**Servidor de tarefas**

A aplicação do servidor de tarefas é uma API feita utilizando .Net 6 no qual possui um controlador responsável pelo CRUD das tarefas, contendo algumas rotas.

**GET**

A computer screen shot of a program code

Description automatically generated

Esse é a rota utilizada para obter todas as tarefas que já foram criadas, ela obtém todas as tarefas no banco de dados um status 200 OK contendo as mesmas, caso algum erro aconteça é retornado um status 500.

**GET com ID**

A computer screen shot of a program code

Description automatically generated

Essa é a rota utilizada para obter uma tarefa com um id específico, ela retorna um 200 OK com a tarefa caso a mesma seja encontrada, se a tarefa não for identificada na base de dados, é retornado um status 404 e por fim caso um erro aconteça será retornado um status 500.

**POST**

A computer screen with white and blue text

Description automatically generated

Rota utilizada para criação de uma tarefa, a mesma é inserida no banco e logo em seguida é retornado um status 200 com a tarefa criada, e caso algum erro aconteça é retornado um status 500.

**POST de concluído**

A screen shot of a computer program

Description automatically generated

Rota utilizada para marcar uma tarefa como concluída ou como incompleta, nela é obtido a tarefa com o id enviado, se a tarefa for identificada no banco e não estiver concluída, a mesma é finalizada e retornado um status 200 com a tarefa atualizada, e vice-versa caso a tarefa já esteja concluída. Caso um erro aconteça é retornado um status 500.

**PUT**

A screen shot of a computer program

Description automatically generated

Rota utilizada para atualizar a descrição e/ou título de uma tarefa, caso a mesma seja identificada no banco de dados, é retornado um status 200 contendo a tarefa atualizada ou status 500 caso um erro aconteça.

**DELETE**

A computer screen with white and green text

Description automatically generated

Rota utilizada para deletar uma tarefa, é recebido o id da tarefa que deve ser deletada caso a mesma seja identificada, é retornado um status 200 caso a mesma seja deletada, um status 500 caso ocorra algum erro e status 404 caso a tarefa não seja identificada.

**Base de dados**

A base de dados utilizada foi o SQLite, o objetivo desse banco de dados ter sido escolhido foi pela facilidade de ter o mesmo em qualquer ambiente, por ser somente um arquivo torna se fácil a configuração e portabilidade dele. Esse banco contém uma única tabela para serem salvos o domínio tarefas, a configuração é feita automaticamente no momento que o servidor de tarefas é iniciado, ele roda o script abaixo:

A computer screen shot of a black screen

Description automatically generated

**Ferramenta ORM (Dapper)**

No servidor de tarefas foi utilizado a biblioteca Dapper, essa ferramenta é um micro ORM (Object Relational Mapping, ou Mapeamento Objeto Relacional em português) feita para mapear as colunas do banco de dados com as propriedades do nosso domínio tarefa, decidimos usar ele pela sua simplicidade de configuração e alta performance para execução de scripts no banco de dados.

**Domínio tarefa**

A screen shot of a computer program

Description automatically generated

Uma das principais partes do servidor de tarefas é o domínio tarefa em si, nele contém toda a lógica feita para registrar quando uma tarefa foi criada, concluída etc. Foi seguido as boas práticas de OO (Orientação a Objetos) deixando com que não seja uma estrutura fácil de ser manipulada externamente, o próprio domínio fornece todas as funcionalidades necessários para se modificar.