

A descoberta do conhecimento  
é um dos acontecimentos mais  
importante da vida. A  
inteligência e a criatividade  
humana é insuperável!



# Computational Logic Using Python

## AULA 02

# Introdução à Lógica Computacional

Conjunto de princípios e regras que permitem a criação de algoritmos para resolver problemas de forma eficiente.

Envolve conceitos como **proposições, operadores lógicos, estruturas condicionais e repetitivas**, e serve como base para qualquer linguagem de programação.

A lógica computacional é baseada na **Lógica Matemática** e na **Lógica Booleana**, permitindo que computadores **tomem decisões e executem ações de maneira estruturada e previsível**.

# LÓGICA BOLEANA

A Lógica Booleana é um sistema matemático que trabalha com valores verdadeiros e falsos, geralmente representados como:

- Verdadeiro (True ou 1)
- Falso (False ou 0)

Essa lógica é a base da programação e dos circuitos digitais, permitindo a construção de expressões condicionais e tomadas de decisão.

# LÓGICA BOLEANA

A Lógica Booleana é um sistema matemático que trabalha com valores verdadeiros e falsos, geralmente representados como:

- Verdadeiro (True ou 1)
- Falso (False ou 0)

Essa lógica é a base da programação e dos circuitos digitais, permitindo a construção de expressões condicionais e tomadas de decisão.

# Introdução à Lógica Computacional

- Proposições e valores lógicos (Verdadeiro ou Falso).
- Operadores lógicos: E (AND), OU (OR), NÃO (NOT).
- Tabela-verdade para operadores lógicos.

## Exemplo de Tabela-Verdade

| A | B | A AND B | A OR B | NOT A |
|---|---|---------|--------|-------|
| V | V | V       | V      | F     |
| V | F | F       | V      | F     |
| F | V | F       | V      | V     |
| F | F | F       | F      | V     |

# Testando operadores lógicos

```
programa {  
    funcao inicio() {  
        logico A = verdadeiro  
        logico B = falso  
  
        escreva("A AND B: ", A e B, "\n")  
        escreva("A OR B: ", A ou B, "\n")  
        escreva("NOT A: ", nao A, "\n")  
    }  
}
```

Saída esperada:

```
A AND B: falso  
A OR B: verdadeiro  
NOT A: falso
```

# Importância da Lógica Computacional na Programação

## 1. Facilita a resolução de problemas

- Ensina a dividir um problema complexo em partes menores e solucionáveis.
- Exemplo: Um sistema de login pode ser dividido em **entrada de dados**, **validação de credenciais** e **resposta ao usuário**.

## 2. Permite a criação de algoritmos eficientes

- Algoritmos bem estruturados economizam **tempo e processamento**.
- Exemplo: Um bom algoritmo de busca em uma lista pode reduzir o tempo de resposta do sistema.

## 3. Fundamental para qualquer linguagem de programação

- Independente da linguagem (Python, Java, C, etc.), a lógica computacional **não muda**.
- Se você entende lógica, aprende qualquer linguagem mais facilmente.

# Importância da Lógica Computacional na Programação

## 4. Evita erros e facilita a depuração (debugging)

- Programas bem estruturados têm menos bugs e são mais fáceis de manter.
- Exemplo: Se um programa está falhando, um bom entendimento lógico permite encontrar rapidamente a causa do erro.

## 5. É usada em Inteligência Artificial, Machine Learning e Ciência de Dados

- Algoritmos de IA e Machine Learning dependem de uma boa modelagem lógica para funcionarem corretamente.
- Exemplo: Regras lógicas são usadas para classificar dados e tomar decisões automáticas.



# Exemplo Prático: Como a Lógica Computacional é Aplicada

**Problema:** Um sistema precisa verificar se um aluno passou de ano.

**Regras:**

- O aluno precisa de nota maior ou igual a 7.
- Se a nota for entre 5 e 6.9, ele está de recuperação.
- Se for menor que 5, ele está reprovado.

# Solução em Portugol

```
programa {  
  funcao inicio() {  
    real nota  
  
    escreva("Digite a nota do aluno: ")  
    leia(nota)  
  
    se (nota >= 7) {  
      escreva("Aprovado\n")  
    } senao se (nota >= 5 e nota < 7) {  
      escreva("Recuperação\n")  
    } senao {  
      escreva("Reprovado\n")  
    }  
  }  
}
```

**Esse código segue uma lógica clara:**

- 1.** Se a nota for maior ou igual a 7, o aluno está aprovado.
- 2.** Se não, mas for maior ou igual a 5, está em recuperação.
- 3.** Caso contrário, está reprovado.