

[vivadecora / ios-teste](#)[Watch 7](#)[Star 0](#)[Fork 1](#)[Code](#)[Issues 0](#)[Pull requests 0](#)[Projects 0](#)[Pulse](#)[Graphs](#)

Branch: master ▾

[ios-teste / README.md](#)[Find file](#)[Copy path](#)

taciogt atualiza os critérios bônus e adiciona critérios de avaliação

6ada54e a day ago

1 contributor

82 lines (52 sloc) | 3.43 KB

[Raw](#)[Blame](#)[History](#)

Teste para candidatos à vaga de desenvolvedor iOS

Teste para candidatos à vaga de desenvolvedor iOS no Viva Decora



Esse teste é público. Candidatos para o teste devem implementar a aplicação solicitada e criar uma issue com um link para um repositório com a solução do mesmo.

Objetivo do teste

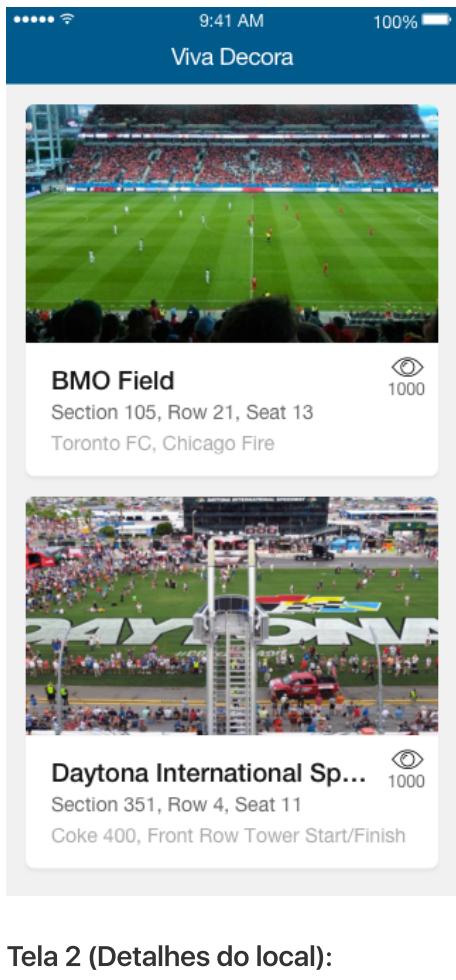
Implementar em Objective-C um aplicativo nativo para iOS que consulte uma API e mostre uma lista de locais (com foto) e uma tela de detalhes.

Requisitos

- Seguir as diretrizes do iOS Human Interface Guidelines
- Fazer cache das imagens
- Se não houver internet, exibir mensagem com botão para tentar novamente
- Tela responsiva
- Touch feedback

Tela 1 (Lista de locais):

Deve ser feita uma lista, em que cada elemento tenha a imagem, o nome do local("venue"), a descrição("note") quando houver e o número de visualizações, conforme imagem abaixo:



Tela 2 (Detalhes do local):

Deve ser feita uma tela de detalhes, que abre a partir do clique em um item da lista da Tela 1. Deve ser exibido a imagem, o nome ("name"), o endereço ("address", "city", "state", "country") e a média de nota ("average_rating"). Quando existir, deve mostrar o conteúdo "stats" e o "link".

This screenshot shows the details page for the BMO Field event. At the top, there is a back button labeled "Voltar" and the name "BMO Field". Below the header is a large thumbnail image of the stadium. The event details are listed below the image:

BMO Field ★
4,4
Section 105, Row 21, Seat 13
Toronto FC, Chicago Fire

Capacity: 100.000
Construction Cost: \$ 27 milion
Opened: 1967
Surface: Grass

[ACESSAR O SITE](#)

Objetivos Bônus:

- Implementação de testes com relatório de cobertura;
- Compartilhamento quanto tiver link de web (na tela de detalhes);
- Sinta-se à vontade para fazer seu bônus.

API:

Essa API permite diversos tipos de consultas. Ela é baseada em fotos postadas por usuários. Eles postam fotos da visão de seus assentos em estádios, concertos e teatros. Para utilizá-la é necessário um cadastro que gera uma chave. Já fizemos o cadastro e já estamos fornecendo a chave direto na url.

A url para obter os dados da Tela 1 (Lista):

```
https://aviewfrommyseat.com/avf/api/featured.php?appkey=f6bcd8e8bb853890f4fb2be8ce0418fa
```

A url para obter a imagem de cada item da lista da Tela 1 é obtida concatenando a url abaixo mais o campo "image" que retorna no json:

```
http://aviewfrommyseat.com/wallpaper/<image>
```

A url para obter os dados da Tela 2 (Detalhes) é obtida concatenando o nome da venue na url no lugar de ":

```
https://aviewfrommyseat.com/avf/api/venue.php?appkey=f6bcd8e8bb853890f4fb2be8ce0418fa&venue=<nome da venue>&info=true
```

Exemplo: <https://aviewfrommyseat.com/avf/api/venue.php?appkey=f6bcd8e8bb853890f4fb2be8ce0418fa&venue=Arthur+Ashe+Stadium&info=true>

A url para obter a imagem da Tela 2 (Detalhes) é obtida concatenando a string do campo "newest_image" na url:

```
http://aviewfrommyseat.com/photos/<newest_image>
```

Documentação completa:

```
http://developer.aviewfrommyseat.com/docs.php#api_featured
```

Critérios de avaliação:

- Organização do código: desacoplamento e legibilidade contam;
- Flexibilidade do sistema para adição/remoção de funcionalidades;

Como vamos avaliar:

- O código do aplicativo será clonado de algum repositório público e será executado através do XCode sobre diferentes dispositivos iOS emulados (a partir do iPhone 5);
- Vamos ler o código;
- Caso haja relatório de cobertura de testes, vamos analisá-lo;