



1. Requisitos Funcionales

RF1 – Autenticación de usuarios

- El sistema debe permitir registrar nuevos usuarios.
- El sistema debe permitir iniciar sesión usando usuario y contraseña.
- El sistema debe generar un **JWT** válido al iniciar sesión.
- El sistema debe validar el JWT en cada petición protegida.

RF2 – Gestión de tareas

- El usuario debe poder **crear tareas**.
- El usuario debe poder **editar** el título, descripción y estado de una tarea.
- El usuario debe poder **eliminar tareas**.
- El usuario debe ver solo **sus propias tareas**, no las de otros usuarios.
- Las tareas deben guardar:
 - título
 - descripción
 - estado (completado / pendiente)

- fecha de creación
- fecha de completado (si aplica)

RF3 – Interfaz Web (React)

- El usuario debe poder iniciar sesión desde un formulario.
- El usuario debe poder registrarse desde un formulario.
- La aplicación debe ocultar el header cuando está en la página de login/register.
- El usuario debe ver un dashboard después del login.
- El usuario debe poder crear, editar y borrar tareas desde una vista visual.

2. Requisitos No Funcionales

RNF1 – Seguridad

- Todo acceso a las tareas debe estar protegido con **JWT**.
- El token debe guardarse de forma segura en localStorage.

RNF2 – Escalabilidad

- La API debe permitir agregar más controladores en el futuro.
- El frontend debe manejar rutas sin recargar la página (SPA).

RNF3 – Mantenibilidad

- El backend debe seguir el patrón **RESTful**.
- El frontend debe estar dividido en componentes reutilizables.
- El código debe estar organizado por carpetas (controllers, models, dto, routes, etc.).

RNF4 – Rendimiento

- Las consultas a la base de datos deben usar **EF Core** con tracking mínimo cuando sea posible.
- La interfaz debe ser ligera y responsiva usando Tailwind.

RNF5 – Usabilidad

- La interfaz debe ser simple, clara y accesible.
- Las acciones importantes deben tener feedback visual.