# Projeto Teste Liven

## Sumário

1. Descrição do Projeto

2. Requisitos

3. Instalação

4. Configuração

5. Endpoints da API

6. Testes

7. Documentação da API

8. Contribuição

9. Licença

## Descrição do Projeto

O projeto Teste Liven é uma API desenvolvida em PHP que permite a manipulação e consulta de dados de usuários e seus endereços. A API fornece endpoints para recuperar endereços com base no país ou no ID do endereço, bem como recuperar endereços associados a um determinado usuário.

## Requisitos

- PHP 7.4 ou superior

- Composer

- MySQL

- Servidor Web (Apache, Nginx, etc.)

## Instalação

1. Clone o repositório:

git clone https://github.com/seu-usuario/liven.git

2. Instale as dependências do Composer:

composer install

3. Configure o banco de dados e execute as migrações necessárias.

Banco se encontra no arquivo banco.sql na raíz do projeto, ou execute o código abaixo:

CREATE DATABASE IF NOT EXISTS `sistema\_endereco`  
DROP TABLE IF EXISTS `enderecos`;  
  
CREATE TABLE `enderecos` (  
 `id` int(11) NOT NULL AUTO\_INCREMENT,  
 `logradouro` varchar(255) NOT NULL,  
 `numero` varchar(10) DEFAULT NULL,  
 `complemento` varchar(100) DEFAULT NULL,  
 `bairro` varchar(100) DEFAULT NULL,  
 `cidade` varchar(100) DEFAULT NULL,  
 `estado` varchar(50) DEFAULT NULL,  
 `pais` varchar(50) DEFAULT NULL,  
 `cep` varchar(20) DEFAULT NULL,  
 `usuario\_id` int(11) NOT NULL,  
 `created\_at` timestamp NOT NULL DEFAULT current\_timestamp(),  
 `updated\_at` timestamp NOT NULL DEFAULT current\_timestamp() ON UPDATE current\_timestamp(),  
 PRIMARY KEY (`id`),  
 KEY `usuario\_id` (`usuario\_id`),  
 CONSTRAINT `enderecos\_ibfk\_1` FOREIGN KEY (`usuario\_id`) REFERENCES `usuarios` (`id`) ON DELETE CASCADE  
) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;  
  
/\*Table structure for table `usuarios` \*/  
  
DROP TABLE IF EXISTS `usuarios`;  
  
CREATE TABLE `usuarios` (  
 `id` int(11) NOT NULL AUTO\_INCREMENT,  
 `nome` varchar(100) NOT NULL,  
 `email` varchar(100) NOT NULL,  
 `senha` varchar(255) NOT NULL,  
 `created\_at` timestamp NOT NULL DEFAULT current\_timestamp(),  
 `updated\_at` timestamp NOT NULL DEFAULT current\_timestamp() ON UPDATE current\_timestamp(),  
 PRIMARY KEY (`id`),  
 UNIQUE KEY `email` (`email`)  
) ENGINE=InnoDB AUTO\_INCREMENT=8 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

## Configuração

1. No arquvio conig/config.php configure a variável $chave\_secreta para receber sua chave para o token (Recomendado salvar uma variável de ambiente)

2. Verifique se o arquivo `conexao.php` está configurado corretamente para conectar-se ao seu banco de dados.

3. Configure o servidor web para apontar para o diretório do projeto.

## Endpoints da API

### Recuperar Endereços por País

\*\*URL:\*\* `/endereco/getendereco.php?country={country\_code}`

\*\*Método:\*\* `GET`

\*\*Parâmetros:\*\*

- `country` (opcional): Código do país para filtrar os endereços.

\*\*Exemplo de Requisição:\*\*

```sh  
curl -X GET "http://localhost/liven/endereco/getendereco.php?country=BR"  
```

\*\*Resposta:\*\*

- \*\*200 OK\*\*:  
 ```json  
 {  
 "enderecos": [  
 {  
 "id": 1,  
 "usuario\_id": 1,  
 "pais": "BR",  
 "endereco": "Rua A, 123"  
 },  
 ...  
 ]  
 }  
 ```

- \*\*404 Not Found\*\*:  
 ```json  
 {  
 "erro": "Nenhum endereço encontrado para o país especificado"  
 }  
 ```

### Recuperar Endereço por ID

\*\*URL:\*\* `/endereco/getendereco.php?id={endereco\_id}`

\*\*Método:\*\* `GET`

\*\*Parâmetros:\*\*

- `id` (obrigatório): ID do endereço.

\*\*Exemplo de Requisição:\*\*

```sh  
curl -X GET "http://localhost/liven/endereco/getendereco.php?id=1"  
```

\*\*Resposta:\*\*

- \*\*200 OK\*\*:  
 ```json  
 {  
 "id": 1,  
 "usuario\_id": 1,  
 "pais": "BR",  
 "endereco": "Rua A, 123"  
 }  
 ```

- \*\*404 Not Found\*\*:  
 ```json  
 {  
 "erro": "Endereço não encontrado para o ID especificado"  
 }  
 ```

### Recuperar Endereços por ID de Usuário

\*\*URL:\*\* `/usuario/getusuario.php?usuario\_id={usuario\_id}`

\*\*Método:\*\* `GET`

\*\*Parâmetros:\*\*

- `usuario\_id` (obrigatório): ID do usuário.

\*\*Exemplo de Requisição:\*\*

```sh  
curl -X GET "http://localhost/liven/usuario/getusuario.php?usuario\_id=1"  
```

\*\*Resposta:\*\*

- \*\*200 OK\*\*:  
 ```json  
 {  
 "enderecos": [  
 {  
 "id": 1,  
 "usuario\_id": 1,  
 "pais": "BR",  
 "endereco": "Rua A, 123"  
 },  
 ...  
 ]  
 }  
 ```

- \*\*404 Not Found\*\*:  
 ```json  
 {  
 "erro": "Nenhum endereço encontrado para o usuário especificado"  
 }  
 ```

## Testes

Os testes são realizados usando PHPUnit. Para executar os testes, utilize o seguinte comando:

vendor/bin/phpunit

### Testes de Endpoints

\*\*EnderecoTest.php\*\*  
Testa os endpoints relacionados aos endereços:

- `testGetEnderecosPorPais()`: Testa a obtenção de endereços por país.

- `testGetEnderecoPorId()`: Testa a obtenção de um endereço por ID.

### UsuarioTest.php

Testa os endpoints relacionados aos usuários:

- `testGetEnderecosPorUsuarioId()`: Testa a obtenção de endereços por ID de usuário.

## Documentação da API – implementação incompleta

A documentação da API é gerada usando Swagger. Para acessar a documentação, abra o Swagger UI e aponte para o arquivo `swagger.php`:

```sh  
http://localhost/swagger/dist/  
```

Certifique-se de que o arquivo `swagger.php` esteja configurado corretamente para gerar a documentação da API. {Essa parte ficou incomplete pois não consegui finalizar}