

Análisis de Requisitos

Requisitos Funcionales y No Funcionales

Requisitos Funcionales:

1. **Gestión de Empleados:** La aplicación debe permitir a los usuarios agregar, editar y eliminar empleados.
2. **Generación de Horarios:** Debe ser posible crear, modificar y visualizar los horarios de los empleados.
3. **Control de Fichaje:** Los empleados deben poder fichar su entrada y salida a través de la aplicación.
4. **Autenticación y Autorización:** El sistema debe tener un mecanismo de login seguro para los empleados y administradores.
5. **Notificaciones:** La aplicación debe enviar notificaciones a los empleados sobre cambios en sus horarios.
6. **Informes:** Generación de informes de asistencia y horarios.

Requisitos No Funcionales:

1. **Seguridad:** La aplicación debe asegurar que los datos de los empleados están protegidos contra accesos no autorizados.
2. **Usabilidad:** La interfaz debe ser intuitiva y fácil de usar para usuarios con diferentes niveles de habilidades tecnológicas.
3. **Rendimiento:** La aplicación debe ser capaz de manejar múltiples solicitudes simultáneamente sin degradar su rendimiento.
4. **Compatibilidad:** Debe ser compatible con los principales navegadores web y dispositivos móviles.
5. **Disponibilidad:** El sistema debe estar disponible el 99.9% del tiempo.

Breve Descripción y Diseño de la Arquitectura

Tecnología Empleada:

- **Frontend:** React.js para la construcción de la interfaz de usuario.
- **Backend:** Node.js con Express.js para manejar las solicitudes del servidor.
- **Base de Datos:** MongoDB para el almacenamiento de datos.
- **Autenticación:** JSON Web Tokens (JWT) para la autenticación segura.
- **PWA:** La aplicación será una Progressive Web App para ofrecer una experiencia similar a la de una aplicación nativa.

Descripción de la Arquitectura:

La arquitectura de la aplicación está basada en el modelo cliente-servidor. El cliente (frontend) interactúa con el servidor a través de APIs RESTful. El servidor maneja las operaciones CRUD y la autenticación, mientras que la base de datos almacena la información de los empleados y sus horarios. La implementación PWA permite que la aplicación sea instalable y funcione sin conexión.

Temporalización de Todas las Tareas

Fase de Planificación:

- Definición de Requisitos: 2 días
- Diseño de la Arquitectura: 1 día
- Planificación del Proyecto: 2 días

Fase de Desarrollo:

- Implementación del Frontend: 3 semanas
 - Diseño de la interfaz: 1 semana
 - Desarrollo de componentes principales: 1 semana
 - Integración de funcionalidades: 1 semana
- Implementación del Backend: 3 semanas
 - Configuración del servidor y base de datos: 1 semana
 - Desarrollo de APIs: 1 semana
 - Integración de autenticación: 1 semana

Fase de Pruebas:

- Pruebas Unitarias: 2 días
- Pruebas de Integración: 2 días
- Pruebas de Usuario: 2 días

Fase de Despliegue:

- Configuración del Entorno de Producción: 1 semana
- Despliegue y Monitoreo Inicial: 1 semana

Fase de Documentación:

- Redacción de Manuales de Usuario: 3 días
- Redacción de Documentación Técnica: 3 días

Cumplimiento de la Temporalización

En este apartado se debe describir si se ha podido cumplir la planificación inicial o no y cuáles han sido los obstáculos encontrados.

Durante el desarrollo del proyecto, no se logró cumplir en ningún momento el tiempo objetivo. Factores externos como la familia, el trabajo y algún que otro indeseado (como labeltec) ocuparon una gran parte de ese tiempo.

Además la idea inicial del proyecto, a pesar de ser una buena idea y haberla detallado, no era compatible con los recursos disponibles: 1 integrante y poco tiempo. Así fue como en el último momento empecé de cero con un proyecto más humilde y optimista, teniendo una perspectiva más realista de la carga de trabajo, tomando mejores decisiones.

Admito que el hecho de no querer más integrantes en el equipo no fue sólo por no poder comprometerme con el tiempo debido a mi trabajo, y que una parte de mí pensaba que podía hacerlo sólo y que trabajar con otros compañeros iba a dar más problemas que beneficio. Me equivocaba. Ahora soy consciente de que la mejor decisión era formar un grupo y realizar un proyecto realista que no cumpla con las expectativas fantasiosas que tuve al embarcarme en este proyecto.

Principales Problemas al Desarrollar el Sistema

1. **Integración de Autenticación:** Se encontraron dificultades para implementar el sistema de autenticación de JWT debido a errores en la gestión de tokens.
2. **Pruebas de Usuario:** Algunos usuarios reportaron problemas de usabilidad que no fueron previstos inicialmente, lo que requirió revisiones en la interfaz.
3. **Compatibilidad PWA:** Hubo desafíos en asegurar que la aplicación funcionara correctamente como PWA en diferentes dispositivos y navegadores.

Estos problemas se abordaron mediante un proceso iterativo de pruebas y corrección, asegurando finalmente una solución robusta y eficiente.