**DATOS DEL ESTUDIANTE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Apellidos y Nombres: | Rodríguez Anchante Sandro Steven | ID: | 1555585 | |
| Dirección Zonal/CFP: | Ica – Ayacucho | | | |
| Carrera: | Ingeniería de Software con Inteligencia Artificial | Semestre: | | V |
| Curso/ Mód. Formativo: | FullStack Developer Software | | | |
| Tema de Trabajo Final: | Entregable 01 – Implementación de CRUD, (BACKEND / FRONTED) sobre Cursos | | | |

1. **INFORMACIÓN**

* **Identifica la problemática del caso práctico propuesto.**

**El objetivo principal es implementar un Sistema de Gestión de Oferta Educativa que permita la organización estructurada y la administración digital de los cursos. Esto implica:**

1. **Clasificar el catálogo de cursos mediante una jerarquía clara de Categoría > Subcategoría > Curso.**
2. **Digitalizar el proceso de Registro, Actualización y Consulta (CRUD) de la información clave de cada curso (fechas, docente, precio).**
3. **Garantizar la integridad referencial de los datos, asegurando que cada curso esté correctamente asociado a una subcategoría y que las inscripciones se vinculen de forma válida a un curso existente.**

* **Identifica propuesta de solución y evidencias.**

**Desarrollo de una Aplicación Web Full-Stack con una API RESTful que gestionará la información mediante un modelo de Base de Datos Relacional de cuatro tablas interconectadas para soportar la gestión completa de los Cursos y sus Inscripciones.**

**Evidencias de la Solución (Modelo de Datos Relacional): Se utilizarán cuatro tablas normalizadas (Categoría, Subcategoría, Curso, Inscripción) con conexiones de Llaves Foráneas (FK) para modelar la relación jerárquica y transaccional del negocio.**

* **Respuestas a preguntas guía (ESTO ES DEL ENTREGABLE PASADO)**

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

|  |  |
| --- | --- |
| Pregunta 01: |  |
|  | |
| Pregunta 02: |  |
|  | |
| Pregunta 03: |  |
|  | |
| Pregunta 04: |  |
|  | |
| Pregunta 05: |  |
|  | |

**2. PLANIFICACIÓN DEL TRABAJO**

* **Cronograma de actividades:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **N°** | **ACTIVIDADES** | **CRONOGRAMA** | | | | | |
|  |  |  |  |  |  |
|  | **Configuración Inicial** | 15/10 |  |  |  |  |  |
|  | **Base de Datos y Modelado** | 15/10 | 16/10 |  |  |  |  |
|  | **Desarrollo del Backend (API CRUD)** |  | 16/10 | 17/10 | 18/10 |  |  |
|  | **Desarrollo del Frontend (Vistas y Pruebas)** |  |  | 17/10 | 18/10 | 19/10 |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

* **Lista de recursos necesarios:**

|  |  |
| --- | --- |
| 1. **MÁQUINAS Y EQUIPOS** | |
| **Descripción** | **Cantidad** |
| Laptop Personal | 1 |
| Conexión a Internet / Wi-fi | 1 |
|  |  |
|  |  |

|  |  |
| --- | --- |
| 1. **HERRAMIENTAS E INSTRUMENTOS** | |
| **Descripción** | **Cantidad** |
| Visual Studio Code (VS Code) + ThunderClient | 1 |
| SQL Developer | 1 |
| Navegador Web | 1 |
|  |  |

|  |  |
| --- | --- |
| 1. **MATERIALES E INSUMOS** | |
| **Descripción** | **Cantidad** |
| Node.js | 1 |
| Express.js | 1 |
| Archivos Staticos(JS, HTML y CSS) | 1 |
|  |  |

**3. DECIDIR PROPUESTA**

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

|  |
| --- |
| **PROPUESTA DE SOLUCIÓN** |
| **Link del Proyecto en Respositorio GITHUB (Mas Cambios Verificar Aqui):**  [**https://github.com/sandro060606/Entregable\_FullStack\_Gestion\_Cursos-Public.git**](https://github.com/sandro060606/Entregable_FullStack_Gestion_Cursos-Public.git)  **La propuesta consiste en construir un Sistema Web Full-Stack usando Node.js y Express.js, estructurado para gestionar la oferta educativa de manera relacional.**   1. **Modelo de Datos Relacional: Se utilizará un motor de bases de datos relacional (como SQLite) con cuatro tablas para garantizar la integridad referencial y la correcta clasificación:**    * **Jerarquía de Clasificación: Categoría se relaciona con Subcategoría (1:N), y Subcategoría se relaciona con Curso (1:N).**    * **Registro Transaccional: Curso se relaciona con Inscripción (1:N).** 2. **Backend (API CRUD): Se desarrollará una API RESTful con Express.js que brindará *endpoints* para realizar las operaciones CRUD completas sobre la entidad principal, Curso, y las entidades de soporte (Categoría, Subcategoría, Inscripción).** 3. **Frontend: Una interfaz desarrollada con tecnologías web estándar (HTML/CSS/JS) consumirá la API para permitir a los administradores crear, ver, editar y eliminar cursos de forma ágil y con validación de datos (ej. selección de Subcategoría y Docente usando FKs).** |

**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.**

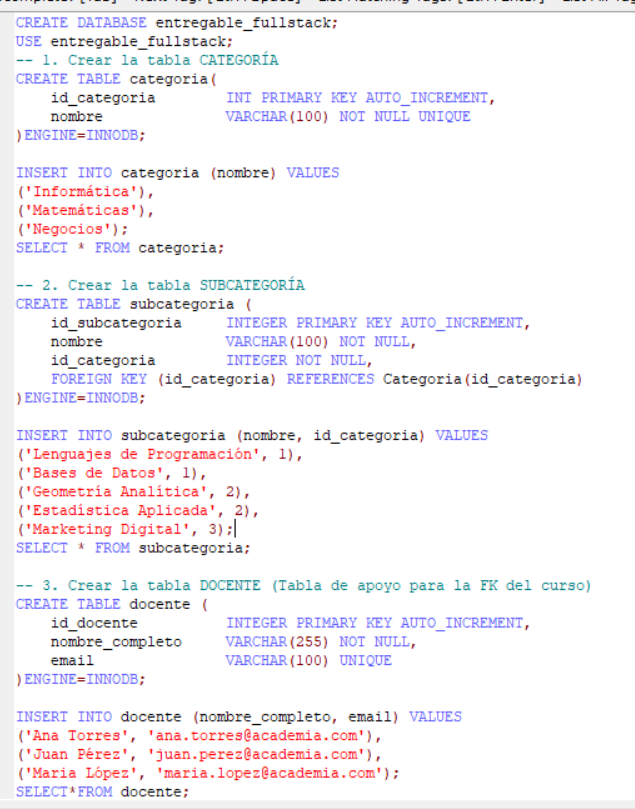
**INSTRUCCIONES:** Ser lo más explícito posible. Los gráficos ayudan a transmitir mejor las ideas. Tomar en cuenta los aspectos de calidad, medio ambiente y SHI.

|  |  |
| --- | --- |
| **OPERACIONES / PASOS / SUBPASOS** | **NORMAS TÉCNICAS - ESTANDARES / SEGURIDAD / MEDIO AMBIENTE** |
| **Crear la BD y sus respectivas tablas (Categoría, subcategoría, cursos y docente)** |  |
| **Inicializar Proyecto con Node.js e instalaciones** |  |
| **Implementar “*endpoints”* para el CRUD de Cursos (/api/cursos). También se necesitarán “*endpoints”* READ para Categoría, Subcategoría y Docente para poblar los formularios del Frontend.** |  |
| **GET - Usar JOIN para fusionar Curso con los nombres de la Subcategoría, Categoría y el Docente asociado.** |  |
| **POST/PUT / - La API debe validar que tanto el id\_subcategoria como el id\_docente existen antes de registrar el Curso (Integridad de datos).** |  |
| **DELETE - Permitir la eliminación ya que la tabla Inscripción fue excluida, evitando problemas de integridad referencial.** |  |
| **HTML/CSS/JS. Usar fetch para comunicar con la API.** |  |
| **Usar los *endpoints* READ de Subcategoría y Docente para rellenar los selectores (dropdowns) en los formularios de creación/edición de Curso, asegurando el ingreso de FKs válidas.** |  |
|  |  |
|  |  |
|  |  |
|  |  |

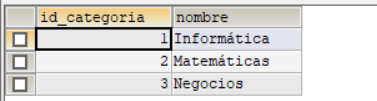
**DIBUJO / ESQUEMA / DIAGRAMA DE PROPUESTA**

**(Adicionar las páginas que sean necesarias)**

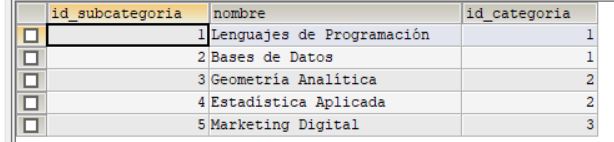
Creación de la BD, sus tablas para almacenar la Información:



Registros de Categoría:

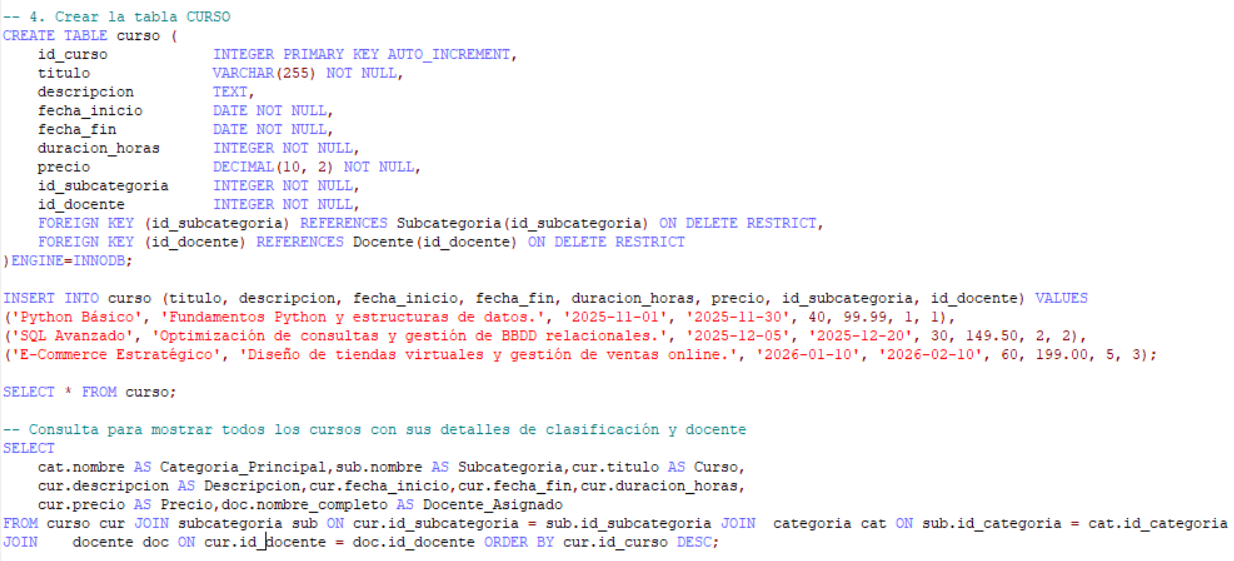


De SubCategoria:

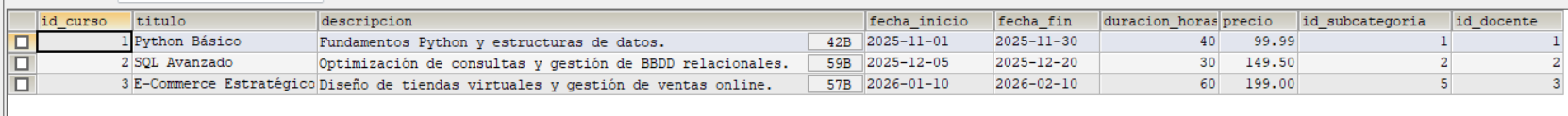


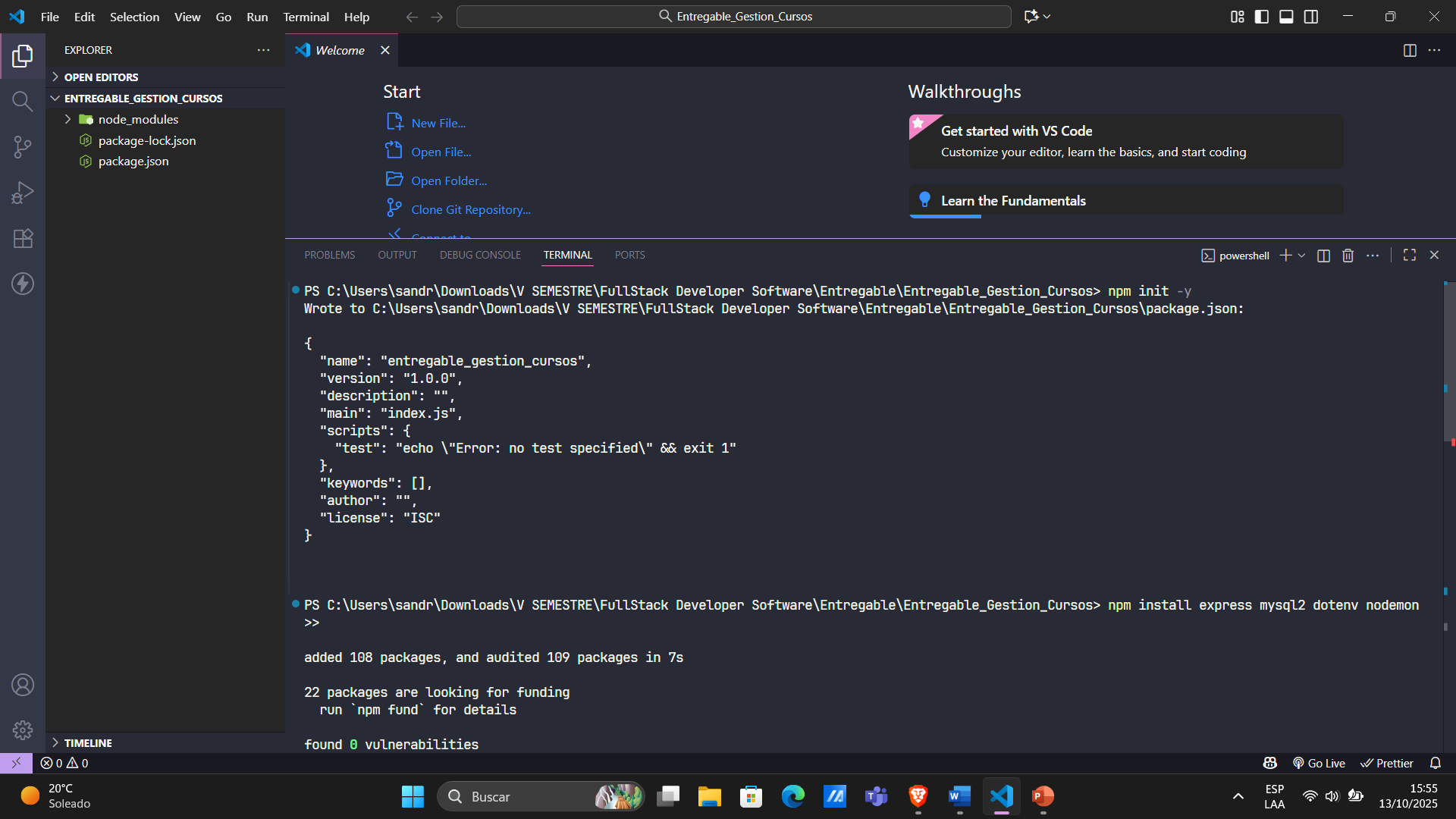
De Docente:

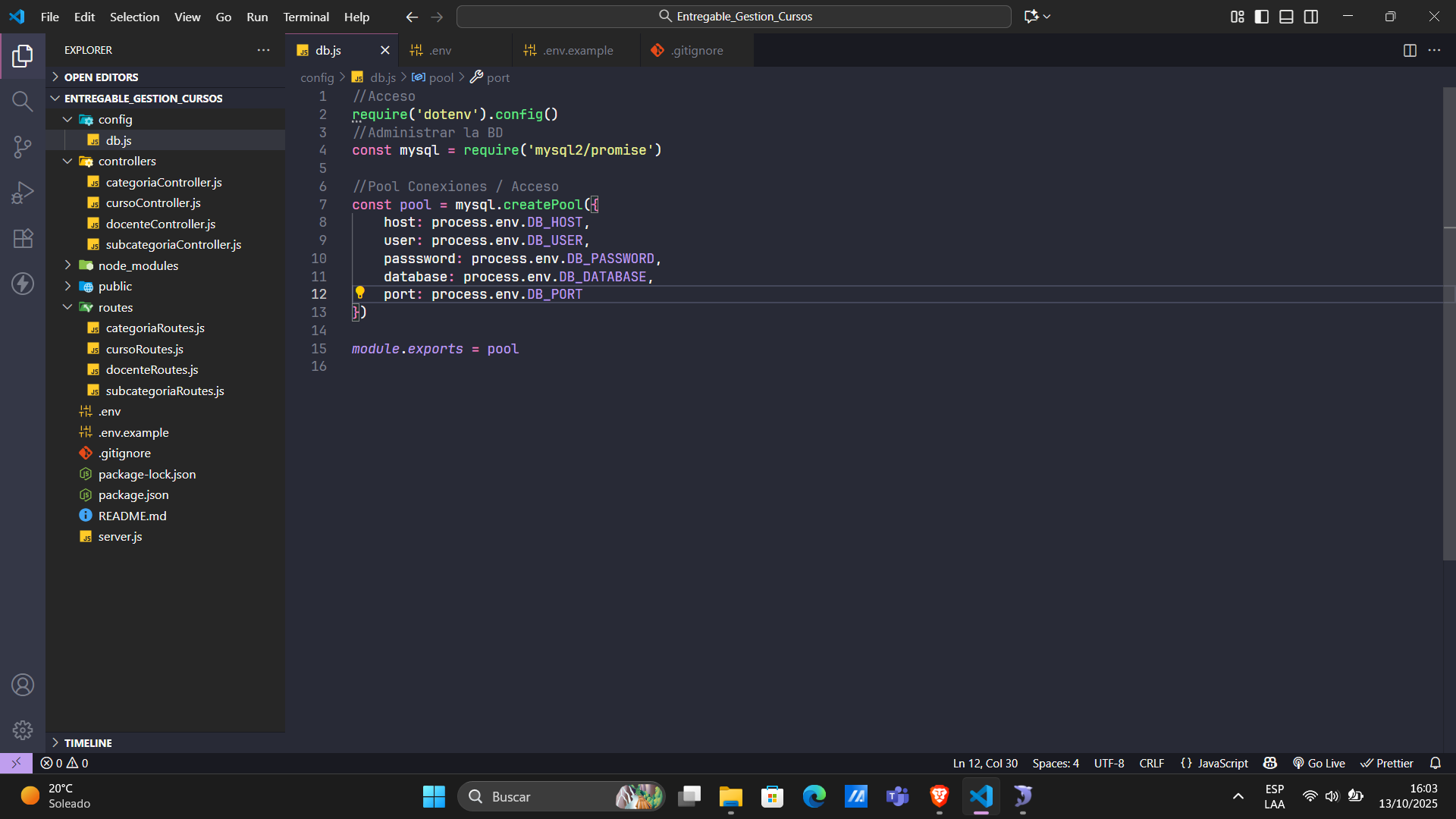


Tabla de Cursos:

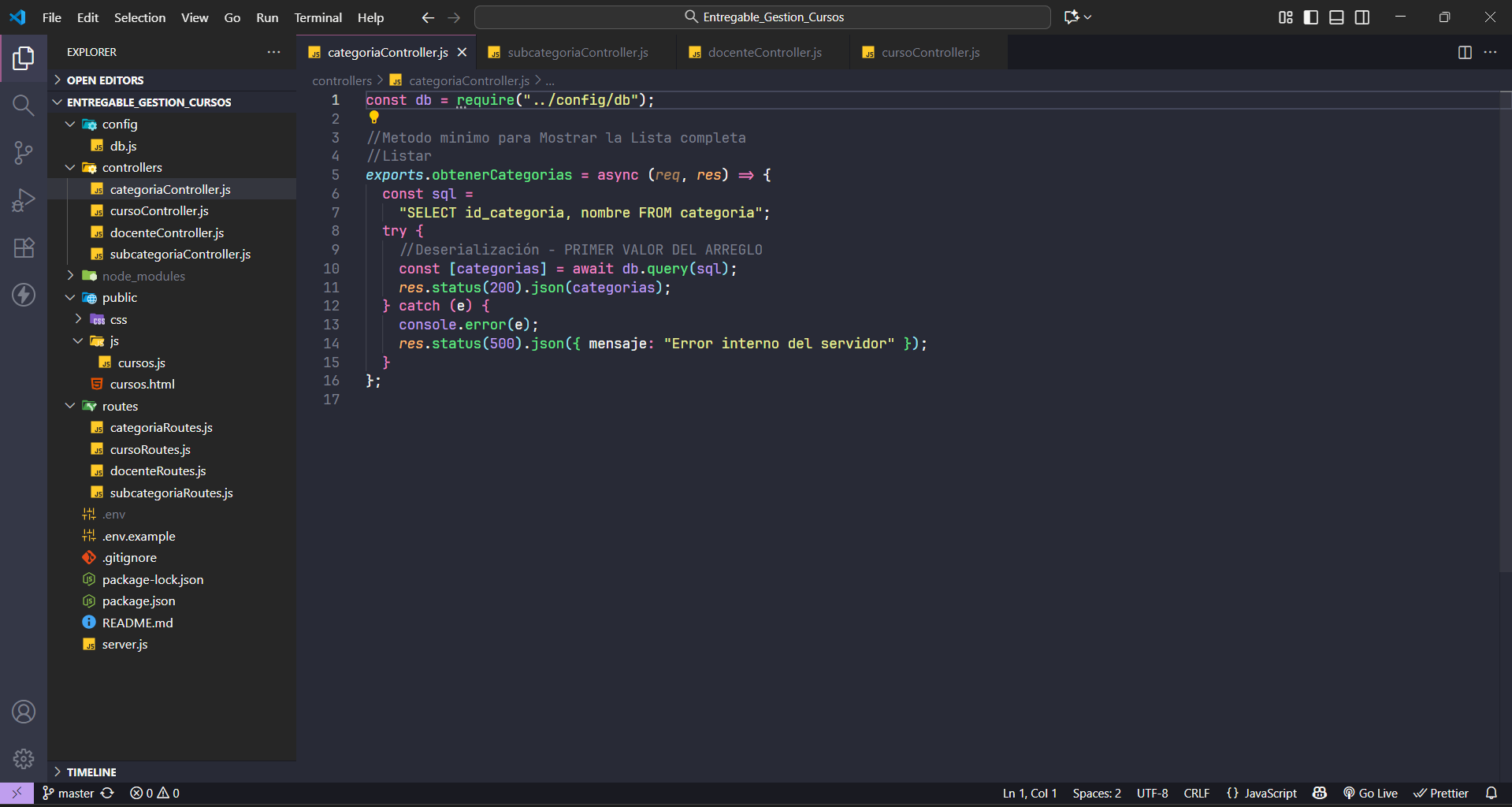
Registros de Curso:



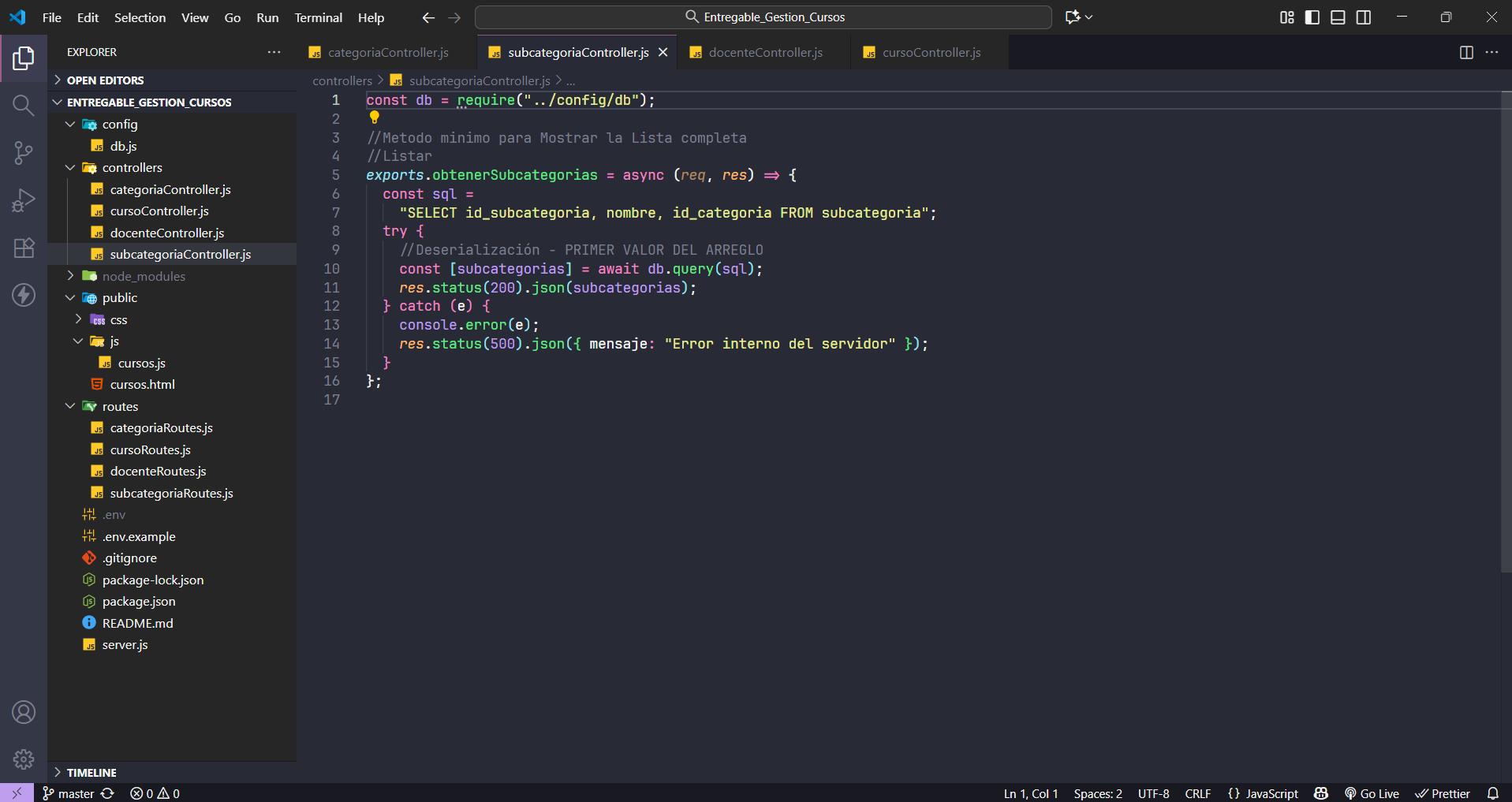
Configuraciones Básicas del Proyecto  


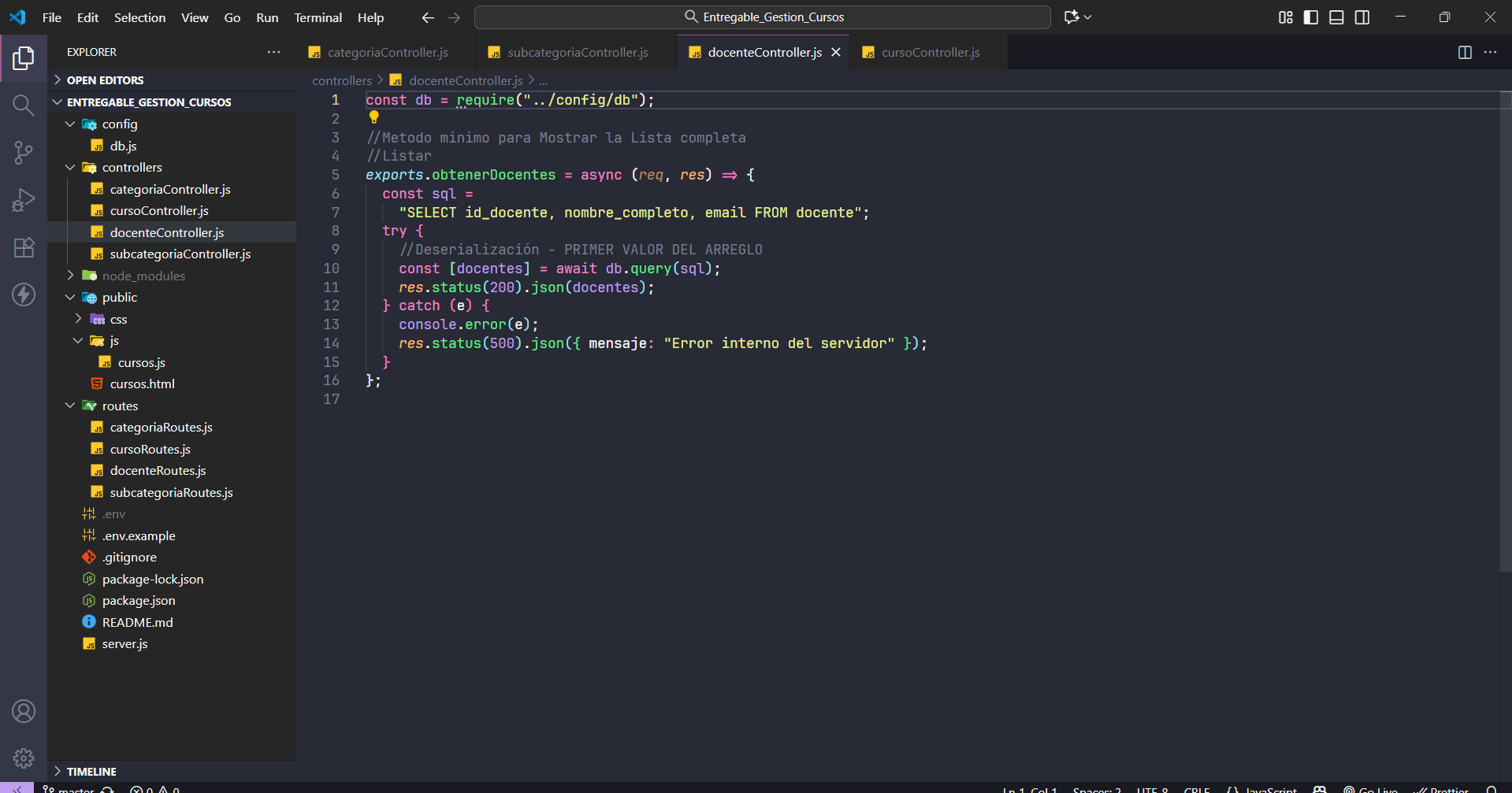


Controllers:  
CategoriaController:

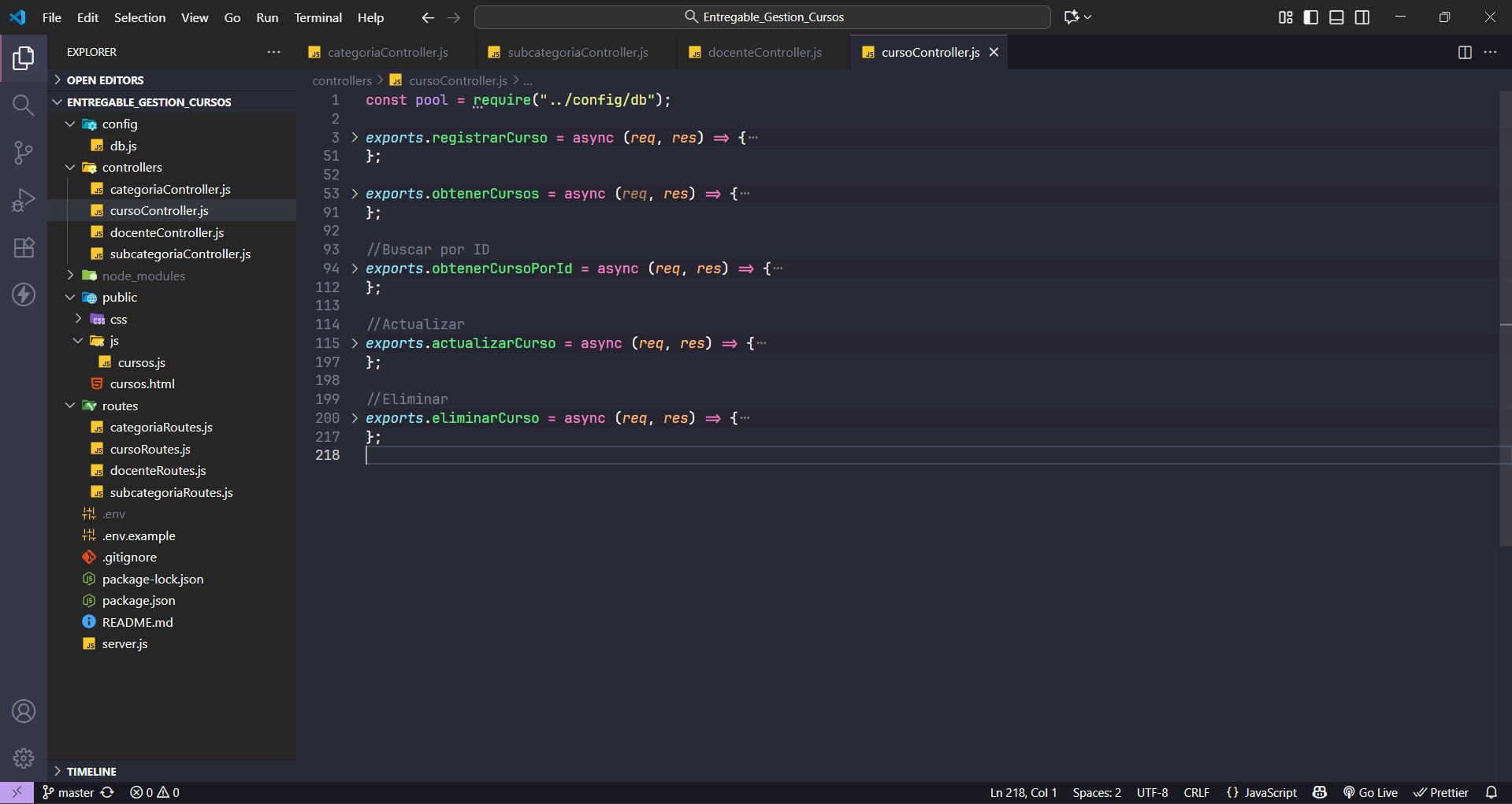


SubCategoriaController:

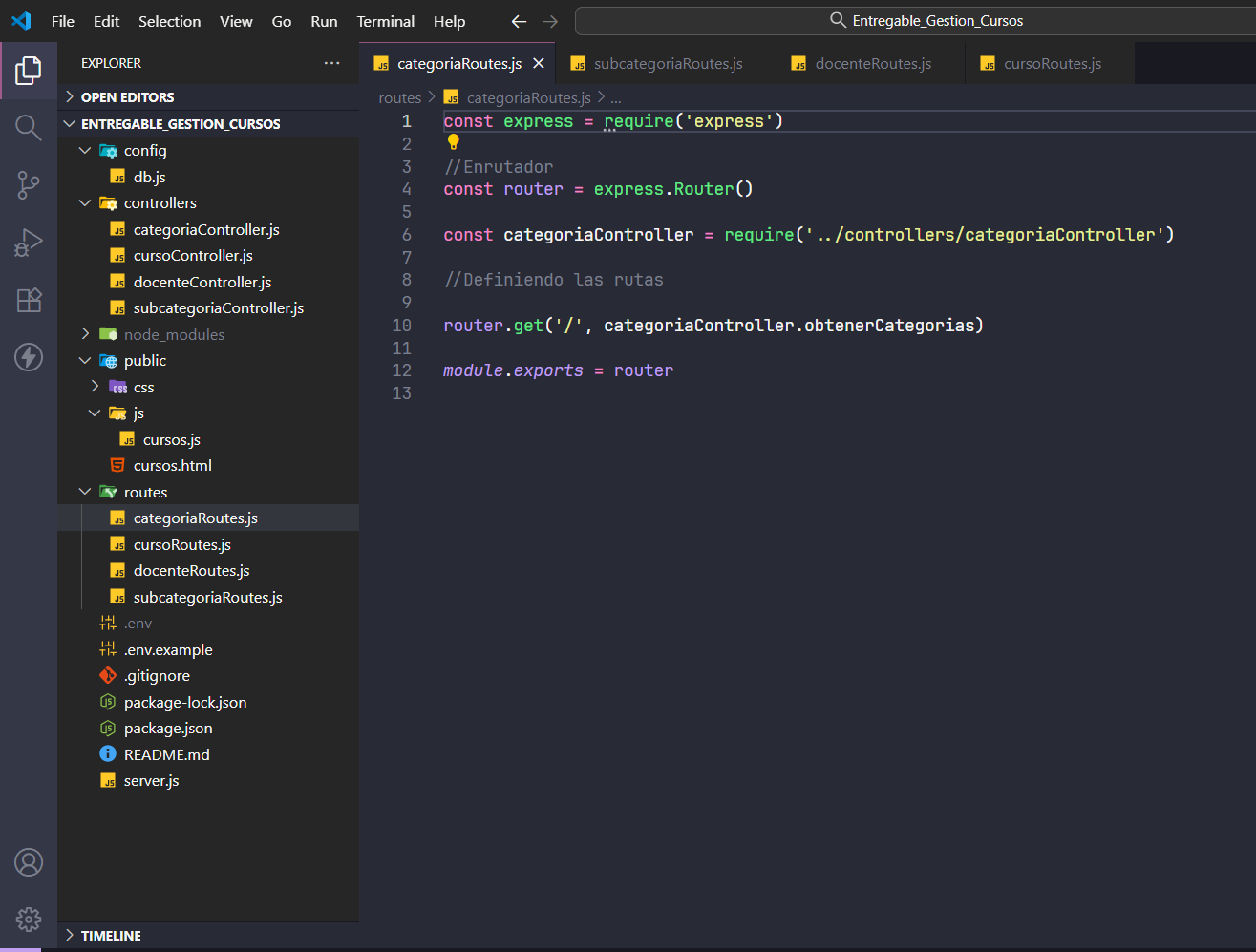


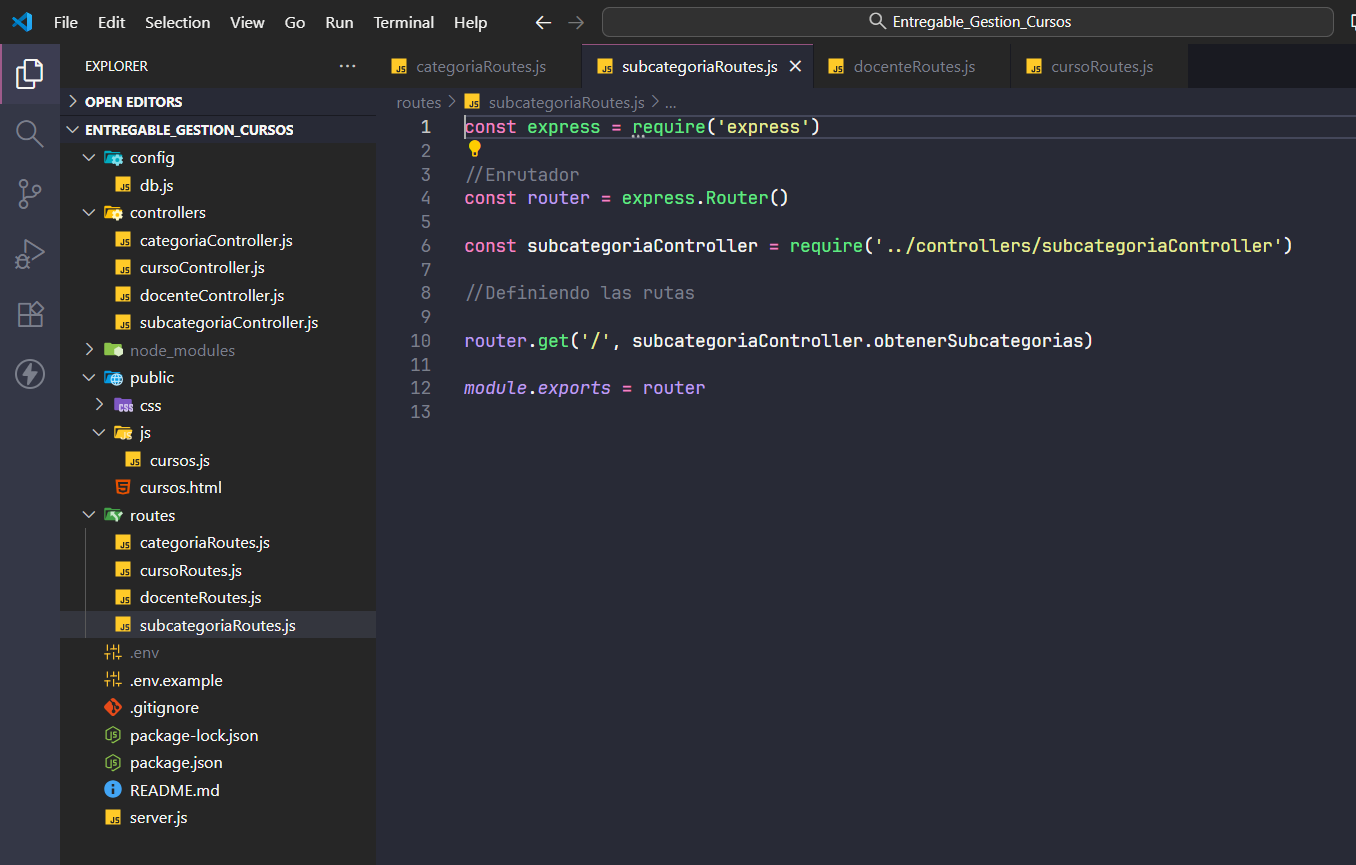
DocenteController:  


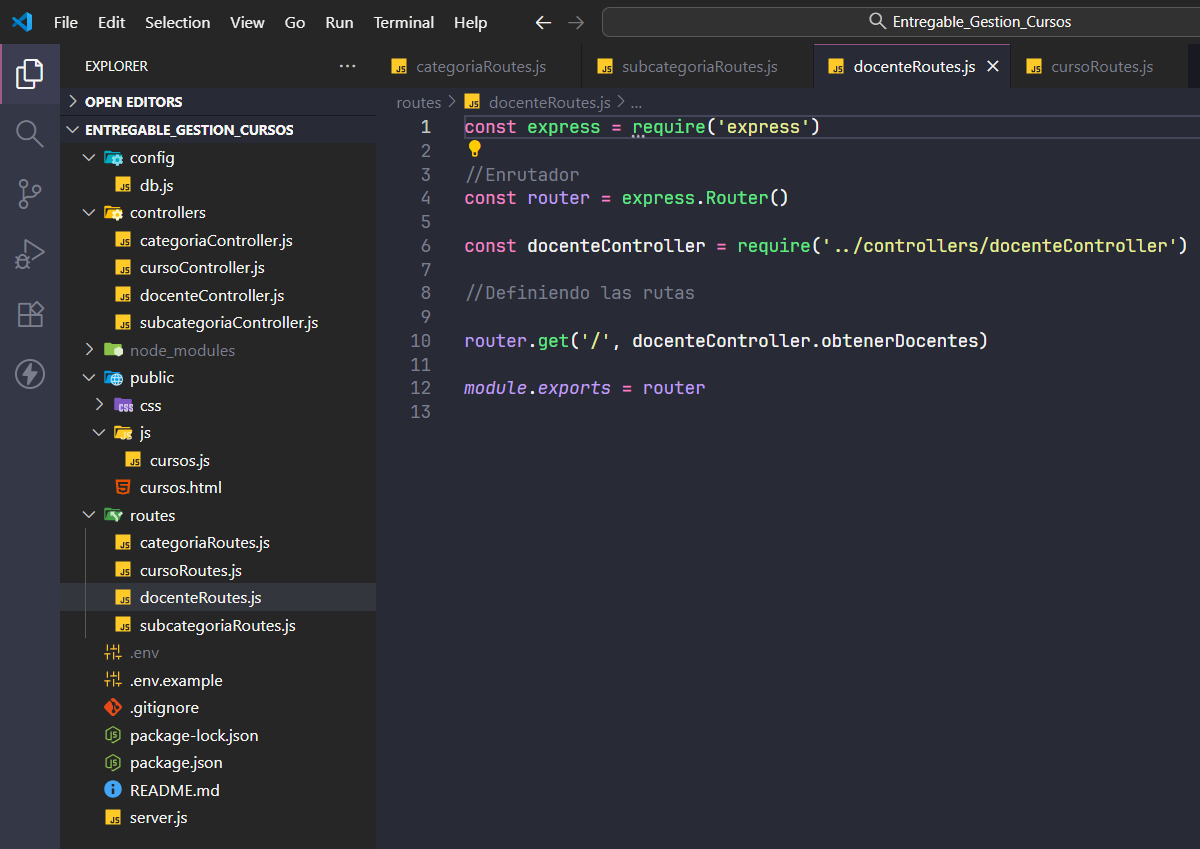
CursoController:



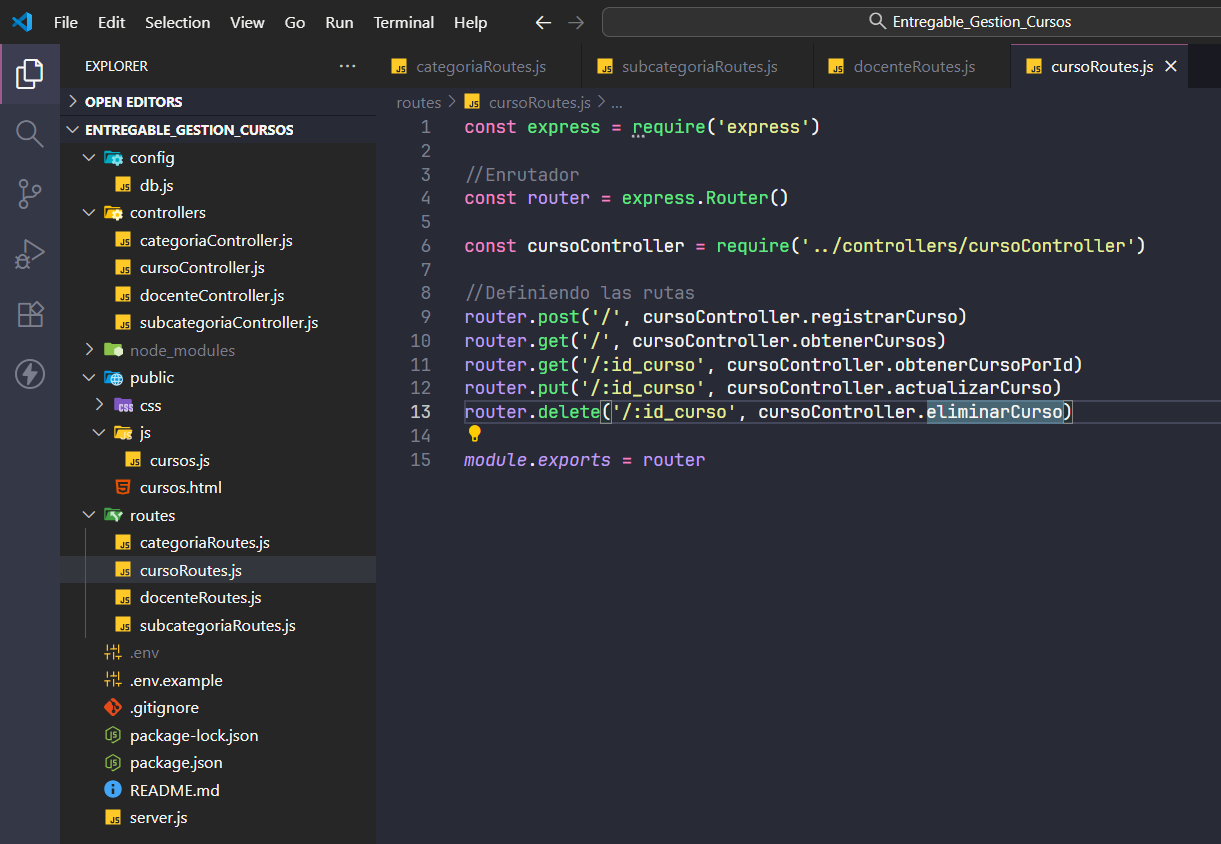
Routes:

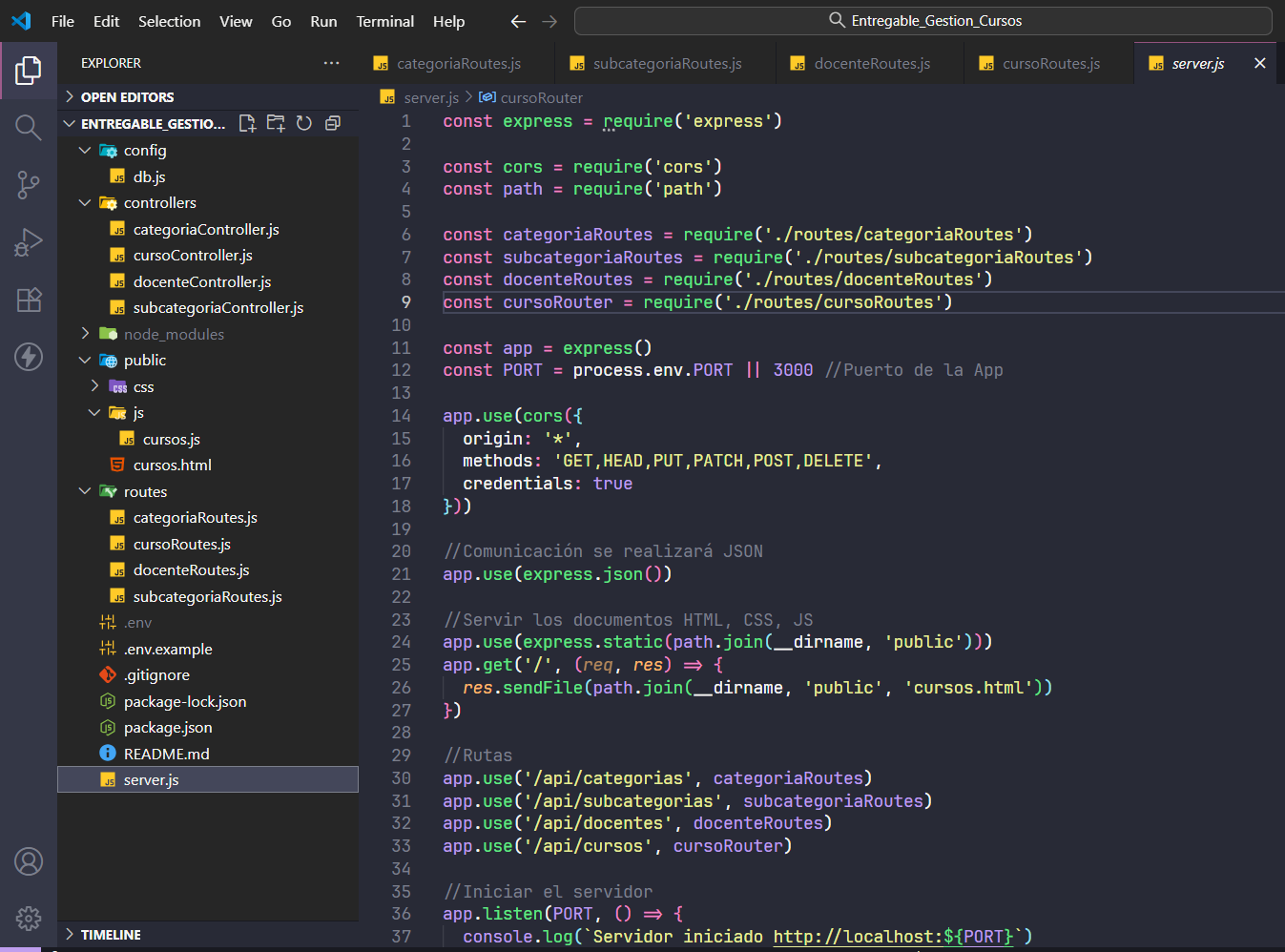
CategoriaRoutes:  


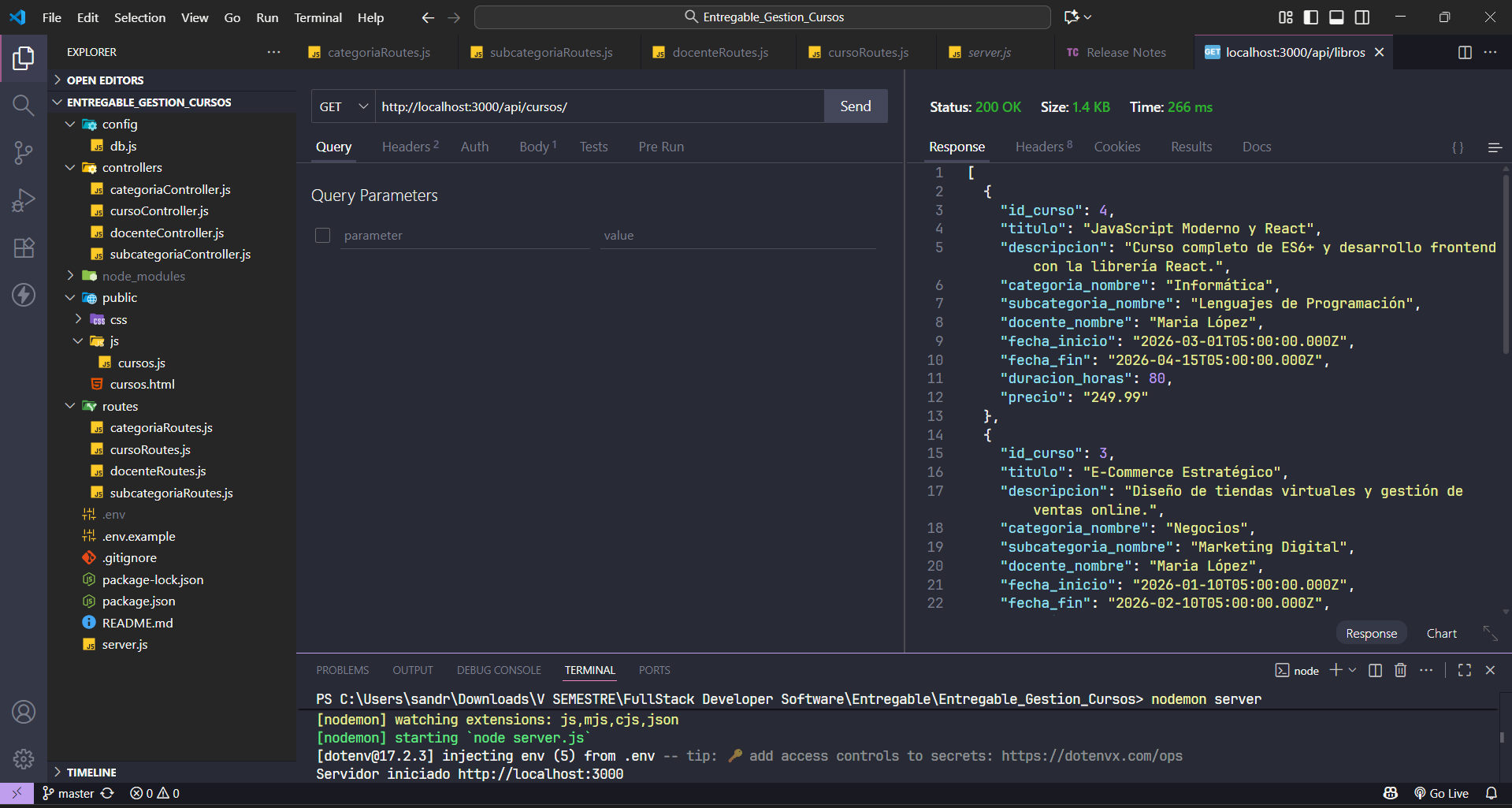
SubCategoriaRoutes:  


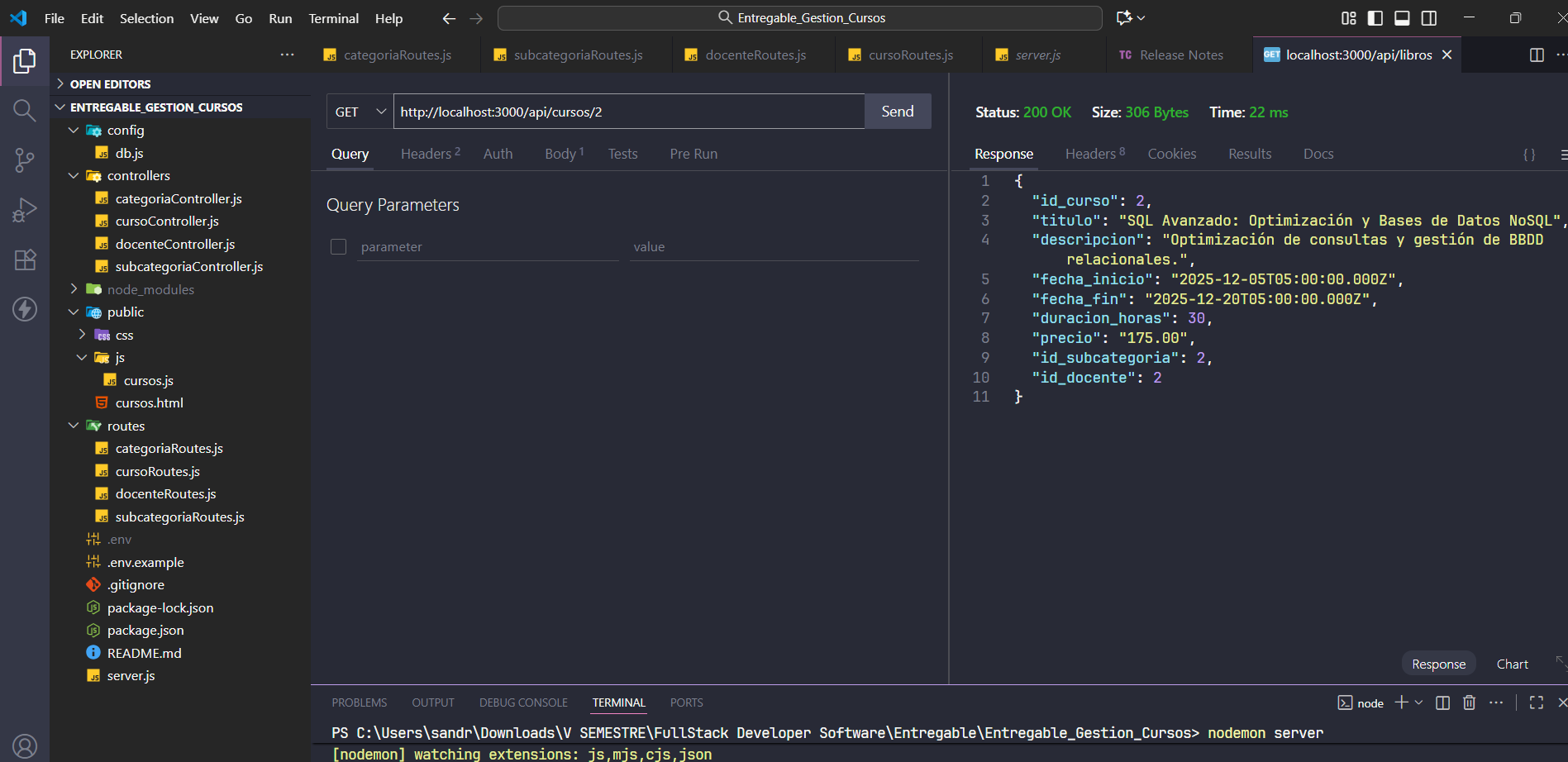
DocenteRoutes:  


CursoRoutes:

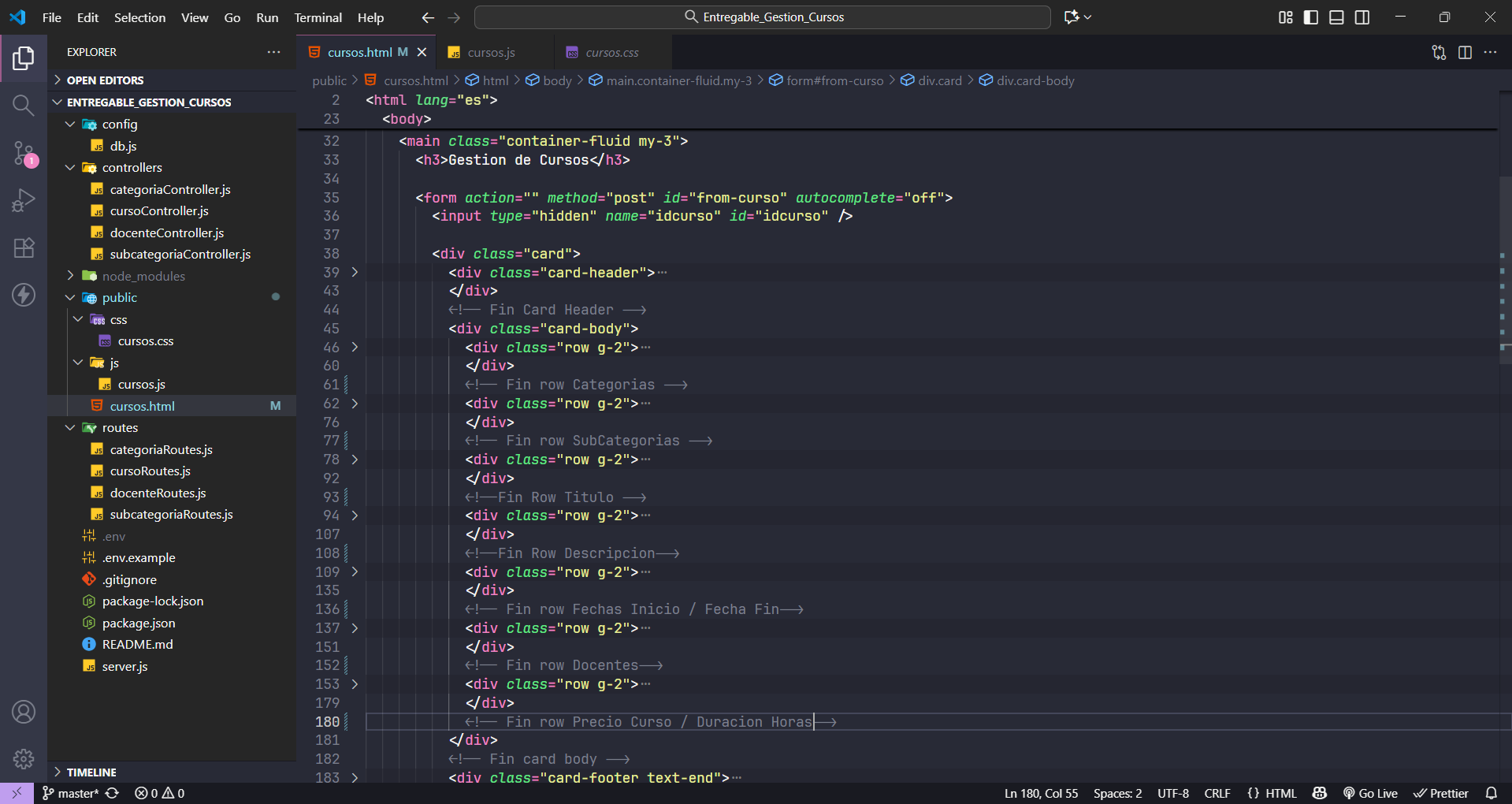


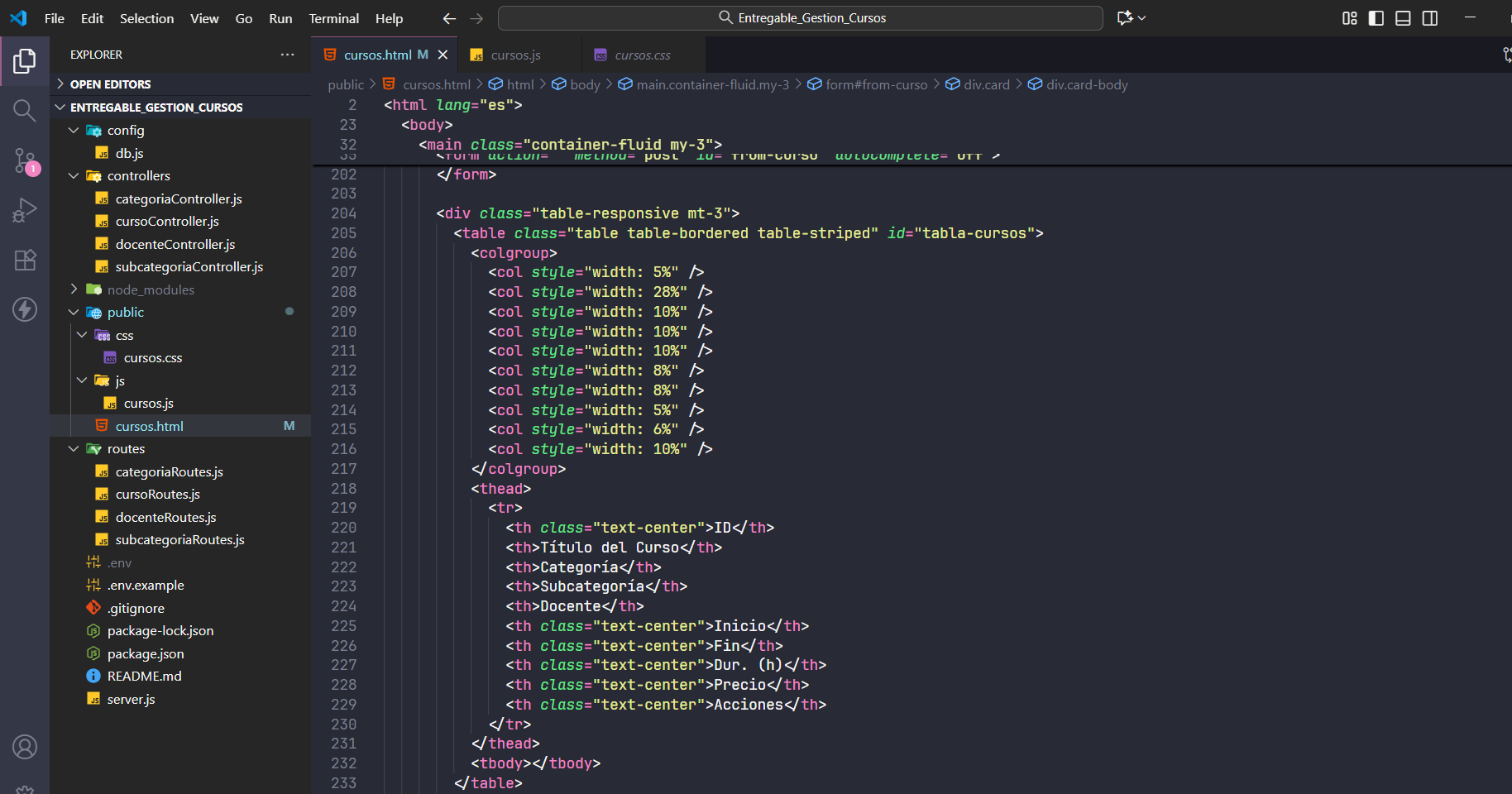
Server.js:  
  
  
Ejecucion en el Backend (ThunderClient)

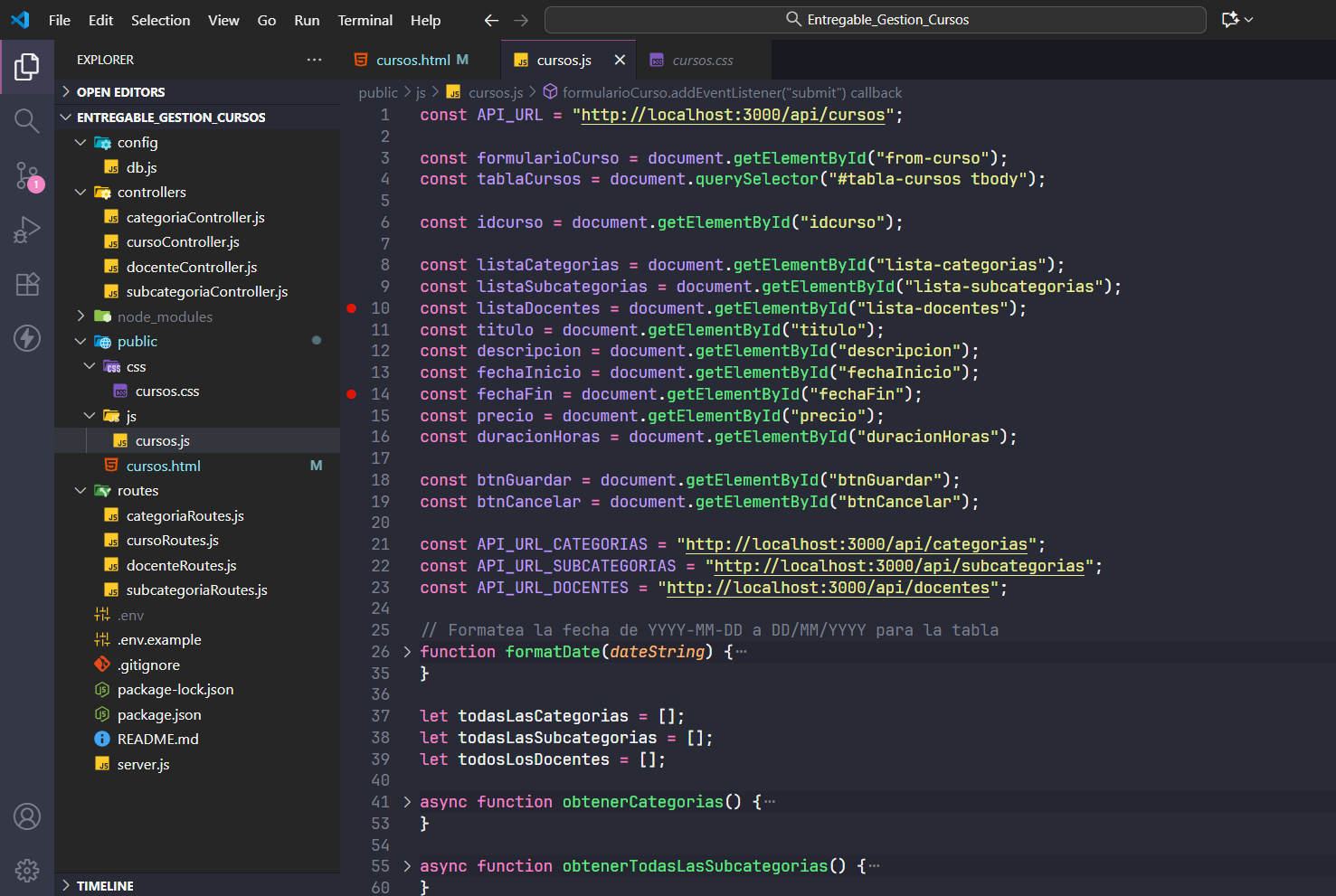


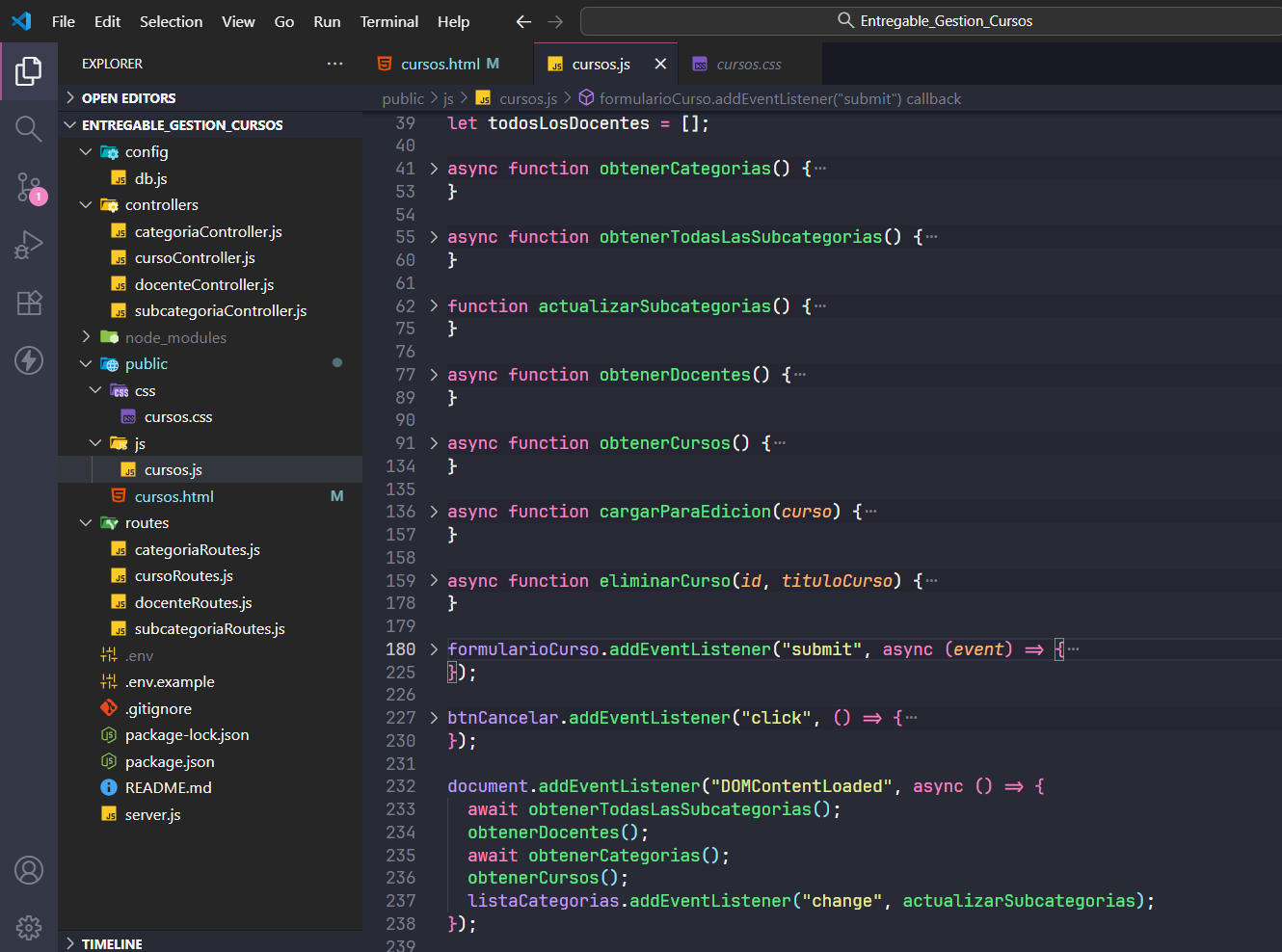


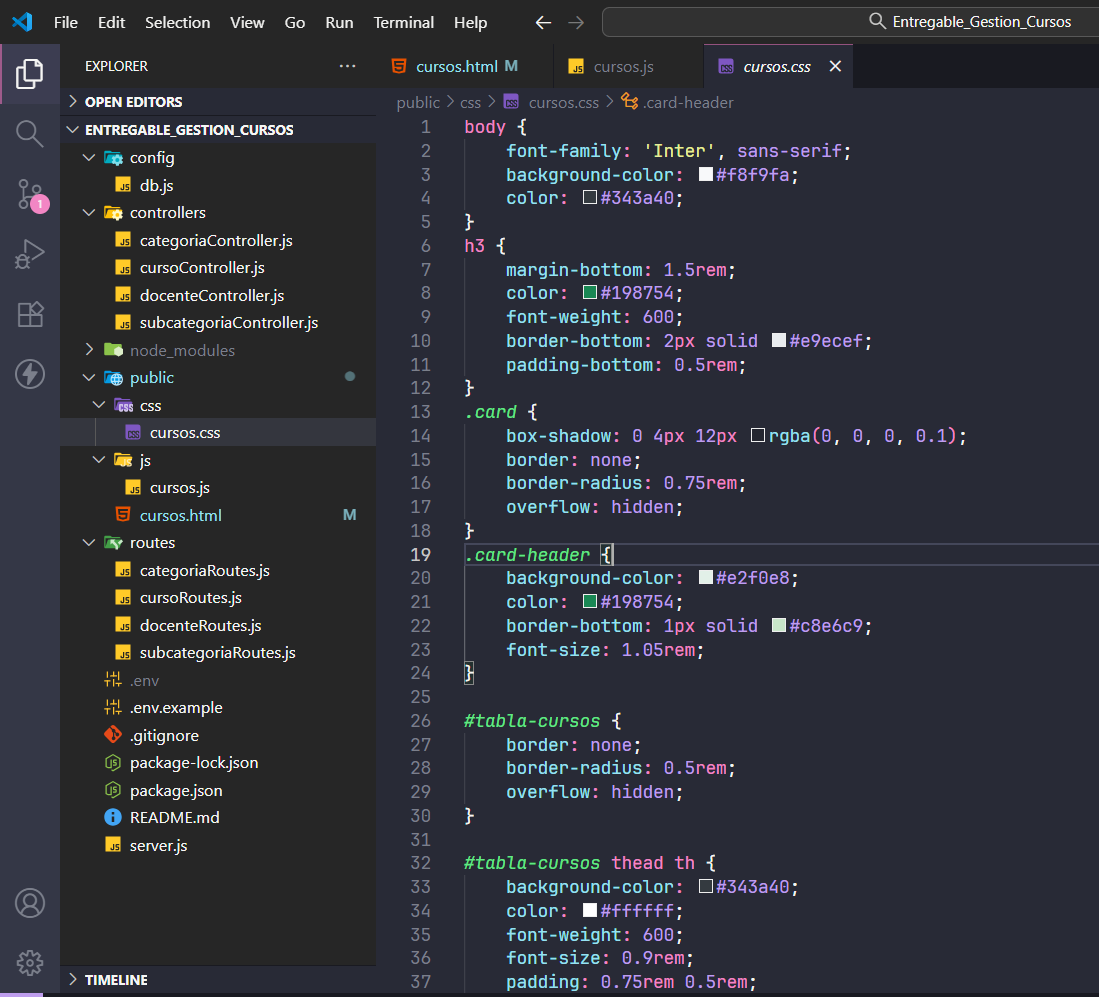
Fronted:

Curso.html:  




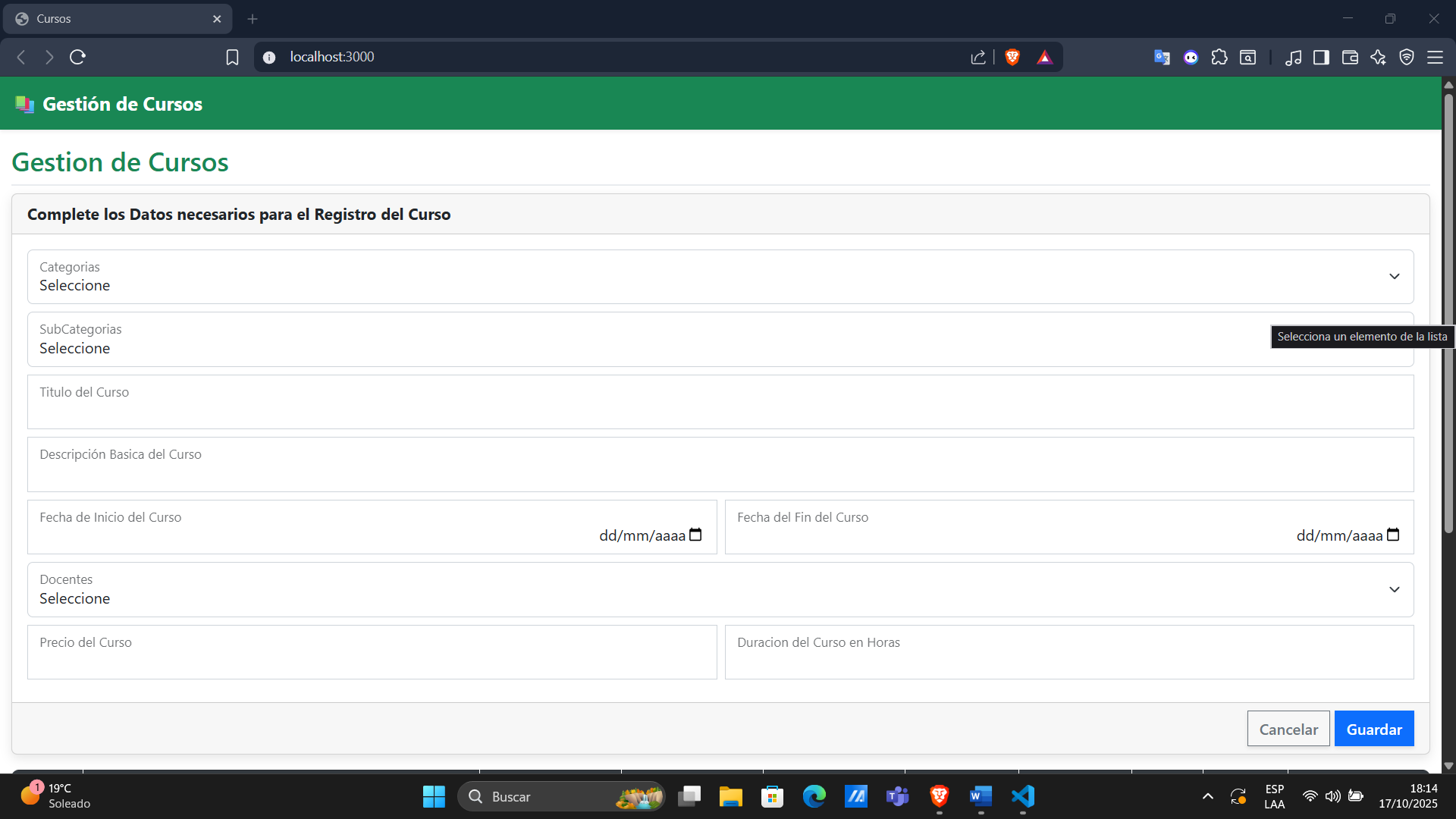
Curso.js:  


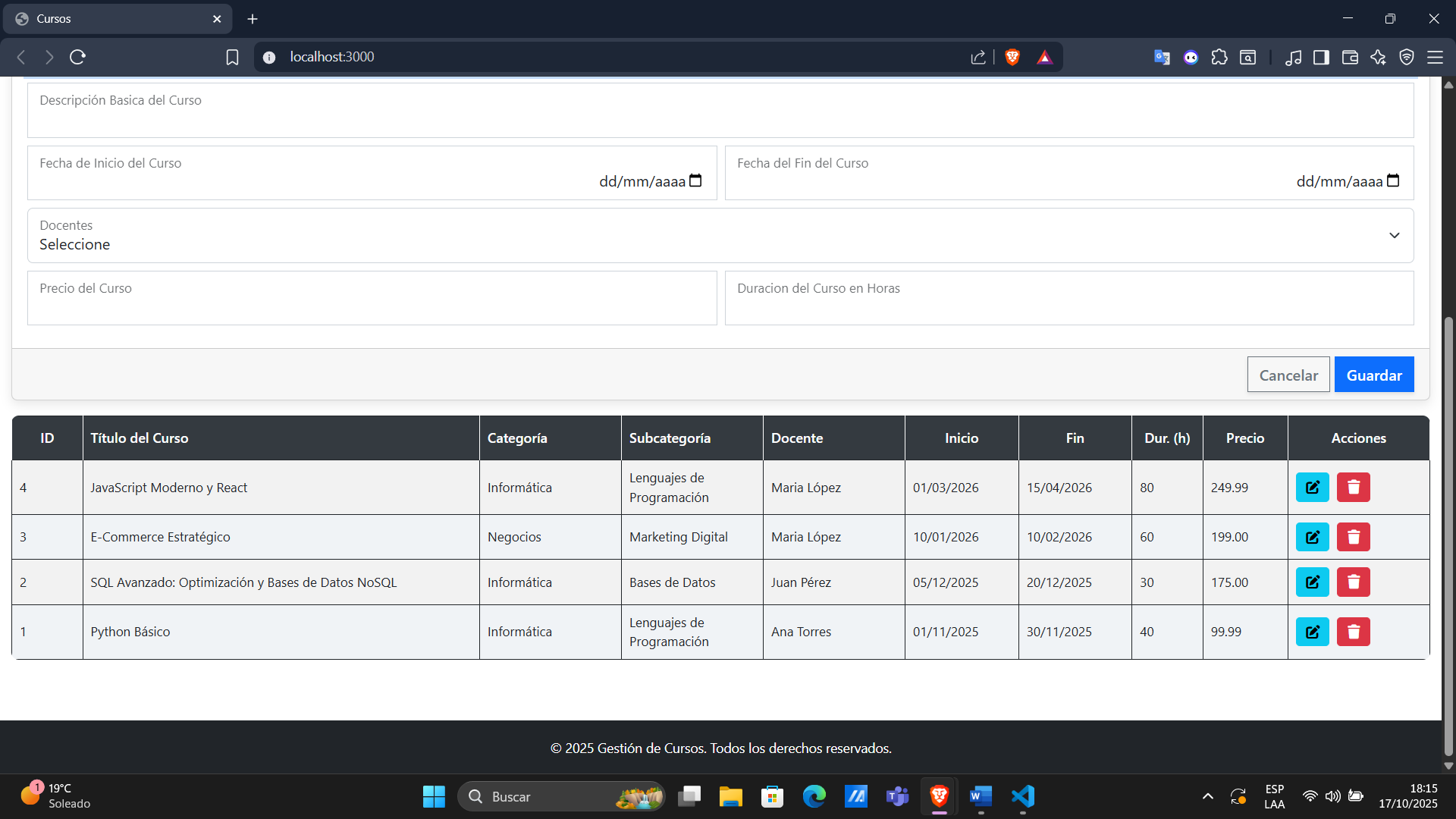


Curso.Css:  


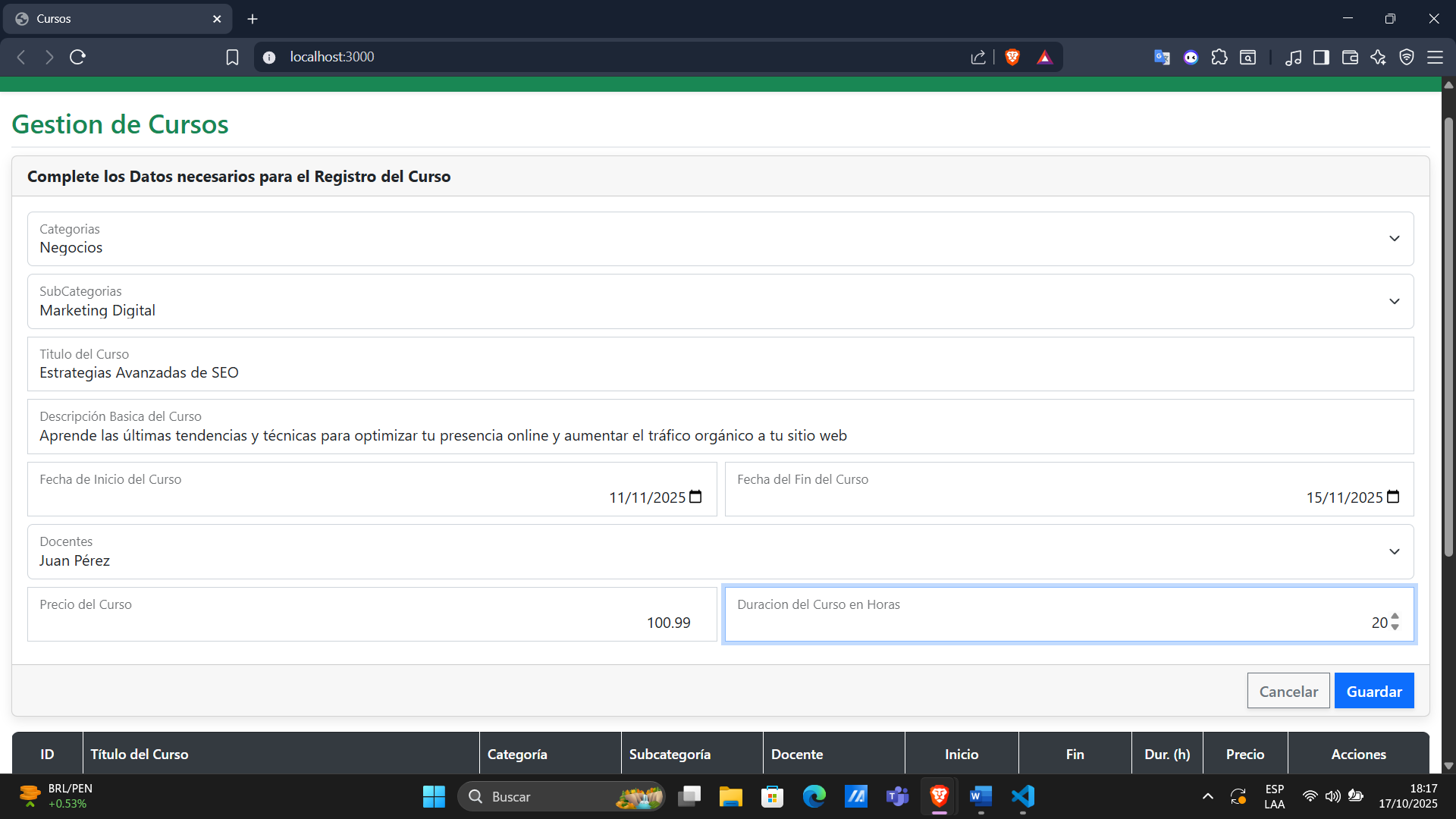
Ejecucion Fronted (Navegador):

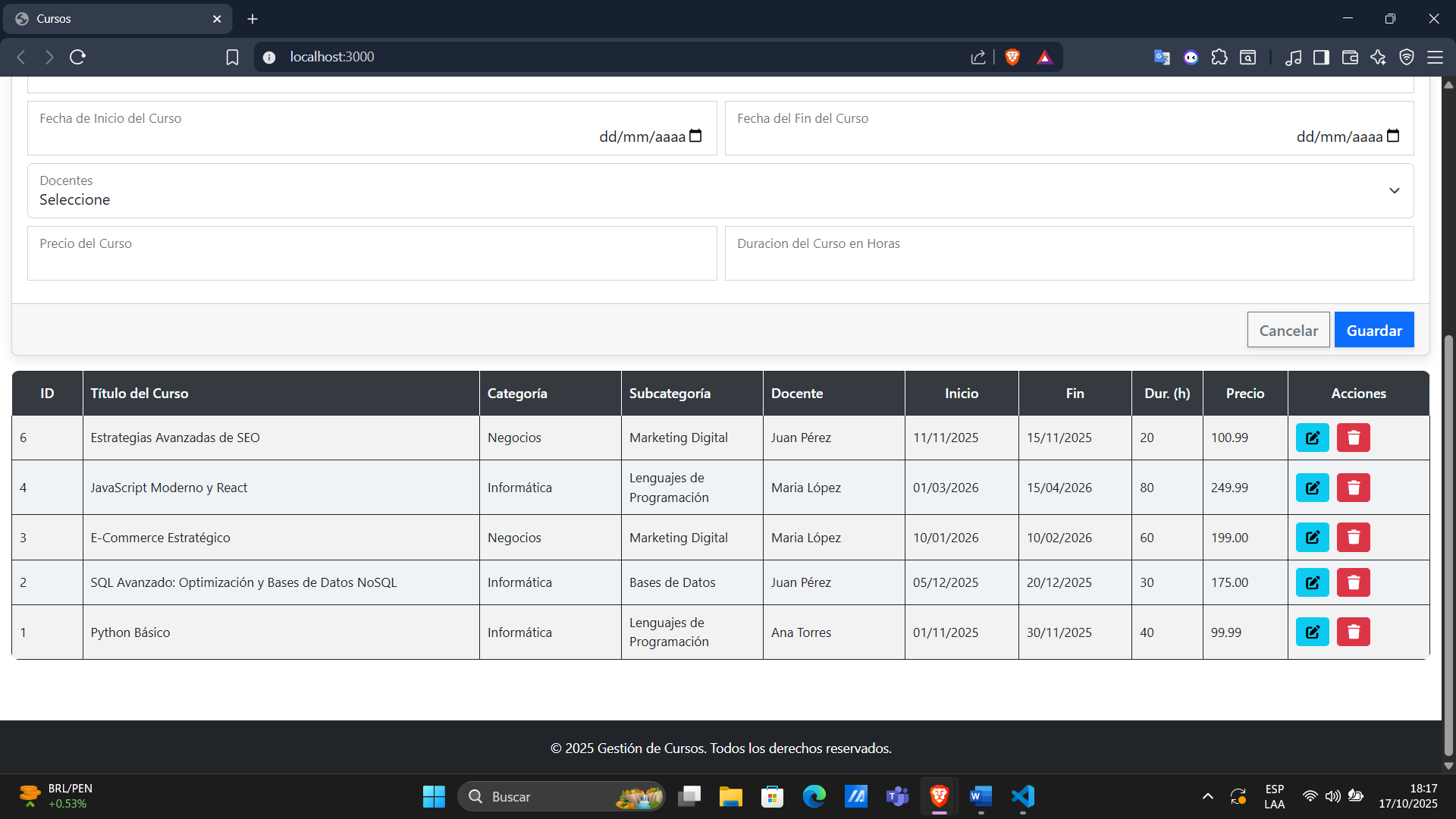
GET



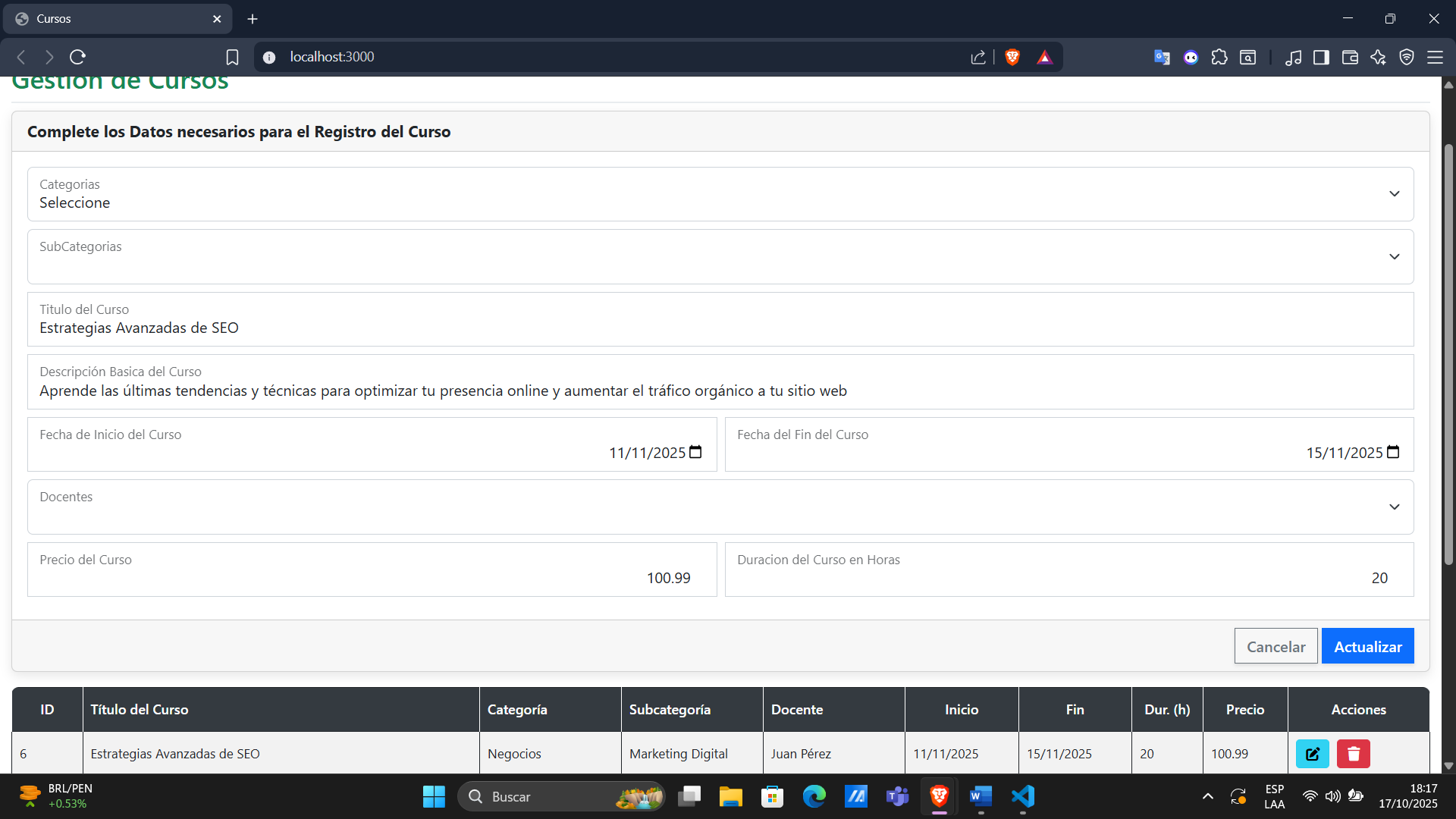


POST:

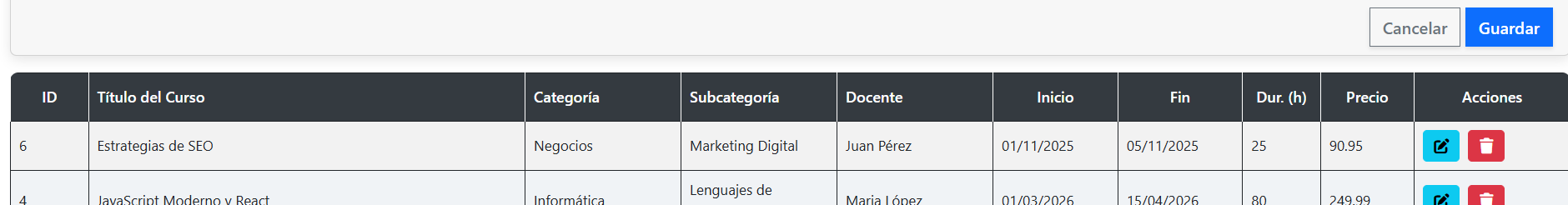


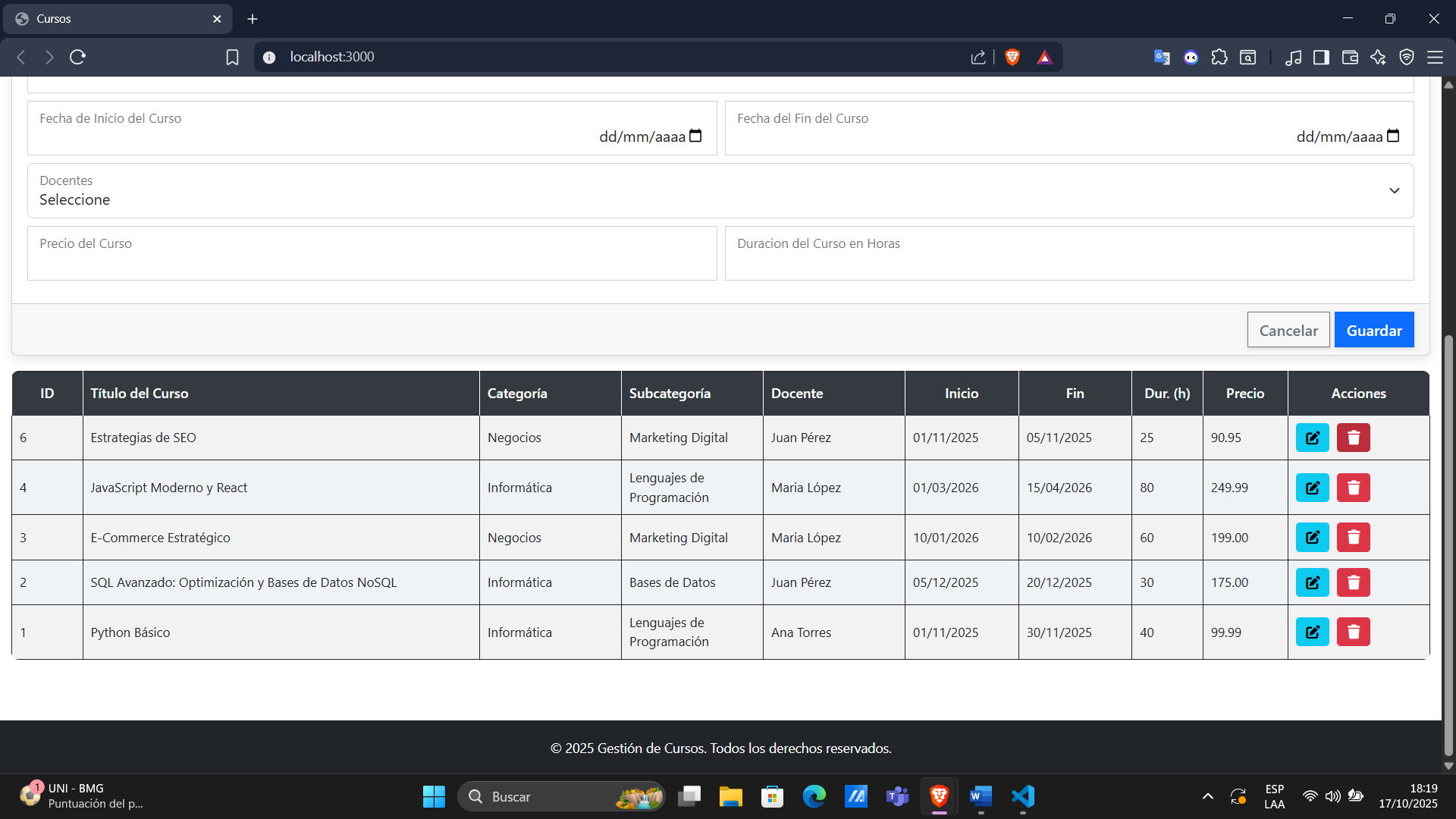


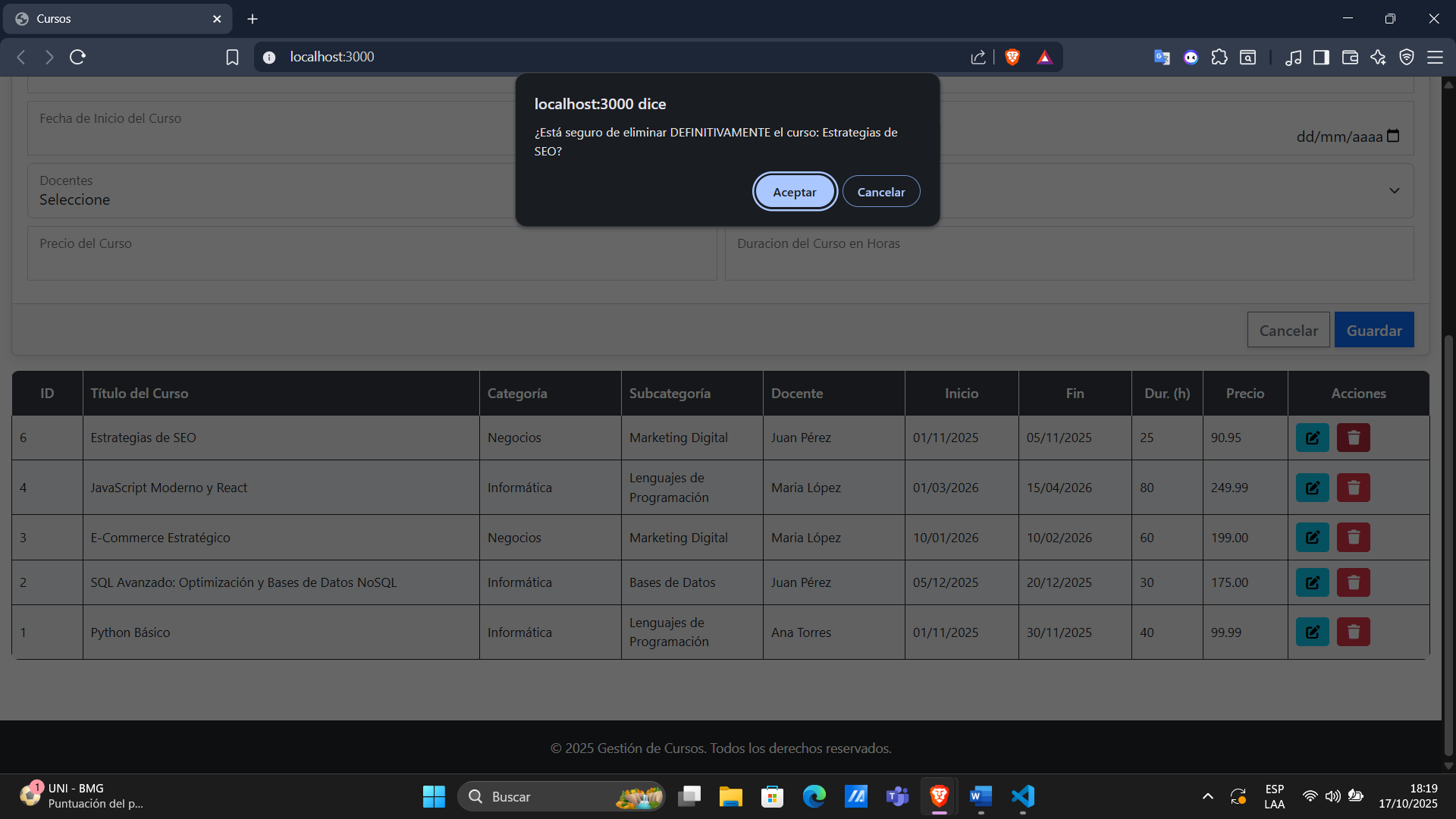
PUT:







DELETE:  




|  |  |  |
| --- | --- | --- |
| C:\Unidad_D\Nuevo Logo\SENATI_FF-01 Modificado.jpg | Entregable 1 - Software Web de Gestion de Cursos | |
| Rodríguez Anchante, Sandro Steven | [ESCALA] |

**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 | 3 |
| Relevancia de la propuesta de solución | La propuesta responde adecuadamente al problema planteado y es relevante para el contexto del caso práctico. | 8 | 6 |
| Viabilidad técnica | La solución es técnicamente factible, tomando en cuenta los recursos y conocimientos disponibles. | 6 | 6 |
| Cumplimiento de Normas | La solución cumple con todas las normas técnicas de seguridad, higiene y medio ambiente. | 3 | 2 |
| **PUNTAJE TOTAL** | | **20** | **17** |

