

JavaScript:

Consumindo e tratando dados de uma API



alura

Assincronicidade

Aula 2

alura

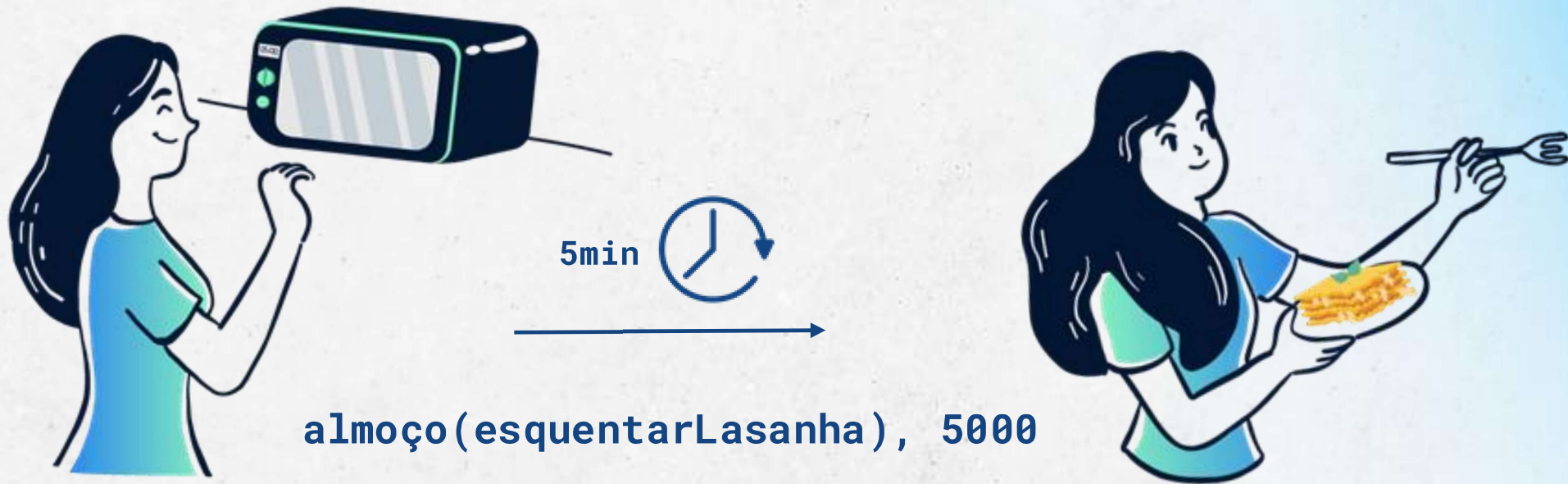
Callbacks

Aula 2.1



alura

Eu estava com fome e coloquei uma lasanha no microondas por 5 minutos. Após esse tempo, eu farei uma ação, que nesse caso vai ser a de comer. Esse processo de espera é algo que podemos visualizar no javascript com os callbacks, que podem ser interpretados como ações que você faz que irão resultar em outra ação.



`almoço(esquentarLasanha), 5000`

`almoço(esquentarLasanha)`

`almoço()`

O callback é uma função que é passada como argumento para uma outra função (higher-order function) e, geralmente, só é executada quando alguma operação é concluída ou quando um evento específico ocorre. O cenário de uso mais comum é, certamente, quando as requisições são finalizadas.

almoço(esquentarLasanha, 5000)

(Callback)

(Higher-order
function)

almoço(**esquentarLasanha**
)

almoço()

Fetch API

Aula 2.2



alura

Ao iniciarmos o trabalho, nos deparamos com um problema entre os dados de endereço dos usuários da AluraBooks



Os endereços estavam despadronizados:

Cliente 1	Rua Getúlio Vargas, número 1520
Cliente 2	R. G. Vargas, 1520
Cliente 3	R. Getúlio Vargas, 1520
Cliente 4	G. Vargas - n 1520

Como vamos resolver essa
situação?





Vamos usar uma API chamada [ViaCEP](#).

Ao entrar no site conseguimos mais informações de como usá-la, que é através de um link que contém o número do cep do cliente:

...

viacep.com.br/ws/01001000/json/

...

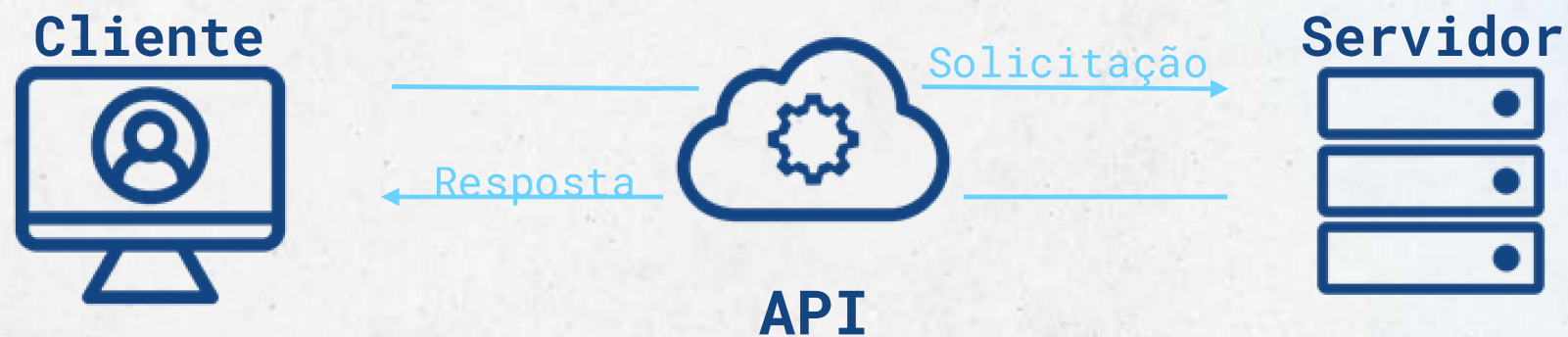
API

Interface de Programação de Aplicações

Mecanismos que permitem que dois componentes de software se comuniquem usando um conjunto de definições e protocolos.

API

Interface de Programação de Aplicações



API

Interface de Programação de Aplicações

Cliente



Solicitação



Pagar boleto

Resposta

Servidor



API

Interface de Programação de Aplicações

Cliente



Solicitação



API

Resposta

Servidor



API

Interface de Programação de Aplicações

Cliente



Requisição de CEP



API

Resposta com dados

Servidor



VIA CEP

Promises

Aula 2.3



alura

Promise

Promise

Definem uma ação que vai ser executada no futuro e ela pode ser resolvida (com sucesso) ou rejeitada (com erro).

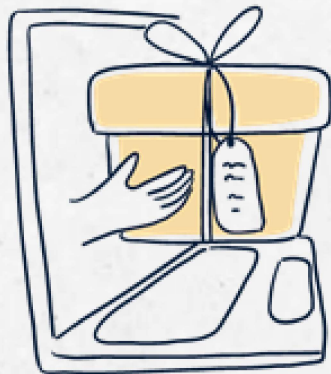


Resolvida



Rejeitada

Promise



Imagine que você recebeu a notificação de que sua encomenda saiu para entrega. A notificação foi uma promessa feita pela transportadora.



Anatomia da Promise

```
...  
const entrega = new Promise(function  
(resolve, reject) {  
  if (recebeu == true) {  
    resolve('Ana recebeu a encomenda!')  
  } else {  
    reject('Não foi possível receber a  
encomenda!')  
  }  
})  
...
```

Promise

