

História e evolução da Java

Para entender Java, é essencial compreender as razões de sua criação, sua evolução e sua herança. Assim como outras linguagens Java possui muitos elementos de suas predecessoras. O desenvolvimento de Java se deu por a necessidade de se adaptar a novos ambientes e usos, e melhorias na arte da programação, foi derivada de C++, ela herdou a sintaxe de C e recursos orientados a objetos de C++, foi feita para resolver problemas que precisavam de grandes quantidades de códigos.

A linguagem C revolucionou a programação ao oferecer uma alternativa estruturada, eficiente de alto nível para a criação de sistemas. Antes de C, era preciso escolher entre facilidade de uso vs poder, segurança vs eficiência e rigidez vs extensibilidade, linguagens apresentavam limitações, Assembly era poderoso, mas difícil de aprender. Por isso em 1970, a demanda era maior que a quantidade de programadores, e os computadores ficavam mais acessíveis de modo que eles próprios criavam o C.

A evolução da linguagem C para C++, criada para substituir a linguagem de montagem, C tornou-se um padrão na programação, combinando eficiência, estrutura e facilidade de uso. Em 1989, foi padronizado pela ANSI, linguagem feita de programadores para programadores. No início de 1980, com programas cada vez mais complexos, surgiu C++, que trouxe a programação orientada a objetos (OOP), feita para agrupar dados, reutilizar código, adaptar novos usos e necessidades.

A evolução de C para Java se deu pela necessidade de gerenciar programas mais complexos, para resolver isso, Bjarne Stroustrup criou o C++ em 1979, inicialmente chamado de "C com classe", mas C, mas com COOP integrada, já nos anos de 1990, a popularidade da internet pediu mudanças, a Sun Microsystems criou o Java, com o objetivo de criar um sistema livre de plataforma para dispositivos eletrônicos, feita para funcionar em qualquer sistema, ainda hoje e dia. Java surgiu para criar programas que funcionassem em qualquer CPU, em 1993, os criadores do Java perceberam que as demandas enfrentadas na portabilidade em dispositivos era as mesmas da internet, isso mudou o foco do projeto para a web, sendo essencial para o desenvolvimento online. Com isso, comparando a sintaxe de C e C++, ela foi feita para ser mais segura, portátil e eficiente. Seu sucesso veio da necessidade de aplicativos de multiprocessamento, sendo um dos mais importantes atualmente.

O Java não foi criado para substituir o C++, mas sim para resolver problemas diferentes: principalmente programas portáveis na internet. Sua evolução não ocorreu devido pela demanda por portáveis multiprocessamento, mas também para suportar nativo a multithreading e uma biblioteca simplificada, o Java inspirou novas linguagens, como o C#, criado pela Microsoft para o .Net Framework. Ambas compartilham sintaxes similares, isso mostra que o Java redifiniu as linguagens.

Uma das evoluções do Java foram os applets, pequenos programas que rodam diretamente no navegador ou mesmo baixados na internet. A chegada da web e melhorou as coisas e a serem mais interativas, de seu para diminuir o tempo e reduzir a carga dos servidores.

O Applet de segurança tem um papel importante no início do Java, pois possibilita a transição de programas feitos pela internet, diferentes de dados passivos, como e-mails e arquivos baixados. Ele permite que um código possa executar e executar no navegador do usuário.

Sempre houve riscos, pois códigos maliciosos poderiam roubar dados. Java resolveu esse problema por meio de Sandbox, onde o código baixado é executado em ambientes separados, sem acesso ao sistema operacional. Diferente de outras linguagens o Java transforma seu código Traduzido para Bytecode, que pode ser executado em qualquer dispositivo que tenha JVM instalado.

Diferente de outras linguagens o Java compila seu código em Bytecode, onde é executado e interpretado pela Java Virtual Machine (JVM), se o dispositivo tiver instalado pode executar o programa. O JVM cria um ambiente chamado sandbox, que impede que o programa tenha acesso irrestrito ao sistema operacional. A utilização do Bytecode é mais lenta que o código nativo, porém, o JIT, traz o conceito de Just-in-Time (JIT) Compilation.

O JIT converte parte do Bytecode em código nativo agilizando a execução.

Ahead-of-Time (AOT): Uma abordagem que compila Bytecode em código nativo antes da execução, foi introduzida no JDK 17 e removeu os applets AOT, evoluindo por meio dos parâmetros `-Xlink` e `-Xpackage`:

`-Xlink (JDK 9)`, cria imagem personalizada do JRE,
`-Xpackage (JDK 16)`, cria pacotes nativos instaláveis.

Java Buzzwords são característicos - chaves que definem a linguagem. Além de segurança e portabilidade, outras qualidades incluem: Java foi projetado para ser fácil de aprender e usar. Programadores com experiência em C e C++ terão uma transição fácil devido à sintaxe semelhante. Java não foi feito para ser compatível com outras linguagens, permitindo uma abordagem única e pragmática, a natureza multiplataforma da JVM exige que o Java seja compatível em diferentes plataformas.

Diferenças de C e C++, que devem operar naturalmente a localização e deslocalização de memória. Java faz eficientemente por meio de garbage collection, reduzindo falhas, e também diminui os erros relacionados ao tempo de execução por meio de (OOP), permitindo que o tempo seja gerenciado, Multithreading: permite a criação de programas que executam várias tarefas ao mesmo tempo, faz isso de forma fluida. A versão Java 1.1 trouxe novas melhorias como adição de novas bibliotecas, foi a versão Java 2 (J2SE), trouxe bibliotecas gráficas, Java Swing e o Collections Framework, além de melhorias na Java Virtual Machine, o Java é uma linguagem que sempre está em evolução, a versão do (Java 5), trouxe novas melhorias como: Parametrização de tipos, anotação para o compilador, conversão automática de tipos primitivos, enumerações, novo estilo de loop, já a versão (Java 6), trouxe aprimoramentos das bibliotecas e otimização no tempo de execução.

O primeiro lançamento foi o Java 2 que trouxe atualizações para o sistema, linguagem não-Java, foi lançada pela Oracle, foram desenvolvidas pequenas melhorias no "projeto Coih", pequenas alterações que tiveram grande impacto como: String e de declaração Switch, suporte para literais binários, expansão de "Try", que gerencia exceções, recursos como Stream, adição de Fork/Join Framework, importante para programação paralela, versão (Java 8), trouxe a expressão lambda, trazendo recursos de programação funcional e simplicidade no código, versão (Java 9) introduziu o conceito de módulos, que organiza o código em módulos e o framework Jlink, que serve para criar imagens de tempo de execução, Também trouxe o JShell, ferramenta para a experimentação interativa e aprendizado de Java, fim da atualização dos applets, a versão "Java 10", trouxe a inferência de variáveis locais, permitindo a simplificação de código com menos no valor atribuído, Java 11, suporte "var" para expressões lambda e recursos dos suporte applets, a versão 12 a 13 não teve alterações significativas, versão Java 14 trouxe expressão Switch, versão 15 trouxe blocos de texto, por fim a versão 17 trouxe classes e interfaces abstratas e o Open-Public do JDK, assim se encerra a história do Java.