

# Programação De Computadores

Professor : Yuri Frota

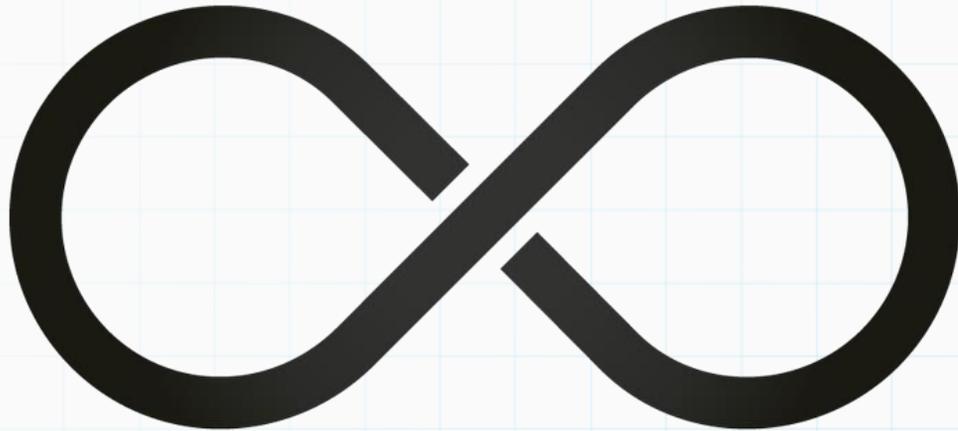
[www.ic.uff.br/~yuri/prog.html](http://www.ic.uff.br/~yuri/prog.html)

yuri@ic.uff.br

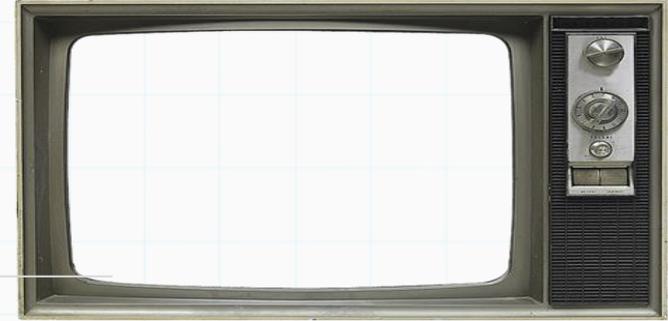
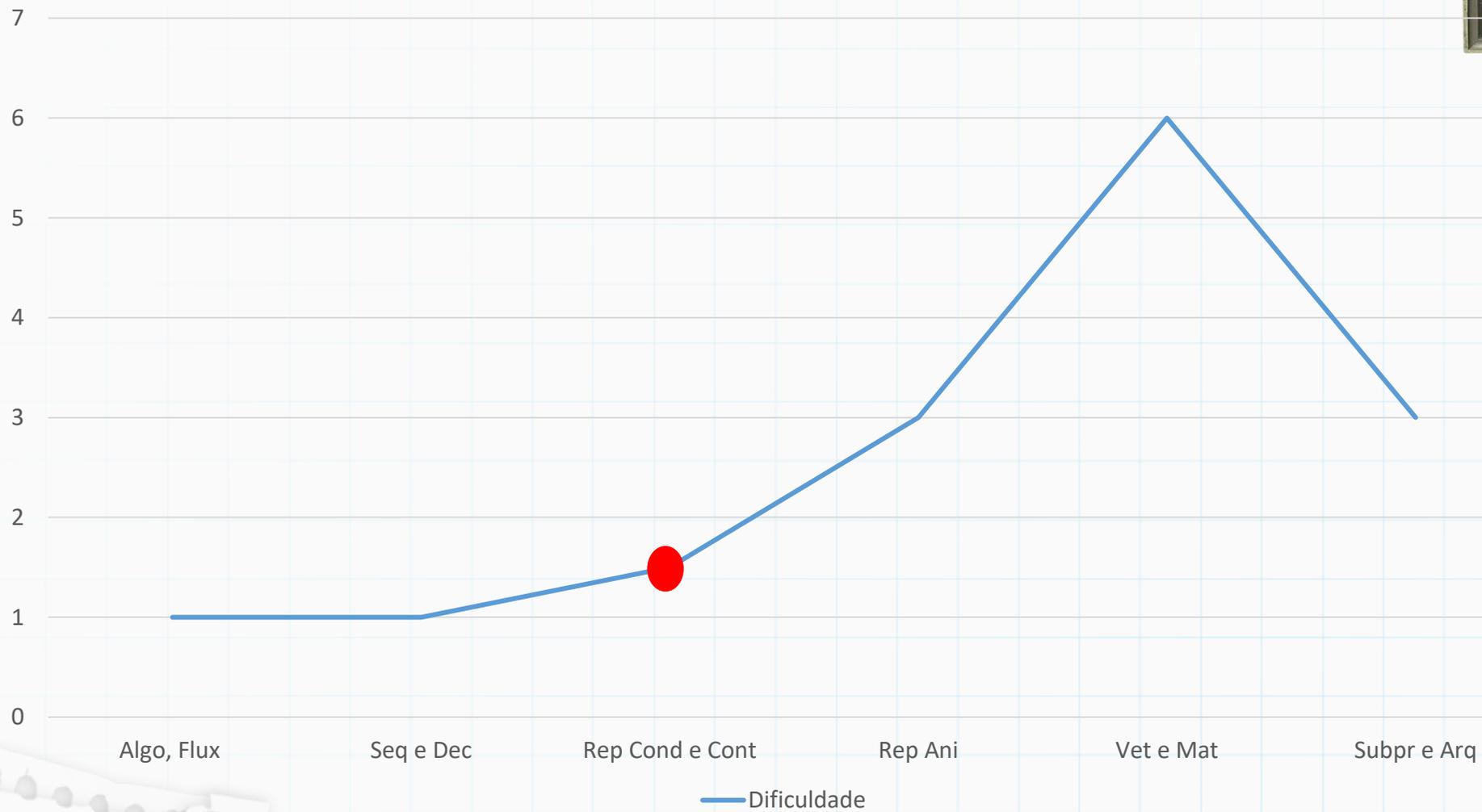


# Repetição

Um comando de repetição é aquele que permite repetir um determinado bloco de comandos. Existem dois tipos de repetição: as condicionais e as contáveis.



## Percepção de Dificuldade dos Alunos



# Repetição Condicional

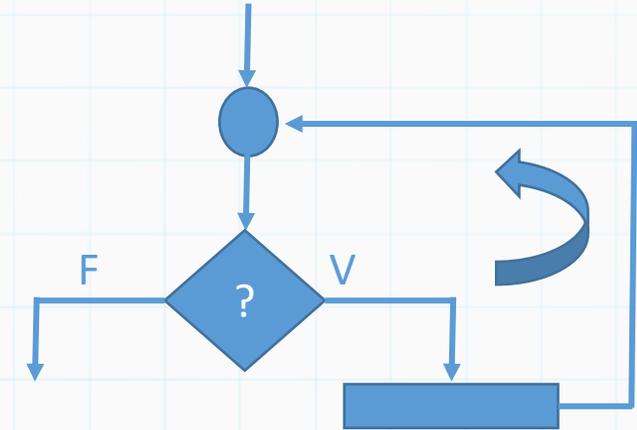
Comando While:

Executa o bloco de instruções enquanto a condição for verdadeira



Portugol

```
...  
enquanto CONDIÇÃO  
faça  
  INSTRUÇÃO 1;  
  INSTRUÇÃO 2;  
  ...  
  INSTRUÇÃO N;  
...
```



# Repetição Condicional

Comando While:

Executa o bloco de instruções enquanto a condição for verdadeira

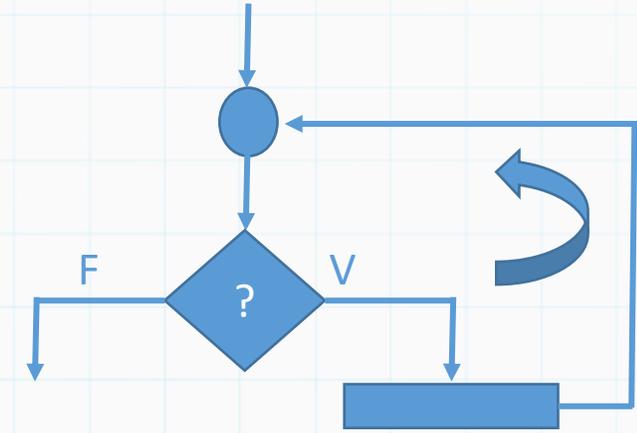


Portugol

```
...  
enquanto CONDIÇÃO  
faça  
    INSTRUÇÃO 1;  
    INSTRUÇÃO 2;  
    ...  
    INSTRUÇÃO N;  
...
```

Python

```
...  
while CONDIÇÃO:  
    INSTRUÇÃO 1  
    INSTRUÇÃO 2  
    ...  
    INSTRUÇÃO N  
...
```



# Repetição Condicional

Comando While:

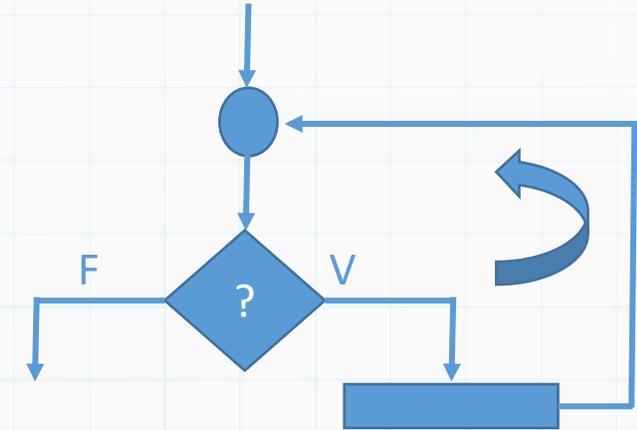
Executa o bloco de instruções enquanto a condição for verdadeira

Portugol

```
...  
enquanto CONDIÇÃO  
faça  
    INSTRUÇÃO 1;  
    INSTRUÇÃO 2;  
    ...  
    INSTRUÇÃO N;  
...
```

Python

```
...  
while CONDIÇÃO:  
    INSTRUÇÃO 1  
    INSTRUÇÃO 2  
    ...  
    INSTRUÇÃO N  
...
```



- A condição é uma expressão booleana que pode fazer uso de quaisquer operadores

# Repetição Condicional

Comando While:

Executa o bloco de instruções enquanto a condição for verdadeira

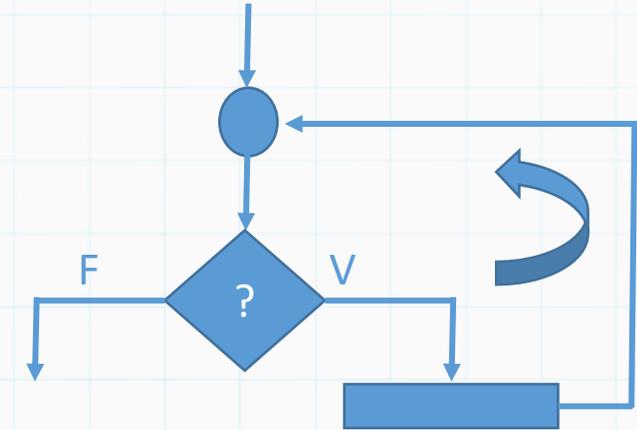


Portugol

```
...  
enquanto CONDIÇÃO  
faça  
    INSTRUÇÃO 1;  
    INSTRUÇÃO 2;  
    ...  
    INSTRUÇÃO N;  
...
```

Python

```
...  
while CONDIÇÃO:  
    INSTRUÇÃO 1  
    INSTRUÇÃO 2  
    ...  
    INSTRUÇÃO N  
...
```

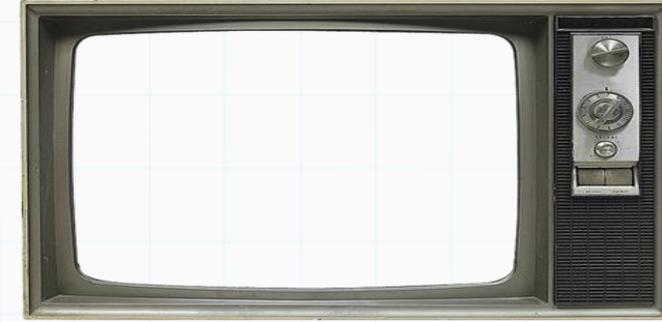


- A condição é uma expressão booleana que pode fazer uso de quaisquer operadores
- O bloco de instruções é delimitado por indentação

# Repetição Condicional

Comando While:

Executa o bloco de instruções enquanto a condição for verdadeira

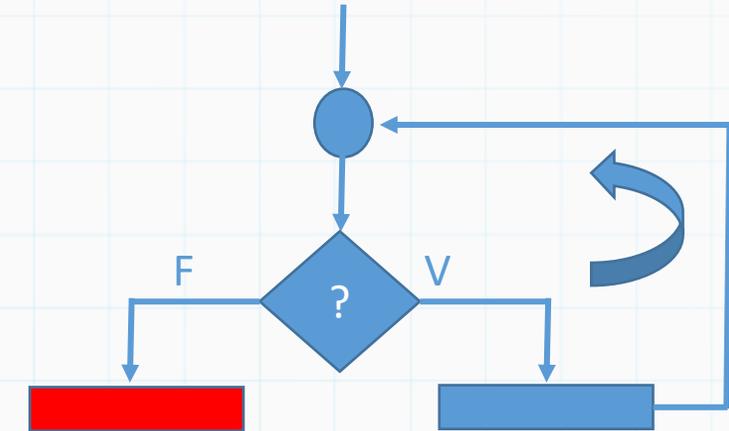


Portugol

```
...  
enquanto CONDIÇÃO  
faça  
    INSTRUÇÃO 1;  
    INSTRUÇÃO 2;  
    ...  
    INSTRUÇÃO N;  
...
```

Python

```
...  
while CONDIÇÃO:  
    INSTRUÇÃO 1  
    INSTRUÇÃO 2  
    ...  
    INSTRUÇÃO N  
...
```



- A condição é uma expressão booleana que pode fazer uso de quaisquer operadores
- O bloco de instruções é delimitado por indentação
- Deve haver algum processo dentro do bloco de comandos que torne a condição falsa para que a repetição seja encerrada

- Quando a condição se torna falsa, a próxima instrução após o bloco do **while** é executada

# Repetição Condicional

Comando While:

Executa o bloco de instruções enquanto a condição for verdadeira



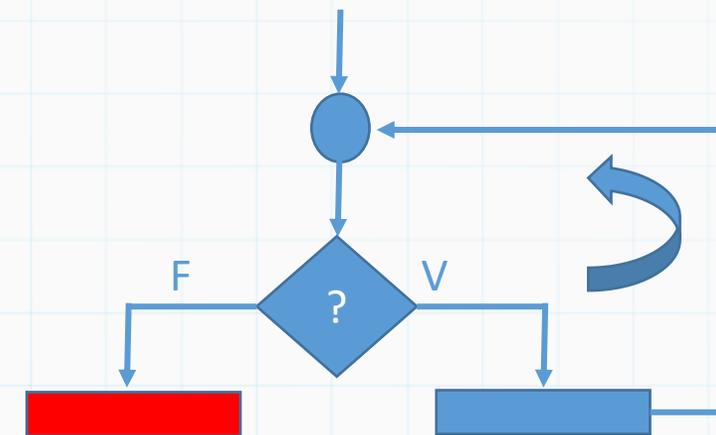
Portugol

```
...  
enquanto CONDIÇÃO  
faça  
    INSTRUÇÃO 1;  
    INSTRUÇÃO 2;  
    ...  
    INSTRUÇÃO N;  
...
```

Python

```
...  
while CONDIÇÃO:  
    INSTRUÇÃO 1  
    INSTRUÇÃO 2  
    ...  
    INSTRUÇÃO N  
...
```

- A condição é uma expressão booleana que pode fazer uso de quaisquer operadores
- O bloco de instruções é delimitado por indentação
- Deve haver algum processo dentro do bloco de comandos que torne a condição falsa para que a repetição seja encerrada

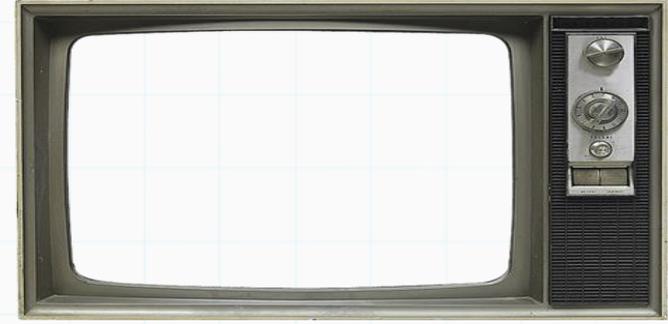
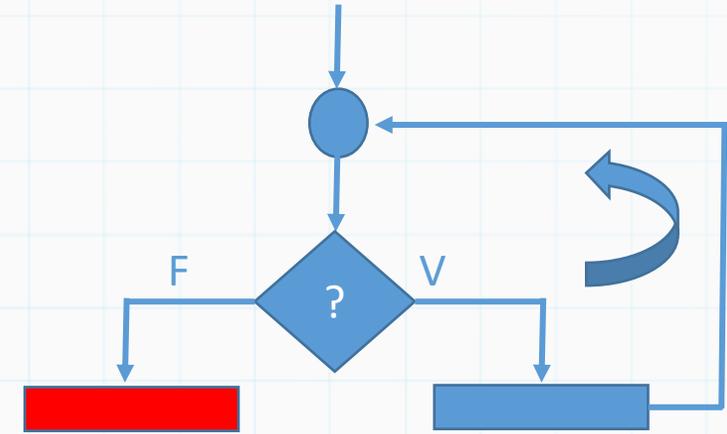


- Quando a condição se torna falsa, a **próxima instrução** após o bloco do **while** é executada.
- Se a condição do **while** for falsa desde o início, o bloco de instruções **nunca é executado**

# Repetição Condicional

Exemplo: Programa que imprime todos os números de 1 a 100

```
1 numero = 1
2 while numero <= 100:
3     print(numero)
4     numero = numero + 1
```



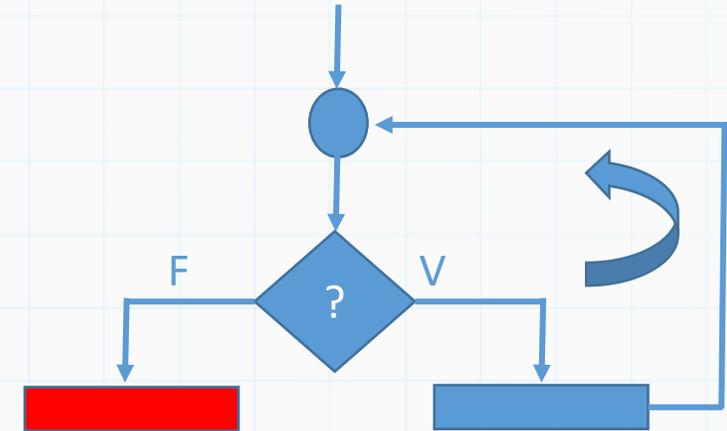
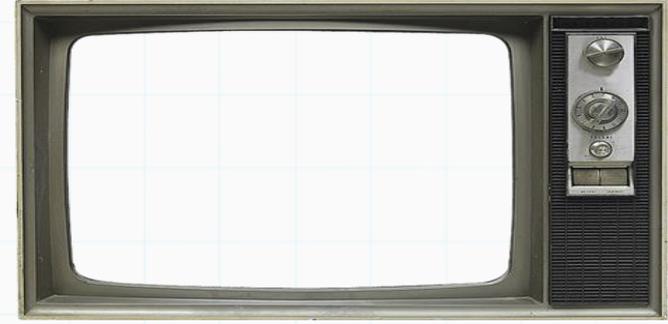
# Repetição Condicional

Exemplo: Programa que imprime todos os números de 1 a 100

```
1 numero = 1
2 while numero <= 100:
3     print(numero)
4     numero = numero + 1
```

Exemplo: Programa que imprime os  $n$  primeiros números

```
1 n = int(input("Digite um número: "))
2 numero = 1
3 while numero <= n:
4     print(numero)
5     numero = numero + 1
```



# Repetição Condicional

Exemplo: Programa que imprime todos os números de 1 a 100

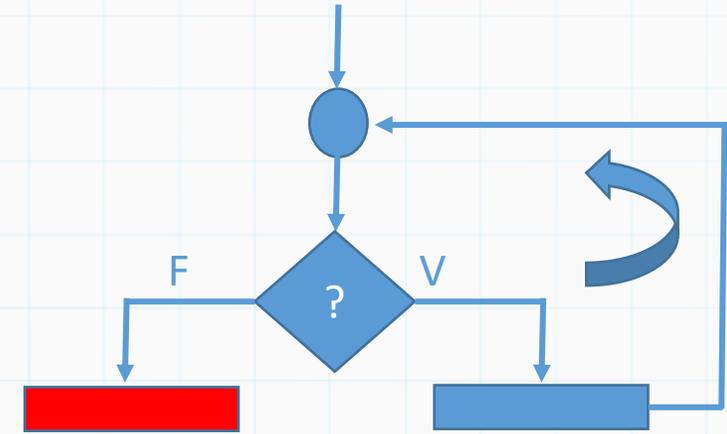
```
1 numero = 1
2 while numero <= 100:
3     print(numero)
4     numero = numero + 1
```

Exemplo: Programa que imprime os  $n$  primeiros números

```
1 n = int(input("Digite um número: "))
2 numero = 1
3 while numero <= n:
4     print(numero)
5     numero = numero + 1
```

Exemplo: E esse agora ?

```
1 a = 5
2 while a == a:
3     a = a + 1
4 print(a)
```

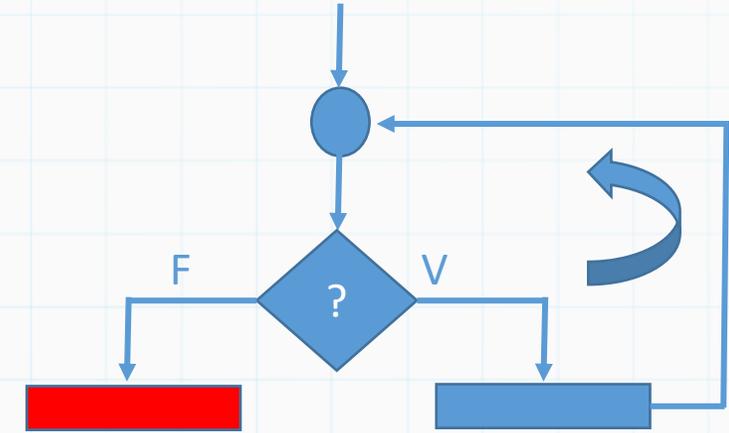


# Repetição Condicional

Exemplo: Programa que imprime a quantidade de números pares de 100 até 200, incluindo-os.

```
1 num = 100
2 contador_pares = 0
3 while num <= 200:
4     if num % 2 == 0:
5         contador_pares = contador_pares + 1
6         num = num + 1
7 print(contador_pares)
```

[código](#)



# Repetição Condicional

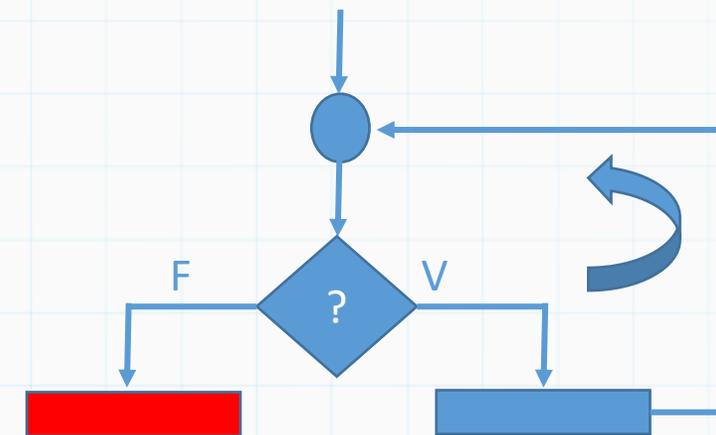
Exemplo: Programa que imprime a quantidade de números pares de 100 até 200, incluindo-os.

```
1 num = 100
2 contador_pares = 0
3 while num <= 200:
4     if num % 2 == 0:
5         contador_pares = contador_pares + 1
6     num = num + 1
7 print(contador_pares)
```

[código](#)

```
1 num = 100
2 contador_pares = 0
3 continua = True
4
5 while continua:
6     if num % 2 == 0:
7         contador_pares = contador_pares + 1
8     num = num + 1
9     if (num > 200):
10        continua = False
11
12 print(contador_pares)
```

[código](#)



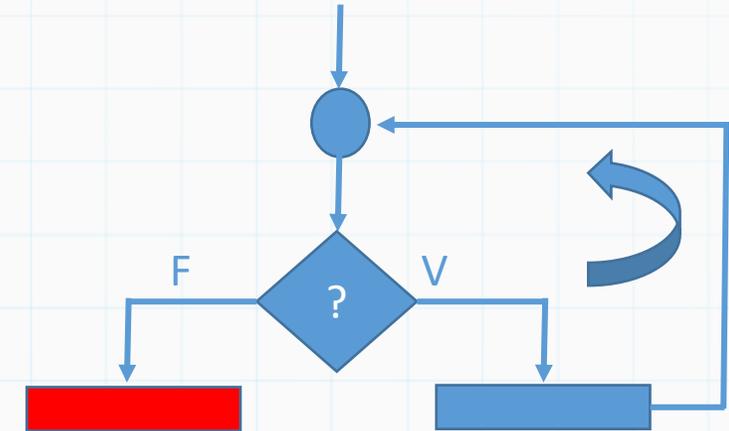
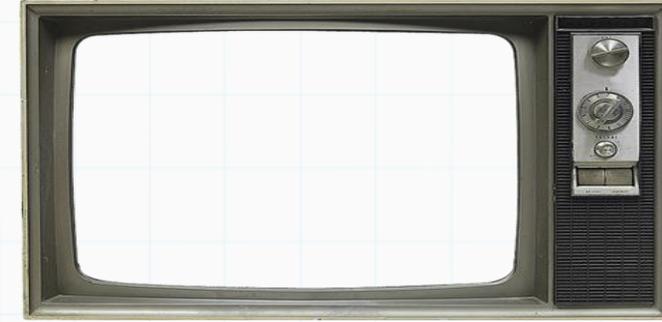
também podemos  
usar uma variável  
lógica para controlar  
o loop

# Repetição Condicional

E se agora quiséssemos saber a quantidade de pares entre quaisquer números dados

```
1 num1 = int(input('Entre com o valor inicial: '))
2 num2 = int(input('Entre com o valor final: '))
3
4 contador_pares = 0
5 while num1 <= num2:
6     if num1 % 2 == 0:
7         contador_pares = contador_pares + 1
8         num1 = num1 + 1
9 print(contador_pares)
```

[código](#)



# Repetição Condicional

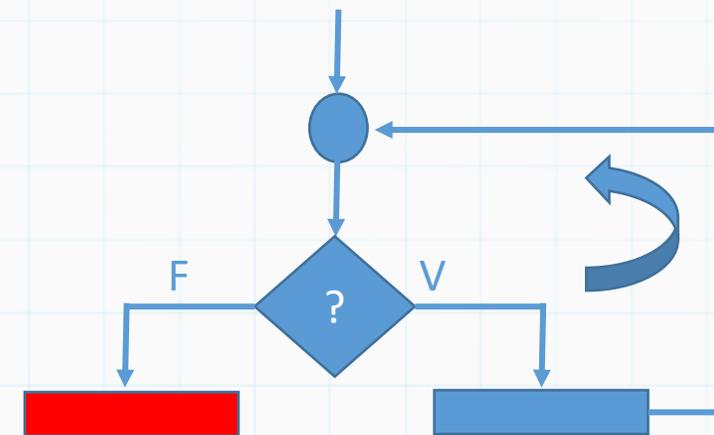
E se agora quiséssemos saber a quantidade de pares entre quaisquer números dados

```
1 num1 = int(input('Entre com o valor inicial: '))
2 num2 = int(input('Entre com o valor final: '))
3
4 contador_pares = 0
5 while num1 <= num2:
6     if num1 % 2 == 0:
7         contador_pares = contador_pares + 1
8     num1 = num1 + 1
9 print(contador_pares)
```

[código](#)

```
1 num1 = int(input('Entre com o valor inicial: '))
2 num2 = int(input('Entre com o valor final: '))
3
4 continua = True
5 contador_pares = 0
6 while (continua):
7     if num1 % 2 == 0:
8         contador_pares = contador_pares + 1
9     num1 = num1 + 1
10
11     if (num1 > num2):
12         continua = False
13 print(contador_pares)
```

[código](#)



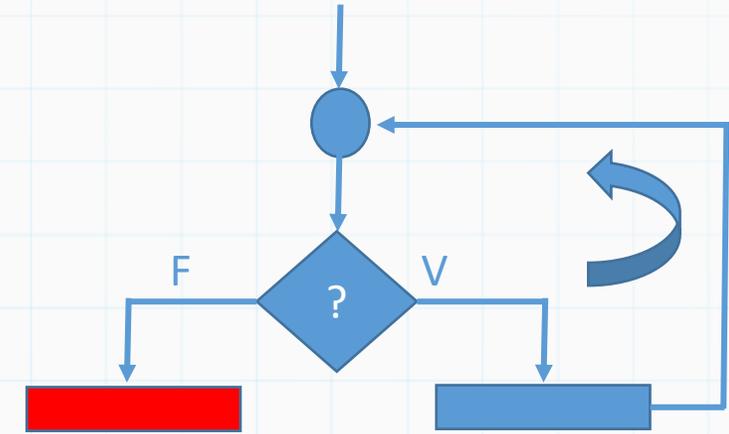
também podemos  
usar uma variável  
lógica para controlar  
o loop

# Repetição Condicional

E se fosse agora a soma de todos os números pares entre dois números quaisquer, incluindo-os ?

```
1 num1 = int(input('Entre com o valor inicial: '))
2 num2 = int(input('Entre com o valor final: '))
3
4 soma = 0
5 while num1 <= num2:
6     if num1 % 2 == 0:
7         soma = soma + num1
8     num1 = num1 + 1
9 print(soma)
```

[código](#)



Não temos só mais um contador, temos um acumulador

# Repetição Condicional

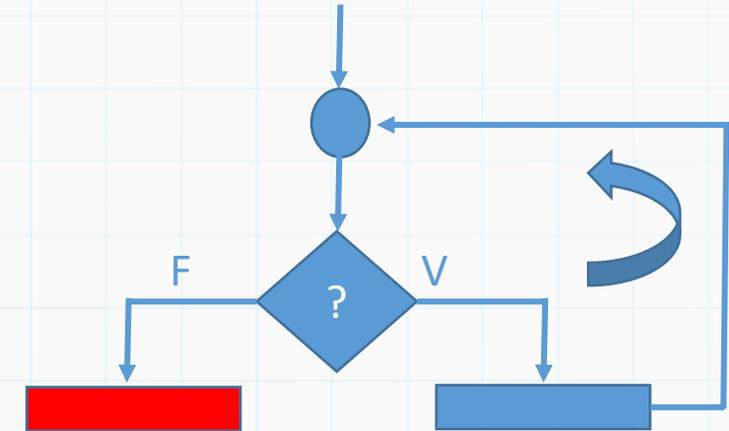
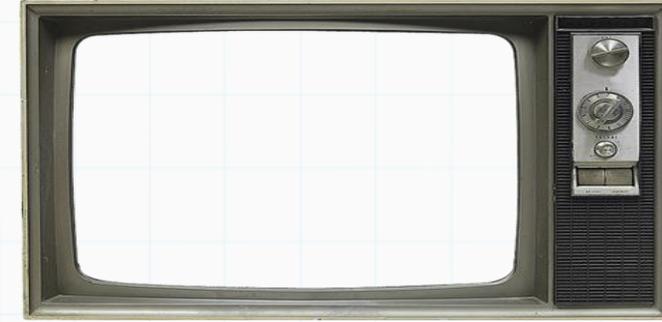
E se fosse agora a soma de todos os números pares entre dois números quaisquer, incluindo-os ?

```
1 num1 = int(input('Entre com o valor inicial: '))
2 num2 = int(input('Entre com o valor final: '))
3
4 soma = 0
5 while num1 <= num2:
6     if num1 % 2 == 0:
7         soma = soma + num1
8     num1 = num1 + 1
9 print(soma)
```

```
1 num1 = int(input('Entre com o valor inicial: '))
2 num2 = int(input('Entre com o valor final: '))
3
4 soma = 0
5 continua = True
6 while continua:
7     if num1 % 2 == 0:
8         soma = soma + num1
9     num1 = num1 + 1
10
11     if (num1 > num2):
12         continua = False
13 print(soma)
```

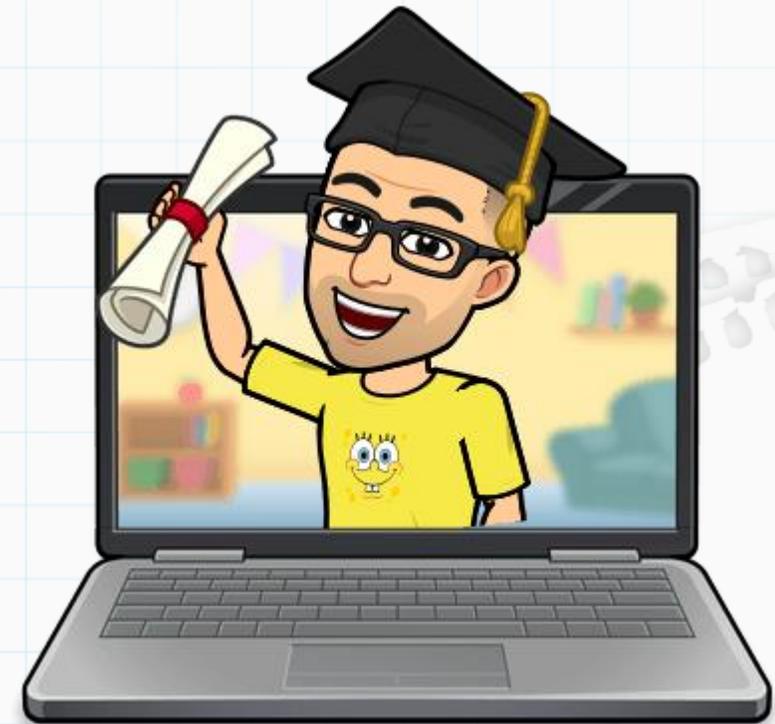
[código](#)

[código](#)



também podemos  
usar uma variável  
lógica para controlar  
o loop

Até a próxima



Slides baseados no curso de Vanessa Braganholo