Introdução a Linguagem de Programação

**Linha de Evolução**

O Hardware limita o Software

1948

Primeira linguagem de máquina (programação)

Assembly

1950

COBOL

FORTRAM

LISP

1960 – 70

Paradigmas

C

Prolog

ML (Machine Learning)

1990

Linguagens de Alto Nível

C#

Java

JavaScript

Python

2000

Novos Conceitos

TypeScript

GO

**Problemas Computacionais**

Cada época possui seu problema e eles vão evoluindo com avanço da capacidade e poder computacional

- Problemas de decisão

Caractere lógico – sim ou não

- Problemas de busca

Relacionamento binário – x está em A?

- Problemas de otimização

Maximizar ou Minimizar uma função ou recurso.

**O que é uma linguagem de programação?**

É um método padronizado composto de um conjunto de regras sintáticas e semânticas de implementação de um **código fonte**.

**Código fonte**

Pode ser traduzido ou interpretado

conjunto de palavras com regras

Como o computador entende o programa?

Um programa é um amontoado de palavras se não for possível que o computador entenda.

**Tradução**

Programa fonte (linguagem de alto nível) 🡪 Compilador (executa a análise do programa escrita) traduzindo para linguagem de máquina (linguagem de baixo nível) 🡪 Execução do programa objeto (produto final que é lido pela máquina)

- Possui uma execução mais rápida

- Produz programas menores

Ex. C++ e Java

Interpretação

Executa o programa fonte diretamente

- Maior flexibilidade

- Mais lenta

Ex. Ruby, Python, JavaScript

Características importantes durante a criação de um programa

**Desenvolvimento de Programas**

*- Legibilidade (leitura)*

Facilidade de leitura

Compreensão

Ortogonalidade 🡪 Coerência nas instruções

Definição adequada das estruturas

*- Redigibilidade (escrita)*

Simplicidade da escrita

Suporte e abstração

Reuso do código

Expressividade

*- Confiabilidade*

Verificação de tipos

Trata exceções

Uso de ponteiros

Compatibilidade dentre compiladores

*- Custo*

Treinamento

Codificação

Compilação

Execução

Infraestrutura

Outras características: Atualizações, Uso de API, Disponibilidade de Ferramentas, Comunidade Ativa e Adoção pelo Mercado.

Análise de Código

**Léxica**

Leitura do código fonte caractere por caractere letra por letra e agrupar os caracteres em lexemas produzindo uma sequência de símbolos léxicos conhecidos como tokens.

Particionar 🡪 Classificar 🡪 Eliminar

Particionar: Identificar os **elementos** léxicos que são os tokens e vai agrupa-los

Classificar: Esses elementos são identificadores, palavras reservadas, número, strings... todos conteúdos relevantes para codificação do programa

Eliminar: Caracteres de espaços em branco, comentários coisas que não são necessárias para o funcionamento do programa

**Sintáxico**

Remete a forma.

Componente do sistema linguística que interligam os constituintes da sentença, atribuindo-lhe uma estrutura.

Varia de linguagem para linguagem.

**Semântica**

Remete ao significado

É o estudo do significado. Incide sobre a relação entre os significantes, como: palavras, frases, sinais e símbolos.

Paradigmas da Programação

**O que é?**

Forma de resolução de problemas com diretrizes e limitações especificas de cada paradigma utilizando linguagem de programação.

**Classificação**

Orientação a Objeto

Procedural: chamadas sucessivas e procedimento separados

Funcional: instruções são baseadas em funções

Estruturada: estrutura de blocos aninhados

Computação Distribuída: funções executadas de forma independentes

Lógico

**Paradigma Estruturado**

Ênfase em instruções sendo executadas em sequência

- Sequencia

- Decisão: testes lógicos

- Iteração: funções, laços e condições

Ex. C

*Utilização*

- Problemas simples e diretos

- Aprender programação

**Orientação a Objeto**

Paradigma de programação baseado na utilização de objetos e suas interações. Análogo ao muno real.

Objeto: é descrito por características especificas (o que tenho), comportamentos (o que sou capaz de fazer) e estado (como faço).

Método é uma função associada a uma classe

O que tenho (características) são os atributos

O que sou capaz de fazer (comportamentos) são os métodos

Como faço são os estados

Pilares da Orientação a Objeto

- Encapsulamento

- Polimorfismo

- Abstração

- Herança

Classe filha herda características, atributos da classe mãe

Também pode possuir características, comportamentos e atributos próprios e pode até sobrescrever o que foi herdado

Cada paradigma tem suas vantagens

POO

- Reuso de código

PE

- Problemas específicos e diretos