SENATI

PLAN DE TRABAJO DEL ESTUDIANTE

RDA
RECURSO DIDÁCTICO PARA EL APRENDIZAJE

DATOS DEL ESTUDIANTE

Apellidos y Nombres:	Alcala Meneses Luis Orlando	ID:	001463684
Dirección Zonal/CFP:	Zonal Ica-Ayacucho-Filial Chincha		
Carrera:	Ingenieria de Software con IA	Semestre	5
Curso/ Mód. Formativo:	FULLSTACK DEVELOPER SOFTWARE		
Tema de Trabajo Final:	EDUPLUS		

1. INFORMACIÓN

- Identifica la problemática del caso práctico propuesto.

La botica "nova salud", ubicada en una zona con alta demanda, ha experimentado un crecimiento en su clientela que ha generado dificultades en la gestión del inventario y en la atención al cliente. actualmente, todos los procesos se realizan de forma manual, lo que provoca errores frecuentes,desabastecimientos y tiempos de espera prolongados. esta situación afecta la eficiencia del negocio y la satisfacción de los clientes. por ello, se propone implementar un software web que permita automatizar y centralizar el control de stock, ventas y atención, con el fin de optimizar la gestión interna y mejorar la competitividad de la botica.

Propuesta de solución

Implementar un software web centralizado que automatice la gestión del inventario, las ventas y la atención al cliente. Este sistema incluirá funcionalidades
como actualización en tiempo real del stock, alertas automáticas para reposición de productos, registro de ventas y una interfaz amigable para mejorar el proceso de atención al cliente.
Objetivo:
Desarrollar una aplicación web que permita la gestión centralizada del inventario la automatización del registro de ventas y la optimización del proceso de atenciónal cliente.

- **Respuestas a preguntas guía**

  Durante el análisis y estudio del caso práctico, debes obtener las respuestas a las interrogantes:

Pregunta 01: *¿Cuáles son los principales desafíos operativos que enfrenta la botica "FarmaPlus" debido a la falta de un sistema de gestión de inventariosautomatizado?*

Errores en conteo de stock, ventas no registradas correctamente, desabastecimientos frecuentes, demoras en atención al cliente y ausencia de reportes confiables.

Pregunta 02: *¿Qué características específicas debe incluir el software web para atender las necesidades de control de stock y atención al cliente en "FarmaPlus"?*

Control automatizado del stock, registro y seguimiento de ventas, interfaz fácil de usar,alertas de stock bajo y reportes exportables.

Pregunta 03: ¿De qué manera la implementación de un software web para la gestión de inventarios puede impactar en la reducción del desabastecimiento y la mejora del servicio al cliente?

Aumenta la eficiencia, reduce errores humanos, mejora el control del inventario y permite una atención más rápida y ordenada al cliente.

Pregunta 04:	*¿Qué beneficios adicionales, aparte del control de inventarios, podría ofrecer el software para incrementar la competitividad de "FarmaPlus" en su mercado?*

Estadísticas para toma de decisiones, seguimiento del rendimiento de empleados,historial de ventas, escalabilidad para futuras funciones como pagos electrónicos.

Pregunta 05:	*¿Qué métricas se pueden utilizar para evaluar la efectividad del software en mejorar los tiempos de respuesta al cliente y reducir los errores en la gestión de inventario?*

Tiempo promedio de atención, porcentaje de productos sin stock, reducción de errores en inventario, número de ventas sin fallas y satisfacción del cliente.

2. PLANIFICACIÓN DEL TRABAJO

- **Cronograma de actividades:**

N°	ACTIVIDADES	CRONOGRAMA					
		Lun	Mar	Mie	Ju	Vi	Sab
1	Conoce y aplica prácticas de desarrollo de software colaborativo	x					
2	Usa entorno de ejecución backend con JavaScript		x				
3	Usa tecnología frontend con JavaScript			x			
4	Diseña y crea servicios API RESTful				x		

- **Lista de recursos necesarios:**

1. MÁQUINAS Y EQUIPOS	
Descripción	**Cantidad**
Laptop	1

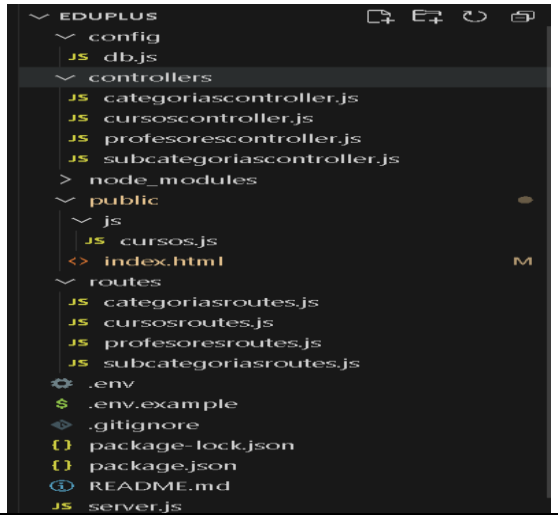
2. HERRAMIENTAS E INSTRUMENTOS	
Descripción	**Cantidad**
❖ Visual Studio code	1
❖ Node.Js	
❖ GitHub	

3. MATERIALES E INSUMOS	
Descripción	**Cantidad**
💧 Energía	1
💧 Documentación de SENATI	1
💧 Internet	

3. DECIDIR PROPUESTA

- **Describe la propuesta determinada para la solución del caso práctico**

PROPUESTA DE SOLUCIÓN

La propuesta consiste en implementar Eduplus, una plataforma web (Node.js + MySQL) para gestionar categorías, subcategorías, profesores y cursos.
1.Objetivo
Implementar una aplicación web para gestionar categorías, subcategorías, profesores y cursos con un backend REST (Node.js + Express + MySQL) y un frontend ligero con Bootstrap y Font Awesome, mejorando la presentación y usabilidad del panel administrativo.
Alcance
• **Conexión y consultas a MySQL correctas.**
• **Endpoints REST para categorías, subcategorías, profesores y cursos.**
• **Panel en public/index.html con:**
  ○ **Listado de categorías y cursos.**
  ○ **Formularios en modales para crear registros.**
  ○ **Estilos con Bootstrap y iconos con Font Awesome (agregados).**
• **Estructura del proyecto organizada (config, controllers, routes, public).**
2) Arquitectura
Patrón MVC ligero / por capas
 

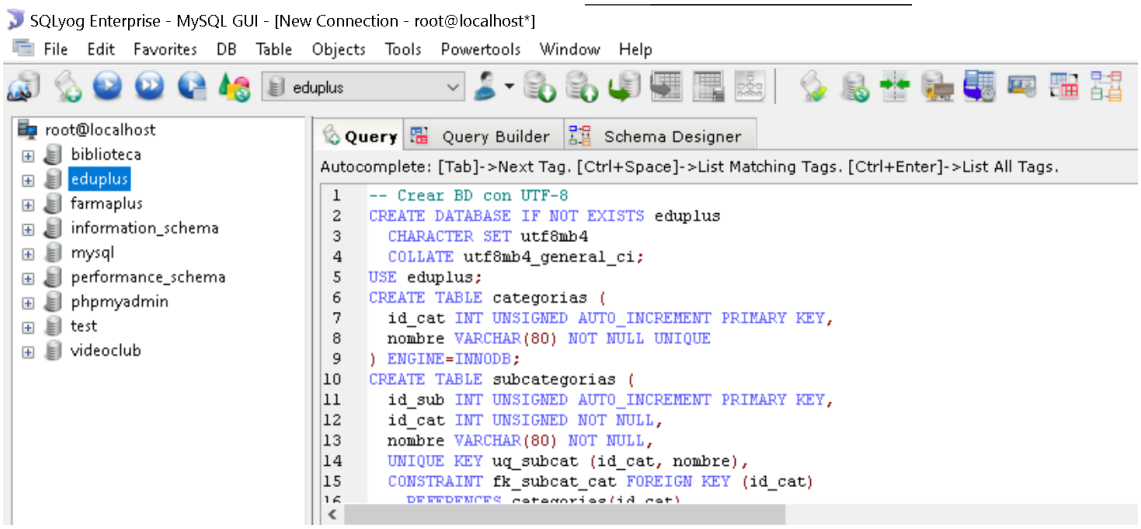
3) Configuración:

Variables de entorno (.env)

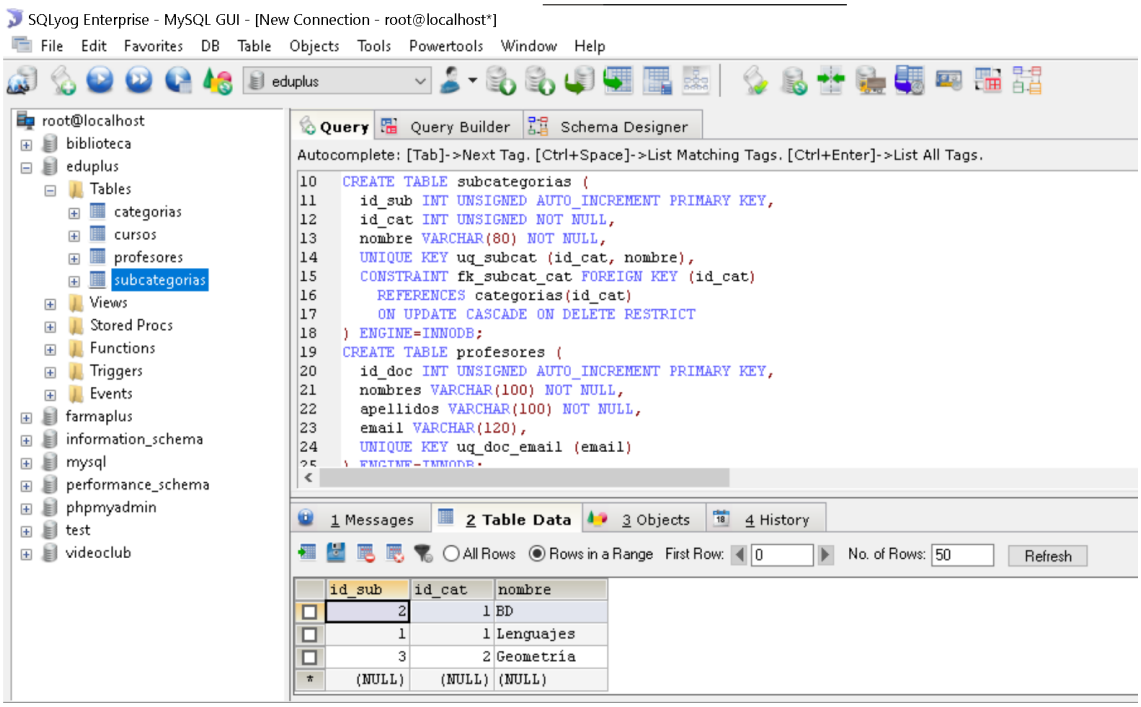
.env

```
DB_HOST=localhost
DB_USER=root
DB_PASSWORD=
DB_DATABASE=eduplus
DB_PORT=3306
DB_POOL_SIZE=10
```

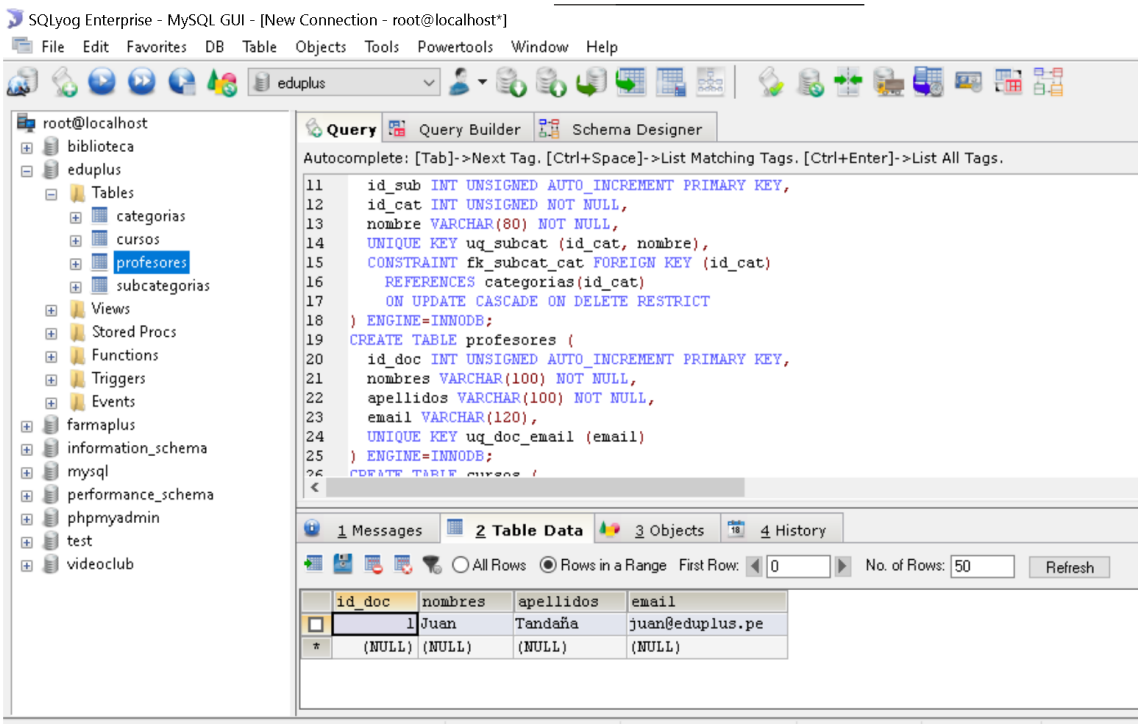
4) Diseño de la base de datos:EDUPLUS



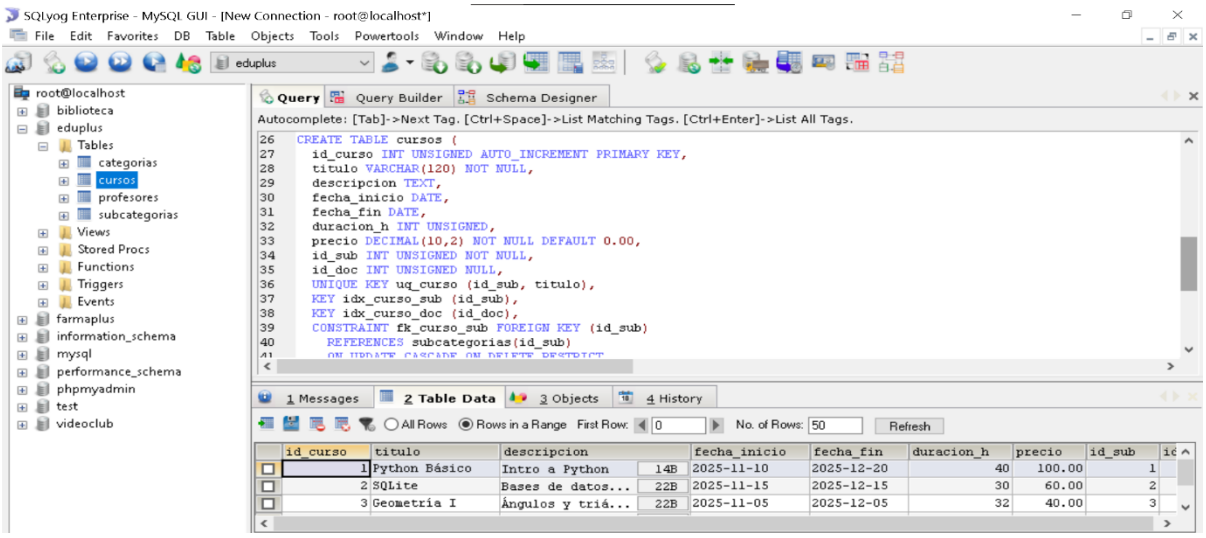
Subcategorias:



Profesores:



Cursos:



5) Ejecución:

npm init -y

npm install express mysql2 dotenv

Levantar:

nodemon server.js

6) Configuración del Servidor Backend

-Archivo db.js — Conexión con la Base de Datos

Este archivo permite establecer la conexión entre el servidor backend y la base de datos MySQL usando un pool de conexiones. Esto mejora el rendimiento porque mantiene conexiones abiertas y reutilizables en lugar de abrir una nueva cada vez que se hace una consulta.

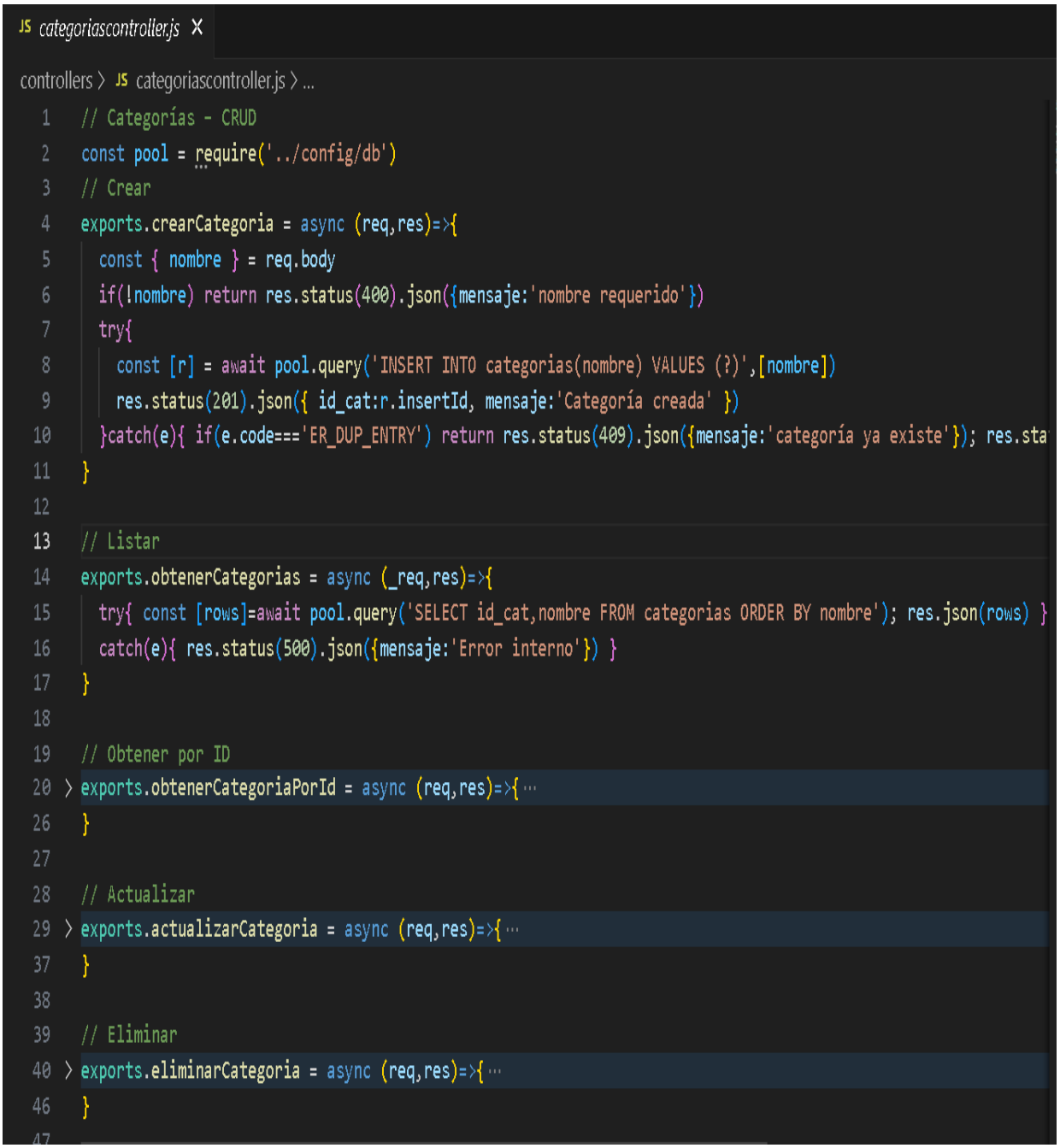
db.js

config > db.js > ...

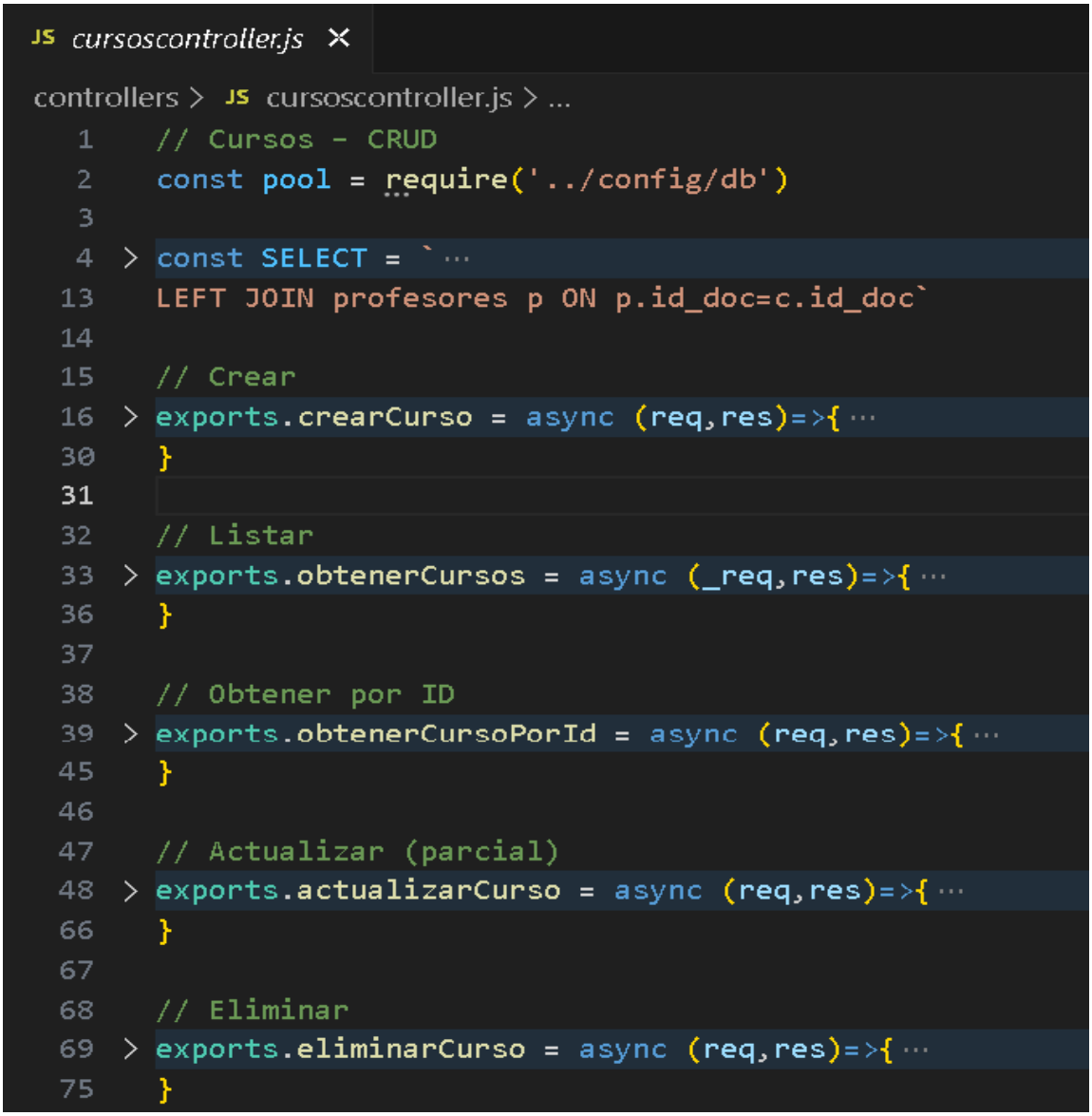
```
require('dotenv').config();

//Administrar la BD(promesesas=proceso en curso..)
const mysql = require('mysql2/promise');

//Pool de conexiones  = acceso
const pool = mysql.createPool({
  host: process.env.DB_HOST || '127.0.0.1',
  user: process.env.DB_USER || 'root',
  password: process.env.DB_PASSWORD || '',
  database: process.env.DB_DATABASE || 'eduplus',
  port: Number(process.env.DB_PORT || 3306),
  waitForConnections: true,
  connectionLimit: Number(process.env.DB_POOL_SIZE || 10),
  queueLimit: 0,
  multipleStatements: false,  // evita SQL injection por lotes
  dateStrings: true           // fechas como string (sin líos de TZ)
});

//Aprovechar el recurso en otra parte de la App
module.exports = pool;
```

-Archivo controller-categoriascontroller.js — Lógica del CRUD

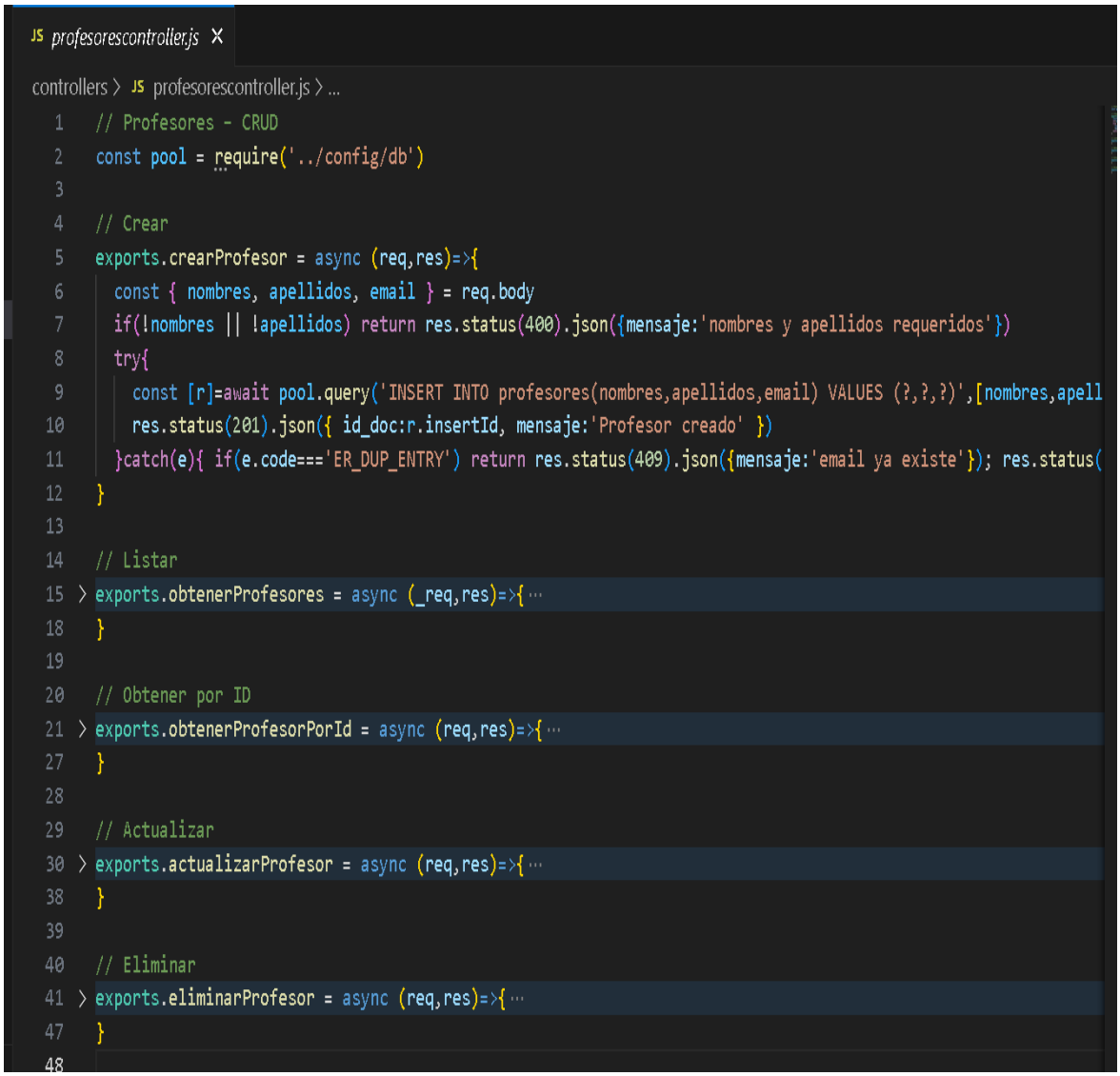
Este archivo contiene la lógica de negocio de la aplicación, es decir, las funciones que interactúan directamente con la base de datos MySQL para crear, listar, buscar, actualizar y eliminar registros de la tabla Eduplus.



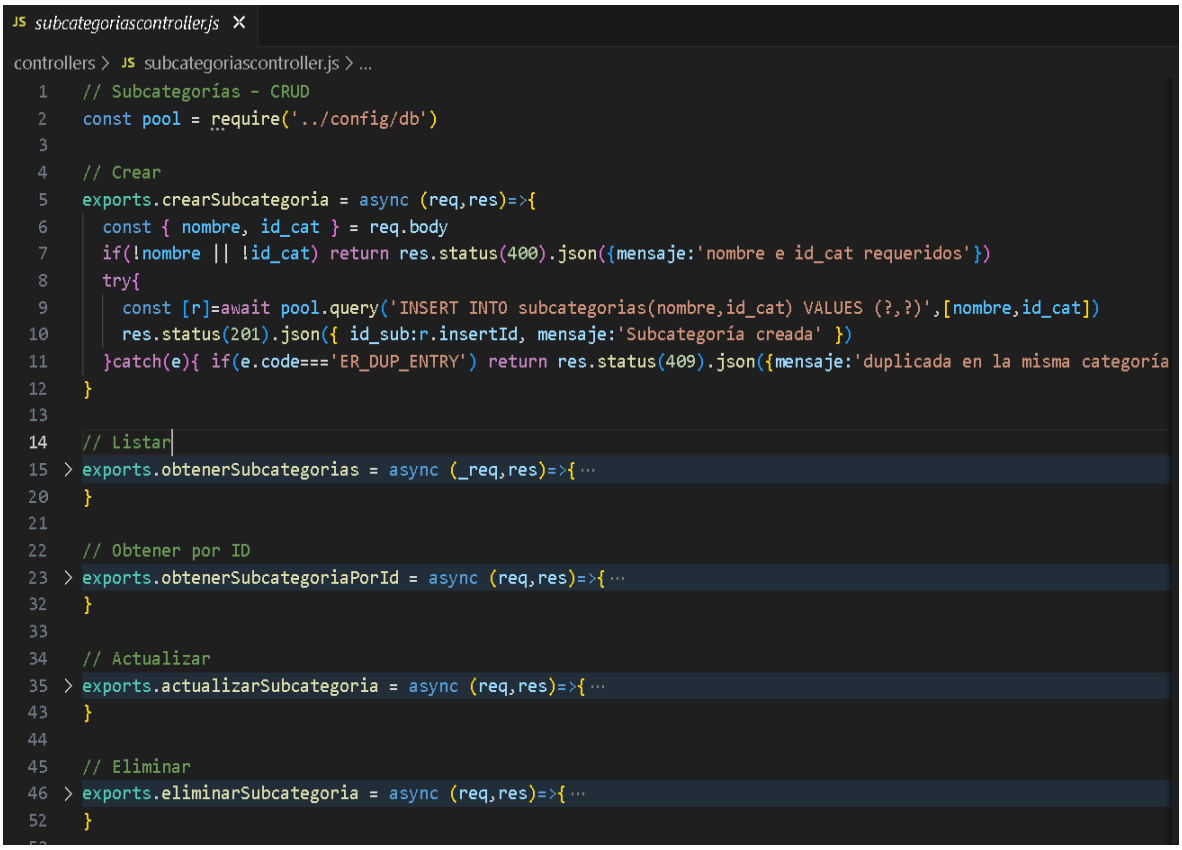
-Archivo controller-cursoscontroller.js — Lógica del CRUD



Archivo controller-profesorerescontroller.js — Lógica del CRUD



Archivo controller-subcategoriascontroller.js — Lógica del CRUD



```
subcategoriascontroller.js
controllers > subcategoriascontroller.js > ...
1   // Subcategorías - CRUD
2   const pool = require('../config/db')
3
4   // Crear
5   exports.crearSubcategoria = async (req,res)=>{
6     const { nombre, id_cat } = req.body
7     if(!nombre || !id_cat) return res.status(400).json({mensaje:'nombre e id_cat requeridos'})
8     try{
9       const [r]=await pool.query('INSERT INTO subcategorias(nombre,id_cat) VALUES (?,?)',[nombre,id_cat])
10      res.status(201).json({ id_sub:r.insertId, mensaje:'Subcategoría creada' })
11    }catch(e){ if(e.code==='ER_DUP_ENTRY') return res.status(409).json({mensaje:'duplicada en la misma categoría
12  }
13
14  // Listar
15  exports.obtenerSubcategorias = async (_req,res)=>{ ···
20  }
21
22  // Obtener por ID
23  exports.obtenerSubcategoriaPorId = async (req,res)=>{ ···
32  }
33
34  // Actualizar
35  exports.actualizarSubcategoria = async (req,res)=>{ ···
43  }
44
45  // Eliminar
46  exports.eliminarSubcategoria = async (req,res)=>{ ···
52  }
```

Routes

Archivo categoriaroutes.js — Definición de Rutas de la API

Este archivo define las rutas principales que permiten interactuar con el backend de Eduplus. Cada ruta está asociada a una función específica del controlador (categoriascontroller.js), lo que facilita la organización del código siguiendo el patrón MVC (Modelo – Vista – Controlador).

```
categoriasroutes.js
routes > categoriasroutes.js > ...
1   const express = require('express');
2   const router = express.Router();
3   const categoriaController = require('../controllers/categoriascontroller');
4
5   // Crear categoría
6   router.post('/', categoriaController.crearCategoria);
7
8   // Listar todas las categorías
9   router.get('/', categoriaController.obtenerCategorias);
10
11  // Obtener categoría por ID
12  router.get('/:id', categoriaController.obtenerCategoriaPorId);
13
14  // Actualizar categoría
15  router.put('/:id', categoriaController.actualizarCategoria);
16
17  // Eliminar categoría
18  router.delete('/:id', categoriaController.eliminarCategoria);
19
20  // Exportar el router
21  module.exports = router;
```

Archivo server.js — Configuración del Servidor Backend

Este archivo es el corazón del proyecto. Se encarga de levantar el servidor con Express, conectarlo a la base de datos y registrar las rutas de la API.

```
JS server.js
JS server.js > ...
1   // Cargar variables de entorno
2   require('dotenv').config()
3
4   // Mensaje de control al iniciar
5   console.log('Archivo server.js ejecutándose...')
6
7   // Importar dependencias principales
8   const express = require('express')
9
10  // Importar rutas de Eduplus
11  const categoriasRoutes     = require('./routes/categoriasroutes')
12  const subcategoriasRoutes  = require('./routes/subcategoriasroutes')
13  const profesoresRoutes     = require('./routes/profesoresroutes')
14  const cursosRoutes         = require('./routes/cursosroutes')
15
16  // Importar pool de conexión a la BD
17  const pool = require('./config/db')
18
19  // Inicializar la aplicación Express
20  const app = express()
21
22  // Definir el puerto del servidor (por .env o por defecto 3000)
23  const PORT = process.env.PORT || 3000
24
```

Consola: Evidencia en consola de la correcta conexión del servidor backend con la base de datos MySQL. El mensaje “ Conexión a la base de datos correcta” confirma que el pool establecido en db.js funciona correctamente.

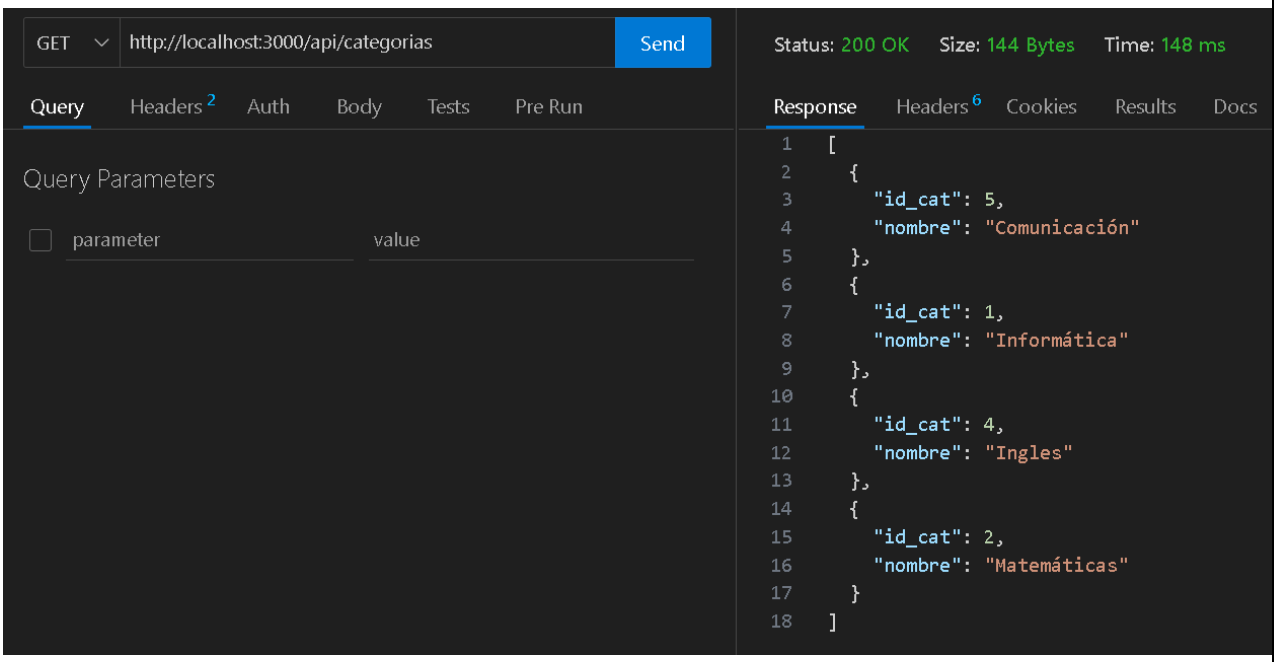
El mensaje “ Servidor iniciado en http://localhost:3000” indica que el servidor Express se está ejecutando sin errores y está listo para recibir peticiones en el puerto configurado.

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS        node

PS C:\Users\Luis\Downloads\Eduplus> nodemon server.js
[nodemon] starting `node server.js`
[dotenv@17.2.3] injecting env (6) from .env -- tip: sync secrets across teammates & machines: https://dotenvx.com/ops
Archivo server.js ejecutándose...
[dotenv@17.2.3] injecting env (0) from .env -- tip: sync secrets across teammates & machines: https://dotenvx.com/ops
✓ Server cargando...
Servidor iniciado en http://localhost:3000
✓ Conexión a la base de datos correcta
```

Endpoints REST Implementado

GET /api/categorias→ listar categorias.

Este endpoint permite obtener un listado completo de todas los Eduplus almacenadas en la base de datos. Cuando se realiza una solicitud GET a esta ruta, el servidor responde con un arreglo en formato JSON que contiene cadacategoriasregistrada, incluyendo sus atributos: id, nombre.



6) Resultados

- **CRUD funcional para las entidades definidas.**
- **Interfaz con Bootstrap y Font Awesome mejorada visualmente.**
- **Conexión a BD estable (pool MySQL).**

El proyecto Eduplus logró un backend REST confiable (Node.js + Express + MySQL) y un panel web usable con Bootstrap y Font Awesome, cumpliendo el CRUD de categorías, subcategorías, profesores y cursos. La arquitectura modular (config–controllers–routes–public) facilita el mantenimiento y la escalabilidad, y deja preparado el sistema para incorporar mejoras sin romper lo construido. La base de datos está correctamente normalizada y la conexión estable asegura integridad en las operaciones

4. EJECUTAR

- **Resolver el caso práctico, utilizando como referencia el problema propuesto y las preguntas guía proporcionadas para orientar el desarrollo.**
- **Fundamentar sus propuestas en los conocimientos adquiridos a lo largo del curso, aplicando lo aprendido en las tareas y operaciones descritas en los contenidos curriculares.**

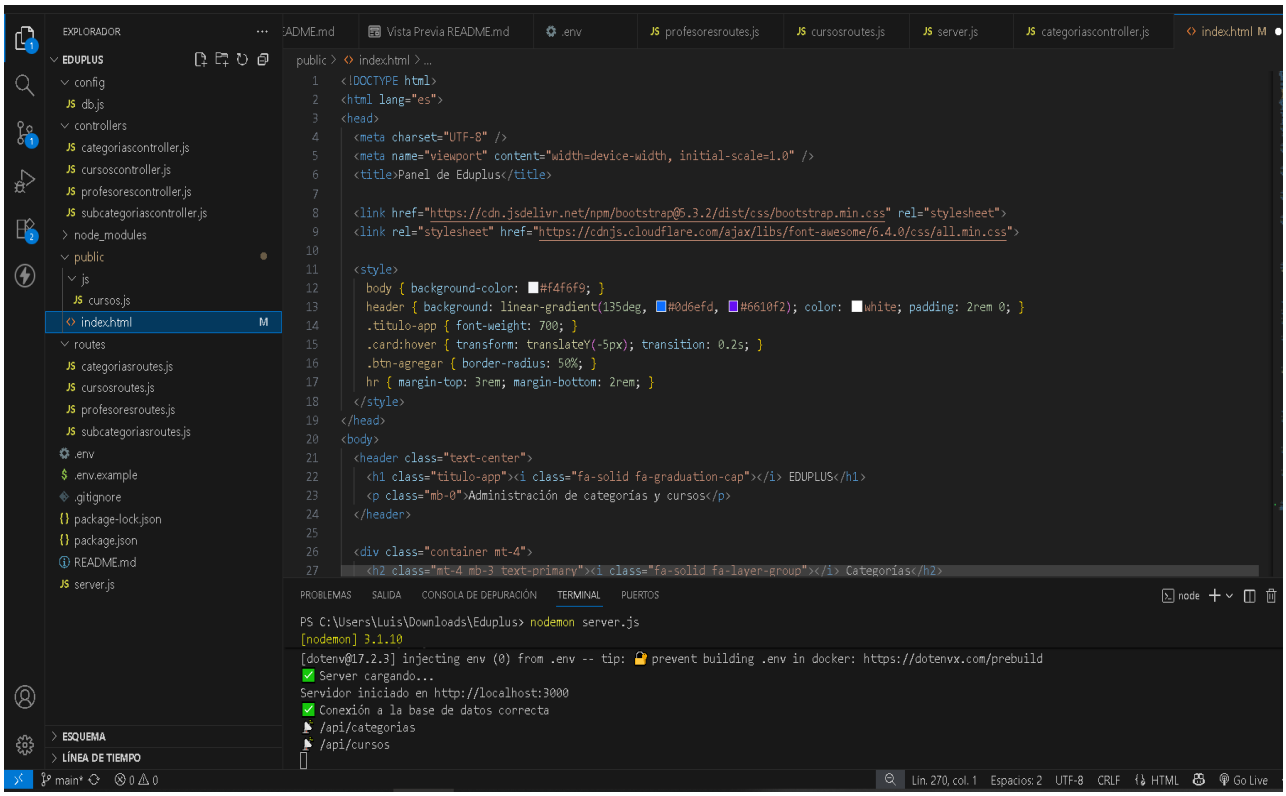
INSTRUCCIONES: Ser lo más explícito posible. Los gráficos ayudan a transmitir mejor las ideas. Tomar en cuenta los aspectos de calidad, medio ambiente y SHI.

OPERACIONES / PASOS / SUBPASOS	NORMAS TÉCNICAS - ESTANDARES / SEGURIDAD / MEDIO AMBIENTE
Paso 1: Se definió la BD eduplus en MySQL con charset utf8mb4.	**ISO/IEC 23894: Gestión de riesgos en tecnologías de la información**
Paso 2: Se crearon tablas: categorías, subcategorías, profesores y cursos con claves foráneas.	**ISO/IEC 23053: Marco para el uso de aprendizaje automático (machine learning)**
Paso 3: Se cargó semilla inicial (categorías, subcategorías, profesor y cursos).	**ISO/IEC 24029: Evaluación de la robustez de sistemas de IA.**
Paso 4Se configuró .env con variables de conexión a MySQL y puerto.	**ISO/IEC 27001: Gestión de la seguridad de la información.**
Paso 5Se instaló y configuró Express y mysql2 en el proyecto.	**ISO/IEC 20000: Gestión de servicios de TI.**
Paso 6. Se creó config/db.js con pool de conexiones y lectura de .env.	**ISO/IEC 12207: Procesos del ciclo de vida del software.**
Paso 7: Se implementaron rutas REST para categorías (GET/POST/DELETE).	**ISO/IEC 25010: Calidad del producto software.**
Paso 8:. Se implementaron rutas REST para subcategorías (GET/POST).	**SO/IEC 9126: Calidad del software.**
Paso 9: Se implementaron rutas REST para profesores (GET/POST).	**ISO/IEC 27032: Seguridad en el ciberespacio.**
Paso 10: Se implementaron rutas REST para cursos (GET/POST/DELETE).	**ISO/IEC 27005: Gestión de riesgos de seguridad de la información.**
Paso 11: Se configuró server.js: middlewares, registro de rutas y estáticos desde /public.	**ISO/IEC 15408 : Criterios comunes para la evaluación de seguridad de productos y sistemas de TI.**
Paso 12: Se construyó public/index.html como panel de administración.	**ISO 9001: Sistemas de gestión de calidad.**
Paso 13: Se integró Bootstrap y Font Awesome para estilos e íconos en el front.	**ISO 21500: Gestión de proyectos.**
Paso 14 Se programaron fetch en el front para listar/crear/eliminar (modales y botones).	**ISO 14001: Gestión ambiental.**
Paso 15: Se probó la API con Thunder Client y se verificó el CRUD funcionando. **.**	**ISO 50001: Gestión de la energía.**

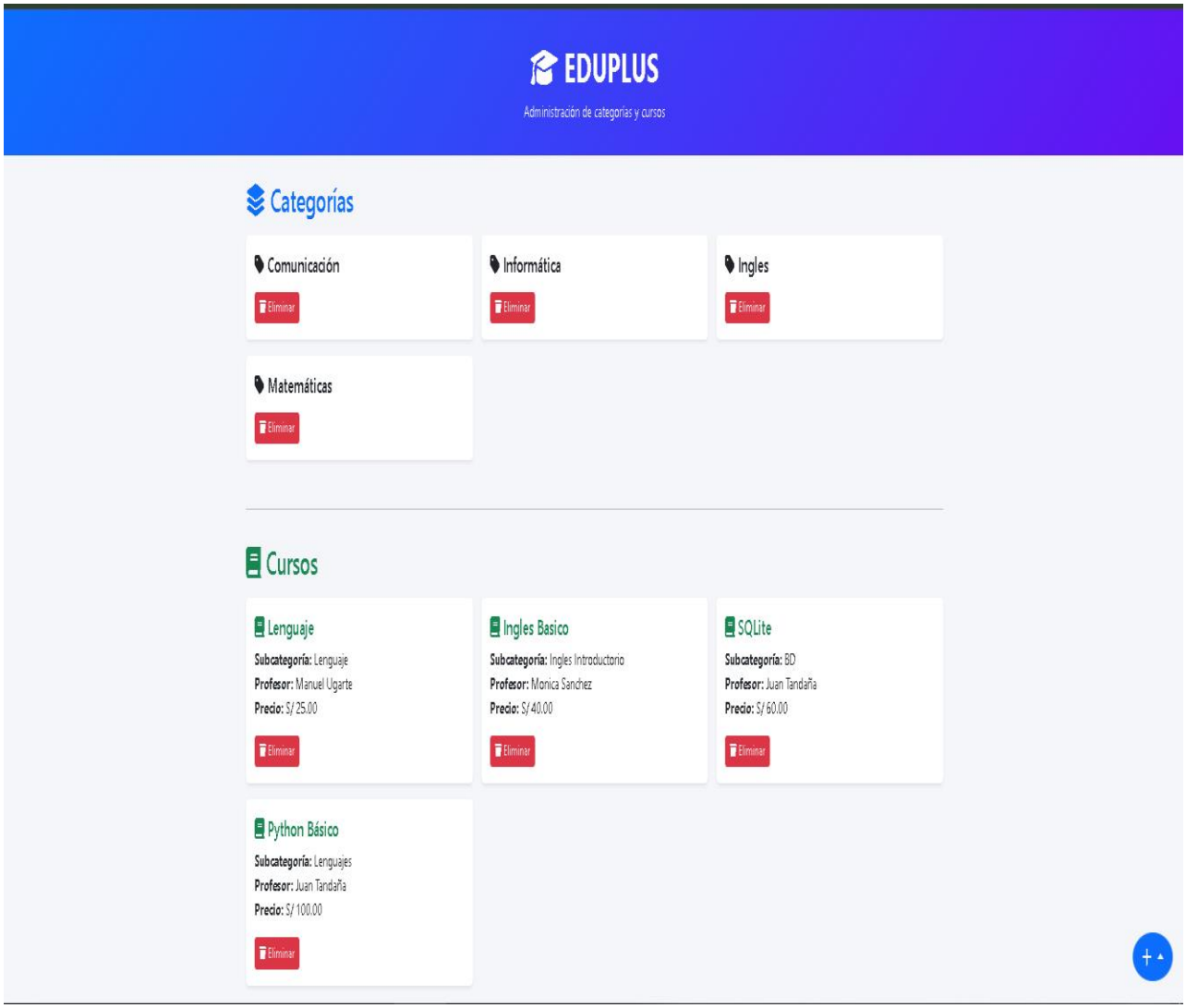
DIBUJO / ESQUEMA / DIAGRAMA DE PROPUESTA

(Adicionar las páginas que sean necesarias)

DIBUJO 1:



DIBUJO 2:


EDUPLUS
Administración de categorías y cursos
Categorías
Comunicación
Eliminar
Informática
Eliminar
Ingles
Eliminar
Matemáticas
Eliminar
Cursos
Lenguaje
Subcategoría: Lenguaje
Profesor: Manuel Ugarte
Precio: S/ 25.00
Eliminar
Ingles Basico
Subcategoría: Ingles Introductorio
Profesor: Monica Sanchez
Precio: S/ 40.00
Eliminar
SQLite
Subcategoría: BD
Profesor: Juan Tandaña
Precio: S/ 60.00
Eliminar
Python Básico
Subcategoría: Lenguajes
Profesor: Juan Tandaña
Precio: S/ 100.00
Eliminar

5. CONTROLAR

- **Verificar el cumplimiento de los procesos desarrollados en la propuesta de solución del caso práctico.**

EVIDENCIAS	CUMPLE	NO CUMPLE
• ¿Se identificó claramente la problemática del caso práctico?	☒	☐
• ¿Se desarrolló las condiciones de los requerimientos solicitados?	☒	☐
• ¿Se formularon respuestas claras y fundamentadas a todas las preguntas guía?	☒	☐
• ¿Se elaboró un cronograma claro de actividades a ejecutar?	☒	☐
• ¿Se identificaron y listaron los recursos (máquinas, equipos, herramientas, materiales) necesarios para ejecutar la propuesta?	☒	☐
• ¿Se ejecutó la propuesta de acuerdo con la planificación y cronograma establecidos?	☒	☐
• ¿Se describieron todas las operaciones y pasos seguidos para garantizar la correcta ejecución?	☒	☐
• ¿Se consideran las normativas técnicas, de seguridad y medio ambiente en la propuesta de solución?	☒	☐
• ¿La propuesta es pertinente con los requerimientos solicitados?	☒	☐
• ¿Se evaluó la viabilidad de la propuesta para un contexto real?	☒	☐

6. VALORAR

- **Califica el impacto que representa la propuesta de solución ante la situación planteada en el caso práctico.**

CRITERIO DE EVALUACIÓN	DESCRIPCIÓN DEL CRITERIO	PUNTUACIÓN MÁXIMA	PUNTAJE CALIFICADO POR EL ESTUDIANTE
Identificación del problema	Claridad en la identificación del problema planteado.	3	
Relevancia de la propuesta de solución	La propuesta responde adecuadamente al problema planteado y es relevante para el contexto del caso práctico.	8	
Viabilidad técnica	La solución es técnicamente factible, tomando en cuenta los recursos y conocimientos disponibles.	6	
Cumplimiento de Normas	La solución cumple con todas las normas técnicas de seguridad, higiene y medio ambiente.	3	
PUNTAJE TOTAL			


SENATI