

JavaScript: Consumindo e tratando dados de uma API



alura

Como funciona o JavaScript?

Aula 1



alura

**Nesse curso, seremos
desenvolvedores Front-end
aqui da Alura, especificamente
trabalharemos com Javascript.**



Importando arquivos .js

Aula 1.2

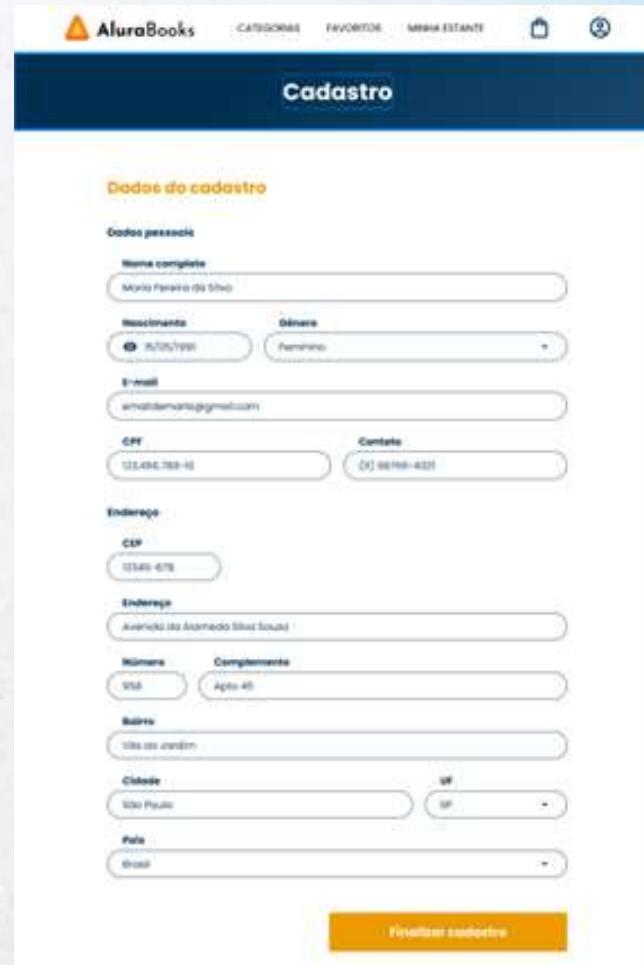


alura

O primeiro passo é criar e importar o arquivo javascript.

Vamos abrir e testar pelo navegador através da ferramenta do desenvolvedor (f12).

Usaremos a página de cadastro do site Alura Books.



Síncrono e assíncrono?

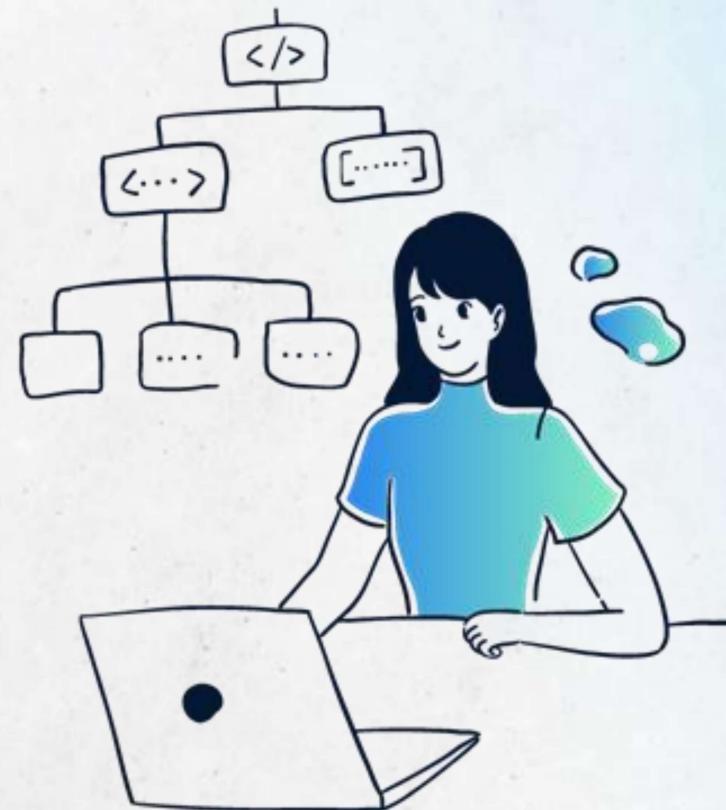
Aula 1.3



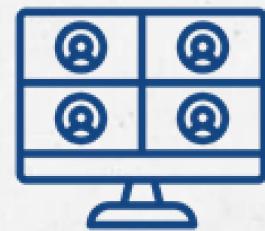
alura

Síncrono ou assíncrono

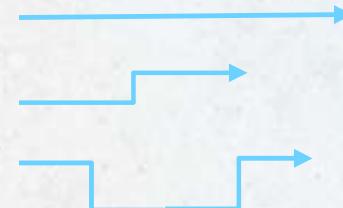
- Tendência de trabalho home office
- Comunicação pode ser por videochamada (**síncrona**) ou via chat (**assíncrona**)



Síncrono ou assíncrono



Videochamadas

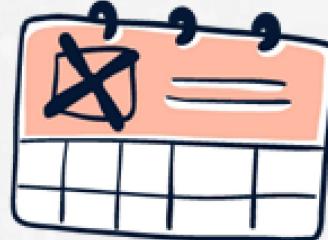


Mensagens de texto

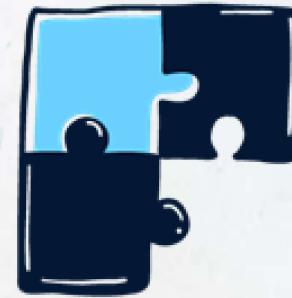
**Quais são os caminhos que você pode seguir enquanto
aguarda a resposta de um colega?**



**Faltar seus
compromissos
pessoais**



ou



**Se ocupar em
outras tarefas até
obter resposta**



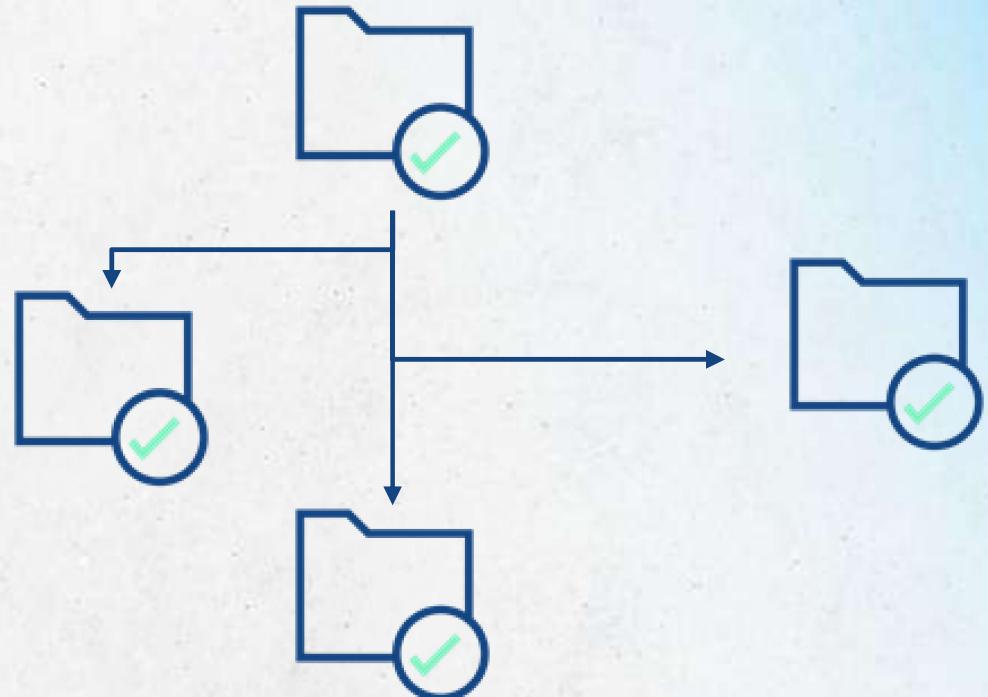
Isso é equivalente as decisões que podemos tomar em um sistema:

Síncrono



Uma tarefa é concluída após a outra (padrão do JavaScript)

Assíncrono



Há tarefas executadas em "segundo plano"

Event Loop e Call Stack

Aula 1.4



alura

Task Queue



Event Loop

```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
  console.log("Tudo bem?");
  console.log("Vou te mandar uma solicitação!");
  console.log("Solicitação recebida!");
}
mandaMensagem();
console.log("Tchau tchau!")
```

Call Stack

Podemos
destrinchar o fluxo
de execução de
tarefas em
JavaScript em três
partes: **Event Loop**,
Call Stack e **Task
Queue**.

Task Queue



Event Loop

Call Stack

O Event Loop é um ciclo que monitora e executa as ações que mandamos para o JavaScript. O processo de leitura do código só é finalizado quando não existem mais ações a serem executadas.

```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
    console.log("Tudo bem?");
    console.log("Vou te mandar uma solicitação!");
    console.log("Solicitação recebida!");
}
mandaMensagem();
console.log("Tchau tchau!")
```

Task Queue



Event Loop

Call Stack

A pilha de chamadas (call stack) é um mecanismo que organiza como irá funcionar o script quando existem muitas funções: qual função está sendo executada, quais estão sendo chamadas dentro de alguma função, etc.

```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
    console.log("Tudo bem?");
    console.log("Vou te mandar uma solicitação!");
    console.log("Solicitação recebida!");
}
mandaMensagem();
console.log("Tchau tchau!")
```

Task Queue

Call Stack



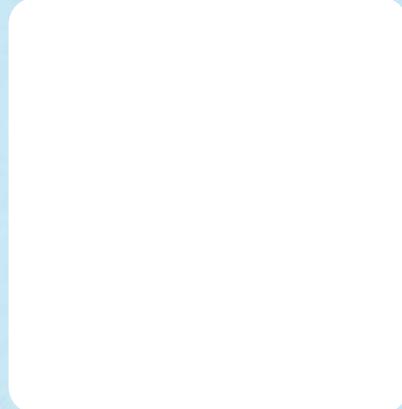
Event Loop

```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
    console.log("Tudo bem?");
    console.log("Vou te mandar uma solicitação!");
    console.log("Solicitação recebida!");
}
mandaMensagem();
console.log("Tchau tchau!")
```

A task queue é a fila de tarefas assíncronas. Se algo precisa ocorrer em segundo plano ou mais tarde, é nessa fila que ele será adicionado e executado mais tarde.

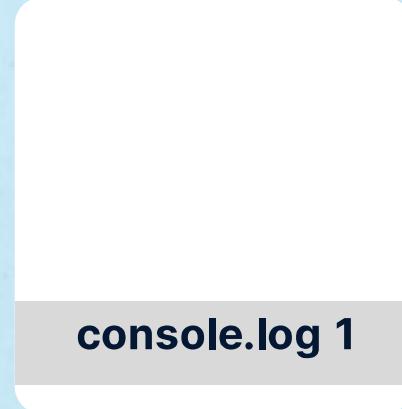
Exemplo síncrono

Task Queue



Event Loop

Call Stack



```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
    console.log("Tudo bem?");
    console.log("Vou te mandar uma solicitação!");
    console.log("Solicitação recebida!");
}
mandaMensagem();
console.log("Tchau tchau!")
```

Task Queue

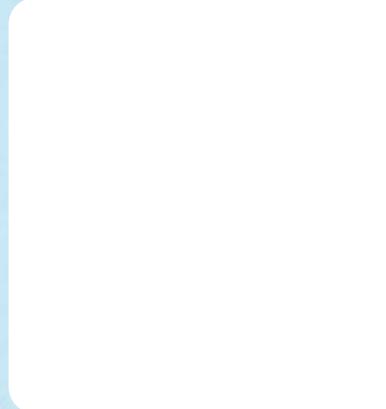
Call Stack



Event Loop

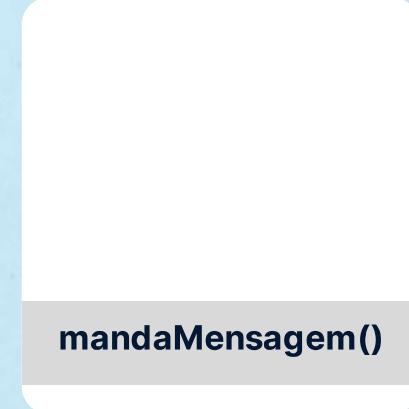
```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
    console.log("Tudo bem?");
    console.log("Vou te mandar uma solicitação!");
    console.log("Solicitação recebida!");
}
mandaMensagem();
console.log("Tchau tchau!")
```

Task Queue



Event Loop

Call Stack



```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
    console.log("Tudo bem?");
    console.log("Vou te mandar uma solicitação!");
    console.log("Solicitação recebida!");
}
mandaMensagem();
console.log("Tchau tchau!")
```

Task Queue



Event Loop

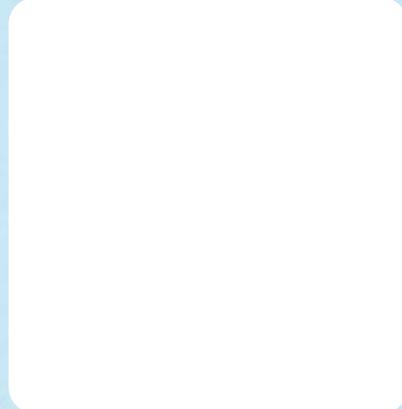
Call Stack

console.log 1

mandaMensagem()

```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
    console.log("Tudo bem?");
    console.log("Vou te mandar uma solicitação!");
    console.log("Solicitação recebida!");
}
mandaMensagem();
console.log("Tchau tchau!")
```

Task Queue



Event Loop

Call Stack



```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
    console.log("Tudo bem?");
    console.log("Vou te mandar uma solicitação!");
    console.log("Solicitação recebida!");
}
mandaMensagem();
console.log("Tchau tchau!")
```

Task Queue



Event Loop

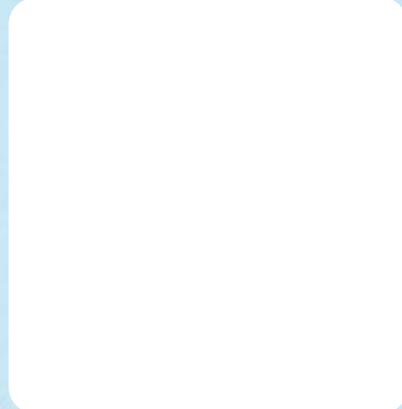
Call Stack

console.log 2

mandaMensagem()

```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
    console.log("Tudo bem?");
    console.log("Vou te mandar uma solicitação!");
    console.log("Solicitação recebida!");
}
mandaMensagem();
console.log("Tchau tchau!")
```

Task Queue



Event Loop

Call Stack



```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
    console.log("Tudo bem?");
    console.log("Vou te mandar uma solicitação!");
    console.log("Solicitação recebida!");
}
mandaMensagem();
console.log("Tchau tchau!")
```

Task Queue



Event Loop

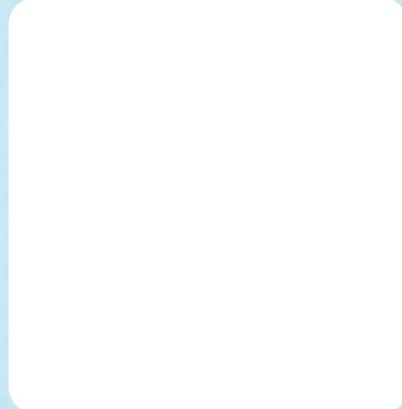
Call Stack

console.log 3

mandaMensagem()

```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
    console.log("Tudo bem?");
    console.log("Vou te mandar uma solicitação!");
    console.log("Solicitação recebida!");
}
mandaMensagem();
console.log("Tchau tchau!")
```

Task Queue



Event Loop

Call Stack



```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
    console.log("Tudo bem?");
    console.log("Vou te mandar uma solicitação!");
    console.log("Solicitação recebida!");
}
mandaMensagem();
console.log("Tchau tchau!")
```

Task Queue

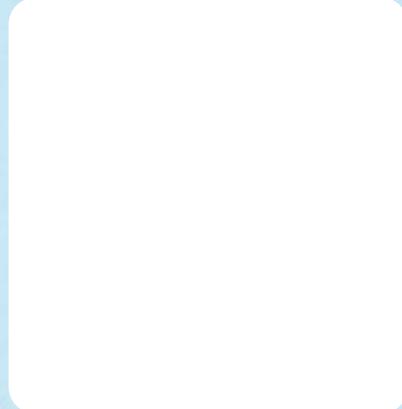
Call Stack



Event Loop

```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
    console.log("Tudo bem?");
    console.log("Vou te mandar uma solicitação!");
    console.log("Solicitação recebida!");
}
mandaMensagem();
console.log("Tchau tchau!")
```

Task Queue



Event Loop

Call Stack



```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
    console.log("Tudo bem?");
    console.log("Vou te mandar uma solicitação!");
    console.log("Solicitação recebida!");
}
mandaMensagem();
console.log("Tchau tchau!")
```

Task Queue

Call Stack



Event Loop

```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
    console.log("Tudo bem?");
    console.log("Vou te mandar uma solicitação!");
    console.log("Solicitação recebida!");
}
mandaMensagem();
console.log("Tchau tchau!")
```

Exemplo assíncrono

Task Queue

Call Stack



Event Loop

```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
    console.log("Tudo bem?");
    console.log("Vou te mandar uma solicitação!");
    console.log("Solicitação recebida!");
}
setTimeOut(mandaMensagem, 5000);
console.log("Tchau tchau!")
```

Task Queue



Event Loop

Call Stack



```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
    console.log("Tudo bem?");
    console.log("Vou te mandar uma solicitação!");
    console.log("Solicitação recebida!");
}
setTimeOut(mandaMensagem, 5000);
console.log("Tchau tchau!")
```

Task Queue

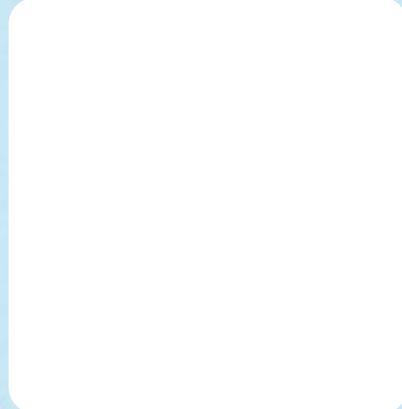
Call Stack



Event Loop

```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
    console.log("Tudo bem?");
    console.log("Vou te mandar uma solicitação!");
    console.log("Solicitação recebida!");
}
setTimeOut(mandaMensagem, 5000);
console.log("Tchau tchau!")
```

Task Queue



Event Loop

Call Stack



setTimeout() de 5s

```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
    console.log("Tudo bem?");
    console.log("Vou te mandar uma solicitação!");
    console.log("Solicitação recebida!");
}
setTimeout(mandaMensagem, 5000);
console.log("Tchau tchau!")
```

Task Queue

setTimeout() de 5s



Event Loop

Call Stack

```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
    console.log("Tudo bem?");
    console.log("Vou te mandar uma solicitação!");
    console.log("Solicitação recebida!");
}
setTimeout(mandaMensagem, 5000);
console.log("Tchau tchau!")
```

Task Queue

setTimeout() de 5s



Event Loop

Call Stack

console.log 4

```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
    console.log("Tudo bem?");
    console.log("Vou te mandar uma solicitação!");
    console.log("Solicitação recebida!");
}
setTimeout(mandaMensagem, 5000);
console.log("Tchau tchau!")
```

Task Queue

setTimeout() de 5s

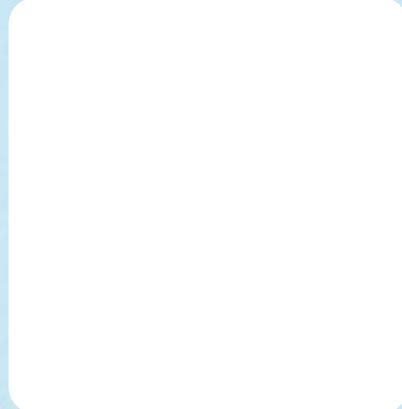
Call Stack



Event Loop

```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
    console.log("Tudo bem?");
    console.log("Vou te mandar uma solicitação!");
    console.log("Solicitação recebida!");
}
setTimeout(mandaMensagem, 5000);
console.log("Tchau tchau!")
```

Task Queue



Event Loop

Call Stack



```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
    console.log("Tudo bem?");
    console.log("Vou te mandar uma solicitação!");
    console.log("Solicitação recebida!");
}
setTimeOut(mandaMensagem, 5000);
console.log("Tchau tchau!")
```

Task Queue



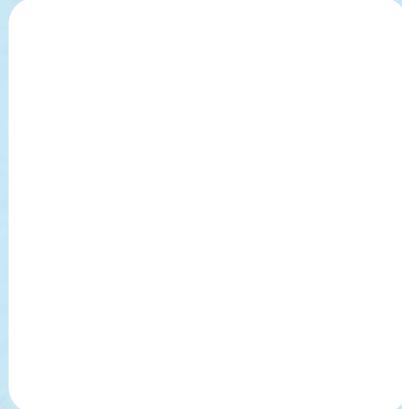
Event Loop

Call Stack

```
console.log 1  
trocaMensagem()
```

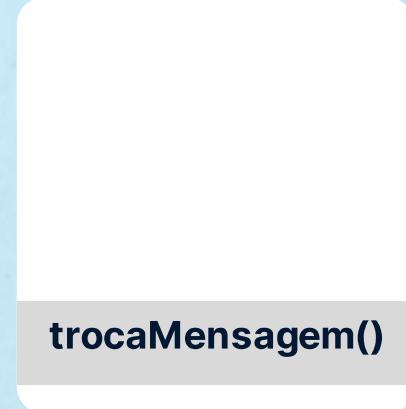
```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
    console.log("Tudo bem?");
    console.log("Vou te mandar uma solicitação!");
    console.log("Solicitação recebida!");
}
setTimeOut(mandaMensagem, 5000);
console.log("Tchau tchau!")
```

Task Queue



Event Loop

Call Stack



```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
    console.log("Tudo bem?");
    console.log("Vou te mandar uma solicitação!");
    console.log("Solicitação recebida!");
}
setTimeOut(mandaMensagem, 5000);
console.log("Tchau tchau!")
```

Task Queue



Event Loop

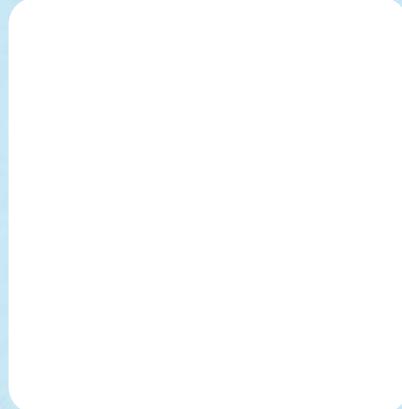
Call Stack

console.log 2

trocaMensagem()

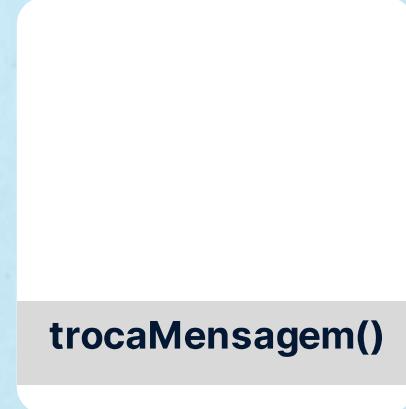
```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
    console.log("Tudo bem?");
    console.log("Vou te mandar uma solicitação!");
    console.log("Solicitação recebida!");
}
setTimeOut(mandaMensagem, 5000);
console.log("Tchau tchau!")
```

Task Queue



Event Loop

Call Stack



```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
    console.log("Tudo bem?");
    console.log("Vou te mandar uma solicitação!");
    console.log("Solicitação recebida!");
}
setTimeOut(mandaMensagem, 5000);
console.log("Tchau tchau!")
```

Task Queue



Event Loop

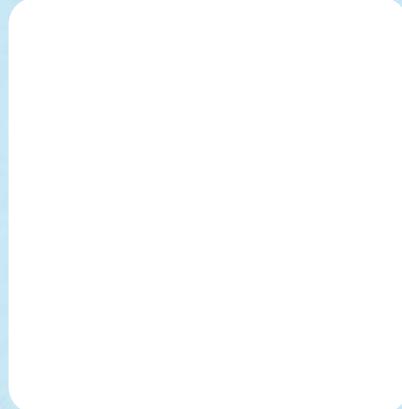
Call Stack

console.log 3

trocaMensagem()

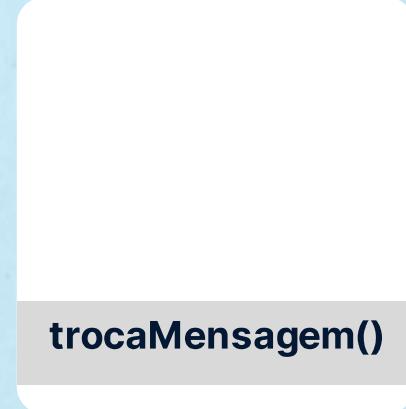
```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
    console.log("Tudo bem?");
    console.log("Vou te mandar uma solicitação!");
    console.log("Solicitação recebida!");
}
setTimeOut(mandaMensagem, 5000);
console.log("Tchau tchau!")
```

Task Queue



Event Loop

Call Stack



```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
    console.log("Tudo bem?");
    console.log("Vou te mandar uma solicitação!");
    console.log("Solicitação recebida!");
}
setTimeOut(mandaMensagem, 5000);
console.log("Tchau tchau!")
```

Task Queue

Call Stack



Event Loop

```
console.log("Mandando oi pro amigo!");
function mandaMensagem() {
    console.log("Tudo bem?");
    console.log("Vou te mandar uma solicitação!");
    console.log("Solicitação recebida!");
}
setTimeOut(mandaMensagem, 5000);
console.log("Tchau tchau!")
```