

```
Route::get('/Gerenciamento', [GerenciamentoController::class, 'index']);
Route::post('/Gerenciamento', [GerenciamentoController::class, 'store']);
Route::get('/Gerenciamento/{id}', [GerenciamentoController::class, 'show']);
Route::put('/Gerenciamento/{id}', [GerenciamentoController::class, 'update']);
Route::delete('/Gerenciamento/{id}', [GerenciamentoController::class, 'destroy']);
```

- A primeira linha define uma rota GET com o padrão /Gerenciamento que chama o método index do controlador GerenciamentoController.

**Descrição:** Esta rota é responsável por listar todas as tarefas no sistema de gerenciamento. Ela chama o método index do GerenciamentoController, que retornará a visualização ou resposta adequada com as tarefas listadas.

```
"id": 1,
"Titulo": "física",
"Descrição": "MRU",
>Status": "finalizado",
"created_at": "2023-07-03T20:29:53.000000Z",
"updated_at": "2023-07-03T20:43:14.000000Z"

"id": 4,
"Titulo": "lp",
"Descrição": "Álge",
>Status": "finalizado",
"created_at": "2023-07-03T21:01:24.000000Z",
"updated_at": "2023-07-04T19:25:32.000000Z"
```

ilustração da listagem das tabelas

- A segunda linha define uma rota POST com o padrão /Gerenciamento que chama o método store do controlador GerenciamentoController.

**Descrição:** Esta rota é responsável por criar uma nova tarefa no sistema de gerenciamento. Ela chama o método store do GerenciamentoController, que tratará da lógica de criação da tarefa com base nos dados fornecidos.

```

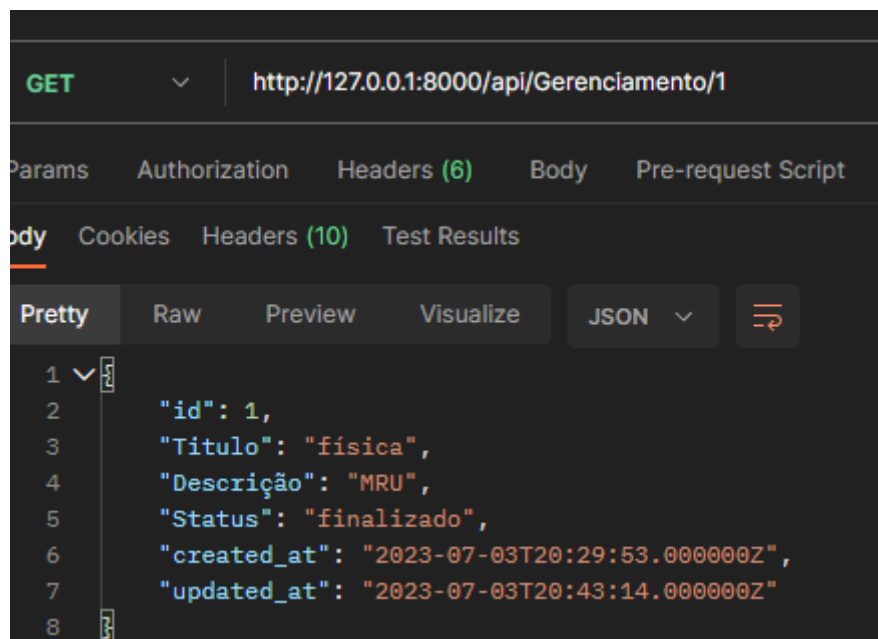
    "Titulo": "jkhtk",
    "Descrição": "Álgeb",
    "Status": "finalizado",
    "updated_at": "2023-07-04T19:23:29.000000Z",
    "created_at": "2023-07-04T19:23:29.000000Z",
    "id": 6

```

ilustração de uma tabela criada

- A terceira linha define uma rota GET com o padrão /Gerenciamento/{id} que chama o método show do controlador GerenciamentoController. O {id} é um parâmetro dinâmico que será substituído por um valor específico.

**Descrição:** Esta rota é responsável por obter os detalhes de uma tarefa específica no sistema de gerenciamento. Ela chama o método show do GerenciamentoController, passando o parâmetro {id} para identificar a tarefa desejada.



The screenshot shows a REST client interface with a GET request to `http://127.0.0.1:8000/api/Gerenciamento/1`. The response is displayed in JSON format, showing details for a task with ID 1.

```

1  {
2    "id": 1,
3    "Titulo": "física",
4    "Descrição": "MRU",
5    "Status": "finalizado",
6    "created_at": "2023-07-03T20:29:53.000000Z",
7    "updated_at": "2023-07-03T20:43:14.000000Z"
8  }

```

- A quarta linha define uma rota PUT com o padrão /Gerenciamento/{id} que chama o método update do controlador GerenciamentoController.

**Descrição:** Esta rota é responsável por atualizar os dados de uma tarefa existente no sistema de gerenciamento. Ela chama o método update

```
"id": 1,  
"Titulo": "lp",  
"Descrição": "Álge",  
"Status": "finalizado",  
"created_at": "2023-07-03T20:59:33.000000Z",  
"updated_at": "2023-07-03T20:59:33.000000Z"
```

do GerenciamentoController, passando o parâmetro {id} para identificar a tarefa que deve ser atualizada.

- A quinta linha define uma rota DELETE com o padrão /Gerenciamento/{id} que chama o método destroy do controlador GerenciamentoController.

**Descrição:** Esta rota é responsável por excluir uma tarefa do sistema de gerenciamento. Ela chama o método destroy do GerenciamentoController, passando o parâmetro {id} para identificar a tarefa que deve ser excluída.

## Criação das tabelas

```
public function up(): void  
{  
    Schema::create('gerenciamentos', function (Blueprint $table) {  
        $table->id();  
        $table->string('Titulo');  
        $table->string('Descrição');  
        $table->string('Status');  
        $table->timestamps();  
    });  
}
```

Neste trecho de código está a ser criada a tabela gerenciamento com as seguintes colunas id, titulo, descrição, status e timestamp.

Schema::create('gerenciamentos', function (Blueprint \$table)

O método create() do Schema é usado para criar uma nova tabela no banco de dados. Nesse caso, a tabela está sendo chamada de "gerenciamentos". O argumento do create() é uma função anônima que recebe um objeto Blueprint como parâmetro. O objeto Blueprint é usado para definir a estrutura da tabela.

`$table->id()`: O método `id()` é usado para criar uma coluna de identificação única (geralmente chamada de "id") na tabela. Essa coluna é geralmente usada como a chave primária da tabela.

`$table->string('Titulo')`: O método `string()` é usado para criar uma coluna de texto (string) chamada "Titulo" na tabela "gerenciamentos". Essa coluna armazenará o título da tarefa.

`$table->string('Descrição')`: O método `string()` é usado novamente para criar uma coluna de texto chamada "Descrição" na tabela. Essa coluna armazenará a descrição da tarefa.

`$table->string('Status')`: Mais uma vez, o método `string()` é usado para criar uma coluna de texto chamada "Status" na tabela. Essa coluna armazenará o status da tarefa (concluída ou não concluída).

`$table->timestamps()`: O método `timestamps()` cria automaticamente duas colunas chamadas "created\_at" e "updated\_at". Essas colunas são usadas para rastrear quando um registro foi criado e quando foi atualizado pela última vez. O Laravel preencherá automaticamente essas colunas com as informações de data e hora correspondentes.