

DOCUMENTAÇÃO DO SISTEMA TaskFlow API

Luiz Otávio Portes Avelar

Introdução

Este documento apresenta a análise de requisitos, arquitetura e tecnologias utilizadas no desenvolvimento da TaskFlow API, um sistema simples para gerenciamento de tarefas integrado com autenticação JWT.

O objetivo é fornecer uma visão clara sobre como o sistema funciona, quais problemas resolve e como foi projetado.

Objetivo do Sistema

O sistema permite que usuários:

- Criem contas.
- Façam login recebendo um token JWT.
- Gerenciem suas tarefas pessoais.
- Tenham segurança, isolamento de dados e facilidade de uso.

É um sistema backend preparado para aplicações futuras (web, mobile ou desktop).

Público Alvo

- Equipes de desenvolvimento
- Professores e avaliadores
- Estudantes analisando arquitetura de APIs

Escopo do Projeto

O sistema tem:

- API REST
- Autenticação via JWT
- Persistência de dados com Entity Framework Core
- Uso de SQL Server rodando em *container Docker*

Análise de Requisitos

Requisitos Funcionais (RF)

	Descrição
RF01	Criar usuário
	O sistema deve permitir criar usuários com email e senha.
RF02	Login
	O usuário deve conseguir fazer login e receber um token JWT.
RF03	Listar tarefas
	Permitir listar todas as tarefas do usuário autenticado.
RF04	Criar tarefa
	Permitir criar uma nova tarefa associada ao usuário logado.
RF05	Atualizar tarefa
	Permitir editar título, descrição e status.
RF06	Excluir tarefa
	Permitir deletar uma tarefa existente.
RF07	Buscar tarefa por ID
	Permitir buscar uma tarefa específica.
RF08	Garantir isolamento
	O usuário só pode acessar suas próprias tarefas.

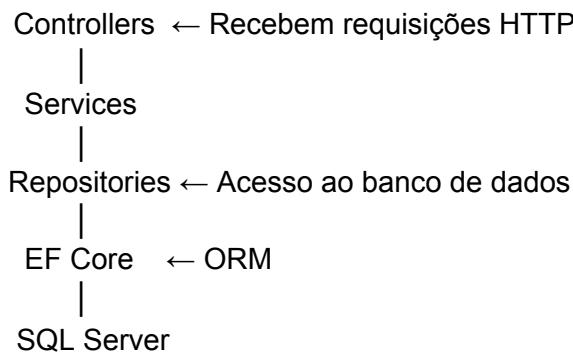
Requisitos Não Funcionais (RNF)

Descrição		
RNF01	Segurança	A autenticação deve usar JWT com chave segura.
RNF02	Integridade	Banco de dados deve armazenar dados de forma consistente.
RNF03	Manutenibilidad e	Código organizado em camadas (Controllers, Repositórios, Models).
RNF04	Desempenho	API deve responder rapidamente a comandos CRUD.
RNF05	Escalabilidade	O sistema deve poder crescer no número de usuários.
RNF06	Confiabilidade	API deve registrar erros adequadamente.
RNF07	Padronização	Endpoints REST obedecem boas práticas de nomes e rotas.
RNF08	Portabilidade	Fácil implantação em outros computadores via Docker.

Arquitetura do Sistema

A arquitetura segue um padrão simples:

Arquitetura em Camadas



Fluxo de Autenticação JWT

1. Usuário envia email + senha
2. API valida credenciais
3. Gera um token JWT contendo o ID do usuário
4. Em cada nova requisição, o token é enviado no header
5. Middleware valida o token

Componentes Principais

Componente	Função
Controller	Recebe e responde às requisições
Repository	Lida com o banco de dados
Models	Representam as entidades
DTOs	Estruturas para entrada/saída
AppDbContext	Conexão com BD via EF Core
Swagger	Documentação automática
JWT Service	Geração e validação de tokens

Tecnologias Utilizadas

Tecnologia	Uso
.NET 8 Web API	Estrutura principal da aplicação
C#	Linguagem
Entity Framework Core	Acesso ao banco
SQL Server (Docker)	Banco de dados
JWT Authentication	Segurança

BCrypt.Net	Hash de senhas
Swagger	Testes e documentação
Git + GitHub	Controle de versão
Docker Desktop	Infraestrutura do SQL

Modelos do Sistema

Entidade Usuário

Id
Nome
Email
SenhaHash
CriadoEm

Entidade Tarefa

Id
Usuarioid
Titulo
Descricao
Status
CriadoEm

Fluxo Geral de Uso

- 1) Criar conta
- 2) Fazer login e pegar token
- 3) Usar token no Swagger
- 4) Criar, listar, editar ou excluir tarefas

Conclusão

A TaskFlow API é uma aplicação simples, escalável e segura, seguindo boas práticas de desenvolvimento de APIs REST.