

# Fundamentos de Java

Introdução ao Java



Softblue  
cursos online

---

---

---


---

---

---

---

Tópicos Abordados



- História do Java
- JVM (Java Virtual Machine)
  - Vantagens da máquina virtual
- Bytecode
- Versões
- Ramificações
- Onde usar Java e onde não é necessário usar Java
- Nomenclatura: JVM, JRE e JDK

---

---

---


---

---

---

---

Como Tudo Começou



- Busca de inovações tecnológicas pela Sun (1992)
- Time liderado por James Gosling
- Interpretador para trabalhar com diversos aparelhos eletrônicos (como videocassete, TV e aparelhos de TV a cabo)
- A ideia não funcionou

---

---

---

---

---

---

---

## O Java e a Internet



- Com o aparecimento da Internet, a Sun resolveu usar sua ideia para rodar aplicações dentro dos browsers
- Grande quantidade de sistemas operacionais e browsers
- Na época os browsers só renderizavam HTML
- Criação das applets

---

---

---

---

---

---

---

## No Final das Contas...



- Java nasceu com um objetivo
- Foi lançado com outro objetivo
- Firmou-se como tecnologia de desenvolvimento de aplicações no lado servidor
- Em 2009 a Oracle comprou a Sun Microsystems e passou a cuidar do desenvolvimento da linguagem Java

---

---

---

---

---

---

---

## Logotipo e Mascote



---

---

---

---

---

---

---

## Java e a JVM



- JVM
  - Java Virtual Machine
- A Máquina Virtual é uma camada intermediária entre o sistema operacional e a aplicação
- A aplicação se comunica apenas com a JVM

---

---

---

---

---

---

---

## Quando Não Existe Máquina Virtual



- Numa linguagem como C ou Pascal, o código é compilado para uma plataforma específica
- Toda vez que a plataforma muda é necessário recompilar o código
- Às vezes é necessário até alterar o código
  - Interfaces gráficas Windows X Linux
- A aplicação fica atrelada à plataforma sobre a qual ela executa

---

---

---

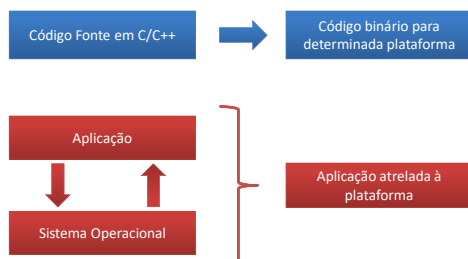
---

---

---

---

## Aplicações Atreladas à Plataforma



---

---

---

---

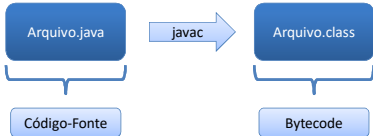
---

---

---

## Bytecode

- O bytecode é uma linguagem entendida pela JVM
- A geração do bytecode é feita através da compilação do código Java



```

graph LR
    A[Arquivo.java] -- javac --> B[Arquivo.class]
    subgraph C [Código-Fonte]
        A
    end
    subgraph D [Bytecode]
        B
    end
  
```

---

---

---

---

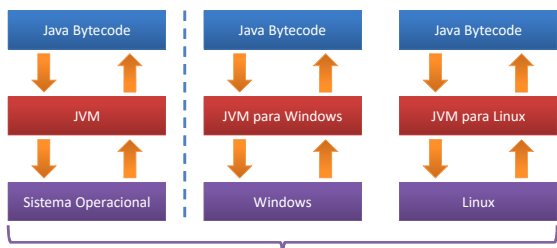
---

---

---

---

## Mesmo Código entre Plataformas



```

graph TD
    subgraph SO [Sistema Operacional]
        JB1[Java Bytecode]
        J1[JVM]
        SO1[Sistema Operacional]
        JB1 --> J1
        J1 --> SO1
        SO1 --> JB1
    end
    subgraph W [Windows]
        JB2[Java Bytecode]
        J2[JVM para Windows]
        W1[Windows]
        JB2 --> J2
        J2 --> W1
        W1 --> JB2
    end
    subgraph L [Linux]
        JB3[Java Bytecode]
        J3[JVM para Linux]
        L1[Linux]
        JB3 --> J3
        J3 --> L1
        L1 --> JB3
    end
  
```

Ao mudar a plataforma, o bytecode não precisa ser alterado

---

---

---

---

---

---

---

---

## Vantagens da Máquina Virtual

- Isolamento total da aplicação
- Como tudo passa pela JVM, é possível obter métricas e trabalhar com otimização
- Garbage collection
- Princípio WORA
  - “Write Once, Run Anywhere”

---

---

---

---

---

---

---

---

## A JVM é uma Especificação



- Diversas empresas implementam a JVM
  - Oracle, IBM, etc.
- É possível trocar de JVM sem a necessidade de recompilar os códigos das aplicações

---

---

---

---

---

---

---

## A Performance do Java



- A JVM usa dois elementos para otimizar a performance
  - HotSpot
    - Identifica código bastante executado
  - JIT (Just in Time Compiler)
    - Compila o código identificado pelo HotSpot para instruções nativas da plataforma
- Mito da performance
  - “Java é uma linguagem com baixa performance”
  - Existem estudos que provam que o Java pode ser mais rápido em algumas situações do que C ou C++

---

---

---

---

---

---

---

## Ramificações do Java



- **Java SE** (Standard Edition)
  - Base do Java
  - Ambiente de execução e bibliotecas comuns
- **Java EE** (Enterprise Edition)
  - Aplicações corporativas e internet
- **Java ME** (Micro Edition)
  - Dispositivos móveis

---

---

---

---

---

---

---

### Onde Usar Java



- Aplicações de médio e grande porte
- Aplicações que virão a se expandir no futuro
- Ambientes heterogêneos

---

---

---

---

---

---

---

### Onde Não é Necessário Usar Java



- Para pequenas aplicações com poucos desenvolvedores, outras linguagens podem ser mais produtivas

---

---

---

---

---

---

---

### Por que Adotar o Java



- Linguagem madura
- Extensa lista de bibliotecas gratuitas
  - Manipulação de XML
  - Logging
  - Manipuladores de texto
  - Geração de relatórios
  - etc.
- Grande comunidade de desenvolvedores
- Documentação farta

---

---

---

---

---

---

---

### Nomenclaturas: JVM



- Java Virtual Machine
- Máquina virtual do Java
- Instalar apenas a JVM não é possível

---

---

---

---

---

---

---

### Nomenclaturas: JRE



- Java Runtime Environment
- JVM + bibliotecas
- Utilizada para executar aplicações Java

---

---

---

---

---

---

---

### Nomenclaturas: JDK



- Java Development Kit
- JRE + kit completo de desenvolvimento
- Utilizado pelos desenvolvedores para compilar e executar aplicações Java

---

---

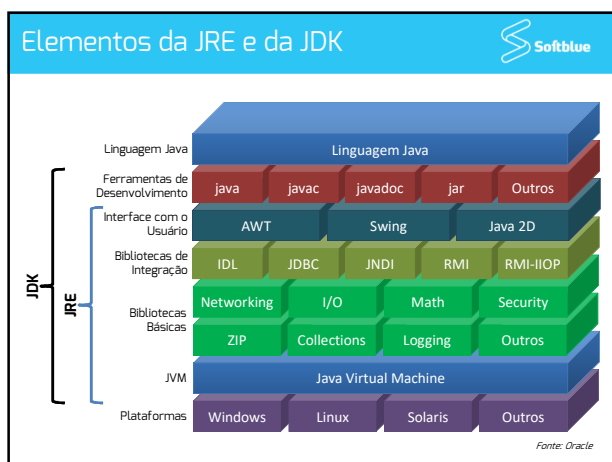
---

---

---

---

---




---

---

---

---

---

---

---

---




---

---

---

---

---

---

---

---