

**Bootcamp: Desenvolvedor(a) React****Desafio**

<b>Módulo 2</b>	<b>React I</b>
-----------------	----------------

**Objetivos**

Exercitar os seguintes conceitos trabalhados no Módulo:

- ✓ Implementação de aplicações com JavaScript e React.
- ✓ Utilização de Function Components.
- ✓ Renderização de dados.
- ✓ Cálculos diversos.
- ✓ Utilização de Back End.

**Enunciado**

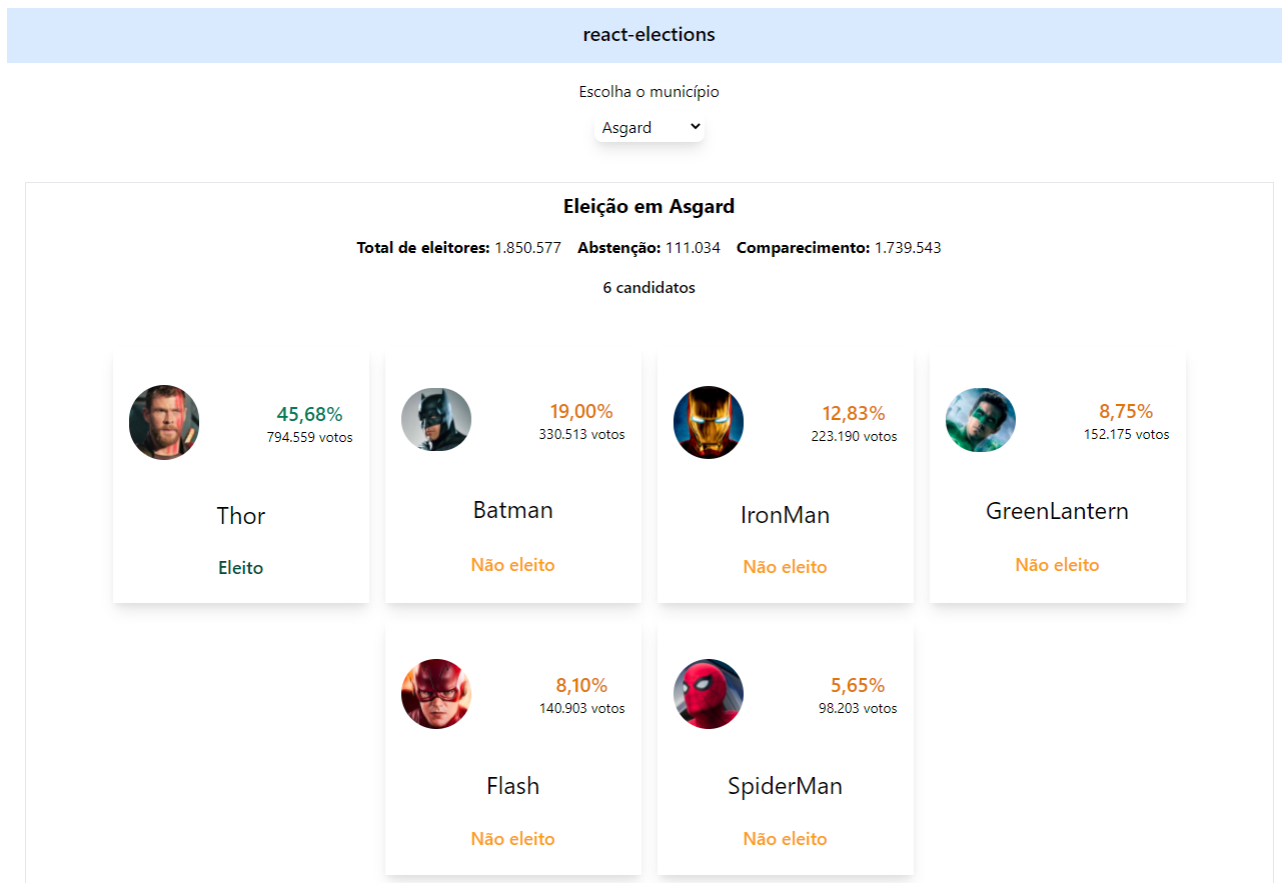
Construa, utilizando o React, uma aplicação para informar os resultados de eleições em 5 municípios.

**Atividades**

Os alunos deverão desempenhar as seguintes atividades:

- **Implementar**, utilizando o React, uma aplicação denominada "react-elections", que mostrará o resultado de eleições em 5 municípios. Os dados sobre as eleições serão fornecidos pelo professor no **Fórum de Avisos do Módulo**.

- A imagem abaixo ilustra um exemplo de implementação com o município "Asgard", que não será considerado no questionário.



Considerações importantes:

- **Imagens** – será fornecida, no **Fórum de Avisos do Professor**, uma pasta **img** com imagens dos candidatos. Esta pasta deve estar localizada dentro da pasta **public** do projeto de Front End e a url do componente de imagem deve ser acessada como **"/img/superman.png"**, por exemplo.
- **Estruturas de dados** – será fornecido, no **Fórum de Avisos do Professor**, um Back End fake com **json-server**, que irá monitorar o arquivo **elections.json**, que contém:
  - Um **array** denominado **cities**, que contém dados sobre os 5 municípios e informações sobre total de eleitores (**votingPopulation**), abstenções (quem não votou – **absence**) e presenças (diferença entre eleitores e abstenções – **presence**).

- Um **array** denominado **candidates**, que contém dados sobre os candidatos. O campo **username** pode ser utilizado para localizar a imagem correspondente do candidato. **Observação:** para simplificar o processo de geração dos dados de forma aleatória, considere que um mesmo candidato pode participar da eleição de mais de um município.
- Um **array** denominado **election**, que contém o vínculo entre município e candidato e informações sobre a votação. Este é o maior array e, na prática, deve ser filtrado e ordenado (maior votação >>> menor votação) para refletir determinada eleição em determinado município.
- O Back End on-line dará acesso às seguintes rotas/urls, que "devolvem" os dados conforme as descrições dos tópicos acima:

```
\{^_^}/ hi!  
  
Loading elections.json  
Done  
  
Resources  
http://localhost:3001/cities  
http://localhost:3001/candidates  
http://localhost:3001/election  
  
Home  
http://localhost:3001
```

- Há basicamente duas estratégias para a busca dos dados:
  1. Você pode buscar todo o back-end e trabalhar com todos os dados em memória.
  2. Você pode buscar por **cities** e **candidates**, mantendo-os em memória. A partir da cidade selecionada pelo usuário, você traz os dados referentes às eleições daquele município com a seguinte url:

<http://localhost:3001/election?cityId=d2dab6a2-3029-45a5-89f2-fcbaee387758> (neste exemplo, trago o resultado da votação de "Smallville", considerando os dados gerados no momento da criação deste documento, ou seja, o id (d2dab6a2-3029-45a5-89f2-fcbaee387758) pode possuir um outro valor em outras ofertas deste Desafio.

### Dicas e sugestões (são opcionais, mas recomendadas)

- Utilize o **projeto-base** fornecido pelo professor.
- "Levante" o back-end da mesma forma que fiz nas videoaulas.
- Utilize a tag `<select>` + `<option>`'s para renderizar a troca de cidades. Recomendo isolar isso em um componente `<Select>`. Informações importantes podem ser vistas [aqui](#).
- Há um projeto de **exemplo** que elaborei (utilizando Materialize CSS) que mostra a implementação da tag `<select>` na prática. Este projeto se encontra [aqui](#).
- Não será cobrado nada a respeito da interface gráfica, ou seja, implemente-a da maneira que você preferir.
- No questionário, serão feitas perguntas sobre o React e sobre o projeto implementado.
- Os valores foram gerados aleatoriamente e são totalmente fictícios.
- O cálculo de **percentual de votação** de cada **candidato** foi feito considerando os **votos válidos (comparecimento)**.
- Os valores dos percentuais de votação dos candidatos podem estar com aproximação. Na minha implementação, fiz o arredondamento dos cálculos para duas casas decimais. Sendo assim, garanto que não farei *pegadinhas* no questionário, do tipo 58,01% ou 58,02% como alternativas. Considere a alternativa cujo valor mais se aproxima do seu cálculo.
- Caso alguma questão esteja incorreta, solicitarei ao IGTI que a anule. Esse processo **não é "instantâneo" e leva algum tempo para acontecer – geralmente no final do módulo.**

Quando acontecer, **todos os alunos ganharão a pontuação da questão anulada, independente se tiverem acertado a questão ou não.**

- Demonstrarei minha implementação de forma didática durante a Aula Interativa 2.