

Desafio iOS Stone

Este é o desafio para posição de dev iOS no time da Stone. Muito obrigado pelo interesse em fazer parte da nossa equipe, estamos ansiosos para ver o resultado do seu desafio!

Submissão

- O repositório do desafio deve ser privado, por favor não compartilhe com ninguém.
- O prazo limite é 23/12/22 às 23h59
- Formato de entrega: você deve enviar um arquivo zip ou compartilhar o link do código do Drive/Dropbox com a gente.

Desafio

O desafio consiste em desenvolver um aplicativo iOS em Swift (ou Objective-C). Deve-se consumir uma API pública do Rick & Morty: <https://rickandmortyapi.com>. Sinta-se à vontade para escolher entre REST e GraphQL.

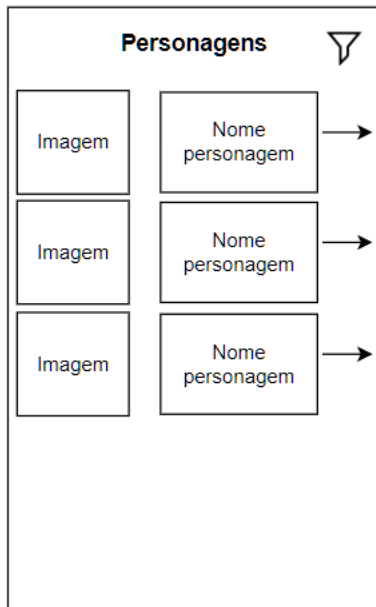
Você não precisa começar o projeto do zero, você pode utilizar o seguinte template: <https://github.com/stone-co/template-desafio-ios>. Fique à vontade para modificar a estrutura de módulos ou o que julgar necessário.

⚠ As imagens abaixo não devem ser utilizadas como referência para o seu layout, sinta-se livre e utilize sua criatividade.

Parte 1: Personagens

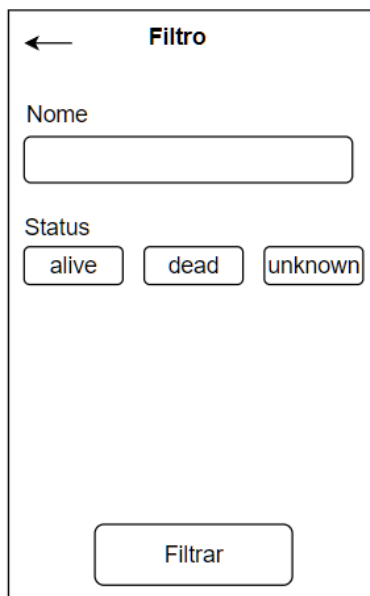
1.1 Listagem de personagens

Utilize o endpoint [/character](#) para montar uma lista de personagens. Cada item da lista deve exibir a foto e o nome dos personagens.



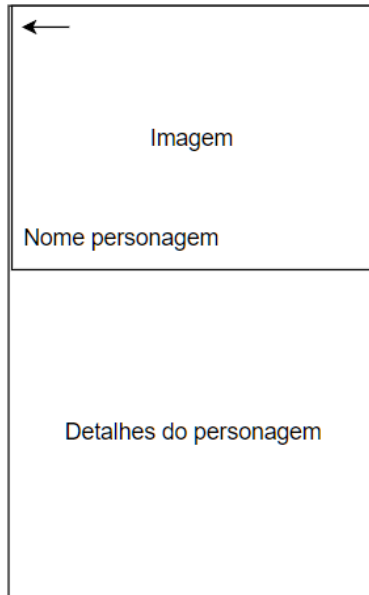
1.2 Filtros de personagem

Utilize o mesmo endpoint de listagem (agora passando [parâmetros](#)) para filtrar os personagens por *name* e *status*.



Parte 2: Detalhes do personagem

Ao clicar em um personagem, navegue para tela de detalhes e exiba informações adicionais como *species*, *gender* e *type*.



O que será avaliado

- Uso obrigatório da linguagem de programação Swift ou Objective-C;
- Implementação dos requisitos obrigatórios;
- Para construção da interface **é necessário** o uso do UIKit (que faz parte do nosso dia-a-dia). O uso de ViewCode é opcional, porém contará pontos;
- Testabilidade;
- Legibilidade;
- Manutenibilidade;

Requisitos

Obrigatório

- Tela de listagem de personagens (com paginação)
- Tela de detalhes do personagem
- Filtro de personagens

Opcional

- Escrever testes unitários e de UI
- Suporte ao modo offline (cachear localmente response da API)
- Suporte a rotação de tela (orientações *portrait* e *landscape*)

Boas práticas

- README explicando as tomadas de decisões do projeto (dedique um tempo aqui);
- Respeite os princípios de SOLID e Clean Code;
- Uso de Dependency Injection (assistida ou não por frameworks);
- Uso de uma arquitetura de responsabilidades bem definida (como MVVM ou MVP);
- Entregue uma interface otimizada, que faz uso das técnicas mais modernas relacionadas aos toolkits de UI nativos.

Resultado

A avaliação do desafio vai ser feita num período máximo de 7 dias.

Mesmo em caso de reprovação nós enviaremos um feedback mostrando oportunidades de melhoria e de crescimento.