

 caelum
apresenta

As Aventuras do Cangaceiro JavaScript

Episódio 2: Desmistificando o Redux



com Gabriel Leite

O cangaceiro

Gabriel Leite

Formado em Ciência da
Computação pela Universidade
Federal do Rio de Janeiro.

Palestrante, professor,
desenvolvedor e líder/mentor de
instrutores na Caelum Rio e na Alura.



...nas horas vagas

No episódio anterior...

Desbravando o sertão do React



estado

```
<div class="uiToggle _4962 _3nzl _24xk">
  <a class="jewelButton _3eoa" href="#">
    <span class="jewelCount">
      <span class="_51lp hidden_elem _3z_5 _5ugh">1</span>
      <i class="accessible_elem">Solicitações de amizade</i>
    </span>
  </a>
  <div class="__tw toggleTargetClosed _3nzk uiToggleFlyout"></div>
</div>
```

```
import React from 'react';
import './index.css';
function FbJewelButton() {
  return (
    <div className="uiToggle _4962 _3nzl _24xk">
      <a className="jewelButton _3eoaa" href="#">
        <span className="jewelCount">
          <span className="_51lp hidden_elem _3z_5 _5ugh">0</span>
          <i className="accessible_elem">Solicitação de amizade</i>
        </span>
      </a>
      <div className="__tw_toggleTargetClosed _3nzk uiToggleFlyout"></div>
    </div>
  );
}

export default FbJewelButton;
```

```
import React, { Component } from 'react';

class FbJewelButton extends Component {

  render() {
    return (
      /* Elementos JSX omitidos. */
      <span className="jewelCount">
        <span className="_51lp hidden_elem _3z_5
_5ugh">{this.state.notifications}</span>
        <i className="accessible_elem">{this.props.name}</i>
      </span>
      /* Elementos JSX omitidos. */
    );
  }
}

export default FbJewelButton;
```

```
import React, { Component } from 'react';
```

```
class FbJewelButton extends Component {
```

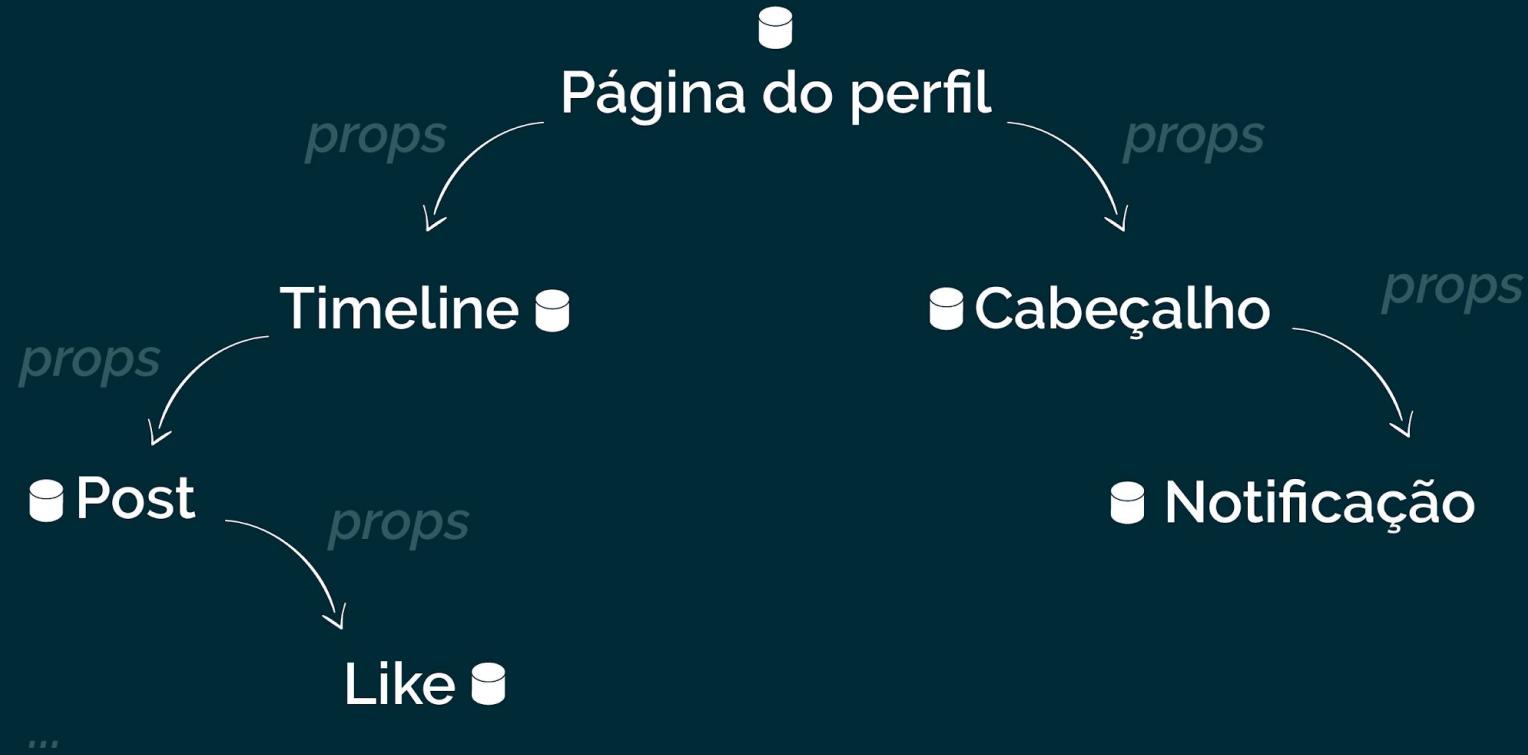
```
    constructor() {
        super();
        this.state = {
            notifications: 0
        };
    }
}
```

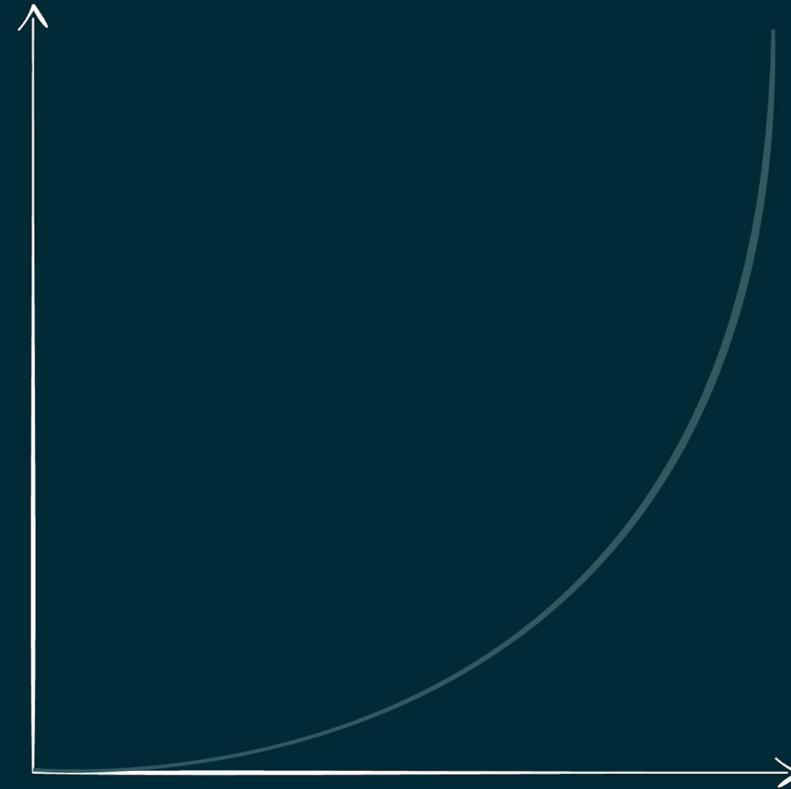
// Restante do código omitido.

```
}
```

```
export default FbJewelButton;
```

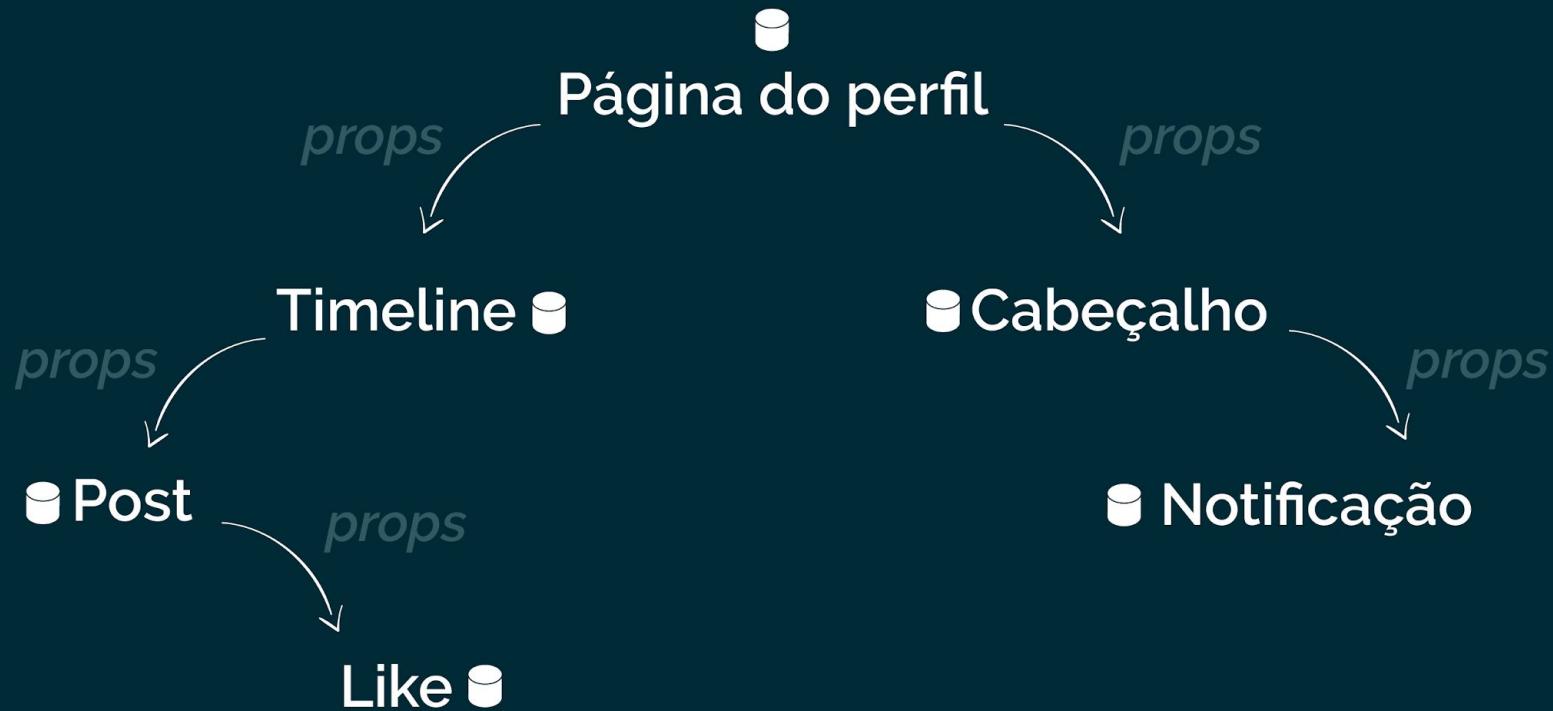
Por que usar Redux?



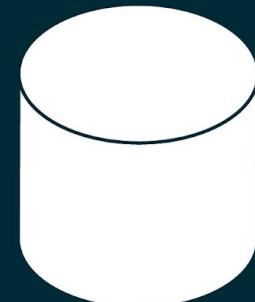
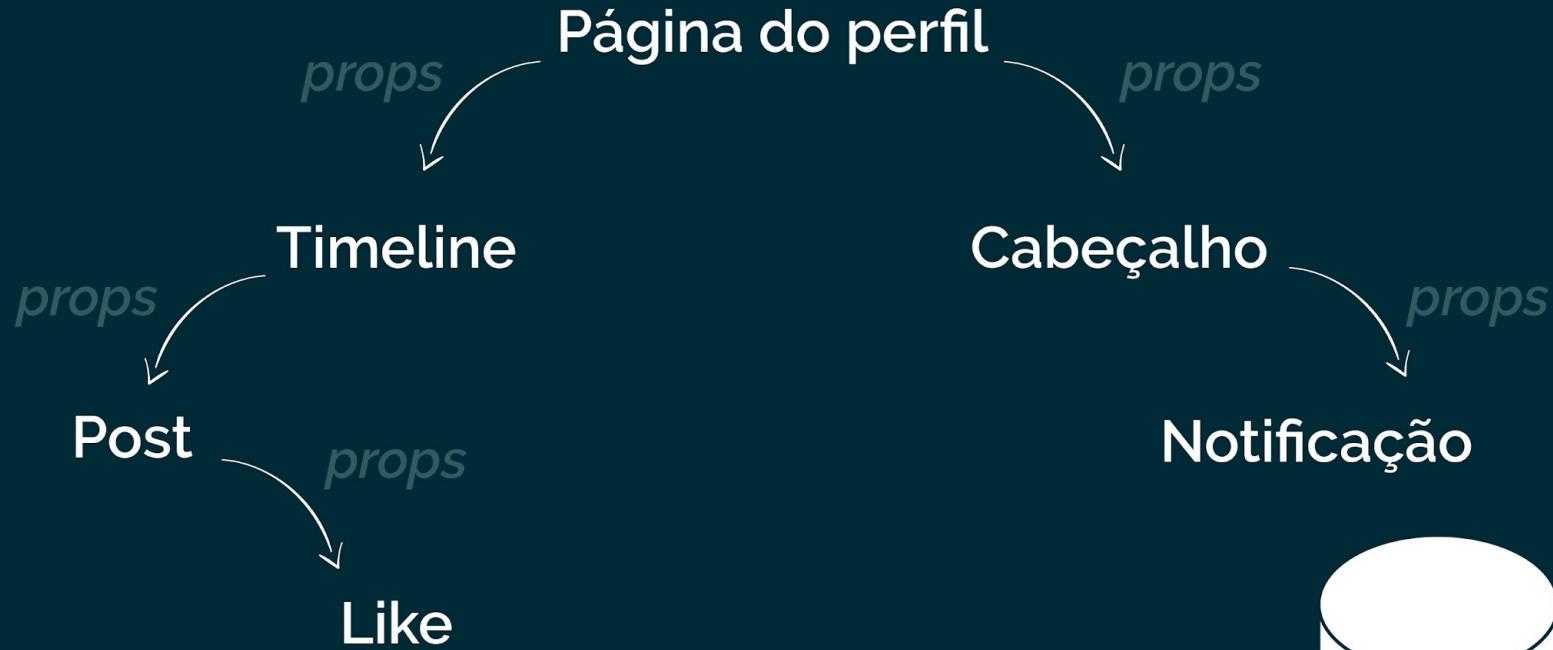


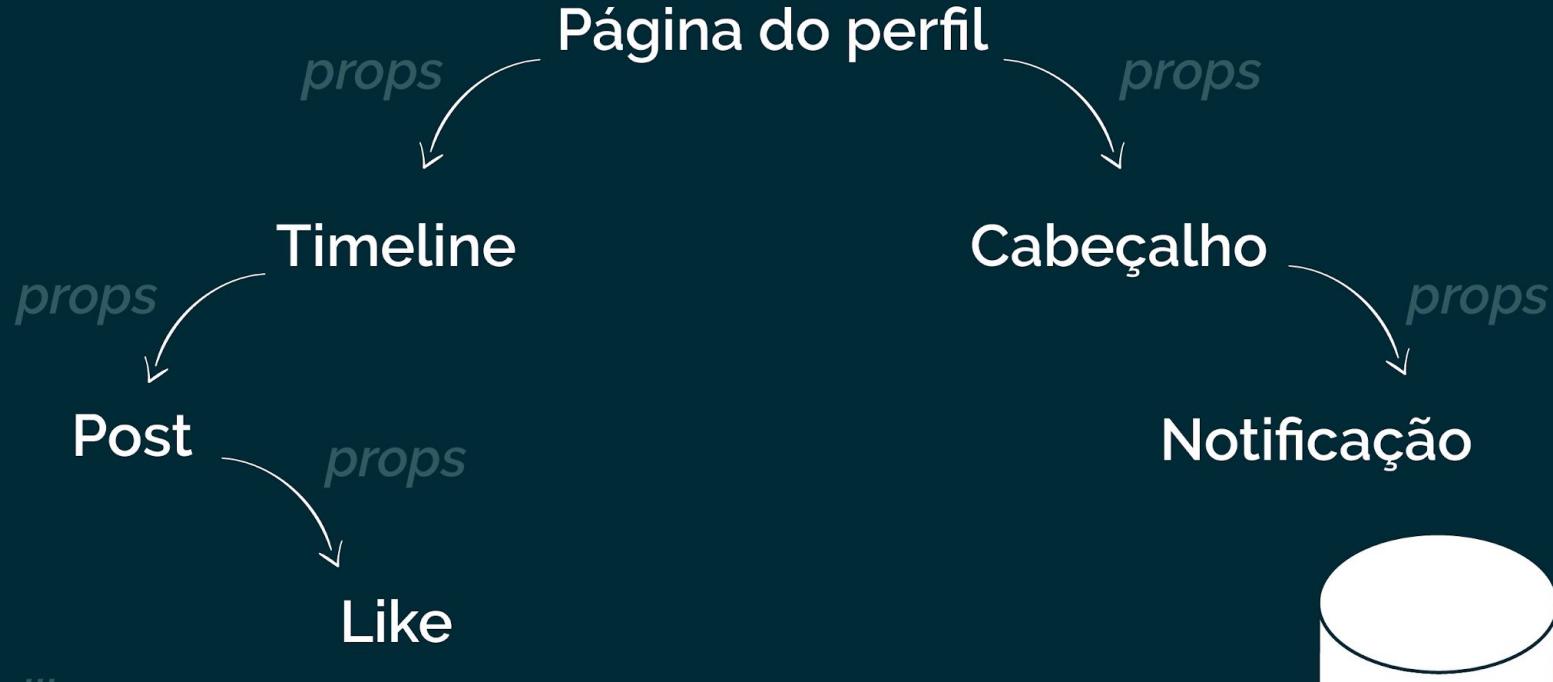
Linhas
de código

Caos



...





Episódio de hoje

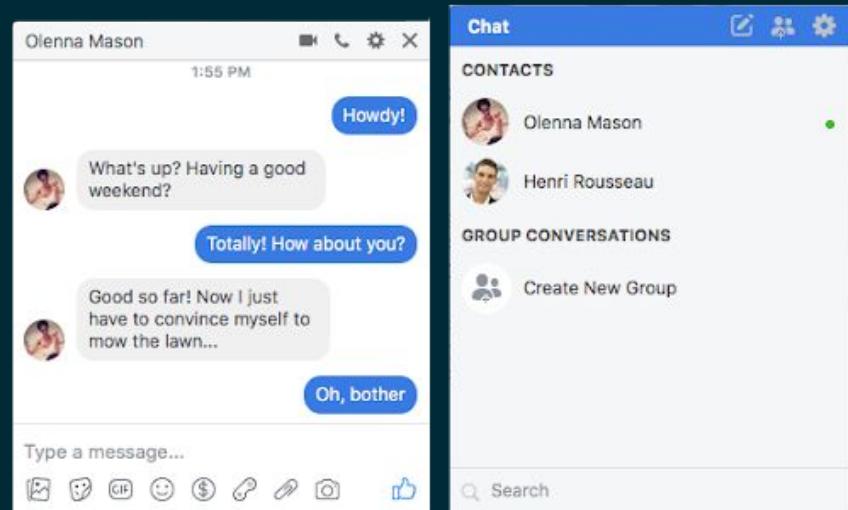
Parte 1: Entendendo o Redux

Um caso prático:
Chat do Facebook

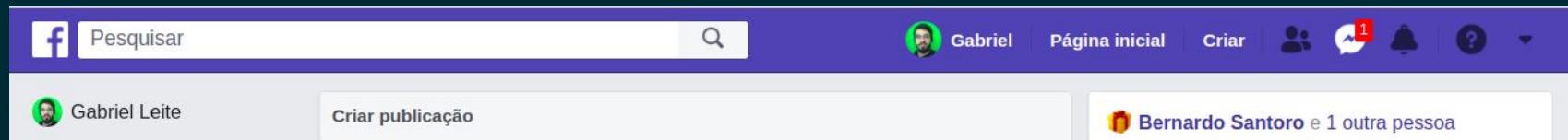
Nova mensagem no Chat

```
function newMessageHandler(message) {  
  const chatTab = ChatTabs.getTab(message.threadID);  
  
  if (chatTab.isOpen()) {  
    chatTab.appendMessage(message);  
  }  
}
```

1. Adicionar a mensagem ao chat se a aba estiver aberta



Nova mensagem no Chat



```
function newMessageHandler(message) {  
  const chatTab =  
    ChatTabs.getTab(message.threadID);  
  
  if (chatTab.isOpen()) {  
    chatTab.appendMessage(message);  
  }  
  
  if (!chatTab.hasFocus()) {  
    UnseenCount.increment();  
  }  
}
```

1. Adicionar a mensagem ao chat se a aba estiver aberta
2. Incrementar o contador de novas mensagens se a aba da conversa não tiver o foco

Nova mensagem no Chat

The screenshot shows a Facebook Messenger conversation. At the top, there's a purple header bar with the Facebook logo, a search bar, and user navigation links like 'Gabriel', 'Página inicial', 'Criar', and a profile icon. Below the header, the left sidebar shows a group chat named 'Bate-papos' and a message from 'Gabriel Leite' with a timestamp of '02:59'. The main conversation area shows messages from 'Gabriel Leite' with URLs. On the right side, there's a sidebar for 'Gabriel Leite' with options like 'OPÇÕES', 'Pesquisar na conversa', 'Editar apelidos', 'Alterar cor', 'Alterar emoji', 'PRIVACIDADE E SUPORTE', and 'ARQUIVOS COMPARTILHADOS'. A message input field at the bottom allows users to type a message.

Gabriel Leite
Online no Messenger

https://m.facebook.com/story.php?story_fbid=10156854917117058&id=558757057

21 DE OUT DE 2019 20:25

https://m.facebook.com/story.php?story_fbid=517381398808339&id=139400679939748

<https://www.facebook.com/1198530643491609/posts/2811810975496893/>

22 DE OUT DE 2019 19:51

<https://www.facebook.com/1428488850698604/posts/2425142544366558/>

<https://www.facebook.com/165975420254553/posts/1256297147889036/>

SEG, 20:11

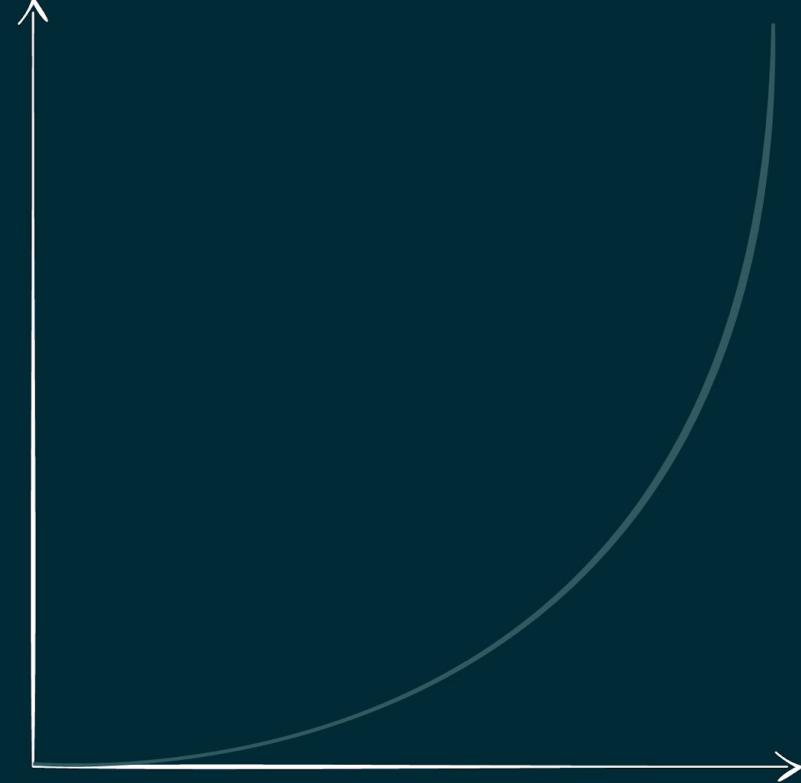
<https://www.facebook.com/134672973222186/posts/2833795626643227/>

Digite uma mensagem...

Nova mensagem no Chat

```
function newMessageHandler(message) {  
  const chatTab = ChatTabs.getTab(message.threadID);  
  
  if (chatTab.isOpen()) {  
    chatTab.appendMessage(message);  
  }  
  
  const messagesView = Messages.getOpenView();  
  const threadID = messagesView.getThreadID();  
  
  if (threadID == message.threadID) {  
    messagesView.appendMessage(message);  
  }  
  
  if (!chatTab.hasFocus() || threadID != message.threadID) {  
    UnseenCount.increment();  
  }  
}
```

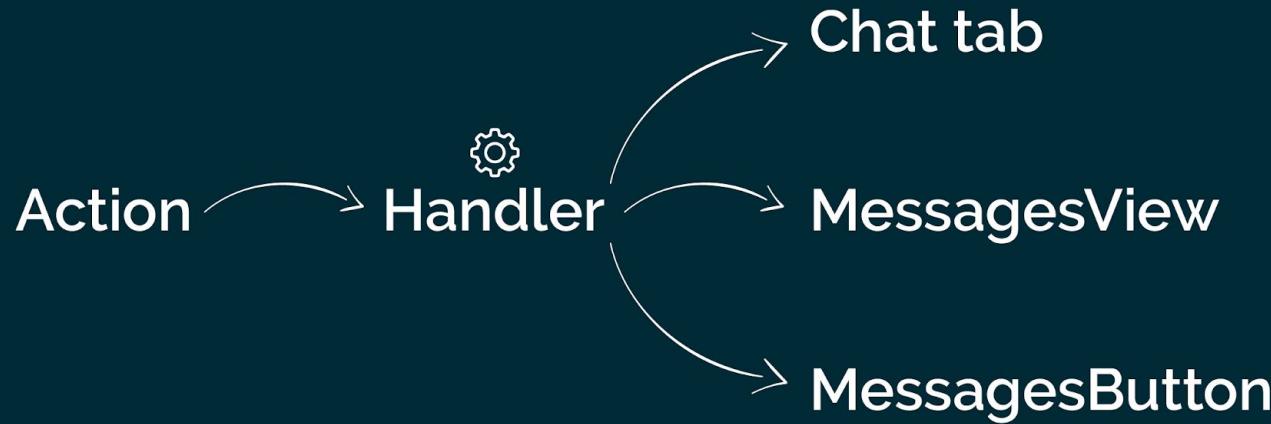
1. Adicionar a mensagem ao chat se a aba estiver aberta
2. Incrementar o contador de novas mensagens se a aba da conversa não tiver o foco
3. Se a tela principal de mensagens estiver aberta, adicionar a nova mensagem
4. Caso contrário, incrementar o contador de novas mensagens



Linhas
de código

Caos

Avaliando a situação

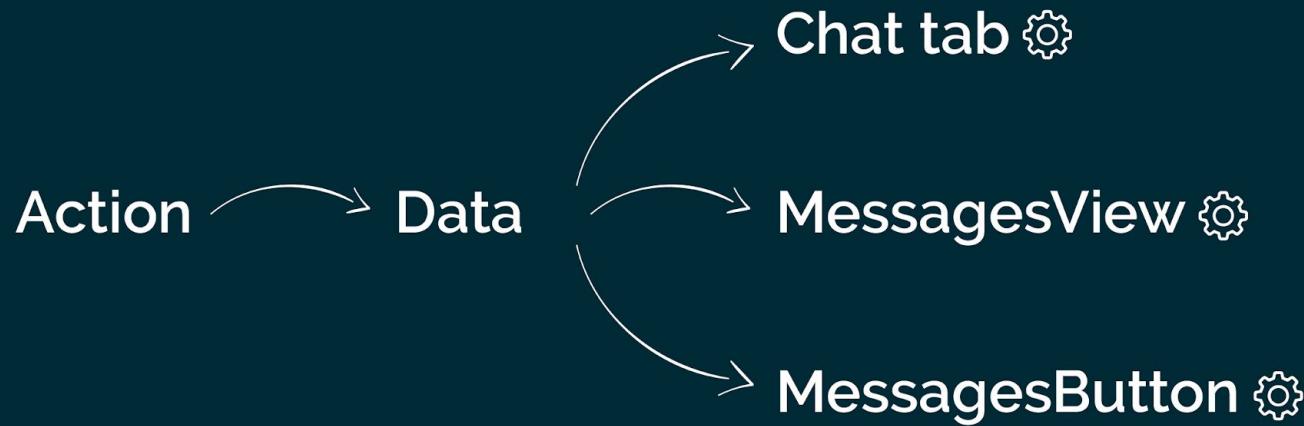


Como resolver esta situação?

“Simplicidade é pré-requisito para confiabilidade.”

- Edsger Dijkstra

O que gostaríamos





Parte 2: Preparando o ambiente

De posse do Node

```
$ npm install -g create-react-app  
$ create-react-app <nome-do-projeto>  
$ cd <nome-do-projeto>  
$ npm start
```

Instalando o Redux

```
$ npm install redux --save
```

Parte 3: Utilizando o Redux na aplicação



Criando o Reducer

```
function messagesReducer(state, action) {  
  if (action.type == 'ADD_MESSAGE') {  
    return [ ...state, action.payload.message ];  
  }  
}
```

Criando a store

```
function createStore(reducer) {  
  const messages = [];  
  const subscribers = [];  
  
  return {  
    dispatch: ???,  
    subscribe: ???,  
    getState: ???  
  };  
}
```

Criando o getState

```
function getState() {  
  return messages;  
}
```

Criando o subscribe

```
function subscribe(callback) {  
    subscribers.push(callback);  
}  
subscribers = [];
```

Criando o dispatch

```
function dispatch(action) {  
  messages = reducer(messages, action);  
  subscribers.forEach(callback => {  
    callback();  
  }) ;  
}
```

```
function createStore(reducer) {  
  const messages = [];  
  const subscribers = [];  
  
  function dispatch(action) {  
    // ...  
  }  
  
  function subscribe(callback) {  
    // ...  
  }  
  
  function getState() {  
    // ...  
  }  
  
  return {  
    dispatch,  
    subscribe,  
    getState  
  };  
}
```

De volta ao createStore

Em algum componente...

1. Se increvendo na store

```
store.subscribe(() => {  
  console.log('Store changed');  
}) ;
```

2. Obtendo o estado atual

```
store.getState();
```

3. Enviando uma ação

```
store.dispatch({  
  type: 'ADD_MESSAGE',  
  payload: {  
    message: { ... }  
  }  
}) ;
```

Criação da store na prática

```
// ./src/store.js

import { createStore } from 'redux';

function messagesReducer(state, action) {
  if (action.type == 'ADD_MESSAGE') {
    return [ ...state, action.payload.message ];
  }
}

const store = createStore(messagesReducer);
```



```
import React from 'react';
import './index.css';
function FbJewelButton() {
  return (
    <div className="uiToggle _4962 _3nzl _24xk">
      <a className="jewelButton _3eoa" href="#">
        <span className="jewelCount">
          <span className="_51lp hidden_elem _3z_5 _5ugh">0</span>
          <i className="accessible_elem">Solicitação de amizade</i>
        </span>
      </a>
      <div className="__tw_toggleTargetClosed _3nzk uiToggleFlyout"></div>
    </div>
  );
}

export default FbJewelButton;
```

```
import React, { Component } from 'react';
```

```
class FbJewelButton extends Component {
```

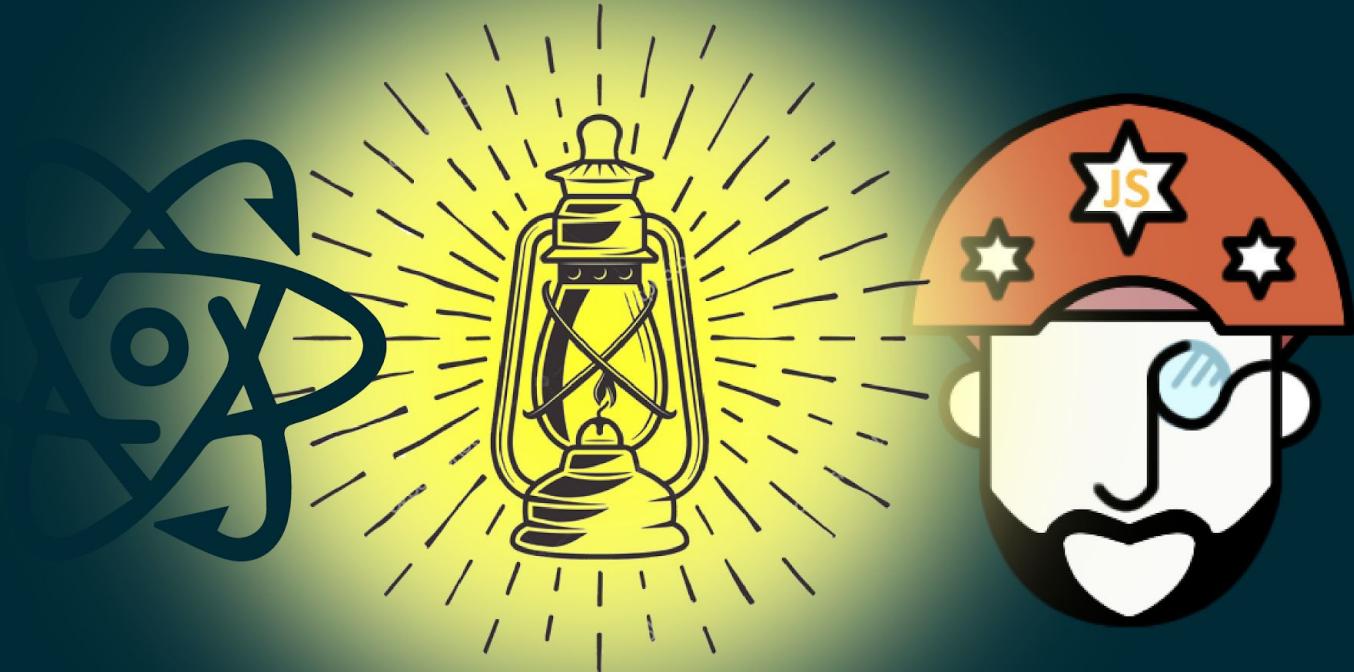
```
    constructor() {
        super();
        this.state = {
            notifications: 0
        };
    }
}
```

```
// Restante do código omitido.
```

```
}
```

```
export default FbJewelButton;
```

Continua no
próximo episódio...



Obrigado!