

Nombre y apellidos del alumno: Adrian Martin Pereira

Título del proyecto: Zenit (Provisional)

## 1. Descripción.

El proyecto consistiría en la creación de una plataforma web de gestión de proyectos orientada a facilitar el trabajo en equipo y la organización de tareas dentro de proyectos. La plataforma permitirá a los usuarios realizar las siguientes acciones:

- Gestión de Proyectos: Crear y editar proyectos, asignar tareas a miembros del equipo y gestionar su progreso
- Gestión de tareas: Crear, asignar, editar y eliminar tareas dentro de los proyectos. Cada tarea tendrá un estado (pendiente, en progreso, completada).
- Comunicación en tiempo real: Los miembros podrán comentar en las tareas dentro de los proyectos proporcionando actualizaciones y facilitando la colaboración.
- Notificaciones: El sistema enviara notificaciones de cambios importantes, como tareas asignadas, actualizaciones o fechas limite.
- Roles de usuario: Habrá al menos dos roles:
  - Administrador: Gestión completa de proyectos, tareas y usuarios.
  - Miembro: Acceso para gestionar las tareas asignadas, pero sin privilegios para modificar proyectos o gestionar otros usuarios.

La plataforma se desarrollará utilizando:

- **Frontend:**
  - HTML, CSS y JavaScript para diseñar y desarrollar una interfaz de usuario intuitiva y dinámica.
  - Uso de tecnologías modernas como React, Vue.js, o Vanilla JS para la gestión de componentes y mejorar la interactividad.
- **Backend:**
  - Node.js con Express.js, que proporcionará la lógica del servidor y una API RESTful para gestionar proyectos, tareas y usuarios.
  - Manejo de autenticación y autorización mediante JSON Web Tokens (JWT) para garantizar la seguridad.
- **Base de datos:**
  - MySQL para almacenar los datos relacionados con proyectos, tareas y usuarios.
  - Uso de Sequelize (ORM) o Knex.js (Query Builder) para interactuar con la base de datos y gestionar relaciones entre tablas (proyectos, tareas y usuarios).

## 2. Contenidos curriculares.

### Desarrollo Frontend:

- Diseño e implementación de interfaces con HTML y CSS.
- Interactividad mediante JavaScript:
  - Manipulación del DOM y manejo de eventos.
  - Comunicación con el backend mediante fetch() para consumir APIs RESTful.
  - Uso de frameworks o librerías modernas como React o Vue.js para una interfaz dinámica y reactiva.

### Desarrollo Backend:

- Implementación de una API RESTful utilizando Node.js con Express.js:
  - Gestión de proyectos, tareas y usuarios.
  - Endpoints para realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar).
- Manejo de roles y permisos de usuarios:
  - Implementación de middleware en Node.js para controlar accesos según el rol (Administrador o Miembro).
- Validación y gestión de formularios:
  - Validación de datos enviados desde el frontend en el backend utilizando herramientas como Joi o lógica personalizada en Node.js.
- Implementación de notificaciones:
  - Notificaciones en tiempo real utilizando Socket.IO para eventos como asignación de tareas, actualizaciones o fechas límite.
- Manejo de sesiones y autenticación:
  - Uso de JSON Web Tokens (JWT) para la autenticación y autorización de usuarios.

### Base de Datos:

- Diseño y gestión de base de datos con MySQL
  - Creación de tablas en SQL para almacenar proyectos, tareas y usuarios.
  - Definición de relaciones entre los datos:
    - Un proyecto puede tener muchas tareas.
    - Un usuario puede estar asignado a muchas tareas.

## Comunicación en tiempo real

- Implementación de WebSockets con Socket.IO para funciones como:
  - Actualizaciones en tiempo real de comentarios en tareas.
  - Envío inmediato de notificaciones a los usuarios.

### 3. Objetivos.

- Objetivo Principal: Desarrollar una plataforma web sencilla y funcional para la gestión de proyectos, tareas y equipos de trabajo, con un sistema de roles y permisos.
- Objetivos Específicos:
  1. Implementar una interfaz visual intuitiva para gestionar proyectos y tareas.
  2. Implementar la capacidad de asignar tareas a miembros del equipo y seguir su progreso.
  3. Crear una base de datos estructurada para almacenar proyectos, tareas y usuarios.
  4. Desarrollar un sistema de notificaciones para alertar sobre cambios importantes.
  5. Asegurar que la plataforma sea accesible desde cualquier dispositivo moderno.

### 4. Recursos necesarios.

#### Hardware:

- Ordenador personal para desarrollo.
- Acceso a un servidor web (local o remoto) para pruebas y despliegue.

#### Software:

- Node.js y Express.js para el desarrollo backend.
- MySQL como sistema de gestión de bases de datos.
- HTML, CSS y JavaScript para el desarrollo frontend.
- IDE Visual Studio Code para la edición de código.
- Plataforma para gestión de versiones como GitHub.

#### Recursos Adicionales:

- Herramienta de diseño como Figma para crear un diseño preliminar de la interfaz.
- Sequelize o Knex.js para interactuar con la base de datos MySQL desde el backend.
- Postman o herramienta similar para probar y documentar las APIs RESTful.

## 5. Resultados esperados.

El resultado esperado es el desarrollo de un prototipo funcional que permita:

- La creación y gestión de proyectos.
- La asignación y gestión de tareas dentro de proyectos.
- La implementación de roles (administrador y miembro) con permisos diferenciados.
- Un sistema de notificaciones para mantener a los usuarios informados sobre cambios importantes.
- Un diseño visualmente intuitivo y accesible.

Este prototipo podrá servir como base para futuras ampliaciones, como la implementación de nuevas funcionalidades o mejoras en el rendimiento.

## 6. Temporalización.

### Fase 1 - Planificación y Diseño (20 horas)

- Definición de requisitos funcionales y no funcionales. (6 horas)
- Investigación y planificación de funcionalidades adicionales futuras. (6 horas)
- Diseño de la estructura de la base de datos y arquitectura del sistema. (8 horas)

---

### Fase 2 - Desarrollo Backend con Node.js y MySQL (40 horas)

- Instalación y configuración de Node.js, Express.js y MySQL. (4 horas)
- Creación de la API RESTful para proyectos, tareas y usuarios, incluyendo operaciones CRUD. (20 horas)
- Implementación de relaciones entre proyectos, tareas y usuarios. (6 horas)
- Validaciones avanzadas para datos de entrada. (6 horas)
- Testeo inicial de la API. (4 horas)

---

### Fase 3 - Desarrollo Frontend con JavaScript (30 horas)

- Diseño de la interfaz de usuario con enfoque en usabilidad y responsive design. (10 horas)
- Implementación de formularios para gestión de proyectos y tareas con validación en frontend. (10 horas)
- Desarrollo de vistas dinámicas para mostrar proyectos y tareas, utilizando DOM y manipulación de eventos. (10 horas)

## Fase 4 - Roles y Permisos (20 horas)

- Implementación de un sistema básico de roles (administrador y miembro). (8 horas)
  - Restricción de acceso a funcionalidades según el rol del usuario. (8 horas)
  - Validación de roles en el frontend y backend. (4 horas)
- 

## Fase 5 - Notificaciones (15 horas)

- Implementación de un sistema básico de notificaciones internas. (6 horas)
  - Opcional: Notificaciones por correo utilizando nodemailer. (6 horas)
  - Testeo del sistema de notificaciones. (3 horas)
- 

## Fase 6 - Pruebas y Debugging (20 horas)

- Pruebas de integración del sistema completo (frontend y backend). (10 horas)
  - Realización de pruebas de usabilidad para mejorar la experiencia de usuario. (6 horas)
  - Corrección de errores y optimización del sistema. (4 horas)
- 

## Fase 7 - Despliegue y Documentación (15 horas)

- Despliegue de la aplicación en un servidor con Docker. (5 horas)
  - Redacción de la documentación técnica (API, instrucciones de instalación). (5 horas)
  - Creación de un manual de usuario con ejemplos visuales. (5 horas)
- 

## Fase 8 - Ajustes y Mejoras (10 horas)

- Implementación de mejoras según el feedback recibido. (6 horas)
  - Test de estrés para asegurar la capacidad del sistema con varios usuarios. (4 horas)
- 

**Total estimado: 150 horas**

### 7. Tutor(a) individualizado(a).

Me gustaría contar con Alberto como tutor, ya que durante el curso ha dado bastante retroalimentación en los trabajos hechos y ha ofrecido una buena orientación. Además, dado que el proyecto estará completamente enfocado en JavaScript me vendría bien su experiencia y ayuda al desarrollar el mismo.



## Plantilla propuesta de proyecto

En Valladolid, 10 de Enero de 2025.

Fdo. D. Adrián Martín Pereira

Alumno de 2º DAW