

# Documentação da API de Gerenciamento de Tarefas

## Visão Geral

A API de Gerenciamento de Tarefas foi desenvolvida utilizando o .NET Core 8.0 e ASP.NET Core. Esta API permite realizar operações de gerenciamento de tarefas, como listar, criar, atualizar e excluir tarefas, além de funcionalidades para gerenciar usuários, incluindo criação de usuários, login e busca de usuário por ID.

Esta documentação oferece uma visão detalhada da API de Gerenciamento de Tarefas, descrevendo os endpoints disponíveis, os requisitos não funcionais e as tecnologias utilizadas.

## Endpoints

A API possui os seguintes endpoints:

### Tarefas

**POST /api/Tasks/CriarTarefa:** Cria uma nova tarefa.

**PUT /api/Tasks/Atualizar/{id}:** Atualiza uma tarefa existente com o ID fornecido.

**DELETE /api/Tasks/Deletar/{id}:** Exclui uma tarefa.

**GET /api/Tasks/ObterPorTitulo/{titulo}:** Retorna uma lista de tarefas com base no título fornecido.

**GET /api/Tasks/ObterPorID/{id}:** Retorna os detalhes de uma tarefa específica com o ID fornecido.

**GET /api/Tasks/ObterTodas:** Retorna todas as tarefas cadastradas.

### Usuários

**POST /api/users/CriarUsuario:** Cria um novo usuário.

**PUT /api/users/login:** Realiza o login do usuário e retorna um token de autenticação JWT.

**GET /api/users/{id}:** Retorna os detalhes de um usuário específico com o ID fornecido.

## Requisitos Não Funcionais

**Segurança:** A API utiliza autenticação JWT (JSON Web Token) para proteger os endpoints e garantir a segurança das operações.

**Desempenho:** A API foi otimizada para garantir um alto desempenho, utilizando cache para consultas frequentes e implementando técnicas de otimização de consulta no banco de dados.

**Escalabilidade:** A arquitetura da API foi projetada para ser facilmente escalável, permitindo o aumento da capacidade de processamento e armazenamento conforme necessário.

**Documentação:** A API está devidamente documentada utilizando Swagger, facilitando a compreensão dos endpoints e operações disponíveis.

## **Tecnologias Utilizadas**

**Plataforma:** .NET Core 8.0

**Framework Web:** ASP.NET Core

**Banco de Dados:** Entity Framework Core com SQLite

**Autenticação:** JWT (JSON Web Token)

**Testes localmente:** Postman

**Documentação Automática:** Swagger

## **Exemplos de Uso:**

### **Criar uma Nova Tarefa**

POST /api/Tasks/CriarTarefa

Content-Type: application/json

```
{  
  "Title": "Nova Tarefa",  
  "Description": "Descrição da Nova Tarefa"  
}
```

### **Atualizar uma Tarefa Existente**

PUT /api/Tasks/Atualizar/{id}

Content-Type: application/json

```
{  
  "Title": "Título Atualizado",  
  "Description": "Nova Descrição da Tarefa"  
}
```

### **Excluir uma tarefa**

DELETE /api/Tasks/Deletar/{id}

### **Obter uma Tarefa por Título**

GET /api/Tasks/ObterPorTitulo/{titulo}

### **Obter uma Tarefa por ID**

GET /api/Tasks/ObterPorID/{id}

### **Obter Todas as Tarefas**

GET /api/Tasks/ObterPorID/{id}

### **Criar um Novo Usuário**

POST /api/users/CriarUsuario

Content-Type: application/json

```
{  
  "FullName": "Lara",  
  "Password": "12345",  
  "Role": "gerente",  
  "Email": "ls@gmail.com"  
}
```

## **Realizar Login do Usuário**

PUT /api/users/login

Content-Type: application/json

```
{  
  "Username": "usuario",  
  "Password": "senha"  
}
```

## **Obtém um usuário pelo ID**

GET /api/users/{id}

## **Como Executar Localmente**

Para executar a API localmente, siga as etapas abaixo:

1. Certifique-se de ter o .NET Core SDK instalado em sua máquina
2. Clone o repositório do projeto
3. Navegue até o diretório do projeto em seu terminal
4. Execute o comando `dotnet run`
5. Acesse a API por meio do navegador ou de um cliente HTTP