

# Desafio da **Maximize**

## Perfil Backend Challenge

O problema que você deve resolver é a entrega de um feed de notícias para uma aplicação com o uso de API, utilizando ferramentas open-source. **Como ferramentas para esse desafio, você deve usar a linguagem PHP e framework Laravel ou Lumen.**

Neste documento você encontrará algumas telas com requisitos visuais estabelecidos para o projeto. Você também encontrará contratos que deverão ser respeitados pelo seu front-end para a busca dos dados de notícias. Esta lista é apresentada para o usuário numa página, onde é possível clicar em cada um dos itens para acessar a matéria apresentada.

A entrega deve ser feita por um **repositório público** criado por você, em até **7 dias** após o recebimento deste teste, e com **instruções claras** de como rodar o projeto.

## Regras de negócio

Cada item matéria deve ser exibido no feed com um título, uma descrição e uma imagem de capa opcional. Ao clicar em qualquer lugar do item, o usuário deverá ser redirecionado para a página da matéria.

## API

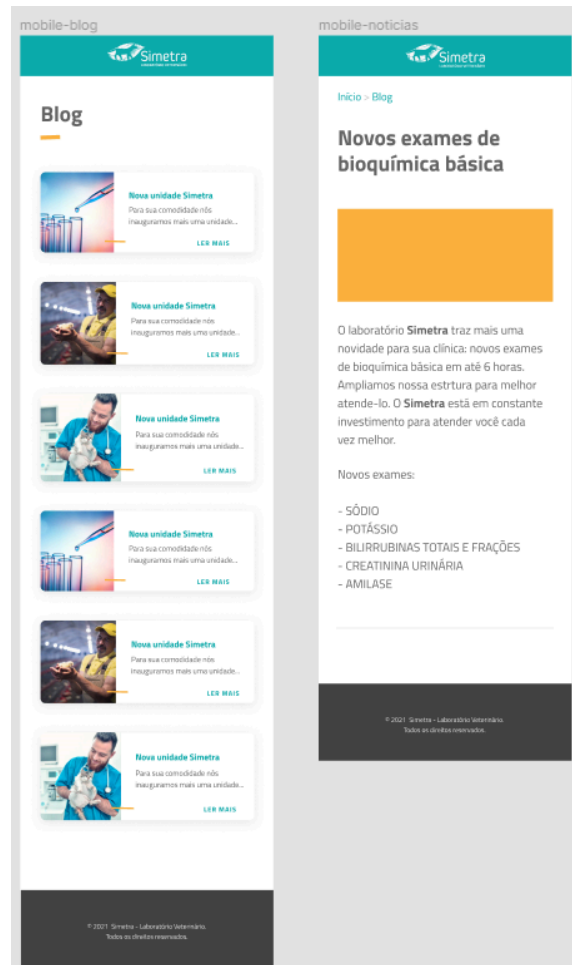
A API deve possuir dois endpoints:

1. O primeiro deve entregar uma lista de matérias
  - A lista deve conter paginação;
  - Cada matéria deve conter os campos:
    - 'titulo'
    - 'descricao'
    - 'imagem'
    - 'data\_de\_publicacao'
2. O segundo deve entregar a matéria completa
  - Onde a matéria deve conter os campos:
    - 'titulo'
    - 'descricao'
    - 'texto\_completo'
    - 'imagem'
    - 'data\_de\_publicacao'

## Premissas

O [link](#) é um guia para o mesmo layout abaixo, mas você pode fazer alterações de acordo com o seu senso estético.

**Atenção:** note que o rótulo das telas contém a palavra “mobile” mas a expectativa é de desenvolvimento de uma Aplicação Web. O desenvolvimento do frontend será considerado, mas lembre-se que o teste é de **BACKEND**. Nada de perder o foco, em?



## O que será analisado?

Seu código será observado por desenvolvedores que avaliarão: **a implementação do código, simplicidade e clareza da solução, a arquitetura, estilo de código e a documentação.**

A automação da infraestrutura também é importante. O ambiente deve ser containerizado para permitir a configuração e teste pelos nossos desenvolvedores.

## Dicas

1. Use ferramentas e bibliotecas open-source, mas documente as decisões de uso delas e os porquês;
2. Automatize o máximo possível: lints, hints, formatadores de código são excelentes ferramentas para alcançar seus resultados ;
3. Em caso de dúvidas, não hesite em perguntar;
4. Registre tudo no sistema de versionamento e na documentação::
  - a. Testes que foram executados;
  - b. Ideias que gostaria de implementar se tivesse tempo (explique como você as resolveria caso houvesse mais tempo);
  - c. Decisões que foram tomadas e seus porquês;
  - d. Arquiteturas que foram testadas e os motivos de terem sido modificadas ou abandonadas;
  - e. Crie um arquivo HISTORY.md no repositório para registrar essas reflexões e decisões;
  - f. Quanto mais detalhes sobre a jornada, melhor!

## Observações finais

Caso não consiga terminar o projeto, fique tranquilo, avaliamos somente os pontos concluídos. Qualquer dúvida é só entrar em contato!

**Boa sorte!**

