

Plan de Proyecto

Sistema de Gestión de Horarios Académicos

Universidad Industrial de Santander

Autores:

Diego Medina

Jesús Ramírez

Janer Vega

Fecha: 3 de mayo de 2024

Índice

1. Fase 1: Planificación y Diseño	2
1.1. Actividades	2
1.2. Tiempo estimado	2
2. Fase 2: Configuración del Entorno y Estructura del Proyecto	2
2.1. Actividades	2
2.2. Tiempo estimado	2
3. Fase 3: Implementación del Modelo de Datos e inicio del desarrollo de las vistas	3
3.1. Actividades	3
3.2. Tiempo estimado	3
4. Fase 4: Desarrollo de la Capa de Servicio	3
4.1. Actividades	3
4.2. Tiempo estimado	3
5. Fase 5: Implementación de Controladores, Endpoints API y navegación	3
5.1. Actividades	3
5.2. Tiempo estimado	4
6. Fase 6: Seguridad, autenticación e integración	4
6.1. Actividades	4
6.2. Tiempo estimado	4
7. Fase 7: Pruebas	4
7.1. Actividades	4
7.2. Tiempo estimado	4
8. Fase 8: Documentación, optimización y despliegue	4
8.1. Actividades	4
8.2. Tiempo estimado	5
9. Fase 9: Retroalimentación y Mejoras	5
9.1. Actividades	5
9.2. Tiempo estimado	5

1. Fase 1: Planificación y Diseño

1.1. Actividades

- Definir requisitos detallados del sistema, incluyendo las funcionalidades del back-end.
- Diseñar el modelo de datos para la base de datos, considerando las entidades relacionadas con la gestión de horarios académicos.
- Especificar la arquitectura del back-end, determinando las tecnologías a utilizar y los endpoints de la API necesarios para la gestión de horarios.
- Definir y estudiar las funcionalidades que tendrá la experiencia de usuario del aplicativo.
- Diseñar cada una de las vistas del aplicativo web en Figma.

1.2. Tiempo estimado

1-2 semanas.

2. Fase 2: Configuración del Entorno y Estructura del Proyecto

2.1. Actividades

- Configurar el entorno de desarrollo, incluyendo la instalación del JDK, MySQL, Angular, Node.js, NPM y el IDE necesario.
- Generar el proyecto base utilizando Spring Initializer y configurar las dependencias necesarias.
- Generar el proyecto usando Angular CLI y crear los módulos principales para el proyecto.
- Configurar Visual Studio Code para hacer más eficientes algunos aspectos del desarrollo.
- Organizar la estructura del proyecto en paquetes, siguiendo las mejores prácticas de diseño.

2.2. Tiempo estimado

1 semana.

3. Fase 3: Implementación del Modelo de Datos e inicio del desarrollo de las vistas

3.1. Actividades

- Definir las entidades JPA relacionadas con la gestión de horarios académicos y mapearlas a la base de datos.
- Crear los repositorios para el acceso a datos, permitiendo realizar operaciones CRUD sobre las entidades relacionadas con los horarios.
- Configurar la conexión a la base de datos MySQL en el back-end.
- Desarrollar las vistas principales de aplicativo.

3.2. Tiempo estimado

2 semanas.

4. Fase 4: Desarrollo de la Capa de Servicio

4.1. Actividades

- Implementar la lógica de negocio en servicios, definiendo las operaciones necesarias para la gestión de horarios.
- Integrar los servicios con los repositorios para acceder a los datos de la base de datos.
- Realizar pruebas unitarias para verificar el correcto funcionamiento de los servicios.
- Desarrollar la lógica de negocio en los componentes usando TypeScript.

4.2. Tiempo estimado

2-3 semanas.

5. Fase 5: Implementación de Controladores, End-points API y navegación

5.1. Actividades

- Desarrollar controladores REST para manejar las solicitudes HTTP relacionadas con la gestión de horarios académicos.
- Implementar la lógica necesaria en los controladores para procesar las solicitudes y enviar las respuestas adecuadas.
- Realizar pruebas de integración para verificar la funcionalidad completa de la API.
- Definir las rutas de la aplicación y configurar el enrutador de Angular.

- Implementar la navegación entre las diferentes vistas y componentes.

5.2. Tiempo estimado

3-4 semanas.

6. Fase 6: Seguridad, autenticación e integración

6.1. Actividades

- Configurar Spring Security para proporcionar autenticación y autorización en la aplicación.
- Implementar mecanismos de seguridad para proteger los endpoints de la API relacionados con la gestión de horarios.
- Realizar pruebas de seguridad para asegurar la protección adecuada de los recursos.
- Integrar el aplicativo con el back-end y API para obtener y enviar datos.

6.2. Tiempo estimado

2 semanas.

7. Fase 7: Pruebas

7.1. Actividades

- Escribir pruebas unitarias para validar el funcionamiento de cada componente tanto del frontend como del backend.
- Realizar pruebas de integración para verificar la interacción entre los diferentes módulos del sistema.
- Ejecutar pruebas de rendimiento para evaluar la capacidad de respuesta y estabilidad del sistema bajo carga.
- Identificar y corregir errores (bugs) en el sistema.

7.2. Tiempo estimado

2-3 semanas.

8. Fase 8: Documentación, optimización y despliegue

8.1. Actividades

- Documentar la API con Swagger o Springdoc OpenAPI para facilitar su comprensión y uso.
- Preparar el entorno de producción y desplegar la aplicación en un servidor adecuado.

- Crear un manual de instalación y un manual de usuario para ayudar en el despliegue y el uso del sistema.
- Mejorar el rendimiento del aplicativo mediante técnicas como el lazy loading, optimización de imágenes, etc.

8.2. Tiempo estimado

2 semanas.

9. Fase 9: Retroalimentación y Mejoras

9.1. Actividades

- Recoger feedback de los usuarios para identificar áreas de mejora en el sistema.
- Realizar ajustes y mejoras en función de la retroalimentación recibida, priorizando las características más solicitadas.
- Realizar pruebas finales para asegurar que las mejoras implementadas no afecten negativamente al funcionamiento del sistema.

9.2. Tiempo estimado

Variable, según necesidades.