



## Desafio Programador Full Stack PHP/Angular

### Questão 1: CRUD gerenciamento de empresas

Implemente uma API REST para realizar o gerenciamento de funcionários e clientes de um conjunto de empresas. Neste cenário, cada funcionário ou cliente pode pertencer a uma ou mais empresas, assim como cada empresa pode possuir um ou mais funcionários/clientes. As operações básicas de um CRUD deverão ser implementadas com os verbos HTTP e códigos de retorno HTTP apropriados, tanto para funcionários e clientes, como para empresas.

#### Requisitos:

- Ao exibir os dados de uma empresa deverão ser exibidos os funcionários e clientes vinculados a mesma;
- Ao exibir os dados de um funcionário deverão ser mostradas as empresas as quais ele está associado;
- Ao exibir os dados de um cliente deverão ser mostradas as empresas as quais ele está associado;
- Os campos de cada objeto funcionário são: login, nome, cpf, e-mail, endereço e senha.
- Os campos de cada objeto cliente são: login, nome, cpf, e-mail, endereço e senha.
- Os campos de cada objeto empresa são: nome, cnpj e endereço.
- Deverá ser possível enviar um documento de identificação para cada funcionário ou cliente cadastrado no sistema.
- Os documentos devem estar em formato pdf ou jpg e deverão fazer parte dos dados do funcionário/cliente.
- Procure realizar um tratamento de erros adequado.

### Questão 2: CRUD gerenciamento de empresas

Conecte o serviço desenvolvido na questão 1 a uma aplicação front-end desenvolvida com o framework Angular 17 ou superior. Você deve:

- Criar o CRUD de serviços.
- Tratar os erros de forma genérica (preparando-os para mais serviços).
- Impedir o cadastro quando o campo login ou nome possuir acentuação.

#### Informações Gerais:

- É fácil encontrar um código de exemplo pronto para este fim na Internet. Você pode se basear nestes códigos, mas espera-se que o sistema final entregue seja robusto e estável, o que normalmente não é o caso de códigos de exemplo. Utilize seu conhecimento e os métodos adequados para garantir/provar esta robustez e estabilidade;
- Comentários em código são bem vindos;
- A linguagem utilizada deverá ser php 7+, preferencialmente usando o framework Laravel 10+.
- Para a banco de dados você pode escolher entre Postgres e MySql
- Para o servidor WEB recomendamos Apache HTTP 2.2 ou superior.
- Para consumir o serviço REST você deve utilizar o Postman e exportar suas requisições em formato .json, enviando o arquivo junto com o projeto.

## Entrega:

- Criar uma imagem docker (<https://laravel.com/docs/12.x>) contendo o projeto completo e suas dependências, publicar no GIT ou fazer deploy em algum servidor e disponibilizar o link para acesso.

## Endereços Úteis:

- <https://laravel.com/>
- <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods>
- <https://angular.io/>
- <http://www.php.net/>
- <https://httpd.apache.org/>
- <https://www.postgresql.org/>
- <https://www.getpostman.com/>
- <https://www.docker.com/>
- <https://laravel.com/docs/12.x>

© BRy Tecnologia.

Todos os direitos reservados. Proibida distribuição, comercialização e cópia.