**Documentación Mega3D**

**por Quinteros R Mateo**

**Descripción General**

**Mega3D** es una plataforma web diseñada para la distribución y gestión de cursos en 3D a través de un modelo de suscripción. El proyecto está concebido para brindar una experiencia de usuario envolvente, permitiendo a los usuarios explorar y aprender mediante cursos que incluyen lecciones detalladas, materiales visuales en 3D, y otros recursos educativos. La aplicación es accesible tanto para usuarios regulares como para administradores, cada uno con sus propias vistas y funcionalidades.

El sistema está construido sobre una arquitectura cliente-servidor que separa claramente la lógica del frontend y el backend. El frontend está diseñado para ser responsive, asegurando que los usuarios puedan acceder a la plataforma desde cualquier dispositivo, mientras que el backend se encarga de gestionar las operaciones de la base de datos y las comunicaciones entre el servidor y los clientes.

A continuación, se ofrece una descripción completa de las funcionalidades del proyecto, la estructura del sistema, y un Esquema de Requerimientos del Sistema (ERS) que cubre tanto los aspectos técnicos como funcionales de Mega3D.

**Requerimientos Funcionales**

**1. Página Principal (Vista Pública):**

* **Descripción:** La página principal sirve como la puerta de entrada a la plataforma Mega3D. En esta vista, los usuarios pueden obtener una visión general de la plataforma, explorar los cursos disponibles, y entender las ventajas de suscribirse.
* **Características:**
  + **Listado de Cursos:** Muestra un listado de cursos, categorizados por temas o categorías (por ejemplo, animación, modelado 3D, etc.). Los usuarios pueden filtrar cursos según sus intereses.
  + **Búsqueda Dinámica:** Los usuarios pueden utilizar una barra de búsqueda para encontrar cursos específicos. La búsqueda se realiza en tiempo real, mostrando resultados instantáneamente.
  + **Información sobre Suscripciones:** La página también contiene información detallada sobre los planes de suscripción disponibles, destacando las ventajas de cada plan.
  + **Acceso Rápido al Registro/Inicio de Sesión:** Botones de acceso para iniciar sesión o registrarse, asegurando que los usuarios puedan ingresar fácilmente a sus cuentas.

**2. Página de Detalles del Curso:**

* **Descripción:** Los usuarios pueden hacer clic en un curso específico para ver más detalles, incluyendo una descripción completa, las lecciones incluidas, y el material adicional disponible.
* **Características:**
  + **Descripción Completa del Curso:** Incluye información como el nombre del instructor, la duración del curso, el número de lecciones, y una descripción general.
  + **Vista Previa de Lecciones:** Los usuarios pueden obtener una vista previa de algunas lecciones, incluso sin estar registrados. Esto permite a los usuarios evaluar el contenido antes de suscribirse.

**3. Autenticación de Usuarios:**

* **Descripción:** Mega3D permite a los usuarios registrarse y autenticarse para acceder al contenido, para proporcionar acceso a las funciones restringidas para usuarios sin registrarse.
* **Características:**
  + **Registro de Nuevos Usuarios:** Los usuarios pueden crear una cuenta proporcionando su correo electrónico, nombre de usuario, y contraseña. La plataforma incluye validación de datos en el lado del cliente y servidor para asegurar la integridad de los datos.
  + **Inicio de Sesión:** Los usuarios registrados pueden autenticarse proporcionando su correo electrónico y contraseña. Se implementan medidas de seguridad como hash de contraseñas y sesiones seguras para proteger la información del usuario.

**4. Gestión de Usuarios y Cursos para Administradores:**

* **Descripción:** Los administradores tienen acceso a vistas y funcionalidades exclusivas que les permiten gestionar el contenido de la plataforma, incluyendo la creación, edición y eliminación de cursos, lecciones y categorías.
* **Características:**
  + **Gestión de Cursos:** Los administradores pueden añadir nuevos cursos, editar los existentes, o eliminarlos. Esto incluye la capacidad de subir material adicional, modificar descripciones, y reorganizar el contenido.
  + **Gestión de Lecciones:** Los administradores pueden añadir o modificar lecciones dentro de los cursos, ordenarlos de posición y eliminarlos asegurando que el contenido esté siempre actualizado y relevante.
  + **Gestión de categorías:** los administradores pueden hacer CRUD a las categorías en el panel de control.

**5. Vista de Cursos Suscritos:**

* **Descripción:** Los usuarios suscritos pueden acceder a una vista personalizada de los cursos en los que están matriculados, permitiendo un fácil seguimiento de su progreso y acceso rápido al contenido.
* **Características:**
  + **Progreso del Curso:** Los usuarios pueden ver su progreso dentro de cada curso, incluyendo qué lecciones han completado y cuáles están pendientes.
  + **Acceso Directo a Lecciones:** El menú permite acceder directamente a cualquier curso de tu lista, sin necesidad de navegar a través de múltiples páginas.

**Especificación de Requerimientos del Sistema (ERS)**

* **Arquitectura del Sistema:** Mega3D está basado en una arquitectura cliente-servidor, donde el frontend y el backend están claramente separados para mejorar la escalabilidad y el mantenimiento del sistema. El frontend se comunica con el backend mediante una REST API, que maneja todas las operaciones CRUD (Crear, Leer, Actualizar, Eliminar).
* **Tecnologías del Frontend:**
  + **HTML y CSS:** Utilizados para estructurar y diseñar la interfaz de usuario. El diseño es responsive para asegurar que la plataforma sea accesible desde dispositivos móviles y de escritorio.
  + **JavaScript:** Se utiliza para manejar la interacción del usuario con la plataforma, incluyendo la validación de formularios, la navegación dinámica, y las peticiones asíncronas al backend.
* **Tecnologías del Backend:**
  + **Node.js:** El backend está implementado en Node.js, utilizando Express.js para gestionar las rutas y middleware.
  + **Base de Datos:** La persistencia de los datos se gestiona mediante una base de datos relacional en Postgres. El sistema incluye modelos para usuarios, cursos, lecciones y categorías, asegurando que los datos estén normalizados y sean fácilmente accesibles.
  + **REST API:** El backend expone una serie de endpoints REST que permiten al frontend realizar operaciones como registrar usuarios, autenticar sesiones, y gestionar el contenido de los cursos.

**Estructura Detallada del Proyecto**

**Backend:**

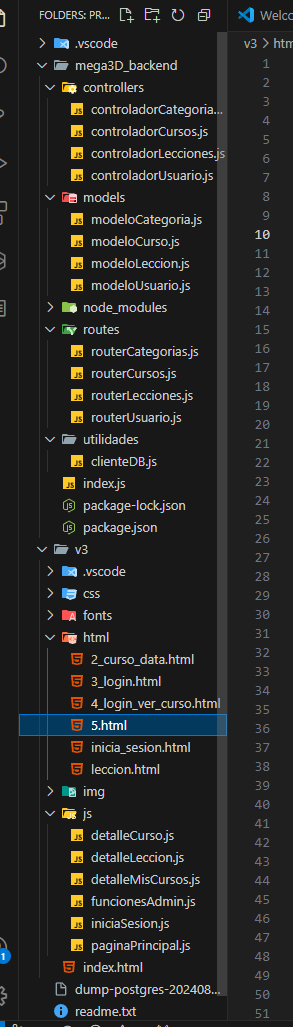
* **controllers:** La carpeta controllers contiene los controladores que gestionan la lógica de negocio y la interacción con los modelos de datos. Estos controladores son responsables de coordinar la interacción entre las rutas y los modelos.
  + **controladorCategorias.js:** Gestiona todas las operaciones relacionadas con las categorías de cursos, como crear nuevas categorías o listar las existentes.
  + **controladorCursos.js:** Responsable de manejar la creación, actualización, eliminación, y consulta de cursos. Este controlador interactúa directamente con el modelo de curso para asegurar la coherencia de los datos.
  + **controladorLecciones.js:** Maneja las operaciones relacionadas con las lecciones dentro de un curso. Permite a los administradores añadir nuevas lecciones, editar las existentes, y reorganizar el orden de las lecciones.
  + **controladorUsuario.js:** Gestiona las operaciones relacionadas con los usuarios, incluyendo el registro, autenticación, y la gestión de sesiones de usuario.
* **models:**
  + **modeloCategoria.js:** Define la estructura de los datos para las categorías de cursos, incluyendo atributos como el nombre de la categoría y su descripción.
  + **modeloCurso.js:** Define los atributos de un curso, como el título, la descripción, la categoría a la que pertenece, y las lecciones asociadas.
  + **modeloLeccion.js:** Define la estructura de una lección, incluyendo el título, el contenido, y su asociación con un curso específico.
  + **modeloUsuario.js:** Define los atributos de los usuarios registrados en la plataforma, incluyendo su nombre, correo electrónico, contraseña (encriptada), y el estado de la suscripción.
* **routes:** Las rutas definen los endpoints de la API que los clientes (frontend o aplicaciones móviles) pueden utilizar para interactuar con la plataforma. Cada ruta está asociada con un controlador que gestiona la lógica detrás de las solicitudes HTTP.
  + **routerCategorias.js:** Define las rutas para operaciones CRUD relacionadas con categorías.
  + **routerCursos.js:** Contiene las rutas para la gestión de cursos, incluyendo la creación de nuevos cursos, actualización de información, y eliminación de cursos.
  + **routerLecciones.js:** Define las rutas para la gestión de lecciones dentro de un curso.
  + **routerUsuario.js**: Contiene las rutas para la gestión de usuarios, como registro, autenticación, y recuperación de contraseñas.
* **utilidades:**
  + **clienteDB.js:** Este archivo contiene la configuración de la conexión a la base de datos, incluyendo la información del host, puerto, y credenciales de acceso. Se asegura que la aplicación se conecte correctamente a la base de datos cada vez que se inicie.
  + **index.js:** Es el punto de entrada de la aplicación backend. Aquí se configuran y arrancan los servidores, se importan y aplican los middlewares necesarios, y se inician las rutas.

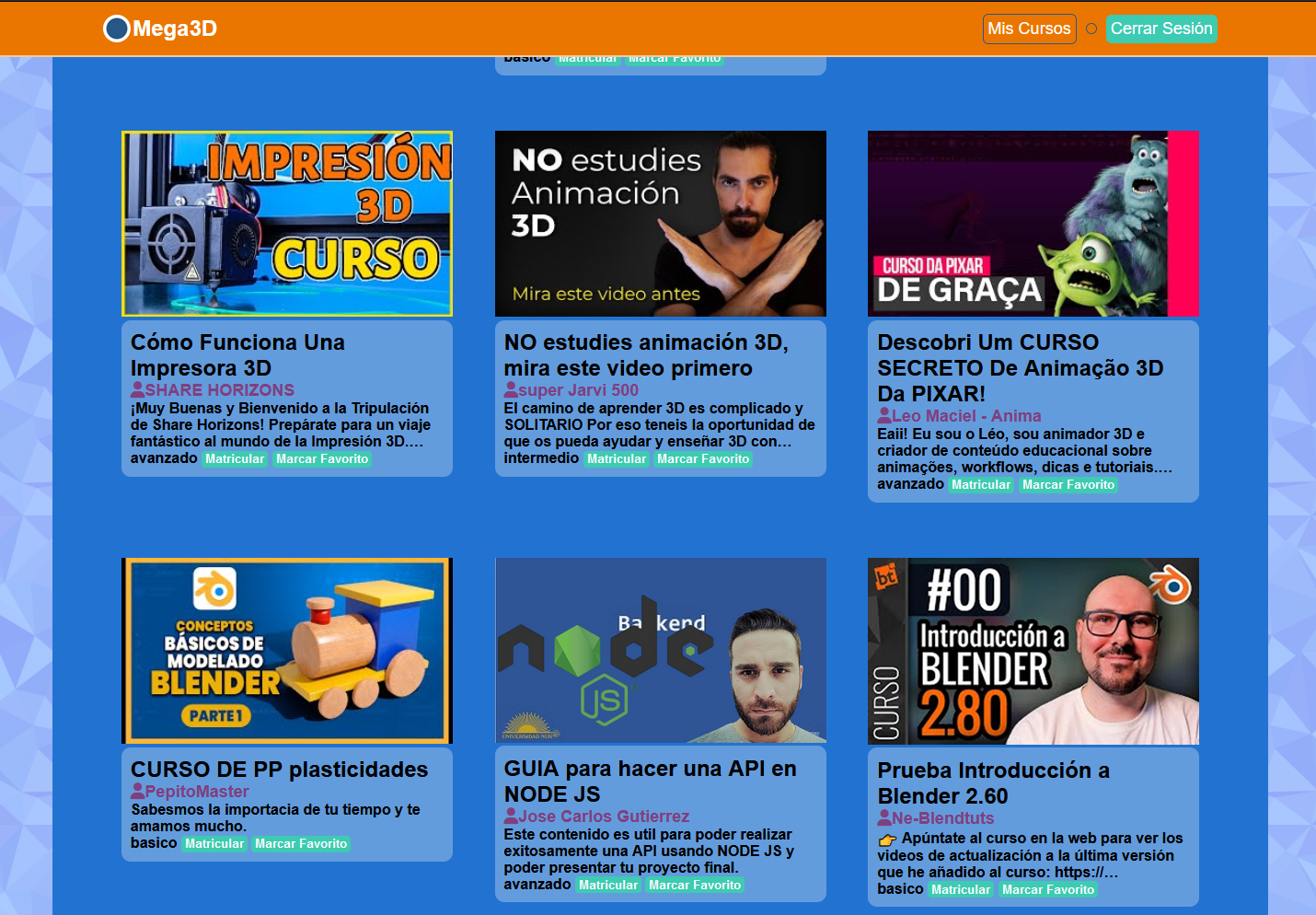
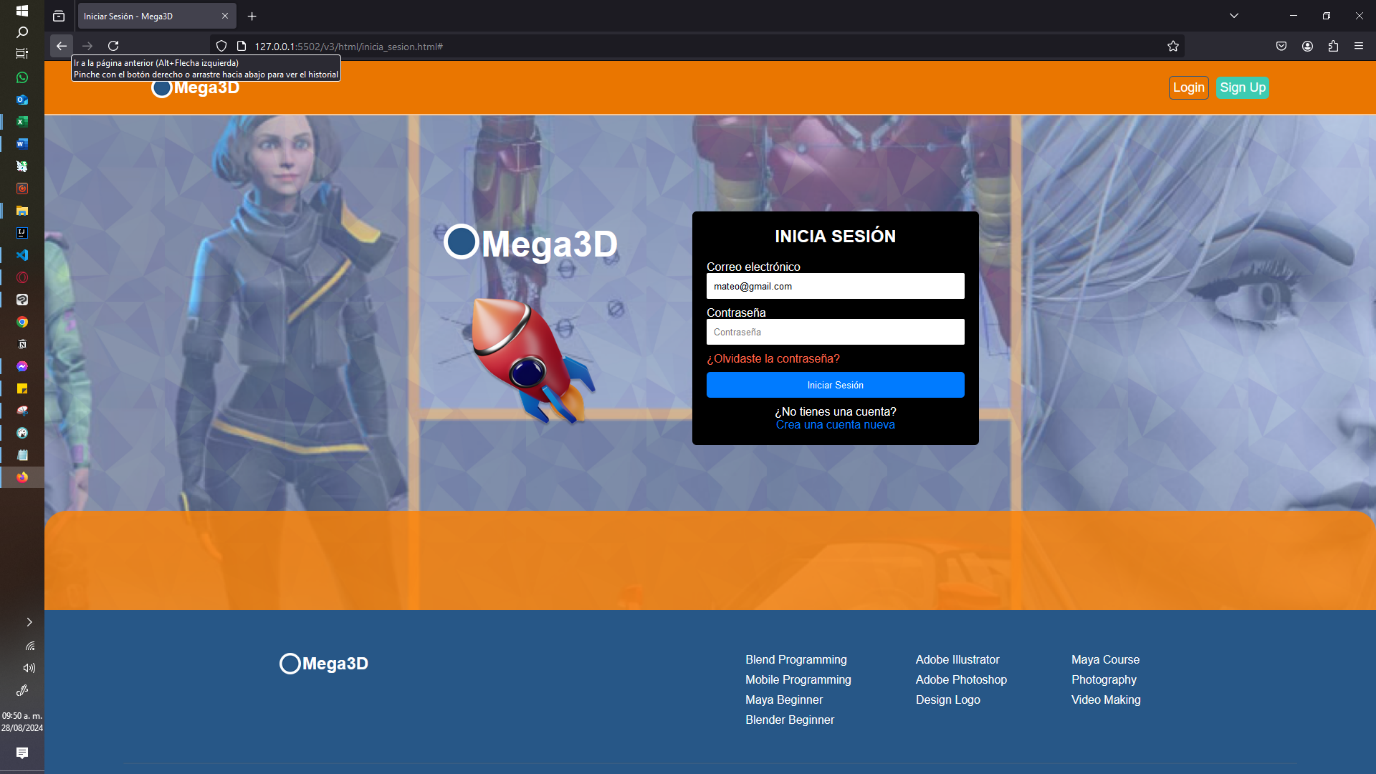
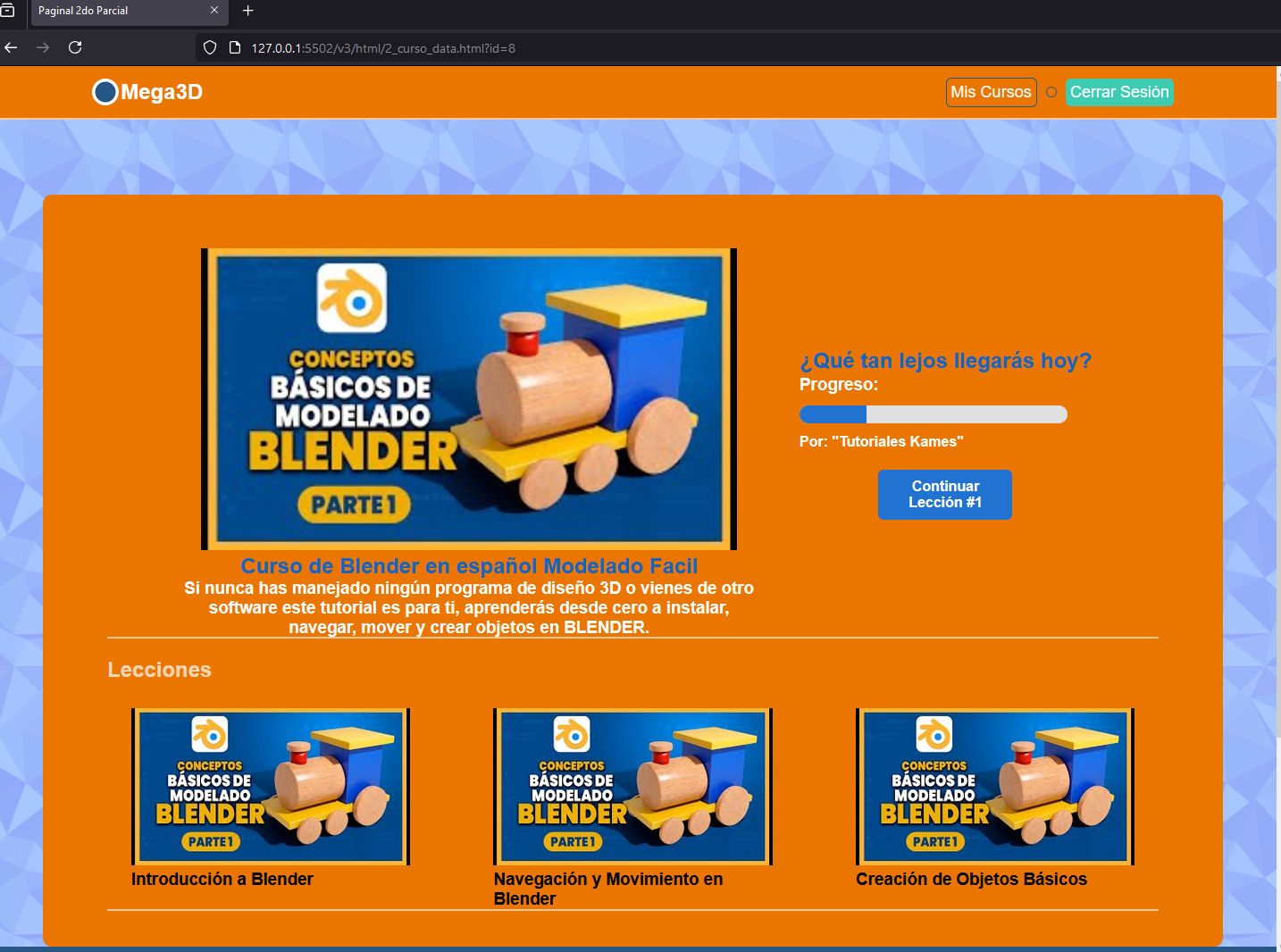
**Frontend**

El frontend se ocupa de la presentación y la interacción con el usuario. Está organizado para soportar una experiencia de usuario fluida y eficiente en diferentes dispositivos.

* **v3/**: Carpeta principal del frontend que alberga todo lo relacionado con la interfaz de usuario.
  + **html/**: Contiene los archivos HTML que representan las diferentes páginas del sitio.
    - **2\_curso\_data.html**: Página que muestra detalles específicos de un curso, incluyendo las lecciones disponibles en listadas.
    - **3\_login.html**: Pagina que muestra la lista de cursos a los cuales te has suscrito, asi como información de cada uno de ellos y tu progreso.
    - **4\_login\_ver\_curso.html**: pagina de pruebas, no usada en el proyecto final.
    - **5.html**: Pagina de administrador, donde se enlistan los cursos, sus lecciones y categorías. Esta pagina te permite hacer crud sobre todo el contenido de la página. Esta está reservada solo para usuarios administradores.
    - **inicia\_sesion.html**: Página formulario para registrarse o iniciar sesión, esta valida las credenciales antes de poder pasar a siguientes validaciones.
    - **leccion.html**: Página específica que muestra el contenido de una lección seleccionada dentro de un curso, es decir, muestra un blog de texto o un video de la lección, junto con más información y una portada.
  + **js/**: Carpeta que contiene los archivos JavaScript responsables de la interacción dinámica del sitio.
    - **detalleCurso.js**: Maneja la lógica y funcionalidad para la página de detalle de un curso, incluyendo la carga de datos del curso y la navegación entre lecciones.
    - **detalleLeccion.js**: Se encarga de la lógica para mostrar el contenido de una lección, gestionando la navegación y el seguimiento del progreso.
    - **detalleMisCursos.js**: Administra la página donde el usuario puede ver el progreso en sus cursos inscritos y acceder a las lecciones correspondientes.
    - **funcionesAdmin.js**: Contiene las funciones necesarias para las operaciones administrativas, como la creación y modificación de cursos y lecciones.
    - **iniciaSesion.js**: Gestiona el proceso de autenticación del usuario, validando credenciales y redirigiendo al usuario al área correspondiente.
    - **paginaPrincipal.js**: Lógica asociada con la página de inicio, incluyendo la carga de cursos destacados y manejo de eventos de navegación.
  + **img/**: Directorio donde se almacenan las imágenes utilizadas en el sitio, como logos, banners, y miniaturas de cursos.
  + **css/**: Carpeta destinada a los estilos CSS que definen la apariencia visual del sitio.
    - **styles.css**: Archivo principal que contiene los estilos generales del sitio.
    - **cursos.css**: Estilos específicos para la página que lista los cursos disponibles.
    - **admin/curso-lecciones.css**: Estilos para las interfaces de gestión de cursos y lecciones.

**ANEXOS**

****

****  **** **** 