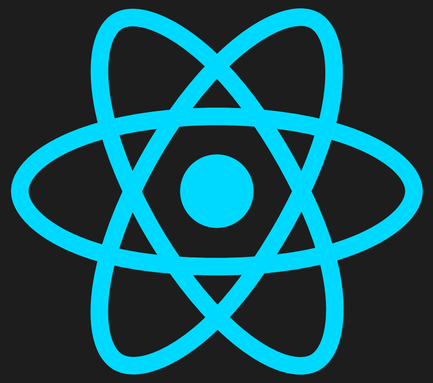


React

DESENVOLVEDOR MULTIPLATAFORMA

WEB & MOBILE

RECAPITULANDO

Propriedades

```
var nomeVariavel = “Pesquise aqui”;

<SearchField texto={nomeVariavel + “...”} />
```

Para acessar as propriedades do componente

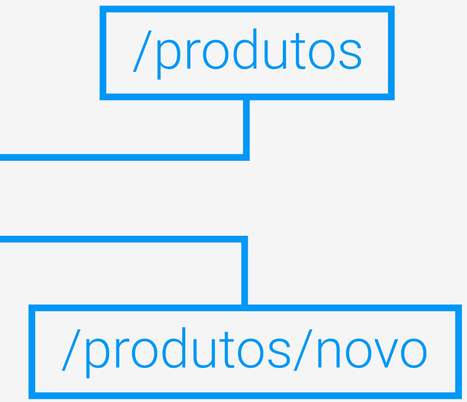
```
// SearchField

<input placeholder={this.props.texto} />
```

RECAPITULANDO

Rotas

```
<Router history={browserHistory} />
  <Route path=“/produtos” component={Produtos}>
    <IndexRoute component={ListaProdutos} />
    <Route path=“novo” component={NovoProduto} />
  <Route />
<Router />
```



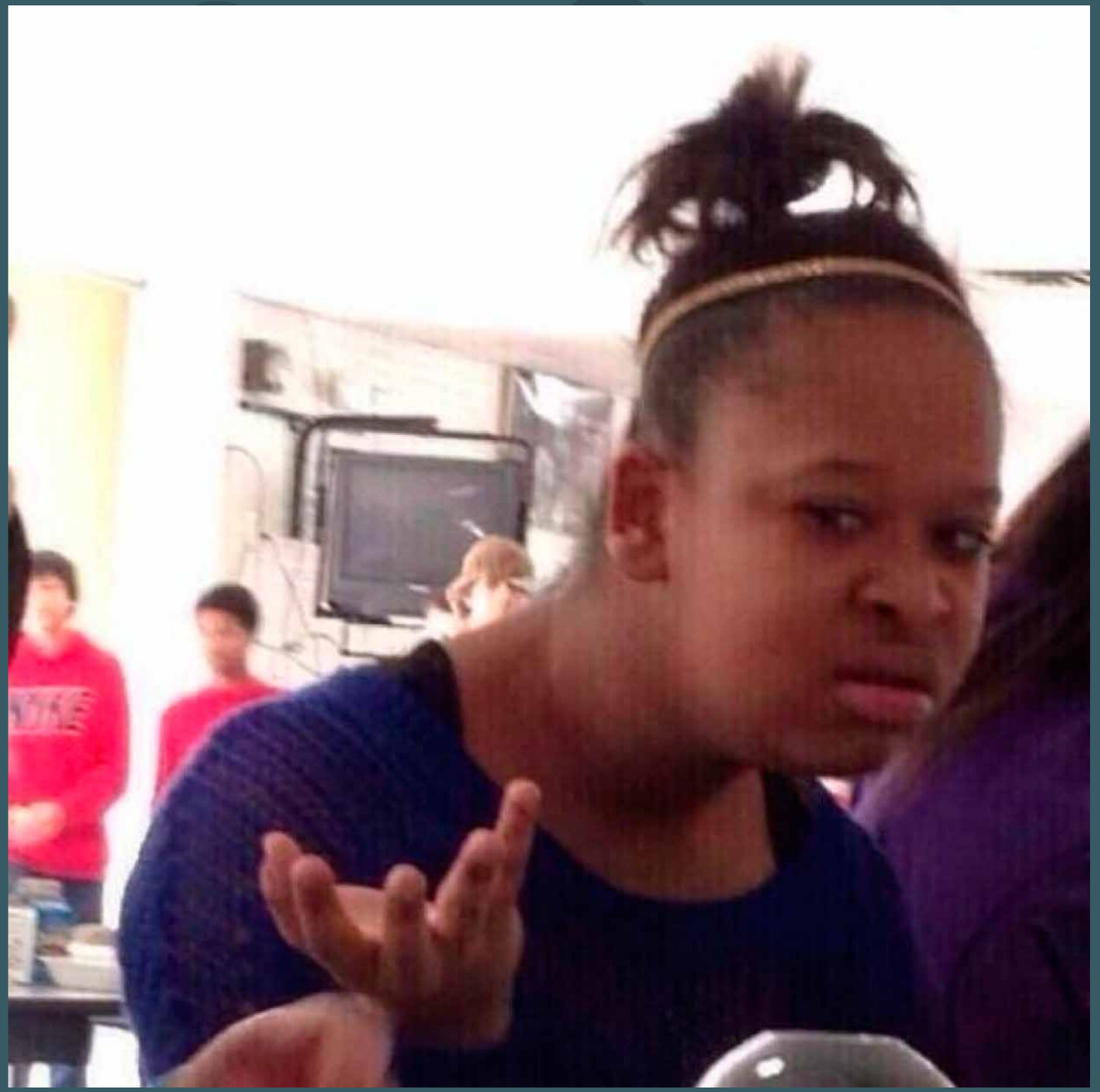

REST

apiary

INTEGRANDO O FRONT-END

FETCH

```
fetch(“url_da_api”)
  .then(response => response.json())
  .then(response => this.setState({
    propriedade: response
  }));
```



CICLO DE VIDA

- componentWillMount
- componentDidMount
- componentWillReceiveProps
- componentWillReceiveProps
- shouldComponentUpdate
- componentWillUpdate
- componentDidUpdate
- componentWillUnmount

https://facebook.github.io/react/docs/component-specs.html

PROMISES

São usadas para o processamento assincrono.
Estão sempre em um dos três estados:

pendente (pending):
estado inicial, ainda não concluido ou rejeitado.

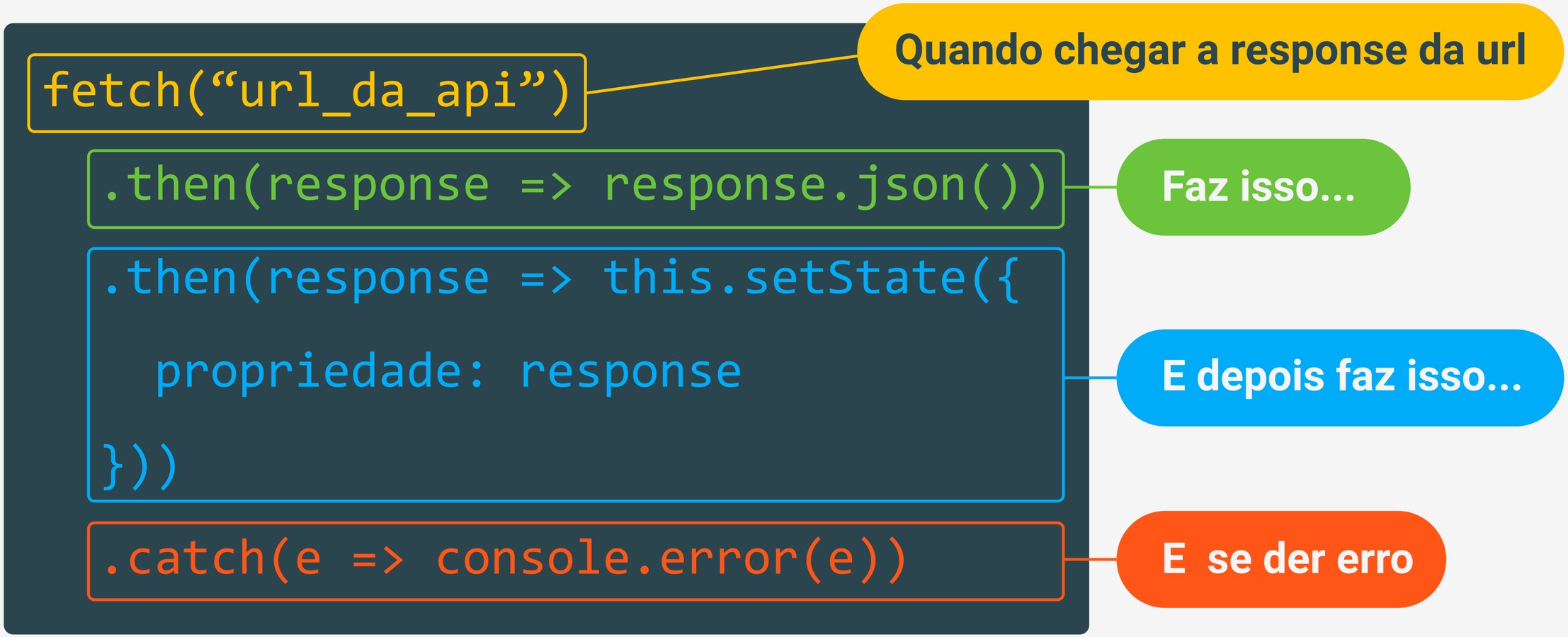
realizada (fulfilled):
sucesso na operação

rejeitado (rejected):
falha na operação

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Promise

PROMISES

Você pode encadear eventos no promise



ARROW FUNCTIONS

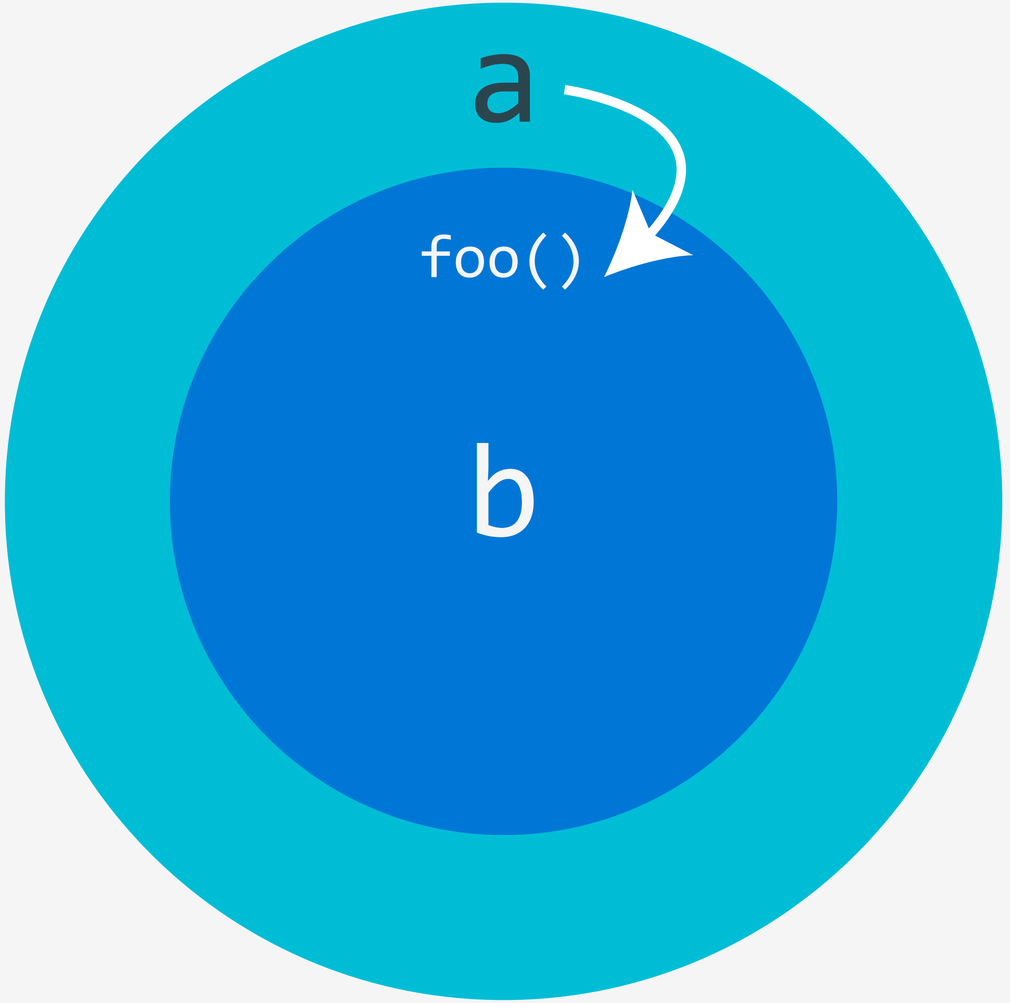
- Não criam um escopo separado para a função
- São sempre anônimas
- São uma forma abreviada para criar funções

ESCOPO JS

```
var a = 1;
function nomeFuncao() {
    console.log(a); // 1
    var b = 2;
}
console.log(b); // undefined
```

ESCOPO JS

```
var a = 1;
function foo() {
    console.log(a); // 1
    var b = 2;
}
console.log(b); // undefined
```



ARROW FUNCTIONS

```
function nomeFuncao() {


}
```

=

```
var nomeFuncao = () => {


}
```

ARROW FUNCTIONS

SÃO SEMPRE ANÔNIMAS

```
nomeFuncao() => {
    ...
});
```

```
var nomeFuncao = () => {
    ...
}
```

ARROW FUNCTIONS

```
.forEach(function(foo) {
    ...
});
```

=

```
.forEach((foo) => {
    ...
});
```

ARROW FUNCTIONS

```
.map(function(foo) {
    return foo + 1;
});
```

=

```
.map(foo => foo + 1);
```

Parênteses são opcionais caso só tenha um parâmetro na função

STATES

STATES

São objetos no JS e disparam atualizações na renderização

Podem ser definidos quando o componente é criado

Pode sofrer mudanças dentro do componente

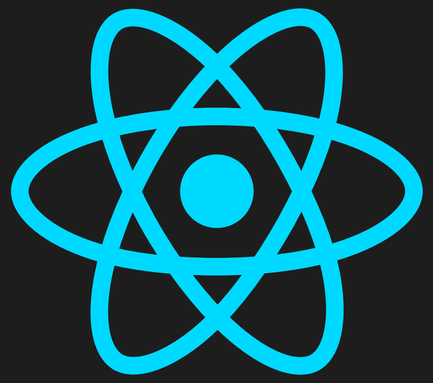
VS

PROPS

São objetos no JS e disparam atualizações na renderização

São as “configurações” do seu componente

Só é mudado pelo componente pai, nunca por ele mesmo



React

Obrigado!

DESENVOLVEDOR MULTIPLATAFORMA

WEB & MOBILE