**Sistema de Gestión de Tareas Personales (To-Do List)**

**Descripción**

Aplicación web integral para la gestión de tareas personales desarrollada con Angular (Frontend) y .NET Core (Backend). Permite a los usuarios crear, editar, eliminar y organizar sus tareas diarias de manera eficiente.

**Tecnologías Utilizadas**

**Backend**

* **ASP.NET Core 8.0** - Framework web
* **Entity Framework Core** - ORM para base de datos
* **SQL Server LocalDB** - Base de datos
* **LINQ** - Consultas a la base de datos
* **Swagger** - Documentación de API

**Frontend**

* **Angular 18** - Framework de frontend
* **Bootstrap 5** - Framework CSS
* **Font Awesome** - Iconos
* **TypeScript** - Lenguaje de programación
* **RxJS** - Programación reactiva

**Características**

**Funcionalidades Backend (REST API)**

* **GET /api/tareas** - Obtener todas las tareas
* **GET /api/tareas/{id}** - Obtener tarea específica
* **POST /api/tareas** - Crear nueva tarea
* **PUT /api/tareas/{id}** - Actualizar tarea existente
* **DELETE /api/tareas/{id}** - Eliminar tarea

**Consultas LINQ Implementadas**

* Filtrar tareas por estado (completadas/pendientes)
* Ordenar tareas por fecha límite o prioridad
* Búsqueda de tareas por título o descripción

**Funcionalidades Frontend**

* Listado de tareas con filtros y búsqueda
* Formulario para crear nuevas tareas
* Edición de tareas existentes
* Marcar tareas como completadas
* Eliminación de tareas
* Validaciones de formulario
* Interfaz responsiva

**Estructura del Proyecto**

Examen Progra 6

├── Backend/

│ ├── TodoApi/

│ │ ├── Controllers/

│ │ │ └── TareasController.cs

│ │ ├── Data/

│ │ │ └── TodoContext.cs

│ │ ├── Models/

│ │ │ └── Tarea.cs

│ │ ├── Program.cs

│ │ ├── TodoApi.csproj

│ │ └── appsettings.json

│ └── README.md

├── Frontend/

│ ├── src/

│ │ ├── app/

│ │ │ ├── components/

│ │ │ │ ├── task-list/

│ │ │ │ └── tarea-form/

│ │ │ ├── models/

│ │ │ │ └── tarea.model.ts

│ │ │ ├── services/

│ │ │ │ └── tarea.service.ts

│ │ │ ├── app.component.ts

│ │ │ └── app.module.ts

│ │ ├── index.html

│ │ ├── main.ts

│ │ └── styles.css

│ ├── angular.json

│ └── package.json

└── README.md

**Instalación y Configuración**

**Prerrequisitos**

* .NET 8.0 SDK
* Node.js (versión 18 o superior)
* SQL Server LocalDB
* Angular CLI (npm install -g @angular/cli)

**Backend (.NET Core)**

1. **Navegar al directorio del backend:**

cd Backend/TodoApi

1. **Restaurar dependencias:**

dotnet restore

1. **Crear la base de datos:**

dotnet ef database update

1. **Ejecutar la aplicación:**

dotnet run

**Frontend (Angular)**

1. **Navegar al directorio del frontend:**

cd Frontend

1. **Instalar dependencias:**

npm install

1. **Ejecutar la aplicación:**

ng serve

**Uso de la Aplicación**

**Crear una Nueva Tarea**

1. Completa el formulario en el panel izquierdo
2. Ingresa título (obligatorio), descripción (opcional), fecha límite y prioridad
3. Haz clic en "Crear Tarea"

**Gestionar Tareas**

* **Completar tarea:** Marca el checkbox junto a la tarea
* **Editar tarea:** Haz clic en el botón de edición (lápiz)
* **Eliminar tarea:** Haz clic en el botón de eliminación (basura)

**Filtros y Búsqueda**

* **Filtrar por estado:** Selecciona "Todas", "Pendientes" o "Completadas"
* **Ordenar:** Por fecha de creación, fecha límite o prioridad
* **Buscar:** Escribe en el campo de búsqueda para filtrar por título o descripción

**API Endpoint**

**Tareas**

**GET /api/tareas -** Obtener todas las tareas

**Query parameters:** completada, orderBy, buscar

**GET /api/tareas/{id} -** Obtener tarea por ID

**POST /api/tareas -** Crear nueva tarea

**PUT /api/tareas/{id} -** Actualizar tarea

**DELETE /api/tareas/{id} -** Eliminar tarea

**Modelo de Datos**

interface Tarea {

id: number;

titulo: string;

descripcion: string;

fechaLimite: Date;

completada: boolean;

prioridad: number; // 1=Baja, 2=Media, 3=Alta

fechaCreacion: Date;

}

**Desarrollo**

**Comandos Útiles**

**Backend:**

**# Crear migración**

dotnet ef migrations add InitialCreate

**# Actualizar base de datos**

dotnet ef database update

**# Compilar**

dotnet build

**# Ejecutar tests**

dotnet test

**Frontend:**

**# Generar componente**

ng generate component components/nuevo-componente

**# Generar servicio**

ng generate service services/nuevo-servicio

**# Compilar para producción**

ng build --prod

**# Ejecutar tests**

ng test