**aplicación de gestión de tareas (To-Do List)**

**Historias de Usuario**

**1. Creación de Tareas**

* **Descripción:** Como usuario, quiero crear nuevas tareas para gestionar mis actividades.
* **Tareas:**
  1. Crear formulario de nueva tarea con campos: título, descripción, fecha de vencimiento, prioridad.
  2. Implementar endpoint para crear tareas en el backend.
  3. Guardar tarea en la base de datos.
  4. Mostrar tareas creadas en la interfaz del usuario.

**2. Listado de Tareas**

* **Descripción:** Como usuario, quiero ver una lista de todas las tareas o tareas por usuario.
* **Tareas:**
  1. Implementar endpoint para obtener todas las tareas del usuario en el backend.
  2. Crear componente en React para mostrar la lista de tareas.
  3. Manejar paginación y filtros (opcional).
  4. Actualizar lista en tiempo real tras la creación, edición o eliminación de tareas.

**3. Edición de Tareas**

* **Descripción:** Como usuario, quiero editar las tareas existentes para actualizar la información.
* **Tareas:**
  1. Crear formulario de edición de tarea.
  2. Prellenar formulario con datos existentes de la tarea.
  3. Implementar endpoint para actualizar tarea en el backend.
  4. Guardar cambios en la base de datos.
  5. Actualizar lista de tareas en el frontend tras la edición.

**4. Eliminación de Tareas**

* **Descripción:** Como usuario, quiero eliminar tareas que ya no necesito.
* **Tareas:**
  1. Implementar botón de eliminación en cada tarea de la lista.
  2. Mostrar confirmación antes de eliminar.
  3. Implementar endpoint para eliminar tarea en el backend.
  4. Remover tarea de la base de datos.
  5. Actualizar lista de tareas en el frontend tras la eliminación.

**5. Marcado de Tareas como Completadas o canceladas**

* **Descripción:** Como usuario, quiero marcar tareas como completadas o canceladas para mantener un seguimiento.
* **Tareas:**
  1. Implementar checkbox o botón para marcar tareas como completadas o canceladas.
  2. Implementar endpoint para actualizar estado de la tarea en el backend.
  3. Guardar estado de completado o cancelado en la base de datos.
  4. Visualmente diferenciar tareas completadas o canceladas en la lista.

**6 Filtros y Búsqueda de Tareas**

* **Descripción:** Como usuario, quiero buscar y filtrar por tareas o por usuarios para encontrar rápidamente la información.
* **Tareas:**
  1. Implementar barra de búsqueda en el frontend.
  2. Implementar filtros por fecha, prioridad y estado.
  3. Implementar endpoints en el backend para búsqueda y filtrado.
  4. Actualizar lista de tareas basada en criterios de búsqueda y filtros.

* **Requisitos**
  + La api del backend debe de tener basic auth
  + Si una tarea no esta completada o cancelada se debe asumir que esta pendiente
  + El frontend debe ser responsive y intuitivo
  + Para el modelado de las tablas en el backend utilizar un ORM (JPA e hibernate)
  + Se debe de contar con una tabla de usuarios, con id y nombre y asociarle las tareas a los usuarios

**Funcionalidades:**

1. **CRUD de Tareas**
   * Crear, leer, actualizar y eliminar tareas.
   * Cada tarea debe tener un título, descripción, fecha de vencimiento y estado (completada/no completada).
2. **Filtrado y Ordenamiento**
   * Filtrar tareas por estado (completadas/no completadas).
   * Ordenar tareas por fecha de vencimiento o por prioridad.
3. **Interfaz de Usuario Amigable**
   * Un diseño simple y responsivo para la gestión de tareas.

**Historias de Usuario**

**1. Creación de Tareas**

* **Descripción:** Como usuario, quiero crear nuevas tareas para gestionar mis actividades.
* **Tareas:**
  1. Crear formulario de nueva tarea con campos: título, descripción, fecha de vencimiento, prioridad.
  2. Implementar endpoint para crear tareas en el backend.
  3. Guardar tarea en la base de datos.
  4. Mostrar tareas creadas en la interfaz del usuario.

**2. Listado de Tareas**

* **Descripción:** Como usuario, quiero ver una lista de todas mis tareas.
* **Tareas:**
  1. Implementar endpoint para obtener todas las tareas del usuario en el backend.
  2. Crear componente en React para mostrar la lista de tareas.
  3. Manejar paginación y filtros (opcional).
  4. Actualizar lista en tiempo real tras la creación, edición o eliminación de tareas.

**3. Edición de Tareas**

* **Descripción:** Como usuario, quiero editar las tareas existentes para actualizar la información.
* **Tareas:**
  1. Crear formulario de edición de tarea.
  2. Prellenar formulario con datos existentes de la tarea.
  3. Implementar endpoint para actualizar tarea en el backend.
  4. Guardar cambios en la base de datos.
  5. Actualizar lista de tareas en el frontend tras la edición.

**4. Eliminación de Tareas**

* **Descripción:** Como usuario, quiero eliminar tareas que ya no necesito.
* **Tareas:**
  1. Implementar botón de eliminación en cada tarea de la lista.
  2. Mostrar confirmación antes de eliminar.
  3. Implementar endpoint para eliminar tarea en el backend.
  4. Remover tarea de la base de datos.
  5. Actualizar lista de tareas en el frontend tras la eliminación.

**5. Marcado de Tareas como Completadas**

* **Descripción:** Como usuario, quiero marcar tareas como completadas para mantener un seguimiento.
* **Tareas:**
  1. Implementar checkbox o botón para marcar tareas como completadas.
  2. Implementar endpoint para actualizar estado de la tarea en el backend.
  3. Guardar estado de completado en la base de datos.
  4. Visualmente diferenciar tareas completadas en la lista.

**6 Filtros y Búsqueda de Tareas**

* **Descripción:** Como usuario, quiero buscar y filtrar tareas para encontrar rápidamente la información.
* **Tareas:**
  1. Implementar barra de búsqueda en el frontend.
  2. Implementar filtros por fecha, prioridad y estado.
  3. Implementar endpoints en el backend para búsqueda y filtrado.
  4. Actualizar lista de tareas basada en criterios de búsqueda y filtros.