|  |  |  |
| --- | --- | --- |
| **CORPORACIÓN UNIVERSITARIA PARA EL DESARROLLO EMPRESARIAL Y SOCIAL**  **– CUDES –** | | |
|  | **Programa**: TG. Desarrollo de Software | **Examen**: Validación |
| **Asignatura**: DESARROLLO WEB - LABORATORIO DE PROGRAMACIÓN |
| **Docente**: Samir Vidal | **Fecha**: 28 08 2023 |
| **Nombre Estudiante**: | |
| **Documento de Identidad**: | **Nota**: |
| **Artículo 56: Parágrafo 2:** La calificación definitiva será de cero punto cero (0.0) cuando en cualquier prueba de evaluación el estudiante sea sorprendido en fraude o intento de fraude, ya sea haciéndolo o cooperando con él. | |

**Objetivo**

Implementación de un sistema de gestión de tareas (To-Do List) utilizando Laravel como framework de backend para crear una API y desarrollar vistas que consuman dicha API.

**Requerimiento**

* **Descripción General del Sistema:** Desarrollar una aplicación web que permita a los usuarios gestionar su lista de tareas pendientes. Los usuarios podrán registrarse, iniciar sesión, agregar nuevas tareas, marcar tareas como completadas y eliminar tareas. La aplicación constará de un backend desarrollado en Laravel que expondrá una API para la manipulación de las tareas, y un frontend que consumirá esta API para mostrar la interfaz de usuario.
* **Backend con Laravel:** Crear un backend utilizando el framework Laravel. Definir los siguientes endpoints de la API:

- **/api/register** - Permitirá a los usuarios registrarse en la plataforma.

- **/api/login** - Permitirá a los usuarios iniciar sesión.

- **/api/tasks** - Endpoints para la gestión de tareas (crear, listar, marcar como completadas, eliminar).

Implementar un sistema de autenticación para asegurar los endpoints.

* **Base de Datos:** Diseñar y configurar una base de datos que almacene la información de los usuarios y las tareas. La tabla de tareas debe contemplar algunos campos como título, descripción y estado (pendiente o completada).
* **Vistas de Consumo:** Desarrollar las vistas utilizando HTML, CSS y JS. Debería incluir:

- Página de registro de usuario.

- Página de inicio de sesión.

- Página principal de la lista de tareas, mostrando las tareas pendientes y completadas.

- Formulario para agregar nuevas tareas.

- Opciones para marcar tareas como completadas y eliminar tareas.

* **Interacción Frontend-Backend:** Utilizar JavaScript para interactuar con la API desde el frontend. Emplear técnicas como Fetch API o librerías como Axios para realizar peticiones a los endpoints de la API. Actualizar dinámicamente las vistas en función de las respuestas de la API.
* **Validación de Datos:** Implementar validación tanto en el frontend como en el backend para asegurar que los datos ingresados por los usuarios sean válidos y seguros.
* **Documentación:** Proporcionar documentación clara y concisa sobre cómo ejecutar el proyecto localmente, incluyendo la configuración de la base de datos, la migración de tablas y la ejecución del servidor Laravel. También se debe explicar cómo se estructuran los endpoints de la API y cómo interactúan con las vistas.

**Plazo**

* **Avances:** miércoles 27 de septiembre 2023
* **Final:** miércoles 1 de noviembre 2023