

# Introdução à Programação

## Lógica de Programação

Entenda a importância da lógica de programação como a base para qualquer linguagem. Aprenda a estruturar seu pensamento para resolver problemas de forma eficiente antes de se aprofundar em linguagens específicas.

## Conceitos Fundamentais

Explore conceitos essenciais como variáveis e estruturas de repetição. Compreenda como esses elementos são utilizados para automatizar tarefas e manipular dados de forma eficaz em qualquer linguagem de programação.

## Ferramentas de Programação

Familiarize-se com ferramentas como Portugol, Google Colab e Jupyter Notebook. Descubra como essas plataformas podem facilitar seu aprendizado e prática de programação, preparando-o para o uso de linguagens como Python.

Quais linguagens veremos no curso? Python, SQL e um pouco de R.

### PRINCIPAIS LINGUAGENS PARA CADA ÁREA



#### DESENVOLVIMENTO WEB

- JavaScript
- HTML/CSS
- Python
- Ruby



#### CIÊNCIA DE DADOS

- Python
- R



#### DESENVOLVIMENTO SOFTWARE

- JAVA
- C#



# PORQUE ESTUDAR LÓGICA DE PROGRAMAÇÃO?

**O CONCEITO:** POR TRÁS DAS LINGUAGENS TEMOS UMA ESTRUTURA DE CONCEITO E FLUXO MUITO CONSOLIDADA QUE É COMPARTILHADA ENTRE PRATICAMENTE TODAS LINGUAGENS..

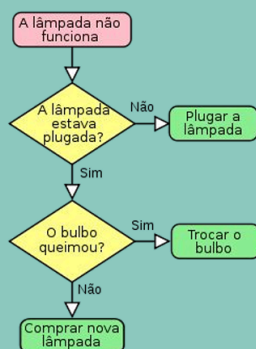
**SE EU APRENDER LÓGICA APRENDEREI TODAS LINGUAGENS?**  
NÃO, MAS COM TODA CERTEZA SERÁ MUITO MAIS FÁCIL APRENDER QUALQUER LINGUAGEM NOVA UMA VEZ QUE A BASE DA LÓGICA DA PROGRAMAÇÃO ESTÁ BEM CONSTRUÍDA.

## O QUE APRENDEREMOS AGORA?

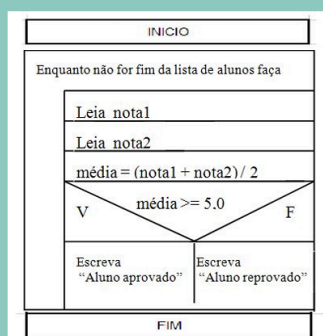
POR ENQUANTO NÃO PROGRAMAREMOS EM UMA LINGUAGEM ESPECIFICA MAS APRENDEREMOS A LÓGICA POR TRÁS DOS CÓDIGOS QUE GERAM COMANDOS AS MÁQUINAS.

## APRENDIZADO DA LÓGICA DE PROGRAMAÇÃO

### FLUXOGRAMA



### NASSI SHNEIDERMAN



### PORTUGOL

```
programa {  
  funcao inicio() {  
    inteiro valor = 1  
  
    escolha (valor) {  
      caso 0: //testa se o valor é igual a 0  
        escreva ("o valor é igual a 0")  
        pare  
      caso 1: //testa se o valor é igual a 1  
        escreva ("o valor é igual a 1")  
        pare  
      caso 2: //testa se o valor é igual a 2  
        escreva ("o valor é igual a 2")  
        pare  
      caso contrario:  
        escreva ("o valor não é igual a 0, 1 ou 2")  
    }  
  }  
}
```

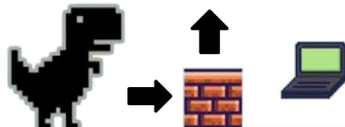
MÉTODO QUE IREMOS APRENDER

## COMO ESTRUTURAR UM ALGORITMO?

### ALGORITMO

Algoritmos são uma sequência de comandos que criam uma sequência de ações que deve ser executada pela máquina. Podemos entender como uma **rotina**.

UM ALGORITMO PARA O DINOSSAURO CHEGAR AO COMPUTADOR:



### Algoritmo Nível 1

Avançar para Frente  
Pular  
Avançar para Frente  
Pegar o Notebook  
Fim Algoritmo

### INDEXAÇÃO

Processo de organizar a estrutura de dados para permitir acesso eficiente aos comandos descritos.

## COMO ESTRUTURAR UM ALGORITMO?

### CONDICIONAL

Estruturas condicionais nos permitem que tomemos decisões baseadas em condições, geralmente representadas pelo "SE" e "SE NÃO".



### Algoritmo Nível 2

Avançar para Frente  
Olhar para o alto  
Se estiver vindo o meteoro  
Avançar para Trás  
Senão  
Avançar para Frente  
Pegar Notebook  
Fim-Se  
Fim Algoritmo

## VARIÁVEL:

- **O que é?**  
É um símbolo que armazena um valor.
- **Valores armazenados:**  
Podem armazenar valores dos mais diversos tipos, palavras, números e afins,
- **Regras:**  
As variáveis não podem conter caracteres especiais, espaço, acentuação ou palavras reservadas.
- **Tipo de dado:** Antes de declararmos nossa variável é importante expressarmos que tipo de dado ela se refere.

### INTEIRO:

Todo e qualquer número inteiro.

**Ex:** 1,2,3,1000,1239  
IDADES, QUANTIDADES

### REAL:

Todo e qualquer número real.

**Ex:** 1.77 | 1.885 | 235.4  
PREÇOS, ALTURAS, PESO

### CADEIA:

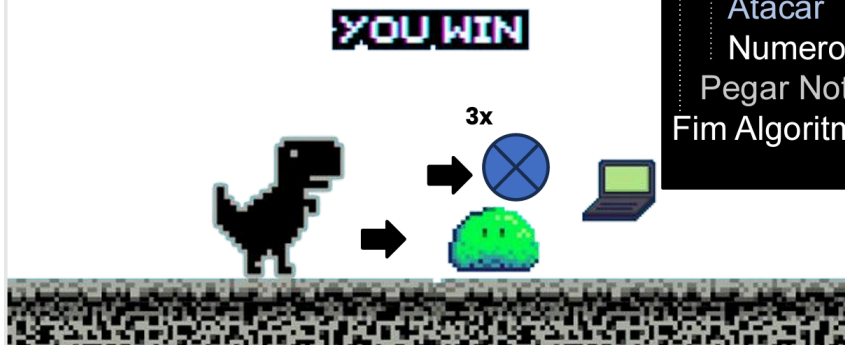
Palavras de texto.

**Ex:** Ana, Joao, EBAC  
NOMES, CIDADES, TIMES

## COMO ESTRUTURAR UM ALGORITMO?

### REPETIÇÃO:

Para facilitar nossas tarefas, quando temos que repetir um processo por determinadas X utilizamos estruturas de repetição "Enquanto", "Para".



Algoritmo FaseFinal  
NumeroAtaque INTEIRO

**NumeroAtaque = 0**

Avançar para Frente

Enquanto NumeroAtaque < 3:

Avançar para Frente

Atacar

NumeroAtaque + 1

Pegar Notebook

Fim Algoritmo

## A Lógica da programação para Ciência de Dados



- Logicamente o Cientista não construirá Jogos.
- Programação voltada para dados que podem ser numéricos ou categóricos.
- Utiliza a mesma lógica e conceitos vistos na aula, mas geralmente envolvendo aplicações matemáticas.
- Facilitação ao trabalhar com alto volume de dados (Big Data).
- Ao invés de estarmos aplicando a jogos estaríamos pedindo, por exemplo, que o algoritmo realizasse um cálculo repetidas vezes.



# Fluxograma

Uma representação visual de um processo ou algoritmo, usando símbolos padronizados para descrever as etapas e a sequência de ações necessárias para resolver um problema ou executar uma tarefa.

# Portugol

Uma ferramenta educacional que permite a prática de lógica de programação em português, facilitando o aprendizado de conceitos básicos antes de migrar para linguagens de programação mais complexas como Python.

# Compreensão da Lógica de Programação

Antes de mergulhar em qualquer linguagem de programação, é crucial entender a lógica de programação. Isso serve como um alicerce para resolver problemas de forma eficiente e compreender algoritmos. A lógica de programação ajuda a estruturar o pensamento e a criar soluções sistemáticas para problemas complexos.

# Uso de Ferramentas de Ensino

Ferramentas como fluxogramas e Portugol são excelentes para iniciantes praticarem lógica de programação. Elas permitem que os alunos visualizem o fluxo de execução de um programa e experimentem com algoritmos simples em português, facilitando a transição para linguagens como Python.

# Importância das Variáveis

Variáveis são fundamentais na programação, permitindo o armazenamento e manipulação de dados. Compreender como declarar e usar variáveis, bem como os diferentes tipos de dados, é essencial para qualquer programador iniciante.

# Estruturas de Repetição

Estruturas de repetição, como "enquanto" e "para", são cruciais para automatizar tarefas repetitivas. Elas permitem que um bloco de código seja executado várias vezes, economizando tempo e esforço na programação.

## Exemplo de utilização no mercado de trabalho:

**Automação de Processos:** A lógica de programação e o uso de estruturas de repetição são frequentemente utilizados para automatizar processos em empresas, como a geração de relatórios ou o processamento de grandes volumes de dados.

**Desenvolvimento de Aplicações:** Compreender variáveis e lógica de programação é essencial no desenvolvimento de aplicações, onde é necessário manipular dados e criar funcionalidades que respondam a diferentes condições e entradas do usuário.

## Como o Portugol ajuda no aprendizado da lógica de programação?

O Portugol ajuda no aprendizado da lógica de programação ao permitir que os alunos pratiquem a escrita de algoritmos em português. Isso facilita a compreensão dos conceitos lógicos sem a barreira adicional de uma nova linguagem, preparando os alunos para transitar para linguagens como Python.

## O que são variáveis e por que são importantes na programação?

Variáveis são espaços de armazenamento que mantêm dados que podem ser alterados durante a execução de um programa. Elas são importantes porque permitem que os programas manipulem dados de forma dinâmica, facilitando a automação de tarefas e a execução de operações complexas.

## Como as estruturas de repetição são utilizadas na programação?

Estruturas de repetição, como "enquanto" e "para", são usadas para automatizar tarefas repetitivas em um programa. Elas permitem que um bloco de código seja executado várias vezes, economizando tempo e esforço ao evitar a necessidade de escrever código repetitivo manualmente.

## Por que é importante praticar a lógica de programação com diferentes variáveis e cenários?

Praticar a lógica de programação com diferentes variáveis e cenários é importante porque ajuda a consolidar o aprendizado e a desenvolver uma compreensão mais profunda dos conceitos. A prática constante permite que os alunos experimentem e aprendam a aplicar a lógica de programação de forma eficaz em diferentes contextos.

## Qual é a importância de uma base sólida em lógica de programação?

Uma base sólida em lógica de programação é crucial para evitar erros comuns e aplicar conhecimentos de forma eficaz. Compreender a lógica subjacente aos algoritmos permite que os programadores desenvolvam soluções eficientes e robustas, independentemente da linguagem de programação utilizada.

## Competência:

Compreender e aplicar a lógica de programação para resolver problemas e desenvolver algoritmos eficientes utilizando ferramentas como Portugol e plataformas como Google Colab.

## Habilidades:



Definir variáveis e estruturas de repetição para automatizar tarefas repetitivas em algoritmos.

## 2

Descrever a lógica de programação utilizando fluxogramas e a ferramenta Portugol para facilitar o aprendizado.