



Pensamento Computacional

2023

Pensamento Computacional (27 horas)

CONTEÚDO PROGRAMÁTICO

1. **Pensamento Computacional**
2. **Algoritmos:** conceitos, variáveis e tipos de dados; expressões lógico-aritmético-relacionais.
3. **Linguagem de Programação:** Python
4. **Estruturas de controle:** seleção e repetição.
5. **Estruturas de dados homogêneos:** vetores e matrizes.
6. **Funções** e recursividade.
7. **Arquivos.**
8. **Boas práticas** de programação.

PROF. DR. VILMAR ABREU JUNIOR



- Graduado em Ciência da Computação – PUCPR
- Especialista em Gestão de TI - FAE
- Mestrado em CiberSegurança – PUCPR
- Doutorado em CiberSegurança – PUCPR
- **Especialista em Desenvolvimento Seguro**
- Lattes:

<http://lattes.cnpq.br/8126149863316538>

Online ao vivo

- ◇ **Período:** Tarde
- ◇ **Dia da semana:** Terça-feira e Quinta-feira
- ◇ **Horário:** 14h00 às 17h00
- ◇ **Gravação:** Todas as aulas serão gravadas

Cronograma

Aula	Data	Horário
#1	06/06/2023	14h00 – 17h00
#2	13/06/2023	14h00 – 17h00
#3	15/06/2023	14h00 – 17h00
#4	20/06/2023	14h00 – 17h00
#5	22/06/2023	14h00 – 17h00
#6	27/06/2023	14h00 – 17h00
#7	29/06/2023	14h00 – 17h00
#8	06/07/2023	14h00 – 17h00
#9	11/07/2023	14h00 – 17h00

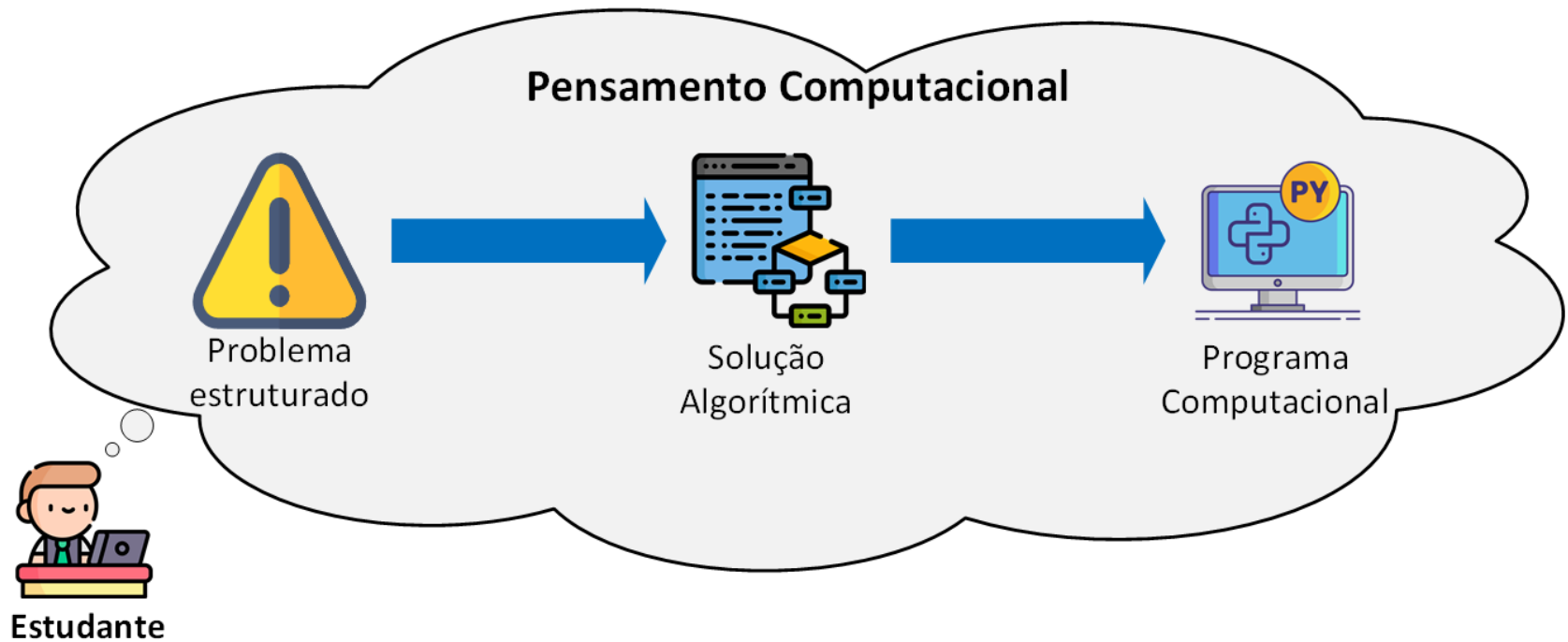
Informações Importantes

Link para Matrícula	https://432f5d.mannesoftprime.com.br/webcandidato/index.php?c=94B421E869CAA15276262E65EAE2235
Ambiente Virtual de Aprendizagem (AVA)	https://pucpr.instructure.com/
Plataforma de Aulas Remotas	Microsoft Teams – Link disponível no AVA (Preferencialmente utilizar o software instalado no computador)
Canal de Comunicação Oficial	Ambiente Virtual de Aprendizagem

Avalie seus conhecimentos

- ◆ Acesse o site: <https://www.menti.com/>
- ◆ Preencha com o código: 43202330

Mapa mental



Programação

Identificar
um
problema



Planejar a
solução para
o problema



Implementar a
solução (algoritmo)
usando uma
linguagem de
programação



Algoritmo

- Sequência de passos, bem definido, que visa resolver um problema
- Desenvolver um algoritmo demanda:
 1. Saber como solucionar um problema
 2. Converter uma solução em um algoritmo
 3. Implementar este algoritmo em uma linguagem de programação
 4. Validar e testar o algoritmo

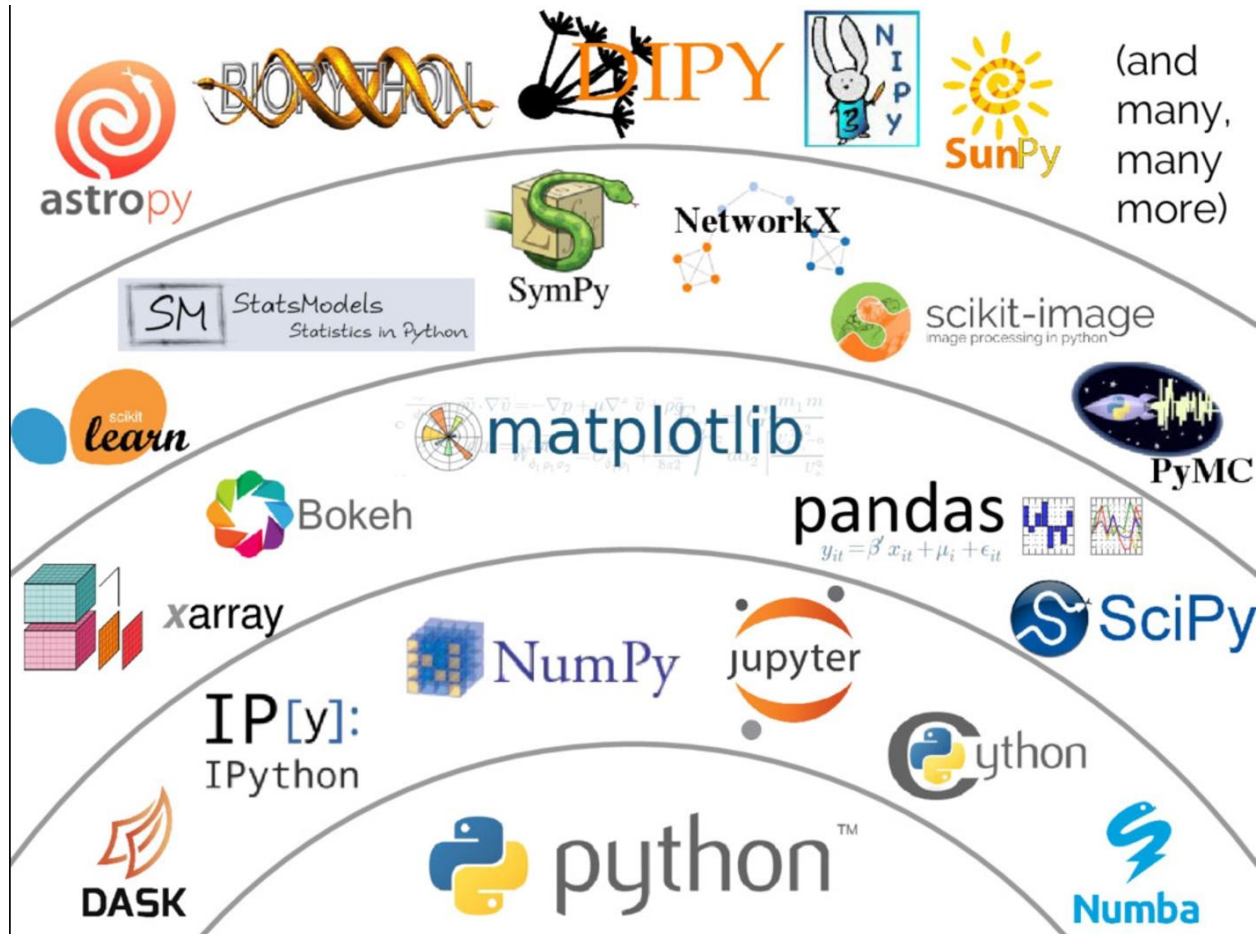
Linguagem de programação

- Infelizmente, se digitarmos qualquer solução proposta no computador, ele não vai conseguir executar o nosso algoritmo
- Por isso, precisamos “traduzir” nossas soluções para uma linguagem que o computador entenda
- Desta forma, uma linguagem de programação é uma linguagem utilizada para controlar máquinas
- Como linguagens humanas, nós temos regras sintáticas e semânticas para determinar estrutura e significado
- Note: linguagens de programação ainda são intermediárias, pois o computador entende apenas códigos binários (0s e 1s)

Linguagens de programação



Python



Ambiente



Alternativa (temporária): <https://repl.it/languages/python3>

Variáveis

- Começam com letras ou _
- Não podem começar com número
- Apenas alfanuméricos (a-z, A-Z, 0-9 e _)
- Case-sensitive
- Evitar caracteres compostos (acentuados).

Variáveis (Boas práticas)

- Usar nomes significativos de variáveis
- Utilizaremos a prática conhecida como camelCase:

custoPassagem

valorHora

nomeCliente

pesoProduto

- Evitar:

x

variavel1

Tipos de dados

- **Inteiro (int)**: toda e qualquer informação numérica que pertença ao conjunto dos inteiros;
Ex: Idade de uma pessoa, número de alunos na sala
- **Real (float)**: toda e qualquer informação numérica que pertença ao conjunto dos reais;
Ex: Altura de uma pessoa, peso
- **Caracteres (string)**: toda e qualquer informação composta por um conjunto de caracteres alfanuméricos;
Ex: Nome, endereço
- **Lógico (bool)**: informação que assume somente dois valores possíveis (verdadeiro ou falso)
Ex: Porta aberta ou fechada, lâmpada acesa ou apagada

Entrada e Saída

Para que os algoritmos possam apresentar seus resultados e também possam obter dados (por exemplo solicitando-o ao usuário), comandos especiais de entrada e saída são necessários.

Entrada (dados do teclado)

- **variável = input(<texto informativo entre aspas>)**
- A variável receberá como conteúdo o valor lido, ou seja, digitado pelo teclado. O conteúdo lido é um dado dito “bruto”, do tipo string.
- **variável = int(input())**
Utilizado para ler um número inteiro do teclado.
- **variável = float(input())**
Utilizado para ler um número real do teclado.

Saída

- `print(variável)`
- `print("O nome é " + nome)`
- `print("A idade é " + str(idade))`

ATIVIDADE I

Crie um algoritmo que recebe quatro informações: e armazena em variáveis: Nome, CPF, Telefone e Ano de Nascimento.

Em seguida, imprima na tela esses valores formatados.



Expressões Aritméticas

- Operadores Aritméticos

Chamamos de operadores aritméticos o conjunto de símbolos que representa as operações básicas da matemática, a saber:

Operador	Função	Exemplos
+	Adição	$2 + 3$, $x + y$
-	Subtração	$4 - 2$, $n - m$
*	Multiplicação	$3 * 4$, $a * b$
/	Divisão	$10 / 2$, $x1 / x2$

- Atribuição de variáveis

Para a atribuição de um valor à uma variável, utiliza-se o sinal de '=',

ATIVIDADE 2

Crie um algoritmo que calcula o IMC e imprime o resultado.



Opções de impressão formata (print)

```
print("IMC com vírgula:", imc)
print("IMC concatenado: " + str(imc))
print("IMC com round:", round(imc, 2))
print("IMC formato antigo: %.2f" % imc)
#formato antigo com várias variaveis
print("Para a altura %.2f e peso %.2f o IMC é %.3f" % (altura, peso, imc))
#formato novo
print(f"Para a altura {altura} e peso {peso} o IMC é {imc}")
#formato novo definindo casas decimais
print(f"Para a altura {altura:.2f} e peso {peso:.2f} o IMC é {imc:.2f}")
```

Exercícios

- 1) Escreva um algoritmo em Python para calcular a idade de alguém, sabendo-se seu ano de nascimento.
 - Utilize o ano de maneira fixa
 - Ou use a função: `datetime.date.today().year`
 - 2) Escreva um algoritmo em Python para calcular o valor, em reais, que deve ser pago por um cliente de uma locadora de carros. Sabe-se que:
 - O valor de locação de cada carro é 180,00 reais por dia;
 - O cliente pode locar um único carro por vários dias.
 - 3) Leia do teclado a temperatura em Celsius e imprima o equivalente em Fahrenheit.
- Fórmula: $(X \text{ }^{\circ}\text{C} \times 9/5) + 32$**
- 4) Escrever um algoritmo para calcular a média de 4 notas.
 - 5) Calcular sua idade em meses

Operadores Relacionais

- Utilizados para realizar comparações entre dois valores. Tais valores são representados por constantes, variáveis ou expressões aritméticas.

Operador	Função	Exemplos
==	Igual a	3 == 3, x == y
>	Maior que	5 > 4, x > y
<	Menor que	3 < 6, x < y
>=	Maior ou igual a	5 >= 3, x >= y
<=	Menor ou igual a	3 <= 5, x <= y
!=	Diferente de	8 != 9, x != y

- O resultado obtido é sempre um valor lógico (**True** ou **False**).

Exercícios #1

$$x = 2 * 4 == 24 / 3$$

Exercícios #2

$$x = 15 \% 4 < 19 \% 6$$

Exercícios #3

$$x = 3 * 5 / 4 \leq 3 ** 2 / 0.5$$

Exercícios #4

$$x = 2 + 8 \% 7 \geq 3 * 6 - 15$$

Expressões Lógicas

- Operadores Lógicos

Utilizaremos três conectivos (operadores) básicos para a formação de novas proposições lógicas compostas a partir de outras proposições lógicas simples: E, OU e NAO.

- and
- or
- not

- Tabela Verdade

Conjunto das possibilidades combinatórias entre os valores de diversas variáveis lógicas, as quais se encontram em apenas duas situações (Verdadeiro ou Falso), e um conjunto de operadores lógicos

Tabela Verdade

p	q	not p	p AND q	p OR q
F	F	V	F	F
F	V	V	F	V
V	F	F	F	V
V	V	F	V	V

Ordem de Precedência

- Hierarquia de tratamento dos operadores.
- Para os operadores lógicos:
 - Antes: não
 - Depois: e, ou
- Para uso geral:
 - Antes: parênteses
 - operadores aritméticos (+, -, *, /)
 - operadores relacionais (>, <, >=, <=, ==, !=)
 - Depois: operadores lógicos (not, and, or)
- Ao realizarmos cálculos em qualquer linguagem de programação, precisamos levar em conta a precedência que temos entre as operações

Dica #1: na dúvida, monte a equação no papel, adicione parênteses, e depois a transcreva no computador

Exemplos

- Avalie as seguintes expressões no Python:
 $5 < 4 + 3$
 $12 + 1 \geq 12$
 $16 * 2 == 32$
 $16 != 16$
 $5 \geq 6$
- Encontre o valor lógico das seguintes expressões:
 $(V \text{ e } F) \text{ ou } (V \text{ ou } \text{nao } F)$
 $\text{nao } (V \text{ ou } V) \text{ e } (F \text{ e } F)$
 $(\text{nao } (5 != 10/2) \text{ ou } V) \text{ e } (2 - 5 > 5 - 2 \text{ ou } V)$

Bibliografia

FORBELLONE, André Luiz Villar; EBERSPÄCHER, Henri Frederico. Lógica de programação: a construção de algoritmos e estruturas de dados. 3. ed. São Paulo: Pearson Prentice Hall, 2005.

MANZANO, José Augusto N. G.; OLIVEIRA, Jayr Figueiredo de. Algoritmos: lógica para desenvolvimento de programação de computadores. 24. ed., rev. São Paulo: Érica, 2010.

MENEZES, Nilo Ney Coutinho. Introdução à programação com Python: algoritmos e lógica de programação para iniciantes. 1. ed. São Paulo: Novatec, 2010.



PUCPR
GRUPO MARISTA

Contato:

vilmar.abreu@pucpr.br