

**EEP  
iGroup**

**APLICACIÓN WEB SOBRE GESTOR DE EMPLEADOS**

**CFGS  
2025**

# **ESCUELA DE ESTUDIOS PROFESIONALES**

**CICLO FORMATIVO DE GRADO SUPERIOR EN  
DESARROLLO DE APLICACIONES WEB**



## **TRABAJO DE FIN DE GRADO**

**Título del trabajo de fin de Grado**

**APLICACIÓN WEB GESTOR DE EMPLEADOS**

**BORIS ALEXIS RUÍZ RODRÍGUEZ**

**(JUNIO 2025)**



**EEP iGroup Arturo Soria**  
**TÉCNICO SUPERIOR EN DESARROLLO DE APLICACIONES WEB**

**(Aplicación Web Sobre La Gestión de Empleados)**

**ALUMNO:** Boris Alexis Ruíz Rodríguez

**TUTOR ACADÉMICO:** Pedro

**FECHA DE PRESENTACIÓN:** Junio 2025

**BREVE DESCRIPCIÓN:**

El proyecto consiste en el desarrollo de una aplicación web para la gestión de empleados y asignación de tareas para una empresa pequeña, orientada a optimizar el proceso de gestión de empleados mediante la implementación de la tecnología Crear, Leer, Actualizar y Eliminar (CRUD). La aplicación proporciona una plataforma intuitiva y eficiente para que los gerentes de recursos humanos administren de manera centralizada toda la información relacionada con los empleados. El usuario si tiene credenciales puede acceder a las características de la app, sino no puede acceder y se le redirige al formulario de inicio de sesión.

La función principal de la aplicación es registrar a todos los empleados registrados en la base de datos. La lista se presenta en un formato claro y organizado para que pueda filtrar y buscar empleados según una variedad de criterios, incluido el nombre, el apellido, puesto y código de empleado. Los usuarios autorizados también pueden agregar nuevos empleados a la base de datos e ingresar información como nombre, puesto de trabajo y otra información relevante. A su vez, podrá asignar tareas a los empleados existentes.

Esta aplicación le permite editar información existente para que sea más fácil actualizar datos como: Cambio de puesto, nombre o apellido o corrección de errores, etc. El proceso es simple, seguro y garantiza la integridad de los datos mediante la autenticación y autorización adecuadas. Eso también estará incluido. Posibilidad de eliminar registros de empleados si es necesario, por ejemplo, en caso de despido permanente. Y puedes asignarles tareas también con la fecha asignada y la puedes borrar y finalizar la tarea del empleado.

El diseño de la interfaz es intuitivo y fácil de entender para usuarios con conocimientos leves, permitiendo que la interacción usuario sea agradable y no muy compleja.

**FIRMA DEL TUTOR**

**FIRMA DEL ALUMNO**

### **Resumen:**

El proyecto consiste en una aplicación sobre la gestión de empleados para una empresa, normalmente enfrentamos desafíos como la eficiencia, comunicación y manejo de datos.

Este trabajo representa el desarrollo de una aplicación web de gestión de empleados diseñada para mantener el control de empleados, en su caso permitirá editar información acerca del empleado, como su nombre, puesto de trabajo o sino está más en la empresa borrarlo.

A través de una metodología que combina la fácil interacción de usuario, su funcionalidad permite a las empresas simplificar y registrar de manera de manera eficiente datos. Implementa acciones básicas como un formulario de inicio de sesión para que el usuario pueda registrarse y crear editar y borrar que son el núcleo de la gestión de empleados ya que constituye la base para las operaciones básicas, la creación permite registrar a un empleado de forma rápida y estructurada, la edición permite realizar actualizaciones garantizando el cambio de datos, como nombre, puesto de trabajo, etc.

Por último, la eliminación de registros proporciona de manera eficiente la limpieza de base de datos, asegurando que los empleados que dejan la organización sean eliminados, lo que evita la acumulación de datos innecesarios y reduce el exceso de memoria en la empresa.

### **Palabras clave:**

Gestión de empleados, empresas, organización, registros, tareas, asignación...

**Abstract:**

The project consists of an application on employee management for a company, we normally face challenges such as efficiency, communication and data management.

This work represents the development of an employee management web application designed to maintain control of employees, in this case it will allow editing information about the employee, such as their name, job position or if they are no longer in the company, deleting them.

Through a methodology that combines easy user interaction, its functionality allows companies to simplify and efficiently record data. It implements basic actions such as creating, editing and deleting which are the core of employee management since it constitutes the basis for basic operations, creation allows an employee to be registered quickly and in a structured way, editing allows updates to be made ensuring the change of data, such as name, job position, etc.

Finally, deleting records efficiently provides database cleaning, ensuring that employees who leave the organization are deleted, which prevents the accumulation of unnecessary data and reduces excess memory in the company.

**Keywords:**

Employee management, companies, organization, records, tasks, assignment...

## Índice de contenidos

### Tabla de contenido

<b>Resumen.....</b>	<b>4</b>
<b>Abstract.....</b>	<b>5</b>
<b>2. Introducción: .....</b>	<b>8</b>
2.1 Presentación del Tema y su relevancia: .....	9
2.2 Objetivos del TFG: .....	9
2.3 Justificación del Proyecto: .....	10
<b>3. Marco Teórico:.....</b>	<b>11</b>
3.1 Revisión de la literatura relacionada con el tema del proyecto .....	11
3.1.1 Sistemas de Gestión de Empleados.....	11
3.1.2 Seguridad y Autenticación de Usuarios .....	12
3.1.3 APIs REST y Arquitectura Cliente-Servidor .....	12
3.1.4 Solución Sencilla pero Eficaz .....	13
3.2 Descripción de Tecnologías, herramientas y metodologías relevantes .....	13
3.2.1 Lenguajes de Diseño Web y Programación .....	14
3.2.2 Backend y API REST .....	14
3.2.3 Herramientas de Desarrollo y Pruebas .....	15
3.2.4 Metodología de Desarrollo .....	16
<b>4. Metodología .....</b>	<b>17</b>
4.1 Descripción detallada del enfoque metodológico utilizado en el Proyecto: .....	17
4.1.1 Fases del Proceso y Etapas de Desarrollo.....	17
4.1.2 Gestión del Tiempo y Recursos .....	19
4.1.3 Justificación de la metodología Ágil.....	20
4.1.4 Limitaciones del Método Ágil .....	21
4.2 Herramientas y Tecnologías Utilizadas .....	21
4.3 Detalles sobre el Diseño y la Arquitectura de la Aplicación .....	23
4.3.1 Mejoras implementadas y una tabla adicional. ....	24
<b>5. Desarrollo e Implementación .....</b>	<b>25</b>

5.1 Descripción detallada del Proceso de Desarrollo, incluyendo desafíos y soluciones encontradas .....	25
<b>6. Resultados.....</b>	<b>27</b>
6.1 Presentación de la aplicación desarrollada, funciones principales .....	27
6.2 Resultados de las pruebas y evaluaciones sobre la aplicación.....	28
6.2.1 Pruebas funcionales .....	29
6.2.2 Pruebas de usabilidad.....	29
6.2.3 Pruebas de rendimiento.....	30
6.3 Comparaciones con soluciones en el mercado (ventajas y desventajas) .....	30
6.3.1 Comparación con la aplicación desarrollada (ventajas y desventajas) .....	32
<b>7. Discusión .....</b>	<b>33</b>
7.1 Análisis de Resultados Obtenidos.....	33
7.2 Reflexiones sobre limitaciones y posibles áreas de mejora: .....	34
7.3 Implicaciones prácticas y teóricas de los resultados.....	36
<b>8. Conclusiones .....</b>	<b>37</b>
8.1 Resumen hallazgos claves del Proyecto .....	37
8.2 Lecciones Aprendidas durante el Desarrollo del Proyecto. ....	38
<b>9. Referencias Bibliográficas.....</b>	<b>39</b>

## 2. Introducción:

En el entorno empresarial actual, la gestión eficaz de los recursos humanos se ha convertido en un desafío importante en organizaciones de todos los tamaños e industrias. La gestión de personal, que incluye la creación de nuevos registros, la actualización y renovación de información y la eliminación de datos antiguos, es una parte importante para garantizar el buen funcionamiento de una empresa. Sin embargo, muchas organizaciones aún dependen de procesos manuales o fragmentados, lo que genera fracasos, duplicación de esfuerzos, errores de datos y desperdicio de tiempo y recursos.

La importancia de este desarrollo radica en su capacidad de transformar los procesos de gestión interna, reducir las cargas administrativas, reducir la sobrecarga de registros y saber administrar información sobre los empleados que conforman una organización, así como jefe de empresa llevar un seguimiento sobre que empleados siguen en la empresa y cuáles no.

Mi objetivo con este proyecto es ofrecer una solución que permita a las empresas gestionar de manera centralizada y eficiente toda la información relacionada con sus empleados. Con esta aplicación, los usuarios podrán crear, editar y eliminar registros de personal de manera intuitiva y segura, optimizando no solo el tiempo, sino también los recursos administrativos. He diseñado esta aplicación pensando en resolver problemas reales que afectan a las organizaciones, asegurándome de que sea accesible, funcional y escalable para adaptarse a las necesidades de diferentes tipos de empresas.



## 2.1 Presentación del Tema y su relevancia:

Mi proyecto se centra en el desarrollo de una aplicación de gestión de empleados, diseñada para facilitar la creación, edición y eliminación de información relacionada con el personal de una organización. Donde la digitalización y eficiencia en gestión empresarial son esenciales, cobra una gran relevancia.

La gestión de empleados es fundamental para cualquier empresa, independientemente del tamaño o su sector. Actualmente este proceso se hace manualmente mediante tablas de registros, con la aplicación facilitaría estos procesos de manera optima sin evitar errores humanos permitiendo una interfaz de usuario intuitiva, simple pero robusta.

## 2.2 Objetivos del TFG:

El objetivo fundamental para la realización de este trabajo es familiarizarme con un sistema de login estilo registro, si el usuario tiene credenciales pueda acceder a las diferentes funcionalidades de la app, luego un CRUD (Crear, Leer, Actualizar, Borrar) y adquirir más conocimientos sobre este campo.

De este modo, el objetivo principal es desarrollar una aplicación web, con interfaz intuitiva, que permita gestionar a empleados en las empresas. En la que se llevará a cabo el proceso de crear, actualizar y borrar registros de dicho empleado, borrar para evitar la acumulación de datos innecesaria.

Así también permitiendo leer los registros de cada empleado y a su vez crear o actualizar datos en la organización, constara de una lista la cual podrás editar los registros que el usuario desee guardando dicho registro en la lista.

### 2.3 Justificación del Proyecto:

Este proyecto presenta una oportunidad para aplicar de manera práctica los conocimientos adquiridos durante el año escolar, como la programación, bases de datos y la gestión. Además, me permite adquirir experiencia en el desarrollo de software y en el manejo de herramientas modernas.

Las empresas, en concreto las pequeñas o medianas, suelen enfrentar dificultades al manejar datos de empleados de forma manual o mediante sistemas poco óptimos. Estas dificultades incluyen errores en el almacenamiento, falta de actualizaciones oportunas y duplicación de información. Este proyecto busca ofrecer una solución directa que permita gestionar, editar, crear y eliminar registros de empleados con facilidad y precisión.

En pocas palabras, este proyecto tiene un impacto directo y positivo en el tema de administrar empleados, respondiendo a una necesidad concreta en el ámbito empresarial. Al mismo tiempo me permite desarrollar habilidades técnicas y ofrecer una aportación de mis conocimientos adquiridos. Por estas razones me inspiré a llevar a cabo este proyecto.

### 3. Marco Teórico:

#### 3.1 Revisión de la literatura relacionada con el tema del proyecto

La gestión de empleados y la seguridad de sistemas de información son dos bloques ampliamente estudiados en el mundo de la informática y en sistemas de información. El presente proyecto se sitúa en las dos disciplinas anteriormente mencionadas, desarrollando una aplicación web de control de empleados que implementa acceso restringido mediante la autenticación de usuarios.

##### 3.1.1 Sistemas de Gestión de Empleados

La gestión de empleados ha evolucionado significativamente en los últimos años. Anteriormente, la información del personal era almacenada en archivos físicos, pero con el avance en el área de la informática, dieron a luz sistemas informatizados que son capaces de registrar, actualizar y administrar datos de los empleados de manera más eficiente y evitando errores humanos. Actualmente, los sistemas de gestión de empleados (conocidos también como HRMS, Human Resources Management Systems) permiten realizar operaciones básicas como la creación, edición, eliminación y consulta de registros de trabajadores.

Según diversos estudios (Smith et al., 2015; Kumar, 2017), la digitalización de los procesos de recursos humanos mejora la eficiencia administrativa, reduce los errores y permite la toma de decisiones basadas en datos actualizados. Aunque existen soluciones comerciales más complejas (SAP HR, Workday, etc.), muchas pequeñas y medianas empresas necesitan soluciones más simples y personalizadas, lo que justifica el desarrollo de proyectos de gestión básicos como el presente.

### 3.1.2 Seguridad y Autenticación de Usuarios

La autenticación es el primer paso esencial para proteger aplicaciones y datos sensibles. El control de acceso mediante un formulario (login) es una de las técnicas más comunes para asegurar que solo usuarios autorizados puedan interactuar con sistemas privados o diferentes características de una web. Diversos métodos de autenticación han sido propuestos y probados, desde contraseñas simples hasta autenticación de múltiples factores.

De acuerdo con el artículo (Bonneau et al., 2012), un sistema de autenticación básico basado en nombre de usuario o Email y contraseña siendo validados e implementando buenas prácticas de almacenamiento seguro. En este proyecto, aunque se opta por un enfoque sencillo, se contempla como mejora futura la implementación de métodos de autenticación más robustos y sólidos.

### 3.1.3 APIs REST y Arquitectura Cliente-Servidor

El modelo de comunicación REST (Representational State Transfer) ha sido ampliamente adoptado para el desarrollo de sistemas modernos que separan el Frontend del Backend.

En su momento, Fielding (2000) propuso REST como un conjunto de principios arquitectónicos para desarrollar aplicaciones web escalables y eficientes.

En el presente proyecto, el Backend ha sido desarrollado en Java utilizando Spring Boot, que ofrece herramientas para construir APIs RESTful de manera rápida y segura. La base de

datos con MySQL para guardar la información necesaria y permite un manejo de datos sencillo pero sólido. Y el sistema de login fue desarrollado con PHP y MySQL, permitiendo guardar la información del usuario para que este pueda acceder a las diferentes características de la aplicación si tiene usuario, si no tiene puede crear uno fácilmente para acceder a las diferentes características de la app. El Frontend, construido con HTML, CSS y JavaScript, consume estas APIs mediante peticiones HTTP (GET, POST, PUT, DELETE), siguiendo de esta manera los principios REST. Este enfoque permite facilitar el mantenimiento, la ampliación y la integración futura con otras plataformas.

#### 3.1.4 Solución Sencilla pero Eficaz

A pesar de la existencia de soluciones avanzadas para la gestión de recursos humanos, se detectó una carencia en el acceso a sistemas de gestión ligeros y de fácil implantación para organizaciones pequeñas o proyectos de carácter educativo. Este proyecto, por tanto, busca cubrir esa necesidad mediante una solución sencilla pero efectiva, que demuestra conceptos fundamentales de arquitectura web moderna, protección de datos mediante autenticación y manipulación de información mediante servicios REST.

### 3.2 Descripción de Tecnologías, herramientas y metodologías relevantes

Para el desarrollo del presente proyecto, se han empleado diversas tecnologías y herramientas ampliamente utilizadas en el desarrollo web actual. La selección de cada uno fue guiada por criterios como la facilidad de uso, comunidad de soporte, rendimiento, compatibilidad y experiencias previas del desarrollador. En los siguientes apartados, se describen cada una de ellas, junto con su función específica dentro del proyecto.

### 3.2.1 Lenguajes de Diseño Web y Programación

- **HTML5:** Se utiliza para estructurar el contenido de la interfaz web. Permite definir los elementos visuales básicos como tablas, formularios y botones. Fue elegido por ser el estándar actual para el desarrollo Frontend.
- **CSS3:** Se utilizó para dar estilo a la interfaz del usuario. Permite mejorar la experiencia visual del usuario con estilos personalizados, animaciones y diseño limpio sin depender de frameworks externos como Bootstrap.
- **JavaScript:** Se usa para agregar interactividad al Frontend, como la carga dinámica de datos, el manejo de eventos (editar, borrar, agregar) y la comunicación con la API REST. Fue elegido por su compatibilidad con todos los navegadores y su facilidad para integrarse con HTML.

### 3.2.2 Backend y API REST

- **Java con Spring Boot:** Framework utilizado para desarrollar el backend del proyecto donde se expone una API REST que permite realizar operaciones CRUD sobre los empleados. Spring Boot fue seleccionado por su potencia, modularidad, y por ser una tecnología madura en el ámbito empresarial. Además, ofrece

integración sencilla con bases de datos, el uso de dependencias y control de rutas HTTP.

- MySQL: El Sistema de gestión de bases de datos usado para almacenar la información de los empleados y usuarios. Se ha optado por MySQL debido a su estabilidad, rendimiento, fácil uso y soporte para consultas SQL. Además, es ampliamente compatible con PHP y Java.
- PHP: Lenguaje utilizado para implementar el sistema de login. La lógica de autenticación se desarrolla en PHP y se conecta con MySQL para verificar las credenciales. Fue elegido por su facilidad de integración con bases de datos, especialmente para sistemas web sencillos.

### 3.2.3 Herramientas de Desarrollo y Pruebas

- Postman: Herramienta utilizada para realizar pruebas a la API REST desarrollada en Java. Permite enviar peticiones HTTP (GET, POST, PUT, DELETE) y verifica que los endpoints funcionen correctamente. Su interfaz amigable y su capacidad para simular peticiones la convierten en una herramienta estándar para este tipo de tareas.
- XAMPP: Servidor local usado para ejecutar el sistema de login en PHP. Incluye Apache, MySQL y PHP en un solo paquete, lo que facilita el desarrollo y pruebas del login en un entorno controlado.
- Visual Studio Code: Para editar el Frontend (HTML, CSS Y JS) y el login PHP.

- Eclipse: Para programar el backend con Java Spring Boot y gestionar las dependencias del proyecto Java mediante Maven.

### 3.2.4 Metodología de Desarrollo

El proyecto ha sido desarrollado siguiendo un enfoque iterativo y progresivo, inspirado en metodologías ágiles como Scrum, aunque de forma simplificada. Se establecieron tareas semanales con objetivos definidos (semana 1: Crear la base de datos; semana 2: Diseñar el sistema de login; semana 3: Elaborar las APIs o desarrollar la interfaz web; semana 4: Desarrollar la interfaz de usuario para el frontend; semana 5: Realización de pruebas creando, editando y eliminando registros), se hizo un seguimiento del progreso a través de pruebas y revisión de funcionalidades.

Este enfoque permitió realizar mejoras continuas, validar funcionalidades de forma incremental y adaptarse a nuevas ideas durante el desarrollo del proyecto.

En conclusión, la combinación de tecnologías elegida responde a un equilibrio entre accesibilidad, funcionalidad y relevancia académica. La arquitectura cliente-servidor, el uso de APIs REST y la separación de lógica de negocio en distintos lenguajes (PHP para login, Java para backend, JS para frontend) hacen el proyecto un ejemplo completo y realista de una aplicación web.



## 4. Metodología

### 4.1 Descripción detallada del enfoque metodológico utilizado en el Proyecto:

El desarrollo del presente proyecto se ha realizado siguiendo un enfoque metodológico basado en los principios de las metodologías ágiles, concretamente una adaptación simplificada del modelo Scrum. Este tipo de enfoque fue especialmente útil en el contexto de trabajo individual, permitiendo dividir el proyecto en fases manejables, fomentar una planificación iterativa y asegurar la evolución progresiva del sistema.

Scrum se caracteriza por su estructura basada en iteraciones cortas llamadas sprints, y su orientación hacia la entrega continua de valor funcional. Aunque normalmente está pensado para equipos, sus principios pueden aplicarse de manera eficaz en proyectos académicos personales, manteniendo una estructura organizada del trabajo y un enfoque flexible ante posibles mejoras o cambios.

#### 4.1.1 Fases del Proceso y Etapas de Desarrollo

El desarrollo del proyecto se organizó en distintas etapas, cada una con tareas específicas y objetivos definidos. A continuación, se mostrarán las fases más relevantes:

##### **Fase 1: Análisis y Planificación**

- Identificación del problema y definición de los objetivos del proyecto.
- Revisión de la literatura existente sobre aplicaciones de gestión y sistemas de autenticación.
- Selección de tecnologías a utilizar (Java, Spring Boot, PHP, MySQL, HTML, CSS, JavaScript).

- Diseño preliminar de la base de datos y definición de los requerimientos funcionales.

## **Fase 2: Diseño del Sistema**

- Elaboración del modelo entidad-relación de la base de datos.
- Diseño de la arquitectura del sistema con separación de capas: Frontend, backend y base de datos:
- Definición de endpoints para la API RESTful.
- Maquetación inicial de la interfaz de usuario.

## **Fase 3: Implementación del Backend y Login**

- Desarrollo del backend en Java con Spring Boot, con operaciones CRUD sobre empleados.
- Implementación del sistema de login con PHP conectado a MySQL para controlar el acceso.
- Verificación de autenticación antes de permitir el acceso a los registros de empleados.
- Desarrollo y prueba de los endpoints utilizando Postman.

## **Fase 4: Desarrollo del Frontend**

- Construcción de la interfaz web con HTML, CSS puro y JavaScript.
- Implementación de una tabla dinámica que muestra los datos de empleados.
- Desarrollo de formularios para editar, eliminar y agregar nuevos elementos.
- Comunicación entre el Frontend y la API mediante peticiones “fetch()”.

## **Fase 5: Integración y Pruebas**

- Integración de las partes de sistema: login, API REST y frontend.

- Pruebas funcionales completas para verificar el correcto flujo de autenticación y gestión de empleados.
- Corrección de errores detectados en pruebas internas.
- Revisión del comportamiento de la aplicación en diferentes navegadores.

#### **Fase 6: Documentación y Preparación del TFG**

- Redacción de la memoria del proyecto.
- Aplicar lo aprendido.
- Revisión final del contenido.

#### **4.1.2 Gestión del Tiempo y Recursos**

El tiempo se distribuyó de forma equilibrada entre análisis, desarrollo y documentación. El uso de herramientas como Visual Studio Code, Eclipse, XAMPP, MySQL y Postman permitió una gestión eficiente del entorno de desarrollo. Asimismo, la reutilización de conocimientos en PHP y Java facilitó una implementación fluida y eficaz.

Durante el desarrollo de la aplicación intenté implementar “password\_hash()” y “password\_verify()” para la encriptación de contraseñas en PHP para asegurar las contraseñas. Sin embargo, surgieron problemas de compatibilidad o validación y, aunque no llegué a resolverlos del todo a tiempo para la entrega, entiendo perfectamente la importancia de esa práctica y cómo debería haberse implementado correctamente en un entorno real. No obstante, las contraseñas fueron encriptadas correctamente utilizando otro método y no son solo texto plano. Sé que MD5 no es seguro actualmente, pero lo utilicé por simplicidad durante el desarrollo y soy consciente de que en un entorno real esto debe ser reemplazado por algoritmos como bcrypt con password\_hash().

Consideré incluir un sistema de roles para diferenciar entre usuarios normales y administradores, pero por motivos de tiempo me centré en asegurar que las funciones principales estuvieran estables. Es una funcionalidad que se podría añadir fácilmente en el futuro usando roles en la base de datos y validaciones en las vistas, al ser una aplicación cooperativa decidí no implementar roles y que un usuario con credenciales validas pueda acceder sin problemas.

En conclusión, el enfoque metodológico aplicado ha permitido una organización clara y estructurada del trabajo, promoviendo la entrega progresiva de funcionalidades y facilitando el control del desarrollo del proyecto en todo momento. Gracias a esta metodología adaptada, se ha logrado alcanzar los objetivos previstos de manera ordenada, eficiente, clara y dentro del tiempo disponible.

#### 4.1.3 Justificación de la metodología Ágil

Scrum es una metodología ágil que se enfoca en la entrega incremental del producto, donde el desarrollo se organiza en bloques de tiempo los cuales tienen por nombre “sprints”. Aunque normalmente se aplica en equipos de trabajo, sus principios también son útiles para la elaboración de proyectos individuales, ya que fomentan a la planificación clara, control del progreso y la mejora continua. Este enfoque permitió dividir el desarrollo de tareas complejas, priorizar algunas y asegurar una evolución constante del proyecto.

#### 4.1.4 Limitaciones del Método Ágil

Aunque el enfoque de la metodología ágil fue útil de aplicar, trabajar individualmente implicó ciertas limitaciones, como por ejemplo el formulario de validación de usuarios, la gestión de empleados, pero aplicando esta metodología se logró sacar el trabajo adelante.

En pocas palabras, el enfoque metodológico empleado permitió organizar el proyecto de forma clara, estructurada y progresiva, garantizado que todas las funcionalidades previstas fueran implementadas correctamente a tiempo. La adaptación de Scrum a un entorno académico demostró ser eficaz y adecuado para el desarrollo de este proyecto.

#### 4.2 Herramientas y Tecnologías Utilizadas

Para el desarrollo de la aplicación de gestión de empleados se ha hecho uso de un conjunto de herramientas y tecnologías seleccionadas por su eficacia, compatibilidad y facilidad de uso. Estas herramientas han permitido implementar un sistema funcional tanto en el lado del servidor como en el cliente, garantizando una experiencia de usuario fluida, así como una arquitectura robusta y mantenible.

El lenguaje principal utilizado en el backend ha sido Java, debido a su versatilidad, estabilidad y orientación a objetos. Combinado con el framework Spring Boot, ha permitido estructurar una API RESTful capaz de realizar operaciones CRUD sobre los registros de empleados de manera eficiente. Spring Boot ofrece una configuración mínima y una integración rápida con bases de datos, lo que ha facilitado el desarrollo y pruebas del sistema. Por otro lado, para la autenticación de usuarios se ha optado por utilizar PHP, una tecnología muy extendida en

el desarrollo web que ofrece una rápida integración con bases de datos MySQL. PHP ha sido utilizado para validar el acceso a la aplicación, permitiendo que solo usuarios que tengan una cuenta y estén registrados puedan visualizar o modificar los datos de la aplicación.

Para la parte del cliente, el desarrollo de la interfaz web se ha realizado utilizando HTML y CSS, sin recurrir a frameworks como Bootstrap. Esto ha permitido diseñar una interfaz ligera, clara y personalizada, adaptada al propósito del proyecto. El comportamiento dinámico de la interfaz se ha logrado mediante JavaScript, que se ha encargado de enviar peticiones a la API del backend, renderizar los datos en una tabla interactiva, y gestionar los formularios para añadir, editar o eliminar empleados. En su tiempo se evaluó usar Angular para el frontend pero

La base de datos utilizada ha sido MySQL, un sistema de gestión de bases de datos relacional que ofrece alto rendimiento y compatibilidad con las tecnologías seleccionadas. A través de MySQL se han definido las tablas necesarias para almacenar tanto los datos de los empleados como las credenciales de acceso, garantizando la persistencia de la información y permitiendo realizar consultas complejas de forma eficiente.

En cuanto a las herramientas de desarrollo, se ha utilizado Visual Studio Code para la creación del frontend y la lógica en PHP, aprovechando su ligereza, extensiones útiles y su compatibilidad con múltiples lenguajes. Para el desarrollo del backend en Java se ha trabajado con Eclipse, un entorno de desarrollo integrado muy completo que ofrece soporte específico para Spring Boot, autocompletado eficaz, pruebas y depuración avanzada. A su vez, XAMPP ha sido la herramienta empleada para montar un servidor local que integró Apache, PHP y MySQL, permitiendo ejecutar de forma sencilla el sistema de login. Postman ha sido fundamental para la validación de los distintos endpoints REST creados en el backend, permitiendo simular peticiones HTTP y verificar el correcto funcionamiento del API. Por su parte, MySQL ha

facilitado el diseño y gestión de la base de datos a través de una interfaz gráfica que permite modelar tablas, ejecutar scripts SQL y mantener un control visual del esquema.

La elección de estas tecnologías no fue porque sí. Se ha priorizado el uso de herramientas de código abierto, con gran respaldo de la comunidad y bien documentadas. Esta combinación ha permitido desarrollar un sistema completo que integra control de acceso mediante autenticación, manipulación de datos en tiempo real y una interfaz amigable. Las herramientas elegidas han contribuido a optimizar tanto el proceso de desarrollo como los resultados obtenidos, asegurando un correcto cumplimiento de los objetivos planteados en el proyecto.

### 4.3 Detalles sobre el Diseño y la Arquitectura de la Aplicación

El diseño de la aplicación se ha pensado de manera funcional y con un enfoque en la facilidad de uso. Desde el inicio, se estableció una arquitectura que se divide en tres capas definidas: parte visual (frontend), lógica de negocio (backend) y persistencia (base de datos). Esta estructura permite una clara separación de responsabilidades, lo que facilita tanto el mantenimiento como la escalabilidad del sistema.

Respecto a las bases de datos, se ha optado por un diseño simple pero efectivo. Se crearon dos tablas principales: una para los usuarios del sistema, que se encarga de gestionar el inicio de sesión, y otra para los empleados. La tabla de empleados incluye campos como id, nombre, apellido, puesto y codigo\_empleado, lo que permite realizar operaciones CRUD de manera organizada. Por otro lado, la tabla de usuarios contiene campos como id, usuario y contraseña, siendo esta última almacenada e encriptada. La integridad de los datos se asegura mediante claves primarias y validaciones tanto del lado del cliente como del servidor.

La arquitectura del software se ha desarrollado bajo el modelo cliente-servidor. En el lado del servidor, el backend ha sido construido con Java y Spring Boot, ofreciendo una API RESTful que actúa como intermediario entre el cliente y la base de datos. Este enfoque permite al cliente realizar operaciones a través de peticiones HTTP como GET, POST, PUT y DELETE. Además, se ha implementado un sistema de autenticación en PHP que controla el acceso al sistema, asegurando que solo los usuarios que tengan una cuenta puedan acceder a las diferentes características de la aplicación.

La interfaz de usuario (UI) fue diseñada utilizando HTML y CSS, presentando un enfoque especial a la estética, el orden visual y la claridad de los elementos. Se ha evitado el uso de frameworks como Bootstrap para crear un diseño propio, ligero y adaptado a las necesidades específicas de la aplicación. La experiencia de usuario (UX) ha sido lo más primordial, asegurando que la interacción del usuario con la aplicación sea fluida y agradable.

En pocas palabras, el diseño de la aplicación tiene como propósito buscar el equilibrio entre simplicidad, funcionalidad y ser robusto. Las decisiones tomadas en cuanto al tema de arquitectura, estructura de datos y experiencia de usuario responden a criterios prácticos que garantizan un uso accesible y una base sólida para futuras mejoras o integraciones.

#### 4.3.1 Mejoras implementadas y una tabla adicional.

Para que la app no sea tan simple y que no solo consista en usuarios que se loguean y añadir o editar registros, se optó la idea de crear una tabla de tareas que tiene su id, nombre, descripción, fecha de asignación y como clave foránea, el id del empleado asociado, esta mejora permite que el usuario pueda seleccionar un empleado mediante un desplegable y asignarle una tarea, cuando la tarea es asignada puede verlas mediante un botón que dice “Ver tareas”, esto



despliega las tareas que tiene cada empleado, si no tiene tareas aparece que no tiene tareas con la opción de añadirle una, abajo del desplegable aparece una barra de texto el cual puede añadir el nombre de la tarea y asignarla a la persona que corresponda si la tarea fue hecha hay un botón para eliminarla, sino queda como pendiente y con la fecha en la que fue asignada. A su vez, se añadió un desplegable amigable en la que el usuario puede ver su perfil, documentos de la empresa y como punto final un botón para cerrar su sesión que esto destruye la sesión en PHP como `session_destroy()`. Para la parte del login se añadió un botón de “ver contraseña”, para que el usuario pueda ver si metió bien la contraseña o no. Estas mejoras aportan significativamente a la aplicación y no queda como contenido simple y sin objetivos claros.

## 5. Desarrollo e Implementación

### 5.1 Descripción detallada del Proceso de Desarrollo, incluyendo desafíos y soluciones encontradas

El desarrollo de la aplicación se realizó de manera progresiva, siguiendo una planificación de trabajo basada en la metodología Agile, con iteraciones cortas que permitieron realizar pruebas continuas y ajustar funcionalidades según las necesidades. Desde el principio se establecieron objetivos claros y concretos: implementar un sistema de autenticación que permitiera el acceso restringido a usuarios que tengan una cuenta o estén registrados y desarrollar una interfaz de gestión de empleados funcional y fácil de usar.

El primer paso fue diseñar la estructura general del sistema, tanto backend y frontend. Se comenzó con la creación de la base de datos en MySQL, en el cual definimos las tablas necesarias para usuarios y empleados. Durante esta fase, se encontraron ciertos desafíos, como

por ejemplo como gestionar el código de empleado ya que es el identificador, al final se estableció como identificador único para evitar problemas.

Posteriormente se desarrolló el sistema de login utilizando PHP y MySQL. Esta parte incluyó validación de usuario y protección contra accesos no autorizados. Uno de los principales retos fue que el usuario no pudiera acceder a la app si no tenía una cuenta o credenciales. Para ello, se implementaron validaciones de sesión y redirecciones automáticas, logrando un sistema básico pero efectivo en tema de control de acceso.

Una vez establecido el sistema de login, se empezó a desarrollar el backend de la aplicación utilizando Java y Spring Boot. Se realizó una API RESTful que permite realizar operaciones CRUD sobre la entidad “Empleado”. En este punto, se encontraron varios desafíos técnicos durante esta etapa, como por ejemplo al integrar Spring Boot con la base de datos y al diseñar los endpoints para búsquedas específicas por nombre, apellido, puesto y código. Gracias a la documentación de Spring y al uso de herramientas como Postman, fue posible probar los endpoints de manera efectiva y corregir errores con el mapeo de datos y la conversión de tipos.

En el desarrollo del frontend se realizó con HTML, CSS y JavaScript sin necesidad de utilizar frameworks adicionales. Esto presentó algunos retos en cuanto al diseño visual de la app. Para lograr una buena experiencia de usuario, se creó una tabla amigable que muestra todos los empleados registrados y permite, editar o eliminar registros. También se añadieron formularios emergentes para añadir nuevos empleados o modificar los que ya existían, todo con validación básica por parte del lado del cliente. Al no utilizar Bootstrap, el diseño se trabajó a mano, lo cual llevo a más tiempo, pero permitió mayor control estético.

Otro gran desafío fue garantizar la validez de datos al editar o eliminar empleados, se implementaron mensajes de confirmación y se mejoró la gestión de errores para asegurar si hubiera un fallo fuera visible para el usuario y pudiera resolverse rápido.

Finalmente, se realizaron pruebas completas del sistema, incluyendo casos de uso como acceso con credenciales validas y correctas, intentos no autorizados, agregar nuevos empleados, edición y eliminación de datos. Todos los puntos fueron revisados y arreglados hasta alcanzar su estado funcional y estable.

En resumen, el proyecto fue desarrollándose paso a paso, superando los problemas técnicos anteriormente mencionados como validaciones de usuario, conexión entre backend y frontend, control de errores y material visual, hasta lograr y conseguir una aplicación funcional, segura y fácil de usar.

## 6. Resultados

### 6.1 Presentación de la aplicación desarrollada, funciones principales

La aplicación desarrollada es un sistema de gestión de empleados accesible únicamente para usuarios que posean credenciales válidas. Una vez el usuario accede mediante el login, se le redirige al panel principal donde puede visualizar, buscar, añadir, editar, eliminar registros de empleados que no sirvan y cerrar sesión. Todo esto de manera fácil e intuitiva

Entre las funcionalidades importantes se muestra un formulario de login el cual el usuario si tiene cuenta puede acceder a los registros y si no le pide que se cree una, una vez el usuario registrado, se muestra un mensaje de bienvenida con su nombre, una barra de búsqueda para que se le resulte más cómoda buscar un registro y una tabla que muestra todos los empleados

registrados, luego tiene un botón para añadir un empleado el cual abre un formulario para insertar los datos necesarios, el código de empleado es único. Junto a cada empleado y sus datos, se incluyen botones como “editar” y “eliminar” los cuales permiten modificar los datos o eliminar un dato con confirmación previa. Todos los datos se actualizan en tiempo real lo cual permite mayor comodidad sin necesidad de recargar la página. Y en la pestaña de arriba un botón de “Cerrar Sesión” que permite al usuario guardar los cambios y destruir su sesión al darle al botón.

Una característica que destacar es su simplicidad visual, desarrollada sin frameworks de diseño como por ejemplo Bootstrap, lo cual refuerza el dominio del código HTML, CSS y JavaScript. A pesar de todo esto, la interfaz es clara, limpia y funcional, lo que hace que la experiencia del usuario sea agradable.

Para terminar, lo que hace esta aplicación especialmente útil es el enfoque de gestión simple, directa y segura de empleados, ideal para pequeñas empresas, departamentos o responsables de recursos humanos que necesiten tener un control básico de su personal sin depender de sistemas complejos o costosos. Además, combina la autenticación de usuarios y operaciones CRUD completas sobre los empleados, lo que lo hace especialmente útil para simular entornos de gestión realistas. Está desarrollado utilizando tecnologías abiertas y conocidas (PHP, Java Spring Boot, HTML, CSS y JS), lo que también hace que sea fácil de ampliar o modificar.

## 6.2 Resultados de las pruebas y evaluaciones sobre la aplicación

Durante el desarrollo de la aplicación, se realizaron diversas pruebas con el objetivo de garantizar su correcto funcionamiento, la detección de posibles errores, arreglar la experiencia

del usuario. Estas pruebas se dividieron principalmente en tres secciones: funcionales, usabilidad y de rendimiento. A continuación, se mostrarán los objetivos mencionados:

### 6.2.1 Pruebas funcionales

Las pruebas funcionales se enfocan en garantizar que cada funcionalidad principal de la aplicación funcione correctamente. Se verificó lo siguiente: iniciar sesión con credenciales válidas, mostrar una lista completa de empleados, buscar por nombre, apellido, puesto de trabajo o código, crear un nuevo empleado, editar un empleado existente y eliminar permanentemente un registro.

Las pruebas se llevaron a cabo utilizando datos simulados y varios escenarios con campos correctamente completados y entradas en blanco o no válidas. En todos los casos, la aplicación reacciona como se esperaba. El registro en la base de datos se actualiza correctamente, se recibe una respuesta consistente del backend (SpringBoot) y el frontend responde sin errores. Este comportamiento confirma que el punto final está funcionando correctamente y que el flujo de operación CRUD completo está implementado de manera sólida.

### 6.2.2 Pruebas de usabilidad

En este apartado, evaluamos la facilidad de uso de la aplicación realizando pruebas con varios usuarios que no son expertos en el tema. Primero se les pide que creen una cuenta (usuario y contraseña) una vez realizado este proceso, se les pide que inicien sesión, creen un nuevo

empleado y luego lo editen. La mayoría de las personas pudieron completarlo sin ayuda, lo cual refleja que el diseño es intuitivo y fácil de usar.

El panel principal es simple, con botones de manera clara visibles y claramente etiquetados, lo que facilita una comprensión inmediata. La tabla es clara y no muestra demasiada información, una barra de búsqueda con botones para buscar un dato específico y un botón para ver todos los registros. Como mejora, descubrimos que algunos campos podrían incluir una validación visual más clara (por ejemplo, cuando un campo está vacío o mal escrito), aunque esto no impidió que los usuarios realizaran tareas básicas.

### 6.2.3 Pruebas de rendimiento

Aunque la app no fue hecha para un ambiente de alto rendimiento, se hicieron pruebas con una tabla que contenía más de 50 empleados falsos. La carga de la tabla, búsquedas y edición se hicieron muy rápido sin mostrar problemas notables. Al estar creada con herramientas sencillas y tener una estructura simple, su funcionamiento es suficiente para un lugar pequeño o escolar.

## 6.3 Comparaciones con soluciones en el mercado (ventajas y desventajas)

La aplicación desarrollada para la gestión de empleados se coloca en el mercado donde existen diversas soluciones que ofrecen funcionalidades similares. A continuación, se presenta

una comparación con algunas de las herramientas más destacadas en el ámbito de la gestión de RRHH, considerando aspectos como funcionalidades, facilidad de uso, escalabilidad y coste.

Las siguientes soluciones son:

### **1. Endalia**

Endalia es un software integral de RRHH y nómina que abarca todos los procesos relacionados con la gestión de personal. Está orientado a empresas de más de 30 empleados y destaca por su capacidad para gestionar de manera conjunta los procesos de RRHH y la nómina, evitando duplicidades y facilitando la integración de datos.

#### **Ventajas:**

- Facilidad de uso y rápida implantación.

#### **Desventajas:**

- Orientado a empresas medianas y grandes, lo que puede resultar excesivo para pequeñas empresas o proyectos académicos.

### **2. Zoho People**

Zoho People es una solución flexible y personalizable, orientada a pequeñas y medianas empresas. Permite la gestión de diversos procesos de RRHH, incluyendo el seguimiento del tiempo, la gestión de licencias y la evaluación del desempeño.

#### **Ventajas:**

- Interfaz amigable y fácil de usar.

**Desventajas:**

- Algunas funcionalidades avanzadas requieren planes de pago superiores.
- Compleja de usar y no ofrece servicios avanzados

**3. Woffu**

Woffu es una herramienta especializada en el registro horario de los trabajadores, orientada a pequeñas empresas. Ofrece soluciones para la gestión de ausencias, vacaciones y control de presencia.

**Ventajas:**

- Especialización en el control de horario.
- Simplicidad y facilidad de uso.

**Desventajas:**

- Funciones limitadas en comparación con soluciones integrales.
- Puede no cubrir todas las necesidades de gestión de RRHH.

**6.3.1 Comparación con la aplicación desarrollada (ventajas y desventajas)**

La aplicación desarrollada en el presente trabajo se centra en la gestión básica de empleados, incluyendo funcionalidades como el inicio de sesión, visualización de empleados, búsqueda, creación, edición y eliminación de registros. En comparación con las soluciones anteriormente mencionadas:

**Ventajas:**

- Simplicidad y enfoque en funcionalidades esenciales.



- Facilidad de implementación y uso.

**Desventajas:**

- Limitaciones en funcionalidades avanzadas como la gestión de nómina, evaluación del desempeño o integración con otros sistemas.

**Aspectos de mejora:**

- Desarrollar una interfaz más intuitiva y adaptable a dispositivos móviles.

En conclusión, mientras que las soluciones en el mercado ofrecen una amplia gama de funcionalidades y están orientadas a diferentes tamaños de empresas, la aplicación desarrollada se coloca como una herramienta sencilla y funcional, adecuada para pequeñas organizaciones o como una base para futuros desarrollos más complejos.

## 7. Discusión

### 7.1 Análisis de Resultados Obtenidos

Para llevar a cabo el desarrollo de esta aplicación sobre gestión de empleados ha dado lugar a una solución funcional, centrada en la facilidad, accesibilidad y utilidad para el usuario con conocimientos básicos de gestión del personal. A lo largo del proceso, se lograron los siguientes objetivos del proyecto: Un sistema de login seguro, permitir la visualización, edición,

creación y eliminación de registros de empleados, y garantizar que solo los usuarios autorizados puedan acceder a ciertos datos.

Desde el punto de vista técnico, el uso de tecnologías como HTML, CSS, JavaScript, PHP, Java y MySQL ha permitido una integración efectiva entre frontend y backend. El almacenamiento persistente de datos mediante una base de datos relacional ha demostrado ser adecuado, y la lógica implementada para las operaciones CRUD con Java ha funcionado correctamente en todas las pruebas realizadas. La aplicación no se enfrenta a soluciones empresariales sofisticadas, pero sí evidencia que, con recursos escasos, se puede desarrollar un sistema operativo, seguro y práctico. Adicionalmente, su estructura modular permite la opción de expandir o incorporar nuevas funcionalidades, tales como informes o administración por roles.

Para concluir, los resultados alcanzados evidencian una adecuada realización del proyecto, con un producto final operativo, verificado y en concordancia con los objetivos establecidos desde el comienzo. El estudio evidencia que la aplicación cumple de manera apropiada con su objetivo y deja espacio para el futuro avance tecnológico.

## 7.2 Reflexiones sobre limitaciones y posibles áreas de mejora:

Durante el desarrollo del presente proyecto, se identificaron diversas limitaciones, no impidieron alcanzar los objetivos principales, pero llegaron a condicionar ciertos aspectos del resultado final. Una de las principales limitaciones se encuentra el tiempo disponible. Al tratarse de un proyecto académico con una duración determinada, ciertas decisiones debieron tomarse priorizando la funcionalidad sobre la complejidad o profundidad técnica, por ejemplo, la

posibilidad de asignar unos aspectos a la app, como por ejemplo tareas a los empleados o diversas que podría aportar mayor control en estos aspectos.

Respecto a los recursos, el proyecto se llevó a cabo empleando herramientas de código abierto y ambientes locales, lo que resulta beneficioso en cuanto a accesibilidad, pero restringente al simular un ambiente de producción real. No se llevó a cabo una integración constante ni se implementó la aplicación en la nube, lo que habría facilitado la evaluación de factores como el desempeño con varios usuarios al mismo tiempo o la capacidad de expansión de la base de datos. En el tema de roles (admin vs usuario), decidí no implementarlos porque mi idea desde un principio fue elaborar una app cooperativa donde un usuario si tiene credenciales validas pueda acceder a las diferentes funciones del programa. En un futuro se evaluará la posibilidad de añadir roles como por ejemplo el admin podrá acceder a las funciones de la aplicación que es editar, borrar y añadir registros y asignar tareas a empleados, mientras que el usuario solo verá una pagina web de bienvenida a la aplicación y las posibilidades que ofrece hacer.

Entre las posibles áreas de mejora se encuentra también la experiencia de usuario (UX). Aunque se diseñó una interfaz funcional y clara, aún hay margen para mejorar la presentación visual, incorporar retroalimentación visual al usuario (mensajes de éxito o error más visibles), y adaptar la interfaz para ser completamente responsiva y accesible desde dispositivos móviles.

En conclusión, las limitaciones detectadas en este proyecto han sido un componente inevitable del proceso de evolución dentro del ámbito académico. No obstante, también han funcionado como oportunidades de aprendizaje y cimientos para mejoras futuras. La aplicación se ha diseñado de tal manera que posibilita ampliar sus funcionalidades y potenciar sus puntos

débiles en versiones futuras, lo que evidencia una base firme sobre la que continuar construyendo.

### 7.3 Implicaciones prácticas y teóricas de los resultados

Los resultados obtenidos durante el desarrollo de esta aplicación tuvieron implicaciones tanto prácticas como teóricas dentro del campo del desarrollo de software, especialmente para la creación de aplicaciones web.

Desde un punto de vista práctico, la aplicación constituye una alternativa funcional y accesible para pequeñas empresas, instituciones educativas o equipos internos que requieren un sistema simple para administrar personal sin necesidad de utilizar software de terceros más complicado o costoso. La capacidad de solo los usuarios autenticados para ingresar al panel de administración fortalece un principio fundamental de la seguridad en aplicaciones web: limitar el acceso a información delicada. Este tipo de implementación, que puede implementarse en contextos reales con escasa inversión, evidencia que incluso con recursos escasos se puede producir valor y utilidad práctica mediante el desarrollo de software.

Para la parte teórica, el proyecto fortalece principios fundamentales del desarrollo web full stack, incluyendo la distinción entre la lógica empresarial, la presentación y el almacenamiento de datos. Adicionalmente, resalta la relevancia del diseño orientado al usuario (UX) al ofrecer una interfaz práctica y entendible. Además, demuestra que la combinación de tecnologías comunes como PHP, MySQL, Java y JavaScript posibilita obtener resultados robustos y escalables sin requerir herramientas demasiado complejas, lo cual es particularmente significativo en entornos académicos o de capacitación.

En resumen, tanto las enseñanzas prácticas como las enseñanzas teóricas resultantes de este proyecto posibilitan establecerlo como una experiencia educativa integral, con resultados que se pueden aplicar en la vida real y que contribuyen a la comprensión del proceso de creación de aplicaciones web simples, pero funcionales y bien organizadas.

## 8. Conclusiones

### 8.1 Resumen hallazgos claves del Proyecto

El desarrollo de esta aplicación sobre la gestión de empleados permitió lograr los objetivos planteados desde el inicio de este mismo. Uno de los logros a destacar ha sido la implementación de un sistema completo y funcional que permite registrar, editar, eliminar y buscar empleados, accesible únicamente mediante credenciales básicas de usuario, garantizando un nivel básico pero eficaz de seguridad al acceso de datos.

Durante el proceso, los conocimientos obtenidos durante el grado se han aplicado, incluyendo la gestión de bases de datos relacionales, la vinculación entre frontend y backend, y la creación de interfaces intuitivas para el usuario. El programa evidencia que, haciendo uso de tecnologías asequibles como PHP, MySQL, JavaScript, Java y HTML/CSS, se pueden desarrollar soluciones prácticas y escalables para situaciones reales.

Además, la organización del trabajo fue fundamental para llevar acabo este proyecto, se aplicó el uso de metodologías ágiles, lo que permitió estructurar el trabajo en fases claras, identificar rápidamente errores y aplicar mejoras de manera continua y eficaz.

Como resultado, el proyecto no cumplió solo los requisitos técnicos, sino que también ha sido una experiencia de aprendizaje significativa. La aplicación final representa una herramienta práctica para el control interno de empleados, y al mismo tiempo refleja mi capacidad para la planificación, desarrollo y documentar el proyecto completo en el tema software.

## 8.2 Lecciones Aprendidas durante el Desarrollo del Proyecto.

El desarrollo de este proyecto me brindó una experiencia muy valiosa tanto para nivel técnico como personal. Una de las principales lecciones ha sido la importancia de una buena y sólida planificación del trabajo. Aunque al principio parecía sencillo, elaborar todas las etapas ha sido un reto que ha requerido disciplina, constancia y capacidad.

Durante el proceso de este proyecto, aprendí un poco sobre la programación en PHP, reforcé tecnologías como JavaScript y Java, y también el uso de bases de datos MySQL, así como el uso de herramientas para probar APIs como Postman, y lenguajes de diseño web como pueden ser HTML y CSS. También adquirí mayor soltura trabajando con arquitecturas cliente-servidor logrando entender el flujo completo de una aplicación web. Además, al integrar la autenticación de usuarios, he aprendido aspectos básicos de seguridad web y protección de accesos.

Uno de los aprendizajes más significativos ha sido solucionar fallos de manera autónoma, explorando documentación, contrastando fuentes y aplicando soluciones a través de prueba y error. Esto ha potenciado mi independencia como programador y me ha brindado seguridad para manejar dificultades técnicas futuras.

Finalmente, este proyecto me ha demostrado que no todo es programación: la experiencia del usuario, la nitidez visual y la facilidad de uso son elementos esenciales que hacen que una

aplicación sea realmente funcional. Indudablemente, estas enseñanzas me beneficiarán en futuros proyectos académicos o laborales.

## 9. Referencias Bibliográficas

- Agile Alliance. (n.d.). *What is Agile Software Development*. <https://www.agilealliance.org/agile101/>
- Beighley, L., & Morrison, M. (2009). *Head First PHP & MySQL*. O'Reilly Media.
- Bonneau, J., Herley, C., Van Oorschot, P. C., & Stajano, F. (2012). The quest to replace passwords: A framework for comparative evaluation of Web authentication schemes. *IEEE Symposium on Security and Privacy*. <https://doi.org/10.1109/SP.2012.44>
- ClickUp. (2025). *Las 10 mejores herramientas de gestión de empleados para 2025*. <https://clickup.com/es-ES/blog/71465/software-de-gestion-de-empleados>
- Endalia. (2025). *Los 7 mejores software de Recursos Humanos 2025 (Comparativa)*. <https://www.endalia.com/news/7-mejores-software-rrhh/>
- Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures* (Doctoral dissertation, University of California, Irvine). [https://www.ics.uci.edu/~fielding/pubs/dissertation/rest\\_arch\\_style.htm](https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm)
- Flanagan, D. (2020). *JavaScript: The Definitive Guide* (7th ed.). O'Reilly Media.
- GitHub. (n.d.). *GitHub Docs*. <https://docs.github.com/>
- Kumar, R. (2017). *Implementing Effective Employee Management Systems*. Springer.
- Mozilla Developer Network. (n.d.). *JavaScript documentation*. <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- Mordor Intelligence. (2025). *Tamaño del mercado de software de gestión de recursos humanos*. <https://www.mordorintelligence.com/es/industry-reports/hr-management-software-personnel-and-organization-market>
- MySQL. (n.d.). *MySQL 8.0 Reference Manual*. <https://dev.mysql.com/doc/>

- Postman. (n.d.). *Postman Learning Center*. <https://learning.postman.com/>
- Smith, J., & Anderson, P. (2015). Digital transformation of human resources management systems. *Journal of HR Technologies*.
- Sommerville, I. (2016). *Ingeniería del software* (10.<sup>a</sup> ed.). Pearson Educación.
- Spring. (n.d.). *Spring Boot Reference Documentation*. <https://docs.spring.io/spring-boot/docs/current/reference/html/>
- W3Schools. (n.d.). *PHP Tutorial*. <https://www.w3schools.com/php/>
- Zendesk. (2025). *12 mejores software de administración de personal para 2025*. <https://www.zendesk.com.mx/internal-help-desk/workforce-management-software/>