

# Aula 01 – Linguagens de Programação e Lógica

---

Disciplina: Algoritmos

Prof. Allbert Velleniche de Aquino Almeida

E-mail: [allbert.almeida@fatec.sp.gov.br](mailto:allbert.almeida@fatec.sp.gov.br)

Site: <http://www.allbert.com.br>



/allbert.almeida

# Resumo da aula

---

- Objetivo
- Definição
- Nível de Linguagem
- Tipos de Linguagem
- Interpretadores X Compiladores
- Lógica Formal e Lógica Booleana
- Lógica de Programação

# Objetivo

---

- O objetivo dessa aula é fazer uma introdução a Linguagem de Programação com o foco nos conceitos e como aplicamos a lógica na investigação da verdade.

# Definição de Linguagem de Programação:

---

- Aurélio: linguagem em que se expressa um **conjunto de ações** de forma aceitável pelo computador;
- Uma linguagem de programação (LP) é uma ferramenta utilizada pelo profissional de informática para **escrever programas**;
  - É um conjunto de instruções a serem executadas pelo computador para realizar um determinado processo.

# Nível de Linguagem

---

- Linguagem de Máquina
- Linguagem de Baixo de Nível
- Linguagem de Alto Nível

# Nível de Linguagem

---



# Linguagem de Máquina

---

- É a única linguagem diretamente compreendida pelo computador, e está intrinsecamente **relacionada ao projeto de hardware** do mesmo. Geralmente, essas linguagens consistem em um **conjunto de caracteres (0s e 1s)**, que definem cada operação elementar que deve ser realizada pelo computador para a execução de uma dada aplicação.

# Linguagem de Baixo de Nível

---

- Dada a dificuldade de se programar em linguagem de máquina, surgiu a linguagem de baixo nível Assembly. O Assembler é o programa que traduz a linguagem Assembly em linguagem de máquina.



# Linguagem de Alto Nível

---

- Essas linguagens possuem instruções simples, normalmente próximas do inglês e das notações matemáticas, para executar tarefas básicas que equivalem, em linguagem de máquina ou mesmo assembly, a um **conjunto de instruções elementares**. Exemplos: Fortran, COBOL, Pascal, C, C++, Java, etc.

# Linguagem de Alto Nível

---

- Linguagens não Estruturadas
- Linguagens Estruturadas
- Linguagens Orientadas a Objetos

# Linguagens não Estruturadas

---

- São aquelas linguagens que não aceitam blocos de estruturas. Um programa, então, é uma **sequência de instruções que se utiliza**, quando necessário, **de desvios (vá para)**.
- Como exemplo dessas linguagens tem-se: Fortran, Cobol, etc.

# Linguagens Estruturadas

---

- São aquelas linguagens que incorporam blocos de **estruturas de repetições** (while, do while, etc.).
- Essas linguagens visam tornar os programas menores, mais claros, mais fáceis de serem compreendidos e depurados, através, principalmente, da **eliminação dos desvios**.
- Exemplo: C, Cobol Estruturado, Pascal, Fortran-77, etc.

# Linguagem Não-estruturada X Estruturada

\*\*\* Potenciação - Desvios \*\*\*\*

Início:

```
real: num, exp, total, contador;  
escreva(" Entre com um número : ");  
leia (num);  
escreva(" Entre com um expoente : ");  
leia (exp);  
total ← 1;  
contador ← 1;
```

Expoente:

```
se (contador > exp) vá para Imprime;  
total ← total * num;  
contador ← contador + 1;  
vá para Expoente;
```

Imprime:

```
escreva ("Resultado : ", total);
```

Fim.

\*\*\* Potenciação - Estruturada \*\*\*\*

Início:

```
real: num, exp, total, contador;  
escreva(" Entre com um número : ");  
leia (num);  
escreva(" Entre com um expoente: ");  
leia (exp);  
total ← 1;
```

**para** contador **de** 1 **até** exp **faça**

```
total ← total * num;
```

**fim\_para**;

```
escreva ("Resultado : ", total);
```

Fim.

# Linguagens Orientadas a Objetos

---

- Essas linguagens expressam um novo modelo dentro da programação. Normalmente, incorporam as estruturas das **linguagens estruturadas** e acrescentam às mesmas a capacidade de lidar com o **paradigma de Objetos**.
- Exemplos: C++, Java.

# Linguagem de Alto Nível

---

- Os programas escritos em linguagens de alto nível devem ser traduzidos para linguagem de máquina através de:
  - Interpretadores
  - Compiladores

# Interpretadores

---

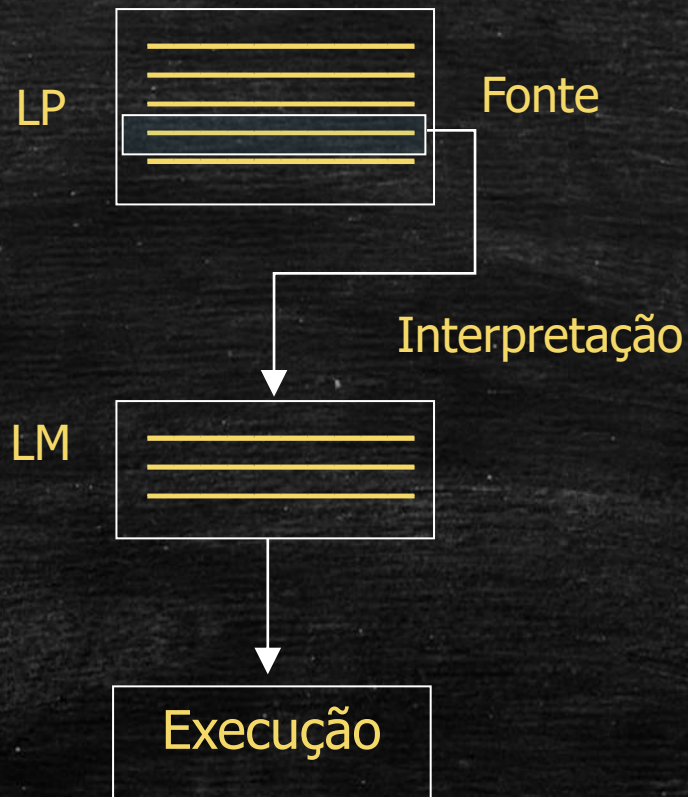
- Um interpretador lê uma instrução de um programa escrito em linguagem de alto nível, faz uma consistência de sua sintaxe e, caso esteja correta, a converte em uma ou mais instruções em linguagem de máquina e as executa. Segue, então, para a próxima instrução e assim sucessivamente até o término da execução do programa.



# Interpretadores

---

## Interpretação pura



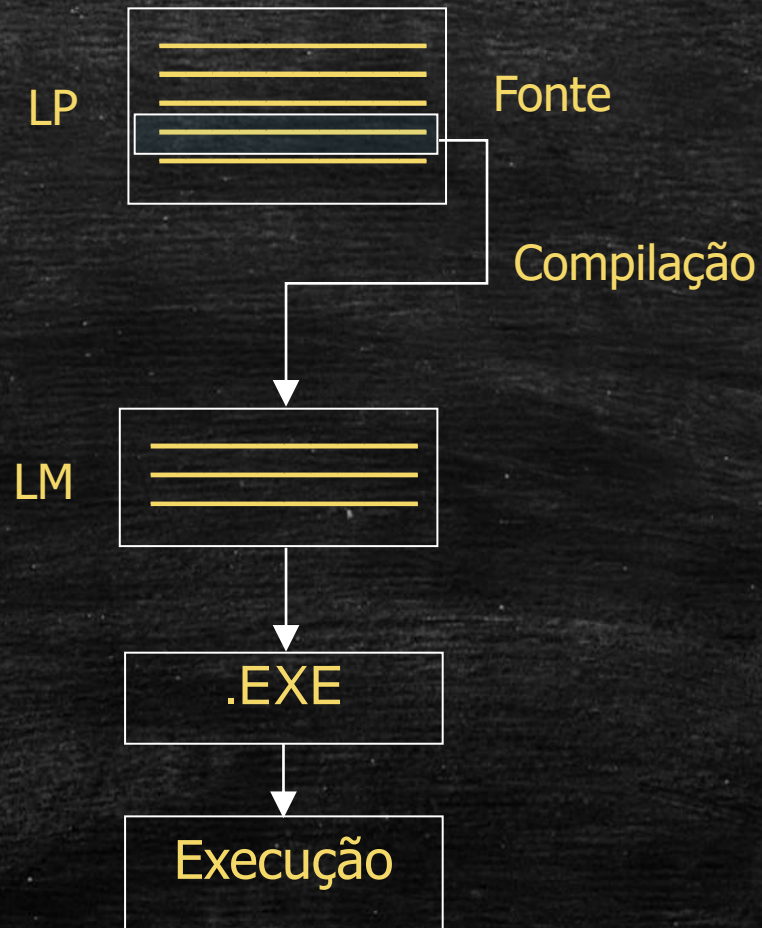
# Compiladores

---

- Um compilador lê uma instrução de um programa escrito (código fonte), **faz uma consistência de sua sintaxe** e, caso esteja correta, a converte em uma ou mais instruções em linguagem de máquina. Em seguida, são agregadas a este código rotinas em linguagem de máquina que possibilitarão sua execução, sendo criado neste ponto o **arquivo executável (.exe)**.

# Compiladores

---



# Lógica - Conceito

---

- A Lógica tem, por objeto de estudo, as leis gerais do pensamento, e as formas de aplicar essas leis corretamente na investigação da verdade.

# Lógica de Programação

---

- A lógica, basicamente, **trata da correção do pensamento**, ou seja, nos ensina a usar os critérios do pensamento.
- Em termos de programação, é a técnica de **encadear o raciocínio para atingir um determinado objetivo ou resolver um dado problema**. Essa forma de encadeamento é chamada, em Lógica, de argumento.

# Argumento

---

- Um argumento é uma sequência de **proposições** na qual uma delas é a **conclusão** e as demais são **premissas**. As premissas justificam a conclusão.
  - Proposições: sentenças afirmativas que podem ser verdadeiras ou falsas
  - Premissas: afirmações disponíveis
- Exemplo:
  - Todo aluno de Computação precisa estudar Lógica. (*premissa*)
  - José é aluno de Computação. (*premissa*)
  - Logo, José precisa estudar Lógica. (*conclusão*)

# Argumento

---

- O objetivo de um argumento é justificar uma afirmação que se faz, ou dar as razões para uma certa conclusão obtida.
- Exemplo:

Você me traiu. Pois, disse que ia estudar e meu irmão lhe viu na boate.
- Um argumento demonstra/prova como a partir dos dados de um problema chegou-se a uma conclusão.

# Argumento: Raciocínio e Inferência

---

- Exercício:

Um rei resolveu dar a liberdade a um de seus três prisioneiros. Mandou trazer três chapéus brancos e dois vermelhos. Vendou os olhos dos prisioneiros, colocou um chapéu em cada um e depois foi retirando a venda dos olhos deles. Ganharia a liberdade aquele que soubesse dizer, de forma convincente, a cor do seu próprio chapéu olhando para os outros prisioneiros. Os dois primeiros não souberam dizer. O terceiro, antes que o rei lhe tirasse a venda dos olhos, afirmou com toda certeza a cor do seu chapéu.

Qual a cor do chapéu do terceiro prisioneiro? Justifique.



# Resposta

Possibilidades \ Prisioneiros	A	B	C
1	B	B	B
2	B	B	V 
3	B	V	B
4	V	B	B
5	B	V	V 
6	V	B	V 
7	V	V	B

# Argumento: Raciocínio e Inferência

---

- Para convencer que você sabe a resposta (que não é um *chute*) você tem de expor as razões que o levaram a conclusão (justificar).



- Um argumento poderia ser considerado uma reconstrução explícita do raciocínio efetuado

# Argumento: Raciocínio e Inferência

---

- Inferência é a relação que permite passar das premissas para a conclusão, um “**encadeamento lógico**”;
- A palavra inferência vem do latim, Inferre, e significa “conduzir para”
- O objeto de estudo da lógica é determinar se a **conclusão de um argumento é ou não decorrente das premissas** (uma inferência).

# Validade de um Argumento

---

- Em um argumento **válido**, as premissas são consideradas provas evidentes da verdade da conclusão, caso contrário **não é válido**.
- Quando é válido, podemos dizer que a conclusão é uma **consequência lógica** das premissas, ou ainda que a conclusão é uma **inferência** decorrente das premissas.

# Validade de um Argumento

---

- **Exemplo 1:** O argumento que segue é válido?

Se eu ganhar na Loteria, serei rico.

Eu ganhei na Loteria.

Logo, sou rico.

→ **Válido**

*(a conclusão é uma decorrência  
lógica das duas premissas.)*

# Validade de um Argumento

---

- **Exemplo 2:** O argumento que segue é válido?

Se eu ganhar na Loteria, serei rico

Eu não ganhei na Loteria

Logo, sou pobre

→ **Não é Válido**

*(a conclusão não é uma decorrência  
Lógica das duas premissas.)*

# Validade de um Argumento

---

- A lógica se preocupa com o relacionamento entre as premissas e a conclusão, ou seja, com a **estrutura e a forma do raciocínio**. A verdade do conteúdo de cada premissa e da conclusão é estudo das demais ciências.

# Dedução e Indução

---

- A Lógica dispõe de duas ferramentas que podem ser utilizadas pelo pensamento na busca de novos conhecimentos: a dedução e a indução, que dão origem a dois tipos de argumentos: **Dedutivos e Indutivos**.



# Argumentos Dedutivos

---

- Exemplos de argumentos dedutivos:

- Todos os mamíferos são mortais.

- Todas as cobras são mortais.

Argumento Inválido

- Todas as cobras são mamíferos.

- Todos os diamantes são duros.

- Alguns diamantes são jóias.

Argumento Válido

- Algumas jóias são duras.

# Argumentos Indutivos

---

- Os Argumentos Indutivos não pretendem que suas premissas forneçam provas cabais da veracidade da conclusão, mas apenas que forneçam indicações dessa veracidade. (**possibilidade, probabilidade**)
- Seguem do **Raciocínio Indutivo**, isto é, obtém conclusões baseada em observações/experiências. Enquanto que um **Raciocínio Dedutivo** exige uma prova formal sobre a validade do argumento.

# Argumentos Indutivos

---

- Exemplo1:

Joguei uma pedra no lago, e ela afundou;

Joguei outra pedra no lago e ela também afundou;

Joguei mais uma pedra no lago, e ela também afundou;

Logo, se eu jogar uma outra pedra no lago, ela vai afundar.

# Argumentos Indutivos

---

- Exemplo2:

A vacina funcionou bem nos ratos.

A vacina funcionou bem nos macacos.

Logo, vai funcionar bem nos humanos.

- Exemplo3:

80% dos entrevistados vão votar no candidato X.

Logo, o candidato X vai vencer as eleições.

# Validade e Verdade

---

- **Verdade e Falsidade:** são propriedades das proposições, nunca dos argumentos
- **Validade ou Invalidade:** são propriedades dos argumentos dedutivos que dizem respeito a inferência ser ou não válida (raciocínio ser ou não correto)

# Validade e Verdade

---

- Exemplo 1

Toda baleia é um mamífero (V)

Todo mamífero tem pulmões (V)

Logo, toda baleia tem pulmões (V)

**→ Argumento válido e a conclusão verdadeira.**

# Validade e Verdade

---

- Exemplo 2

Toda aranha tem seis pernas (F)

Todo ser de seis pernas tem asas (F)

Logo, toda aranha tem asas (F)

→ **Argumento válido e a conclusão falsa**

# Lógica Clássica e Lógica Simbólica

---

- **Lógica Clássica** formula os argumentos em linguagem natural, mas enfrenta problemas de ambiguidade e de construções confusas.
- A **Lógica Simbólica** ou **Lógica Matemática** utiliza símbolos de origem matemática para formular os argumentos. *Trabalho iniciado pelo matemático inglês George Boole (1815 – 1864) – Álgebra Booleana. e consolidado pelo filósofo e matemático alemão Goottlob Frege (1848 – 1895)*



# Expressões lógicas ou booleanas

---

- Uma expressão lógica ou booleana é uma expressão cujos **operadores são lógicos** e cujos **operandos são expressões relacionais**, constantes e/ou variáveis do tipo lógico.
- O resultado da avaliação de uma expressão lógica é sempre um valor **falso ou verdadeiro**;

# Expressões lógicas ou booleanas

---

Operador	Matemática	Usado	Exemplos
Conjunção	e	E	a E b
Disjunção	ou	OU	a OU b
Negação	não	NÃO	NÃO a

## Tabela verdade do operador E

---

Você conhece JAVA	Você conhece C++	Resultado
<b>FALSO</b>	<b>FALSO</b>	<b>FALSO</b>
<b>FALSO</b>	<b>VERDADEIRO</b>	<b>FALSO</b>
<b>VERDADEIRO</b>	<b>FALSO</b>	<b>FALSO</b>
<b>VERDADEIRO</b>	<b>VERDADEIRO</b>	<b>VERDADEIRO</b>

## Tabela verdade do operador OU

---

Você conhece JAVA	Você conhece C++	Resultado
<b>FALSO</b>	<b>FALSO</b>	<b>FALSO</b>
<b>FALSO</b>	<b>VERDADEIRO</b>	<b>VERDADEIRO</b>
<b>VERDADEIRO</b>	<b>FALSO</b>	<b>VERDADEIRO</b>
<b>VERDADEIRO</b>	<b>VERDADEIRO</b>	<b>VERDADEIRO</b>