

Programação III

Aula 1 – Histórico e Visão Geral da Linguagem JAVA



A Linhagem do Java

- **Origem:**
 - Java está relacionado ao C++ (um descendente direto do C).
- **Herança de C:**
 - Sintaxe básica.
 - Alguns conceitos de orientação a objetos.
- **Influências de C++:**
 - Características que definem o Java.
 - Respostas a limitações das linguagens anteriores.
- Java nasceu do processo de **refinamento e adaptação** das linguagens ao longo de décadas.
- Cada inovação visava resolver problemas que as linguagens anteriores não conseguiam superar.
- **Java não é exceção:**
 - Projetado para superar limitações anteriores e atender a novas necessidades.



A Linhagem do Java: C

- **Impacto revolucionário:** Mudou fundamentalmente a forma como a programação era abordada.
- **Necessidade inicial:** Substituir o código assembly com uma linguagem eficiente, estruturada e de alto nível.
- **Principais compromissos ao projetar linguagens:**
 - Facilidade de uso x Poder
 - Segurança x Eficiência
 - Rigidez x Extensibilidade
- **Desafios das linguagens anteriores:**
 - **FORTRAN:** Bom para ciência, mas limitado para código de sistemas.
 - **BASIC:** Fácil, mas pouco estruturado.
 - **Assembly:** Eficiente, porém difícil de usar e depurar.
 - **Pascal:** Estruturado, mas ineficiente para sistemas.



A Linhagem do Java: C++



Origem:

Criado por Bjarne Stroustrup em 1979, nos Bell Labs, inicialmente chamado "C with Classes".



Por que C++ foi necessário?

O sucesso do C era evidente, mas programas maiores se tornavam **incontrolavelmente complexos**.



O crescimento da complexidade impulsionou a busca por novas abordagens.

• Limitações do C:

- Dificuldade em lidar com programas extremamente grandes e complexos.
- Estruturas limitadas para organizar programas de maneira compreensível.

• C++ como evolução:

- **Extensão do C:** mantém suas funcionalidades e adiciona novos recursos.
- A abordagem orientada a objetos ajudou a:
 - Organizar melhor os programas.
 - Tornar projetos maiores **mais gerenciáveis**.



Contexto do surgimento

Final dos anos 80 e início dos 90:

- A **programação orientada a objetos** com **C++** se consolidou como uma solução poderosa.
- C++ combinava a **eficiência** do C com a **orientação a objetos**.
- Parecia ser a linguagem ideal para uma ampla gama de programas.

Novas forças surgiram:

- O **crescimento da Internet** e o surgimento da **World Wide Web** começavam a impulsionar uma nova revolução na programação.

O sucesso de C++ não foi o fim

- **Marco** no caminho da evolução das linguagens.
- O surgimento de novas necessidades, impulsionadas pela Internet, exigia uma nova abordagem para a programação.



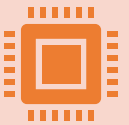
Criação do Java



Origem: Desenvolvido por **James Gosling, Patrick Naughton, Chris Warth, Ed Frank, e Mike Sheridan** na **Sun Microsystems** em 1991.



Primeira versão: Levou 18 meses para ser criada. Originalmente chamada "**Oak**", foi renomeada para "**Java**" em 1995.

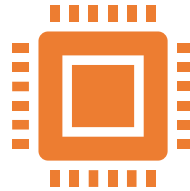


Motivação inicial: Criar uma linguagem **independente de plataforma** para dispositivos eletrônicos, como micro-ondas e controles remotos.

Criação do Java



A Web foi a força motriz que levou Java a seu sucesso global, tornando-se essencial para programação na Internet.



A grande necessidade de portabilidade entre diferentes sistemas e dispositivos foi o que diferenciou Java de C e C++, tornando-o ideal para a nova era da computação distribuída.



Características principais:

**Semelhanças com C e C++:
Multithreading e acesso à Internet.**



Revolução na Programação Web

- Java simplificou a programação para a web e trouxe inovações importantes, como o conceito de **applets**.
 - **Applet:** Programa Java executado automaticamente no navegador ao clicar em um link.
 - Inicialmente, os applets eram usados para **exibir dados**, **processar entradas** ou realizar **funções simples** diretamente no cliente, aliviando o servidor.



Segurança e Portabilidade

- **Segurança:**
 - Java protege programas dinâmicos contra ações maliciosas (vírus, trojans) ao **isolar o código** na **Java Execution Environment (JVM)**, evitando o acesso não autorizado ao sistema.
 - Permitindo que os programas sejam **executados com segurança** no computador do cliente.
- **Portabilidade:**
 - **Execução em Diversos Ambientes:** Java garante que o mesmo código funcione em **diferentes sistemas operacionais, CPUs e navegadores**, sem necessidade de versões separadas.
 - A **JVM** permite que o código seja portátil, mantendo sua funcionalidade em diversos dispositivos.



Bytecode



Portabilidade:

O bytecode é independente da plataforma.

A JVM interpreta e executa o bytecode em qualquer sistema com um **Java Runtime Environment (JRE)**.

Não é necessário recompilar o programa para cada CPU.



Segurança:

JVM cria um ambiente restrito (**sandbox**) que protege o sistema.

Restrições adicionais na linguagem Java reforçam a segurança.

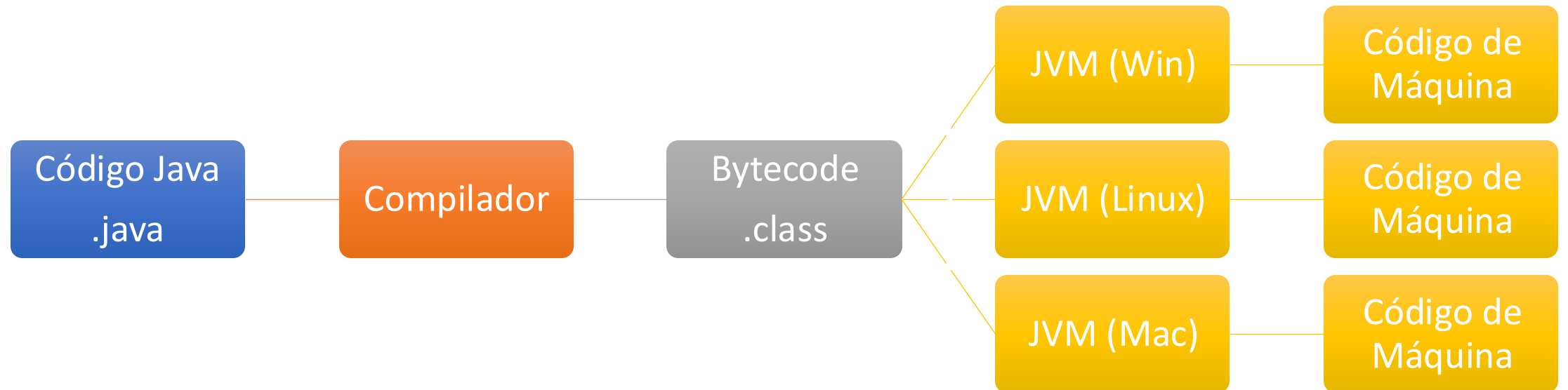


Performance:

Interpretação e Compilação Just-In-Time (JIT):

- A JIT compila partes do bytecode em código executável durante a execução.
- Otimização em tempo real melhora o desempenho sem comprometer a portabilidade ou segurança.

Bytecode



Características Gerais

| | |
|------------------------------|---|
| Simples | Facilmente compreensível e com sintaxe clara. |
| Seguro | Protege contra códigos maliciosos e acessos não autorizados. |
| Portável | Funciona em diversas plataformas sem alterações no código. |
| Orientado a Objetos | Promove modularidade e reutilização. |
| Robusto | Evita falhas com recursos avançados de tratamento de erros. |
| Multithreaded | Suporte integrado para execução de múltiplas tarefas simultaneamente. |
| Neutro em Arquitetura | Projetado para ser independente de hardware. |
| Interpretado | Executado pela JVM, permitindo portabilidade e segurança. |
| Alto Desempenho | Otimizado com técnicas como JIT (Just-In-Time). |
| Distribuído | Ideal para aplicações em redes e sistemas conectados. |
| Dinâmico | Capaz de se adaptar a ambientes em constante mudança. |