

Princípios de Construção de Algoritmos

Conteúdo Programático

Prof. Gabriel Resende Machado



gabrielmachado@unifeso.edu.br



<https://www.linkedin.com/in/machadogabriel>



<https://github.com/UNIFESO-Gabriel/pca>

Professor: Gabriel Resende Machado



- **Formação:**

- **Mestre:** Sistemas e Computação (IME, 2019);
- **Bacharel:** Ciência da Computação (UNIFESO, 2016).

- **Áreas de Especialização:**

- Ciência de Dados;
- Banco de Dados;
- Algoritmos.

- **Principais formas de contato:**

- E-mail e LinkedIn (disponíveis no *slide* de capa);
- Plataforma Canvas (plano de aula, ementa, *slides* de aulas, tarefas e exercícios).

Sobre a Disciplina

- **Objetivos gerais:**

- Compreender sobre a importância da programação considerando todas as áreas do conhecimento, para **desenvolver do raciocínio lógico na resolução algorítmica de problemas;**
- Utilizar uma linguagem de programação de alto nível com aplicações numéricas e não numéricas, **aplicando a lógica de programação em Python com a utilização de bibliotecas e plataformas de desenvolvimento e versionamento.**

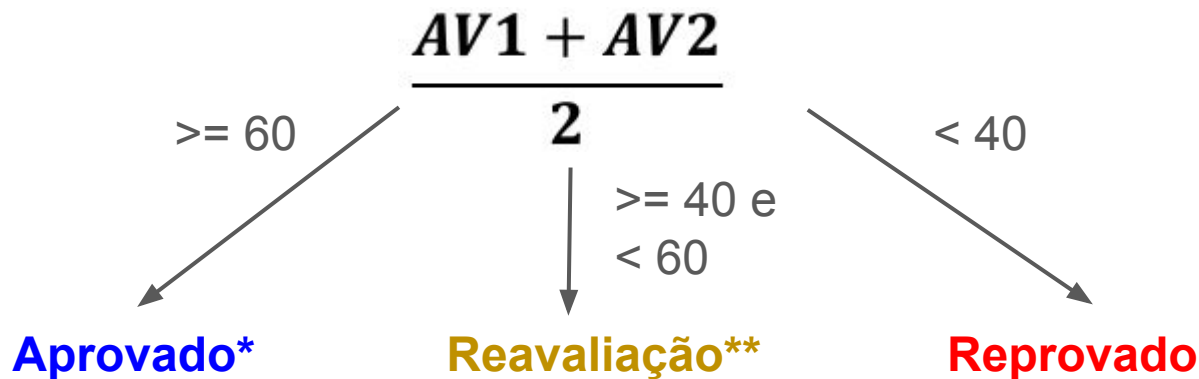
- **Planejamento** (arquivos disponíveis no Canvas):

- **Plano de Ensino:** <https://bit.ly/3M1wZBi>;
- **Plano de Aulas:** a disponibilizar;
- **Plano de Curso:** a disponibilizar.

Avaliações

- **Avaliação 1 (AV1):**
 - **Qstione** (previsto para 02/10/24): **40 pontos**;
 - **Projeto 1** (entrega prevista para 02/10/24 via Canvas): **60 pontos**;
- **Avaliação 2 (AV2):**
 - **Qstione** (previsto para 04/12/24): **40 pontos**;
 - **Projeto 2** (entrega prevista para 04/12/24 via Canvas): **60 pontos**;
- Ambas as avaliações do Qstione serão compostas por 10 questões objetivas e 02 discursivas;
- As avaliações do Qstione compõem 40% da nota trimestral. 60% restantes vêm de trabalhos e/ou atividades.

Cômputo da nota final



- **Avaliações especiais:**

- **2ª chamada (11/12/24):** somente para alunos que faltaram **em uma** das avaliações do Qstione.
- **Reavaliação (18/12/24):** somente para os alunos com a nota final presente no intervalo [40, 60).

* Frequência abaixo dos 75% das aulas resulta em **reprovação por faltas**;

** A prova de reavaliação compreende toda a matéria do curso e é elaborada livremente pelo professor.

Princípios de Construção de Algoritmos

AULA 01 - Introdução

Prof. Gabriel Resende Machado



gabrielmachado@unifeso.edu.br



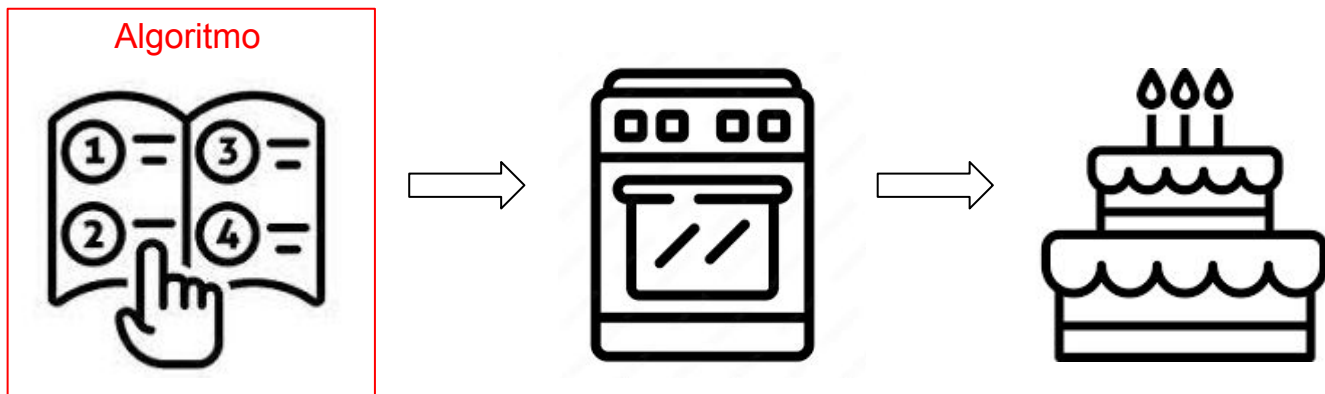
<https://www.linkedin.com/in/machadogabriel>



<https://github.com/UNIFESO-Gabriel/pca>

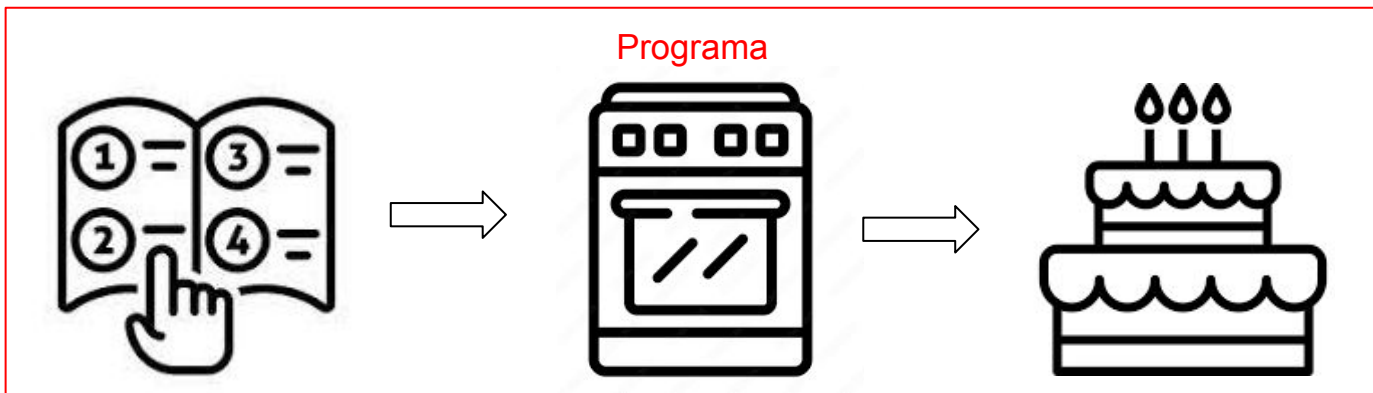
Algoritmo

- **Definição formal:** conjunto das regras e procedimentos lógicos perfeitamente definidos que levam à solução de um problema em um número finito de etapas;
- **Pense em algoritmo como uma receita de bolo:**
 - É um conjunto de instruções bem definidas, passo a passo, para resolver um problema específico;
 - Não leva em consideração a utilização de nenhuma linguagem ou ferramenta específica;
 - Um algoritmo pode ser expresso em português claro, notação matemática ou um **fluxograma**.



Programa e Sistema

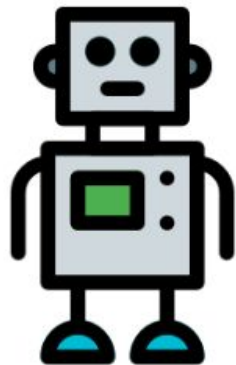
- **Definição formal:** implementação real de um algoritmo em uma linguagem de programação específica, como C, Python, Java ou C++.
 - É como pegar a receita e traduzi-la em instruções que um computador possa entender e executar. O programa usa sintaxe e estruturas específicas da linguagem escolhida para dar vida ao algoritmo.
 - Um programa basicamente recebe uma entrada (*input*) e gera uma saída (*output*).
- **Sistema** é um conceito mais amplo que abrange o programa e seu ambiente. Representa a cozinha onde o bolo é assado.



Exemplo de Programa

Um Primeiro Programa

- O Robô Aspirador de Pó

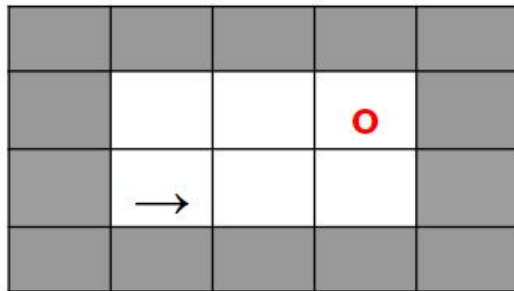


Possível Programa

1. Avançar
2. Avançar
3. Virar 90° Esquerda
4. Avançar
5. Limpar

Conjunto de Instruções

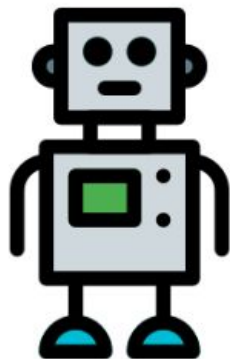
- Avançar
- Virar 90° Esquerda
- Virar 90° Direita
- Limpar



Exemplo de Programa

Um Primeiro Programa

- O Robô Aspirador de Pó

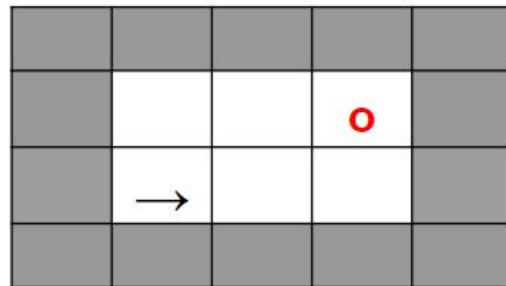


Possível Programa

1. Avançar ←
2. Avançar
3. Virar 90° Esquerda
4. Avançar
5. Limpar

Conjunto de Instruções

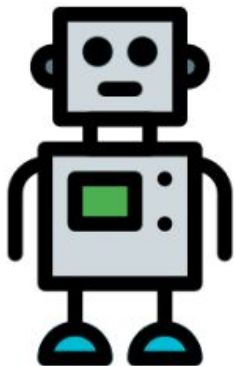
- Avançar
- Virar 90° Esquerda
- Virar 90° Direita
- Limpar



Exemplo de Programa

Um Primeiro Programa

- O Robô Aspirador de Pó

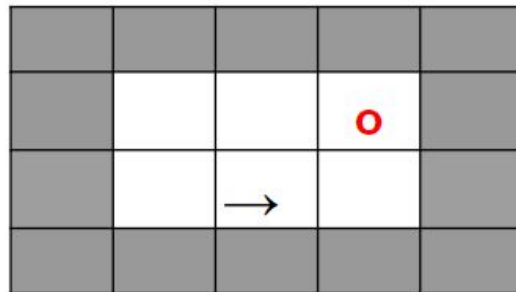


Possível Programa

1. Avançar
2. Avançar ←
3. Virar 90° Esquerda
4. Avançar
5. Limpar

Conjunto de Instruções

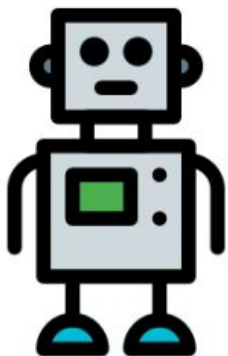
- Avançar
- Virar 90° Esquerda
- Virar 90° Direita
- Limpar



Exemplo de Programa

Um Primeiro Programa

- O Robô Aspirador de Pó

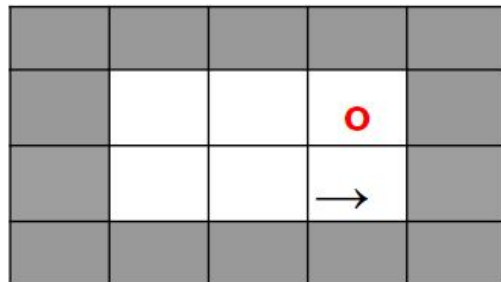


Possível Programa

1. Avançar
2. Avançar
3. Virar 90° Esquerda ←
4. Avançar
5. Limpar

Conjunto de Instruções

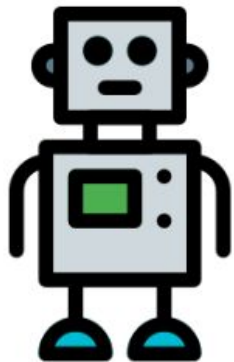
- Avançar
- Virar 90° Esquerda
- Virar 90° Direita
- Limpar



Exemplo de Programa

Um Primeiro Programa

- O Robô Aspirador de Pó

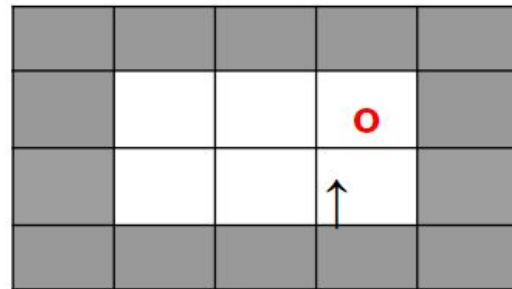


Possível Programa

1. Avançar
2. Avançar
3. Virar 90° Esquerda
4. Avançar ←
5. Limpar

Conjunto de Instruções

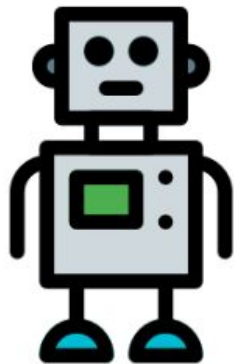
- Avançar
- Virar 90° Esquerda
- Virar 90° Direita
- Limpar



Exemplo de Programa

Um Primeiro Programa

- O Robô Aspirador de Pó

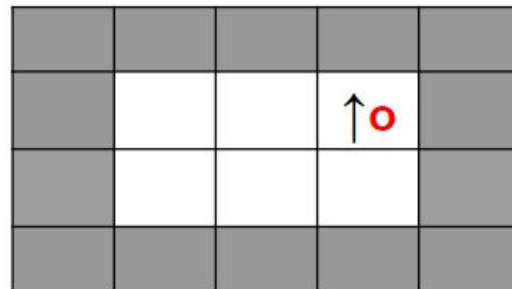


Possível Programa

1. Avançar
2. Avançar
3. Virar 90° Esquerda
4. Avançar
5. Limpar ←

Conjunto de Instruções

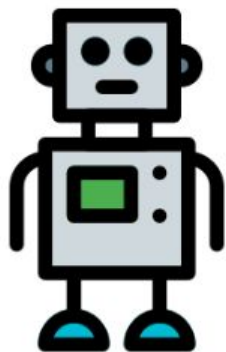
- Avançar
- Virar 90° Esquerda
- Virar 90° Direita
- Limpar



Exemplo de Programa

Um Primeiro Programa

- O Robô Aspirador de Pó

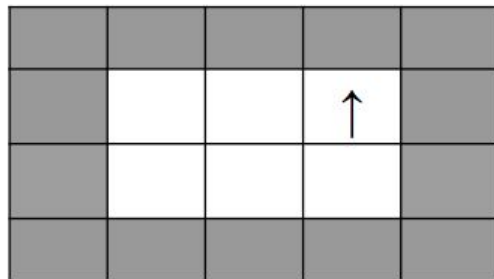


Possível Programa

1. Avançar
2. Avançar
3. Virar 90° Esquerda
4. Avançar
5. Limpar

Conjunto de Instruções

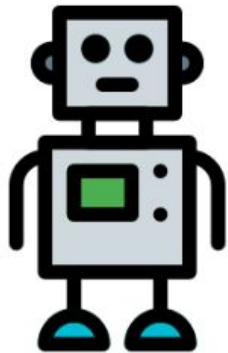
- Avançar
- Virar 90° Esquerda
- Virar 90° Direita
- Limpar



Novo Exemplo de Programa

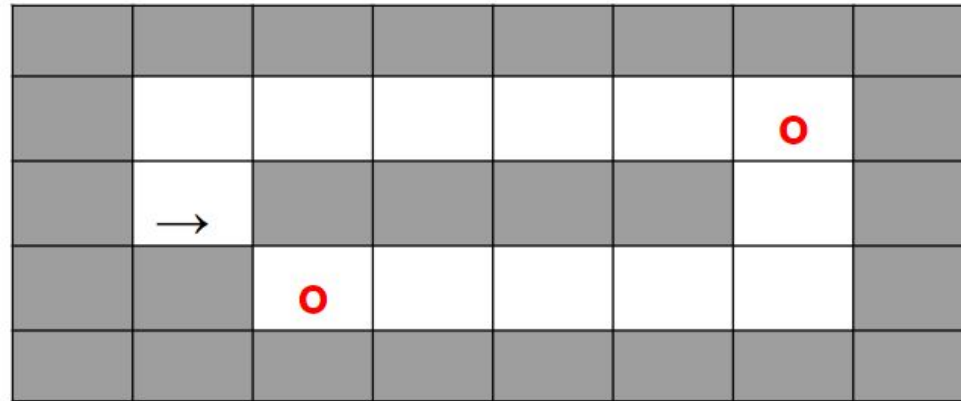
Um Primeiro Programa

- Programando um robô hipotético



Conjunto de Instruções

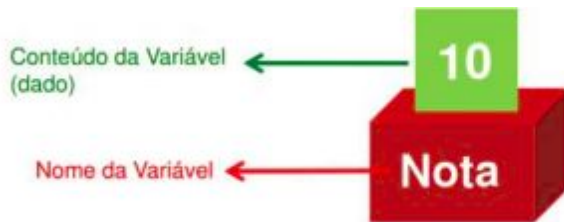
- Avançar
- Virar 90° Esquerda
- Virar 90° Direita
- Limpar



Novo Cenário!

O que é uma variável?

- Uma variável é um **espaço de armazenamento na memória** de um computador que **tem um nome associado a ele**;
- Esses espaços de armazenamento são utilizados para **guardar valores** que podem ser **modificados e acessados** durante a execução de um programa de computador;
- Por enquanto, vamos pensar que uma variável é uma “**caixinha**” e que, dentro dela, é possível **armazenar, consultar ou alterar um determinado valor**.



O que é uma procedimento / função / método?

- Procedimentos são subrotinas que não retornam nenhum valor;
- Funções são subrotinas que retornam um ou mais valores;
- Métodos são procedimentos ou funções presentes dentro de uma classe.

```
#include <stdio.h>
void imprimeCh(char ch, int n){
    int i;
    for(i=0; i < n; i++){
        printf("%c", ch);
    }
    printf("\n");
}

int main(){
    imprimeCh('+', 3);
    imprimeCh('+', 5);
    imprimeCh('=', 7);
    imprimeCh('+', 5);
    imprimeCh('+', 3);
    return 0;
}
```

Função para impressão utilizando dois parâmetros de entrada

Chamada das funções com argumentos

```
1 class Pessoa:
2     def __init__(self, nome, sexo, cpf, ativo):
3         self.nome = nome
4         self.sexo = sexo
5         self.cpf = cpf
6         self.ativo = ativo
7
8     def desativar(self):
9         self.ativo = False
10        print("A pessoa foi desativada com sucesso")
11
12 if __name__ == "__main__":
13     pessoal = Pessoa("João", "M", "123456", True)
14     pessoal.desativar()
```

Laboratório de Fundamentos em TIC

AULA 01 - Linguagem Python

Prof. Gabriel Resende Machado



gabrielmachado@unifeso.edu.br



<https://www.linkedin.com/in/machadogabriel>



<https://github.com/UNIFESO-Gabriel/fundamentos-em-tic>

Características da Linguagem Python

- Python, ao contrário do C, é uma linguagem multi-paradigma;
- Python tem uma sintaxe limpa e fácil de ler, o que torna o código mais legível e compreensível;
- Python é uma linguagem multiplataforma, o que significa que ela pode ser executada em diferentes sistemas operacionais como Windows, macOS, Linux, e muitos outros sem a necessidade de alterações significativas no código;
- Python possui uma vasta coleção de bibliotecas e frameworks que facilitam o desenvolvimento de diversas aplicações em *Desenvolvimento Web*, *Data Science* e *Cyber Security*;
- Python possui uma gestão de memória eficiente, incluindo coleta de lixo automática (*garbage collector*).



Características da Linguagem Python

Java

```
public class hello {  
    public static void main (String [] args) {  
        System.out.println("Hello World");  
    }  
}
```

C

```
#include "stdio.h"  
int main() {  
    printf("Hello World\n");  
}
```

Python

```
print("Hello World")
```



Principais Tipos de Variáveis em Python

1. **int (inteiro):** num = 10;
2. **float (ponto flutuante):** num = 10.5;
3. **complex (complexo):** Números complexos na forma a + bj: num = 1 + 2j;
4. **str (string):** texto = "Olá, Mundo!"
5. **list (lista):** Coleção ordenada e mutável de itens. lista = [1, 2, 3, "a", "b", "c"];
6. **tuple (tupla):** Coleção ordenada e imutável de itens. tupla = (1, 2, 3, "a", "b", "c");
7. **range (intervalo):** Sequência de números gerada a partir de um intervalo. intervalo = range(1, 10);
8. **dict (dicionário):** Coleção não ordenada de pares chave-valor. dict = {"chave1": "valor1", "chave2": "valor2"}
9. **set (conjunto):** Coleção não ordenada de itens únicos. conjunto = {1, 2, 3, 4, 5}
10. **frozenset (conjunto imutável):** Conjunto imutável de itens únicos. conjunto_imutavel = frozenset([1, 2, 3, 4, 5])
11. **Tipo Booleano:** True ou False. resultado = True
12. **Tipo None NoneType:** Ausência de valor ou valor nulo. lista = None;
13. **function:** Funções definidas pelo usuário ou integradas.

```
def minha_funcao():  
    pass
```

Python - Mão na Massa!

1. Escreva um programa em que leia uma sequência de números e exiba a soma deles.
2. Escreva um programa que leia uma lista de números inteiros e exiba-os em ordem crescente;
3. Escreva um programa onde o computador escolhe um número aleatório entre 1 e 100, e o jogador tenta adivinhar. O programa deve dar dicas se o palpite é muito alto ou muito baixo e contar o número de tentativas.
4. Escreva um programa que valide senhas com base nos critérios: tenha pelo menos 8 caracteres, contendo letras maiúsculas, minúsculas, números e caracteres especiais.
5. Faça um programa que conte o número de vogais e consoantes em uma *string* fornecida pelo usuário;
6. Crie um programa que simule as operações básicas de um caixa eletrônico: verificar saldo, fazer depósitos e saques. O programa deve manter um saldo atual e registrar todas as transações.

Python - Mão na Massa!

- Praticar, resolvendo problemas em Python disponíveis na plataforma Beecrowd:
<https://judge.beecrowd.com>.



Princípios de Construção de Algoritmos

AULA 01 - Introdução Dúvidas ou sugestões?

Prof. Gabriel Resende Machado



gabrielmachado@unifeso.edu.br



<https://www.linkedin.com/in/machadogabriel>



<https://github.com/UNIFESO-Gabriel/pca>