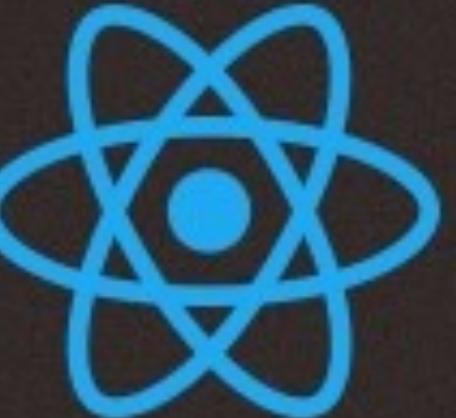


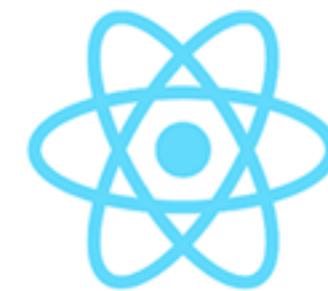
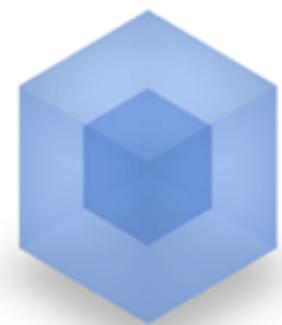
Oficina

 **React**

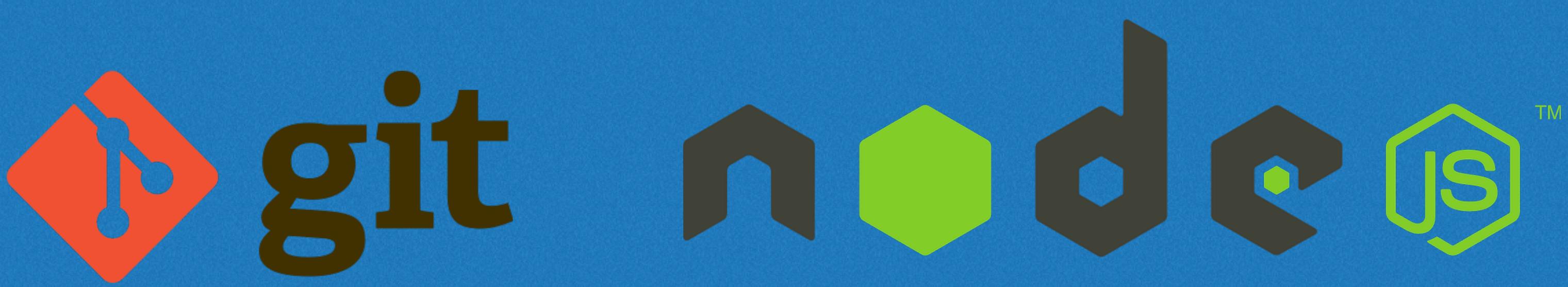
 **Firebase**

Perfil dessa oficina

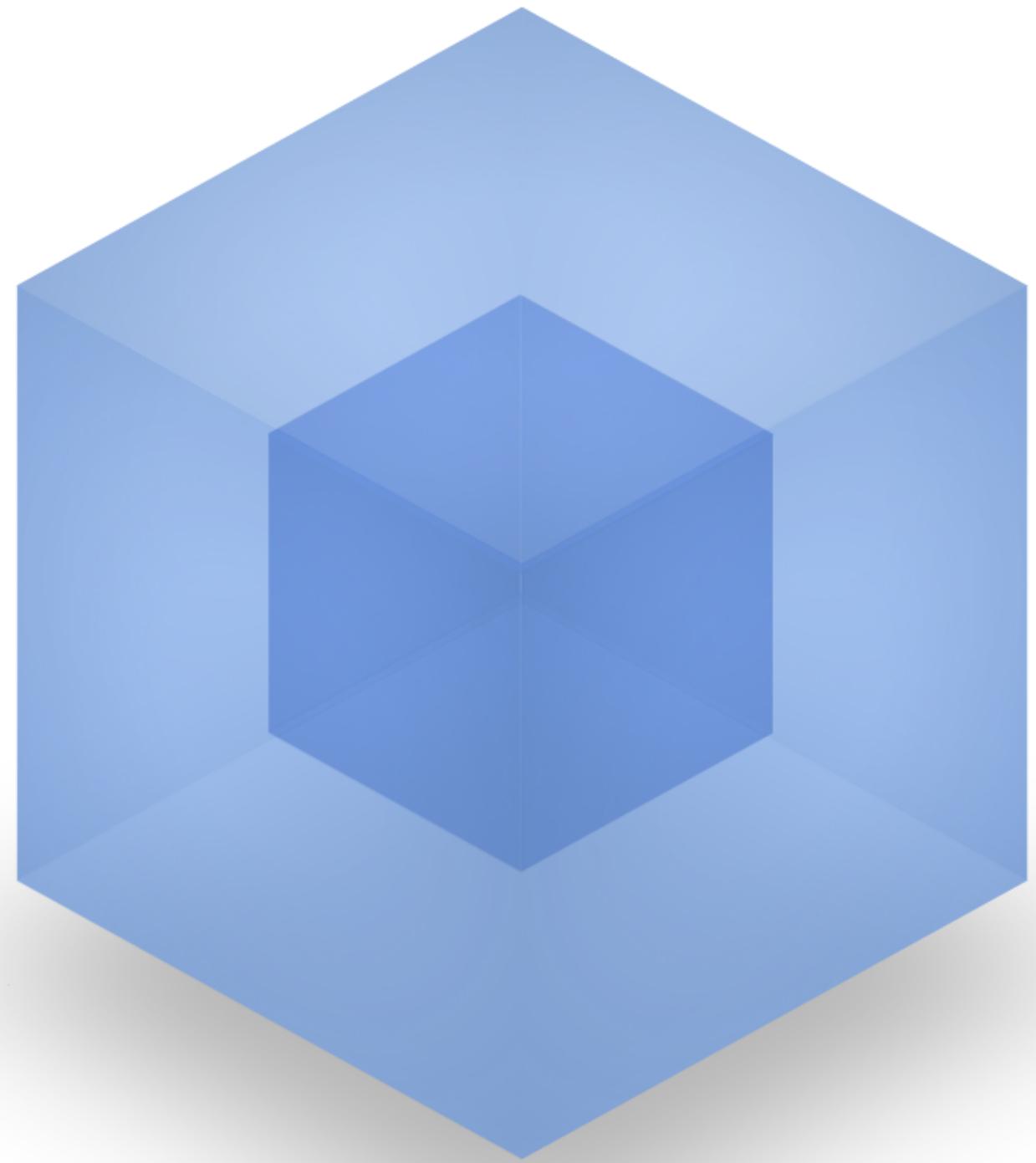
- React
- Webpack e Babel
- React Router
- Redux
- Firebase
- Best practices



Requisitos



<https://github.com/gdg-cascavel/workshop-react-firebase>

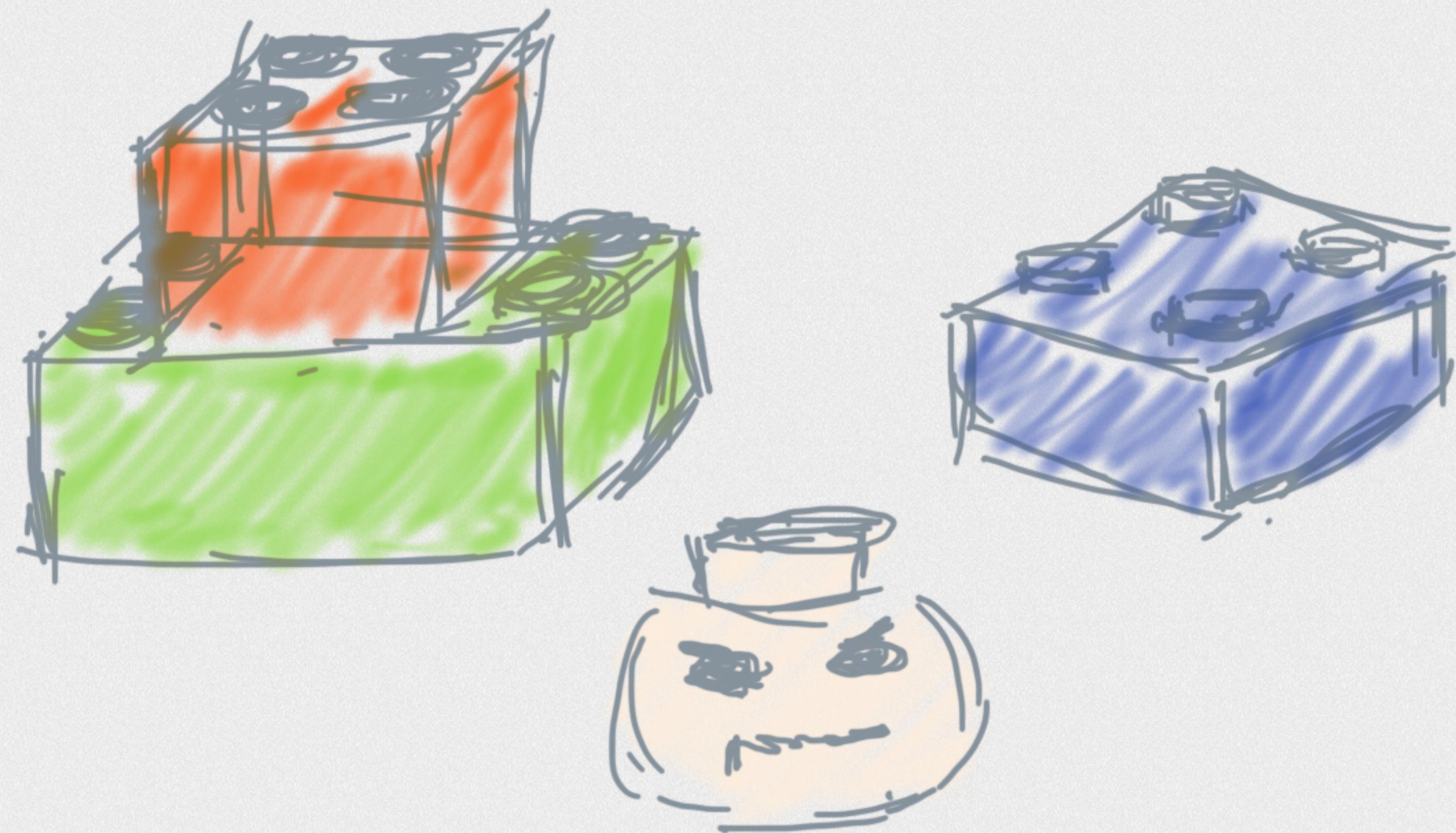


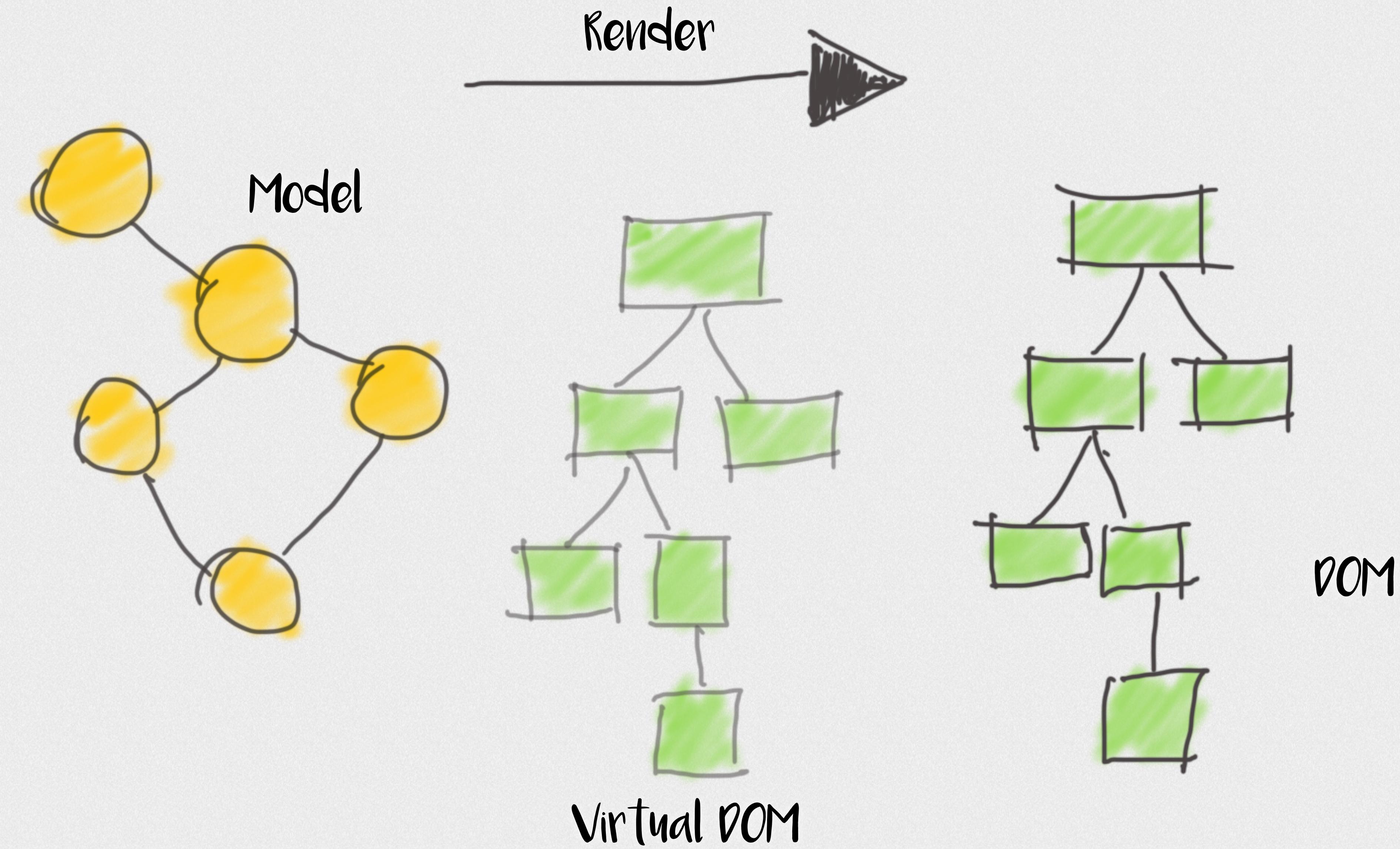
Webpack
Module Bundler

```
$ git clone https://github.com/gdg-cascavel/workshop-react-firebase.git  
$ cd workshop-react-firebase/lesson-01-setup-and-config  
$ npm install  
$ npm start
```


O que é React?
React

*"A javascript library for building
user interfaces"*







```
<div  
  class="page-content"  
  data-active="true"  
/>
```

```
{  
  tagName: 'div',  
  classes: ['page-content'],  
  attributes: {  
    'data-active': true,  
  },  
}
```

```
const HelloWorld = () => (
  <h1>Hello World!</h1>
);

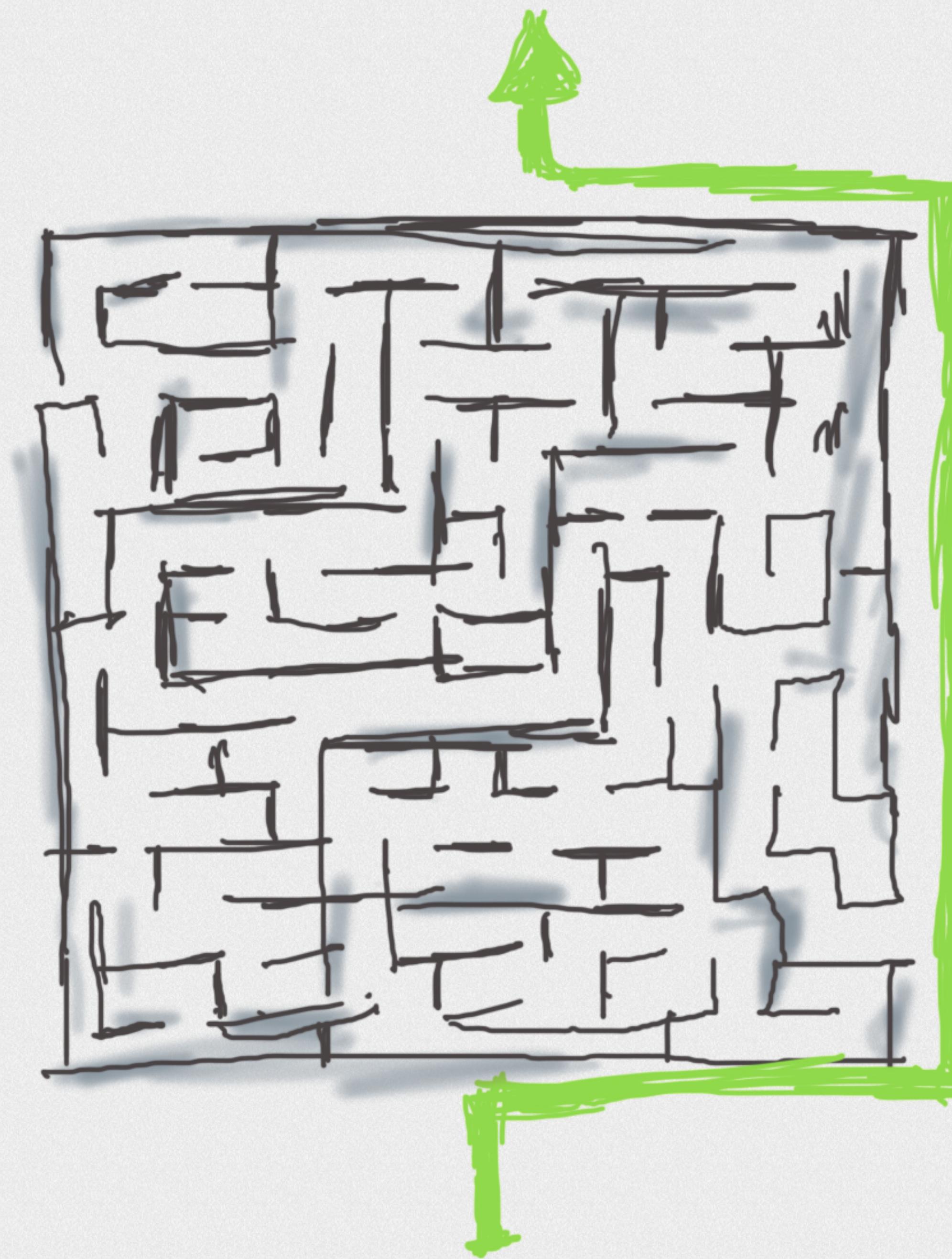
ReactDOM.render(<HelloWorld />, mountNode);
```

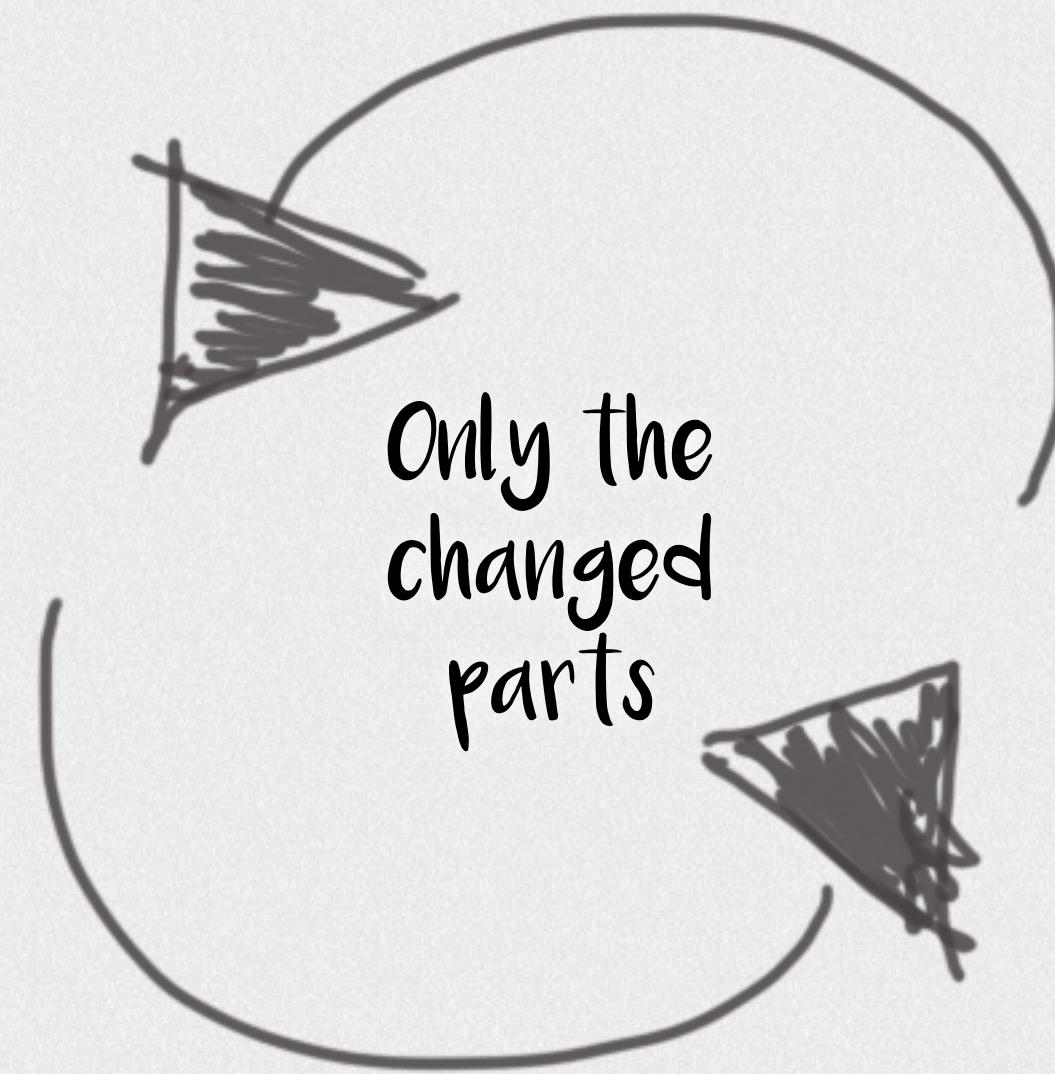
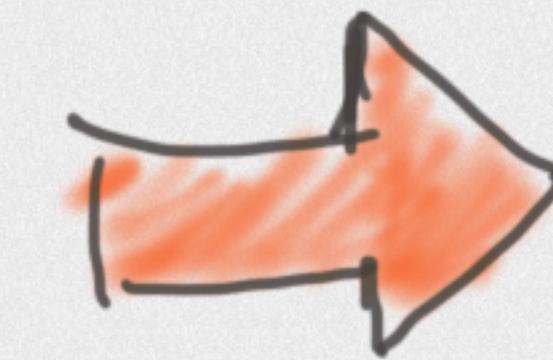
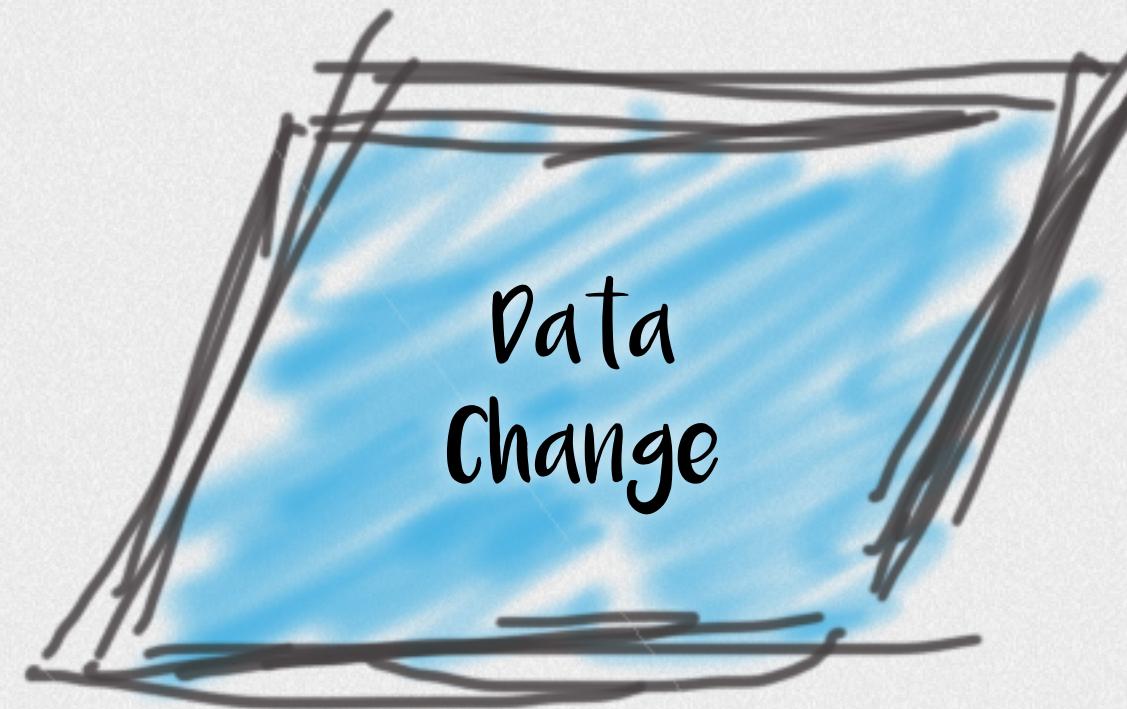
Por que escolher React?

React

Simples

Por que escolher React?





Declarativo

Por que escolher React?

Imperative

Go to kitchen
Open fridge
Remove chicken from fridge

...
Bring food to table

Declarative

I want a dinner with chicken.

Imperative

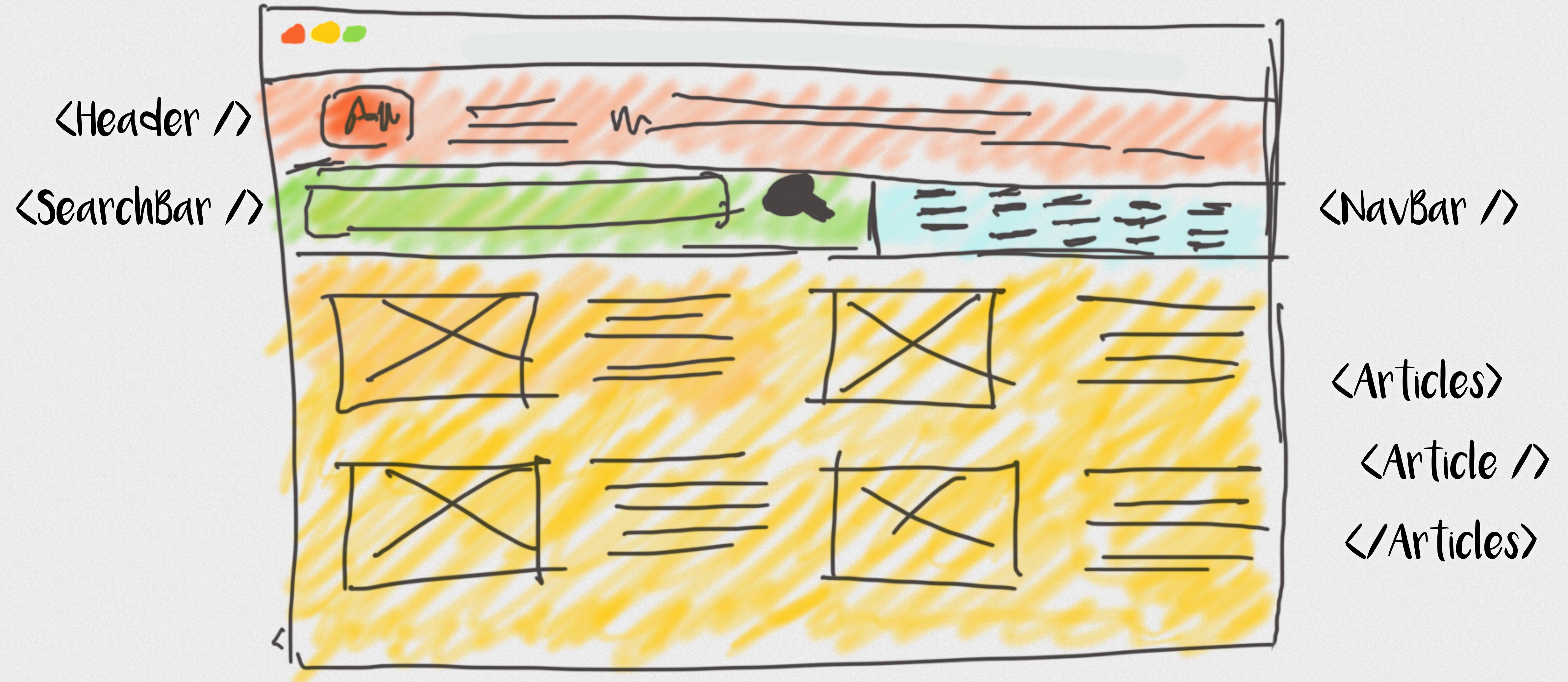
```
if (user.likes()) {  
    if (!hasBlue()) {  
        removeGrey();  
        addBlue();  
    }  
} else {  
    if (hasBlue()) {  
        removeBlue();  
        addGrey();  
    }  
}
```

Declarative

```
if (this.state.liked) {  
    return <BlueLike />;  
} else {  
    return <GreyLike />;  
}
```

Baseado em componente

Por que escolher React?



```
class HelloWorld extends React.Component {  
  render() {  
    return <h1>Hello World!</h1>;  
  }  
}
```

```
ReactDOM.render(<HelloWorld />, mountNode);
```

```
const HelloWorld = () => (  
  <h1>Hello World!</h1>  
)
```

```
ReactDOM.render(<HelloWorld />, mountNode);
```

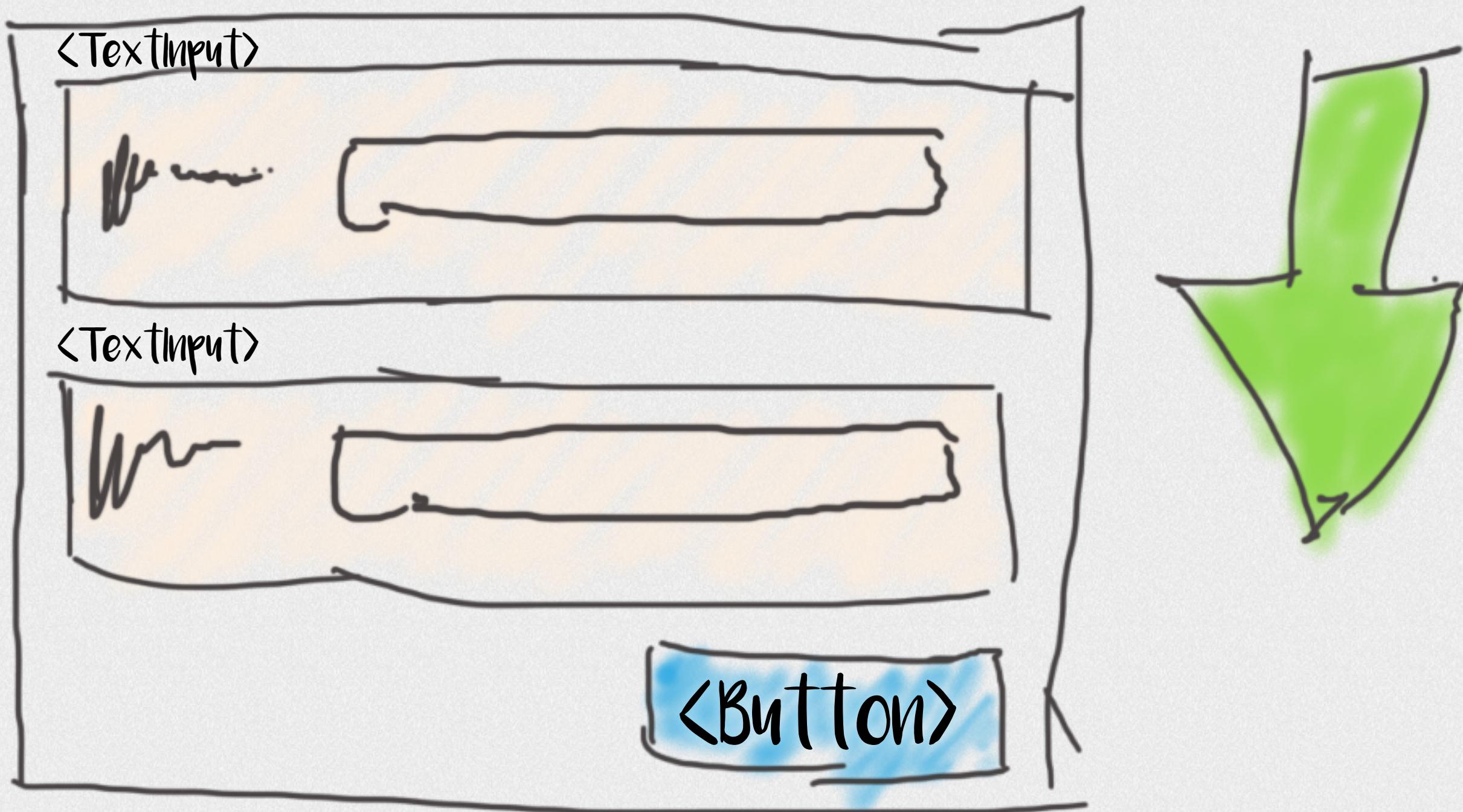
Propriedades

React

```
class HelloMessage extends React.Component {  
  constructor(props) {  
    super(props);  
  }  
  
  render() {  
    return <div>Hello {this.props.name}</div>;  
  }  
};  
  
const HelloMessage = ({ name }) => (  
  <div>Hello {name}</div>  
);  
  
ReactDOM.render(<HelloMessage name="John" />, mountNode);
```



`<Form>`



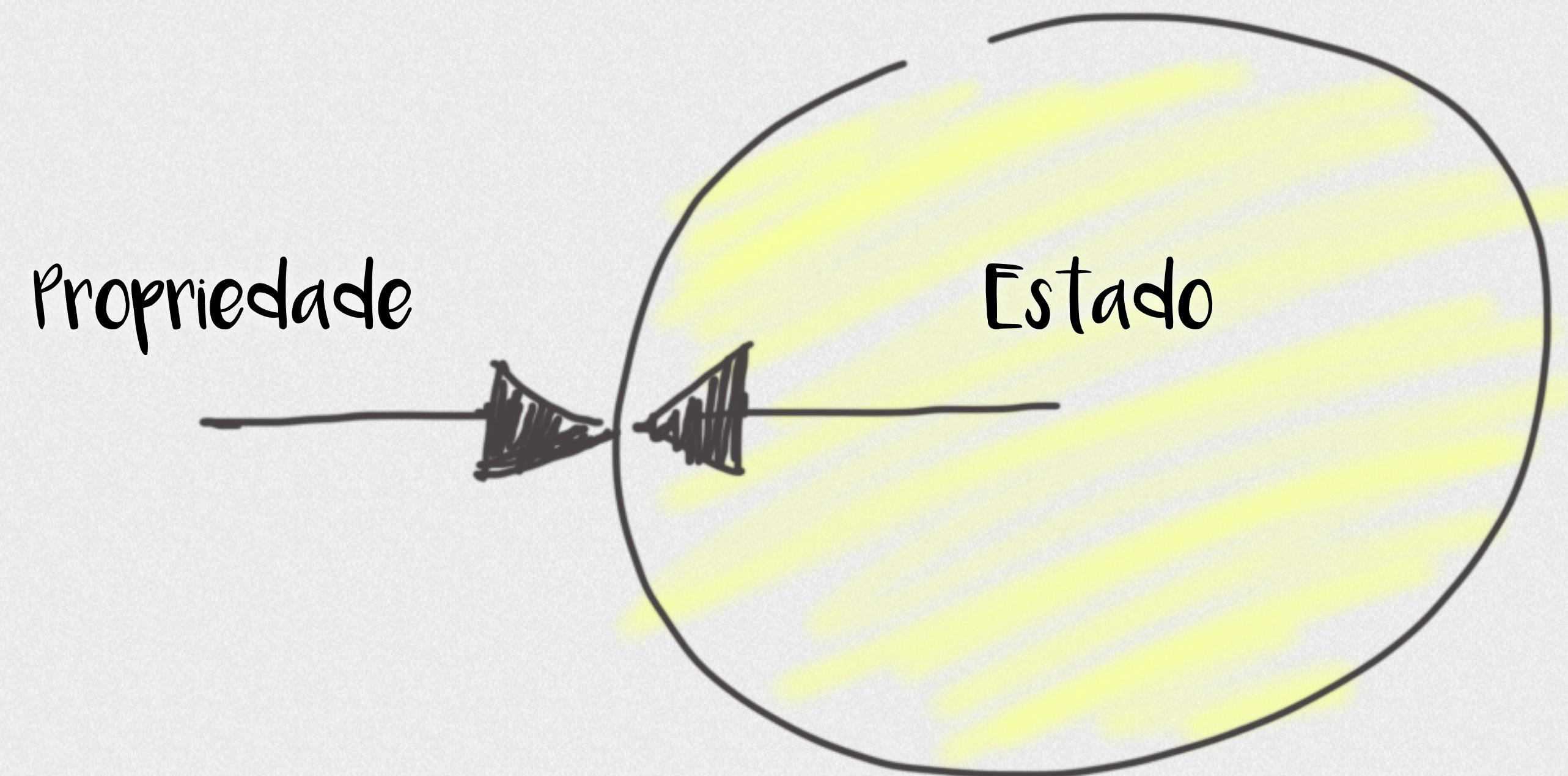
Estado
React

```
class TimedMessage extends React.Component {  
  constructor(props) {  
    super(props);  
  
    this.state = {  
      message: 'Hello',  
    };  
  }  
  
  componentDidMount() {  
    setTimeout(() => {  
      this.setState({  
        message: 'Bye',  
      });  
    }, 2000);  
  }  
  
  render() {  
    return <div>{this.state.message} {this.props.name}</div>;  
  }  
};
```

```
ReactDOM.render(<TimedMessage name="John" />, mountNode);
```

Estado vs Propriedade

React



Methods & Specs

React

```
void setState(  
  function|object nextState,  
  [function callback]  
)
```

```
void replaceState(  
  object nextState,  
  [function callback]  
)
```

```
void forceUpdate(  
  [function callback]  
)
```

```
boolean isMounted()
```

object getInitialState()

object getDefaultProps()

object propTypes

Component and Lifecycle

React

Mounting

```
componentWillMount() {  
    // invoked before rendering  
    // can call setState here, render() will  
    see the updated state and only execute once  
}  
componentDidMount() {  
    // invoked after rendering, can now access  
    this.getDOMNode()  
}  
componentWillUnmount() {  
    // invoked just before a component is  
    unmounted from the DOM  
}
```

Updating

```
componentWillReceiveProps(nextProps) {  
    // invoked when receiving new props; can call  
    this.setState() here  
}  
shouldComponentUpdate() {  
    // invoked before rendering new props or  
    state, can return false to stop re rendering  
}  
componentWillUpdate() {  
    // invoked before rendering new props or  
    state; cannot call this.setState() here  
}  
componentDidUpdate() {  
    // invoked after the component updates are  
    flushed to the DOM  
}
```

The React Way

React

Começar com um mock

The React Way

todos

▼ What needs to be done?

- Quebrar essa tela em componentes ×
- Fazer uma versão estática
- Ver quais são os estados necessários
- Adicionar o fluxo de dados de forma inversa

4 items left

All

Active

Completed

Double-click to edit a todo

Created by petehunt

Part of TodoMVC

Dividir UI em componentes

The React Way

todos

▼ *What needs to be done?*

- Quebrar essa tela em componentes ×
- Fazer uma versão estática
- Ver quais são os estados necessários
- Adicionar o fluxo de dados de forma inversa

4 items left

All

Active

Completed

Double-click to edit a todo

Created by peternor

Part of TodoMVC

Versão estática

The React Way

```
$ npm i todomvc-app-css todomvc-common -S
```

Identifique o estado mínimo
[mas completo]
The React Way

Adicione o fluxo inverso de dados

The React Way



Firebase

O que é Firebase?
Firebase

Por que escolher Firebase?
Firebase

Authentication

Firebase

Realtime Database

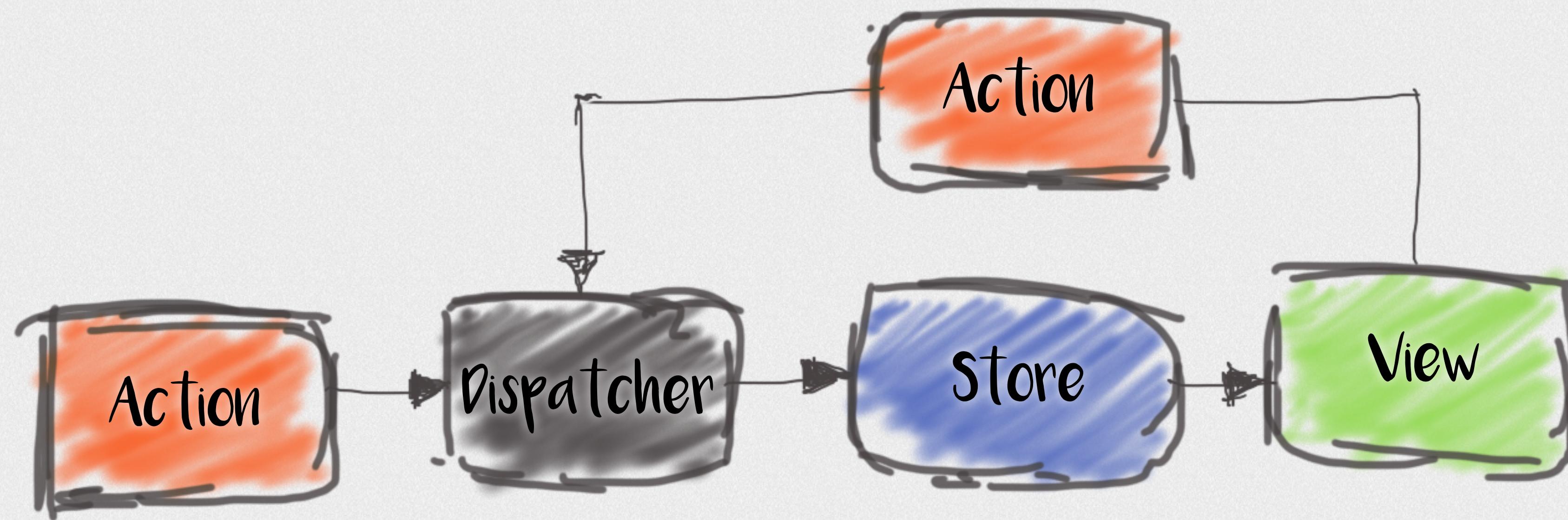
Firebase

```
class TodoApp extends Component {  
  
  constructor() {  
    this.firebaseioRef = firebase.database().ref('todoApp/items');  
  }  
  
  componentWillMount() {  
    this.firebaseioRef.limitToLast(25).on('value', (dataSnapshot) => {  
      let items = [];  
      dataSnapshot.forEach((childSnapshot) => {  
        let item = childSnapshot.val();  
        item['.key'] = childSnapshot.key;  
        items.push(item);  
      });  
  
      this.setState({  
        items: items  
      });  
    });  
  }  
}
```



Redux

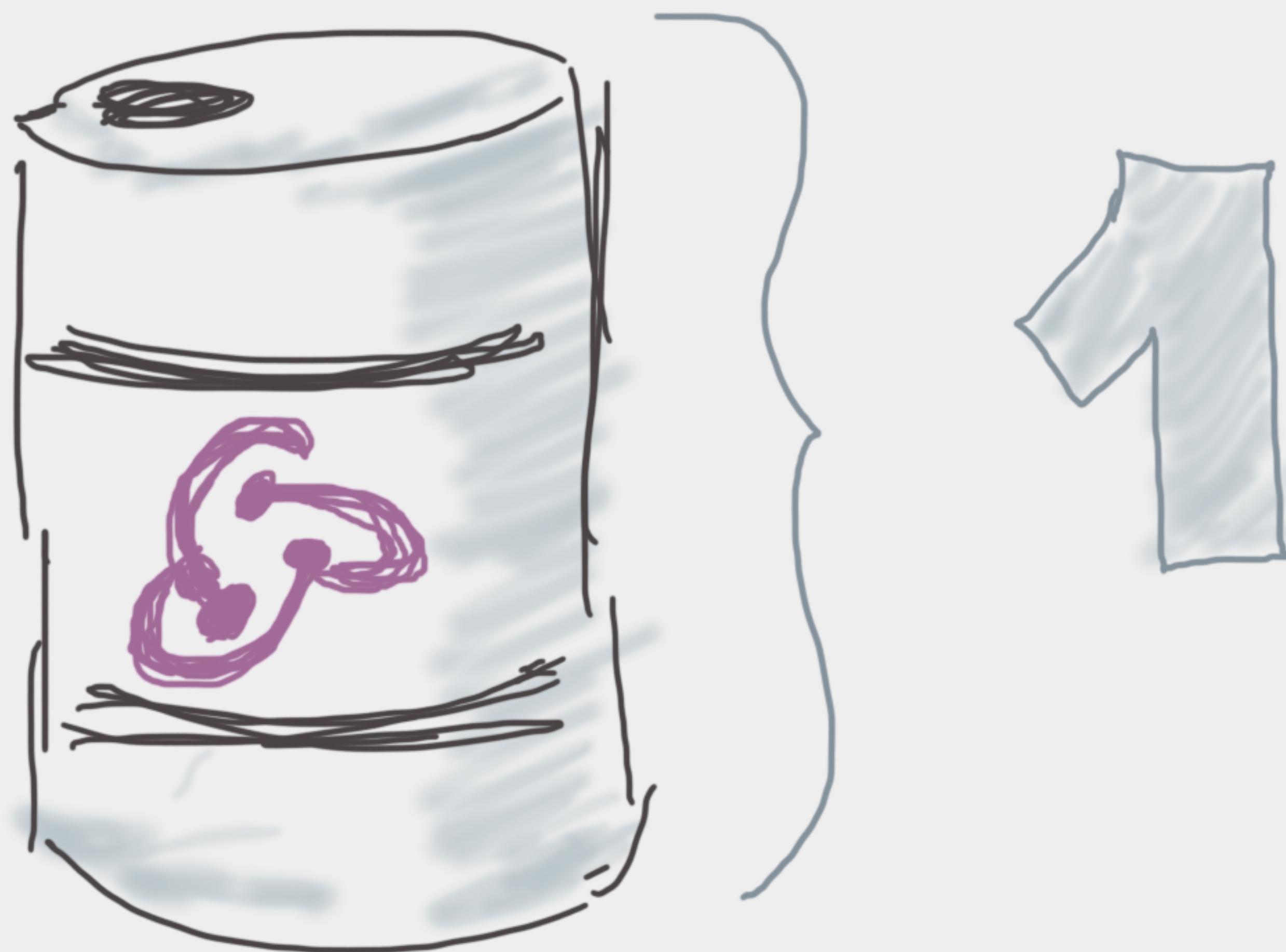
Flux? Mas porque? Redux



O que é Redux

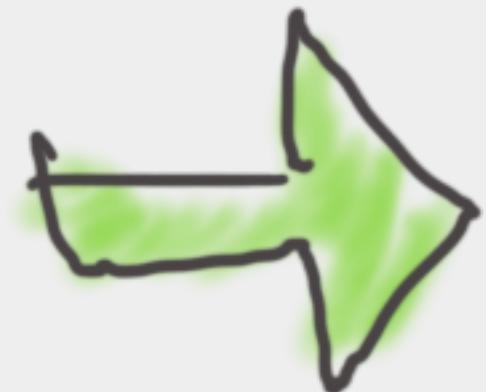
Redux

**"Redux is a predictable state container
for Javascript apps."**



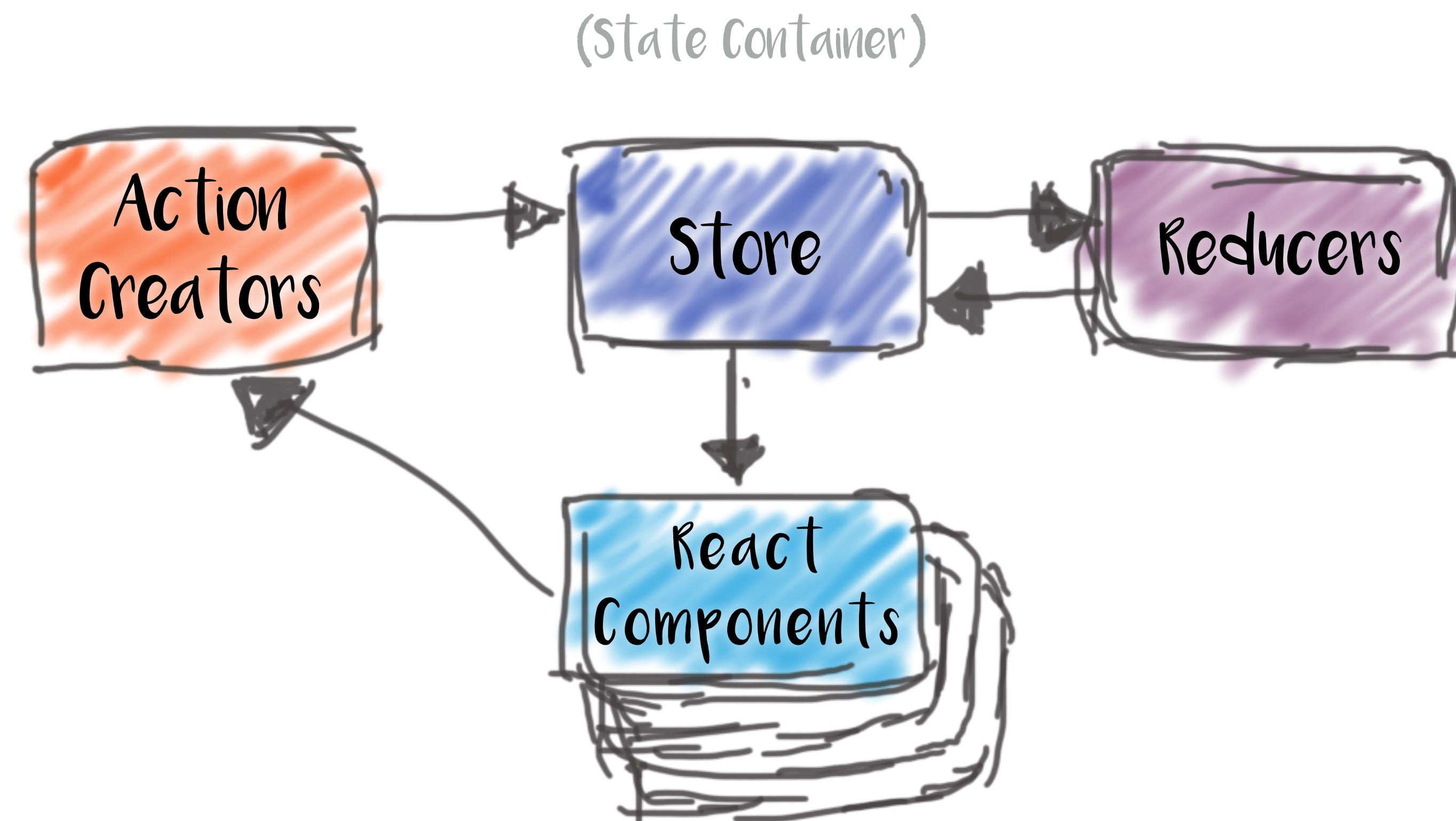
Por que Redux?

Redux



Actions, Reducers e Stores

Redux



Redux Way

Redux

```
$ npm i redux react-redux -S
```

todos

What needs to be done?

- Quebrar essa tela em componentes X
- Fazer uma versão estática
- Ver quais são os estados necessários
- Adicionar o fluxo de dados de forma inversa

4 items left

All Active Completed

Double-click to edit a todo

Created by petehunt

Part of TodoMVC

Adding Redux to our Todo App

UPDATE APP FOLDER STRUCTURE

- actions/
- components/
- containers/
- reducers/
- store/

CONFIGURE THE REDUCERS AND THE STORE

DEFINE THE ACTIONS

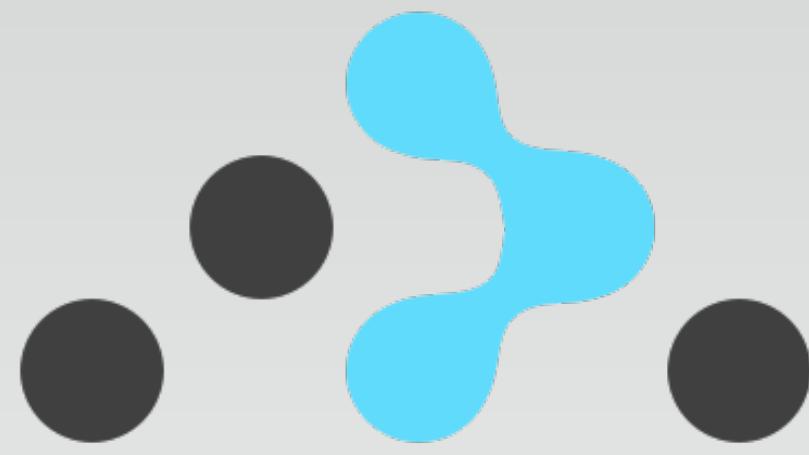
- ADD_TODO
- DELETE_TODO
- EDIT_TODO
- COMPLETE_TODO
- COMPLETE_ALL
- CLEAR_COMPLETED

CONNECT REDUX TO EVERYTHING

Angular



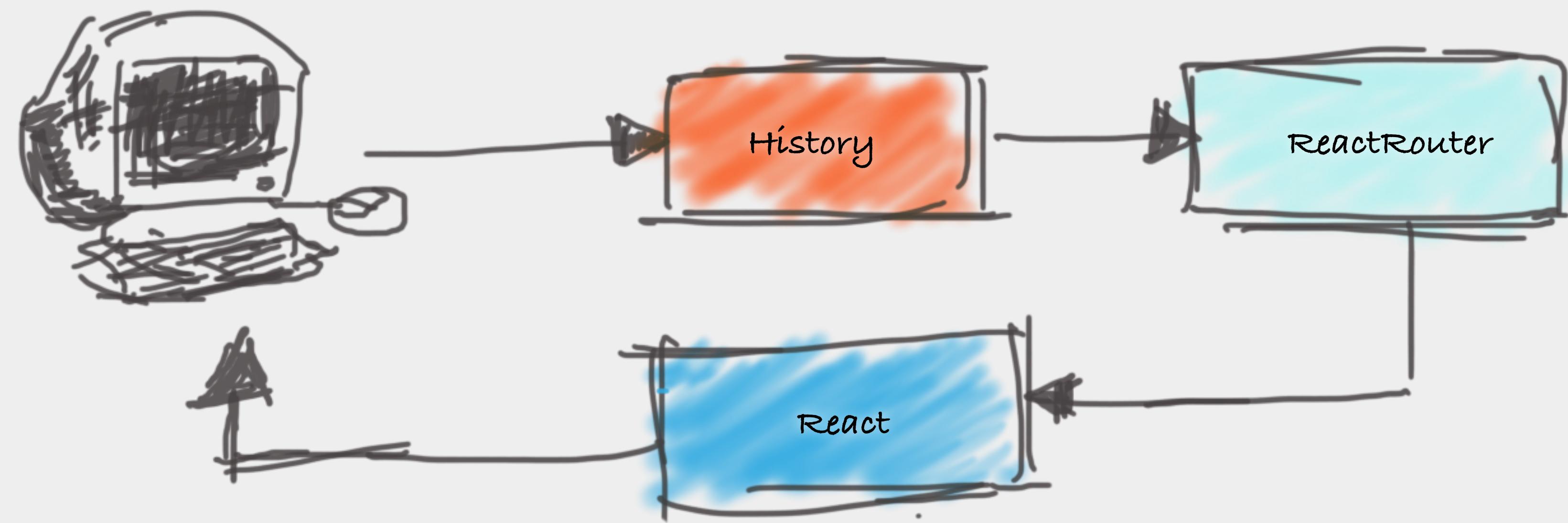
ngRedux



REACT ROUTER

O que é o React Router?

React Router



Agradecimentos

