

Teste técnico para Pessoa Desenvolvedora Back-end Plena

1. Introdução

2. Requisitos gerais

Para realizar o teste, colocamos alguns requisitos para padronizar o processo de desenvolvimento e entrega para avaliação.

2.1. Usar o Laravel Sail

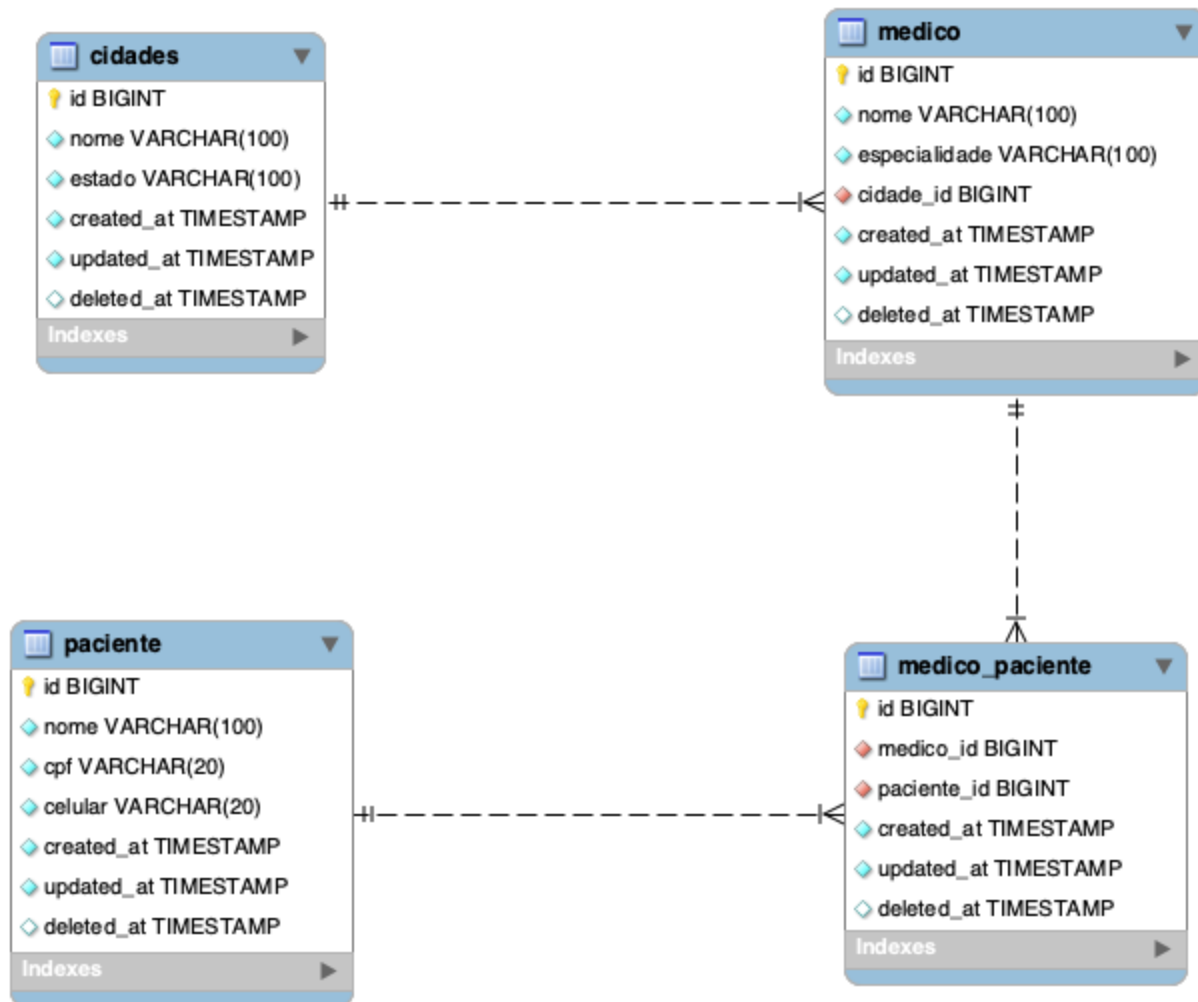
Para executar a aplicação e o banco de dados, pedimos que o projeto seja criado com o Laravel Sail. O único container, além do container da aplicação, que precisa ser instalado é o banco de dados Mysql. As instruções de criação de um projeto com o Laravel Sail estão na documentação.

2.2. Migrations e Seeders

Para criar a estrutura do Banco de dados, com as tabelas necessárias para executar o teste, esperamos que você utilize as Migrations.

Sobre a inserção dos dados, eles devem ser criados através de Factories ou Seeders.

O diagrama abaixo representa as tabelas, colunas e relações que o Banco de dados deve possuir:



2.3. Autenticação com o JWT

Para as rotas que necessitam de autenticação, você deverá utilizar o [JWT](#) como mecanismo de autenticação dos usuários. Considere que a tabela `users` do Laravel para gerenciar os usuários.

2.4. Postman

Um dos critérios da avaliação do teste será a execução dos endpoints no Postman. Para isso, criamos esta [Collection](#). Você pode fazer uma cópia dela e utilizá-la para desenvolver o seu teste.

Na Collection está listado todos os endpoints avaliados, métodos e **exemplos de resultados enviados/retornados**.

3. Aplicação

O teste consiste em realizar o desenvolvimento de uma API para listar, adicionar e atualizar as cidades, médicos e pacientes cadastrados no banco de dados.

Veja a tabela abaixo:

Entidade	Listar	Cadastrar	Atualizar
Cidade	✓	⊘	⊘
Médico	✓	🔒	⊘
Paciente	🔒	🔒	🔒

Legenda:

✓ Público, sem necessidade de autenticação

⊘ Não necessário no projeto

🔒 Privado, apenas usuário autenticado

A parte de listagem das cidades e médicos deverá ser pública. Porém, para cadastrar um novo médico, listar, cadastrar e atualizar um paciente ou vincular um paciente com um médico, só será permitido para os usuários autenticados.

Abaixo está descrito as especificações de cada entidade:

3.1. Cidade

3.1.1. Listar cidades

Para listar as cidades cadastradas, deve ser feito uma requisição **GET** para o *endpoint* `/cidades` e não é necessário estar autenticado.

O retorno esperado é um *Array* de objetos JSON do tipo Cidade.

3.2. Médico

3.2.1. Listar médicos

Para listar todos os médicos cadastrados, deve ser feito uma requisição **GET** para o *endpoint* `/medicos`.

O retorno esperado é um *Array* de objetos JSON do tipo Medico.

3.2.2. Listar médicos de uma cidade

Para listar apenas os médicos de uma cidade específica, deverá ser feito uma requisição **GET** para o *endpoint* `/cidades/{id_cidade}/medicos`.

O retorno esperado é um *Array* de objetos do tipo Medico.

3.2.3. Vincular paciente e médico

Para vincular um paciente e um médico, deverá ser feito uma requisição **POST** para o *endpoint* `/medicos/{id_medico}/pacientes`.

No *body* da requisição deverá conter os parâmetros `medico_id` (com o ID do médico) e `paciente_id` (com o ID do paciente).

Este recurso só deve estar disponível para usuários autenticados.

O retorno esperado é um objeto JSON com os dados do médico e do paciente vinculados.

3.3. Paciente

3.3.1. Listar pacientes do médico

Este é o *endpoint* responsável por retornar todos os pacientes vinculados ao médico. Apenas usuários autenticados tem autorização para acessar esta rota.

Deverá ser feito uma requisição do tipo **GET** para o *endpoint* `/medicos/{id_medico}/pacientes`.

E o retorno esperado é um array de objetos do tipo Paciente.

3.3.2. Atualizar paciente

Este *endpoint* será utilizado para alterar os dados de um paciente específico. Apenas usuários autenticados tem autorização para acessar esta rota.

Somente o nome e o celular podem ser alterados.

Este recurso será acessado através da rota `/pacientes/{id_paciente}` com o método **POST**.

O retorno esperado é os dados do paciente atualizados em formato de objeto JSON.

3.3.3. Adicionar paciente

Este é o *endpoint* responsável por adicionar um novo paciente à base de dados. Apenas usuários autenticados tem autorização para acessar esta rota.

Deverá ser feito uma requisição do tipo `POST` para o *endpoint* `/pacientes`.

O retorno esperado é os dados do novo paciente em formato de objeto JSON.