

IOS – Instituto de  
Oportunidade Social

## Aula JS 04 - Estruturas condicionais



- > Desvios condicionais
- > Estrutura condicional simples (if)
- > Estrutura condicional composta (if-else)
- > Desvios condicionais encadeados
- > Operador ternário
- > Switch

IOS – Instituto de  
Oportunidade Social

Desvios condicionais



- Os **desvios condicionais** decidem o fluxo de execução de programa.
- Uma das tarefas fundamentais de qualquer programa é **decidir o que deve ser executado a seguir**.
- Os **comandos de decisão** permitem determinar qual é a ação a ser tomada com base no resultado de uma expressão condicional.
- Seja em estruturas condicionais ou laços de repetições, sempre será necessário testar uma condição que irá resultar em um valor booleano (**true** ou **false**). Por isso, quase sempre, usamos de operadores de **comparação e/ou lógicos**:

Categoria	Operador	Descrição
Operadores de comparação	==	Igual
	!=	Diferente
	<	Menor que
	<=	Menor ou igual
	>	Maior que
	>=	Maior ou igual
	===	Triplo igual

Categoria	Operador	Descrição
Operadores lógicos	&&	Lógica “and” ou “e”, que retorna verdadeiro se todos os operandos forem verdadeiros.
		Lógica “or” ou “ou”, que retorna verdadeiro se pelo menos um operando for verdadeiro.
	!	Lógica “not” ou “não”, que inverte o valor lógico se é verdadeiro, retorna falso e se é falso retorna verdadeiro.

IOS – Instituto de  
Oportunidade Social

Estrutura condicional simples (if)



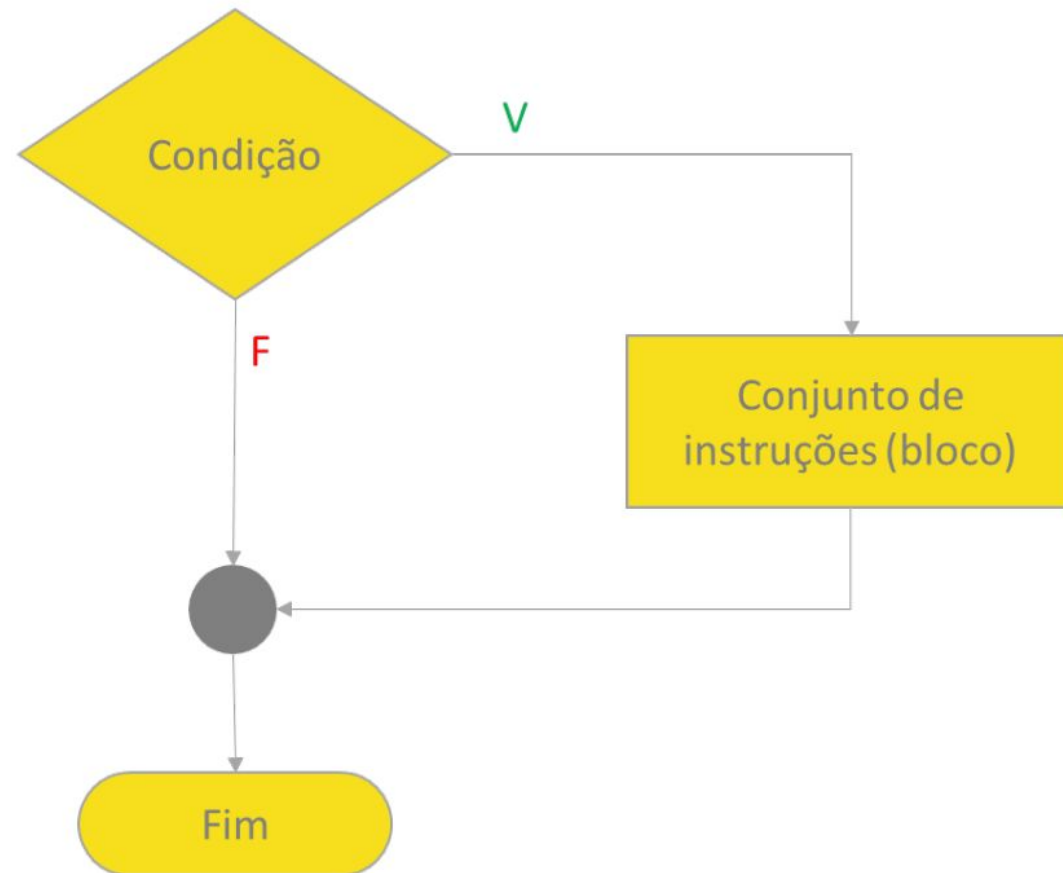
A estrutura condicional simples **if (SE)** é usada para verificar se **dada condição é atendida**:

- **Se for**, um **conjunto de instruções deverá ser executado**;
- **Se não**, o fluxo de execução do algoritmo **seguirá após o fim do bloco de decisão**;



# Estrutura condicional simples (if)

## Exemplo:

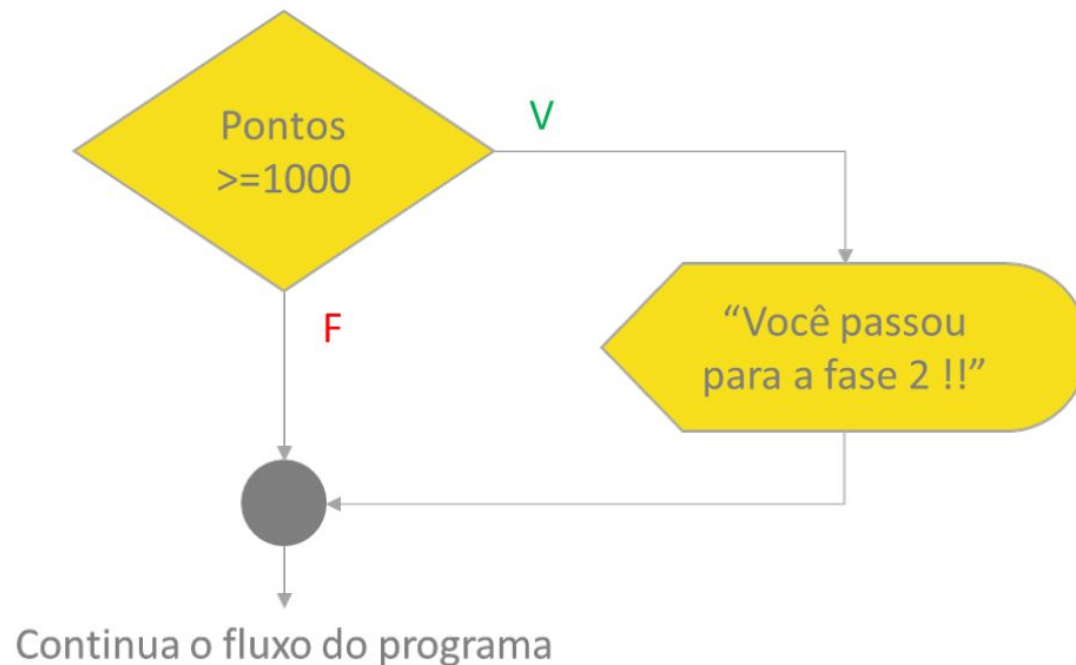


A sintaxe da estrutura **if** é:

```
if (condição) {  
    instrucao1;  
    instrucao2;  
    ...  
    instrucaoN;  
}  
proximaInstrucao;
```

# Estrutura condicional simples (if)

**Exemplo:** Um jogador somente irá **passar para a fase 2 se atingir 1000 pontos.**



> Vamos Praticar

IOS – Instituto de  
Oportunidade Social

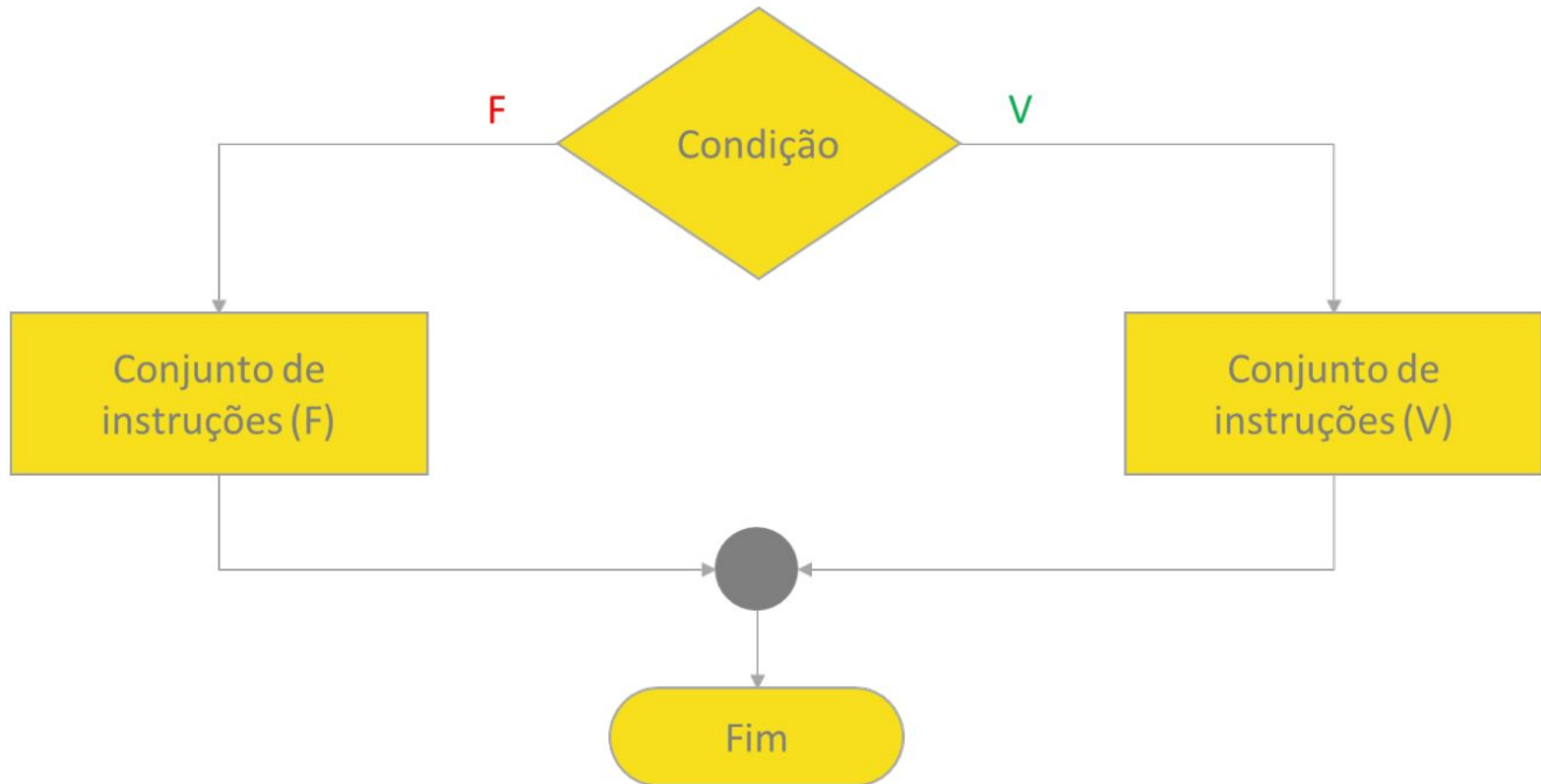
Estrutura condicional composta (if-else)



A **estrutura if-else** prevê **dois conjuntos de instruções para serem executadas** de acordo com a avaliação da condição:

- Um conjunto de instruções que será executado quando a condição resultar em **Verdadeiro**;
- Um conjunto de instruções quando a condição resultar em **Falso**;

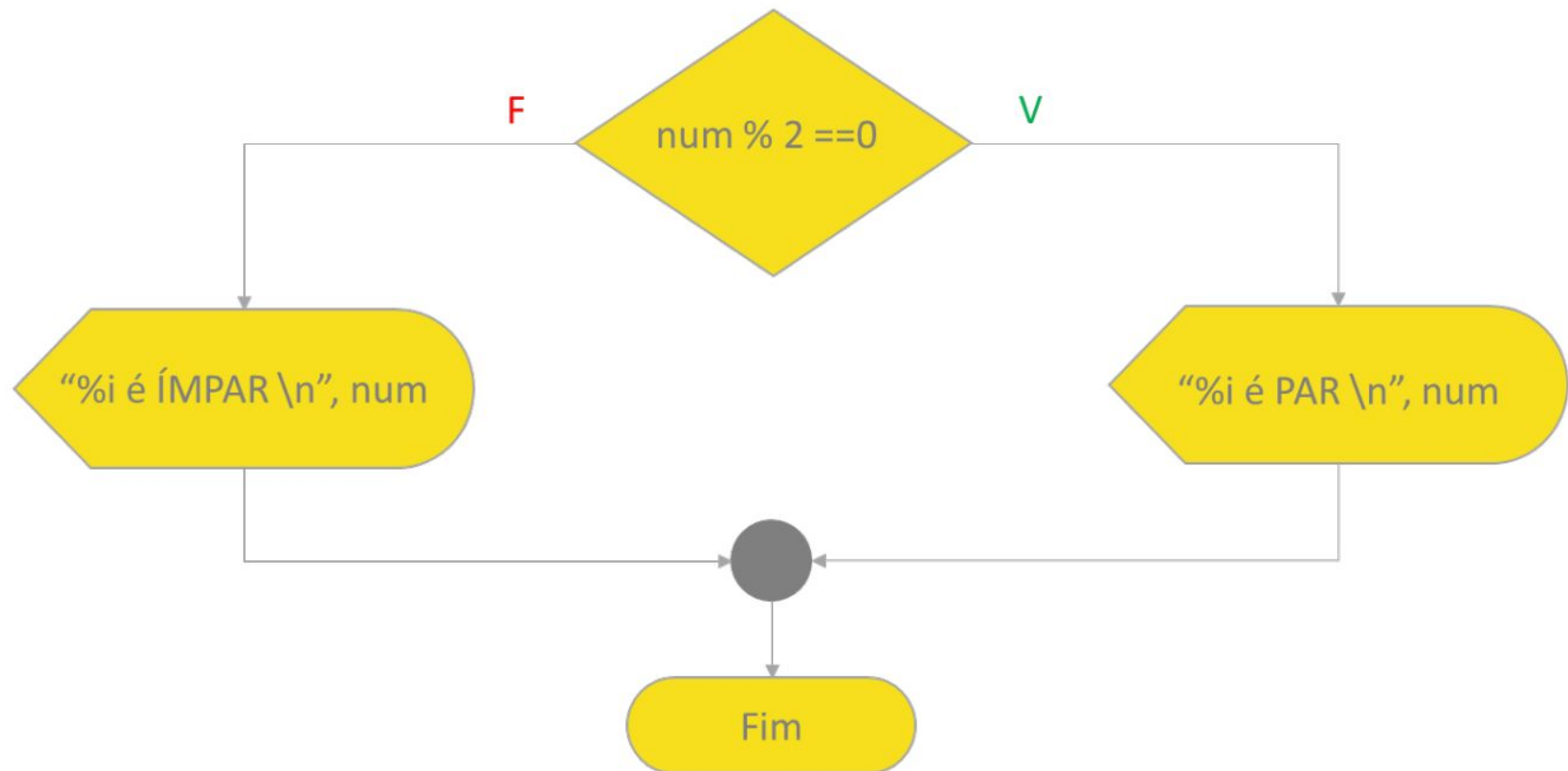
# Estrutura condicional composta (if-else)



## A sintaxe do if-else é:

```
if (condição) {  
    instrucao1;  
    instrucao2;  
}  
else {  
    instrucao3;  
    instrucao4;  
}  
proximaInstrucao;
```

# Estrutura condicional composta (if-else)



> Vamos Praticar



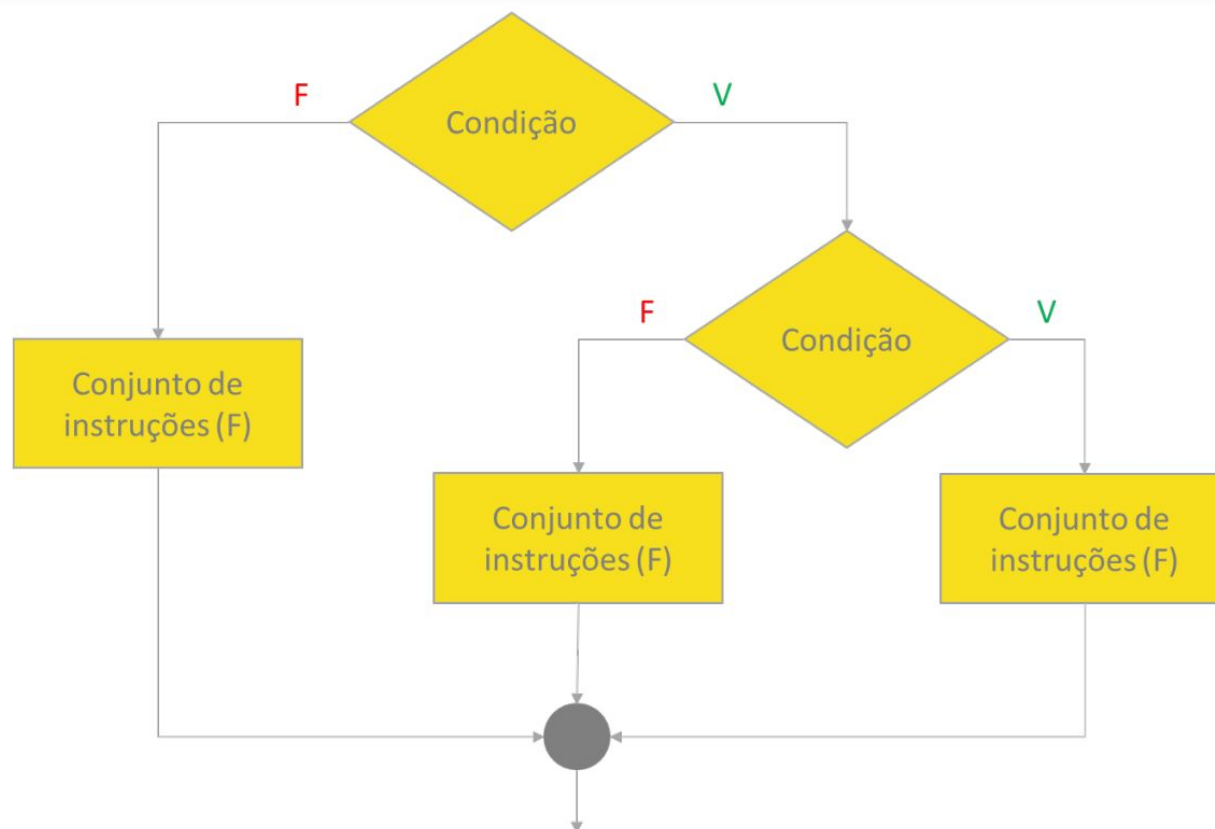
IOS – Instituto de  
Oportunidade Social

## Desvios condicionais encadeados



**Você pode encadear vários desvios condicionais quando for necessário** verificar diversas condições. E cada condição **depende do resultado da condição** anterior. Basicamente, **if dentro de if** (denominado **ifs-elses aninhados**):

# Desvios condicionais encadeados



> Vamos Praticar

IOS – Instituto de  
Oportunidade Social

Operador ternário



O **operador ternário** é uma **instrução equivalente** a estrutura condicional composta **if...else** e a sua vantagem é tornar o código bem mais enxuto. A sua sintaxe é:

```
condição ? expr_1 : expr_2
```

- **condição** é a condição que será testada.
- **expr\_1** é o que fazer quando a condição for verdadeira.
- **expr\_2** é o que fazer quando a condição for falsa.

> Vamos Praticar

IOS – Instituto de  
Oportunidade Social

Switch



O **switch-case** trabalha com **situações mutuamente exclusivas**.  
A sua sintaxe é:

```
switch (expressão) {  
    case valor1:  
        //Instruções executadas quando o resultado da expressão for igual á valor1  
        [break;]  
    case valor2:  
        //Instruções executadas quando o resultado da expressão for igual á valor2  
        [break;]  
    ...  
    case valueN:  
        //Instruções executadas quando o resultado da expressão for igual á valorN  
        [break;]  
    default:  
        //Instruções executadas quando o valor da expressão é diferente de todos os cases  
        [break;]  
}
```

> Vamos Praticar

IOS – Instituto de  
Oportunidade Social

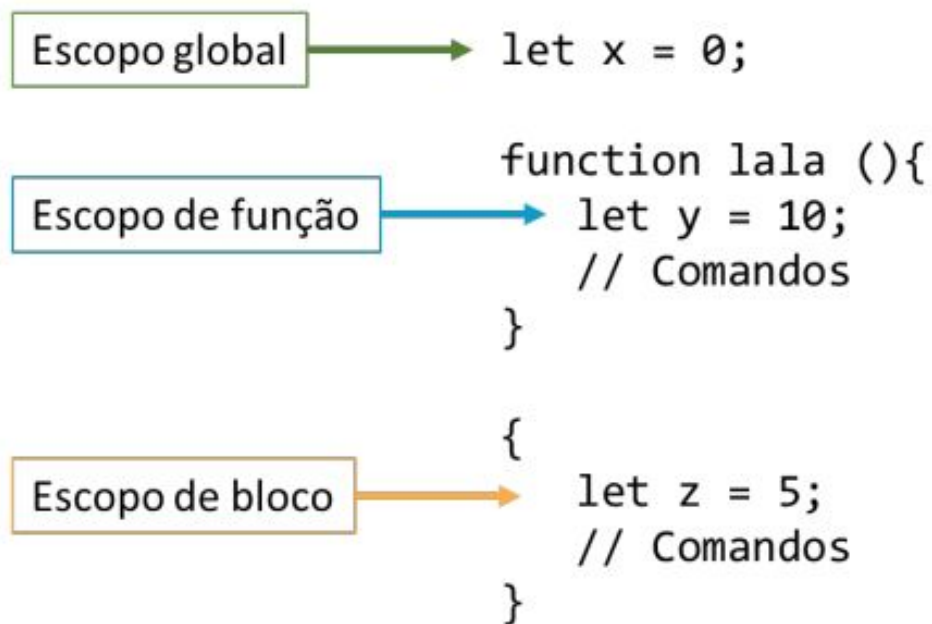
Escopo de variáveis





## > Escopo de Variáveis:

O escopo é o conjunto de regras que determinam o uso e a validade de variáveis nas diversas partes do programa. Um escopo define uma região do programa definida pela abertura e fechamento de chaves { }. JavaScript permite criar variáveis em três escopos: global, de função e de bloco.



IOS – Instituto de  
Oportunidade Social

Exercício



Crie um algoritmo que leia dois números inteiros e verifique qual é maior e qual é o menor e imprima o resultado no console do Browser.

*Exemplo: se `var numero1 = 4` for maior que `var numero2 = 2` então imprima no console a mensagem “numero1 é maior que número2” se for menor imprima que a numero1 é menor.*

Entregue o link do seu gitHub com o exercício no moodle.