Desenvolvimento de Sistemas

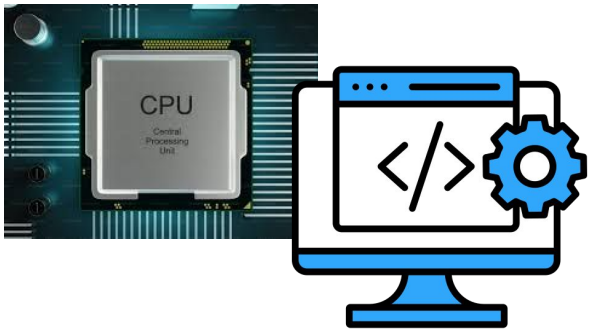
Aula 06 - Compiladores, Interpretadores e IDEs

Lorrany B A Marim

SENAI

A Linguagem do Computador

Nós não falamos binário



- **Linguagem de Máquina:** O computador só entende zeros e uns (linguagem de baixo nível). É difícil para humanos escreverem diretamente nela.

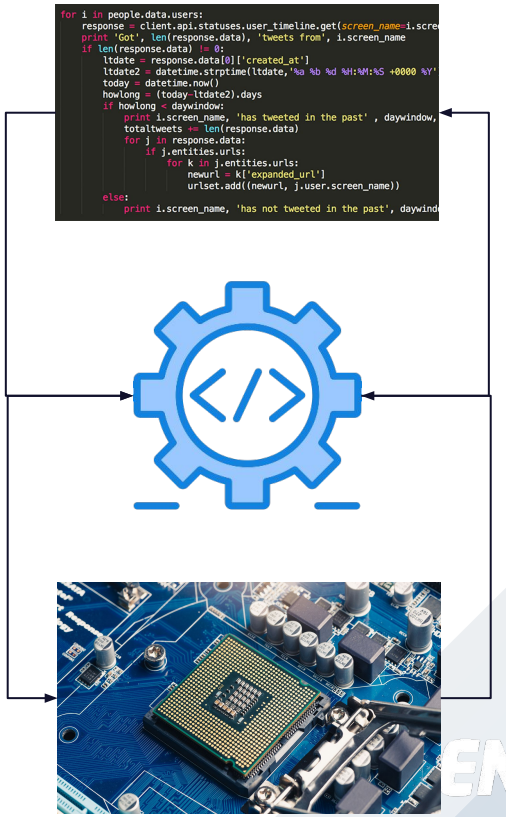
- **Linguagem de Alto Nível:** Linguagens como Python, Java e C# foram criadas para serem lidas e escritas facilmente por humanos.

A Linguagem do Computador

Nós não falamos binário

- **O Problema:** Como fazer o computador (que fala baixo nível) entender o meu código (que está em alto nível)?

- **A Solução:** Precisamos de um tradutor. Existem dois tipos principais: **Compiladores** e **Interpretadores**.

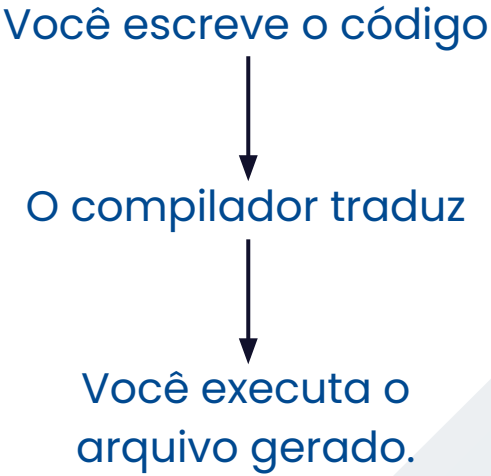


O Compilador

Traduzir tudo de uma vez

- **Definição:** Um compilador lê todo o código fonte de uma vez e o traduz completamente para um arquivo executável (código de objeto) antes de o programa rodar.
- **Vantagem:** Geralmente a execução é mais rápida, pois a tradução já foi feita.
- **Exemplos:** Linguagens como Java, C e C++ usam esse modelo tradicionalmente.

Processo

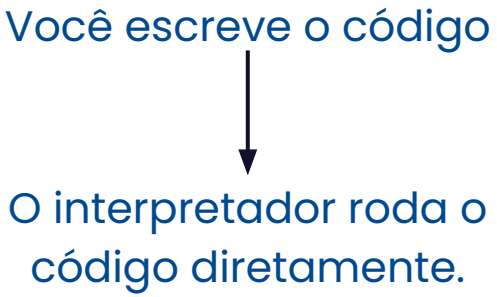


O Interpretador

Traduzir e executar ao mesmo tempo

- **Definição:** Um interpretador lê o código fonte e o executa linha por linha, em tempo real. Ele lê um pouco, traduz e executa.
- **Vantagem:** Desenvolvimento mais rápido e interativo. Se houver um erro, você descobre na hora em que a linha é lida.
- **Exemplos:** Python e JavaScript são linguagens interpretadas.

Processo



O Modelo Híbrido (Bytecode)

O melhor dos dois mundos

- Muitas linguagens modernas usam uma abordagem mista.
- **Bytecode:** O código fonte (ex: .py) é compilado para um código intermediário chamado *Bytecode* (ex: .pyc), que é mais simples que o código fonte, mas não é ainda código de máquina.
- **Máquina Virtual:** Um interpretador lê esse *bytecode* e o executa. Isso ocorre no Java (JVM) e no Python (CPython).
- **Disassembling:** É possível usar ferramentas (como o módulo dis no Python) para ver esse bytecode e entender como o código é processado internamente.

Ambientes de Desenvolvimento IDEs

Onde escrevemos o código?

- **Definição:** Uma IDE (*Integrated Development Environment*) é um software que agrupa as ferramentas essenciais para o desenvolvimento.
- **O Pacote Completo:** Em vez de usar um editor de texto simples (como Bloco de Notas), a IDE integra:
  - Editor de código (com cores e formatação).
  - Compilador ou Interpretador.
  - Depurador (Debugger).
  - Terminal e ferramentas de construção.

Comparando Ferramentas

Editores de Código vs. IDEs Robustas

- **VS Code (Visual Studio Code):**
  - É um editor de código leve, mas que pode se comportar como uma IDE através de extensões.
  - Muito popular para Web (JS, HTML) e Python. Flexível e rápido.
- **IntelliJ / Eclipse:**
  - São IDEs completas e robustas, padrão de mercado para linguagens como Java.
  - Oferecem recursos avançados de refatoração automática e análise profunda do projeto, mas exigem mais memória do computador.

O Poder do “Debugger”

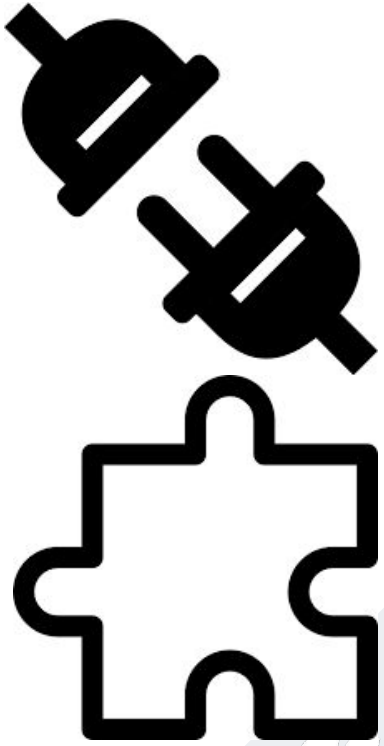
Caçando insetos (Bugs)

- **O que é:** Uma ferramenta que permite pausar a execução do programa e examinar o que está acontecendo "por dentro" naquele exato momento.
- **Breakpoints:** Você marca uma linha de código onde quer que o programa pare.
- **Inspeção:** Quando o programa para, você pode ver o valor das variáveis na memória, ajudando a encontrar erros de lógica ou comportamento inesperado.
- **Passo a passo:** Permite executar o código linha por linha para acompanhar o fluxo de execução.

Plugins e Extensões

Personalizando sua ferramenta

- Nenhuma ferramenta vem pronta para tudo. Usamos plugins para adicionar funcionalidades.
- **Linters:** Plugins que verificam erros de estilo e sintaxe enquanto você digita (ex: *flake8* ou *PyLint* no Python), garantindo que o código siga padrões.
- **Verificação em tempo real:** Muitos editores modernos destacam erros de código diretamente na tela, agindo como corretores ortográficos para programação.



Fontes Utilizadas

- *Downey, A. B. "Pense em Python".*
- *Gamma, E. et al. "Padrões de Projeto: Soluções Reutilizáveis de Software Orientado a Objetos".*
- *Danjou, J. "Python Levado a Sério".*
- *Valente, M. T. "Engenharia de Software Moderna".*


Obrigado!
SENAI