# Imagen que contiene Logotipo Descripción generada automáticamente

# Plataforma de Cursos Online

*by: Javier Berenguer Peidro, Miguel Gutierrez Rico & Alejandro Alberto Jiménez Brundin*

## 1. Documento de Requisitos del Proyecto: Plataforma de Cursos Online

### 1. Título del Proyecto:

**Plataforma de Cursos Online**

### 2. Introducción:

Plataforma de Cursos Online: Administrar cursos, estudiantes y profesores. Los cursos pueden tener varios estudiantes (ManyToMany) y un profesor asignado (ManyToOne). La plataforma permitirá gestionar el contenido educativo, las inscripciones y el progreso de los estudiantes.

### 3. Objetivos del Proyecto:

#### 3.1. Objetivo General:

#### Facilitar el aprendizaje y la gestión de cursos mediante una plataforma intuitiva que permita la interacción entre estudiantes, profesores y administradores.

#### 3.2. Objetivos Específicos:

* Diseñar una interfaz intuitiva y visualmente atractiva para garantizar una experiencia de usuario óptima.
* Permitir la creación de perfiles individuales para estudiantes, profesores y administradores.
* Implementar funcionalidades para inscripción en cursos y gestión de contenidos.
* Proporcionar herramientas para evaluar el progreso de los estudiantes mediante calificaciones y reportes.
* Garantizar la seguridad y privacidad de los datos de los usuarios.

### 4. Requisitos Funcionales:

#### 4.1. Gestor de Usuarios:

* Registro de usuarios con validación de correo electrónico.
* Creación de perfiles de estudiantes con datos como nombre, correo, cursos inscritos y progreso.
* Creación de perfiles de profesores con datos como nombre, área de especialización y cursos asignados.
* Opciones para actualizar y eliminar perfiles.

#### 4.2. Gestión de cursos:

* Creación de cursos con información como título, descripción, duración y recursos adjuntos.
* Asignación de profesores a los cursos.

#### 4.3. Evaluación y seguimiento:

* Creación y evaluación de tareas y exámenes.
* Asignación de profesores a los cursos.
* Inscripción de estudiantes en cursos.

#### 4.4. Comunicación:

* Sistema de mensajería entre estudiantes y profesores.
* Notificaciones sobre actualizaciones en los cursos o fechas importantes.

#### 4.5. Administración:

* Gestión de roles y permisos para administradores, profesores y estudiantes.
* Moderación de contenido subido por los usuarios.

### 5. Requisitos No Funcionales:

#### 5.1. Seguridad:

* Autenticación mediante contraseñas cifradas y sistemas de recuperación.
* Políticas de privacidad estrictas para proteger los datos personales.

#### 5.2. Rendimiento:

* Respuesta de las páginas en menos de 2 segundos bajo condiciones normales de carga.
* Escalabilidad para manejar hasta 10,000 usuarios simultáneos.

#### 5.3. Compatibilidad:

* Diseño responsivo para dispositivos móviles, tablets y ordenadores.
* Compatibilidad con navegadores principales como Chrome, Firefox y Safari.

#### 5.4. Usabilidad:

* Interfaz gráfica con accesibilidad para usuarios con discapacidades.
* Sistema de ayuda y tutoriales integrados.

### 6. Restricciones:

* La plataforma debe ser compatible con sistemas operativos Android e iOS para aplicaciones móviles.
* El desarrollo inicial está limitado a un presupuesto de 10,000 € y un plazo de 6 meses.
* El equipo de desarrollo consta de tres personas: un desarrollador frontend, un desarrollador backend y un coordinador de proyecto.

### 7. Suposiciones Iniciales:

* Los usuarios tendrán acceso a una conexión a Internet estable para utilizar la plataforma.
* El contenido generado por usuarios se mantendrá en línea con las políticas de comunidad.
* El equipo de soporte y moderación será externo y operará después del lanzamiento del MVP.

## 2. Plan de proyecto: Plataforma de Cursos Online

### 1. Título del Proyecto:

**Plataforma de Cursos Online**

### 2. Objetivo del Proyecto:

Planificar y organizar el desarrollo de una plataforma social enfocada en dueños de mascotas, asegurando una ejecución eficiente y cumplimiento de plazos y recursos definidos.

### 3. Cronograma del Proyecto:

El desarrollo del proyecto se dividirá en tres fases principales, cada una con sus respectivas actividades y duración estimada:

#### Fase 1: documentación (1 semana):

* Confirmar el nombre del proyecto
* Completar el documento de requisitos
* Confirmar las especificaciones técnicas

#### Fase 2: Desarrollo (1 mes):

* Diseño de la arquitectura del sistema.
* Creación de la interfaz de usuario (frontend).
* Implementación del backend y servicios.
* Integración de funcionalidades clave (creación de perfiles, creación de cursos, publicación de contenido, y alertas).

#### Fase 3: Despliegue (1 semana):

* Pruebas para comprobar la funcionalidad.
* Implementación del sistema en servidores en la nube.

### 4. Hitos del Proyecto:

1. **Semana 1:** Finalización de la documentación
2. **Semana 3:** Finalización del diseño de la interfaz y la arquitectura.
3. **Semana 4:** Finalización de pruebas y correcciones.

### 5. Recursos del Proyecto:

#### 5.1. Humanos:

* **Desarrollador Frontend:** Responsable del diseño y funcionalidad visual de la aplicación.
* **Desarrollador Backend:** Encargado de implementar la lógica del sistema y la integración con la base de datos.

#### 5.2. Tecnológicos:

* Lenguajes: HTML, CSS, JavaScript, Java, Bootstrap
* Frameworks: Spring Boot, React.
* Herramientas de gestión: GitHub, Trello.
* Infraestructura:H2 database.

#### 5.3. Económicos:

* Presupuesto inicial de 10,000 € para servidores, licencias de software y otros gastos relacionados.

### 6. Roles y Responsabilidades:

#### Desarrollador Frontend:

* Diseñar y construir la interfaz de usuario.
* Asegurar la compatibilidad con diferentes dispositivos y navegadores.

#### Desarrollador Backend:

* Diseñar la arquitectura de la base de datos.
* Implementar las lógicas de negocio.
* Asegurar la seguridad y rendimiento del sistema.

### 7. Indicadores de Éxito:

* Completar el prototipo dentro del tiempo y presupuesto asignado.
* Recibir feedback positivo sobre la experiencia de usuario y funcionalidades.
* Obtener una buena nota durante la fase inicial.

### 8. Suposiciones y Dependencias:

#### Suposiciones:

* El equipo contará con acceso continuo a las herramientas y recursos necesarios.
* Los desarrolladores cumplirán con los plazos acordados.

#### Dependencias:

* Disponibilidad de un entorno de pruebas.
* Acceso a servicios necesarios para el despliegue.

## 3. Especificaciones técnicas: Plataforma de Cursos Online

### 1. Arquitectura:

Hemos definido claramente los requisitos funcionales y no funcionales basándonos en las necesidades de los usuarios. Esto incluye la gestión de cursos, inscripciones, y perfiles de estudiantes y profesores.

### 2. Diseño de la arquitectura:

Decidimos utilizar una arquitectura basada en MVC (Modelo-Vista-Controlador) para separar la lógica de negocio, la interfaz de usuario y la base de datos. Esta arquitectura nos permite escalar y mantener el sistema de forma más sencilla

### 3. Modelo de Datos Eficiente:

Diseñamos un esquema de base de datos que incluye entidades como cursos, estudiantes, profesores y evaluaciones. Consideramos relaciones como ManyToMany para inscripciones de estudiantes a cursos y ManyToOne para asignar un profesor a cada curso.

### 4. Seguridad:

* Autenticación y Autorización: Implementamos un sistema de roles para gestionar permisos de estudiantes, profesores y administradores, utilizando JWT para sesiones seguras.
* Protección de Datos Sensibles: Encriptamos contraseñas y aplicamos buenas prácticas para proteger la información personal.
* Prevención de Ataques Comunes: Implementamos medidas contra inyecciones SQL, XSS y CSRF para garantizar la seguridad del sistema.

### 5. Interfaz de usuario:

Diseñamos una interfaz centrada en la experiencia del usuario, con dashboards claros y herramientas accesibles para estudiantes y profesores. Nos aseguramos de que sea responsiva para funcionar correctamente en dispositivos móviles y ordenadores.