

ALGORITMOS E LÓGICA DE PROGRAMAÇÃO

AULA 8 – ESTRUTURAS DE REPETIÇÃO

WALTER TRAVASSOS SARINHO

@PROF.WALTERTRAVASSOS

WALTER.TRAVASSOS@SECTRAS.EDU.BR

```
RegisterController.php
You, 7 months ago | 1 author (You)
import VueRouter from "vue-router";
import routes from "../routes/routes";
import store from "../store/index";
import vuexI18n from "vuex-i18n";
import enLangFile from "../lang/en";

// Set config file into the global variable
window.config = require("../vue.config");

// Import bootstrap file
require("../bootstrap");

// Set the globally
const Vue = require("vue");

// Import the VueRouter
const Router = require("../router");

// Import the Vuex
const Vuex = require("vuex");

// Import the Vuex-i18n
const VuexI18n = require("vuex-i18n");

// Import the enLangFile
const enLangFile = require("../lang/en");

// Create the Vue instance
const app = new Vue({
  el: "#app",
  router: Router,
  store: Vuex.createStore({
    modules: {
      i18n: VuexI18n({
        locales: ["en"],
        currentLocale: "en",
        messages: {
          en: enLangFile
        }
      })
    }
  })
});
```

Aula de Hoje

Estruturas de repetição:

- while
- for

E o **do while** (faca enquanto) ?

Em Python, não há uma estrutura de repetição chamada "do while", como em outras linguagens de programação, como C ou Java.

Por que o Guido Van Rossum **não** colocou **chaves** na linguagem python?

- Guido van Rossum, o criador da linguagem Python, **optou por utilizar a indentação** como forma de definir blocos de código em vez de chaves para tornar o código mais legível e mais fácil de escrever.
- A ideia era que a falta de chaves obrigaria os programadores a formatar o código de uma maneira consistente e legível. Em vez de ter que se preocupar em digitar chaves corretamente em torno de cada bloco de código, os programadores poderiam simplesmente adicionar uma única indentação para começar um bloco e uma desindentação para terminar o bloco.
- Essa abordagem também torna o código mais compacto e visualmente mais limpo, permitindo que os programadores se concentrem mais no conteúdo do código do que em sua sintaxe. No entanto, é importante lembrar que a indentação correta é crucial em Python, já que ela define a estrutura do código e pode afetar seu comportamento

while

enquanto

while

- Estrutura de repetição que efetua um **loop** (um laço, uma repetição) em um bloco de instruções **enquanto** uma **condição** for verdadeira.
- Trata-se de uma estrutura de repetição baseada em condição.
- Similar a estrutura **enquanto** utilizada no Portugol Studio.

while

Sintaxe:

```
while condição:  
    # Todos os comandos que serão  
    # executados enquanto a condição  
    # da estrutura for verdadeira
```

OBS.: Todos os comandos a serem delimitados pela estrutura deverão estar indentados! Uma indentação poderá ser inserida em uma linha de código a partir da tecla TAB.

Exemplo 1

- Faça um programa que lê repetidamente vários números inteiros e os escreve na tela.
- O mesmo deverá repetir este procedimento enquanto números diferentes de zero forem digitados.
- Quando for digitado o número zero, escreva a palavra “fim”.

```
Digite um número:
9
9
Digite um número: 10
10
Digite um número: 90
90
Digite um número: 5
5
Digite um número: 50
50
Digite um número: 0
fim
```

Exemplo 1 - Resposta

```
1  numero = int (input("Digite um número: \n"))
2
3  while numero !=0:
4      print(numero)
5      numero = int (input("Digite um número: "))
6
7  print("fim")
```

Exemplo 2

- Faça um programa que sempre repetirá a frase “**Você não sabe a senha! =P**” enquanto o usuário não digitar a senha “**naodigo**”.
- Quando a senha “**naodigo**” for digitada, escreva a mensagem “**acesso concedido**”

```
Digite a senha:  
teste  
Você não sabe a senha =P  
Digite a senha:  
12345seis  
Você não sabe a senha =P  
Digite a senha:  
naodigo  
Acesso concedido
```

Exemplo 2 - Resposta

```
1  senha = input("Digite a senha: \n")
2
3  while senha != "naodigo":
4      print("Você não sabe a senha =P")
5      senha = input("Digite a senha: \n")
6
7  print("Acesso concedido")
8
```

Exemplo 3

- Faça um programa que sempre recebe números inteiros e conta quantos números foram iguais a 2.
- Caso o usuário digite o número 0, o programa deverá parar de receber valores e exibir quantos números foram iguais a 2.

```
Digite um número:  
10  
Digite um número:  
2  
Digite um número:  
3  
Digite um número:  
2  
Digite um número:  
0  
0 total de número 2 foi 2
```

Exemplo 3 - Resposta

```
1  numero = int(input("Digite um número: \n"))
2  cont = 0
3
4  while numero != 0:
5      if numero == 2:
6          cont = cont + 1
7          # cont += 1    //funciona
8          # cont++      //não funciona, erro de sintaxe
9          numero = int(input("Digite um número: \n"))
10
11  print("O total de número 2 foi ", cont)
```

Existe a operação **++** ou **--** em python?

- Em Python, não existe a operação **++** ou **--** como em algumas outras linguagens de programação. Esses operadores são chamados de operadores de **incremento** e **decremento**, e são usados para aumentar ou diminuir o valor de uma variável em uma unidade.
- No entanto, podemos utilizar os operadores de atribuição compostos: **i+=1**

Exercícios

Exercício 1

Utilizando a estrutura de repetição while, faça um programa que escreva todos os números entre 0 e 100 (incluindo o 100).

Exercício 1 - Resposta

```
1  n = 0
2
3  while n<=100:
4      print(n)
5      n+=1
```

Exercício 2

- Faça um programa que lê diversos números positivos e, na sequência, escreve o dobro do número digitado.
- Quando um número negativo for digitado (menor que zero), o programa deverá parar de ler números.

```
>_ Console v x Shell x
Digite um número
5
10
Digite um número
6
12
Digite um número
-1
Digitou número negativo
> |
```

Exercício 2 - Resposta

```
1 n = int(input("Digite um número\n"))
2
3 while n >= 0:
4     print(n*2)
5     n = int(input("Digite um número\n"))
6
7 print("Digitou número negativo")
```

```
>_ Console v x Shell x
Digite um número
5
10
Digite um número
6
12
Digite um número
-1
Digitou número negativo
> |
```

Exercício 3

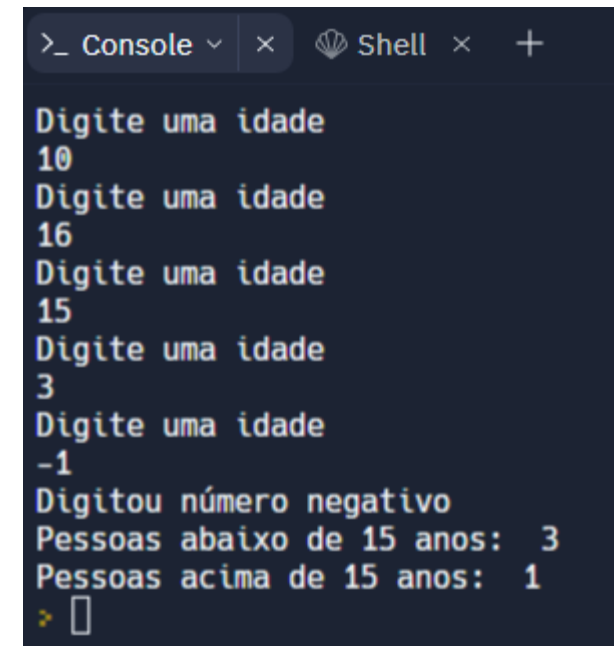
- Faça um programa que lê a idade de várias pessoas, até que uma idade negativa seja digitada.
- O algoritmo deverá calcular e exibir a quantidade de pessoas, de acordo com as faixas etárias apresentadas na tabela abaixo:

FAIXA ETÁRIA	IDADE
1 ^a	≤ 15 anos
2 ^a	Acima de 15 anos

```
>_ Console v x Shell x +
Digite uma idade
10
Digite uma idade
16
Digite uma idade
15
Digite uma idade
3
Digite uma idade
-1
Digitou número negativo
Pessoas abaixo de 15 anos: 3
Pessoas acima de 15 anos: 1
>_
```

Exercício 3 - Resposta

```
1  idade = int(input("Digite uma idade\n"))
2  cont1 = 0
3  cont2 = 0
4
5  while idade >= 0:
6      if idade <= 15:
7          cont1 = cont1 + 1
8      else:
9          cont2 = cont2 + 1
10
11     idade = int(input("Digite uma idade\n"))
12
13     print("Digitou número negativo")
14     print("Pessoas abaixo de 15 anos: ", cont1)
15     print("Pessoas acima de 15 anos: ", cont2)
```

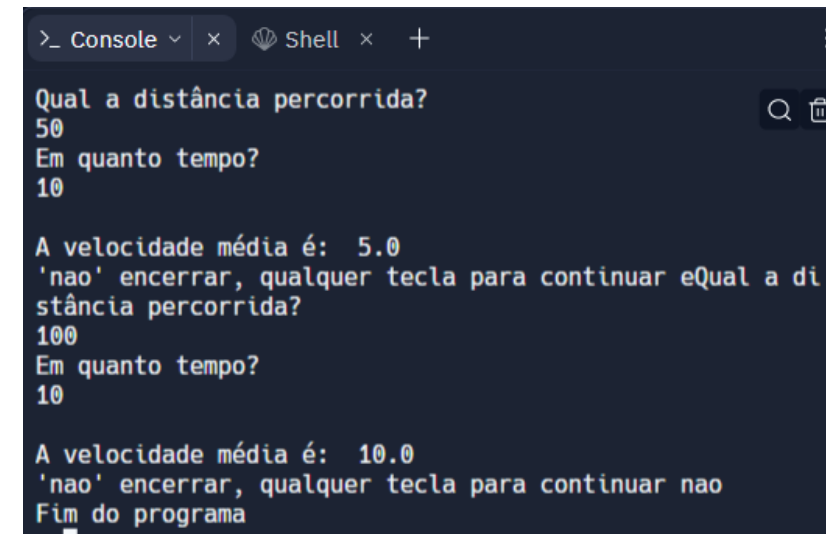


```
>_ Console × Shell × +
Digite uma idade
10
Digite uma idade
16
Digite uma idade
15
Digite uma idade
3
Digite uma idade
-1
Digitou número negativo
Pessoas abaixo de 15 anos: 3
Pessoas acima de 15 anos: 1
> █
```

Exercício 4

Faça um programa que repita as seguintes tarefas, até que a palavra 'nao' seja digitada:

1. Leia a distância percorrida por um atleta.
2. Leia o tempo que o atleta levou para percorrer a distância.
3. Calcule e exiba sua velocidade média:
$$\text{Velocidade} = \text{distancia} / \text{tempo}$$
4. Pergunte ao usuário se o mesmo quer continuar a executar o programa ou digitar 'nao' para sair.



```
>_ Console x Shell x +
Qual a distância percorrida?
50
Em quanto tempo?
10
A velocidade média é: 5.0
'nao' encerrar, qualquer tecla para continuar eQual a di
stância percorrida?
100
Em quanto tempo?
10
A velocidade média é: 10.0
'nao' encerrar, qualquer tecla para continuar nao
Fim do programa
```

Exercício 4 - Resposta

```
1  resposta="sim"
2
3  while resposta!="nao":
4      distancia = int(input("Qual a distância percorrida?\n"))
5      tempo = int(input("Em quanto tempo?\n"))
6      print("\nA velocidade média é: ", (distancia/tempo ))
7      resposta = input("'nao' encerrar, qualquer tecla para continuar ")
8
9  print("Fim do programa")
```

```
>_ Console x Shell x +
Qual a distância percorrida?
50
Em quanto tempo?
10
A velocidade média é: 5.0
'nao' encerrar, qualquer tecla para continuar eQual a di
stância percorrida?
100
Em quanto tempo?
10
A velocidade média é: 10.0
'nao' encerrar, qualquer tecla para continuar nao
Fim do programa
```

Exercício 5

Faça um programa que repita as seguintes tarefas, até que o código 0 seja digitado:

1. Leia o código do produto.
2. Leia a quantidade adquirida.
3. Se o código for:
 - 1 - escreva 'Caderno – R\$ 12.00'
 - 2 - escreva 'Régua – R\$ 2.50'
 - 3 - escreva 'Borracha – R\$ 0.25'
 - 4 - escreva 'Mochila – R\$ 50.00'
4. Calcule e exiba o total a ser pago (valor * quantidade).

```
Qual o código do produto?  
1  
Qual a quantidade?  
5  
Caderno - R$ 12.00 und. Total: R$ 60  
Qual o código do produto?  
5  
Qual a quantidade?  
6  
Código inválido  
Qual o código do produto?  
0  
Qual a quantidade?  
0  
Código inválido  
Fim da transação
```

Exercício 5 - Resposta

```
1  cod = -1
2  while cod!=0:
3      cod = int(input("Qual o código do produto?\n"))
4      qtd = int(input("Qual a quantidade?\n"))
5
6      if cod==1:
7          print("Caderno - R$ 12.00 und. Total: R$ ",(12*qtd))
8      elif cod==2:
9          print("Régua - R$ 2.50 und. Total: R$ ", (2.5*qtd))
10     elif cod==3:
11         print("Borracha - R$ 0.25 und. Total: R$ ", (0.25*qtd))
12     elif cod==4:
13         print("Mochila - R$ 50 und. Total: R$ ", (50*qtd))
14     else:
15         print("Código inválido")
16
17 print("Fim da transação")
```

```
Qual o código do produto?
1
Qual a quantidade?
5
Caderno - R$ 12.00 und. Total: R$ 60
Qual o código do produto?
5
Qual a quantidade?
6
Código inválido
Qual o código do produto?
0
Qual a quantidade?
0
Código inválido
Fim da transação
```

Exercício 6

Faça um programa que receba vários números positivos (enquanto o número 0 não for digitado).

O programa deverá exibir o maior número digitado.

```
Digite um número: 1
Digite um número: 5
Digite um número: 10
Digite um número: 50
Digite um número: 0
Digitou zero
O maior número digitado foi 50
> █
```

Exercício 6 - Resposta

```
1 num = int(input("Digite um número: "))
2 maior = -1
3 while num!=0:
4     if num>maior:
5         maior = num
6     num = int(input("Digite um número: "))
7
8 print("Digitou zero")
9 print("O maior número digitado foi ", maior)
```

```
Digite um número: 1
Digite um número: 5
Digite um número: 10
Digite um número: 50
Digite um número: 0
Digitou zero
O maior número digitado foi 50
> █
```

Exercício 7

- Faça um programa que lê a idade de várias pessoas (enquanto o usuário digitar valores positivos).
- Em seguida, o algoritmo deverá apresentar a quantidade de adolescentes (de 12 a 17 anos).

```
Digite uma idade: 5
Digite uma idade: 12
Digite uma idade: 15
Digite uma idade: 5
Digite uma idade: 16
Digite uma idade: 18
Digite uma idade: 19
Digite uma idade: -1
idade inválida
Número de adolescentes: 3
```

Exercício 7 - Resposta

```
1  idade = int(input("Digite uma idade: "))
2  cont = 0
3  while idade >= 0:
4      if idade >= 12 and idade <= 17:
5          cont += 1
6          idade = int(input("Digite uma idade: "))
7      print("idade inválida")
8      print("Número de adolescentes: ", cont)
9
```

```
Digite uma idade: 5
Digite uma idade: 12
Digite uma idade: 15
Digite uma idade: 5
Digite uma idade: 16
Digite uma idade: 18
Digite uma idade: 19
Digite uma idade: -1
idade inválida
Número de adolescentes: 3
```

for

para

for

- Nesta estrutura, uma variável de controle é inicializada com um valor. Para cada iteração, seu valor é comparado com um valor final.
- A execução da lista de instruções **se repetirá** até que a variável de controle seja maior que o valor final.
- Para cada iteração, a variável de controle é incrementada.
- Similar a estrutura **para** utilizada no **Portugol Studio**.

for

- Sintaxe:

```
for variável in range (início, fim):  
    # Todos os comandos que serão  
    # executados o valor da variável  
    # estiver na faixa (início, fim)
```

- OBS.: Todos os comandos a serem delimitados pela estrutura deverão estar indentados! Uma indentação poderá ser inserida em uma linha de código a partir da tecla TAB.

Exemplo 1

Utilizando a estrutura de repetição **for**, faça um algoritmo que escreva os números entre 1 e 5 (incluindo-os).

Exemplo 1 - Resposta

```
for x in range(1, 6):  
    print(x)
```

O que acontece quando usamos apenas um argumento no range?

```
for x in range(10):  
    print (x)
```

0
1
2
3
4
5
6
7
8
9

for

Para alterarmos a forma de contagem do laço (crescente ou decrescente) e quanto a variável de controle será incrementada ou decrementada, inserimos um terceiro parâmetro no range da estrutura (para definir o valor de passo):

```
for variável in range (início, fim, passo):  
    # Todos os comandos que serão  
    # executados o valor da variável  
    # estiver na faixa (início, fim)
```

Exemplo 2

Utilizando a estrutura de repetição **for**, faça um programa que escreva os números pares entre 1 e 10 (incluindo o 10).

Exemplo 2 - Resposta

```
for i in range(2, 11, 2):  
    print(i)
```

Exercícios

Exercício 1

Utilizando a estrutura de repetição **for**, faça um programa que apresente todos os números ímpares entre 0 a 20.

1
3
5
7
9
11
13
15
17
19

Exercício 1 - Resposta

```
1 ✓ for i in range (1, 20, 2):  
2     print(i)
```

```
1  
3  
5  
7  
9  
11  
13  
15  
17  
19
```

Exercício 2

Faça um programa que lê 10 números, calcula e exhibe:

- O triplo de cada número.
- A mensagem “É positivo”, caso o número digitado seja positivo, ou “É negativo”, caso seja negativo.

```
1
Digite um número: -9
O triplo é: -27
É negativo
2
Digite um número: -8
O triplo é: -24
É negativo
3
Digite um número: 8
O triplo é: 24
É positivo
4
Digite um número: █
```

Exercício 2 - Resposta

```
1 v for i in range (1, 11):
2     print(i)
3     num = int(input("Digite um número: "))
4     print("O triplo é: ", (num*3))
5 v     if num>=0:
6         print("É positivo")
7 v     else:
8         print("É negativo")
```

```
1
Digite um número: -9
O triplo é: -27
É negativo
2
Digite um número: -8
O triplo é: -24
É negativo
3
Digite um número: 8
O triplo é: 24
É positivo
4
Digite um número: █
```

Exercício 3

- Faça um programa que lê a resposta de 15 usuários para a seguinte pergunta: “Você gosta de beterraba? Digite 1 para SIM e 2 para NÃO”.
- Após a coleta da resposta de cada usuário, o programa deverá exibir a quantidade de respostas para cada opção.

```
1 para sim, 2 para não1  
Votou sim  
1 para sim, 2 para não2  
Votou não  
1 para sim, 2 para não1  
Votou sim  
1 para sim, 2 para não2  
Votou não  
1 para sim, 2 para não1  
Votou sim
```

Exercício 3 - Resposta

```
1 print("Enquete: Você gosta de beterraba?")
2 cont1 = 0
3 cont2 = 0
4
5 for x in range(15):
6     voto = int(input("1 para sim, 2 para não"))
7     if voto==1:
8         print("Votou sim")
9         cont1+=1
10    elif voto==2:
11        print("Votou não")
12        cont2+=1
13    else:
14        print("Voto inválido")
15
16 print("Resultado:")
17 print("Votou Sim: ", cont1)
18 print("Votou Não: ", cont2)
```

```
1 para sim, 2 para não1
Votou sim
1 para sim, 2 para não2
Votou não
1 para sim, 2 para não1
Votou sim
1 para sim, 2 para não2
Votou não
1 para sim, 2 para não1
Votou sim
```

Exercício 4

Faça um programa que recebe o preço de 10 produtos e exibe o valor do produto mais caro.

```
Qual o preço do produto?5
Qual o preço do produto?6
Qual o preço do produto?5
Qual o preço do produto?8
Qual o preço do produto?90
Qual o preço do produto?100
Qual o preço do produto?10000
Qual o preço do produto?5
Qual o preço do produto?6
Qual o preço do produto?8
O maior valor foi 10000
```

Exercício 4 - Resposta

```
1 maior = -1;
2 for x in range(10):
3     valor = int (input("Qual o preço do produto?"))
4     if valor>maior:
5         maior = valor
6
7 print("O maior valor foi ", maior)
```

```
Qual o preço do produto?5
Qual o preço do produto?6
Qual o preço do produto?5
Qual o preço do produto?8
Qual o preço do produto?90
Qual o preço do produto?100
Qual o preço do produto?10000
Qual o preço do produto?5
Qual o preço do produto?6
Qual o preço do produto?8
O maior valor foi 10000
```

Exercício 5

Cada espectador de um cinema respondeu a um questionário no qual constava sua opinião em relação ao filme: 3 para ótimo, 2 para bom e 1 para regular.

Faça um programa que receba a opinião de 5 espectadores, calcule e mostre:

- A quantidade de pessoas que responderam ótimo.
- A quantidade de pessoas que responderam bom.
- A quantidade de pessoas que responderam regular.

```
Enquete para o filme
1 - regular, 2 - bom, 3 - ótimo5
Voto inválido
1 - regular, 2 - bom, 3 - ótimo1
Votou regular
1 - regular, 2 - bom, 3 - ótimo2
Votou bom
1 - regular, 2 - bom, 3 - ótimo3
Votou ótimo
1 - regular, 2 - bom, 3 - ótimo1
Votou regular
Resultado
Total regular: 2
Total bom: 1
Total ótimo: 1
```

Exercício 5 - Resposta

```
1 cont1 = 0; cont2 = 0; cont3 = 0
2 print("Enquete para o filme")
3 for x in range(5):
4     voto = int(input("1 - regular, 2 - bom, 3 - ótimo"))
5     if voto==1:
6         cont1+=1
7         print("Votou regular")
8     elif voto==2:
9         cont2+=1
10        print("Votou bom")
11    elif voto==3:
12        cont3+=1
13        print("Votou ótimo")
14    else:
15        print("Voto inválido")
16
17 print("Resultado")
18 print("Total regular: ", cont1)
19 print("Total bom: ", cont2)
```

```
Enquete para o filme
1 - regular, 2 - bom, 3 - ótimo5
Voto inválido
1 - regular, 2 - bom, 3 - ótimo1
Votou regular
1 - regular, 2 - bom, 3 - ótimo2
Votou bom
1 - regular, 2 - bom, 3 - ótimo3
Votou ótimo
1 - regular, 2 - bom, 3 - ótimo1
Votou regular
Resultado
Total regular:  2
Total bom:  1
Total ótimo:  1
```

ALGORITMOS E LÓGICA DE PROGRAMAÇÃO

AULA 8 – ESTRUTURAS DE REPETIÇÃO

WALTER TRAVASSOS SARINHO

@PROF.WALTERTRAVASSOS

WALTER.TRAVASSOS@SECTRAS.EDU.BR

```
RegisterController.php
You, 7 months ago | 1 author (You)
import VueRouter from "vue-router";
import routes from "../routes/routes";
import store from "../store/index";
import vuexI18n from "vuex-i18n";
import enLangFile from "../lang/en";

// Set config file into the global variable
window.config = require("../vue.config");

// Import bootstrap file
require("../bootstrap");

// Set the globally
const Vue = require("vue");

// Import the VueRouter
const Router = require("../router");

// Import the Vuex
const Vuex = require("vuex");

// Import the Vuex-i18n
const VuexI18n = require("vuex-i18n");

// Import the enLangFile
const enLangFile = require("../lang/en");

// Create the Vue instance
const app = new Vue({
  el: "#app",
  router: Router,
  store: Vuex.createStore({
    modules: {
      i18n: VuexI18n({
        locales: ["en"],
        currentLocale: "en",
        messages: {
          en: enLangFile
        }
      })
    }
  })
});
```