favoritos: permitir a los usuarios guardar desarrolladores u ofertas para poder revisarlos más adelante.
- Estadísticas para administradores: incluir paneles con métricas sobre número de usuarios activos, ofertas creadas, aplicaciones realizadas, etc.
- Gestión de administradores: permitir el registro o creación de nuevos usuarios administradores desde la propia plataforma.
- Internacionalización (i18n): adaptar la plataforma a múltiples idiomas para ampliar su alcance internacional.
- Optimización SEO: mejorar la indexación pública de ciertas vistas como /developers para mayor visibilidad.

Estas mejoras no solo aportarían más valor a los usuarios, sino que también permitirían escalar la plataforma a un entorno profesional o comercial. de búsqueda para permitir filtrado por tecnologías específicas.

7. Bibliografía

- Vue.js - <https://vuejs.org>
- Vite - <https://vitejs.dev>
- Node.js - <https://nodejs.org>
- Express - <https://expressjs.com>
- MySQL - <https://www.mysql.com>
- GitHub - <https://github.com>
- JWT - <https://jwt.io>
- Nodemailer - <https://nodemailer.com>
- bcryptjs - <https://www.npmjs.com/package/bcryptjs>
- express-validator - <https://express-validator.github.io>
- Railway - <https://railway.app>
- Vercel - <https://vercel.com>
- Figma - <https://figma.com>
- Creative Commons - <https://creativecommons.org/licenses/by-sa/3.0/es/>

8. Enlaces

Enlaces del proyecto:

1. **Enlace del repositorio de GitHub:** https://github.com/davidstudiodev/tfg_daw_2025

2. **Enlace de la aplicación desplegada:** <https://www.jobsexdevs.com/>

3. **Enlace del diseño en Figma:**

Proyecto:

[https://www.figma.com/design/hbXdLRL5ReWpUFslmXEKUP/TFG?
m=auto&t=SMHOVBK09jCakl5u-1](https://www.figma.com/design/hbXdLRL5ReWpUFslmXEKUP/TFG?m=auto&t=SMHOVBK09jCakl5u-1)

Prototipo:

<https://www.figma.com/proto/hbXdLRL5ReWpUFslmXEKUP/TFG?page-id=2%3A2&node-id=28-6677&p=f&viewport=275%2C213%2C0.08&t=gWXTtlyF7sYuopuj-1&scaling=scale-down&content-scaling=fixed&starting-point-node-id=2%3A3>