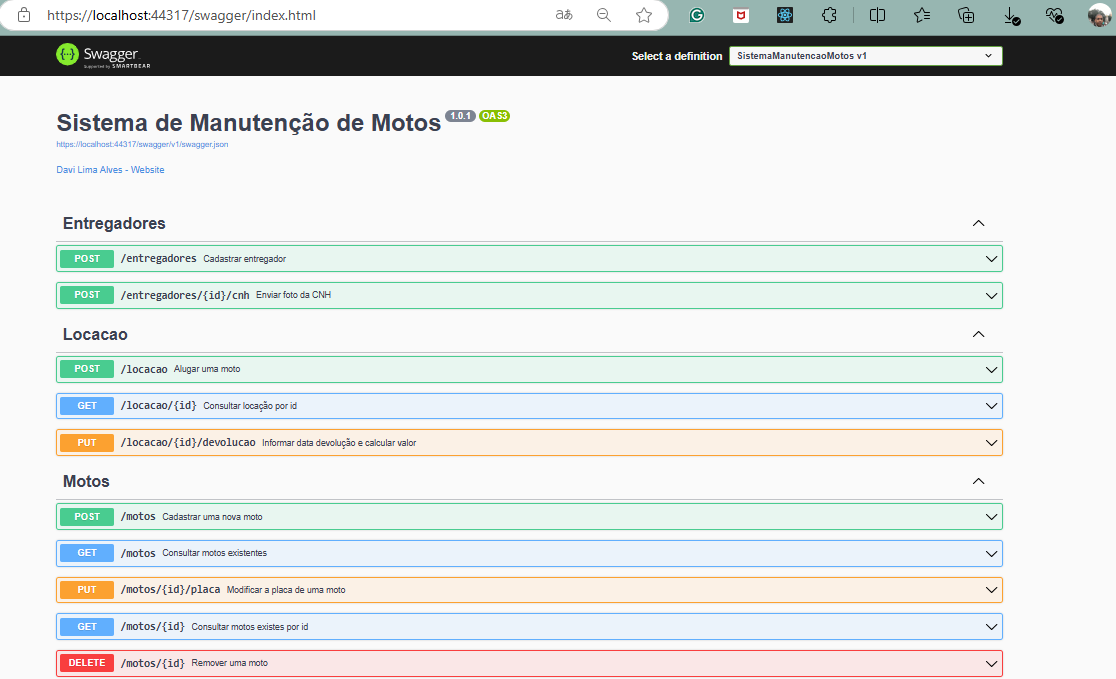
# Sistema de manutenção de motos

## Padrões de projeto e boas práticas

SOLID principles, Clean code, Strategy, Mensageria RabbitMQ, Redis cache e invalidação de cache após atualizar de registro, Postgres, MongoDB, Unit Tests, Integration Tests, Mutation tests, Docker, Controle de acesso via bearer token, Repository, AutoMapper, Entity Framework, SAGA, CQS (futura feature, mas funciona bem com postgers e redis cache no momento)

## Swagger

Descritivo com funcionalidades do web api



## Postman

Disponível na pasta da solução arquivo json

Interface gráfica do usuário, Aplicativo, Email

Descrição gerada automaticamente

### Request token

Na pasta parent de cada controller solicite um novo token para um usuário ser autenticado e permitir realizar requests. O exemplo a seguir mostra a pasta selecionada motos e a solicitação de um token na aba authorization.

user: admin@davinetlive.onmicrosoft.com

password: Mottu2021

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

### Segurança via bearer token

Interface gráfica do usuário, Aplicativo, PowerPoint

Descrição gerada automaticamente

Para autorizar as requests nas pastas entregadores e locação deve ser utilizado o usuário a seguir:

user: user@davinetlive.onmicrosoft.com

password: Mottu2021

## Executar solução

Primeiro passo é colocar os containers dockers do Postgres, Redis, Mongodb e RabbitMQ para executar nas portas utilizadas pela aplicação. Este guia assume que o Docker desktop está instalado no Windows ou o Docker.io no Linux. Seguem os comandos.

### Redis installation in docker (cluster better option)

**docker run --name redis -p 6379:6379 -d redis**

### RabbitMQ installation version latest 4+

**docker run --name rabbitmq -p 5672:5672 -d rabbitmq**

### MongoDB installation in docker (cluster better option) (após instalar pode acessar docker exec -it mongodb mongo)

**docker run --name mongodb -p 27017:27017 -d mongo:latest**

### Postgres installation in docker (cluster better option)

**docker run --name manutencaomoto -p 5432:5432 -e POSTGRES\_USER=moto -e POSTGRES\_PASSWORD=moto -e POSTGRES\_DB=manutencaomoto -d postgres**

#### Criar tabelas no postgres (Executar no cmd na pasta da solução)

**dotnet tool install --global dotnet-ef**

**dotnet ef database update 20241009124749\_initial --project Persistence -s SistemaManutencaoMotos -c Context –verbose**

### Execute solução IIS Express ou outra opção como Docker por exemplo

Ajuste postman environment Variable baseUrl para o endereço e porta da solução em execução.

## Teste de evidências

### Motos

#### Post motos: Interface gráfica do usuário, Texto, Aplicativo, Email Descrição gerada automaticamente

#### Get motos: Tela de computador com texto preto sobre fundo branco Descrição gerada automaticamente

Por placa: Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

#### Get motos/:id

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

#### Put motos/:id/placa

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

#### Delete motos/:id

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

### Entregadores

#### Post entregadores

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

#### Post entregadores/:id/cnh

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

### Locacao

#### Post locacao

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

#### Get locacao/:id

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

#### Put locacao/:id/devolucao

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

## Resultado

Todos endpoints foram executados e confirmados

### Mongo db data

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

### Postgres data

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

### Unit tests, integrations tests, mutation tests e postman for tests

Interface gráfica do usuário, Aplicativo, Word

Descrição gerada automaticamente