



Curso de Java

aula 01

Prof. Anderson Henrique



Apresentação da Linguagem Java

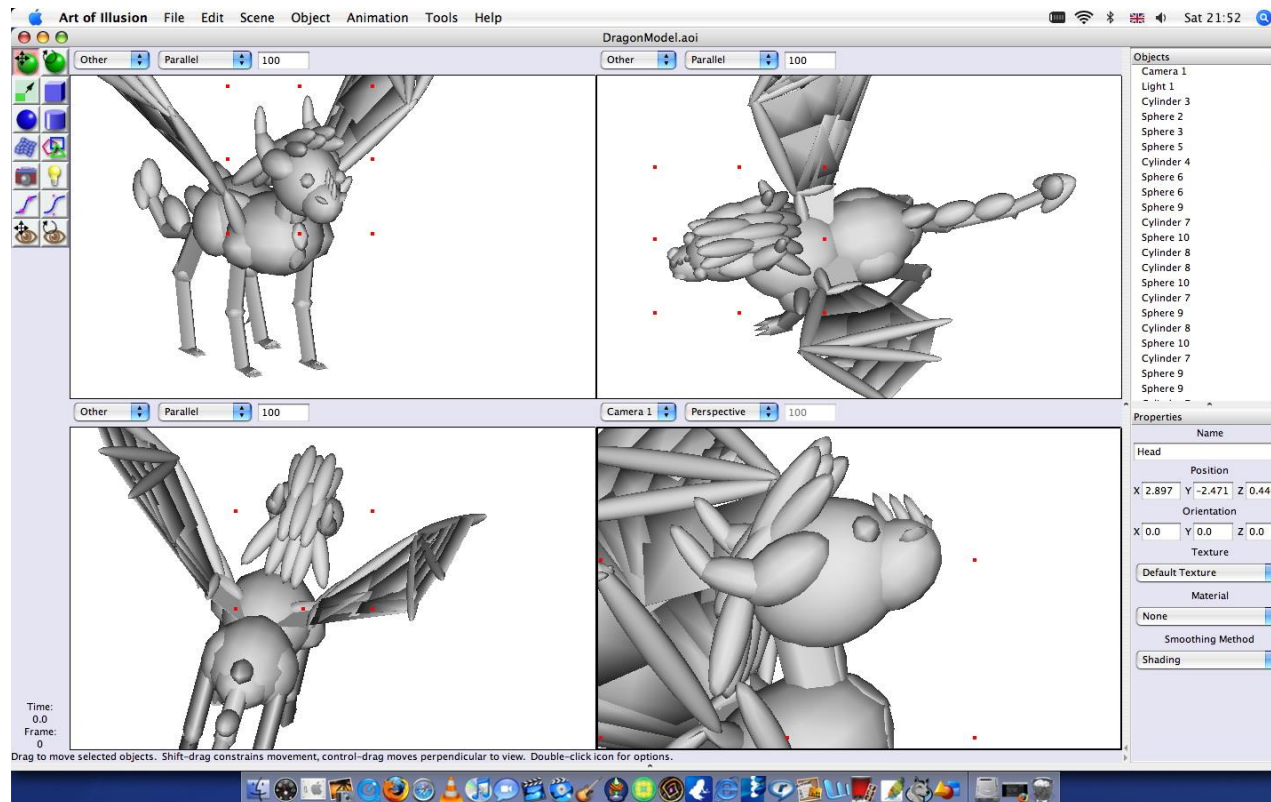
Java é uma linguagem de programação multiplataforma. Lançada em maio de 1995 foi inicialmente considerada por muitos uma linguagem para uso na internet, graças ao fato de ter surgido em um momento em que a internet começava a se propagar mundialmente (e também por causa do seu vínculo com os navegadores HotJava e Netscape).

Mas ela é mais do que isso...



É uma linguagem para o desenvolvimento de sistemas e aplicativos em geral, como programas para escritório, automação comercial, sistemas de gestão e até mesmo jogos.

Só para se ter ideia do poder dessa linguagem, existe um software de modelagem 3D totalmente escrito em Java, disponível gratuitamente para download na internet, denominado **Art of Illusion**.



Java também pode ser considerada uma plataforma completa de desenvolvimento de aplicações. Ela é composta por três versões que agrupam recursos específicos para a construção de diferentes tipos de aplicações.

Versão	Descrição
Standard Edition (jse)	Usada principalmente no desenvolvimento de aplicativos para ambiente desktop, é formada por compilador, ambiente de execução e bibliotecas de classe.
Enterprise Edition (jee)	Possui recursos para o desenvolvimento da aplicação que devem ser executadas em ambiente corporativo ou na web.
Micro Edition (jme)	Utilizada no desenvolvimento de aplicações para dispositivos móveis, como tablets e telefones celulares.

Java SE Downloads



Java Platform (JDK) 8u111 / 8u11

Java™ EE 8 SDK Downloads



Java EE 8 Platform SDK



ORACLE JAVA FOR MOBILE AND EMBEDDED DEVICES - DOWNLOADS

Java for Mobile Devices is a set of technologies that let developers deliver applications and services to all types of mobile handsets, ranging from price efficient feature-phones to the latest smartphones.



Java ME SDK



Java ME Embedded



LWUIT



TECH
soluções em ti

Java é totalmente orientada a objetos e teve seu desenvolvimento baseado nas linguagens C++ e SmallTalk. Embora sua estrutura tenha sido influenciada pela linguagem C++, isso não significa que Java seja uma variação dessa linguagem.

A equipe de desenvolvimento liderada por James Gosling responsável pela sua criação uniu os dois mundos (C++ e SmallTalk) e criou uma linguagem nova, embora empregue conceitos utilizados em ambas.



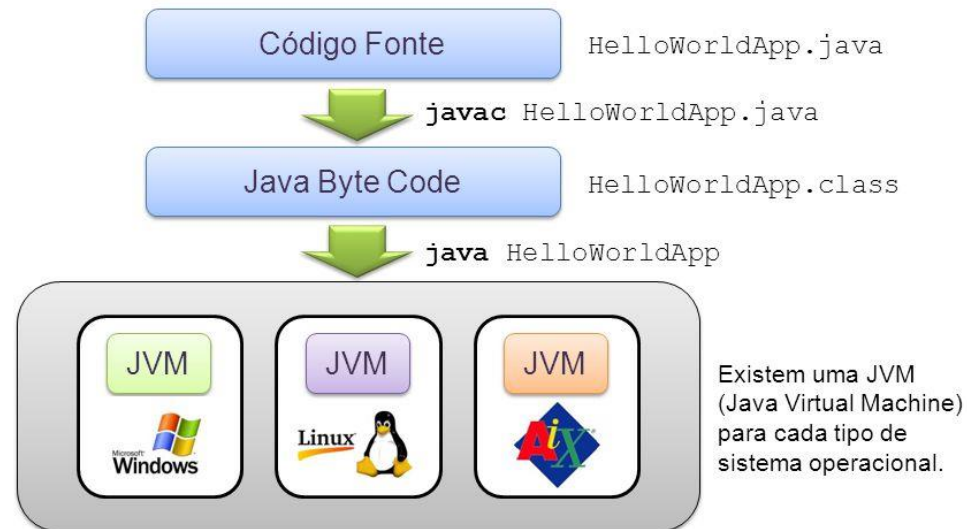
Linguagem relativamente fácil de aprender, principalmente para quem já tem algum conhecimento em C/C++. Entre algumas de suas principais características, podemos citar:

- ☐ Suporte a multiprocessamento;
- ☐ Existência de um coletor de lixo (**Garbage Collector**), que possibilita uso otimizado da memória;
- ☐ Linguagem altamente tipada;
- ☐ Recursos sofisticados para produção de interfaces gráficas;
- ☐ Suporte ao desenvolvimento de aplicações para redes e processamento distribuído;
- ☐ Alta performance;
- ☐ Robustez;
- ☐ Segurança.

Como funciona?

Para executar um programa escrito em Java, precisamos de dois passos. No primeiro, efetuamos a compilação do código-fonte e, no segundo, o código pseudocompilado é rodado pela **Máquina Virtual Java** (JVM – *Java Virtual Machine*), que é específica para os mais diversos tipos de processadores.

Como o Java é executado



O termo pseudocompilado significa que, em Java, não é gerado um programa executável autossuficiente, também conhecido como código nativo de máquina, mas um arquivo intermediário, denominado *bytecode*, que contém instruções que serão interpretadas pela máquina virtual.

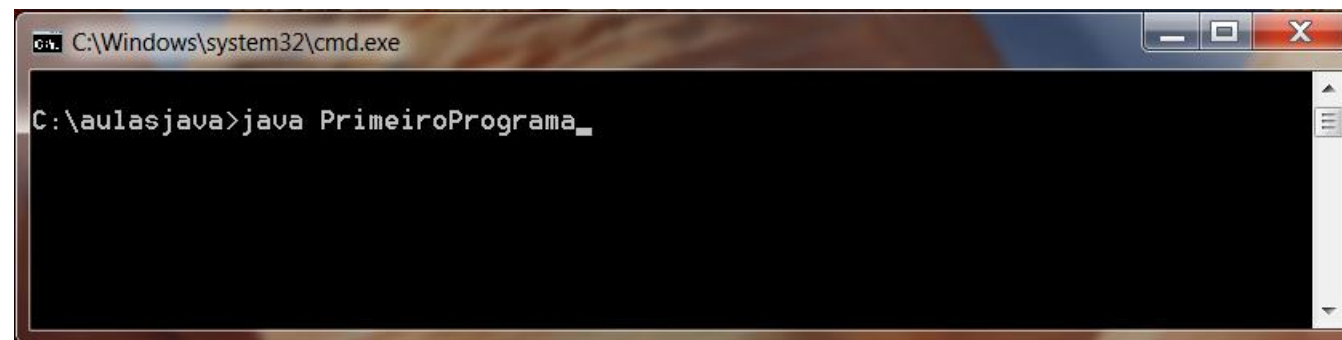


Para executar o código *bytecode*, é preciso utilizar o “lançador” Java. Isso é feito digitando-se o comando **java** na janela de comando (cmd) ou no console terminal do Linux, seguido do nome do arquivo que tem a extensão **.class**



```
C:\Windows\system32\cmd.exe  
C:\aulasjava>javac PrimeiroPrograma.java
```

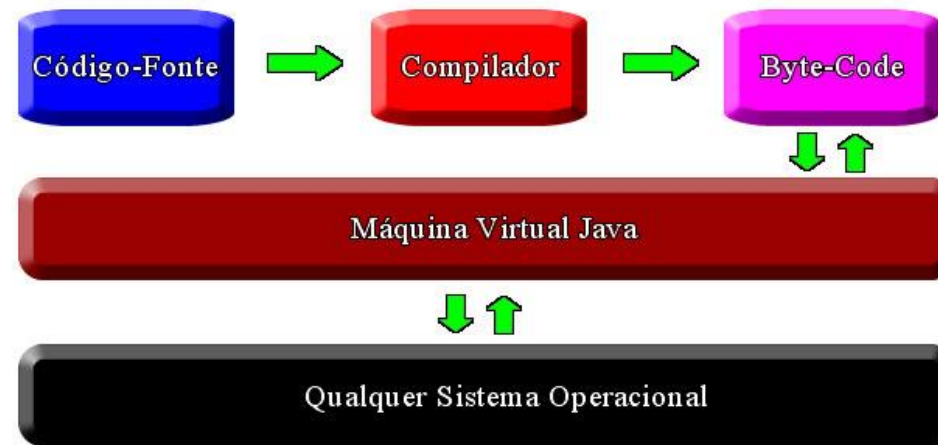
Exemplo de compilação de um código-fonte no CMD



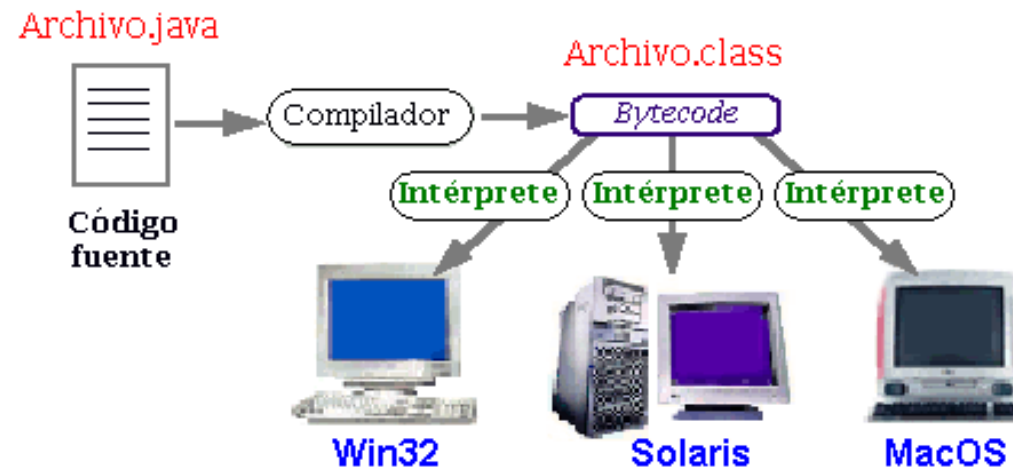
```
C:\Windows\system32\cmd.exe  
C:\aulasjava>java PrimeiroPrograma_
```

Exemplo de execução de um bytecode no CMD

Esse “lançador” faz parte do JRE (**Java Runtime Environment**) e existe uma versão específica para cada plataforma (Windows, MacOS, Linux, Android).



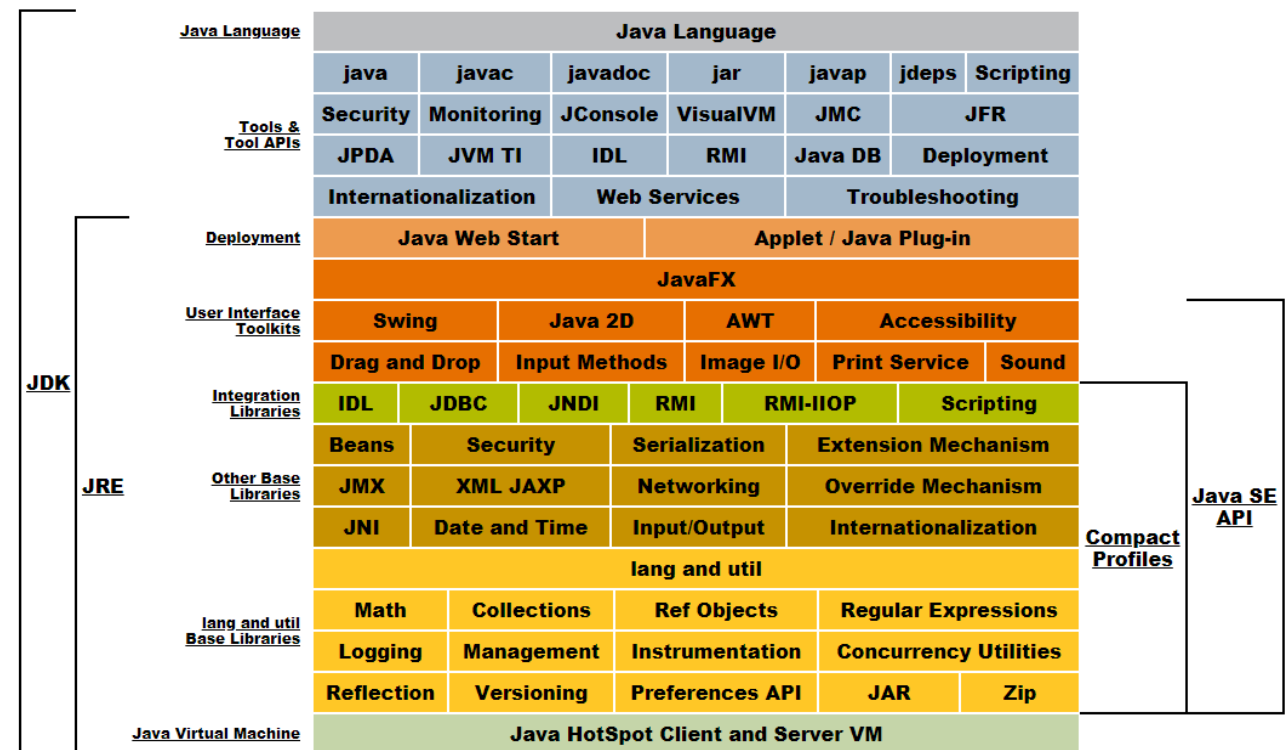
É possível perceber que precisamos criar a aplicação uma única vez, pois o mesmo código *bytecode* é interpretado por diferentes ambientes de forma transparente ao usuário.



“Escreva uma vez, rode em qualquer lugar”, esta é a característica da portabilidade, rodar o programa em diversas plataformas.

Muitos profissionais consideram Java não apenas uma linguagem de programação, mas também uma tecnologia completa, haja vista que o conjunto todo não é formado simplesmente por um compilador, mas por um ambiente completo de desenvolvimento e execução. Podemos considerá-lo uma verdadeira plataforma.

Mais detalhes, ver a documentação do Java...

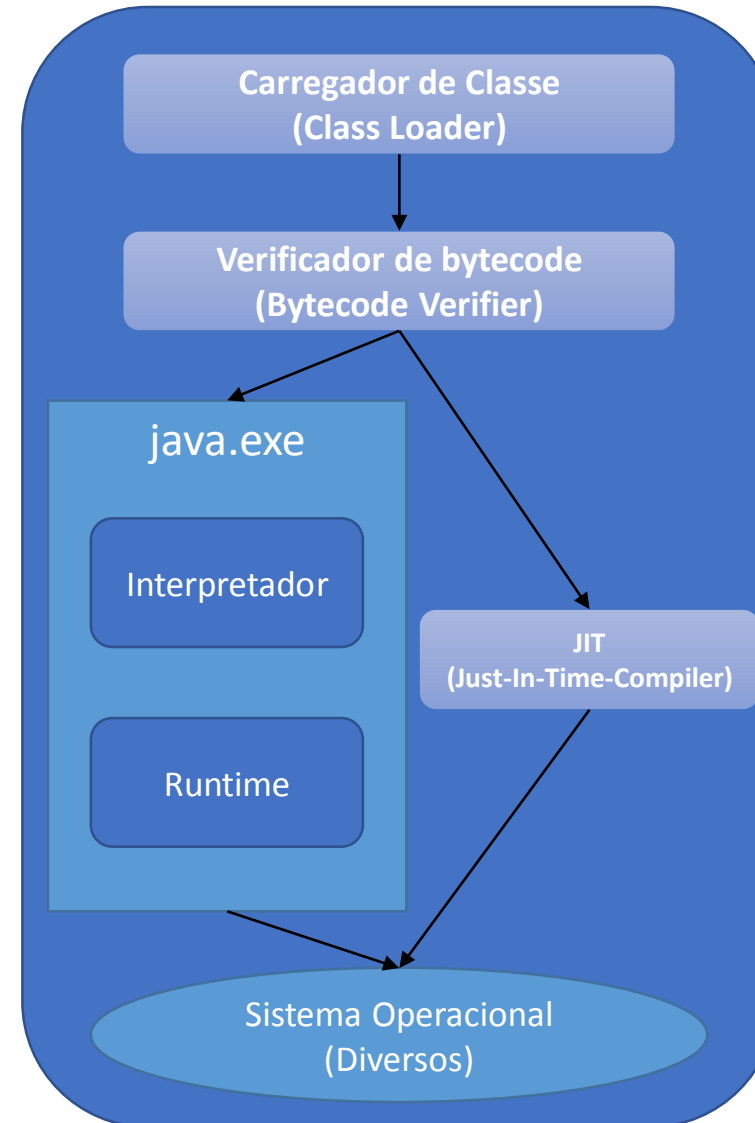


Uma das partes que compõem essa plataforma é o ambiente de execução **JRE**. Trata-se da camada responsável por interpretar os *bytecode* durante a execução do programa.

O **JRE** é formado por diversos componentes, e o principal talvez seja o interpretador, que pode ser identificado pelo arquivo **java.exe**. É ele que deve ser rodado junto com nossas aplicações.

Outros componentes importantes são o carregador de classes (**loader**), responsável pelo carregamento das classes Java que fazem parte do aplicativo, e o verificador de *bytecode* (**verifier**), cuja função é certificar-se de que as classes carregadas são realmente classes Java.

É no JRE que se encontra o coletor de lixo, um mecanismo inteligente responsável pela remoção de objetos e dados da memória quando eles não são mais utilizados pelo aplicativo. Este talvez seja o recurso que mais distingue Java da C++.



Devido a essa forma de execução de programas, seu código nunca acessa diretamente o sistema operacional ou os recursos de hardware, como portas de entrada e saída, memória e periféricos. A única coisa que ele consegue enxergar é a JVM. Essa sim, efetua chamadas ao sistema e também determina o que pode ou não ser feito pelos programas. Isso garante um alto nível de segurança.



Tipos de aplicativo

A versatilidade da linguagem Java é tamanha que ela possibilita a criação de aplicativos dos mais variados tipos e para diferentes ambientes, como equipamentos móveis.

Tipo de aplicativo	Funcionalidades e uso
Autônomo (stand-alone)	Aplicativos que rodam isoladamente ou a partir de um servidor. Na maioria das vezes são aplicações de gestão ou de automação comercial.
Applets	Pequenos programas que podem rodar a partir de um navegador web que suporta a J2SE. Como exemplos podemos criar jogos on-line, animações de sites e teclados virtuais (muito utilizados em internet banking).
Servlets	Aplicações executadas em servidores web por intermédio de navegadores. O uso mais comum ocorre em sistemas de e-commerce (loja virtual) ou em sites dinâmicos que precisam acessar informações de bancos de dados.
Midlets	Aplicações desenvolvidas para dispositivos móveis, como celulares, smartphones e os chamados handhelds (computadores de mão), como o palm e o pocket PC.
javaBeans	Pequenas bibliotecas de códigos que podem ser reutilizados na construção de outros aplicativos, sejam stand-alone, applet, servlet ou midlet.

É importante deixar claro que a linguagem Java não tem nada a ver com o *Javascript*. Este último foi desenvolvido pela Netscape como uma linguagem de *script* para seu navegador, o *Netscape Communicator*. Muitas estruturas são parecidas entre as duas linguagens, mas sua semelhança se restringe a esse ponto.

Colocando em prática

Vamos construir o nosso primeiro programa Java. Para isso, temos que instalar o **JDK** (Java Development Kit) no nosso computador. Esse Kit de Desenvolvimento Java nos oferece todo o suporte necessário a linguagem.

Podemos encontrá-lo no site da ORACLE, <https://www.oracle.com/br/index.html>.

No Menu > Java > Java SE... No link, > Java Platform, Standard Edition

Na aba downloads, logo acima da janela central...



Temos disponível o JDK 10, mas, optaremos pelo 8 encontrado logo abaixo...

Java SE 8u171/ 8u172
Java SE 8u171 includes important bug fixes. Oracle strongly recommends that all Java SE 8 users upgrade to this release. Java SE 8u172 is a patch-set update, including all of 8u171 plus additional bug fixes (described in the release notes).
[Learn more](#) ➔

<ul style="list-style-type: none">▪ Installation Instructions▪ Release Notes▪ Oracle License▪ Java SE Licensing Information User Manual<ul style="list-style-type: none">▪ Includes Third Party Licenses▪ Certified System Configurations▪ Readme Files<ul style="list-style-type: none">▪ JDK ReadMe▪ JRE ReadMe	<div>JDK DOWNLOAD ⬇</div> <div>Server JRE DOWNLOAD ⬇</div> <div>JRE DOWNLOAD ⬇</div>
---	---

Acione o link, JDK download...

Você precisará aceitar o termo de uso, e baixar de acordo com a plataforma que está usando no seu computador...

Java SE Development Kit 8u171

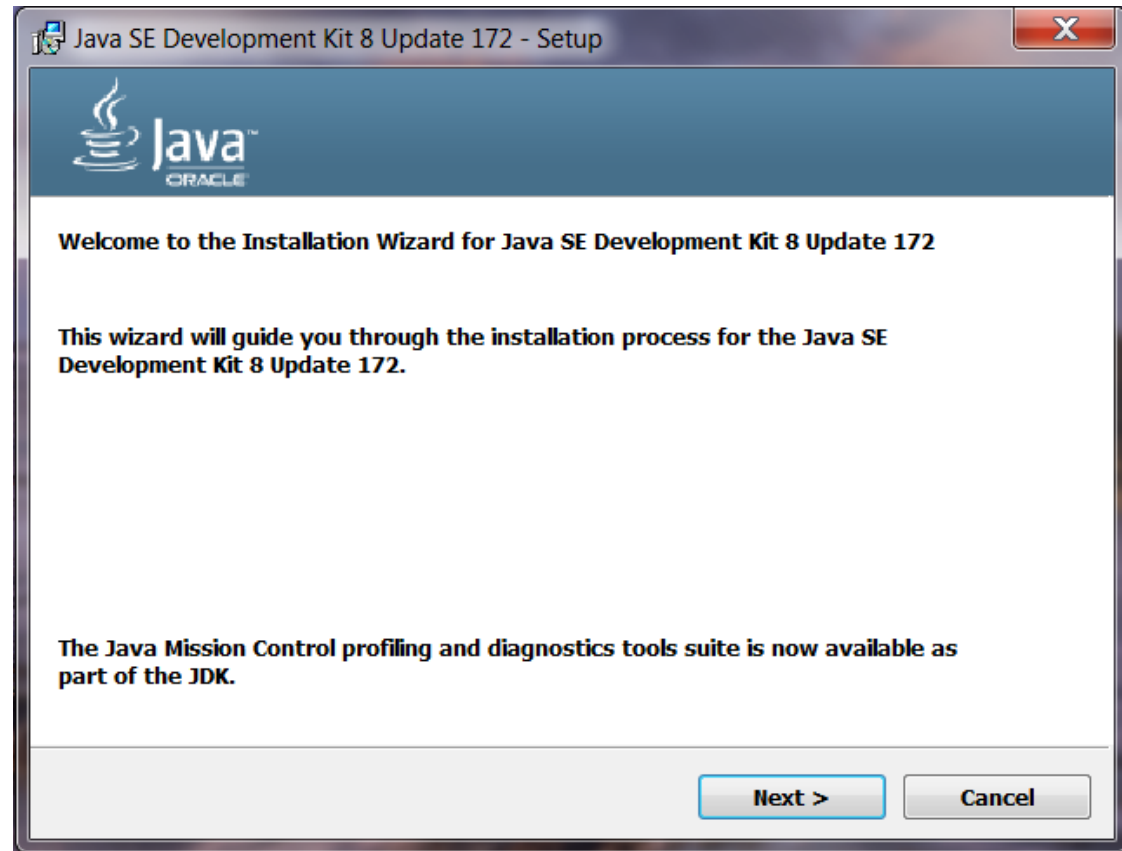
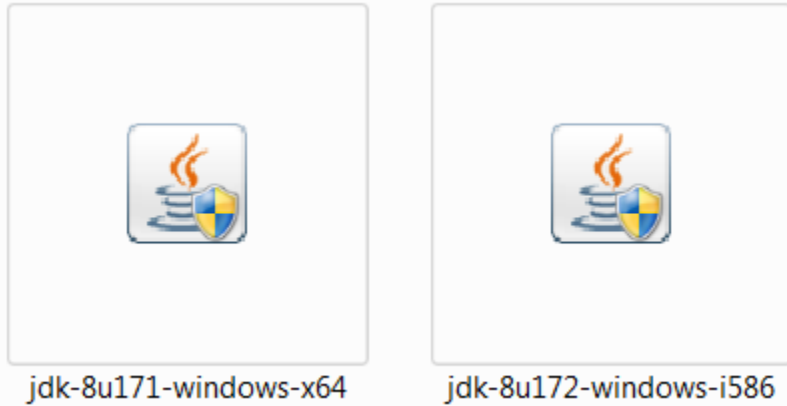
You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

☒ Accept License Agreement ☐ Decline License Agreement

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.97 MB	jdk-8u171-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.89 MB	jdk-8u171-linux-arm64-vfp-hflt.tar.gz
Linux x86	170.05 MB	jdk-8u171-linux-i586.rpm
Linux x86	184.88 MB	jdk-8u171-linux-i586.tar.gz
Linux x64	167.14 MB	jdk-8u171-linux-x64.rpm
Linux x64	182.05 MB	jdk-8u171-linux-x64.tar.gz
Mac OS X x64	247.84 MB	jdk-8u171-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	139.83 MB	jdk-8u171-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.19 MB	jdk-8u171-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	140.6 MB	jdk-8u171-solaris-x64.tar.Z
Solaris x64	97.05 MB	jdk-8u171-solaris-x64.tar.gz
Windows x86	199.1 MB	jdk-8u171-windows-i586.exe
Windows x64	207.27 MB	jdk-8u171-windows-x64.exe

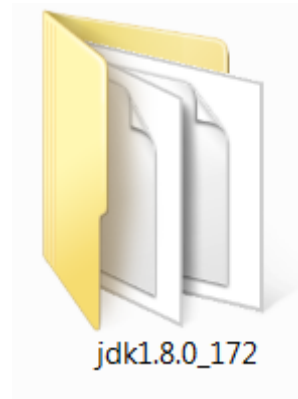
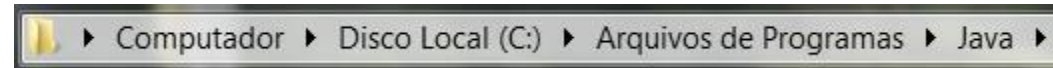
Verifique se o sistema operacional é 32 ou 64 bits, atualmente todos os processadores utilizam a arquitetura 64 bits...

Após baixar o instalador do JDK, clique duas vezes no executável e instale-o em sua máquina...



Pressione next> até que o instalador seja executado...

Após a instalação do JDK, podemos conferir os arquivos que foram instalados acessando o nosso diretório principal C:\ > em Arquivos de Programa > no diretório java >



Como vimos anteriormente, precisaremos dos programas para que as nossas classes Java sejam compiladas, gerando o *bytecode* para que a nossa máquina virtual Java (JVM) possa rodá-los. Estes programas se encontram dentro do diretório do jdk, na pasta bin >

Se acessarmos o diretório bin >, poderemos ver os programas javac e java. Estes serão utilizados durante as aulas de programação...

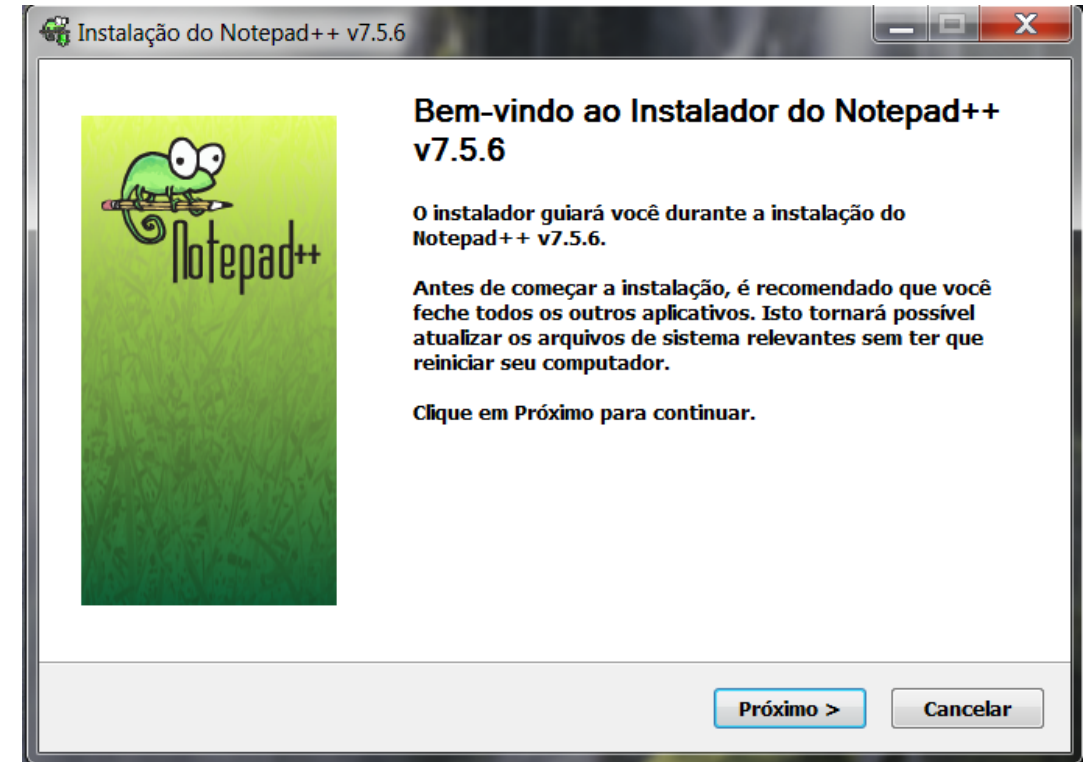
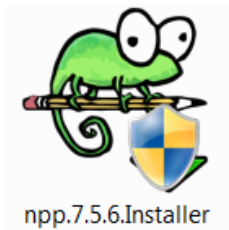


Agora precisamos de um editor de códigos, inicialmente optaremos pelo notepad++ por se tratar de um editor de fácil utilização, leve e com os recursos visuais necessários...

Podemos baixá-lo no site: <https://notepad-plus-plus.org/download/v7.5.7.html>

Notepad++ Installer 32-bit x86: Take this one if you have no idea which one you should take.

Após baixá-lo, execute o instalador do programa...



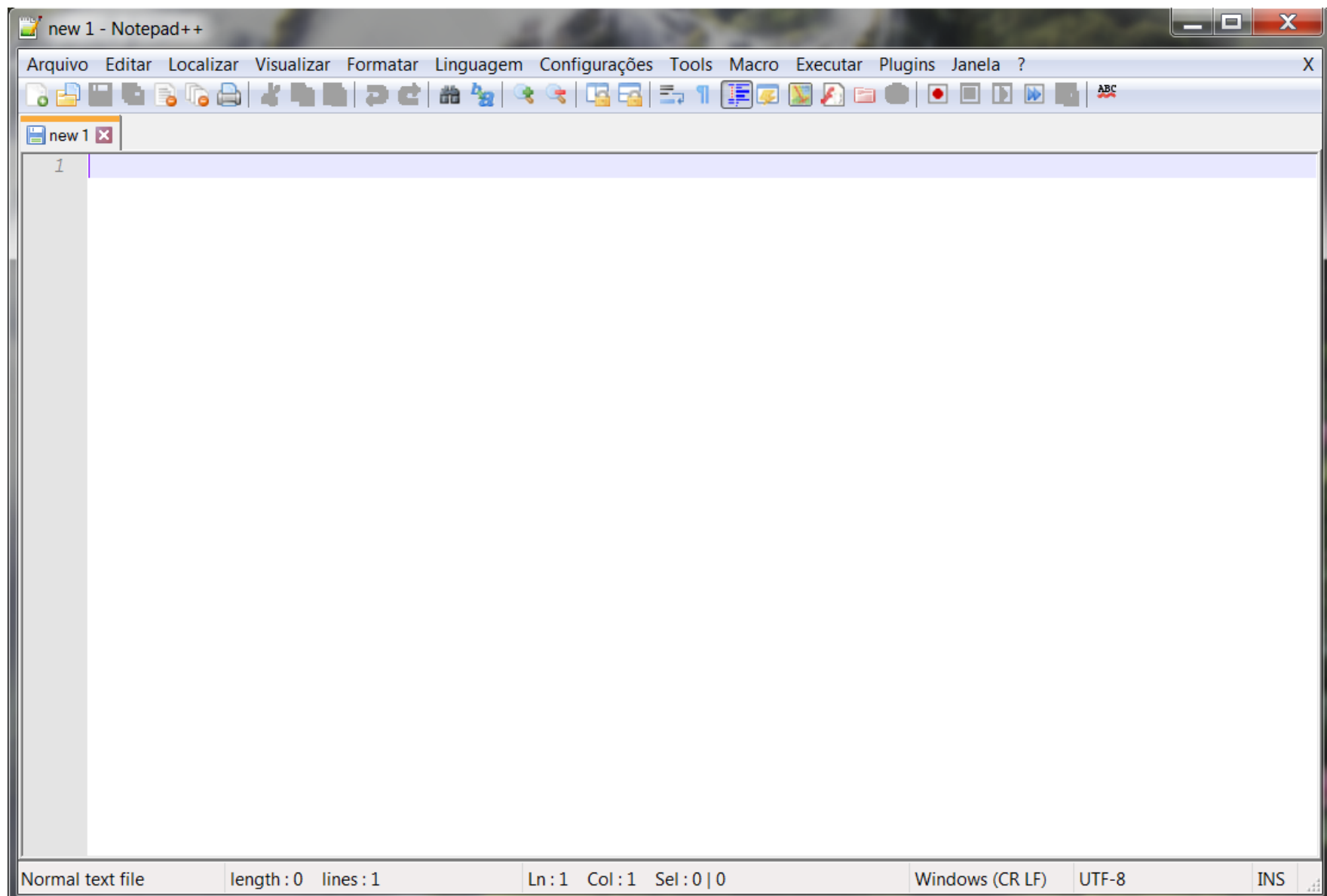
Pressione próximo até a instalação ser concluída, importante marcar a opção que cria um atalho do programa na área de trabalho do sistema operacional, antes de prosseguir com a instalação...

☐ Create Shortcut on Desktop



TECH
soluções em ti

Após a instalação do programa, execute-o...



Prosseguiremos no próximo slide...

Professor: Anderson Henrique

Programador nas Linguagens Java e PHP

