

**DOCUMENTAÇÃO  
CASE TÉCNICO ONFLY**

Guilherme Ferreira Costa

# 1. Passo a passo para rodar a aplicação

1. Clonar o repositório do github: <https://github.com/guilhermefcosta/case-tecnico-onfly>
2. Entrar na pasta do projeto  
**\$ cd case-tecnico-onfly**
3. Subir os containers do banco de dados **mysql** e do **phpmyadmin(localhost:8080)** com o comando:  
**\$ docker-compose up -d**  
  
2.1. Caso dê algum erro, confira se já não tem nenhum serviço usando a porta 3306 do mysql ou a porta localhost:8080. Rode este comando se quiser matar todos os containers do docker:  
**\$ docker kill \$(docker ps -q)**
4. Vamos baixar as bibliotecas usadas no projeto com o comando:  
**\$ composer install**
5. Agora, vamos rodar as migrations para criar as nossas tabelas no banco de dados:  
**\$ php artisan migrate**
6. Com tudo pronto, podemos rodar nosso servidor web com o comando  
**\$ php artisan serve**
7. Pronto, se tudo ocorreu bem estamos prontos para prosseguir. Caso dê algum problema de dependências ou extensões do php e não consiga resolver é só me chamar pelo e-mail [guilherme.ferreira@gmail.com](mailto:guilherme.ferreira@gmail.com) ou pelo whatsapp (31) 99533-9541. **Estou rodando com o php8.1. É IMPORTANTE ESTAR COM A EXTENSÃO DO SQLITE BAIXADA E HABILITADA.**

Para baixar a extensão do sqlite caso não esteja baixada:

**\$ sudo apt install php8.1-sqlite3**

## 2. Arquitetura de código do projeto

Explicando de forma sucinta e prática, temos as rotas definidas no arquivo **api.php**. Os verbos http indicam o que cada endpoint faz.

Por exemplo:

**GET -> Buscar entidade**  
**POST -> Criar entidade**  
**PUT -> Atualizar entidade**  
**DELETE -> Deletar entidade**

Separamos a lógica de negócio em Services, para garantir a coesão e o baixo acoplamento. Assim, alguns services podem ser vistos sendo injetados nas funções do projeto.

Os testes estão no diretório **test**, e foram implementados com **PHPUnit**. Em sua totalidade são testes de feature, para testar em sua grande maioria as regras de negócio do case técnico.

Para rodar os testes basta rodar:

**\$ php artisan test**

Se ocorrer algum erro de extensão, basta rodar o comando inserido no tópico 1.

OS demais componentes de código, como migrations, Models, Controllers se mantêm no local padrão do Laravel.

## 3. Implementação

Na implementação, foi utilizado o pacote **sanctum** padrão do Laravel. Ele nos ajuda na tarefa de autenticação em APIs.

Desta forma, temos as rotas iniciais que são as de **login** e **cadastro de usuario**. Nestas duas rotas não é preciso ter o token de autenticação para fazer request. Assim, estão livres para acesso externo.

Você pode rodar o comando **\$ php artisan db:seed** criar alguns registros de teste para iniciar se preferir.

**OBS:** Uma regra implementada que segue os padrões atuais é que depois da criação do usuário, você precisa logar. Sem isso a aplicação não irá funcionar, pois é no login que é pego o token.

**OBS:** O usuário com o campo **role** = 1 é o usuário administrador, já o usuário com **role** = 2 é o usuário comum.

**Para auxiliar na depuração das rotas, vou deixar no projeto um arquivo chamado:**

>>> Onfly.postman\_collection.json

Este arquivo pode ser importado no seu Postman para testar os endpoints com mais agilidade.

## 4. Envio de e-mail

O disparo de e-mail foi implementado com o suporte do **mailtrap**, que é uma ferramenta que recebe e-mails do nosso ambiente local e os exibe como um webmail, muito útil para testes locais.

**Sempre quando houver criação de uma despesa, haverá novos e-mails sendo criados no mailtrap para o usuário dono da despesa e para os admins.**

**CREDENCIAIS PARA ACESSAR O MAILTRAP:**

Site: <https://mailtrap.io/signin>

E-mail: [guifcosta.contato@gmail.com](mailto:guifcosta.contato@gmail.com)

Senha: Ebd1o2f:

Ao chegar na plataforma é só navegar no caminho: **“Email Testing”** >> **“My Inbox”**.

Assim você verá todos os e-mails sendo criados em tempo real.

## 5. Conclusão

Tendo qualquer dúvida em relação ao projeto estou aqui para esclarecer! Muito obrigado pela oportunidade





