

Challenge 1: Aplicación de Gestión de Tareas

Dev Full Stack Java y Javascript

Descripción del Desafío

Crea una aplicación web de gestión de tareas (To-Do List) donde los usuarios puedan:

1. Crear, leer, actualizar y eliminar tareas.
2. Asignar prioridades a las tareas (baja, media, alta).
3. Marcar tareas como completadas.
4. Filtrar tareas por estado (completadas/no completadas) y por prioridad.

Requisitos

Backend:

1. Crea una API RESTful con las siguientes funcionalidades:
 - **GET /tasks:** Obtener todas las tareas.
 - **POST /tasks:** Crear una nueva tarea.
 - **PUT /tasks/{id}:** Actualizar una tarea existente.
 - **DELETE /tasks/{id}:** Eliminar una tarea.
 - **GET /tasks/filter:** Filtrar tareas por estado y/o prioridad.
2. Utiliza JPA/Hibernate para la persistencia de datos en una base de datos relacional (puedes usar H2 para simplificar).
3. Debe utilizarse Java 11 o superior y Sprign Boot
4. Incluir un Readme donde se indique como configurarlo y correrlo.

Frontend:

1. Crea una interfaz de usuario para interactuar con las API del backend.
2. La interfaz debe permitir:
 - Ver la lista de tareas.
 - Añadir una nueva tarea.
 - Editar una tarea existente.
 - Eliminar una tarea.
 - Filtrar tareas por estado y prioridad.
3. Utilizar Vue.js (Si no conoces Vue, puedes utilizar alguna otra tecnología como React o Angular)
4. Se puede utilizar cualquier Framework de estilos (Tailwindcss, Bootstrap)
5. Usa npm para la gestión de paquetes y la compilación del proyecto.
6. Incluir un Readme donde se indique como configurarlo y correrlo.

Opcionales

- Sistema para autenticación de los servicios (Token, cookies, etc.)
- Interfaz de Login
- Validaciones para el formulario frontend y backend
- Documentar el backend con Swagger
- Test unitarios

Notas Adicionales

- Asegúrate de manejar los errores adecuadamente, proporcionando respuestas claras y significativas para los errores comunes (e.g., 404 Not Found, 400 Bad Request).
- Implementa validaciones en el backend para garantizar que los datos proporcionados son válidos.
- En el frontend, utiliza formularios de validación y proporciona retroalimentación visual a los usuarios.
- Considera la implementación de autenticación básica para el acceso a las funcionalidades de creación, actualización y eliminación de eventos.

Criterios de Evaluación

- **Funcionalidad:** Compleción y corrección de las funcionalidades requeridas.
- **Calidad del Código:** Legibilidad, organización y buenas prácticas en el código tanto del frontend como del backend.
- **Diseño de la API:** Correcta estructuración de los endpoints y manejo de errores.
- **UI/UX:** Calidad del diseño de la interfaz de usuario y experiencia de usuario.

- **Pruebas:** Implementación de pruebas unitarias y/o de integración para el backend y el frontend.
- **Documentación:** Claridad y suficiencia de la documentación del código y del proyecto en general.