

Anteproyecto

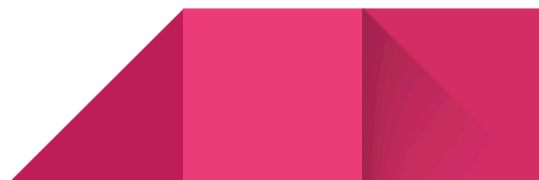
DAW

HR Manager

Nombre del Proyecto: HR Manager

Nombre de la alumna: Andrea Collazo Cacho

Ciclo formativo: Desarrollo de Aplicaciones Web (DAW) - Dual



1. Justificación del Proyecto

Durante mi periodo de prácticas de un curso de programación de sistemas informáticos en una academia, se me encomendó como ejercicio el desarrollo de una aplicación de escritorio para la gestión de recursos humanos en C#. Aunque la aplicación no se llegó a finalizar completamente, sentó las bases de un sistema funcional orientado a la administración de empleados, control de fichajes y vacaciones. Esa experiencia inicial, junto con el uso cotidiano de herramientas como PlexTime y Redmine en mi entorno laboral actual (empresa Plexus), me impulsó a retomar esta idea para darle una evolución profesional y convertirla en una aplicación web completa.

La motivación principal nace de la complejidad y limitaciones de herramientas actuales, que obligan a los usuarios a fichar en sistemas separados, con interfaces poco intuitivas. HR Manager nace con la intención de unificar y simplificar la gestión de recursos humanos, adaptada a pequeñas y medianas empresas.

2. Objetivos del Proyecto

Objetivo General

Desarrollar una aplicación web funcional para la gestión de recursos humanos, que permita administrar trabajadores, proyectos, ausencias y horas trabajadas, con un sistema de roles diferenciados para usuarios y administradores.

Objetivos Específicos

- Migrar la aplicación de escritorio (C#) a una arquitectura web moderna.
- Implementar autenticación segura con roles diferenciados.
- Gestionar empleados, contratos, documentos y proyectos desde un panel de administración.
- Permitir a los usuarios registrar horas trabajadas, descansos y solicitar ausencias.
- Ofrecer funcionalidades de exportación de informes e historial.

3. Tecnologías Empleadas

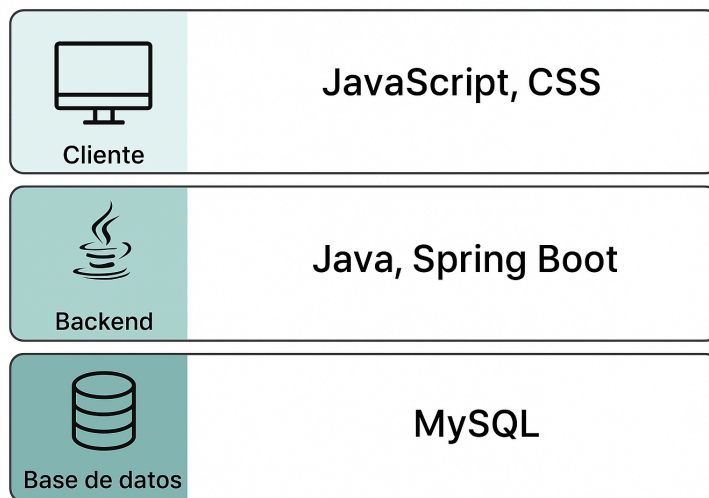
La selección de tecnologías para este proyecto se ha realizado con base en su madurez, soporte, documentación y adecuación a un entorno profesional de desarrollo web moderno:

- **Java con Spring Boot** ha sido elegido para el backend por ser un estándar en aplicaciones empresariales. Permite una estructura organizada, seguridad robusta (Spring Security), y facilita la persistencia con JPA.
- **JavaScript y CSS** para el frontend porque permite crear una interfaz ligera, comprensible y personalizable sin la necesidad de frameworks complejos, ideal para un proyecto de este tipo en entorno educativo.
- **MySQL** como base de datos por su fiabilidad, compatibilidad con Java y su uso extendido tanto en empresas como en formación.
- **JWT (JSON Web Tokens)** para la autenticación, por ser un estándar moderno y seguro en aplicaciones REST.

Este conjunto de tecnologías también facilita el despliegue en plataformas en la nube, permitiendo una transición sencilla a entornos reales si se desea escalar el proyecto.

A continuación, se representa un diagrama de la arquitectura de la aplicación y las tecnologías asociadas por capa:

- **Lenguaje Backend:** Java
- **Framework Backend:** Spring Boot (Spring Security, Spring Data JPA)
- **Frontend:** JavaScript y CSS puro
- **Base de datos:** MySQL
- **Seguridad:** JWT para autenticación y autorización
- **Herramientas:** IntelliJ IDEA, MySQL Workbench, Postman
- **Despliegue previsto:** Servidor local o en la nube (Render).
- **Despliegue final:** Local (Spring Boot embebido) o Render si se desea mostrar online.



4. Análisis Funcional

A continuación, se expresan las funcionalidades del sistema como historias de usuario, para reflejar las necesidades reales de los dos perfiles que utilizarán la aplicación:

Historias de usuario del perfil Administrador:

- Dar de alta, modificar y eliminar empleados para mantener actualizada la plantilla.
- Asignar empleados a proyectos para organizar y distribuir el trabajo.
- Aprobar o rechazar solicitudes de ausencias con comentarios para controlar la disponibilidad del personal.
- Gestionar los contratos de cada trabajador para llevar un control legal y documental.
- Subir y consultar documentos asociados a cada trabajador.
- Generar informes de actividad para cada trabajador y proyecto para evaluar el rendimiento del equipo.

Historias de usuario del perfil Usuario (Empleado):

- Iniciar sesión con mis credenciales para acceder a mi información personal.
- Registrar mis horas trabajadas y descansos cada día para fichar correctamente.
- Solicitar vacaciones o ausencias indicando el tipo y motivo.
- Visualizar el estado de mis solicitudes de ausencias (aprobadas, pendientes, rechazadas) junto con comentarios del administrador.

5. Diseño Técnico

Arquitectura del Sistema

El sistema estará basado en una arquitectura cliente-servidor, distribuida en tres capas:

- **Capa Cliente:** Interfaz web desarrollada en JavaScript y CSS.
- **Capa Backend:** (Lógica de negocio) API REST desarrollada con Java y Spring Boot.
- **Capa de Base de Datos:** Gestión relacional en MySQL usando JPA

Esquema de funcionamiento general:

1. El usuario accede a la aplicación mediante el navegador.
2. El sistema solicita autenticación mediante formulario (JWT).
3. Según el rol, se accede a funcionalidades del panel de usuario o administración.
4. Las acciones invocan peticiones HTTP que interactúan con la lógica de negocio.
5. El backend consulta/modifica datos en la base de datos y responde con JSON.

1. Acceso inicial

Usuario → (Navegador web) → Aplicación web (JavaScript/CSS)

La aplicación se carga en el navegador mediante archivos estáticos (HTML/CSS/JS).

2. Autenticación con JWT

Sistema → Formulario de login → Usuario

Usuario → Credenciales → Backend (Spring Security) → Genera JWT

Spring Boot valida las credenciales contra la base de datos (MySQL).

Si son correctas, genera un JWT con el rol (Usuario/Admin) y lo envía al frontend.

3. Control de acceso basado en roles

Frontend → Decodifica JWT → Redirige según rol

Rol "Admin":

Acceso al panel de administración (gestión de empleados, proyectos, aprobación de ausencias).

Funcionalidades: CRUD de trabajadores, generación de informes, subida de documentos.

Rol "Usuario":

Acceso al panel de empleado: registro de horas, solicitud de ausencias, visualización de estado.

4. Interacción con la API REST (Spring Boot)

Frontend (JavaScript) → HTTP Request → Backend (Spring Boot)

Ejemplos de endpoints:

POST /api/horas → Registra horas trabajadas (empleado).

GET /api/ausencias → Lista de ausencias pendientes (admin).

Headers: Incluyen el JWT (Authorization: Bearer <token>).

5. Lógica de negocio y base de datos

Backend → Spring Data JPA → MySQL

Spring Boot ejecuta consultas JPQL/SQL según la petición:

Ejemplo: `SELECT * FROM TRABAJADORES WHERE rol = 'USER'`.

Los resultados se mapean a objetos Java y se devuelven como JSON.

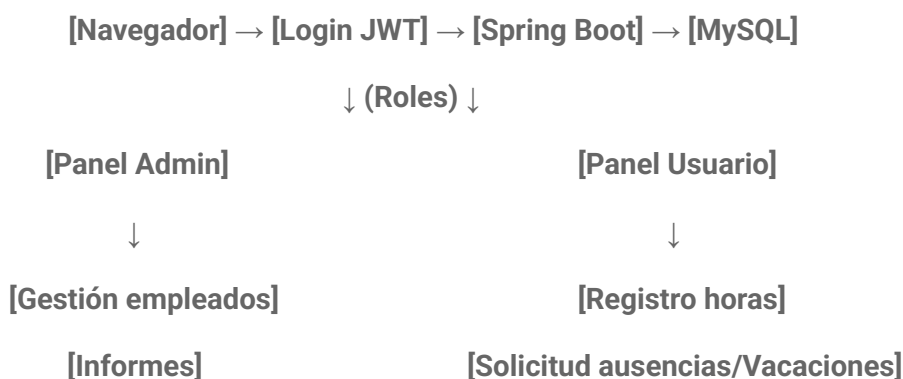
6. Respuesta y actualización de la UI

Frontend → Procesa JSON → Renderiza cambios (JavaScript puro)

Ejemplo:

Tras registrar horas, muestra un mensaje de éxito.

Tras solicitar la ausencia, actualiza el listado con el estado "Pendiente".

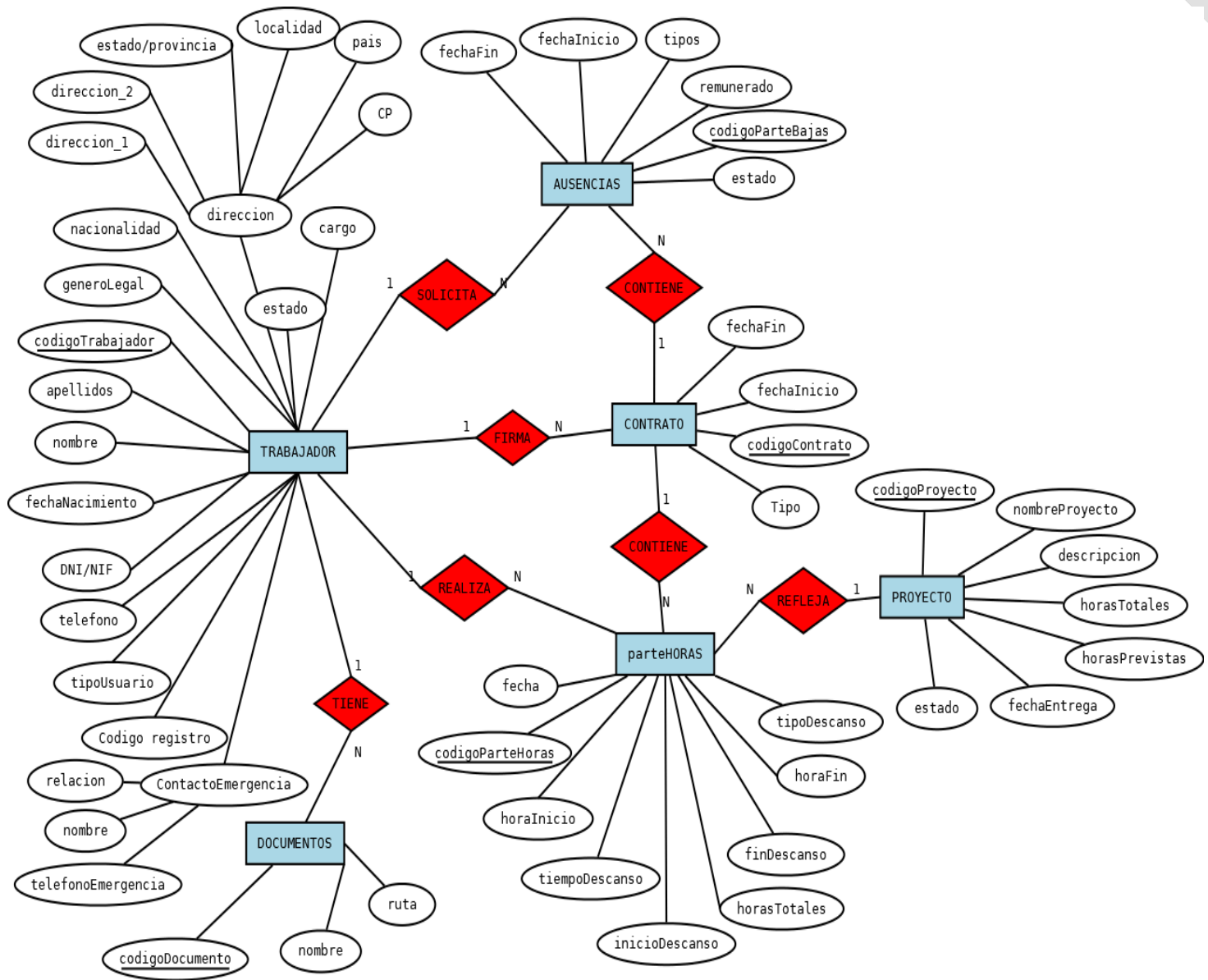


Modelo de datos relacional (MySQL):

El sistema se apoya en un esquema relacional con entidades:

- Trabajadores
- Proyectos
- Contratos
- Documentos
- Ausencias
- Partes de horas

Diagrama ER



6. Proceso de Desarrollo

Metodología

Se utilizará el modelo en cascada, adecuado para un proyecto individual con objetivos bien definidos y tiempos establecidos. Esta metodología permite estructurar el proyecto en fases consecutivas que garantizan una entrega ordenada.

Cada fase finaliza completamente antes de avanzar a la siguiente, lo que facilita el control del progreso y la revisión del cumplimiento de requisitos.

Fases del desarrollo:

1. Análisis y diseño técnico
2. Configuración del entorno de desarrollo
3. Implementación de la base de datos
4. Desarrollo del backend (API REST)
5. Desarrollo del frontend
6. Pruebas funcionales
7. Documentación final

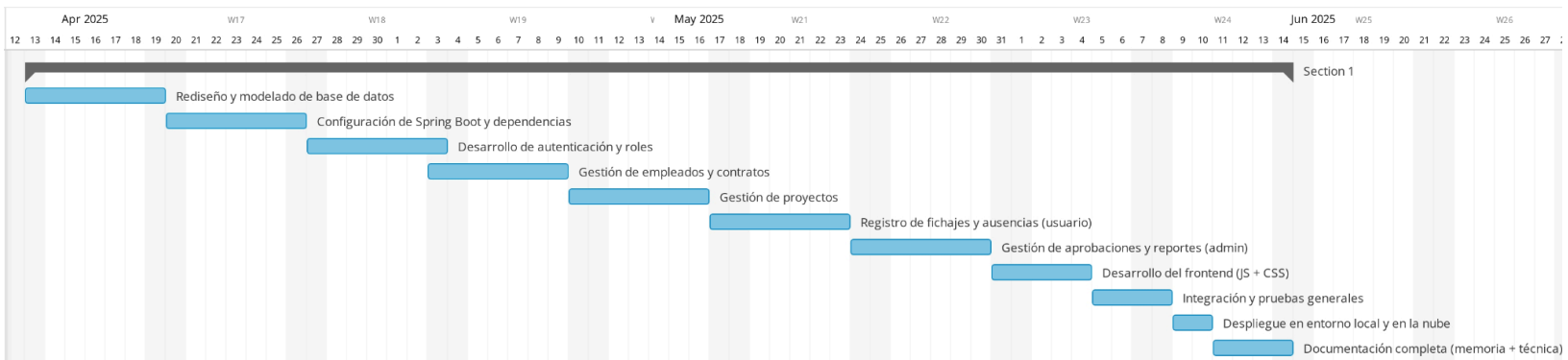
Flujo de trabajo general:

- Desarrollo por módulos funcionales definidos previamente.
- Cada módulo se implementará, probará y documentará antes de pasar al siguiente.
- Se realizarán validaciones manuales al finalizar cada bloque.

7. Planificación Detallada

Se estima una dedicación de 20 horas semanales, desde abril hasta mediados de junio (aproximadamente 8-9 semanas, total de **230 horas**).

	ACTIVITIES	EH	START	DUE
Section 1		230h	13/Apr	14/Jun
1	✓ Rediseño y modelado de base de datos	20h	13/Apr	19/Apr
2	✓ Configuración de Spring Boot y dependencias	20h	20/Apr	26/Apr
3	✓ Desarrollo de autenticación y roles	20h	27/Apr	03/May
4	✓ Gestión de empleados y contratos	20h	03/May	09/May
5	✓ Gestión de proyectos	20h	10/May	16/May
6	✓ Registro de fichajes y ausencias (usuario)	20h	17/May	23/May
7	✓ Gestión de aprobaciones y reportes (admin)	20h	24/May	30/May
8	✓ Desarrollo del frontend (JS + CSS)	40h	31/May	04/Jun
9	✓ Integración y pruebas generales	20h	05/Jun	08/Jun
10	✓ Despliegue en entorno local y en la nube	10h	09/Jun	10/Jun
11	✓ Documentación completa (memoria + técnica)	20h	11/Jun	14/Jun



8. Medios Materiales Necesarios

Hardware:

- Ordenador personal con mínimo 8 GB de RAM
- Conexión estable a Internet

Software:

Herramienta	Finalidad	Licencia	Coste	Fase de uso
IntelliJ IDEA	Desarrollo backend (Java + Spring)	Gratuita (Community)	0 €	Toda la fase de desarrollo
MySQL Workbench	Modelado y consulta de base de datos	Gratuita	0 €	Diseño y pruebas BD
Postman	Pruebas de API REST	Gratuita	0 €	Backend / pruebas
Navegador (Chrome)	Visualización de la aplicación web	Gratuita	0 €	Desarrollo / pruebas

Google Drive	Almacenamiento en la nube	Gratuita (limitada)	0 €	Copias de seguridad / docs
Visual Studio Code (VSCode)	Desarrollo frontend (JavaScript + CSS)	Gratuita	0 €	Desarrollo frontend
Render	Despliegue	Gratuita	0€	Despliegue
Railway	Base de datos	Gratuita	0€	Base de datos

9. Despliegue

Para la fase final del proyecto, se ha optado por un despliegue en la nube utilizando dos servicios complementarios que permiten publicar la aplicación y su base de datos de forma sencilla, accesible y sin costes para proyectos personales o educativos.

Plataforma de despliegue elegida: Render + Railway (MySQL)

La aplicación backend desarrollada en Spring Boot se desplegará en la plataforma cloud **Render**, que permite alojar servicios web escritos en Java, con integración directa desde repositorios Git. Esta plataforma ofrece una capa gratuita suficiente para mantener la aplicación en línea, accesible desde cualquier navegador mediante una URL pública.

La base de datos **MySQL** estará alojada en **Railway**, un servicio de bases de datos en la nube que proporciona instancias de forma gratuita (hasta 500MB), con acceso remoto y gestión mediante panel web. Esto evita depender de un servidor local o mantener MySQL activo manualmente, lo que simplifica el despliegue.

Configuración y acceso

La conexión entre la aplicación y la base de datos se realiza mediante los parámetros que proporciona Railway: host, puerto, nombre de la base de datos, usuario y contraseña. Estos datos se configuran como variables de entorno seguras dentro de Render, evitando la exposición de información sensible en el código fuente.

La aplicación se construye como un fichero `.jar` mediante Maven y se despliega automáticamente desde el repositorio del proyecto al entorno de producción.

Ventajas del entorno seleccionado

- No requiere mantener un servidor encendido localmente.
- Despliegue sencillo, sin configuración compleja de infraestructura.
- Totalmente gratuito en sus versiones básicas.
- Ideal para proyectos de estudiantes y demostraciones públicas.
- Escalable en caso de evolución a una versión comercial o profesional.

10. Conclusiones y Aportaciones Personales

Este proyecto supone la consolidación de conocimientos adquiridos durante el ciclo formativo y en mi experiencia laboral. Me permite cerrar una idea que nació durante mis primeras prácticas y llevarla a un nivel profesional, cubriendo aspectos de backend, frontend, seguridad y base de datos. Al mismo tiempo, responde a una necesidad real detectada en el mercado: una herramienta de recursos humanos más simple, clara y eficaz para PYMEs.

La evolución del proyecto desde su concepción en C# hasta su versión moderna en Java y web representa también mi propio crecimiento como desarrolladora.

11. Perspectiva empresarial: intraemprendimiento, recursos y financiación

11.1 Justificación y necesidad en el entorno productivo

Durante mi experiencia profesional en Plexus, he observado cómo el uso simultáneo de herramientas como PlexTime y Redmine genera fricciones en la gestión diaria del personal, ya que obligan a alternar entre múltiples sistemas con interfaces poco integradas. Esta fragmentación genera pérdidas de tiempo, duplicidad de datos y baja satisfacción del usuario final. En ese contexto, **HR Manager** surge como respuesta directa a una necesidad real detectada desde dentro de la empresa.

El proyecto se concibe como una acción de **intraemprendimiento**, en la que una persona empleada propone, diseña y desarrolla una solución innovadora que puede mejorar los procesos internos. Es, por tanto, una forma de emprendimiento no orientado al lucro personal, sino al **aporte de valor dentro de la empresa** mediante la mejora operativa y tecnológica.

11.2 Asignación de recursos

Recursos humanos

El proyecto está desarrollado de forma individual por la autora, sin la participación de un equipo de trabajo. No obstante, se estiman unas 230 horas de dedicación personal durante el periodo de desarrollo (abril-junio), lo que equivale a una jornada parcial intensiva. No se contempla contratación externa ni subcontratación de servicios.

Recursos materiales

Recurso	Uso	Licencia	Coste estimado
Ordenador personal	Desarrollo completo	Propio	0 €
IntelliJ IDEA	Backend Java (Spring Boot)	Gratuita (Community)	0 €
Visual Studio Code	Frontend JavaScript/CSS	Gratuita	0 €
MySQL Workbench	Diseño base de datos	Gratuita	0 €
Postman	Pruebas API REST	Gratuita	0 €
Google Drive	Copias de seguridad	Gratuita (limitada)	0 €
Render	Despliegue	Gratuita	0€
Railway	Base de datos	Gratuita	0€

Total costes materiales: 0 €

11.3 Financiación

El proyecto no requiere financiación externa. Todos los recursos utilizados son gratuitos o ya disponibles, por lo que **no se prevé endeudamiento ni búsqueda de inversores**. Al tratarse de un proyecto de tipo académico y experimental, desarrollado dentro del entorno profesional, se sostiene mediante recursos propios sin generar costes adicionales.

En caso de evolución hacia un entorno real, Plexus podría considerar su integración como herramienta interna. En tal escenario, la financiación podría provenir del propio presupuesto interno de I+D, sin implicar una comercialización directa.

11.4 Forma jurídica y modelo de negocio

No se contempla la creación de una empresa independiente ni la obtención de beneficios mediante licencias o venta. El proyecto se desarrolla **sin ánimo de lucro y con un enfoque profesional**, como un ejemplo de innovación y mejora continua dentro de una organización ya establecida.

No obstante, si en un futuro se quisiera escalar el producto a otras PYMEs o comercializarlo como servicio SaaS, se podría valorar una estrategia freemium con módulos básicos gratuitos y servicios premium bajo suscripción. Por ahora, esa posibilidad se mantiene como una proyección teórica.

11.5 Conclusión empresarial

HR Manager no solo busca resolver una necesidad técnica, sino también demostrar cómo el talento interno puede generar valor a través de soluciones adaptadas y eficientes. Esta iniciativa se alinea con una cultura de mejora continua, ahorro de costes y empoderamiento del personal técnico, reforzando el papel de la innovación desde dentro de la empresa.

12. Uso de Herramientas de IA en el proyecto

- **¿Qué herramienta de IA generativa usaste (nombre y versión)?**

Se utilizó **ChatGPT 4 de OpenAI (versión abril 2024)** como apoyo para la redacción y organización del anteproyecto.

- **¿Para qué usaste la herramienta?**

Para ayudar a redactar, organizar y estructurar el contenido del documento del anteproyecto, incluyendo esquemas visuales como diagrama de arquitectura. También se empleó para revisar la coherencia técnica del texto en función de los requisitos del profesorado.

- **¿Qué contenidos fueron generados por el agente?**

1. Redacción organizada y formal de las secciones del anteproyecto (con base en contenido y datos aportados por mi).
2. Esquema de arquitectura técnica (figura externa).
3. Apoyo en justificación de tecnologías y estructura metodológica.
4. Revisión del lenguaje técnico y adecuación al formato exigido por el centro.

Nota: Las historias de usuario, la planificación temporal y todos los contenidos de fondo fueron aportados por la autora del proyecto. El uso de la IA se limitó a darles forma estructurada, académica y coherente con el estilo del documento.

- **¿Cómo has utilizado o cambiado la salida de la IA generativa?**

El contenido sugerido fue siempre revisado, corregido y adaptado, el desarrollo previo en C#, y la documentación real del proyecto. Se reorganizaron las secciones para adaptarlas al modelo de anteproyecto proporcionado por el centro.

13. Bibliografía

11. Bibliografía y recursos consultados

- Spring Team. (s.f.). *Spring Boot Project*. Spring.io. Recuperado de <https://spring.io/projects/spring-boot>
- Oracle. (s.f.). *MySQL Documentation*. MySQL. Recuperado de <https://dev.mysql.com/doc/>
- JWT.io. (s.f.). *Introduction to JSON Web Tokens*. Recuperado de <https://jwt.io/introduction>
- W3Schools. (s.f.). *HTML, CSS, JavaScript Tutorials*. Recuperado de <https://www.w3schools.com/>
- OpenAI. (2024). *ChatGPT como apoyo en redacción y estructuración de documentos*.
- Collazo Cacho, A., et al. (2022). *Proyecto de gestión de RRHH en C#* [Proyecto interno de prácticas grupal]. Academia privada (LideraK).
- Xunta de Galicia. (s.f.). *Recursos y apuntes del módulo DAW Dual*.