



*INTRODUÇÃO À COMPUTAÇÃO*

---

*UD II - ALGORITMOS*

*INTRODUÇÃO À ALGORITMOS*

# INTRODUÇÃO À ALGORITMOS



## ELEMENTOS DE COMPETÊNCIA

- Empregar recursos para operar em ambientes humanizados, integrando as dimensões física, humana e informacional deste ambiente operacional.
- Tomar decisões e conduzir ações, em situações de crise.

# UD II - INTRODUÇÃO À ALGORITMOS

## INTRODUÇÃO À ALGORITMOS



### OBJETIVOS

1. Compreender o conceito de algoritmo. (CONCEITUAL)
2. Descrever as relações entre algoritmos e a lógica de programação. (FACTUAL/CONCEITUAL)
3. Apresentar os diferentes tipos de algoritmos. (CONCEITUAL)

# UD II - INTRODUÇÃO À ALGORITMOS

## INTRODUÇÃO À ALGORITMOS



### ATITUDES

1. Organização: capacidade de desenvolver atividades de forma sistemática e eficiente.
2. Dedicação: agir, realizando espontaneamente, com empenho e entusiasmo, as atividades necessárias ao cumprimento da missão.
3. Responsabilidade: capacidade de cumprir suas atribuições assumindo e enfrentando as consequências de suas atitudes e decisões.

- 1) *Um processo de cálculo matemático ou da descrição sistemática da resolução de um grupo de problemas semelhantes, sendo, também, regras formais para a obtenção de um resultado ou da solução de um problema.*
- 2) *Sequência finita de passos ou conjunto de instruções, organizados logicamente, com a finalidade de execução de uma tarefa ou resolução de um problema específico.*



*Um algoritmo se constitui em uma solução possível para certo problema, sendo que outras soluções poderão existir.*

# ALGORITMO DE DESLOCAMENTO PARA O TRABALHO

Cap 2 / Pág:28-29

*Passo 1: abrir a porta da casa*

*Passo 2: ir até a garagem*

*Passo 3: destravar a porta do carro*

*Passo 4: dar partida no carro*

*Passo 5: retirar o carro da garagem*

*Passo 6: pegar o trajeto menos congestionado*

*Passo 7: estacionar o carro na garagem do trabalho*

*Passo 8: abrir a porta do escritório*



*A construção de algoritmos depende fundamentalmente do raciocínio lógico para estruturar passos de forma coerente.*

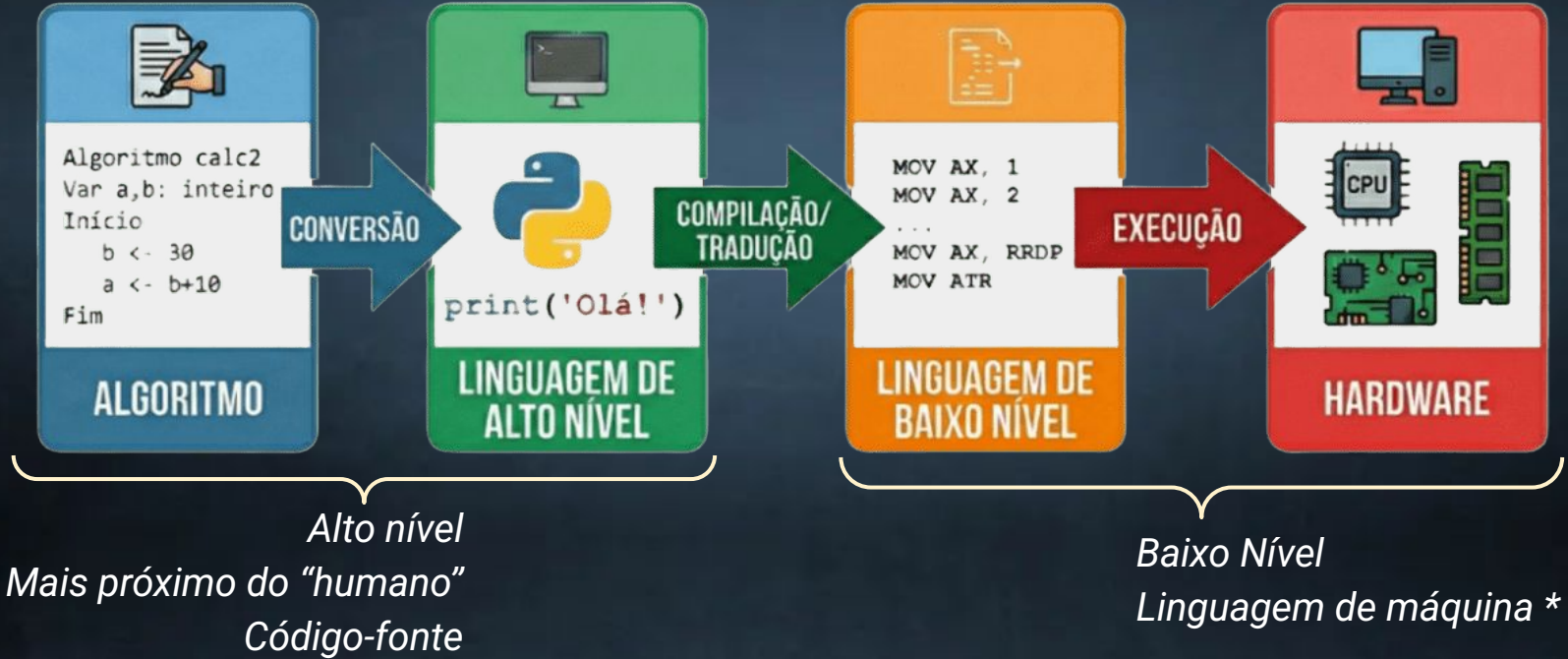
*A lógica é definida como a ciência dos princípios formais do raciocínio (científica).*

*Algoritmos podem ser convertidos para uma linguagem de programação. As linguagens de programação (C, Cobol, Java, Python etc), chamadas linguagens de alto nível, são constituídas por símbolos que mais se aproximam da linguagem humana, e, posteriormente, por meio de um processo de tradução (compilação ou interpretação), esse algoritmo é convertido para a linguagem de máquina.*

*A lógica de programação pode ser definida como um conjunto de técnicas para encadear pensamentos a fim de atingir determinado objetivo.*

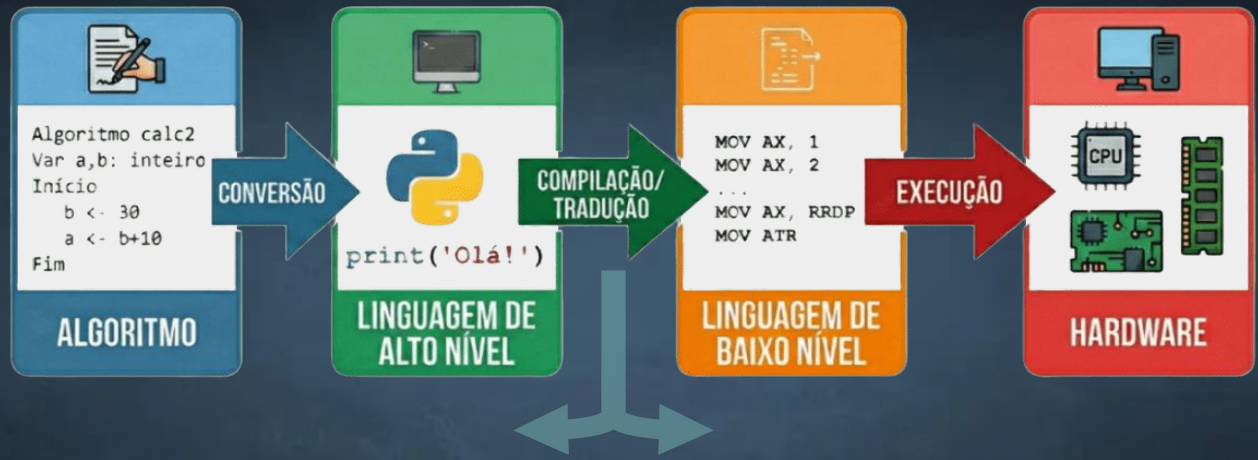
*O algoritmo é uma ferramenta para resolver um problema computacional bem específico, sendo uma sequência de etapas que recebe um valor ou conjunto de valores como entrada e os transforma em saída.*

# NÍVEIS DE LINGUAGEM



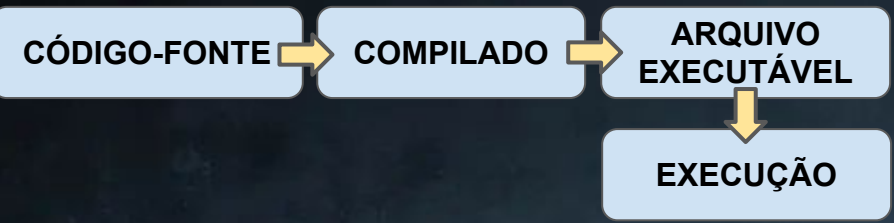
\* específico para um sistema operacional / família de processadores

# COMPILAÇÃO X INTERPRETAÇÃO



*Compilação: converte integralmente o código-fonte de um programa para linguagem de máquina antes de sua execução, gerando um arquivo executável independente.*

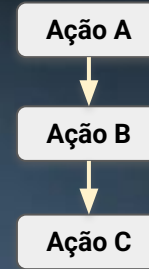
*Interpretação: o código-fonte é traduzido e executado linha a linha, em tempo real, por um programa intermediário, sem a geração prévia de um arquivo executável independente.*



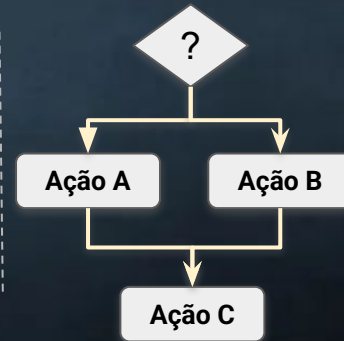
# ESTRUTURAS BÁSICAS DE UM ALGORITMO

Cap 2 / Pág:30

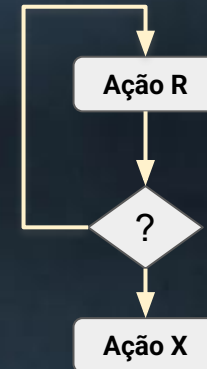
► **SEQUENCIAÇÃO** - As instruções serão executadas sequencialmente, uma após o término da anterior, de cima para baixo e da esquerda para a direita.



► **SELEÇÃO** - o fluxo de execução das instruções a ser seguido é determinado pelo resultado da avaliação de uma ou mais condições, sendo elas uma expressão condicional e/ou lógica que pode ser verdadeira ou falsa.




► **REPETIÇÃO** - o fluxo é alterado por meio da repetição de trechos de código em um número pré-definido de vezes ou enquanto uma determinada condição for satisfeita.



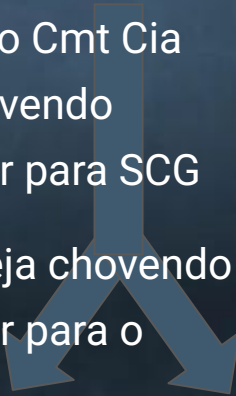
## ▶ SEQUENCIAÇÃO

### Sessão de aula

1. Turma em sala
  2. Preparar sala
  3. Tirar faltas
  4. Apresentar turma
  5. Sentar
  6. Assistir aula
  7. Apresentar por encerramento
- 

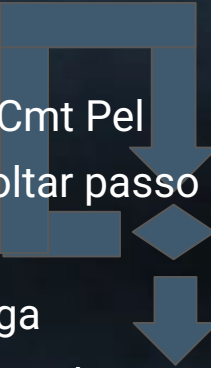
## ▶ SELEÇÃO

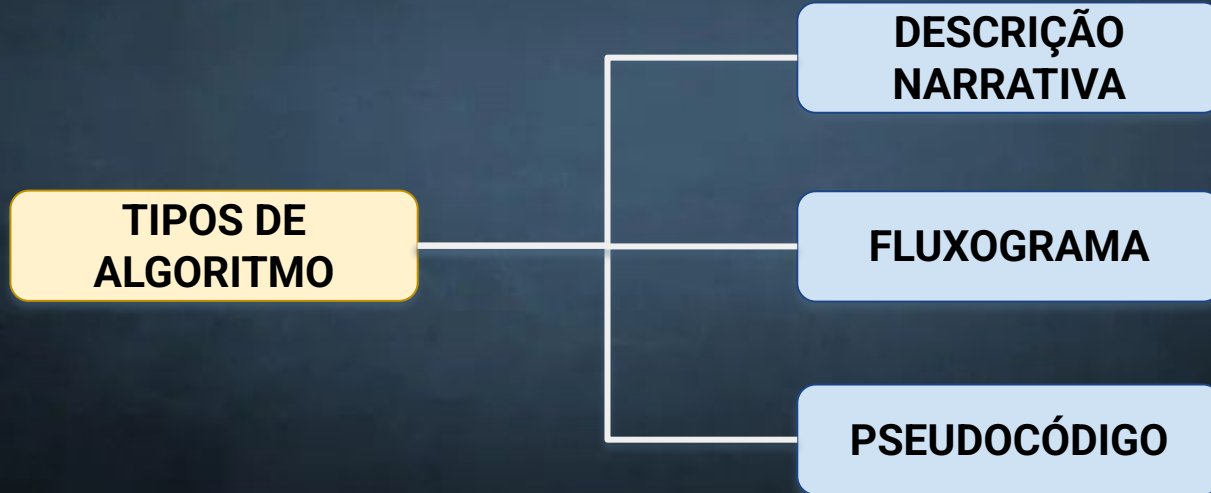
### Atividade conforme clima

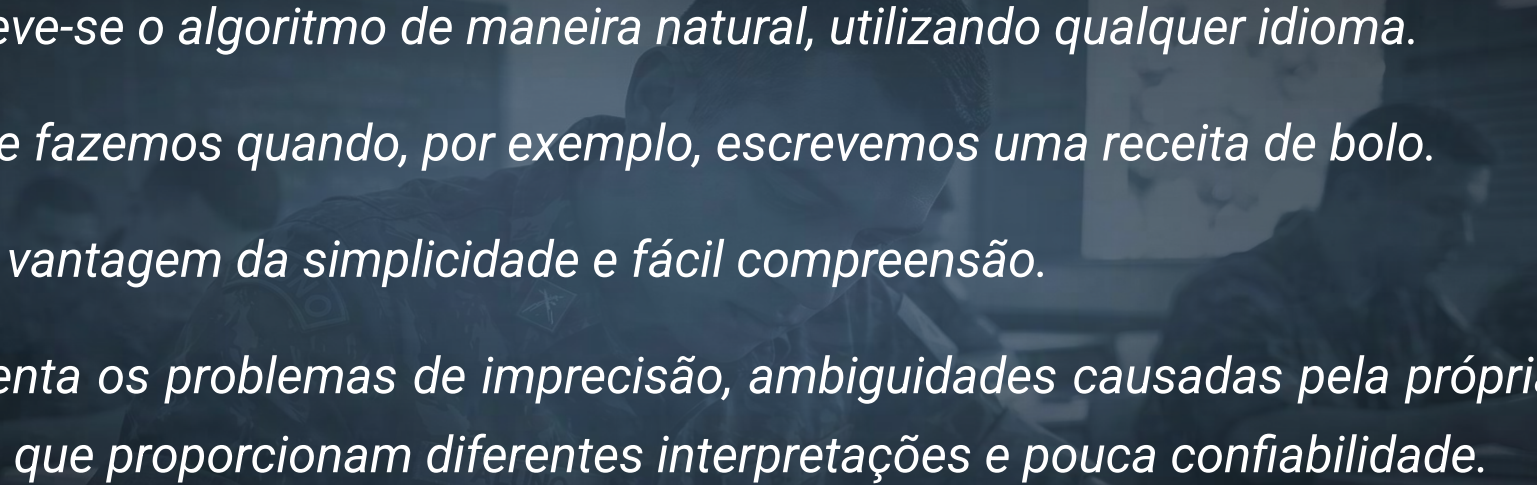
1. Cia pronta na ala
  2. Tiragem de falta
  3. Orientações do Cmt Cia
  4. Se estiver chovendo
    - ↳ 4a. Conduzir para SCG
  - Caso não esteja chovendo
    - ↳ 4b. Conduzir para o Pátio
  5. Formatura
- 

## ▶ REPETIÇÃO

### Inspeção de armamento

1. Desmontar armamento
  2. Limpar e manter
  3. Montar armamento
  4. Deixar aberto
  5. Inspeção pelo Cmt Pel
    - ↳ Se não ok, voltar passo 1
    - ↳ Se ok prossiga
  6. Entregar armamento
- 



- *Descreve-se o algoritmo de maneira natural, utilizando qualquer idioma.*
  - *É o que fazemos quando, por exemplo, escrevemos uma receita de bolo.*
  - *Tem a vantagem da simplicidade e fácil compreensão.*
  - *Apresenta os problemas de imprecisão, ambiguidades causadas pela própria língua, que proporcionam diferentes interpretações e pouca confiabilidade.*
- 
- A background image showing a student in a classroom setting, focused on writing in a spiral notebook. The student is wearing a dark shirt and a watch on their left wrist. The scene is dimly lit, with a blue tint, and other students are faintly visible in the background.

*Exemplo de algoritmo que verifica se um número inteiro é par ou ímpar, usando descrição narrativa:*

*Início*

*(1) Obter o número.*

*(2) Obter o resto da divisão do número por 2.*

*(3a) Se o resto for zero, informe que número é par.*

*(3b) Caso contrário, informe que número é ímpar.*

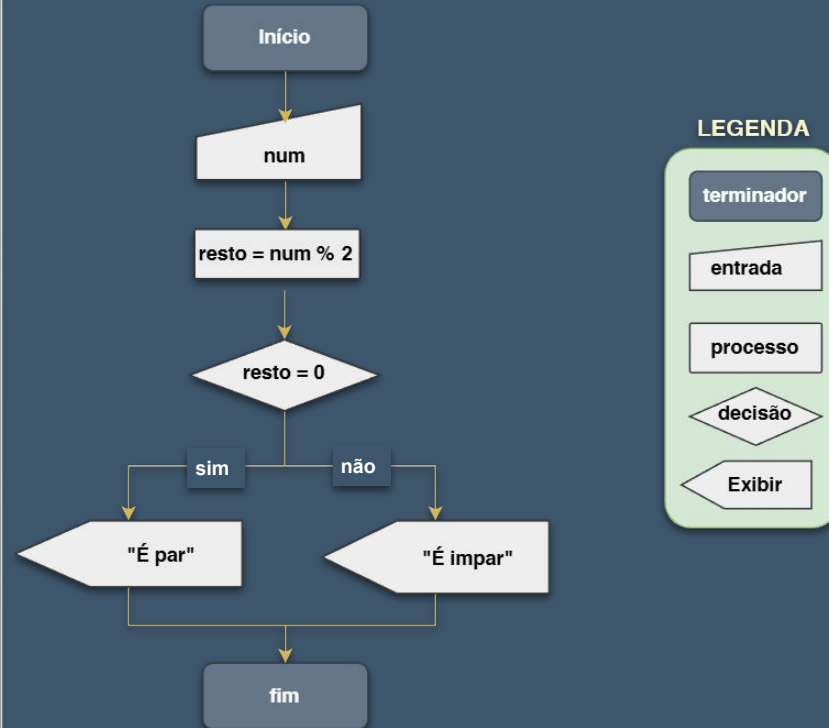
*Fim*

# FLUXOGRAMA (OU DIAGRAMA DE BLOCOS)

Diagrama que utiliza símbolos gráficos padronizados para representar as diversas ações do algoritmo.

Possui a vantagem de que uma figura pode dizer o que seria necessário a utilização de várias palavras, porém, apresenta a desvantagem da necessidade do entendimento do significado de cada símbolo utilizado, além de se tornar complicado para representação de algoritmos grandes.

Exemplo de fluxograma que verifica se um número é par ou ímpar:



- ▶ *Pseudolinguagem de programação para representar um algoritmo, cujas instruções são escritas em qualquer idioma, porém com regras que restringem e estruturam seu uso.*
- ▶ *Não apresenta as ambiguidades de uma língua nem os rigores de uma linguagem de programação. Possibilita a conversão quase direta para uma linguagem de programação.*
- ▶ *Escrito em português é também chamado portugol.*

*Início*

*Leia (num).*

*resto ← num % 2*

*se resto=0 então*

*escreva("Par")*

*senão*

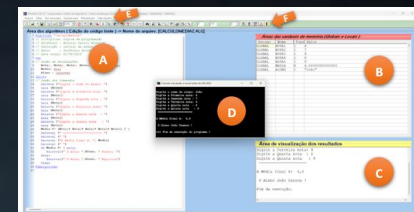
*escreva("Ímpar")*

*fimSe*

*Fim*

- ▶ Algoritmos não são concebidos para execução imediata por computadores. Eles operam em um nível de abstração focado exclusivamente na resolução passo a passo de um problema.
- ▶ Na disciplina de IC, é facultado ao aluno o uso do software Visualg, que permite a execução dos algoritmos escritos, possibilitando a visualização funcional do código. Para tanto, há a necessidade de conhecimento das regras sintáticas do pseudocódigo adotado na disciplina.
- ▶ O anexo “A” desta apostila trata do software Visualg: a instalação\*, uso e guia de referência.

*\* baixe do nosso site, já preparado com os códigos da apostila*



2.1 Utilizando a descrição narrativa, represente um algoritmo para a troca de um pneu furado de um automóvel.

2.2 Elabore um fluxograma que leia duas notas de um aluno (N1 e N2). O algoritmo deve calcular a média aritmética e verificar se o aluno foi aprovado.

2.9 Faça um algoritmo para somar dois números e multiplicar o resultado pelo primeiro número (PSEUDOCÓDIGO).