Prova Técnica

Detalhes de instalação

Ao baixar o arquivo, acesse o local do projeto no terminal e use comando abaixo:

composer install (Para instalar todas as dependências)

É necessário criar o banco de dados manualmente para que as tabelas sejam geradas pelas migrations já configuradas. Após a criação do banco, é importante configurar o arquivo .env ou o Database.php com as informações de conexão do banco de dados. Por fim, basta executar o comando abaixo para que as tabelas sejam criadas automaticamente.

Php spark migrate

E por fim, para iniciar o projeto, usar o comando:

Php spark serve

A url base do projeto é **localhost:8080**

Detalhes sobre o banco de dados

No campo status da tabela pedidos_compra, os seguintes valores numéricos representam os respectivos status do pedido:

0 = Cancelado

1 = Em aberto

2 = Pago

Os campos id_cliente e id_produto na tabela pedidos_compra correspondem aos IDs das tabelas clientes e produtos, respectivamente.

Itens implementados

Desafio 1 completo, *endpoints* de clientes, produtos e pedidos de compra, todos com CRUD completo;

Desafio 2, foram implementados paginação e filtros nos endpoints de listagem /pedidos, /clientes e /produtos para os seguintes campos:

Clientes: nome e razão_social;

Produtos: nome;

Pedidos: nome (nome do cliente associado ao pedido) e status.

A paginação é controlada por dois parâmetros:

- pagina: indica a página atual.

- porPagina: define a quantidade de registros por página.

Todos os filtros e parâmetros de paginação são passados via query string na URL. Exemplo de requisição:

Clientes

Para cadastro do cliente via POST ou atualização do mesmo via PUT, o único campo obrigatório é o nome. No caso de atualizar o cadastro, não é obrigatório atualizar o nome junto, mas não é permitido anular o mesmo, veja os campos abaixo:

```
{
    “parametros”: {
        “nome”: “Vinicius”,
        “cpf”: “55500088877”,
        “razão_social”: “L5 networks”,
        “cnpj”: “12345678912345”
    }
}
```

(GET) | /clientes => Retorna todos os clientes cadastrados.
(GET) | /clientes/1 => Retorna os dados do cliente.
(POST) | /clientes => Cadastra os dados de um cliente.
(PUT) | /clientes/1 => Atualiza os dados de um cliente.
(DELETE) | /clientes/1 => Deleta os dados de um cliente.

Produtos

Para cadastro do produto, o único campo obrigatório é o nome. O campo pode ser adicionado da seguinte forma:

```
{
    “parametros”: {
        “nome”: “Vinicius”
    }
}
```

(GET) | /produtos => Retorna todos os produtos cadastrados.
(GET) | /produtos/1 => Retorna os dados do produto.
(POST) | /produtos => Cadastra os dados de um produto.
(PUT) | /produtos/1 => Atualiza os dados de um produto.
(DELETE) | /produtos/1 => Deleta os dados de um produto.

Pedidos de Compra

Para cadastro do pedido de compra, o campos obrigatórios são os ids do cliente e produto, e precisam existir em suas respectivas tabelas. o status pode ser informado, mas caso não seja, o padrão será 1 => Em aberto, e no final fica assim:

```
{
	“parametros”: {
		“id_cliente”: 1,
		“id_produto”: 4,
		“status”: 2
	}
}
```

(GET) | /pedidos => Retorna todos os pedidos de compra cadastrados.
(GET) | /pedidos/1 => Retorna os dados de um pedido.
(POST) | /pedidos => Cadastra um novo pedido.
(PUT) | /pedidos/1 => Atualiza os dados de um pedido.
(DELETE) | /pedidos/1 => Deleta os dados de um pedido.

Lembrando que esses parâmetros servem tanto para POST quanto para PUT.