



{LET'S REACT}



WOMAKERS  
CODE

{o que é React?}

É uma biblioteca  
em javascript  
para construir  
interfaces  
inteligentes

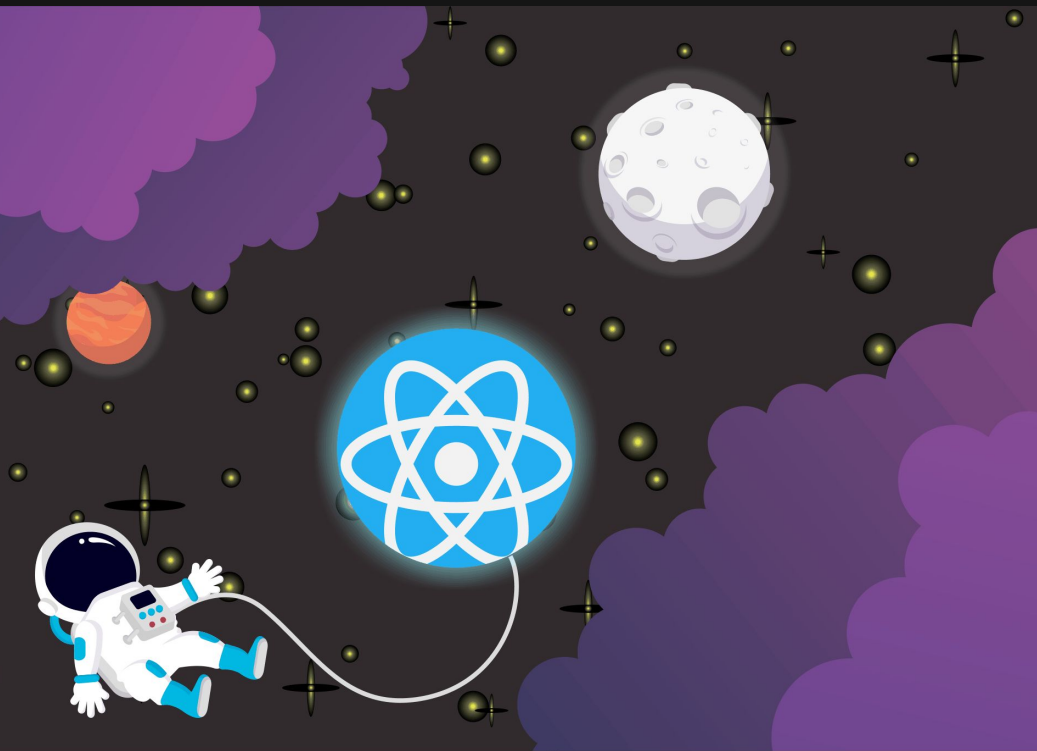


{como é possível?}

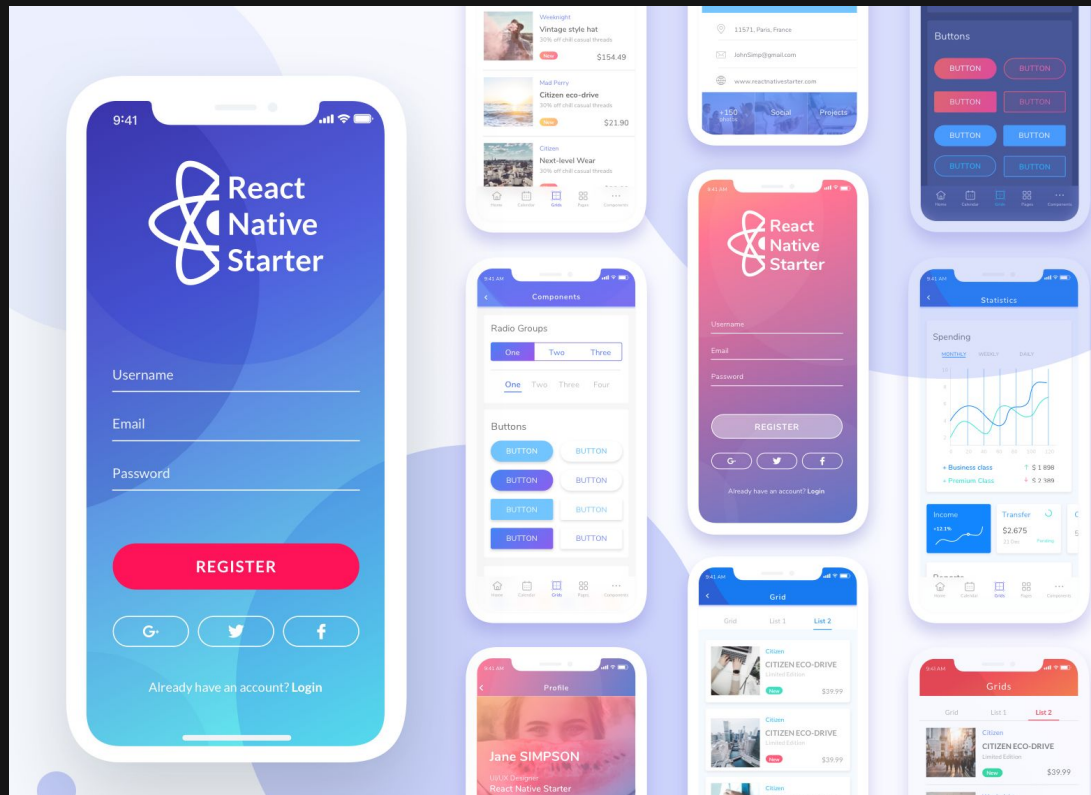
# Criar Componentes



# Gerenciar dados e estados



# Utilizar os conhecimentos para desenvolvimento mobile com React Native



**{primeiros passos}**



{ DOM }

Document Object Model

# { JSX }

Uma sintaxe de javascript com HTML que é utilizada para descrever como a UI deve ser

**permite  
incorporar  
expressões**

```
const name = "Josh Perez";  
const element = <h1>Hello, {name}</h1>  
  
ReactDOM.render(  
  element,  
  document.getElementById("root")  
)
```

**ser utilizada  
dentro de  
funções, em  
loops e ifs**

```
function getGreeting(user) {  
  if (user) {  
    return <h1>Hello, {formatName(user)}!</h1>  
  }  
  return <h1>Hello, Stranger.</h1>  
}
```

## receber atributos

```
const element = <img src={user.avatarUrl}></img>
```

```
const element = <div tabIndex="0"></div>
```

podem conter  
children

```
const element = (  
  <div>  
    <h1>Hello!</h1>  
    <h2>Good to see you here.</h2>  
  </div>  
)
```

# { Props }

São propriedades de um componente que herdam valores para renderizar nos componentes

```
function Welcome(props) {  
  return <h1>Hello, {props.name}</h1>;  
}
```

```
const element = <Welcome name="Faisal Arkan" />;  
ReactDOM.render(  
  element,  
  document.getElementById('root')  
);
```



**Props são somente para leitura!**

**O componente que recebe props nunca deve alterá-las**

{ revisão 1 }

# { Function Components }

Componentes que são simplesmente uma função Javascript.

```
function Welcome(props) {  
  return <h1>Hello, {props.name}</h1>;  
}
```

**mais fácil de testar**

**menor quantidade de código**

**melhoria de performance**

## { Class Components }

Componentes que estendem a classe `Component` do React.

Normalmente contém lógica, gerenciam estado e aceitam métodos do lifecycle do React.

```
class Person extends Component {  
  constructor(props) {  
    super(props);  
    this.state = {  
      myState: true  
    }  
  }  
  
  render() {  
    return (  
      <div>  
        <h1>Hello Person</h1>  
      </div>  
    );  
  }  
}
```

```
export default Person;
```

```
class Welcome extends React.Component {  
  render() {  
    return <h1>Hello, {this.props.name}</h1>;  
  }  
}
```

## { Class Components }

Componentes que estendem a classe  
Component do React.

Normalmente contém lógica, gerenciam  
estado e aceitam métodos do lifecycle do  
React.

## { State }

São propriedades com valores dinâmicos que podem ser alterados com o método `setState()`.

Elas são definidas no `constructor()` da classe de um componente e controladas por ele.



```
class Clock extends React.Component {
  constructor(props) {
    super(props);
    this.state = { date: new Date() };
  }

  render() {
    return (
      <div>
        <h1>Hello, world!</h1>
        <h2>It is {this.state.date.toLocaleTimeString()}.</h2>
      </div>
    );
  }
}

ReactDOM.render(
  <Clock />,
  document.getElementById('root')
);
```

A cada mudança de estado o componente vai renderizar novamente

**ATENÇÃO:**

**setState()** é um método assíncrono, ele agenda um update no estado do componente

<https://reactjs.org/docs/faq-state.html#why-doesnt-react-update-thisstate-synchronously>

## ERRADO:

```
this.setState({  
  counter: this.state.counter + this.props.increment,  
});
```

## CERTO:

```
this.setState((state, props) => ({  
  counter: state.counter + props.increment  
}));
```

## { Class Components }

Componentes que estendem a classe  
Component do React.

Normalmente contém lógica, gerenciam  
estado e aceitam métodos do lifecycle do  
React.

# { Ciclo de Vida }

Funções que são executadas em diferentes momentos da "vida" do componente:

<http://projects.wojtekmaj.pl/react-lifecycle-methods-diagram/>

# { Revisão 2 }

**Props x State**

**Function Component x Class Component**

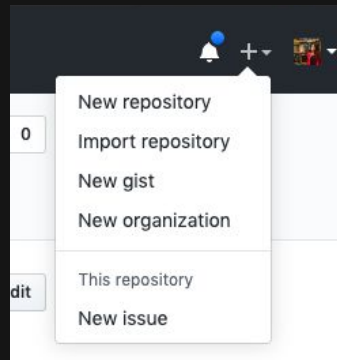
**Quais são os métodos de lifecycle?**

{ como? }

# { setup do projeto - no github }

criar um novo repositório no github

copiar o link do repositório





# { setup do projeto - no terminal }

abrir o terminal

instalar `npm install -g create-react-app`

navegar até onde você quer criar o projeto

`npx create-react-app first-react-app`

entrar na pasta `first-react-app`

# { setup do projeto - no terminal }

iniciar o repositório de git local

adicionar o .gitignore

colocar o node\_modules no gitignore

adicionar os arquivos

commitar os arquivos

adicionar o repositório remoto

mandar os arquivos do repositório local para o remoto

**{ setup do projeto - no terminal }**

**yarn start ou npm start**

# { thinking React }

<https://reactjs.org/docs/thinking-in-react.html>

# Começar com um mock

Imagine que já temos a API em JSON e o design da tela pelo nosso designer

☐ Only show products in stock

Name	Price
<b>Sporting Goods</b>	
Football	\$49.99
Baseball	\$9.99
<b>Basketball</b>	\$29.99
<b>Electronics</b>	
iPod Touch	\$99.99
<b>iPhone 5</b>	\$399.99
Nexus 7	\$199.99

# Os dados da nossa API são assim:



```
[  
  {category: "Sporting Goods", price: "$49.99", stocked: true, name: "Football"},  
  {category: "Sporting Goods", price: "$9.99", stocked: true, name: "Baseball"},  
  {category: "Sporting Goods", price: "$29.99", stocked: false, name: "Basketball"},  
  {category: "Electronics", price: "$99.99", stocked: true, name: "iPod Touch"},  
  {category: "Electronics", price: "$399.99", stocked: false, name: "iPhone 5"},  
  {category: "Electronics", price: "$199.99", stocked: true, name: "Nexus 7"}  
];
```

# 1 - Divida a UI em uma hierarquia de componentes

desenhe quadradinhos e dê nome à eles de acordo com os dados do seu JSON

Search...

☐ Only show products in stock

Name	Price
Sporting Goods	
Football	\$49.99
Baseball	\$9.99
Basketball	\$29.99
Electronics	
iPod Touch	\$99.99
iPhone 5	\$399.99
Nexus 7	\$199.99

## Dica:

Nomenclatura deve ser clara sobre a função do componente!

Será que esse componente vai ser reutilizado?

Nome dos componentes é escrito em PascalCase

Pensar na funcionalidade!



## Temos 5 componentes no nosso App



- **FilterableProductTable (laranja):** Contém a totalidade da tela
- **SearchBar (azul):** Recebe os dados do usuário
- **ProductTable (verde):** Mostra e filtra os resultados de acordo com o que for digitado no campo
- **ProductCategoryRow (turquesa):** Mostra o título de cada categoria
- **ProductRow (vermelho):** Mostra uma linha para cada produto

# Hierarquia

determinar a hierarquia dos componentes

**FilterableProductTable**

**SearchBar**

**ProductTable**

**ProductCategoryRow**

**ProductRow**

## 2 - Construa uma versão estática em React

### Dicas:

criar um arquivo de componentes para colocar todos seus componentes

não precisa ficar bonito! Desapegue das customizações usando CSS

as props devem vir do componente mais "externo" e ser passada para os childrens

### **3 - Identificar o mínimo de estado (valores dinâmicos) na UI**

**Comece pensando em todas as partes de dados ou informações que temos na aplicação**

☐ Only show products in stock

Name	Price
<b>Sporting Goods</b>	
Football	\$49.99
Baseball	\$9.99
<b>Basketball</b>	\$29.99
<b>Electronics</b>	
iPod Touch	\$99.99
<b>iPhone 5</b>	\$399.99
Nexus 7	\$199.99

☐ Only show products in stock

Name	Price
<b>Sporting Goods</b>	
Football	\$49.99
Baseball	\$9.99
<b>Basketball</b>	\$29.99

# Será que meu dado deveria ser um state?

1. É passado por um parent via props?
2. Ele permanece inalterado ao longo do tempo?
3. Você pode calculá-lo com base em qualquer outro estado ou adereços no seu componente?

☐ Only show products in stock

Name	Price
<b>Sporting Goods</b>	
Football	\$49.99
Baseball	\$9.99
<b>Basketball</b>	\$29.99
<b>Electronics</b>	
iPod Touch	\$99.99
<b>iPhone 5</b>	\$399.99
Nexus 7	\$199.99

☐ Only show products in stock

Name	Price
<b>Sporting Goods</b>	
Football	\$49.99
Baseball	\$9.99
<b>Basketball</b>	\$29.99

**vamos usar `state` para gerenciar**

`texto que o usuário digitou na busca`

`o valor do checkbox`

**Dica: usar estado somente quando necessário!**



## 4 - Identificar onde o seu estado deve ficar

1. Encontrar quais componentes renderizam com base nesse state
2. Qual componente tem esses componentes como children e também faz uso desse state?
3. Vale a pena criar um componente para guardar meu estado?

☐ Only show products in stock

Name	Price
Sporting Goods	
Football	\$49.99
Baseball	\$9.99
Basketball	\$29.99
Electronics	
iPod Touch	\$99.99
iPhone 5	\$399.99
Nexus 7	\$199.99

☐ Only show products in stock

Name	Price
Sporting Goods	
Football	\$49.99
Baseball	\$9.99
Basketball	\$29.99
Electronics	
iPod Touch	\$99.99
iPhone 5	\$399.99
Nexus 7	\$199.99

## adicionar **state** na aplicação

adicione o estado inicial da aplicação na `FilterableProductsTable` e passe as variáveis do estado como props para os componentes que precisam delas

**Dicas:** o estado inicial é um objeto que fica dentro do construtor

não importa se a informação vem do state, para o componente que está recebendo ela é uma prop!

**o que acontece se dermos um refresh na  
página?**

**o que acontece se mudarmos o valor  
inicial do filtro para "ball"?**

## 5 - Inverter o fluxo de dados

porque o React ignora o texto que estamos digitando na busca?

Para fazer com que a informação chegue do input na **FilterableProductTable** e altere seu estado precisamos usar **callbacks**

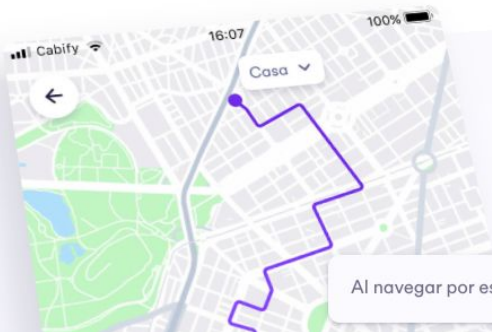
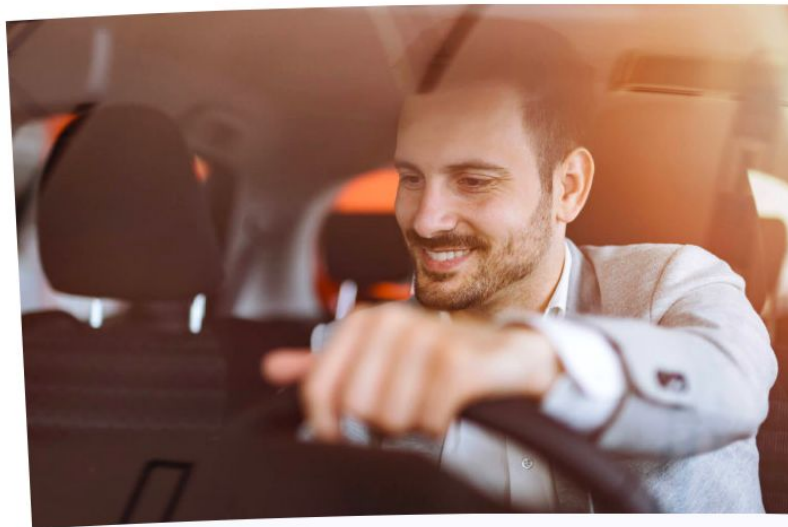
# bora praticar

- identificar quais componentes poderiam ser funções e transformá-los em functional components.
- customizar a aplicação, agora é hora de deixar bonito!
- praticar identificar os componentes dos sites

# Más seguridad, mayor calidad y una ciudad a tu alcance

Conviértete en conductor y empieza a ganar dinero.

Conduce con nosotros



¿POR QUÉ ELEGIR CABIFY?



## Tu seguridad es nuestra prioridad

Para que te muevas sin preocupaciones, todos nuestros viajes están geolocalizados y tanto los conductores como los pasajeros están identificados.

Al navegar por este sitio web, aceptas nuestra [política de cookies](#)







magalu

Nossas lojas Tenha sua loja Regulamentos Acessibilidade

Atendimento Compre no tel: (11) 3508-9900 Meus pedidos

procure por código, nome, marca...



Bem-vindo :)  
Entre ou cadastre-se



Todos os departamentos

Ofertas do dia

Celulares

Móveis

Eletrodomésticos

Tv e Vídeo

Informática

Saldão

Netshoes

Cartão Luiza

#magalu

Quem busca Magalu se dá bem.

magalu.com > ofertas do dia

## Ofertas do dia

### marcas

- ☐ samsung (15)
- ☐ lg (14)
- ☐ brastemp (11)
- ☐ consul (8)
- ☐ philco (7)

[veja todos](#)

### avaliação

- ☐ ★★★★★
- ☐ ★★★★☆
- ☐ ★★★☆☆
- ☐ ★★☆☆☆
- ☐ ★☆☆☆☆

As melhores ofertas e  
preços exclusivos

Digite o seu nome

Digite o seu email

**EU QUERO!**

125 produtos encontrados

Ordenar por: Relevância



Pneu Aro 13" Goodyear  
175/70R13 82T  
★★★★★ (12)  
de R\$ 266,00 por  
**R\$ 179,90**



Notebook Dell Inspiron i14-  
3481-M10S 7ª Geração Intel...  
★★★★★ (3)  
de R\$ 2.399,00 por  
**R\$ 2.260,05 à vista**



Liquidificador Mondial NL-26  
Copo 1,5L  
★★★★★ (74)  
de R\$ 79,90 por  
**R\$ 49,00**



Smart TV LED PRO 43" Full  
HD LG 43LM 631 3 HDMI 2...  
★★★★★ (20)  
de R\$ 1.999,00 por  
**R\$ 1.369,00**

[Premium](#)[Ajuda](#)[Baixar](#)[Inscrever-se](#)[Entrar](#)

# Música para todos.

Milhões de músicas à sua escolha. E nem precisa de cartão de crédito.

[OBTENHA O SPOTIFY FREE](#)[EMPRESA](#)[Sobre](#)[COMUNIDADES](#)[Para Artistas](#)[LINKS ÚTEIS](#)[Ajuda](#)

# referências

<https://medium.com/@Zwenza/functional-vs-class-components-in-react-231e3fbd7108>

<https://codepen.io/gaearon/pen/BwWzwm>

<https://reactjs.org/docs/thinking-in-react.html>