



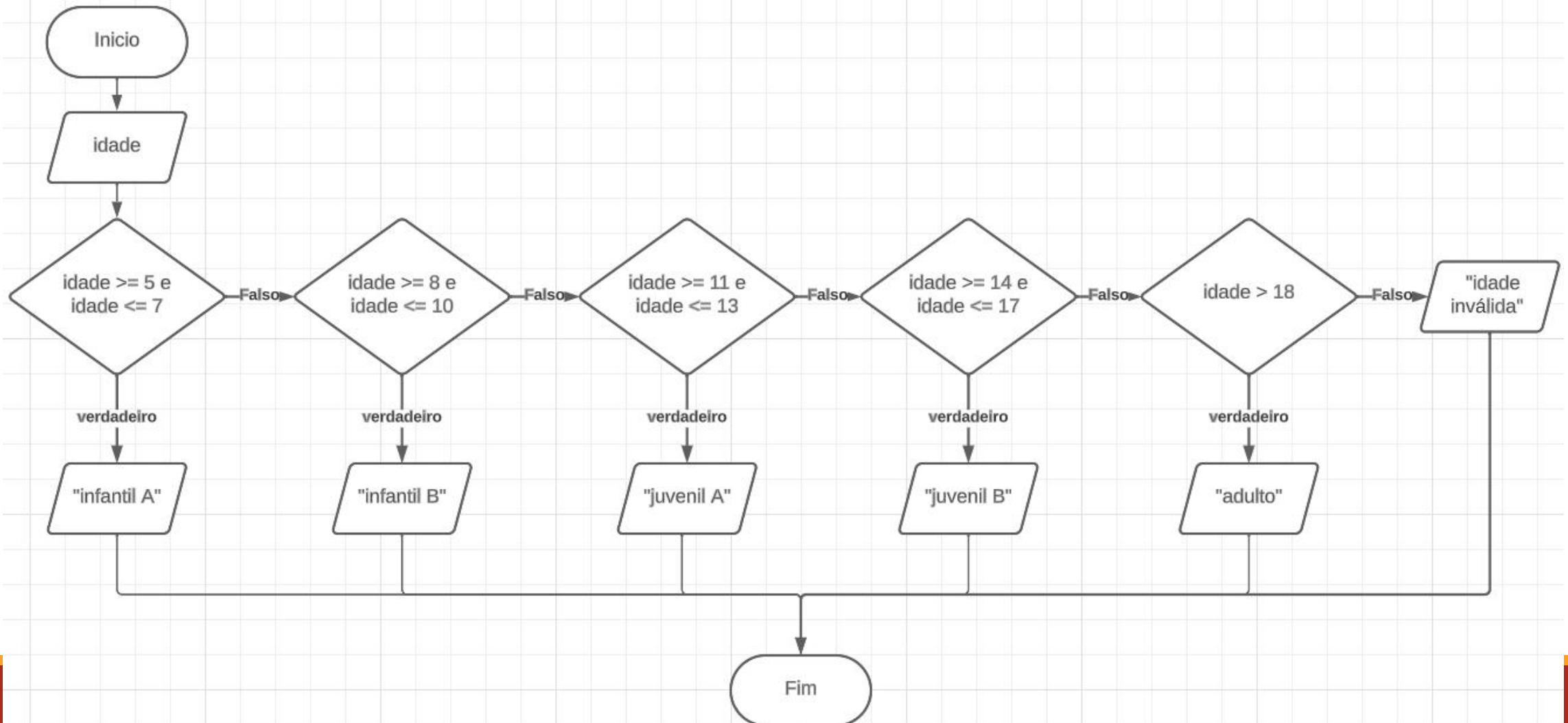
Programação I

LUCAS SAMPAIO LEITE

E os nossos exercícios?

1. Elabore um fluxograma para representar um algoritmo que dada a idade de um nadador classifica-o em uma das seguintes categorias:
 - a) infantil A = 5 - 7 anos
 - b) infantil B = 8 -10 anos
 - c) juvenil A = 11-13 anos
 - d) juvenil B = 14-17 anos
 - e) adulto = maiores de 18 anos

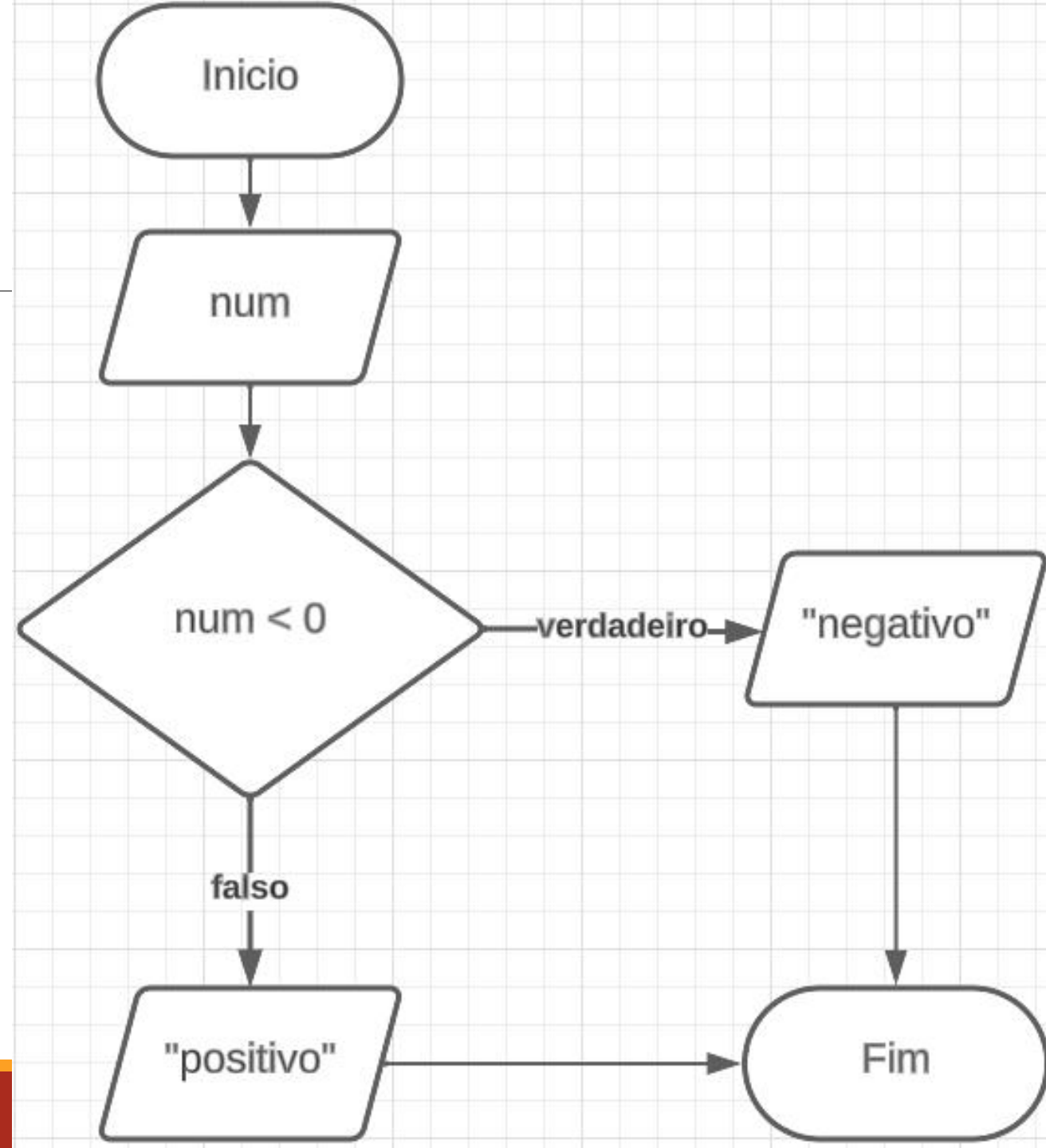
Correção 1:



E os nossos exercícios?

2. Crie um algoritmo que leia um número diferente de zero e diga se este número é positivo ou negativo.
3. Faça um algoritmo que leia a idade de uma pessoa expressa em anos, meses e dias e mostre-a expressa em dias. Leve em consideração o ano com 365 dias e o mês com 30. (Ex: 3 anos, 2 meses e 15 dias = 1170 dias.)
4. Crie um algoritmo que recebe 2 números e multiplica o num1 pelo num2 através de somas repetidas. (ex: 2 e 3 = $2 + 2 + 2$)
5. Crie um algoritmo representado por um fluxograma para exibir todos os números pares de 1 a 100.

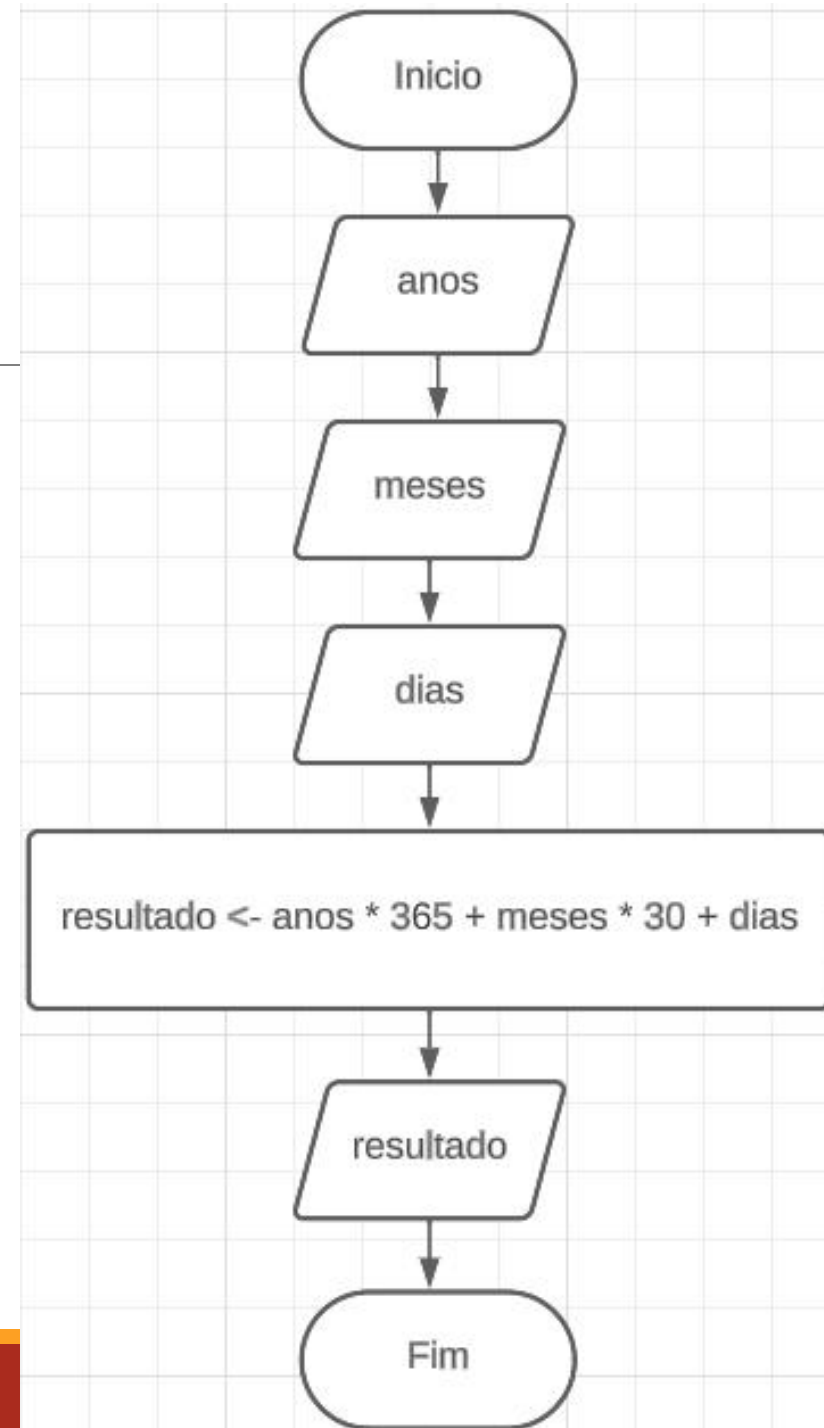
Correção 2:



Correção 2:

```
1 algoritmo "positivo_negativo"
2 var
3     numero: inteiro
4 inicio
5     leia(numero)
6     se numero > 0 entao
7         escreva("O número é positivo.")
8     senao se numero < 0 entao
9         escreva("O número é negativo.")
10    senao
11        escreva("O número é zero.")
12    fim se
13 fim
```

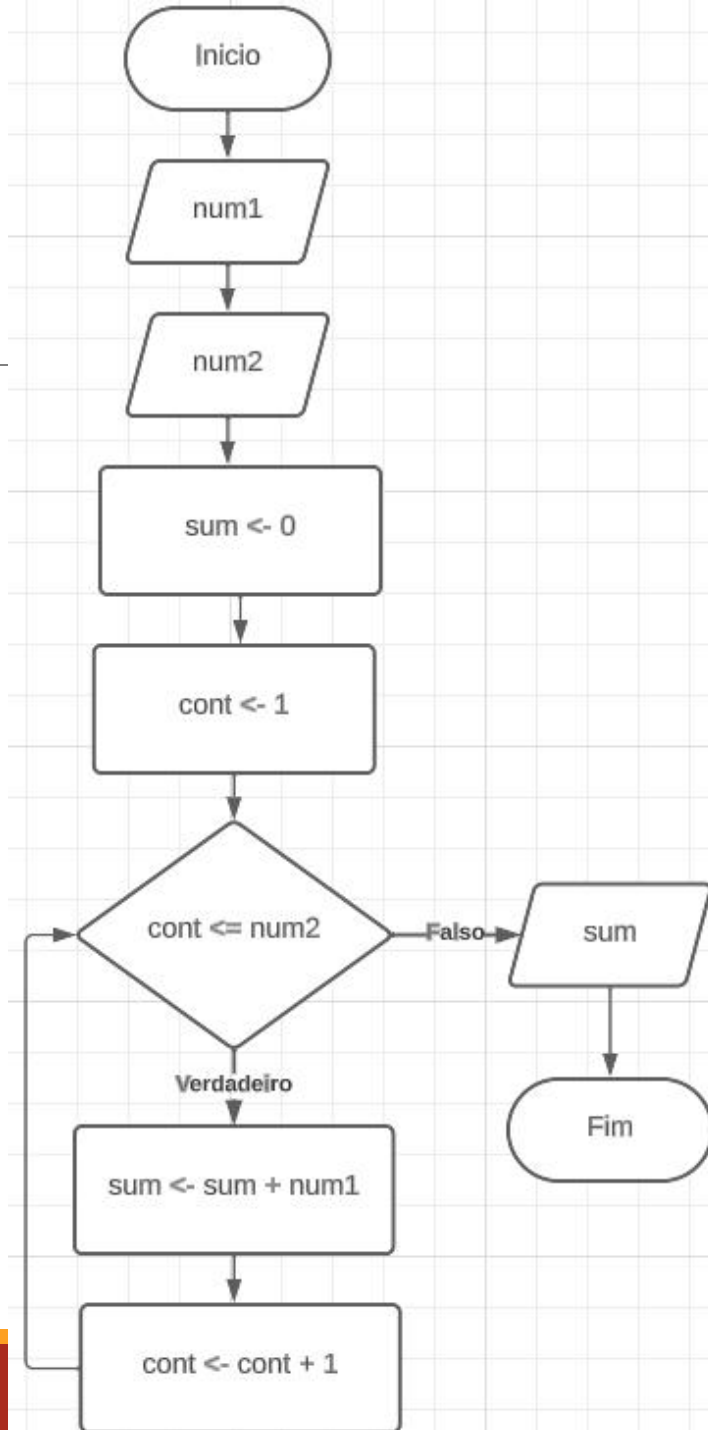
Correção 3:



Correção 3:

```
1 Algoritmo "idade"
2 var
3     anos, meses, dias, resultado: inteiro
4 inicio
5     escreva("Digite o número de anos da idade: ")
6     leia(anos)
7     escreva("Digite o número de meses da idade: ")
8     leia(meses)
9     escreva("Digite o número de dias da idade: ")
10    leia(dias)
11    resultado <- (anos * 365)+(meses * 30)+dias
12    escreva("A idade em dias é: ", resultado)
13 fim
```


Correção 4:



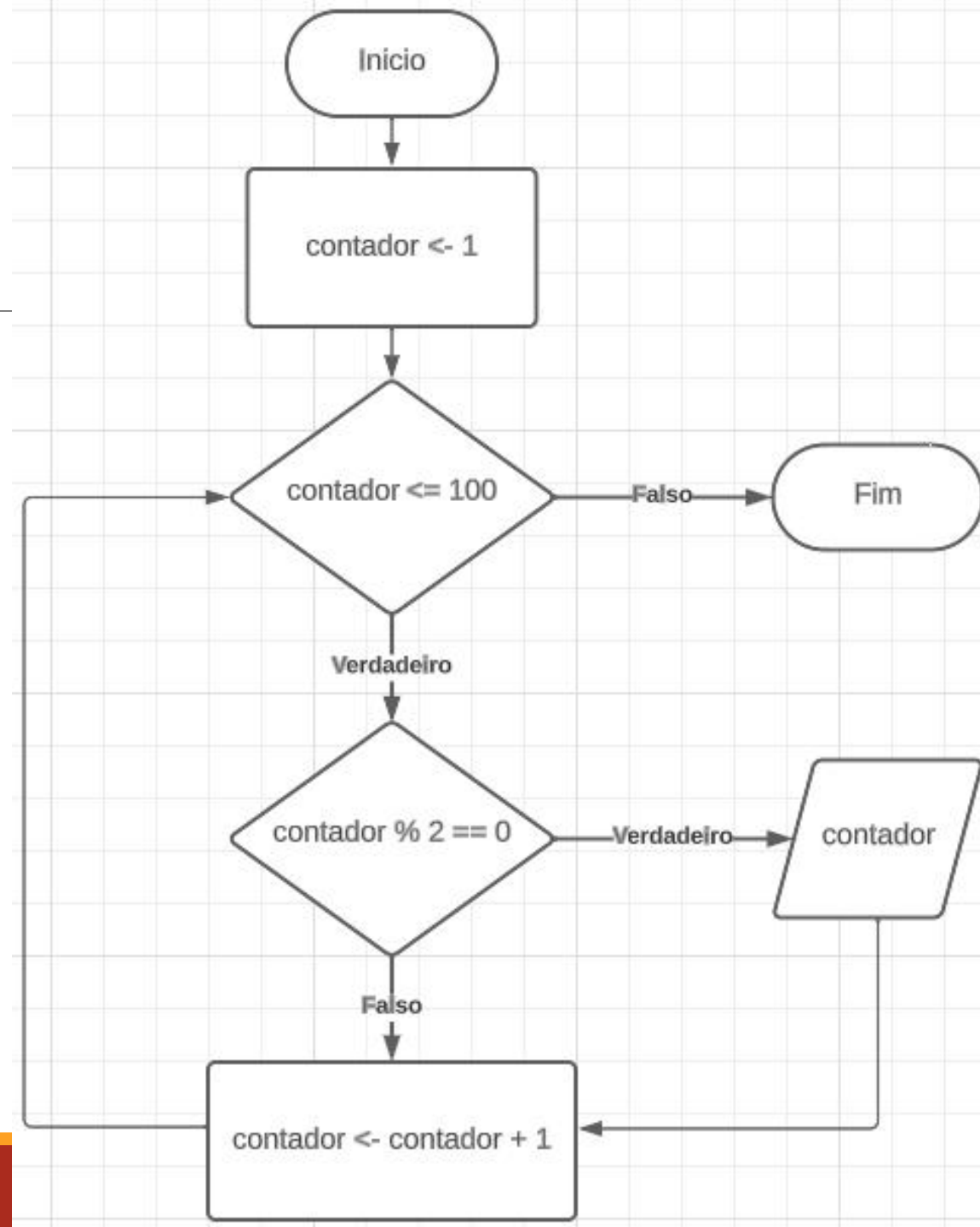
Correção 4:

```
1 algoritmo "mult"
2 var
3     num1, num2, resultado, contador: inteiro
4 inicio
5     escreva("Digite o primeiro número:")
6     leia(num1)
7     escreva("Digite o segundo número:")
8     leia(num2)
9     resultado <- 0
10    contador <- 0
11    enquanto contador < num2 faça
12        resultado <- resultado + num1
13        contador <- contador + 1
14    fim enquanto
15 fim
```

Correção 4:

```
1 algoritmo "mult"
2 var
3     num1, num2, resultado: inteiro
4 inicio
5     escreva("Digite o primeiro número:")
6     leia(num1)
7     escreva("Digite o segundo número:")
8     leia(num2)
9     resultado <- 0
10    enquanto num2 > 0 faça
11        resultado <- resultado + num1
12        num2 <- num2 - 1
13    fim enquanto
14 fim
```

Correção 5:



E os nossos exercícios?

6. João papo-de-pescador, comprou um microprocessador para controlar o rendimento diário do seu trabalho. Toda vez que ele traz um peso de peixes maior que o estabelecido pelo regulamento de pesca do estado de São Paulo (50 quilos) deve pagar uma multa de R\$ 4,00 por quilo excedente. João precisa que você faça um algoritmo que leia a variável “p” (peso de peixes) e verifique se há excesso. Se houver, gravar na variável “e” (excesso) o excesso e na variável “m” o valor da multa que João deverá pagar. Caso contrário mostrar tais variáveis com o conteúdo “0”.
7. A secretária de Meio Ambiente que controla o índice de poluição mantém 3 grupos que são altamente poluentes do meio ambiente. O índice de poluição aceitável varia de 0,05 até 0,25. Se o índice sobe para 0,3 as indústrias do 1º grupo são intimadas a suspenderem suas atividades, se o índice crescer para 0,4 as indústrias do 1º e 2º grupo são intimadas a suspenderem suas atividades, se o índice atingir 0,5 todos os grupos devem ser notificados a paralisarem suas atividades. Faça um algoritmo que leia o índice de poluição medido e emita a notificação adequada aos diferentes grupos de empresas.

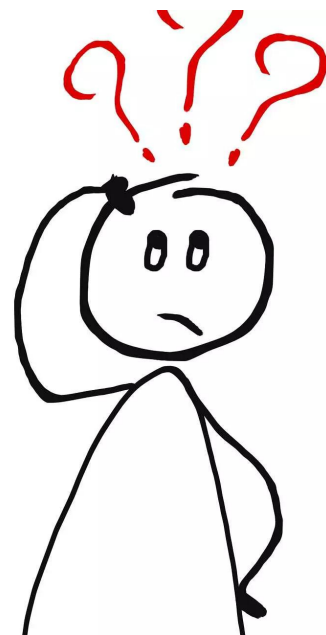
Correção 6:

```
1 algoritmo "pescador"
2 var
3     p, e, m: real
4 inicio
5     escreva("Digite o peso de peixes (em quilos):")
6     leia(p)
7     e <- 0
8     m <- 0
9     se p > 50 entao
10         e <- p - 50
11         m <- e * 4.0
12     fim se
13     escreva("Excesso de peso:", e, "quilos")
14     escreva("Multa a pagar: R$", m)
15 fim
```


Correção 7:

```
1 algoritmo "pescador"
2 var
3     indice: real
4 inicio
5     escreva("Digite o índice de poluição medido:")
6     leia(indice)
7     se indice >= 0.5 entao
8         escreva("Todos os grupos devem paralisar suas atividades.")
9     senao se indice >= 0.4 entao
10        escreva("Indústrias do 1º e 2º grupo devem suspender suas atividades.")
11    senao se indice >= 0.3 entao
12        escreva("Indústrias do 1º grupo devem suspender suas atividades.")
13    senao
14        escreva("Índice de poluição dentro dos limites aceitáveis.")
15    fim se
16 fim
```

O que é um programa?



O que é um programa?

- ❑ Conjunto de instruções que descrevem ações a serem executadas por um computador para resolver um problema ou construir algo;
- ❑ São necessários conhecimentos técnicos de codificação;
- ❑ Um algoritmo é uma abordagem sistemática para resolver um problema específico. Um programa é um conjunto de instruções para um computador seguir.

O que é um programa?

- ❑ Os programas são escritos em uma linguagem de programação;
- ❑ Assim como o português, inglês e o espanhol, possuem regras (sintaxe);
- ❑ Quanto mais linguagens você aprender, mais fácil será aprender novas linguagens (similar com idiomas);
- ❑ Como um programa é criado?
 - ❑ Algoritmo → codificação em uma linguagem de programação → criação do programa (software).

Ambiguidades



Discussão relevante...

- ❑ É comum que se conheça o algoritmo, mas sem conhecer uma linguagem necessária, não será possível criar um programa nessa linguagem:
 - ❑ Exemplo: não sei como construir uma calculadora usando Perl.

Discussão relevante...

- ❑ É comum que se conheça o algoritmo, mas sem conhecer uma linguagem necessária, não será possível criar um programa nessa linguagem:
 - ❑ Exemplo: não sei como construir uma calculadora usando Perl.
- ❑ Também é comum que se conheça a linguagem, mas não o algoritmo, por não conhecer as regras de negócio do que se deseja produzir:
 - ❑ Exemplo: não saberia como construir um programa para decifrar códigos genéticos (em linguagem nenhuma).

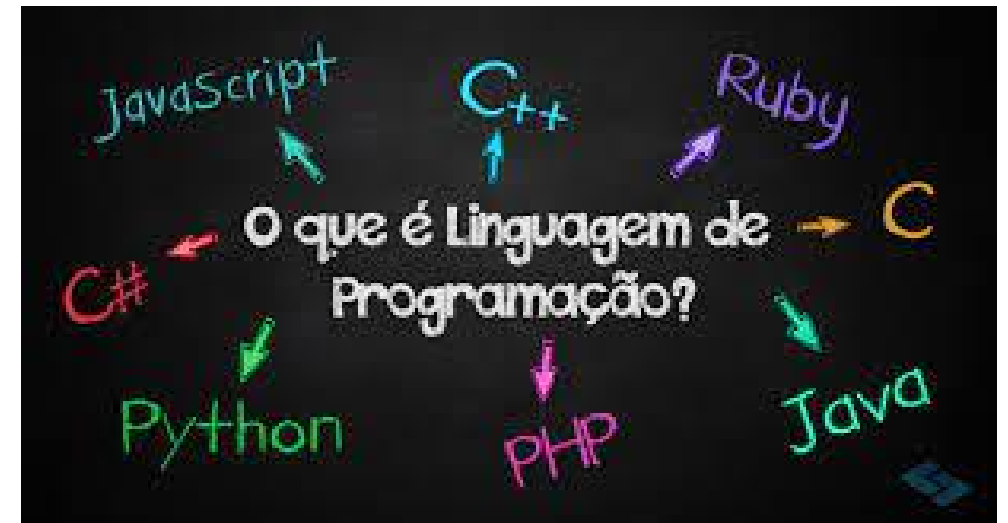
Discussão relevante...

- ❑ É comum que se conheça o algoritmo, mas sem conhecer uma linguagem necessária, não será possível criar um programa nessa linguagem:
 - ❑ Exemplo: não sei como construir uma calculadora usando Perl.
- ❑ Também é comum que se conheça a linguagem, mas não o algoritmo, por não conhecer as regras de negócio do que se deseja produzir:
 - ❑ Exemplo: não saberia como construir um programa para decifrar códigos genéticos (em linguagem nenhuma).

Cuidado para não confundir as coisas e subestimar o seu aprendizado na disciplina!!!

Linguagem de programação

- A linguagem de programação é um método padronizado, formado por um **conjunto de regras sintáticas e semânticas**, de implementação de **um código fonte** - que pode ser compilado e transformado em um programa de computador, ou usado como script interpretado - que informará instruções de processamento ao computador.



Linguagem de programação

- ❑ Sintaxe

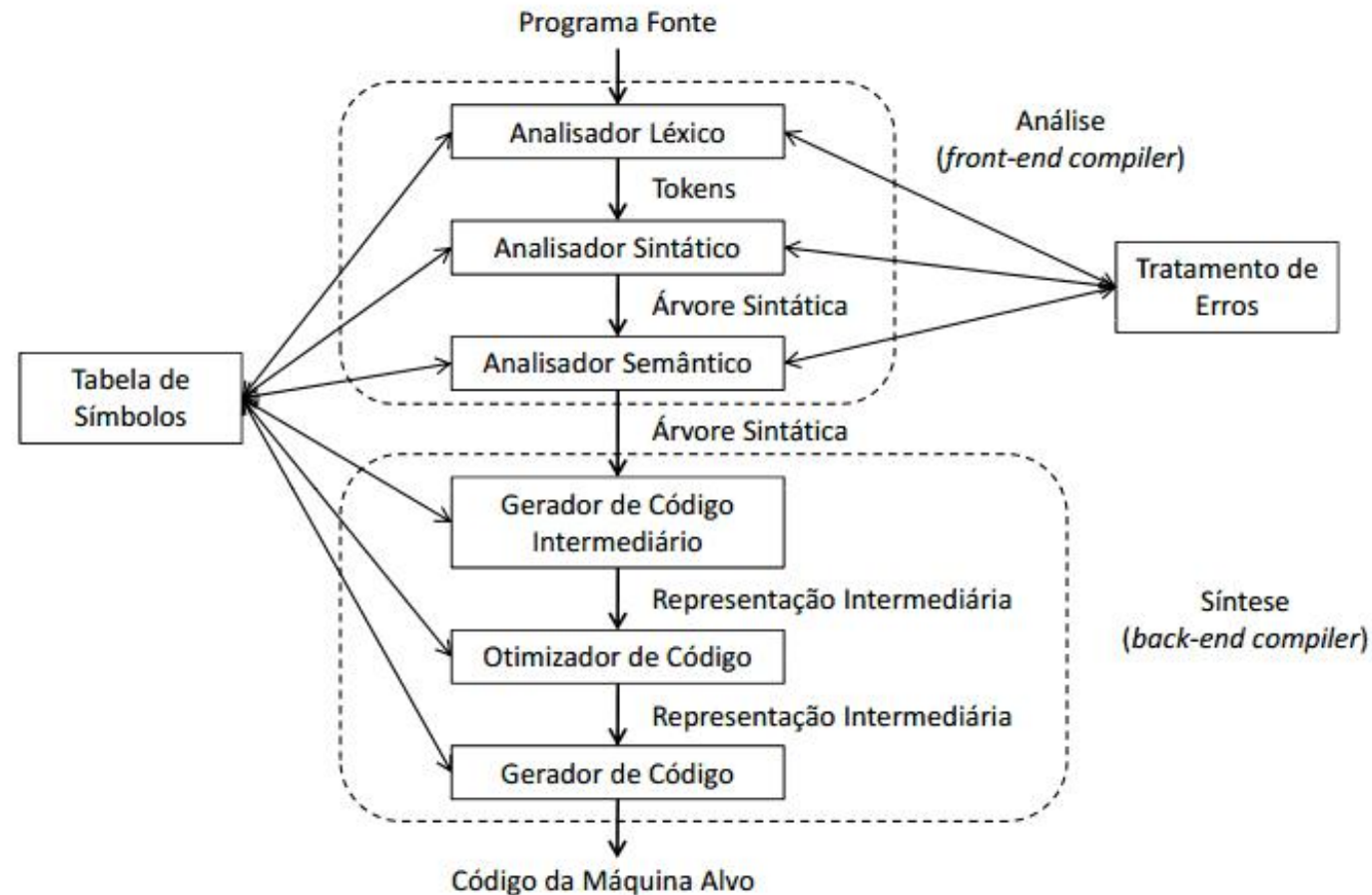
- ❑ Conjunto de regras que definem como um programa pode ser escrito;
 - ❑ Ex: $a = b + c$ está sintaticamente correto?

- ❑ Semântica

- ❑ Define o significado de um programa que esteja sintaticamente correto;
 - ❑ Ex: $a = b + c$ está semanticamente correto?

Um programa pode estar sintaticamente correto, mas semanticamente incorreto.

Linguagem de programação



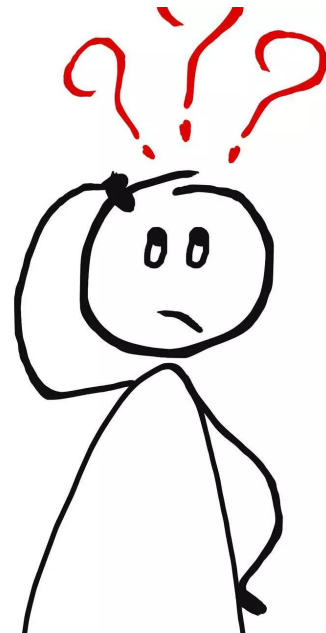
Linguagem de programação

- ❑ Linguagem compilada x interpretada:
 - ❑ A grande diferença entre ambos é a forma como o resultado é gerado.
 - ❑ A interpretação executa o código analisado.
 - ❑ A compilação gera um outro código que será posteriormente usado (executado) por um ambiente que entenda o seu funcionamento.

Linguagem de programação

- ❑ Classificações:
 - ❑ Por paradigmas:
 - ❑ Imperativa/estruturadas;
 - ❑ Funcionais;
 - ❑ Orientadas a objeto;
 - ❑ Multi-paradigmas.
 - ❑ Por nível de abstração:
 - ❑ Linguagem de máquina;
 - ❑ Linguagem de baixo nível;
 - ❑ Linguagem de alto nível.

Por que aprender a programar?



Por que aprender a programar?

- ❑ Possibilidade de automatizar tarefas:
 - ❑ Por meio de um programa uma tarefa pode deixar de ser realizada manualmente para ser resolvida de forma automática.
- ❑ Softwares estão cada vez mais presentes em nossas vidas:
 - ❑ Negócios, automóveis, equipamentos domésticos, wearables, redes sociais, etc.
- ❑ Tendência a estar mais presentes no futuro: inteligência artificial, big data, internet das coisas, etc.

Por que aprender a programar?

- ❑ Programar ensina a resolver problemas:
 - ❑ Diversas estratégias (exemplo: dividir para conquistar);
 - ❑ Cérebro é treinado para pensar a respeito de um problema de forma diferente;
 - ❑ Pensar algoritmicamente;
 - ❑ Requer conhecimentos de lógica, matemática, algoritmos, programação no ambiente específico que se deseja construir o software, além da área de conhecimento em que o programa será aplicado.

Python

- ❑ O Python é uma linguagem de programação de alto nível e interpretada. Portanto, para que se possa programar em Python, é necessário a utilização (junto à IDE escolhida) do interpretador do Python.



Python

- ❑ Python foi lançada por Guido Van Rossum em 1991. Gerenciado pela Python Software Foundation – Modelo de desenvolvimento comunitário e aberto.
- ❑ O nome foi baseado no programa de TV Monty Python's Flying Circus.



Python

- ❑ Um dos principais conceitos que foram levados em conta no projeto da Python é tornar o código dela legível.
- ❑ Para isso o núcleo do projeto era minimalista, com o objetivo de criar um base simples e dar um grande suporte através de bibliotecas.
 - ❑ Cerca de 125.000~137.000 bibliotecas disponíveis.



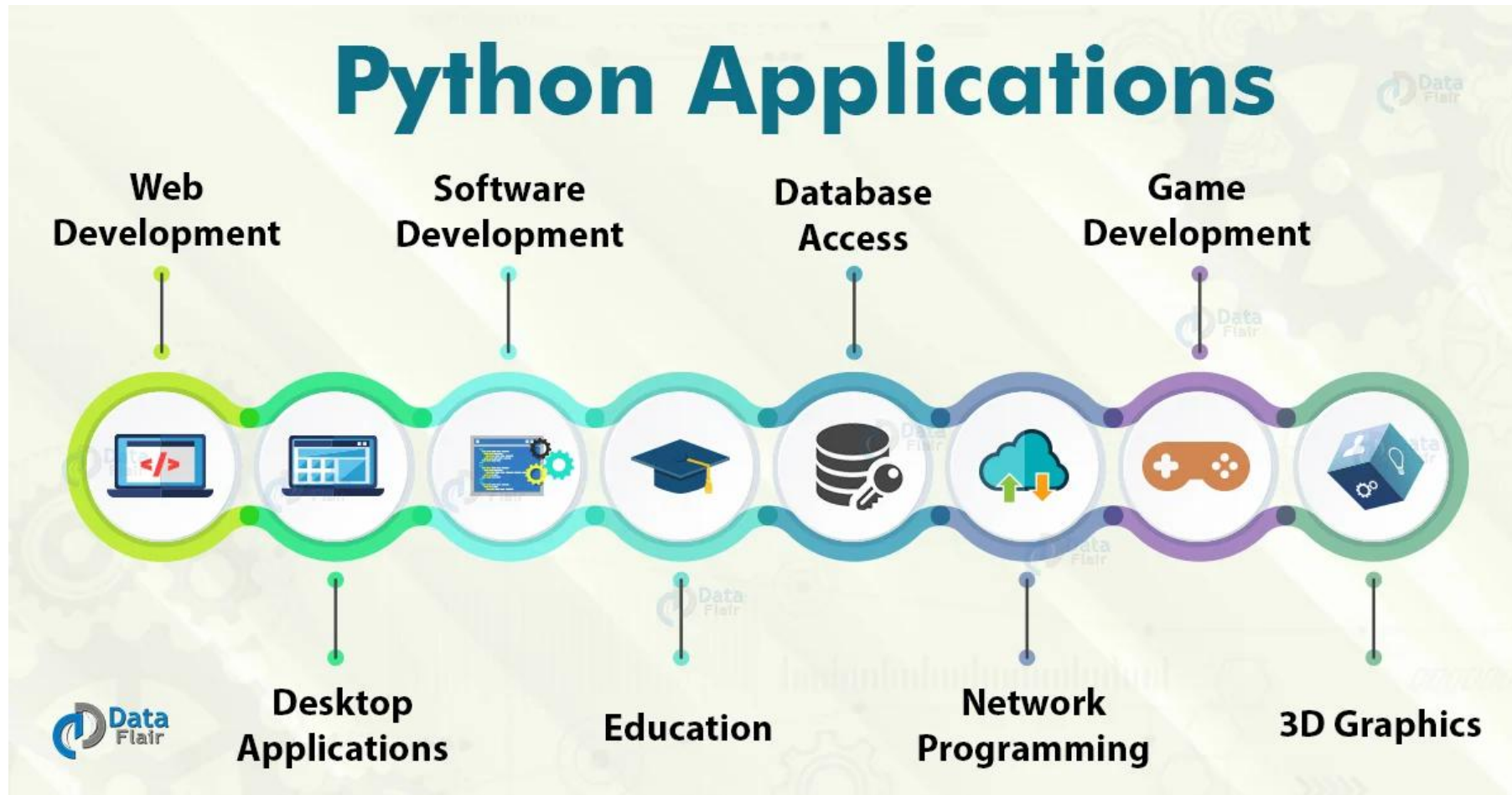
Python (características principais)

- ❑ **Indentação:** a indentação obrigatória, faz com que você tenha que obedecer ao alinhamento padrão para que seu código funcione e seja corretamente interpretado.
- ❑ **Interpretada:** O código gerado pelo desenvolvedor é interpretado para linguagem de máquina somente no momento em que ele é atingido em tempo de execução.

Python (características principais)

- ❑ **Legibilidade:** Ponto tido como fundamental no desenvolvimento da linguagem era ter uma linguagem legível. Isso facilita muito o aprendizado e o desenvolvimento dos programas, pois tenta aproximar a codificação da linguagem natural.
- ❑ **Multiparadigma:** Dar liberdade ao desenvolvedor escolher o paradigma que quisesse é uma realidade da programação em Python. Existe o suporte a *Programação Orientada a Objetos (POO)*, *imperativa* e *funcional*.

Python (aplicações)



Quem usa python?



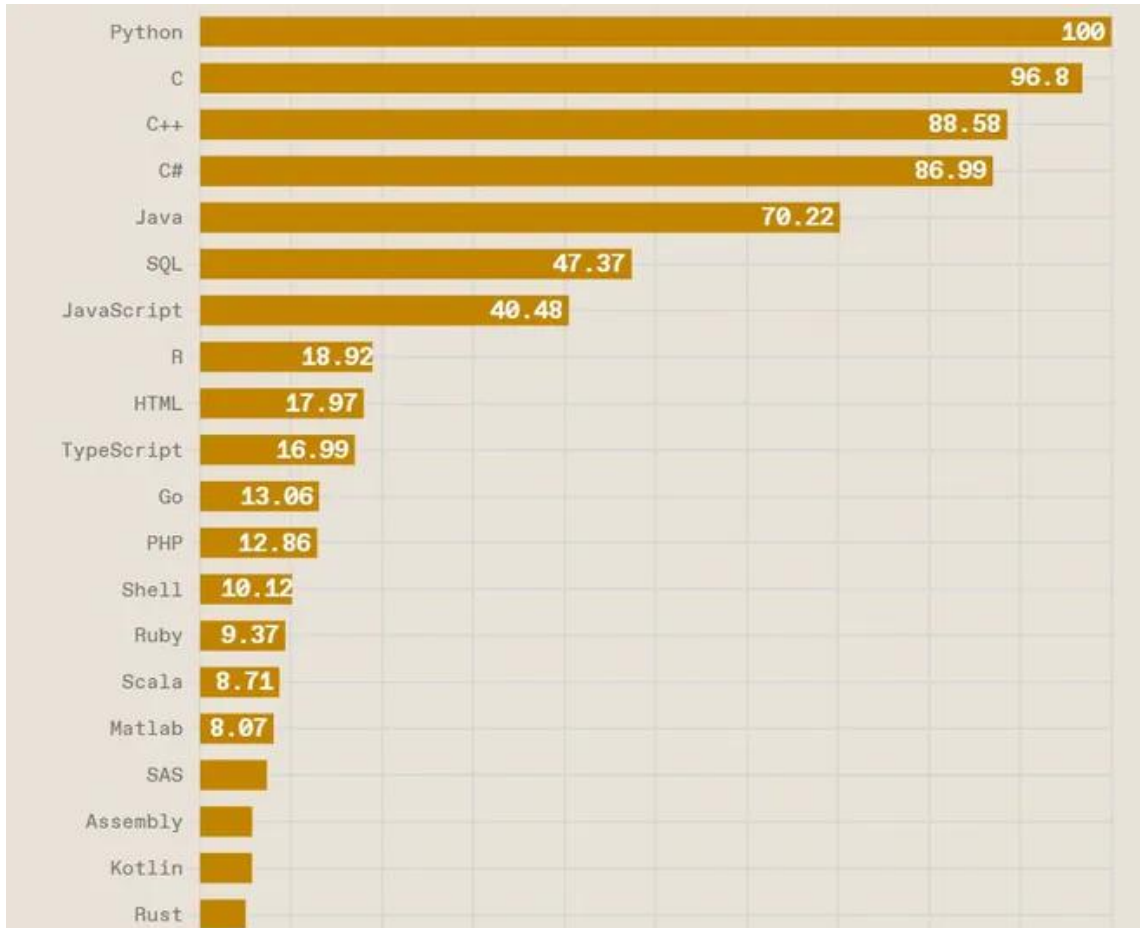
Empresas que utilizam Python?

WIKIPEDIA



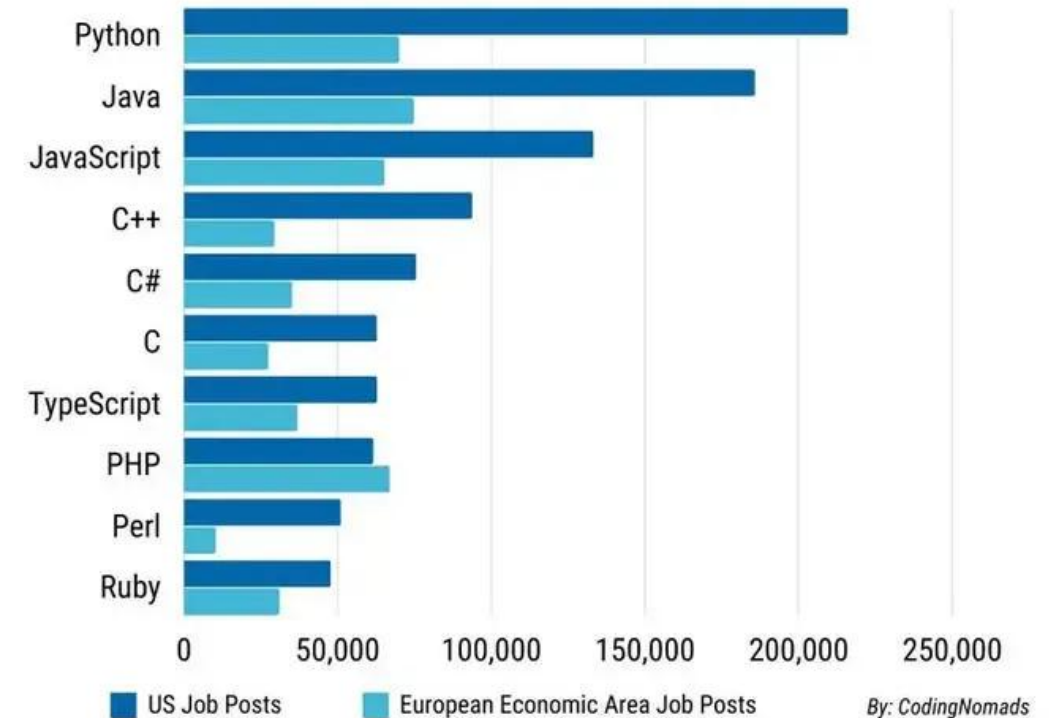
Fonte: <https://conaenge.com.br/curso-python-para-iniciantes/>

Python



Most in-demand programming languages of 2022

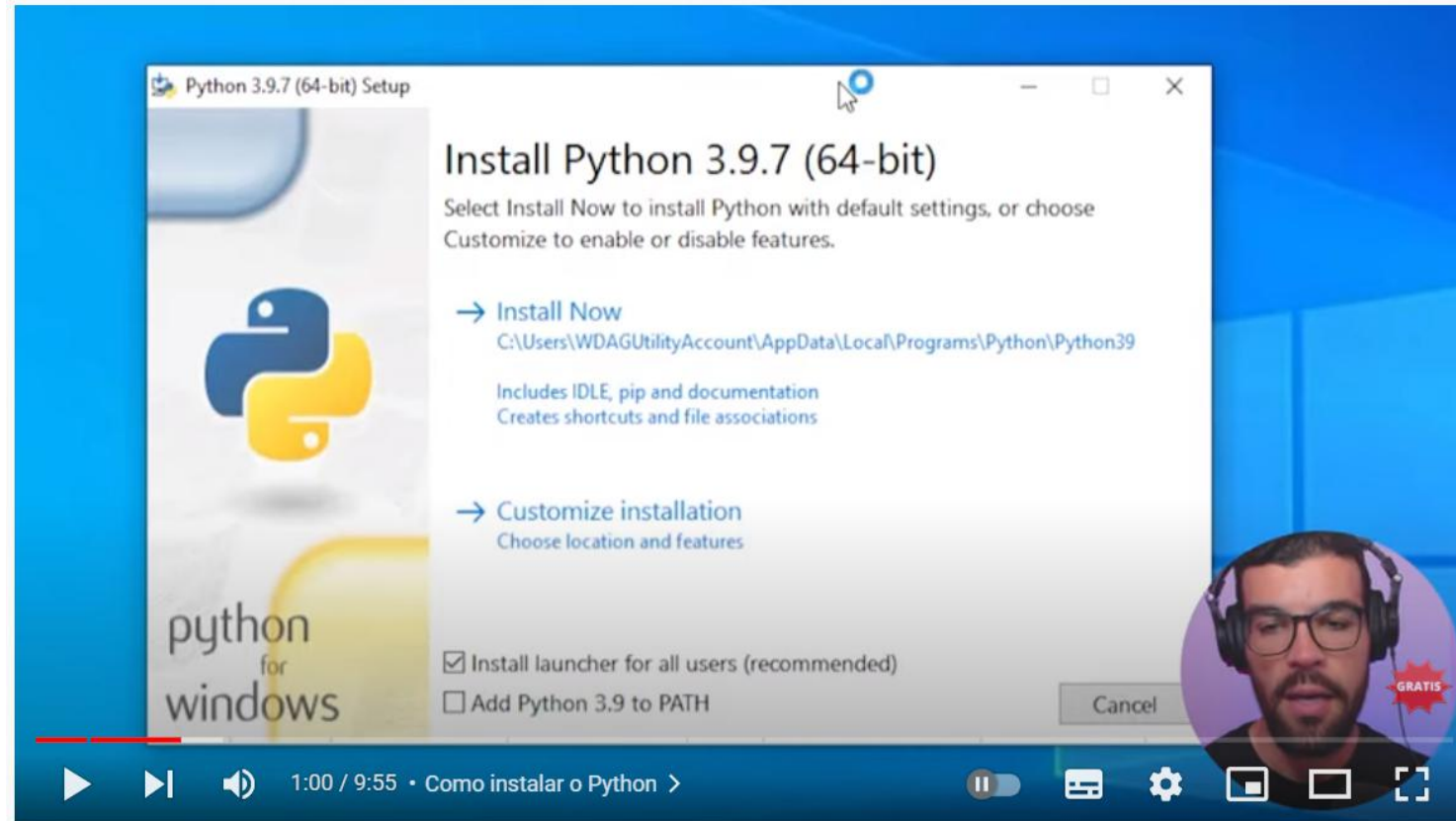
Based on LinkedIn job postings in the USA & Europe



<https://canaltech.com.br/software/ranking-ieee-veja-as-linguagens-de-programacao-mais-populares-de-2022-224274/>

<https://codingnomads.co/blog/the-best-programming-languages-to-learn/>

Instalação e configuração

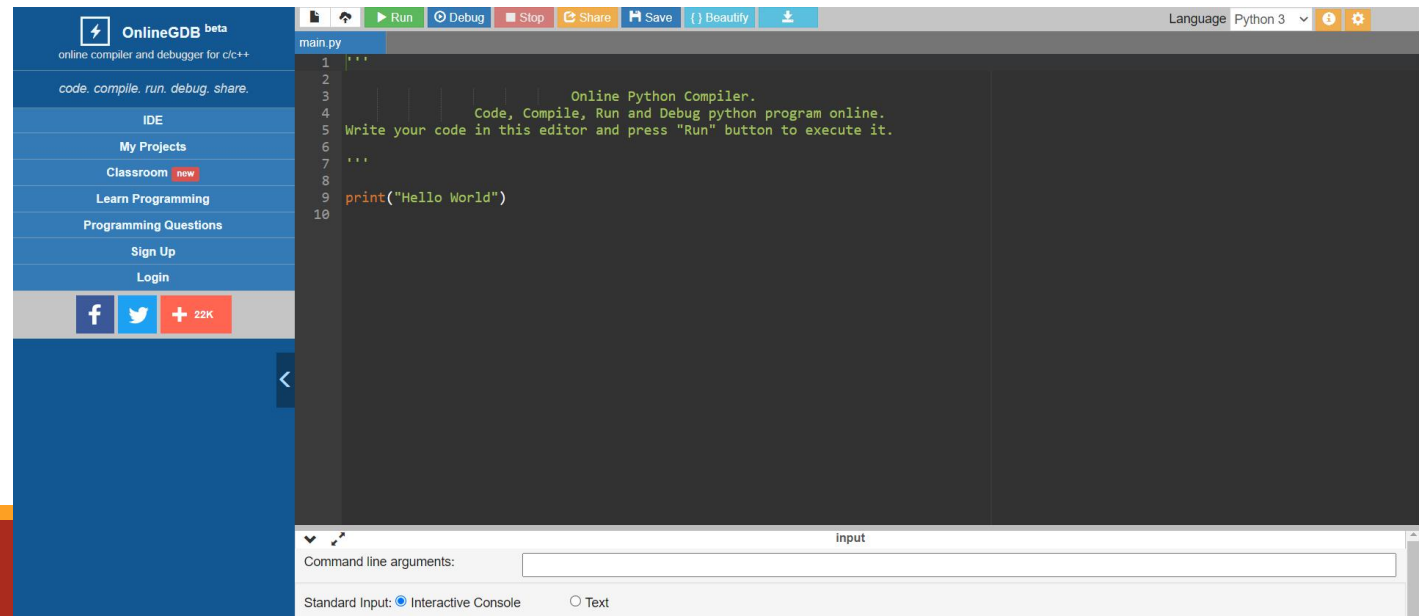


#Python #DevAprender #VsCode

Como Configurar VSCode Para Python [RÁPIDO] em 2021

Python online

- Online GDB: https://www.onlinegdb.com/online_python_compiler
- Replit: <https://replit.com/languages/python3>
- Online Python: <https://www.online-python.com/>
- Programiz: <https://www.programiz.com/python-programming/online-compiler/>

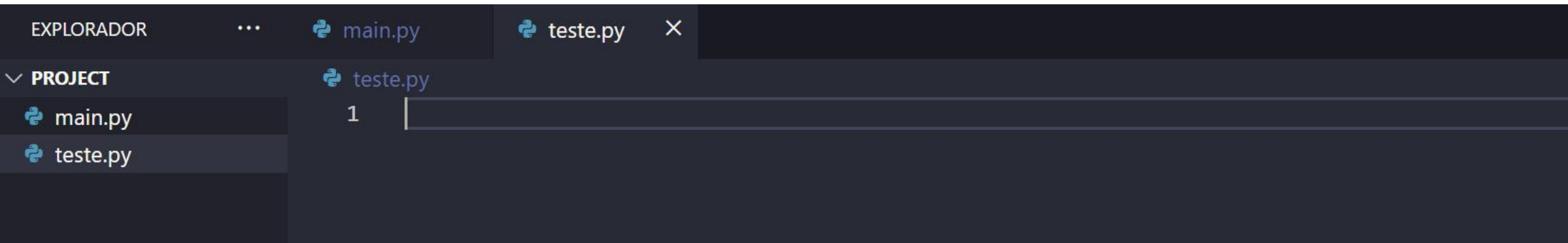
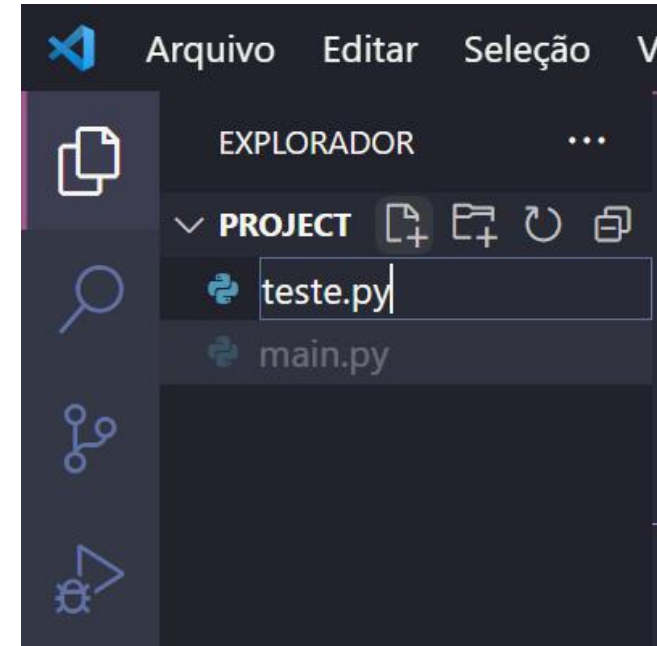
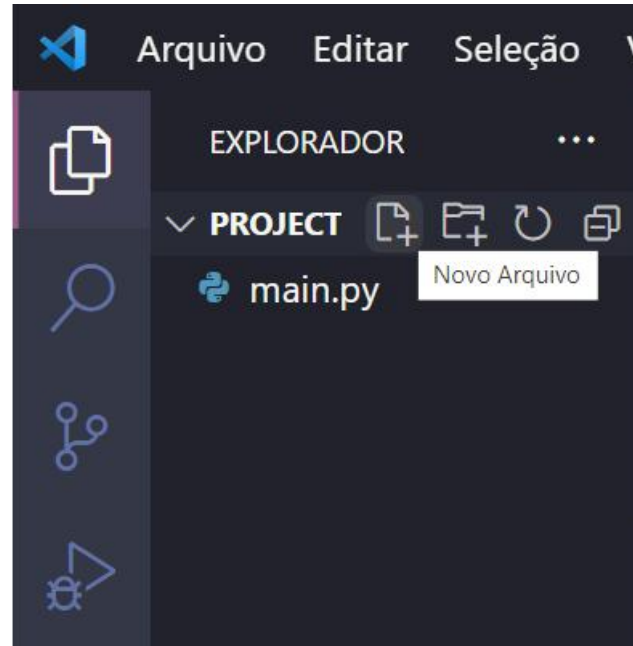


Algumas dicas iniciais

- ❑ Python é case-sensitive (sensível à caixa):
 - ❑ Diferencia maiúsculas de minúsculas;
 - ❑ Logo, print é diferente de Print.
- ❑ Toda vez que abrir aspas (""") não se esqueça de fechá-las;
- ❑ Toda vez que abrir parênteses não se esqueça de fechá-los;
- ❑ Não é necessário o uso de ; ao final das linhas de código.

Criando um primeiro programa

- ❑ Criar um arquivo com a extensão .py
- ❑ Ex: teste.py



Hello World

- ❑ Em Python 3, é necessário o uso da função `print()`:

```
print ('Hello World!')
```

- ❑ Em Python 2, a instrução de impressão não é função:

```
print 'Hello World!'
```

Indentação

- Python usa indentação como delimitação de blocos. A seguir, podemos ver dois exemplos de indentação, apresentando dois modos:

```
main.py > ...  
1  def f():  
2  x = 42  
3  return x  
4  print(f())  
5
```



```
main.py > ...  
1  def f():  
2      x = 42  
3      return x  
4  print(f())  
5
```



Indentação

- ❑ Se o bloco tem apenas um comando, pode-se escrever tudo em uma linha:

```
main.py > ...  
1 resposta = int(input("Digite uma opção: "))  
2 if resposta == 42: print('Opção válida!')  
3
```

Comentários

- ❑ Para colocar comentários no código, utiliza-se hashtag (comentários de uma linha), aspas simples e aspas duplas (comentários de várias linhas).
- ❑ Se o bloco tem apenas um comando, pode-se escrever tudo em uma linha.

```
main.py
1  # Exemplo de comentário em uma linha!
2
3  '''
4  Exemplo de
5  comentário utilizando
6  aspas simples.
7  '''
8
9  """
10 Exemplo de
11 comentário utilizando
12 aspas duplas.
13 """
```

Variáveis

- ❑ Programas manipulam dados (valores) e esses dados são armazenados em variáveis;
- ❑ Uma variável é uma posição na memória referenciada por um identificador (nome);
- ❑ Em Python não precisamos declarar variáveis, basta usá-las no código. Dependendo do valor atribuído a variável ela assumirá o tipo de dado a ela atribuída, por exemplo, inteiro, real, booleano ou string (int, float, bool ou str, respectivamente).
- ❑ A atribuição é realizada utilizando o operador de atribuição (=):
 - ❑ Usado para atribuir um valor a uma variável;
 - ❑ Armazena um valor na posição de memória referenciada por essa variável.

Variáveis

```
main.py > ...  
1    x = 10  
2    nome = 'Lucas'  
3    y = 2.5  
4    sn = True
```

Um nome de variável é uma sequência de letras ($a \rightarrow z$, $A \rightarrow Z$) e números ($0 \rightarrow 9$), que devem sempre começar com uma letra. Apenas letras comuns são permitidas. Letras acentuadas, cedilhas, espaços, caracteres especiais como \$, #, @, etc. são proibidos, exceto para o caractere `_` (sublinhado/underline). No entanto, elas devem necessariamente (convenção) começar com letras minúsculas.

Variáveis

- ❑ É importante também estar atento às palavras reservadas da linguagem, que não podem ser utilizadas para nomear variáveis:

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

Variáveis

❑ Exemplo de atribuição: `nota_p1 = 7.5`

❑ No exemplo, é criada uma variável de nome `nota_p1` que passa a armazenar o valor real 7.5.

❑ A partir do momento em que uma variável é criada, ela pode ser usada em expressões aritméticas e atribuídas a outras variáveis. Exemplo:

```
nota_p1 = 7.5
nota_p2 = 8.0
media = (nota_p1 + nota_p2) / 2
```

Variáveis

- ❑ As variáveis em Python tem um tipo, que é definido no momento em que a variável é criada por um comando de atribuição;
- ❑ Cada tipo define os valores que a variável pode armazenar;
- ❑ Cada tipo ocupa uma certa quantidade de memória.

Variáveis

- ❑ Em Python, pode-se descobrir qual o tipo da variável através do interpretador utilizado. Para isso, basta utilizar a função `type()` e a variável dentro do parênteses

```
main.py
1  print(type(1))
2  print(type(1.1))
3  print(type(1 + 1.1j))
4  print(type('Python'))
5  print(type([1, 1.1, 1 + 1.1j, 'Python']))
6
```

```
<class 'int'>
<class 'float'>
<class 'complex'>
<class 'str'>
<class 'list'>
```

Constantes

- ❑ A regra de nomeação das constantes no Python segue um padrão parecido com as de variáveis, com a diferença de que todas as letras são maiúsculas e separadas por underline “_”.
- ❑ Porém, por possuir tipagem dinâmica, os valores atribuídos à constantes podem ser alterados sem problemas:

```
main.py > ...  
1  MINHA_CONSTANTE = 10  
2  print(MINHA_CONSTANTE) # 10  
3  
4  MINHA_CONSTANTE = 15  
5  print(MINHA_CONSTANTE) # 15  
6
```

Impressão de variáveis e constantes

❑ Para imprimir variáveis e constantes em Python usa-se a função `print ()`.

❑ Exemplo:

```
print ("Minha primeira mensagem em Python")  
print (34+56)
```



```
lucas@lucas-Inspiron-5548:~/Documents,  
objects/teste.py  
Minha primeira mensagem em Python  
90
```

❑ Exemplo:

```
mensagem = "Bom dia!"  
print(mensagem)
```

```
NUMERO_NOTAS = 2  
nota_p1 = 7.5  
nota_p2 = 8.0
```

```
media = (nota_p1 + nota_p2)/NUMERO_NOTAS
```

```
print("A média das notas", nota_p1, "e", nota_p2, "é", media)
```


```
lucas@lucas-Inspiron-5548:~/Dropbox/UPE/codes$  
Bom dia!  
A média das notas 7.5 e 8.0 é 7.75
```



Impressão de variáveis e constantes

- ❑ Para imprimir variáveis e constantes em Python usa-se a função `print ()`.
- ❑ Exemplo:

```
print ("Minha primeira mensagem em Python")  
print (34+56)
```



```
lucas@lucas-Inspiron-5548:~/Documents,  
objects/teste.py  
Minha primeira mensagem em Python  
90
```

- ❑ Exemplo:


```
mensagem = "Bom dia!"  
print(mensagem)
```

```
NUMERO_NOTAS = 2  
nota_p1 = 7.5  
nota_p2 = 8.0
```


```
media = (nota_p1 + nota_p2)/NUMERO_NOTAS
```

```
print("A média das notas", nota_p1, "e", nota_p2, "é", media)
```

Como se pode ver, ao receber uma variável, o comando `print` exibe o valor da variável e não seu nome.



```
lucas@lucas-Inspiron-5548:~/Dropbox/UPE/codes$  
Bom dia!  
A média das notas 7.5 e 8.0 é 7.75
```



Impressão de variáveis e constantes

- ❑ Também é possível imprimir diretamente o valor de uma expressão.
- ❑ Exemplo:

```
2  HORAS_MINUTOS = 60
3
4  print("Duas horas são mesmo que ", 60*2, " minutos.")
5  print("Duas horas são mesmo que ", HORAS_MINUTOS*2, " minutos.")
```



```
lucas@lucas-Inspiron-5548:~/Documents/vsc
objects/teste.py
Duas horas são mesmo que  120  minutos.
Duas horas são mesmo que  120  minutos.
```


Variáveis (exercício)

1. Que tipo de dado seria usado para representar a idade de uma pessoa?
2. Que tipo de dado seria usado para representar a altura de uma pessoa?
3. Que tipo de dado seria usado para representar o sexo de uma pessoa?
4. Que tipo de dado seria usado para representar se uma pessoa está presente ou não?

Mais exercícios...

Questão 1:

Considere as seguintes afirmativas:

I - Uma variável é um elemento sintático do algoritmo que tem como funcionalidade armazenar um determinado valor.

II - Uma Constante é um elemento sintático do algoritmo que tem como funcionalidade armazenar mais de um determinado valor.

III - O valor associado à variável depende do tipo de informação que se deseja manipular.

IV - O valor associado à constante não depende do tipo de informação que se deseja manipular.

Está(ão) correto(s):

Alternativas:

A) apenas o item I.

B) apenas o item III.

C) apenas os itens I e III.

D) apenas os itens II e IV.

E) os itens I, II e III.

Operadores aritméticos

Operação	Operador
Adição	+
Subtração	-
Multiplicação	*
Divisão	/

Operadores aritméticos

Operação	Operador
Exponenciação	**
Parte inteira do resultado da divisão	//
Módulo	%

Ordem de precedência dos operadores aritméticos

Operador	Ordem de resolução na expressão
()	1º
**	2º
*, /, //, %	3º
+, -	4º

Operadores aritméticos

- $a + \frac{b}{b+c} \Rightarrow a + b/(b+c)$
- $b^2 + c^2 \Rightarrow b*b + c*c$
- $\frac{x}{a+\frac{b}{c}} \Rightarrow x/(a+b/c)$

Vamos exercitar?

a) $5+3*2$

b) $5/2$

c) $5//3$

d) $4*(5+2)**3$

e) $420**339$

f) $5\%2$

g) $9**(1/2)$

h) $8**(1/3)$

i) $\text{pow}(2,5)$

Vamos exercitar mais?

1. Como você pode consultar o tipo de uma variável em python?
 - a) Imprimindo o valor e determinando o tipo de baseado no valor exibido.
 - b) Usando a função `type()`.
 - c) Usando o valor em uma equação conhecida e imprimindo o valor resultante.
 - d) Olhando para a declaração da variável.

2. Indique como verdadeiro ou falso:
 - a) Duas variáveis definidas como “teste” e “Teste” são consideradas como idênticas.

Vamos exercitar mais?

1. Classifique o itens abaixo como nome de variável válido (V) ou inválido (F).

- a) teclado_lidinho
- b) germa66
- c) 1º_lugar
- d) PlayerID
- e) int
- f) class

Vamos exercitar mais?

1. Faça um Programa que dado o número de horas trabalhadas e o valor da hora, calcule e mostre o total do salário no referido mês.
2. Você estudou muito pra sua prova de lógica de programação e conseguiu terminar a prova em 1 hora e 34 minutos. Faça um programa que calcula e exibe o tempo de prova decorrido, em minutos e em segundos.
3. Faça um Programa que converta metros para centímetros.

Vamos exercitar mais?

1. Faça um Programa que armazene 4 notas bimestrais, calcule e imprima a média.
2. Faça um Programa que calcule a área de um círculo de raio 5 cm.
3. Faça um Programa que calcule a área de um quadrado, em seguida mostre o dobro desta área para o usuário.
4. As idades das pessoas de um determinado grupo são 10, 12, 15 e 17 anos. Calcule e exiba a média de idade do grupo, e a variação percentual da média das idades caso uma pessoa de 16 anos se junte ao grupo.
5. Faça um Programa que dada a temperatura em graus Fahrenheit, transforme e mostre a temperatura em graus Celsius.

Dúvidas???



Fonte: <https://institutoseculoxxi.com.br/duvidas-entramos-em-contato-com-voce/>