



TaskBuddy Application

Informe Técnico

1 Objetivos del Proyecto

- Desarrollar una aplicación web fullstack que permita gestionar tareas personales.
- Implementar un sistema seguro de login/registro utilizando **JWT**.
- Garantizar que cada usuario acceda únicamente a sus propios datos.
- Proveer una interfaz moderna y responsive creada con **React + Tailwind**.
- Practicar una arquitectura realista basada en **ASP.NET Core + EF Core + React**.

2 Modelo de Datos

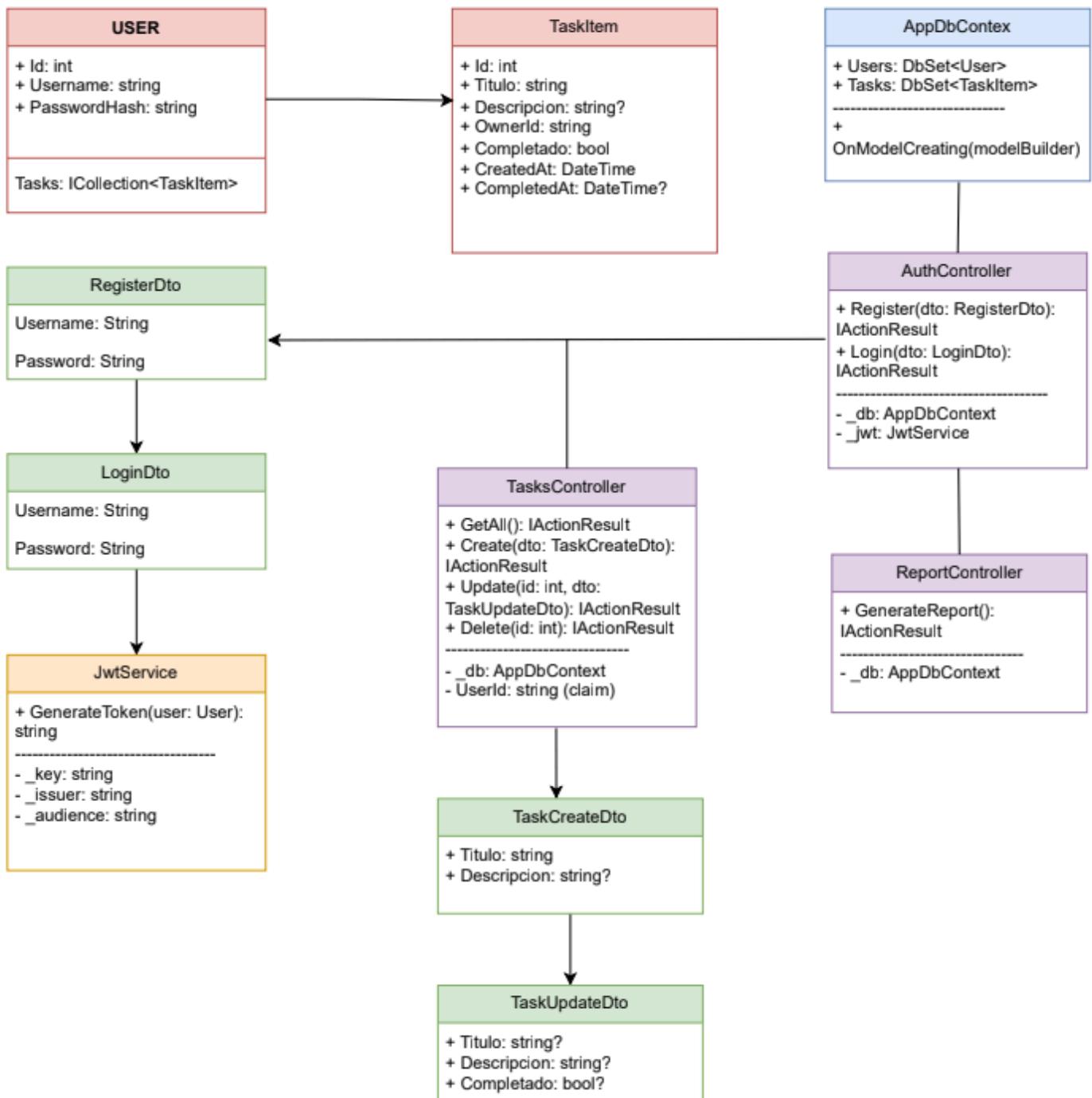
Entidad User

Campo	Tipo	Descripción
Id	string	Identificador único del usuario
Username	string	Nombre de usuario
PasswordHash	string	Hash de la contraseña

Entidad [TaskItem](#)

Campo	Tipo	Descripción
Id	int	Identificador numérico
Titulo	string	Título de la tarea
Descripcion	string	Detalles de la tarea
Completado	bool	Estado de la tarea
CreatedAt	DateTime	Fecha de creación
CompletedAt	DateTime?	Fecha de completado
OwnerId	string	Usuario dueño de la tarea

3 Diagrama de clases



4 Arquitectura

El sistema usa una arquitectura Fullstack con separación clara de responsabilidades:

Frontend (React + Tailwind)



Axios + JWT



Backend (ASP.NET Core Web API)



EF Core



Base de Datos (SQLite)

Base de Datos (SQLite)

Frontend

- React con rutas (react-router-dom)
- Componentes: Login, Register, Dashboard, Tasks, Header
- Axios configurado con token automático

Backend

- ASP.NET Core 8 API
- Controladores:
 - AuthController
 - TasksController
- DTOs
- EF Core con SQLite
- JWT Authentication

Seguridad

- Al iniciar sesión se genera un JWT con:
 - sub: userId
 - exp: fecha de expiración
- Todas las rutas excepto login/register requieren [Authorize].