

### Instruções para execução dos testes

- Todos os códigos e respostas devem estar em um repositório privado no git.
- Este teste é confidencial portanto não deve ser compartilhado, ou de qualquer forma disponibilizado através de qualquer meio, sobe pena de desqualificação do candidato
- O teste deve seguir especificamente as instruções de branches e commits descritas neste documento.
- Não deve ser utilizado respostas prontas da Internet, entendemos que todos os conceitos aqui aplicados devem ser explicados com as palavras do aplicante.
- Compartilhar seu repositório com o usuário no github: **doutbox**

### Regras de branches e commits para desafios práticos

- Cada tarefa disposta no teste deve estar em uma branch separada utilizando a seguinte nomenclatura:
  - nome-do-candidato-framework-numerodaquestao
    - nome-do-candidato: Seu nome separado por hifens
    - framework: laravel ou flutter
    - numerodaquestao: número da questão substituindo pontos por hífen
- O commit deve ser descrito conforme a recomendação:
  - DTB-numerodaquestao - Descrição do commits
    - numerodaquestao: número da questão substituindo pontos por hífen

## 2 Laravel

### 2.1 Perguntas Técnicas

2.1.1 O que são as rotas?

2.1.2 O que são os controllers?

2.1.3 O que são os models?

2.1.4 O que é MVC?

2.1.5 Como podemos gerar rotas automática com Laravel?

2.1.6 Explique o que é o Blade e quais as principais diretivas (cite no mínimo 3).

2.1.7 Explique detalhadamente o que é um guard.

2.1.8 Qual a principal função do composer em um projeto Laravel?

2.1.9 Qual a principal função do Artisan em um projeto Laravel?

2.1.10 Como funciona o Auth do Laravel?

## 2.2 Desafios Práticos

### 2.2.1 DoutBlog

#### Descrição do problema

Um cliente Doutbox requisitou a construção de uma plataforma web capaz de postar notícias e artigos, permitindo que usuários externos também façam postagens, este cliente precisa ainda de um sistema de permissões simples.

#### Desafio

Desenvolver uma aplicação web de blog em Laravel com as seguintes características:

- Todos usuários podem fazer login e se registrar
- Dois níveis de acesso para usuários:
  - 'admin' : tem acesso total ao site e pode ler / escrever / atualizar / excluir qualquer uma das postagens / tornar outros usuários administradores / excluir usuários;
  - 'author' : podem escrever / atualizar / excluir suas próprias postagens.
- Qualquer um pode ler todas as postagens
- As postagens possuem dois status:
  - 'published' (1) : Publicas, devem estar expostas para todos os usuários;
  - 'draft' (0) : Rascunho, devem estar disponíveis apenas para o usuário que criou esta publicação.
- Por padrão toda publicação deve ser criada como 'draft'
- Os usuários podem comentar as postagens (somente após o login)
- Deve-ser utilizado apenas Laravel + Bootstrap + Javascript (no máximo JQuery)

## 2.2.2 Zoológico

### Descrição do problema

Um Zoológico está iniciando sua operação na cidade de Tubarão e está em busca de um sistema para gerenciar os cadastro de animais e registrar as atividades de cada um deles. Você foi um dos selecionados para construir um protótipo de aplicação web que atenda as necessidades do Zoológico.

Inicialmente o zoológico deve permitir que cada animal realize as seguintes ações: comer, passear e dormir. Os animais devem possuir os seguintes atributos: ID, Nome, Espécie e Idade. Para fins de validação no final da execução deve ser exibido no sistema todos os animais cadastrados (e seus atributos), cada animal deve realizar pelo menos duas ações e indicar a situação do animal de acordo com sua idade (0-2 = 'filhote', 2-8 = 'jovem', > 8 = 'idoso'). Por se tratar de um protótipo não é necessário construir conexões com banco de dados.

cachorra, Vampira está comendo  
cachorra, Vampira está passeando  
cachorra, Vampira está dormindo  
Sua idade é: 2 e seu estado é: filhote

girafa, Lole está comendo  
girafa, Lole está passeando  
girafa, Lole está dormindo  
Sua idade é: 5 e seu estado é: jovem

elefante, Buda está comendo  
elefante, Buda está passeando  
elefante, Buda está dormindo  
Sua idade é: 10 e seu estado é: idoso

## 2.2.2 Rotação de Array

### Descrição do problema

Dado um vetor de números inteiros e um número de rotações, execute as rotações à esquerda no vetor. Retorne o vetor atualizada a ser impressa como uma única linha de números inteiros separados por espaço.

Exemplo: Se temos um vetor [1, 2, 3, 4, 5], ele deve ficar [5, 1, 2, 3, 4]. Supondo que o número de rotações seja 4 o resultado esperado é: [1, 2, 3, 4, 5] -> [2, 3, 4, 5, 1] -> [3, 4, 5, 1, 2] -> [4, 5, 1, 2, 3] -> [5, 1, 2, 3, 4]