






CRIANDO COM REDUX TOOLKIT UMA SELEÇÃO DE CAMPEÕES DO LOL EM MINUTOS



Por Marcell Guilherme

SOBRE MIM

- Desenvolvedor na Evolux;
- Fundador da Natal JS;
- Contribuidor free source;
- **Dota2 > LoL;**
- Talvez otaku.

 [@mazuh](#)
 [@marcellguilherme](#)
 [@marcell_mz](#)

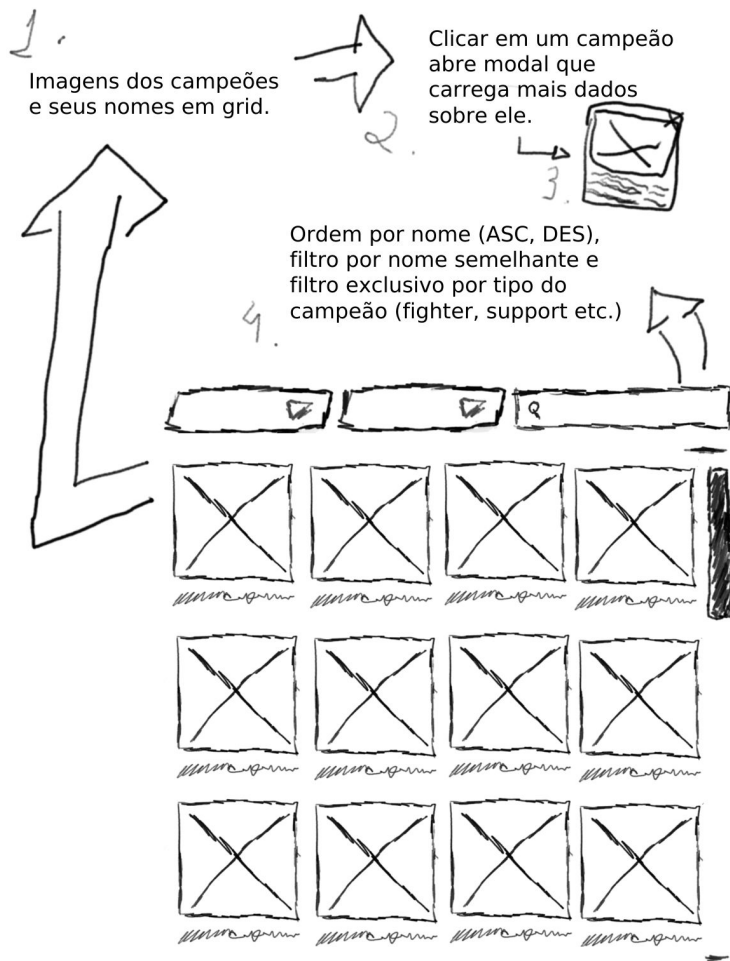


MARCELL G.



O QUE FAREMOS?





SELEÇÃO DE CAMPEÕES

Funções **semelhantes** ao do cliente do jogo, mas obviamente diferente.

Foco na programação, não estilização.

POR QUE REDUX?

E CONTEXT, MOBX, HOOKS...?

POR QUE REDUX?

E CONTEXT, MOBX, HOOKS...?

- **Redux** antigamente era muito “exagerado”, difícil de aprender e verboso de usar, mas hoje tem o **Redux Toolkit**;

POR QUE REDUX?

E CONTEXT, MOBX, HOOKS...?

- **Redux** antigamente era muito “exagerado”, difícil de aprender e verboso de usar, mas hoje tem o **Redux Toolkit**;
- Também usaremos **hooks** para acessar o Redux (por que não?);

POR QUE REDUX?

E CONTEXT, MOBX, HOOKS...?

- **Redux** antigamente era muito “exagerado”, difícil de aprender e verboso de usar, mas hoje tem o **Redux Toolkit**;
- Também usaremos **hooks** para acessar o Redux (por que não?);
- Context API não foi pensada para **estado global**;

POR QUE REDUX?

E CONTEXT, MOBX, HOOKS...?

- **Redux** antigamente era muito “exagerado”, difícil de aprender e verboso de usar, mas hoje tem o **Redux Toolkit**;
- Também usaremos **hooks** para acessar o Redux (por que não?);
- Context API não foi pensada para **estado global**;
- MobX vive no mundo Orientado a Objetos, ao contrário do React e Redux que caminham para o **Funcional e Declarativo**;

POR QUE REDUX?

E CONTEXT, MOBX, HOOKS...?

- **Redux** antigamente era muito “exagerado”, difícil de aprender e verboso de usar, mas hoje tem o **Redux Toolkit**;
- Também usaremos **hooks** para acessar o Redux (por que não?);
- Context API não foi pensada para **estado global**;
- MobX vive no mundo Orientado a Objetos, ao contrário do React e Redux que caminham para o **Funcional e Declarativo**;
- Tudo no Redux é **flexível e fácil de testar**, são só objetos puros e funções;

POR QUE REDUX?

E CONTEXT, MOBX, HOOKS...?

- **Redux** antigamente era muito “exagerado”, difícil de aprender e verboso de usar, mas hoje tem o **Redux Toolkit**;
- Também usaremos **hooks** para acessar o Redux (por que não?);
- Context API não foi pensada para **estado global**;
- MobX vive no mundo Orientado a Objetos, ao contrário do React e Redux que caminham para o **Funcional e Declarativo**;
- Tudo no Redux é **flexível e fácil de testar**, são só objetos puros e funções;
- Mensalmente **atualizado** até hoje.

POR QUE REDUX?

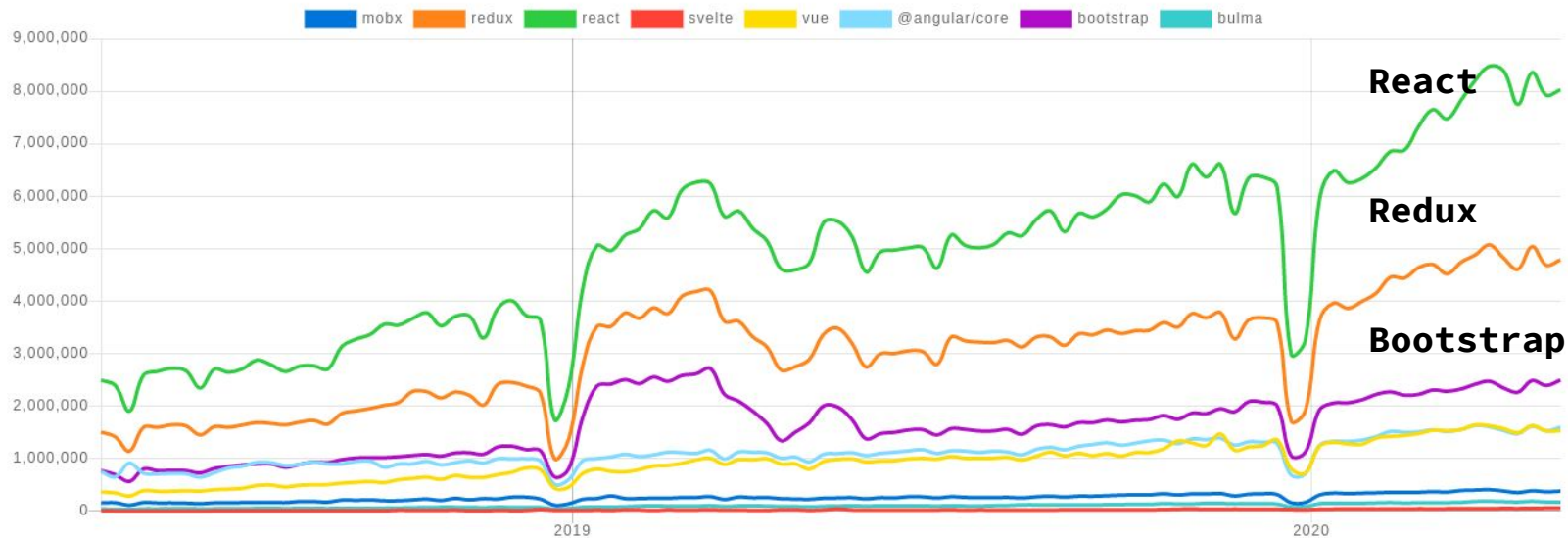
E CONTEXT, MOBX, HOOKS...?

- **Redux** antigamente era muito “exagerado”, difícil de aprender e verboso de usar, mas hoje tem o **Redux Toolkit**;
- Também usaremos **hooks** para acessar o Redux (por que não?);
- Context API não foi pensada para **estado global**;
- MobX vive no mundo Orientado a Objetos, ao contrário do React e Redux que caminham para o **Funcional e Declarativo**;
- Tudo no Redux é **flexível e fácil de testar**, são só objetos puros e funções;
- Mensalmente **atualizado** até hoje.

Extra: você **pode não precisar** de Redux e não há problema nisso! 😊

BÔNUS NÃO-TÉCNICO: COMUNIDADE

Downloads in past 2 Years ▾



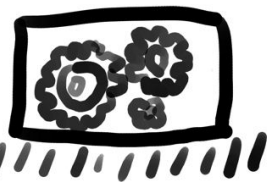
Fonte: NPM Trends (15/maio/2020)

COMO FUNCIONA:

1. Componentes criam e **dispatcham** a ação.



2. Função **reducer** combina a **ação** e o **estado antigo** para retornar um **novo estado**.



Dica: o estado é **encapsulado**, não deve ser lido nem alterado diretamente.



3. O **novo estado** é publicado; e os componentes **inscritos** recebem a alteração.

Dica: toda a árvore terá um **component-pai** de **provedor**.



Dica: todos os componentes inscritos são notificados de que o estado muda, mas só devem re-renderizar caso tenham se interessado na **parte do estado** que sofreu a alteração.

Imagem disponível em:

https://twitter.com/marcell_mz/status/1261419362688303104

MAS REDUX PURO É
TRABALHOSO. 



PRINCIPAIS BIBLIOTECAS

React

Fabricação de interfaces gráficas com paradigma declarativo.

React Bootstrap

Componentes gráficos atômicos já prontos para uso.

Redux Toolkit

Gerenciamento simples e eficiente de estado global.

BORA PROGRAMAR? 📡😊

Usaremos o Node 12.

Manual do Redux Toolkit:

<https://redux-toolkit.js.org/introduction/quick-start>

Dica:

`npx create-react-app lolzinho --template redux`

Do React Bootstrap:

<https://react-bootstrap.github.io/getting-started/introduction>

E da API do LoL/Riot:

<https://developer.riotgames.com/docs/lol>

