

# Condicionalis

## Desenvolver Algoritmos

Msc. Lucas G. F. Alves  
e-mail: [lgfalves@senacrs.com.br](mailto:lgfalves@senacrs.com.br)



# Planejamento de Aula

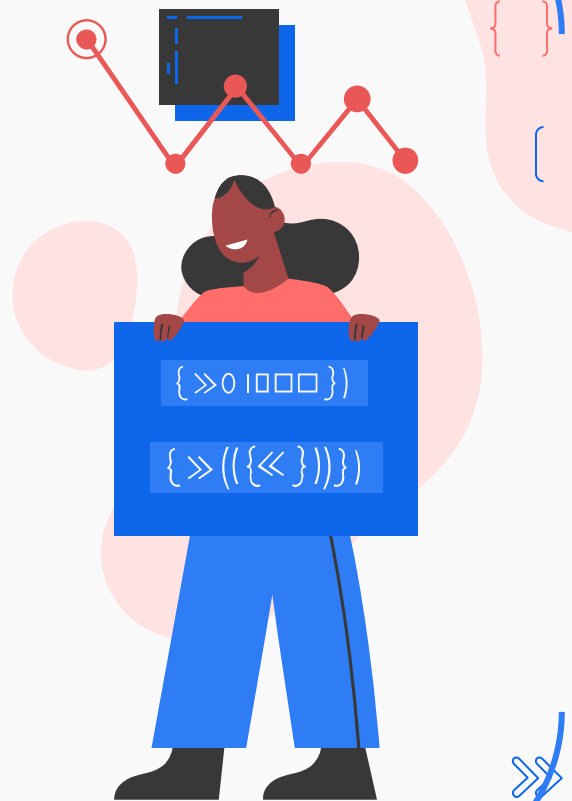
Revisão Condicionais

Revisão Árvores de Condicionais

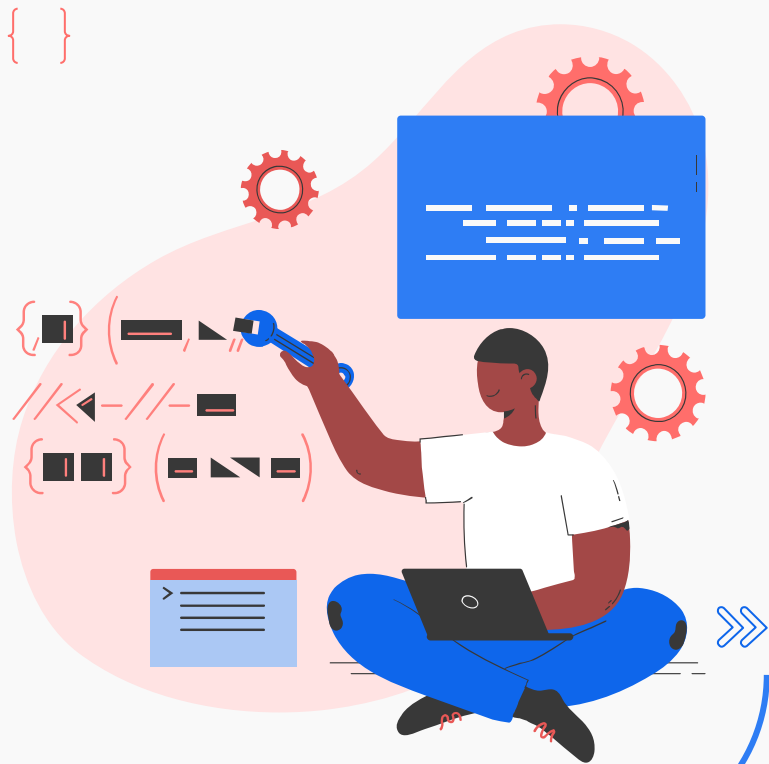
Revisão if else

Revisão switch case

Exercícios de Fixação



# Revisão Condicionais





# Condicionais

[ ]

O que são?

Vocês já repararam quantas vezes ao dia temos que tomar uma decisão **dependendo de outros fatores?**

Nossos cérebros são incríveis e fazem isso de forma automática!

Vamos tentar pegar alguns exemplos e quebrar o processo de raciocínio em pedacinhos:

{ }

{((( {>> })))<< }

- [ ]



# Condicionais

[ ]

O que são?

Eu abro as cortinas para ver o dia lá fora...

- **Se** está chovendo:
  - Saio correndo para recolher as roupas do varal;
- **Senão**:
  - Fico de boa aproveitando o dia bonito;

{ }

{((( {>> } ))<< }

- [ ]



# Condicionais

[ ]

O que são?

Testando um exercício da lista...

- Se passa no teste:
  - Fico feliz e vou pro próximo;
- Senão:
  - Choro e volto para achar o bug ;

{ }

{((({>>}))<<}

-[ ]



# Condicionais

[ ]

O que são?

Estou ficando com fome e resolvo abrir a geladeira...

- **Se** tem alguma coisa além de gelo e ketchup:
  - Fico feliz por ter sido responsável;
  - Preparo um almoço gostoso;
- **Senão:**
  - Peço um hambúrguer por aplicativos ;

{ }

{((({>>}))<<}

- [ ]



# Condicionais

[ ]

O que são?

Condicionais são **estruturas de código** usadas para **fazer escolhas** baseadas em alguns critérios.

Em outras palavras, elas permitem realizar uma determinada **ação** dependendo de uma **condição**.

{ }

**Exemplo:** baseado na **condição** de estar chovendo eu vou realizar a **ação** de recolher a roupa.

{((( {>>} ))<<}

- [ ]





# Condicionais

[ ]

Então

Até agora vimos que o javascript executa linha por linha de código, de forma síncrona e **sequencial**.

Como uma escada, que descemos degrau por degrau, sem poder pular nenhum.

{ }

{((({>>}))<<}

-{ }

Início do código



Etapa 1



Etapa 2

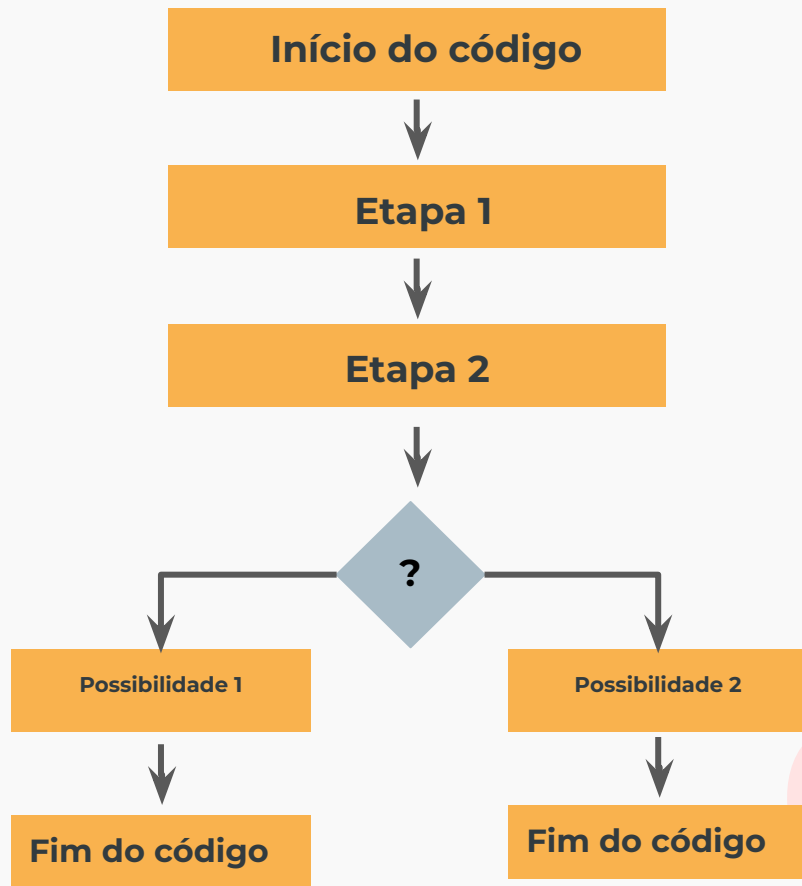


Fim do código

# Condicionais

Então

As condicionais são **estruturas de código** que nos dão o poder de **decidir** se a próxima linha de código deve ser executada ou não.





# Condicionais

[ ]

## Definição de estrutura

Modo como alguma coisa é **construída**, **organizada** ou está disposta: a estrutura de uma empresa.

Aquilo que serve de **base** para algo; armação ou esqueleto: a estrutura de um edifício; a estrutura de uma linguagem de programação.

{ }

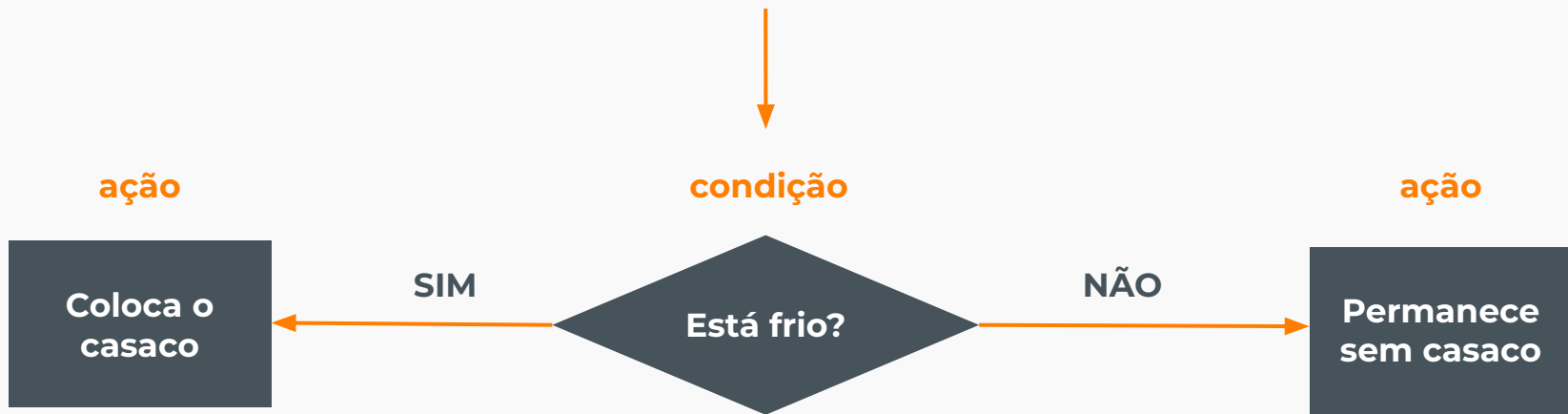
{((({>>}))<<}

-[ ]

# Árvores de Condicionais

Como representar?

Uma maneira esquemática de representar condicionais é utilizando fluxogramas:



{((({>>}))<<)}

-{ }



# Árvores de Condicionais

[ ]

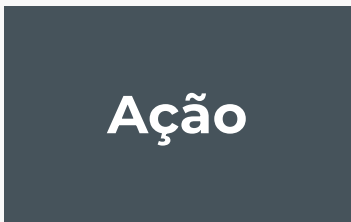
Definição de estrutura

Nos fluxogramas:

Um losango representa uma **condição** para a **tomada de decisão**.

Um retângulo representa a **ação**.

{ }



{ ((({ >> } )) << ) }

- [ ]



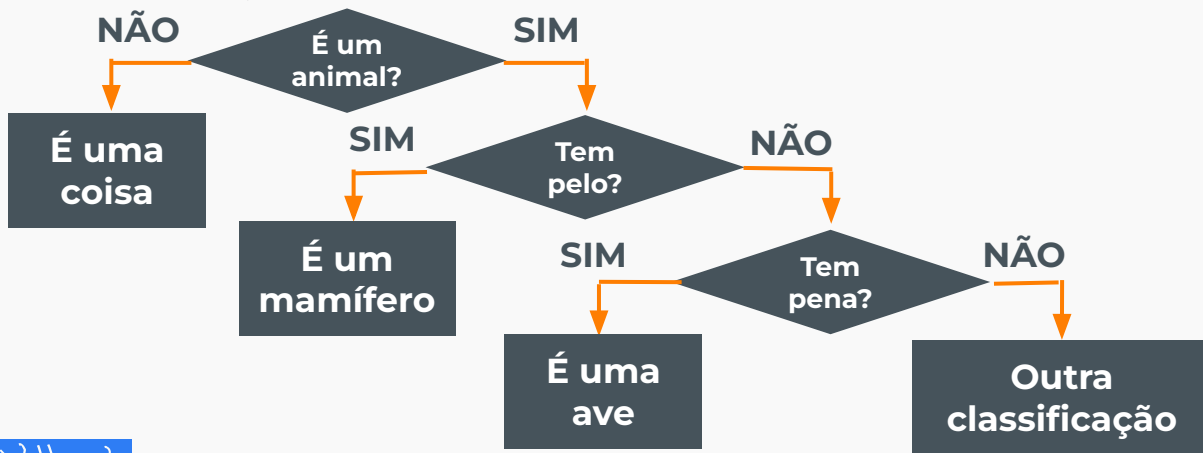
# Árvores de Condicionais



[ ]

## Definição de estrutura

Você pode **encadear** várias condições antes de chegar a uma resposta (por isso chamamos de árvore!)



{ }

{((( {>> } ))<< )}

- [ ]

# Árvores de Condicionais

## Definição de estrutura

Depois de percorrer várias condições, você chega à uma conclusão

E, a partir da conclusão, você pode realizar alguma ação



{((({>>}))<<}

# Árvores de Condicionais

## Definição de estrutura

Depois de percorrer várias condições, você chega à uma conclusão

E, a partir da conclusão, você pode realizar alguma ação



{((({>>}))<<}





# Resumo Condicionais

[ ]

Comparadores são operadores usados para **comparar variáveis** (==, !=, >, <).

**Condicional** é uma estrutura que permite escolher uma **ação dependendo de uma condição**.

**Árvore de condicionais** são uma maneira de se **representar esquematicamente** os condicionais.

{ }

{((({>>}))<<}

-[ ]



# Condicionais em JS

[ ]

Bloco if / else

if / else é a **sintaxe (estrutura)** de programação utilizada para **condicionais**.

Se a **condição for verdadeira** o código **dentro do if** é executado.

{ }

{ ((({ >> }))) << }

- [ ]



# Condicionais em JS

[ ]

if

Todo código da ação vai entre chaves { }

condição simples

```
1  let condicao1 = true
2
3  if (condicao1){
4      // Como o valor da condição é true,
5      // o código desse bloco é executado
6      console.log('Entrei no if 1!')
7  }
```

```
1  let condicao2 = false
2
3  if (condicao2){
4      // Como o valor da condição é false,
5      // o código desse bloco NÃO é executado
6      console.log('Entrei no if 2!')
7  }
```

{((({>>}))<<}

-[ ]



# Exercícios

[ ]

1. Crie um programa que:

a) Recebe 2 números (chamaremos de num1 e num2)

```
const num1 = Number(prompt("Digite o número 1"))
```

```
const num2 = Number(prompt("Digite o número 2"))
```

b) Compara esses números entre si:

```
num1 === num2
```

c) Se os números **forem iguais**, imprime uma mensagem de sucesso

```
if(num1 === num2){
```

```
  console.log("Sucesso")
```

```
}
```

{ }

```
({{{({>>}}))<<}
```

- [ ]



# Condicionais em JS

[ ]

If + else

Todo código da ação vai entre chaves { }

{ }

( { ( ( { >> } ) ) } << )

```
1  let condicao = false
2
3  if (condicao){
4      console.log('Entrei no if!')
5  } else {
6      // Como o valor da condição é false,
7      // o código do bloco else será executado
8      console.log('Entrei no else!')
9  }
```

[ ]



# Exercícios

[ ]

1. Crie um programa que:

a) Recebe 2 números (chamaremos de num1 e num2)

```
const num1 = Number(prompt("Digite o número 1"))
```

```
const num2 = Number(prompt("Digite o número 2"))
```

b) Compara esses números entre si:

```
num1 === num2
```

c) Imprime mensagens dizendo se os números são iguais ou diferentes.

{ }

```
if(num1 === num2){  
    console.log("Os números são iguais!")  
}  
else{  
    console.log("Os números são diferentes!")  
}
```

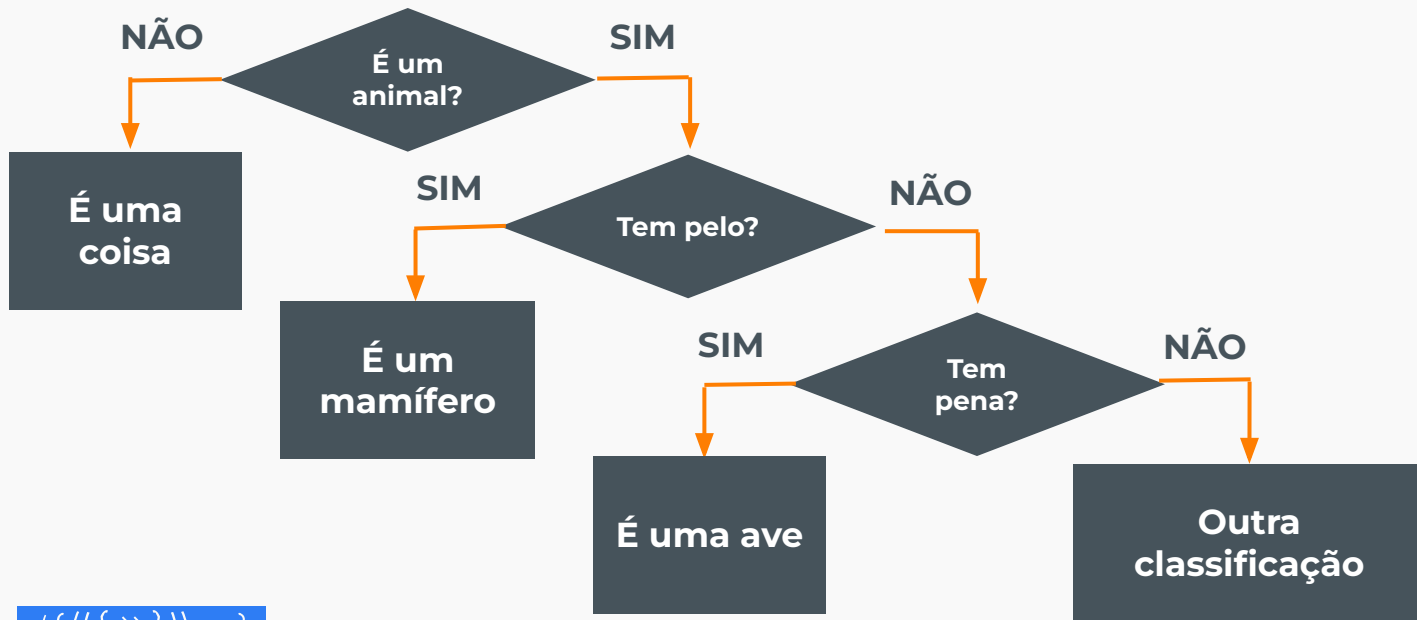
```
{((({>>}))<<}
```

{ }



# Condicionais em JS

Encadeamento de condições



{((({>>}))<<)}

-{ }



# Condicionais em JS

[ ]

if + else + if

Todo código da ação  
vai entre chaves { }

{ }

{((({>>}))<<)}

```
1  let condicao1 = false
2  let condicao2 = true
3
4  if (condicao1){
5      console.log('Entrei no if 1!')
6  } else {
7      // Como o valor da condicao1 é false,
8      // o código do else será executado
9      if (condicao2) {
10         // Como o valor da condicao2 é true,
11         // executaremos esse código!
12         console.log('Entrei no if 2!')
13     }
14 }
```

- [ ]





# Condicionais em JS

[ ]

if + else if

```
1  let condicao1 = false
2  let condicao2 = true
3
4  if (condicao1){
5      console.log('Entrei no if 1!')
6  } else {
7      if (condicao2) {
8          console.log('Entrei no if 2!')
9      }
10 }
```



```
1  let condicao1 = false
2  let condicao2 = true
3
4  if (condicao1){
5      console.log('Entrei no if 1!')
6  } else if (condicao2) {
7      console.log('Entrei no if 2!')
8  }
```

{ }

{((({>>}))<<}

- [ ]



# Condicionais em JS

[ ]

if + else if + else

{ }

{((({>>}))<<)}

```
1  let condicao1 = false
2  let condicao2 = false
3
4  if (condicao1){
5      console.log('Entrei no if 1!')
6  } else if (condicao2) {
7      console.log('Entrei no if 2!')
8  } else {
9      // Como tanto o valor da condicao1 e
10     // da condicao2 são false, executa
11     // os comandos do bloco else
12     console.log('Entrei no else!')
13 }
```

-[ ]



# Exercícios

[ ]

1. Crie um programa que:

a) Recebe 2 números (chamaremos de num1 e num2)

```
const num1 = Number(prompt("Digite o número 1"))
```

```
const num2 = Number(prompt("Digite o número 2"))
```

b) Compara esses números entre si:

```
num1 === num2
```

c) Imprime mensagens dizendo se o primeiro número é **maior/menor/igual** ao segundo número.

{ }

```
if(num1 === num2){ console.log("Os números são iguais!")}
```

```
else if (num1 > num2){console.log("O num1 é maior que o num2")}
```

```
{((( {>>}) )<<)} else{console.log("O num1 é menor que o num2")}
```

[ ]



# Resumo

[ ]

Vimos como a gente pode pedir para o computador realizar condições usando o JavaScript:

Condicional simples (**if**)

Condicional composta (**if/else**)

Condicional aninhada (**if + else if + else**)

{ }

{ ((({ >> } )) << ) }

- [ ]



# Switch Case

[ ]

Onde usar?

Vamos dar uma olhada  
no código:

{ }

{((({>>}))<<}

```
1  let paisDeOrigem
2  if (paisDeOrigem === 'Brasil'){
3    console.log('brasileiro')
4  } else if (paisDeOrigem === 'EUA'){
5    console.log('norte americano')
6  } else if (paisDeOrigem === 'Inglaterra'){
7    console.log('inglês')
8  } else if (paisDeOrigem === 'França'){
9    console.log('francês')
10 } else if (paisDeOrigem === 'Itália'){
11   console.log('italiano')
12 } else if (paisDeOrigem === 'Canadá'){
13   console.log('canadense')
14 } else {
15   console.log('nacionalidade não encontrada')
16 }
```

- [ ]



# Switch Case



[ ]

Comparando com if/else

```
1 let paisDeOrigem
2 if (paisDeOrigem === 'Brasil'){
3   console.log('brasileiro')
4 } else if (paisDeOrigem === 'EUA'){
5   console.log('norte americano')
6 } else if (paisDeOrigem === 'Inglaterra'){
7   console.log('inglês')
8 } else if (paisDeOrigem === 'França'){
9   console.log('francês')
10 } else if (paisDeOrigem === 'Itália'){
11   console.log('italiano')
12 } else if (paisDeOrigem === 'Canadá'){
13   console.log('canadense')
14 } else {
15   console.log('nacionalidade não encontrada')
16 }
```

{ }

```
1 let paisDeOrigem
2 switch (paisDeOrigem){
3   case 'Brasil':
4     console.log('brasileiro')
5     break
6   case 'EUA':
7     console.log('norte americano')
8     break
9   case 'Inglaterra':
10    console.log('inglês')
11    break
12  default:
13    console.log('nacionalidade não encontrada')
14    break
15 }
```

}



# Switch Case



[ ]

## Estrutura

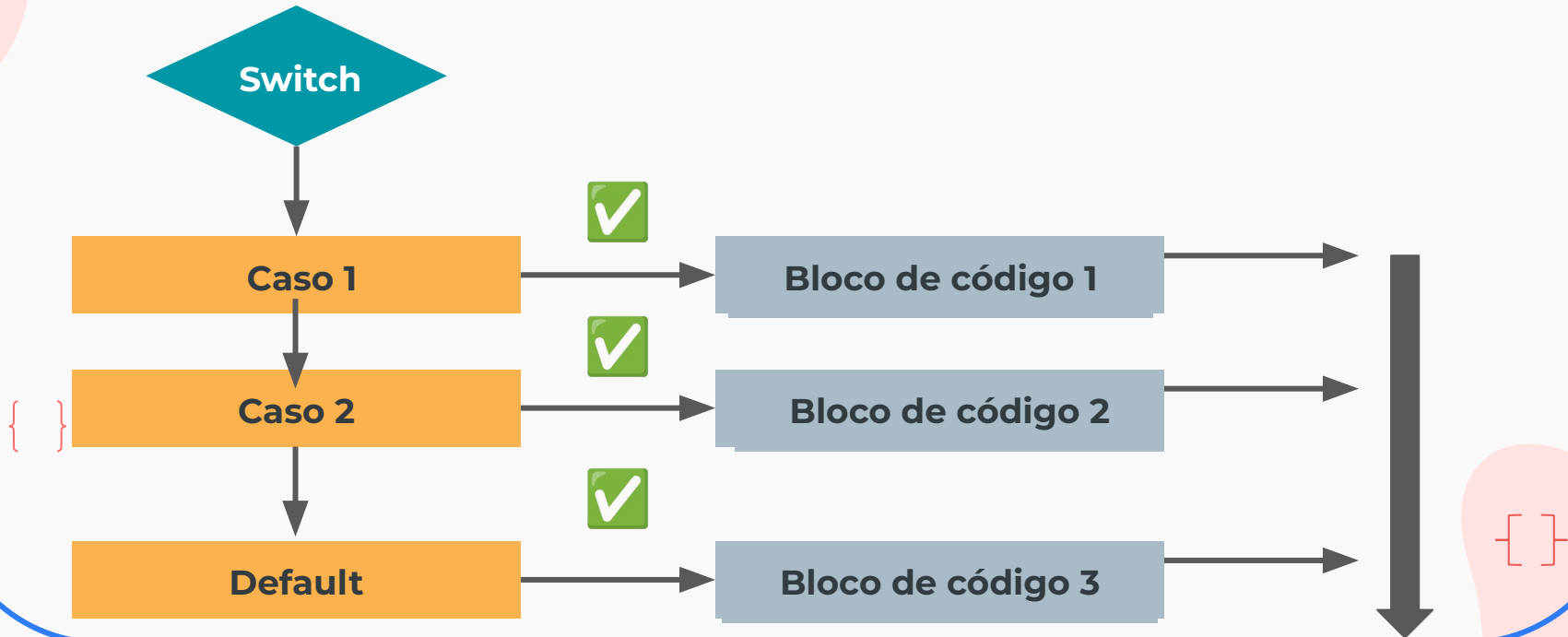
```
1 let paisDeOrigem
2 if (paisDeOrigem === 'Brasil'){
3   console.log('brasileiro')
4 } else if (paisDeOrigem === 'EUA'){
5   console.log('norte americano')
6 } else if (paisDeOrigem === 'Inglaterra'){
7   console.log('inglês')
8 } else if (paisDeOrigem === 'França'){
9   console.log('francês')
10 } else if (paisDeOrigem === 'Itália'){
11   console.log('italiano')
12 } else if (paisDeOrigem === 'Canadá'){
13   console.log('canadense')
14 } else {
15   console.log('nacionalidade não encontrada')
16 }
```

```
1 let paisDeOrigem
2 switch (paisDeOrigem){
3   case 'Brasil':
4     console.log('brasileiro')
5     break
6   case 'EUA':
7     console.log('norte americano')
8     break
9   case 'Inglaterra':
10    console.log('inglês')
11    break
12  default:
13    console.log('nacionalidade não encontrada')
14    break
15 }
```



# Switch Case

Estrutura







# Switch Case

[ ]

## Estrutura

```
1  let paisDeOrigem
2  switch (paisDeOrigem){
3      case 'Brasil':
4          console.log('brasileiro')
5          break
6      case 'EUA':
7          console.log('norte americano')
8          break
9      case 'Inglaterra':
10         console.log('inglês')
11         break
12     default:
13         console.log('nacionalidade não encontrada')
14         break
15 }
```

Conseguimos colocar um caso padrão chamado **default**

O código dentro dele será executado se o valor da variável **não bater com as opções** dos cases

[ ]



# Switch Case

[ ]

## Estrutura

```
1  let paisDeOrigem
2  switch (paisDeOrigem){
3      case 'Brasil':
4          console.log('brasileiro')
5          break
6      case 'EUA':
7          console.log('norte americano')
8          break
9      case 'Inglaterra':
10         console.log('inglês')
11         break
12     default:
13         console.log('nacionalidade não encontrada')
14         break
15 }
```

**break** é a palavra que faz com que a execução do código saia do bloco em questão. Caso não exista o break, o código continuará executando

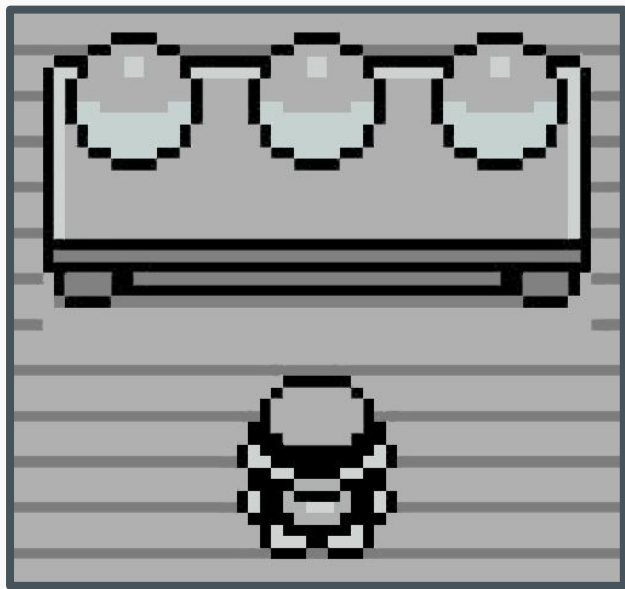
- [ ]



# Exercícios

Escreva um programa que receba o Pokémon inicial escolhido pela pessoa e imprima no console o seu tipo:

```
const pokemon = prompt("Qual o pokemon?")
switch(pokemon){
  case 'Bulsasaur':
    console.log("Bulbasaur(Planta e Veneno)")
    break
  case 'Charmander':
    console.log("Charmander(Fogo)")
    break
  case 'Squirtle':
    console.log("Squirtle(Água)")
    break
}
```





# Exercícios

[ ]

Escreva o programa do chapéu seletor que recebe a aluna e imprima no console a sua casa de Hogwarts:

- Hermione(Grifinória);
- Ana(Lufa-lufa);
- Luna(Corvinal);
- Narcisa(Sonserina)

{ }

{((({>>}))<<}





# Resumo

[ ]

Podemos realizar condições usando no JavaScript o **if + else**. Além disso, podemos usar o **switch case**.

- Switch case **evita códigos verbosos**;
- Só serve para **verificar casos de igualdade**;
- Permite executar bloco de código diferente baseado em **cada opção**;
- **Break** = Para a execução do código quando a **condição é atendida**;

{ }

{((({>>}))<<}

-[ ]



# Condicionalis e Operadores Lógicos



[ ]

## Relembrando Operadores Lógicos

- **&&** : AND - verdadeiro se ambos os operandos forem verdadeiros.
- **||** : OR - falso se ambos os operadores forem falsos.
- **!** : NOT negação - sua função é simplesmente inverter os valores.

{ }

{ ((({ >> } )) << ) }

- [ ]



# Condicionais e Operadores Lógicos



[ ]

Como os **operadores lógicos** retornam booleanos, conseguimos usá-los **diretamente** na condição dos blocos if/else

```
1  let condicao1 = true
2  let condicao2 = false
3
4  if (condicao1 && condicao2){
5      // Entra aqui se ambas forem true
6  }
7
8  if (!condicao1){
9      // Entra aqui se condicao1 for false
10 }
```

{ }

{((({>>}))<<)}

-[ ]



# Exercícios

[ ]

Uma pessoa pode estudar em uma faculdade se:

- Tiver concluído o ensino médio;

```
const ensino = prompt("concluiu o ensino médio?").toLowerCase() === 'sim'
```

- Tiver 18 anos ou mais;

```
const idade = Number(prompt("Digite sua idade")) >= 18
```

- Não estiver cursando outra faculdade;

```
const faculdade = prompt("Você faz faculdade?").toLowerCase() === 'sim'
```

Escreva um programa que receba estes parâmetros e imprima se a pessoa pode ou não estudar nesta faculdade. Receba os dados do usuário pelo prompt.

{ }

```
if(ensino && idade && !faculdade){
```

```
    console.log("Você pode estudar nessa faculdade")
```

```
    {{{({>>}})<<}}else{
```

```
        console.log("Você não pode estudar nessa faculdade")}
```

[ ]





# Desafio

[ ]

Crie um **jogo de "Pedra, Papel, Tesoura, Lagarto, Spock"** no qual o usuário escolhe uma das opções, e o computador escolhe aleatoriamente outra. Utilize um **switch-case** para determinar o vencedor.

O jogo segue estas **regras**: Pedra esmaga Tesoura. Pedra esmaga Lagarto. Papel cobre Pedra. Papel desqualifica Spock. Tesoura corta Papel. Tesoura decapita Lagarto. Lagarto envenena Spock. Lagarto come Papel. Spock vaporiza Pedra. Spock quebra Tesoura.

Gere uma escolha aleatória para o computador:

```
const escolhaComputador = choices[Math.floor(Math.random() * 5)];
```

{ }

Use um switch-case para comparar as escolhas e determinar o vencedor. Utilize operadores lógicos para definir as regras de vitória.

```
{((({>>}))<<}
```

-{ }



# Resumo

[ ]

**Operadores de comparação** são usados para se compararem valores de mais de uma variável. Eles sempre **retornam** um valor **booleano**.

- ==: valor e tipo iguais
- !=: valor ou tipo diferentes
- >: maior que
- >=: maior ou igual que
- <: menor que
- <=: menor ou igual que

{ }

{((({>>}))<<)}

-[ ]



# Resumo

[ ]

**Condicionais** são estruturas que simbolizam decisões tomadas dependendo de certas **condições**.

**Árvore condicional** é uma estrutura **esquemática** que pode ser usada para **facilitar** nossa análise e construção de condicionais

{ }

{((({>>}))<<}

- [ ]



# Resumo

[ ]

**if/else** são blocos que permitem fazer uma condicional. Eles recebem uma condição e o código:

- do **if** é executado se a **condição** for **true**;
- do **else** é executado se a **condição** for **false**;

**switch case** são blocos que permitem simplificar **if/else** apenas no caso de **comparador de igualdade**.

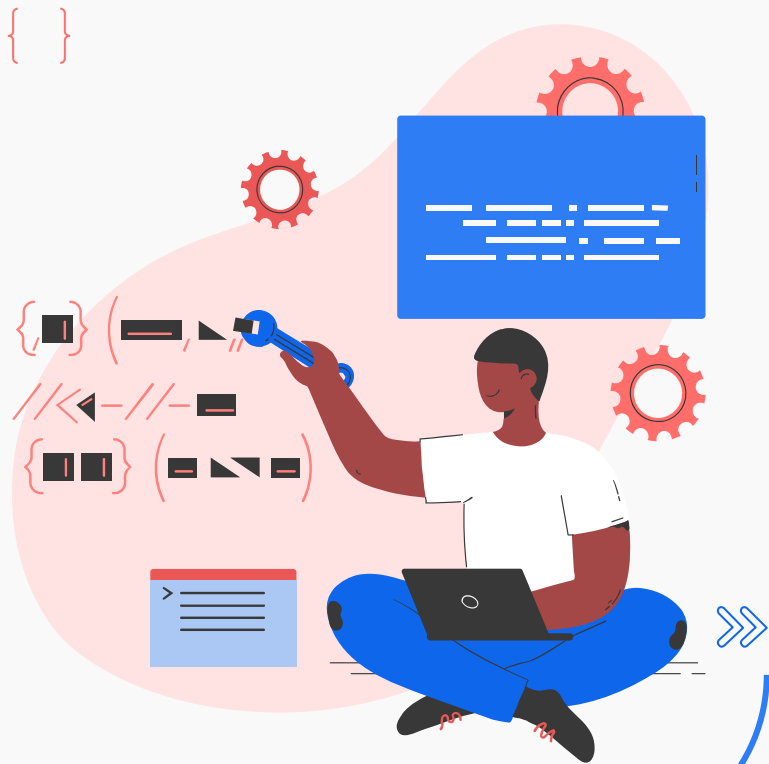
- lembrem-se de escrever o break;

{ }

{((({>>}))<<)}

-[ ]

# Exercícios de Fixação





# Exercícios de Fixação

[ ]

## Exercícios de interpretação de código

1. Teste de números:

- a) Explique o que o código faz. Qual o teste que ele realiza?
- b) Para que tipos de números ele imprime no console "Passou no teste"?
- c) Para que tipos de números a mensagem é "Não passou no teste"?

```
const respostaDoUsuario = prompt("Digite o número que você quer testar")
const numero = Number(respostaDoUsuario)

if (numero % 2 === 0) {
  console.log("Passou no teste.")
} else {
  console.log("Não passou no teste.")
}
```

{ }

[ ]



# Exercícios de Fixação



[ ]

## Exercícios de interpretação

### 2. Mercado

- a) Para que serve o código?
- b) Qual será a mensagem impressa no console, se o valor de fruta for `"Maçã"`?
- c) Qual seria a mensagem impressa no console se comprasse `pera` e retirássemos o `break` que está logo acima do `default` ("BREAK PARA O ITEM c.")?

{ }

{((({>>}))<<)}

```
let fruta = prompt("Escolha uma fruta")
let preco
switch (fruta) {
  case "Laranja":
    preco = 3.5
    break;
  case "Maçã":
    preco = 2.25
    break;
  case "Uva":
    preco = 0.30
    break;
  case "Pêra":
    preco = 5.5
    break; // BREAK PARA O ITEM c.
  default:
    preco = 5
    break;
}
console.log("O preço da fruta ", fruta, " é ", "R$ ", preco)
```



# Exercícios de Fixação

[ ]

## Exercícios de interpretação de código

### 3. Mensagem Secreta

- a) O que a primeira linha está fazendo?
- b) Considere um usuário digitou o número 10. Qual será a mensagem do terminal? E se fosse o número -10?
- c) Haverá algum erro no console? Justifique usando os conceitos de bloco ou escopo.

{ }

{((({>>}))<<)}

```
const numero = Number(prompt("Digite o primeiro número."))

if(numero > 0) {
  console.log("Esse número passou no teste")
  let mensagem = "Essa mensagem é secreta!!!"
}

console.log(mensagem)
```

- [ ]





# Exercícios de Fixação

[ ]

## Exercícios de escrita de código

4. Faça um programa que pergunta ao usuário qual a idade dele e imprima no console se ele/ela pode dirigir (apenas maiores de idade).

a) Faça um `prompt` para receber a idade do usuário e guarde em uma variável.

b) Garanta que essa variável é do tipo `Number`, você deve usar o cast para number para isso.

c) Agora veja se essa idade do usuário corresponde à idade mínima que permite dirigir. Se sim, imprima no console `"Você pode dirigir"`, caso contrário, imprima `"Você não pode dirigir."`

{ }

{((({>>}))<<}

-[ ]



# Exercícios de Fixação

[ ]

## Exercícios de escrita de código

5. Agora faça um programa que verifica que turno do dia um aluno estuda. Peça para digitar M (matutino) ou V (Vespertino) ou N (Noturno). Imprima no console a mensagem "Bom Dia!", "Boa Tarde!" ou "Boa Noite!". Utilize o bloco if/else

6. Repita o exercício anterior, mas utilizando a estrutura de switch case agora.

{ }

{((( {>>} ))<<}

- [ ]



# Exercícios de Fixação

[ ]

## Exercícios de escrita de código

7. Considere a situação: você vai ao cinema com um amigo ou amiga, porém ele/ela só assistirá a um filme com você se ele for do gênero fantasia e se o ingresso está abaixo de 15 reais. Faça um código que pergunta ao usuário qual o gênero de filme que vão assistir e outra pergunta sobre o preço do ingresso, então verifique se seu amigo ou amiga vai topa assistir o filme. Caso positivo, imprima no console a mensagem: "Bom filme!", caso contrário, imprima "Escolha outro filme :("

{ }

{((({>>}))<<}

- [ ]



# Desafio

[ ]

1. Modifique o código do exercício 4 de escrita de código para, antes de imprimir a mensagem "Bom filme!", pergunte ao usuário, pelo prompt qual lanchinho ele vai comprar (pipoca, chocolate, doces, etc) e imprima no console as mensagens "Bom filme!" e "Aproveite o seu [LANCHINHO]", trocando [LANCHINHO] pelo que o usuário colocou no input.

2. Você foi contratado para criar um sistema de vendas de ingressos de jogos de um estádio de futebol. Para esta compra, o usuário deve fornecer algumas informações:

{ }

Nome completo; Tipo de jogo: **IN indica internacional e DO indica doméstico**;

Etapa do jogo: **SF indica semi-final; DT indica decisão de terceiro lugar; e FI**

**{((({>>}))<<)} indica final**. Categoria: pode ser as opções 1, 2, 3 ou 4;

Quantidade de ingressos **CONTINUA ->**

[ ]

# Desafio

2. O seu sistema deve solicitar estas informações ao usuário, através do prompt . Além disso, ele deve imprimir tudo isso, junto com o valor de cada ingresso e o valor total que o usuário tem que pagar (ou seja, o valor unitário do ingresso multiplicado pela quantidade). Abaixo, há a tabela com os valores de cada ingresso. Lembrando que o valor de jogos internacionais é o mesmo de jogos domésticos, mas seus preços devem ser multiplicados pelo valor do dólar (considerar o dólar = R\$5,65)

Público	Jogo	Categoria 1	Categoria 2	Categoria 3	Categoria 4
DO Doméstico	SF - Semifinais	R\$ 1.320,00	R\$ 880,00	R\$ 550,00	R\$ 220,00
	DT - Decisão do 3º lugar	R\$ 660,00	R\$ 440,00	R\$ 330,00	R\$ 170,00
	FI - Final	R\$ 1.980,00	R\$ 1.320,00	R\$ 880,00	R\$ 330,00

{((({>>}))<<}

-{ }

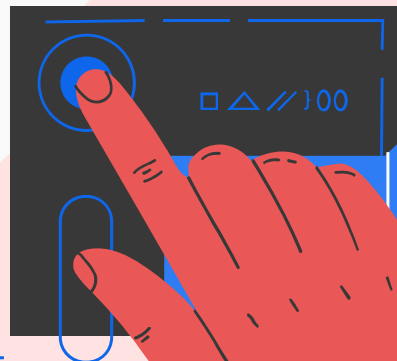
# Obrigado!



E-mail: [lgfalves@senacrs.com.br](mailto:lgfalves@senacrs.com.br)



{({({ >> } ) ) << }

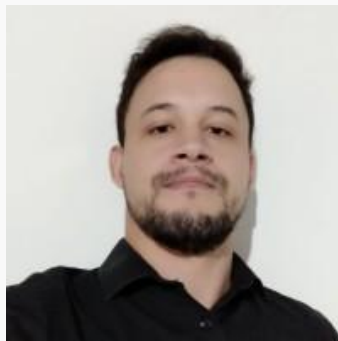


(( { >> 0 i □ □ □ } ))

```
((: 00 - =>> } )  
{ (<1 00 1 000 >> )}  
((: 0)>"< )  
<01 001} +100 0}>  
((: 0)>"< )  
{ (<1 00 1 000 >> )}
```



# Professor



**Lucas G. F. Alves**

