

Multithreading: Java Suporta programação multithreading, permitindo criar programas que executam  
Vários tasks Simultaneamente

Orchestrator-Neutral: Java foi projetado para garantir Longuidade e portabilidade do código

Interpretado e alto desempenho: Java compila para Bytecode, que pode ser executado em qualquer  
Sistema que implemente a JVM.

Distribuído: Java é projetado para o ambiente distribuído de Internet

Dinâmico: Carrega informações de tipo em tempo de execução para verificar e ajustar o acesso a objetos

Desde o lançamento oficial, Java evoluiu rapidamente, diferente de outros sistemas de Software que  
geralmente apresentam melhorias pequenas e incrementais

Java 1.1: adicionou novos elementos de biblioteca, modelou o tratamento de eventos e reconfigurou  
recursos da biblioteca original do Java 1.0.

Java 2: Incluiu Suporte para Swing, Collections framework, melhorias na JVM

Java 2 SE 1: melhorou a funcionalidade existente e o ambiente de desenvolvimento

J2SE 1.4: Introduziu melhorias importantes, como o novo Keyword assert, exceções  
encadeadas e um subsistema de I/O baseado em canais

J2SE 5: Adicionou recursos como o generics, annotations, autoboxing, enumeração, lambda  
each, for-each, import estático, i/o formatado e utilitários de concorrência

J2SE 6: Adicionou melhorias incrementais nas bibliotecas API e melhorias no  
tempo de execução

Java SE 7: trouxe Novas funcionalidades para a linguagem e as bibliotecas API.

Java SE 8: Introduziu expressões lambda, aprimorou a biblioteca API, implementação  
padrão de métodos em interfaces e API de data e hora

Java SE 9: Apresentou módulos, adicionou novos elementos de sintaxe, incluiu  
a funcionalidade JLink, introduziu o JShell, métodos privados em interfaces e catalog OJindex.  
API de Applets foi descontinuada

Java SE 10: trouxe inferência de tipo para variáveis locais, utilizando var

Java SE 11: Adicionou Suporte para var em expressões lambda, API de cliente HTTP,  
Capacidade de executar programas de arquivo único diretamente.

Entre o JDK 11 LTS e o próximo LTS, houve 5 incrementos de recursos, JDK 12 a 16. JDK 12  
e 13 Não adicionam Nada, JDK 14 introduziu a expressão switch, JDK 15 introduziu Placard de texto, JDK 16  
adicionou correspondência de padrões ao instância e a nova palavra-chave record e o Jpackage

Java SE 17 introduziu a capacidade de Selar classes e interfaces, controlando o acesso  
a implementação com as palavras-chave Sealed, permits e Non-Sealed.



Programas dinâmicos apresentam problemas de Segurança e portabilidade. Java resolveu esses problemas ao permitir a execução segura de programas no ambiente de execução Java, prevenindo acesso não autorizados.

A portabilidade é crucial na Internet devido a diversidade de computadores e sistemas operacionais. Java criou um mecanismo para que o mesmo código de aplicação pudesse ser executado em diferentes sistemas, garantindo a portabilidade.

Java Solucionou os problemas de Segurança e portabilidade usando Bytecode, um conjunto otimizado de instruções executado pela Java Virtual Machine (JVM). A JVM interpreta o Bytecode, tornando os programas Java portáveis e seguros, pois JVM cria um ambiente de execução restrito, evitando o acesso irrestrito ao sistema.

Traduzir programas Java para Bytecode facilita a execução de diferentes plataformas, pois apenas a JVM precisa ser implementada para cada plataforma. A tecnologia Hot Spot introduziu o compilador Just-in-time (JIT), que converte partes de Bytecode em código executável em tempo real, aumentando o desempenho sem comprometer a portabilidade e Segurança.

Desde o lançamento oficial do Java, muitas mudanças ocorreram. No início, a Internet era uma novidade, e os Navegadores estavam em rápido desenvolvimento. Os applets foram essenciais para demonstrar o poder do Java, mas seu suporte diminuiu ao longo do tempo, sendo descontinuado a partir do JDR 11.

Java Web Start era uma alternativa aos applets, permitindo a execução de aplicativos Java fora do Navegador, mas também foi descontinuado no JDR 11.

Uma grande mudança no Java envolve a programação de versões. Anteriormente, as grandes versões eram separadas por dois anos ou mais. Após o JDR 9, o intervalo foi reduzido para 6 meses.

A cada 3 anos, ocorre uma versão de suporte a longo prazo, que tem suporte estendido o primeiro foi o JDR 11, seguido pelo JDR 17. Atualmente, os desenvolvedores são programados de março a setembro de cada ano.

Após o lançamento oficial do Java, ficou evidente que Java também seria útil no lado do Servidor, resultando na criação dos servlets, que é um pequeno programa que executa no servidor para gerar conteúdo dinâmico.

Servlets, compilados em Bytecode e executados pela JVM, são altamente portáteis, podendo Ser usados em diferentes ambientes de Servidor. A única exigência é que o Servidor suporte a JVM e um container de Servlets.

Sistemas que moldam o Java incluem: Simples, Seguro, Sua portabilidade, orientado a objetos, entre outros.

Isso mostra como o Java foi projetado para Ser Simples, eficiente, flexível e adequado para vários aplicativos e ambientes.

Simples: Java foi projetado para Ser fácil de aprender e usar por programadores profissionais, criando o objeto. Java adotou um modelo de objeto Simples e fácil de entender.

Robusto: Java foi projetado para criar programas confiáveis, que funcionem em diversos sistemas.



Nos anos 80 e 90, o C++ e a POO se tornaram populares, criando uma sensação de que havia uma linguagem ideal. No entanto, Novas forças estavam moldando a evolução das linguagens de programação.

Então nos anos 90 foi criado o Java, originalmente concebido como Oak, ele foi criado para ser uma linguagem independente de plataforma, que poderia ser usada para criar softwares em dispositivos eletrônicos. A grande limitação do C e C++ era de criar programas para diferentes tipos de CPU, por isso necessitava compilar cada programa especificamente para cada plataforma. e esse foi o ponto de partida para desenvolver o Java.

Os applets Java foram uma das principais aplicações práticas dessa, permitindo que desenvolvedores incorporassem funcionalidades interativas diretamente nas páginas da Web. Embora as applets tenham caído em desuso, o impacto do Java foi duradouro, influenciando o desenvolvimento de tecnologias.

Nos anos 90, a equipe de design do Java percebeu que as programações, problemas de portabilidade na criação de código para controladores também se aplicavam na internet. Assim, o foco do Java mudou de eletrônicos de consumo para a programação na internet, o que contribuiu para o sucesso.

Java foi influenciado por C e C++, adotando a sintaxe parecida e características orientadas a objetos dessas linguagens para atrair programadores experientes. Java se destaca por ser projetado, testado por programadores reais, sendo bem consistente, oferecendo muito controle ao programador.

Seguindo <sup>embora</sup> como Java compartilha Semelhanças com C++, ele possui diferenças práticas significativas. Java AEO é uma versão aprimorada do C++, não é compatível com ele. Em vez disso, foi projetado para resolver um conjunto específico de problemas ~~dest~~ que o C++ não resolve.

As linguagens de programação evoluem para se adaptar as mudanças no ambiente e implementar avanços na programação. O Java surgiu da necessidade de programas independentes de plataforma para a distribuição na internet. Além disso, o Java melhorou o paradigma orientado a objetos do C++, adicionou suporte integrado para multithreading e forneceu uma biblioteca que tornou mais fácil o acesso à internet.

O sucesso do Java influenciou o desenvolvimento de novas linguagens, sendo o C# um exemplo bom. Criado pela Microsoft para o .Net framework, o C# compartilha a mesma sintaxe geral, suporte a programação e modelo de objetos que o Java.

O Java teve um efeito profundo na internet. Além de simplificar a programação web, o Java inovou com o applet, um tipo de programa que pode ser transmitido pela internet e executado dentro de um navegador compatível com o Java. Os applets permitiram mover parte da funcionalidade do servidor para o cliente e eram usados para exibir dados, lidar com o usuário e funções locais.

A criação dos applets foi importante porque expandiu o universo de objetos que podiam ser manipulados no ciberespaço. Diferente dos pacotes passivos, como e-mails, os applets são programas dinâmicos e autônomos, que adicionaram uma nova dimensão emocionante às páginas da web e ilustraram os benefícios do Java. Com o tempo, os applets se tornaram menos importantes, sendo seu suporte removido a partir do MK91.



Renato Portillo Costa RA 24101301

### Resumo: the history an evolution of java

Java tem uma forte ligação com C e C++, herdando a Sintaxe de C e muitos conceitos de orientação a objeto do C++. Seu desenvolvimento foi parte de um longo processo de "refinamento" das linguagens de programação, surgindo para resolver problemas que suas antecessoras não conseguiam superar.

O C revolucionou o mundo da programação, mudando a forma que os programas eram pensados e trabalhavam. Antes de sua criação, as linguagens de programação apresentavam grandes limitações, como a necessidade de escolher entre poder e facilidade de uso, ou entre segurança e eficiência. Linguagens como Fortran, Basic, cobol eram eficientes em algumas áreas, mas faltavam em outras, como na criação de códigos mais complexos e estruturados. A principal limitação era a falta de estrutura, o que resultava em "código espaguete", difícil de entender.

Nas décadas de 1970 a demanda por software estava cada vez maior e ao mesmo tempo, o custo dos computadores estava se expandindo, permitindo mais liberdade para experimentos e criar novos paradigmas. E foi aí que o C nasceu. Criado por Dennis Ritchie para o sistema UNIX baseado em Bcpl.

Por muitos anos, o padrão de fato para a linguagem C foi fornecido com o sistema operacional UNIX. Em 1983 a linguagem foi formalmente padronizada pelo ANSI. C é considerada por muitos como o início da era moderna das linguagens de programação. Ela combinou bem as habilidades conflitantes que as linguagens anteriores tinham dificuldade, resultando na melhor linguagem da época.

Porém o C++ surgiu nos anos 70 e 80 como uma evolução do C, devido a crescente complexidade dos programas. Nas primeiras décadas de programação, a era foi predominantemente com estruturas de dados, mas com o tempo, a programação de baixo nível como o Assembly e linguagens de alto nível, como o Fortran, mas com o aumento na complexidade de projetos, surgiu a programação orientada a objetos (POO), permitindo os programadores escrever programas de maior complexidade.

Porém que C seja uma das grandes linguagens de programação, ela tem um limite de complexidade. À medida que um programa cresce, ele se torna mais difícil de entender e gerenciar. O C++ foi criado para superar esse limite, permitindo que os programadores lidassem com programas maiores.

O C++ foi criado em 1983 por Bjarne Stroustrup, O C++ se trata o C, mantendo todas as suas características e funcionalidades, o que fez ter muito sucesso.