**CURSO JAVA COMPLETO**

**SEÇÃO 2: CONCEITOS DE PROGRAMAÇÃO**

**ALGORITMO:**

Sequência finita de instruções para se resolver um problema.

\*aplica-se a diversas áreas de conhecimento

Exemplo:

Problema: Lavar roupa suja

Algoritmo:

1. Colocar a roupa em um recipiente
2. Colocar um pouco de sabão e amaciante
3. Encher de água
4. Mexer tudo até dissolver todo o sabão
5. Deixar de molho por vinte minutos
6. Esfregar a roupa
7. Enxaguar
8. Torcer

**AUTOMAÇÃO:**

Consiste em utilizar máquina(s) para executar o procedimento desejado de forma automática ou semiautomática.

Resumo: **quando uma máquina realiza o algoritmo**

**COMPUTADOR:**

* Hardware – parte física (a máquina em si)
* Software – parte lógica (programas)
  + Sistemas operacional(Windows, Linux, iOS)
  + Aplicativos (apps de escritório, app de câmera, navegador web)
  + Jogos
  + Utilitários (Antivírus, compactador de arquivos)

**PROGRAMA ~ ALGORITMO**

Programa de computador **são algoritmos** executados pelo computador (em linhas gerais).

Conclusão: o computador é uma máquina que **automatiza** a execução de **algoritmos**.

Qualquer algoritmo? Não. Apenas algoritmos computacionais:

- Processamento de dados

- Cálculos

O que é preciso para se fazer um programa de computador?

Vamos precisar de :

* Uma **linguagem de programação:** regras **léxicas** e **sintáticas** para se escrever o programa.
* Uma **IDE:** software para editar e testar o programa.
* Um **compilador:** software para transformar o **código fonte** em **código objeto.**
* Um **gerador de código** ou **máquina virtual:** software que permite que o programa seja executado.

**LINGUAGEM DE PROGRAMAÇÃO, LÉXICA E SINTÁTICA**

Linguagem de Programação: É um conjunto de regras **léxicas** (ortografia) e **sintáticas** (gramática) para se escrever programas.

**Léxica:** Diz respeito à correção das **palavras** “isoladas” (ortografia).

**Exemplo (português): Linguagem de Programação:**

Correto: Cachorro Correto: main

Errado: Caxorro Errado: maim

**Sintática:** Diz respeito à correção das **sentenças** (gramática).

**Exemplo (Português): Linguagem de Programação:**

O cachorro está com fome x = 2 + y;

A cachorro está com fome x = + 2 y;

**IDE – AMBIENTE INTEGRADO DE DESENVOLVIMENTO**

É um conjunto de software utilizado para a construção de programas.

Exemplo:

C/C++: **Code Blocks**

Java: **Eclipse, NetBeans**

C#: **Microsoft Visual Studio**

Funcionalidades de uma IDE:

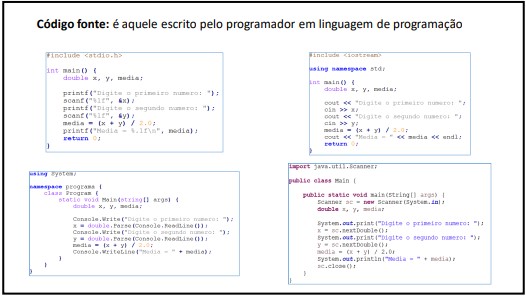
Uma **IDE** oferece várias funcionalidades para facilitar a construção dos programas

Edição de código fonte (endentação, autocompletar, destaque de palavras, etc.)

* Depuração e testes
* Construção do produto final (build)
* Sugestão de modelos (templates)
* Auxilia em várias tarefas do seu projeto
* Etc.

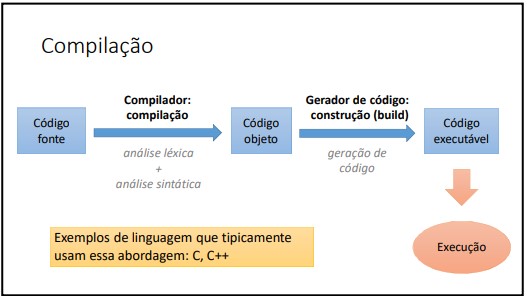
**COMPILAÇÃO E INTERPRETAÇÃO CÓDIGO FONTE E OBJETO MÁQUINA VIRTUAL**

**Código Fonte:** É aquele escrito pelo programador em linguagem de programação. Esse código, entretanto, não é entendido pelo computador nem pelo sistema operacional, ele tem que ser convertido em um código que possa ser executado.



**Compilação:** É um processo que transforma o código fonte no que a gente chama de código objeto esse processo é feito pelo software chamado compilador durante o processo de compilação o compilador faz a análise léxica e sintática , se tiver algum erro léxico ou sintático no código fonte a compilação é abortada e o programador é obrigado a corrigir os erros que porventura existam. Uma vez que o código fonte foi compilado e transformado em código objeto, esse código objeto vai ter que passar por um gerador de código para que seja feito o processo chamado construção(Build) e aí sim esse código executável pode ser executado pelo sistema operacional.

Esse modelo de compilação e execução que esta cedo mostrado na figura abaixo e adotado por algumas linguagens. Exemplo, Linguagem que tipicamente usam essa abordagem: C e C++.



**Interpretação:** É quando tem um software chamado interpretador, ele vai ler um código fonte e vai realizar a análise léxica, análise sintática e geração de código SOB DEMANDA, ou seja, gradualmente o interpretador vai lendo o código fonte traduzindo ele para o código executável e executando, esse modelo de interpretação e execução é usada por várias linguagens, tais como: PHP, JavaScript, Python, Ruby são as linguagens interpretadas.

Diagrama

Descrição gerada automaticamente com confiança média

**Abordagem Híbrida:** É quando tem um código fonte, esse código fonte passa por uma precompilação que faz à análise léxica, análise sintática, gerando o código objeto que recebe o nome de **BYTECODE** que é um código precompilado, porém ao invés de passar por um gerador de código e ser gerado o código executável o Bytecode vai ser executado por uma máquina virtual que faz a interpretação do Bytecode e faz a geração de código SOB DEMANDA e a execução do programa. Essa abordagem hibrida de compilação, interpretação e execução é adotada por algumas linguagens, tais como: Java (JVM), C# (Microsoft.NET Framework). A máquina virtual que executa o Bytecode do Java é chamada de JVM(Java Virtual Maxine) e a máquina virtual que executa o C# é chamado de Microsoft dotNet Framework.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**VANTAGENS**

**Compilação:**

Uma vantagem da compilação é a **Velocidade do programa.** Como todo processo de compilação e geração de código é feito previamente é natural que o programa vai ficar mais rápido, além disso quando você está mexendo com uma linguagem compilada você tem o maior **Auxílio do compilador antes da execução**, porque o compilado faz a análise léxica e sintática e não deixa executar se tiver algum erro léxico ou sintático no programa.

**Interpretação:**

**Flexibilidade de manutenção do aplicativo em produção:** Se você já tem um aplicativo que está rodando, na empresa que te contratou e você precisa alterar alguma coisa no aplicativo, basta trocar um ou mais dos arquivos fontes que o programa continua executando normalmente, não precisa recompilar todo o aplicativo para que ele volte a funcionar.

**Expressividade da linguagem:** quando consegue escrever uma solução elegante e complexa em poucas linhas

**Código fonte não precisa ser recompilado para rodar em plataformas diferentes:** se você quer que seu programa rode em Windows, Linux ou Mac, não precisa mudar nada no código fonte bastando ter um interpretador para cada uma dessas plataformas.

**Abordagem Híbrida do Java e C#:** ela pega algumas vantagens dos dois mundos, as setas grosas mostram as vantagens que são muito bem aproveitadas na abordagem Híbrida e as setas mais finas mostram as vantagens que são parcialmentes aproveitadas pela abordagem Híbrida, como mostra a imagem logo abaixo:

Uma imagem contendo Gráfico

Descrição gerada automaticamente

No caso das **linguagens compiladas**, C e C++: Você escreve seu código fonte, compila esse código fonte para objeto e finalmente gera o código executável. Esse código executável vai ser específico para cada sistema operacional e vai rodar logo em cima do sistema operacional que está rodando em cima do Hardware.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

As **linguagens Interpretadas** já funcionam um pouco diferente: Você escreve seu código fonte e esse código fonte vai ser interpretado pelo interpretador, específico pelo sistema operacional que você estiver usando. Perceba que quando você tem uma linguagem compilada, se você quiser rodar o seu mesmo programa em outro sistema operacional, vai ter que recompilar e gerar outro executável específico para aquele sistema operacional. Enquanto na linguagem interpretada basta trocar o interpretador que o mesmo código fonte vai funcionar em outro sistema operacional.

Interface gráfica do usuário

Descrição gerada automaticamente

As **linguagens Híbridas** como Java e C#, funcionam da seguinte forma: Você escreve seu código fonte, ele é precompilado para o Bytecode finalmente esse Bytecode vai ser executado por uma máquina virtual especifica do sistema operacional quando você estiver usando, se você quiser que seu Bytecode rode em outro sistema operacional, basta trocar a máquina virtual.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

**Resumo:** código fonte é aquele que o programado escreve. E o código objeto é aquele código que já é compilado, faz a análise léxica e sintática, gerado o código objeto. No caso das abordagens híbridas como Java e C# esse código objeto recebe o nome de Bytecode.

Modelo de execução: Compilação, Interpretação e abordagem híbrida, cada um desse modelos tem vantagens e desvantagens. Abordagem híbrida pega a vantagem de cada um, no caso da abordagem híbrida o programa executa com a máquina virtual, no caso da compilação precisa de um gerador de código para gerar o arquivo executável.

Para fazer um programa de computador nós precisamos:

* Linguagem de Programação
* IDE
* Compilador
* Gerador de código ou Máquina Virtual