



**Rafael Ribeiro**  
Head de Engenharia

# Introdução à Promises

Módulo X



Qual o problema em utilizar callbacks?

## Callback hell

```
1 asyncFunction('exemplo', function(erro, dados) {
2   if(erro) { console.error('erro:', erro); return; }
3   asyncFunction('exemplo', function(erro, dados) {
4     if(erro) { console.error('erro:', erro); return; }
5     asyncFunction('exemplo', function(erro, dados) {
6       if(erro) { console.error('erro:', erro); return; }
7       asyncFunction('exemplo', function(erro, dados) {
8         if(erro) { console.error('erro:', erro); return; }
9         asyncFunction('exemplo', function(erro, dados) {
10          if(erro) { console.error('erro:', erro); return; }
11          asyncFunction('exemplo', function(erro, dados) {
12            if(erro) { console.error('erro:', erro); return; }
13          });
14        });
15      });
16    });
17  });
18 });
```



## Paralelismo



```
1  readFile('file1.json', function(erro1, dados1) {  
2      readFile('file2.json', function(erro2, dados2) {  
3          console.log(dados1);  
4          console.log(dados2);  
5      });  
6  });
```

## Paralelismo



```
1 let dados1 = null, dados2 = null;
2
3 readFile('file1.json', function(error, dados) {
4     dados1 = dados;
5     exhibir();
6 });
7
8 readFile('file2.json', function(error, dados) {
9     dados2 = dados;
10    exhibir();
11 });
12
13 function exhibir() {
14     if(dados1 && dados2) {
15         console.log(dados1);
16         console.log(dados2);
17     }
18 }
```



## Promises

É um objeto para trabalhar com a assincronismo em  
javascript



## Por que Promises?

A questão é mais a organização do código do que a funcionalidade



# Ciclo de Vida das Promises

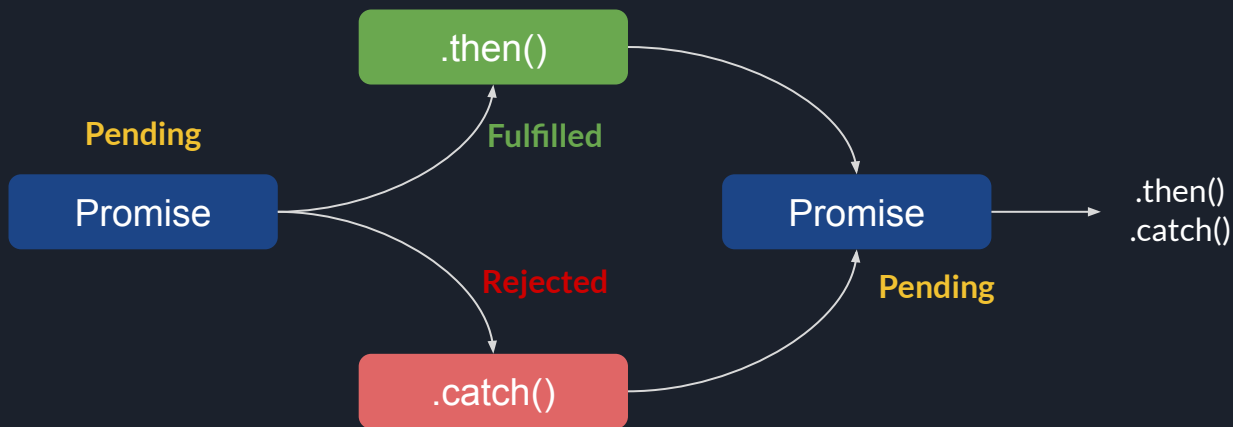
**Pending**: estado inicial

**Fulfilled**: executou com sucesso

**Rejected**: a operação falhou



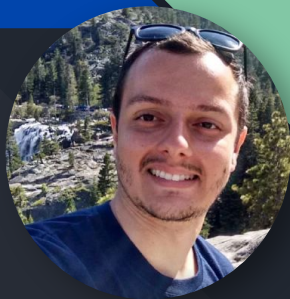
# Ciclo de Vida das Promises





<> Bora para cima!!!! </>

#techacademy #mecomprometo



**Rafael Ribeiro**  
Head de Engenharia

# Introdução à Promises

Módulo X