

Introdução a Linguagem Java

LAB 1

Este laboratório contém as seguintes instruções:

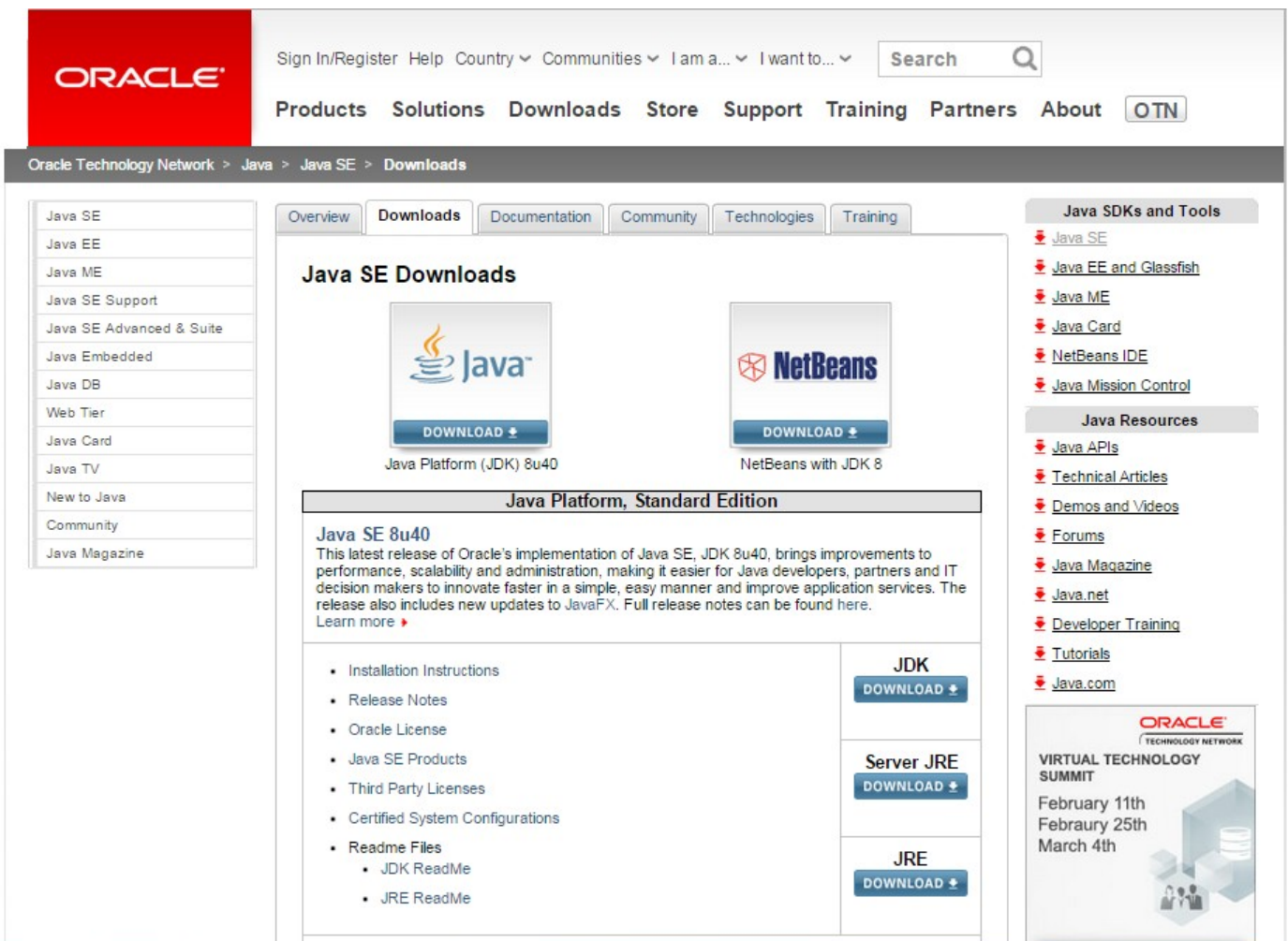
- 1º - Download do JDK 8 (Java Development Kit).
- 2º - Instalação do JDK nas plataformas (Windows e Linux).
- 3º - Configuração das variáveis de ambiente do Java nas plataformas (Windows e Linux).
- 4º - Criação do primeiro programa em Java.

Exercícios

Download do JDK 8.

Para realizar o download do JDK 8 (Java Development Kit) acesse o seguinte link da ORACLE:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>



The screenshot shows the Oracle Java SE Downloads page. The header includes the Oracle logo, navigation links (Sign In/Register, Help, Country, Communities, I am a..., I want to...), a search bar, and a main navigation menu (Products, Solutions, Downloads, Store, Support, Training, Partners, About, OTN). The breadcrumb trail is Oracle Technology Network > Java > Java SE > Downloads. The left sidebar lists various Java SE products and resources. The main content area is titled 'Java SE Downloads' and features two download buttons: 'DOWNLOAD' for Java Platform (JDK) 8u40 and 'DOWNLOAD' for NetBeans with JDK 8. Below these, the 'Java Platform, Standard Edition' section highlights the 'Java SE 8u40' release, describing its improvements in performance, scalability, and administration. A list of links for installation instructions, release notes, Oracle license, Java SE products, third-party licenses, certified system configurations, and readme files (JDK and JRE) is provided. On the right, there are sections for 'Java SDKs and Tools' (listing Java SE, Java EE and Glassfish, Java ME, Java Card, NetBeans IDE, Java Mission Control) and 'Java Resources' (listing Java APIs, Technical Articles, Demos and Videos, Forums, Java Magazine, Java.net, Developer Training, Tutorials, and Java.com). A banner for the 'VIRTUAL TECHNOLOGY SUMMIT' is also visible at the bottom right.

Esta página oferece inúmeras opções de download, por exemplo: **JDK, JRE, Documentação, Código Fonte...**

1 - Clique em Download conforme a imagem abaixo.

Java SE Downloads



Java Platform (JDK) 8u40

2 - Você será redirecionado para a seguinte página.

Java SE
Java EE
Java ME
Java SE Support
Java SE Advanced & Suite
Java Embedded
Java DB
Web Tier
Java Card
Java TV
New to Java
Community
Java Magazine

Overview
Downloads
Documentation
Community
Technologies
Training

Java SE Development Kit 8 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:

- Java Developer Newsletter (tick the checkbox under Subscription Center > Oracle Technology News)
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK MD5 Checksum

Looking for JDK 8 on ARM?
JDK 8 for ARM downloads have moved to the JDK 8 for ARM download page.

Java SE Development Kit 8u40

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 x86	146.85 MB	jdk-8u40-linux-i586.rpm
Linux x86	166.82 MB	jdk-8u40-linux-i586.tar.gz
Linux x64	145.14 MB	jdk-8u40-linux-x64.rpm
Linux x64	165.19 MB	jdk-8u40-linux-x64.tar.gz
Mac OS X x64	221.89 MB	jdk-8u40-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	131.59 MB	jdk-8u40-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	92.83 MB	jdk-8u40-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	139.45 MB	jdk-8u40-solaris-x64.tar.Z
Solaris x64	95.82 MB	jdk-8u40-solaris-x64.tar.gz
Windows x86	175.71 MB	jdk-8u40-windows-i586.exe
Windows x64	180.19 MB	jdk-8u40-windows-x64.exe

Java SDKs and Tools

- [Java SE](#)
- [Java EE and Glassfish](#)
- [Java ME](#)
- [Java Card](#)
- [NetBeans IDE](#)
- [Java Mission Control](#)

Java Resources

- [Java APIs](#)
- [Technical Articles](#)
- [Demos and Videos](#)
- [Forums](#)
- [Java Magazine](#)
- [Java.net](#)
- [Developer Training](#)
- [Tutorials](#)
- [Java.com](#)



VIRTUAL TECHNOLOGY SUMMIT

February 11th
February 25th
March 4th

REGISTER! >


3 - Habilite o download aceitando os termos da licença: **Accept License Agreement**.

4 - Na coluna **Download** clique no JDK conforme a versão do seu Sistema Operacional.

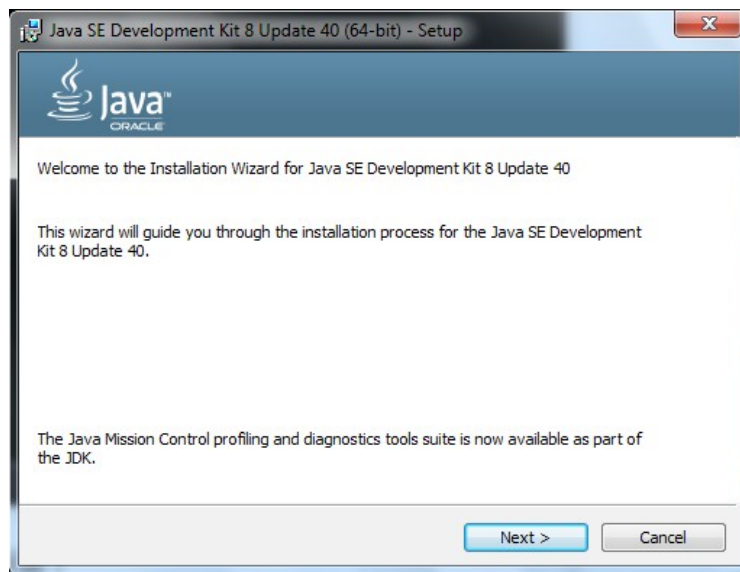
Instalação do JDK na plataforma Windows.

Após realizar o download do JDK (Java Development Kit), iremos realizar sua instalação.

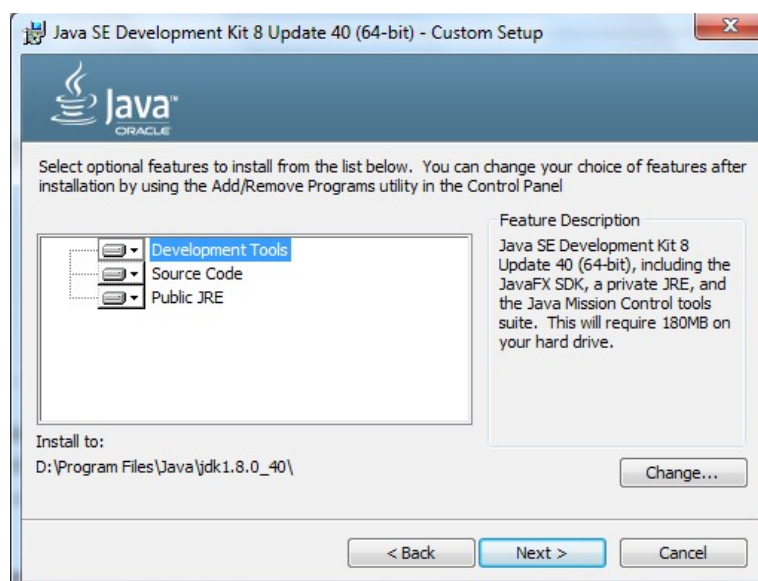
1 - Dê um clique duplo no arquivo **jdk-<versão>-windows-<arquitetura>.exe**, e aguarde o wizard de instalação ser iniciado.

Nome	Data de modificaç...	Tipo	Tamanho
 jdk-8u40-windows-x64	10/03/2015 11:33	Aplicativo	184.519 KB

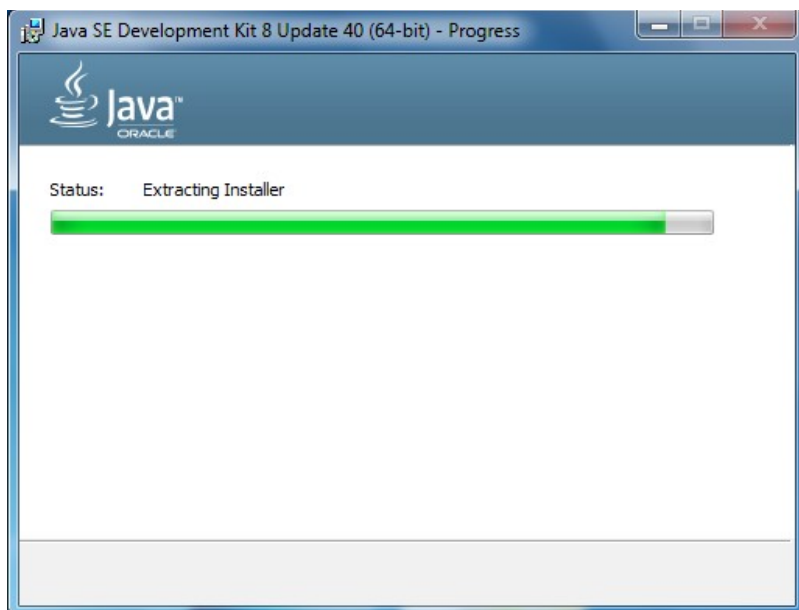
2 - Nesta tela Clique em **Next**.



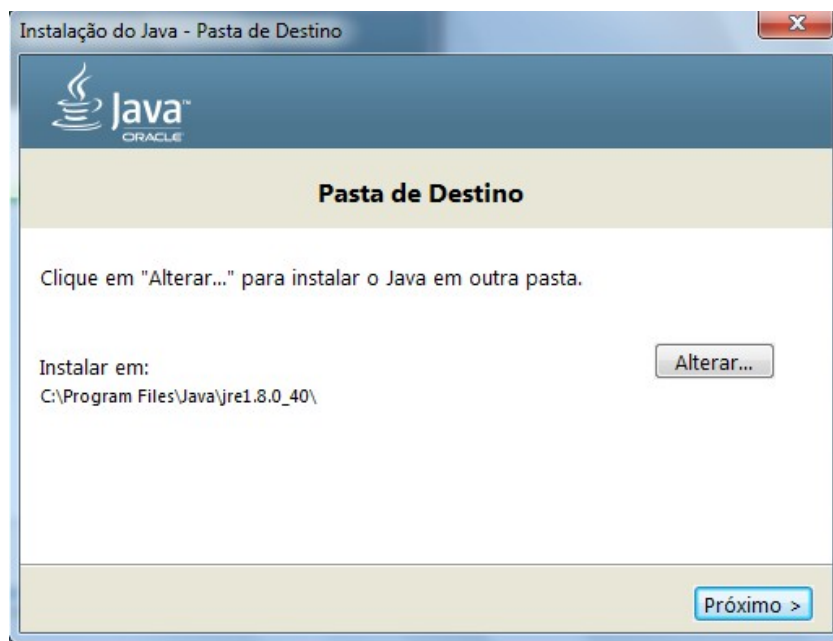
3 - Nesta tela podemos selecionar os recursos opcionais para instalar a partir da lista abaixo. Podemos mudar posteriormente esta escolha através do utilitário Adicionar / Remover Programas no Painel de Controle. Também podemos informar em qual diretório queremos que o Java seja instalado. Clique em **Next**.



4 - O processo de instalação iniciará.



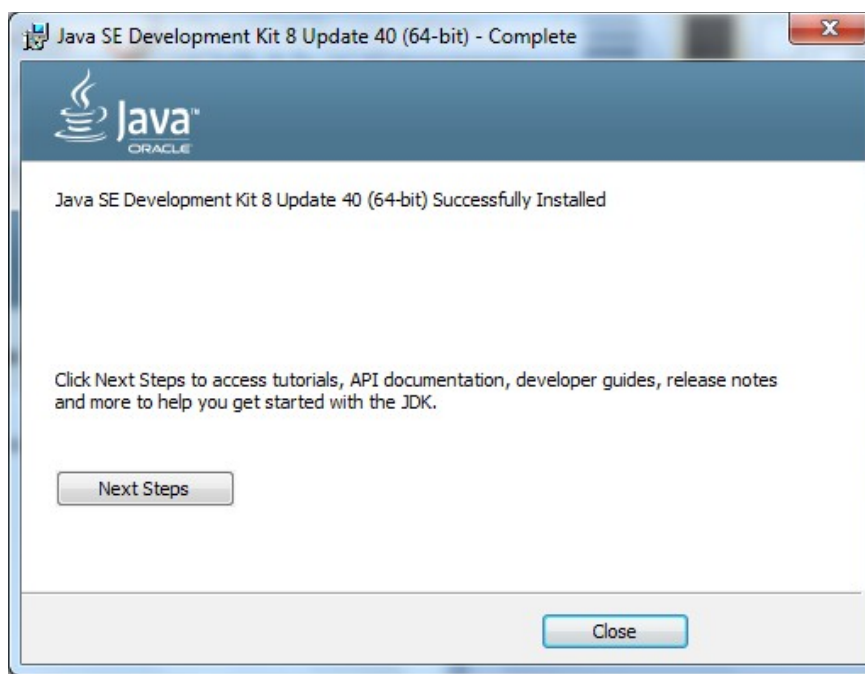
5 - Esta fase só será executada caso na 3ª etapa você não tenha desmarcado a opção **Public JRE**. O JRE (Java Runtime Environment) será instalado. Deixe como está e clique em Next.



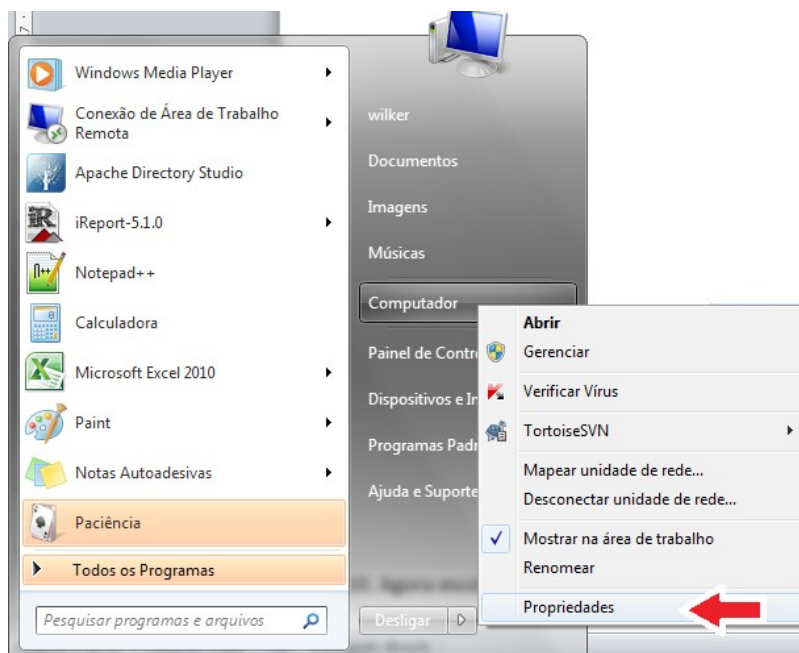
6 - O processo de instalação da **JRE (Java Runtime Environment)** iniciará.



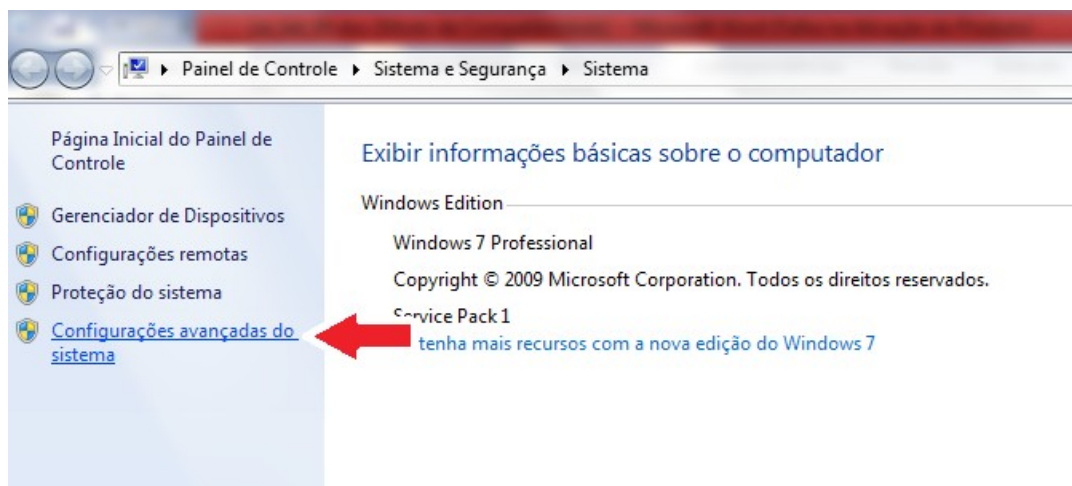
7 - Agora seu **JDK** está instalado. Clique em **Finish**.



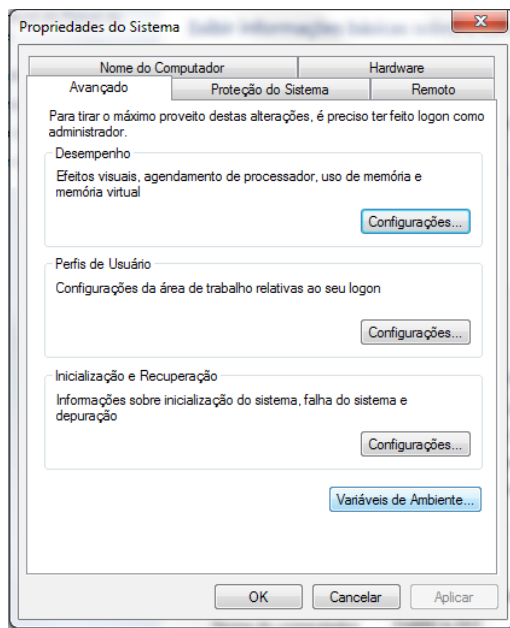
1. Acesse o menu iniciar, clique com o botão direito em **Meu Computador** e selecione a opção **Propriedades**.



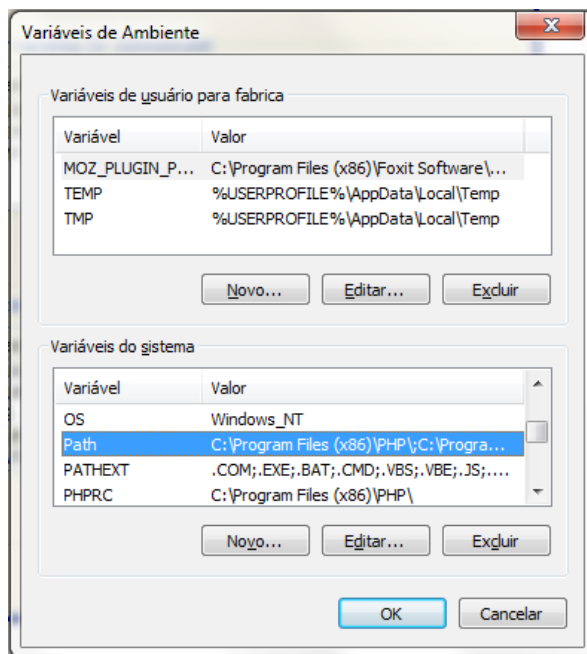
2. Clique em **Configurações avançadas do sistema**.



3. Agora selecione a aba **Avançado**, e depois clique em **Variáveis de Ambiente**.



4. Nesta tela você observará que há duas opções para configurar as variáveis de ambiente do Java: *Variáveis de usuário* e *Variáveis do sistema*. Configurando as variáveis de ambiente do Java na primeira opção, o Java só será executado através da linha de comando se o sistema operacional estiver sendo executado através do usuário **fabrica**. Se as variáveis de ambiente do Java forem configuradas através da segunda opção, independente do usuário que logar nesta máquina, o Java será executado através da linha de comando. Neste caso clique no botão NOVO da segunda opção (Variáveis de sistema).



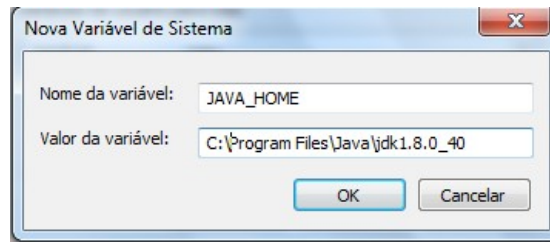
5. A primeira variável de ambiente do Java a ser configurado será a **JAVA_HOME**, através desta variável iremos informar ao sistema operacional onde o Java está instalado.

Informe os seguintes dados:

Nome da Variável: JAVA_HOME.

Valor da Variável: Endereço do diretório onde o JDK está instalado.

Depois clique em **OK**.



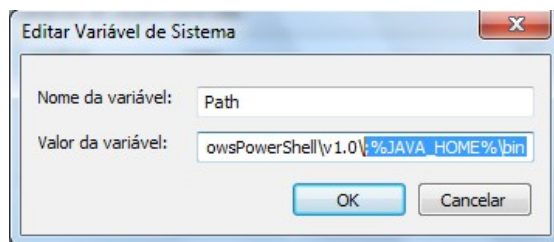
6. Agora devemos informar ao sistema operacional onde estão localizados os executáveis do Java para o mesmo ser executado através da linha de comando, esta configuração é realizada através da variável **path**. Essa variável contém diversos valores que não devem ser apagados, cada valor é separado por um ponto-e-vírgula.

No diretório de instalação do seu JDK há uma pasta chamada **bin**, esta pasta contém os executáveis que o sistema operacional irá encontrar para executar o Java em linha de comando.

Em variáveis de sistema, procure a variável **path**, selecione-a e clique em editar.

No **Valor da variável** precisamos acrescentar um ponto-e-vírgula após o último valor definido e o endereço da pasta **bin** da instalação do Java, que poderá ser assim: **;%JAVA_HOME%\bin**

OBS: **%JAVA_HOME%** é uma variável que contém o endereço de instalação do Java, podemos reutilizá-la e simplesmente informar o diretório bin precedido da barra. **\bin**

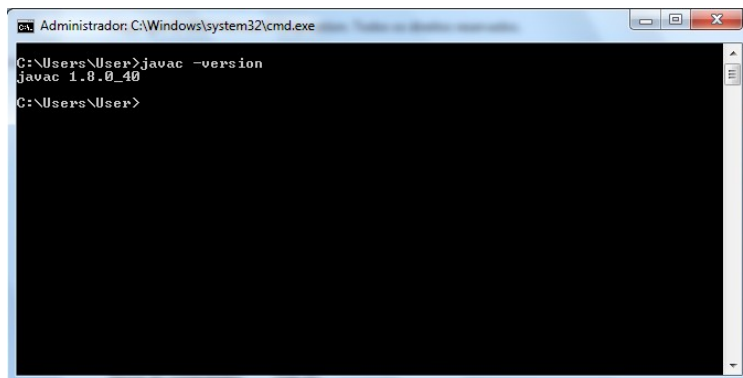


Estas são as duas variáveis que devemos configurar para o sistema operacional poder executar o Java através da linha de comando, após realizar estes passos, clique em OK, feche as demais telas e vamos confirmar se a configuração foi realizada com sucesso.

7. Acesse o menu iniciar, em **Pesquisar Programas e Arquivos**, digite: **cmd** e tecle **Enter**.

Digite **javac -version**, se nossa configuração estiver sendo executada conforme os passos acima citados, a versão do compilador do Java será apresentado conforme imagem abaixo.

Caso não apareça, reinicie sua máquina e execute somente a 7ª etapa, se mesmo assim não aparecer a versão do compilador, reveja os passos e confira se não esqueceu ou pulou nenhum deles e se o caminho colocado nas variáveis estão corretos.



1. Baixe o pacote de instalação para Linux chamado **Linux self-extracting file**, assim o arquivo que você baixará terá a extensão **.bin**.
2. Abra um terminal e faça login como **root** (comando **su**).
3. Crie um diretório chamado **/java** na raiz do seu sistema (**mkdir /java**)
4. Copie o arquivo que você baixou para o diretório **/Java** (**cp /<caminho do arquivo>/jdk-<versão>-linux.bin /java**)
5. Execute o arquivo (**sh jdk<versão>-linux.bin**)
6. Aparecerá na sua tela um acordo de instalação, aperte a tecla **Q** para ir até o final do documento e depois escreva **yes** para aceitar o contrato. Feito isso ele começará a descompactar o JDK em um diretório que ele cria, chamado **jdk<versão>**.
7. Vamos agora criar as variáveis de ambiente, edite o arquivo, use um editor de sua preferência, altere seu arquivo **.bashrc** ou **.bash_profile** (depende da sua distribuição e sua preferência) ou ainda **/etc/bash.bashrc**, considerando que você não tem privilégios de super usuário, faça:

```
> ~/.bashrc  
> vi ~/.bashrc
```
8. Adicione as seguintes linhas no final do arquivo, substituindo **<versão>** pela versão que foi instalado do JDK:

```
JAVA_HOME=/java/jdk<versão>  
PATH=$PATH:$JAVA_HOME/bin  
CLASSPATH=.export JAVA_HOME PATH CLASSPATH
```
9. Execute o comando **source ~/.bashrc**
10. Abra um terminal e teste o **Java Compiler** executando o comando **javac -version**
11. A saída deverá ser parecida com a apresentada abaixo, se não aparecer reveja os passos e confira se foi feito tudo corretamente, inclusive o caminho dos diretórios.

```
javac 1.7.0_25  
javac: no source files  
Usage: javac <options> <source files>  
where possible options include:  
.....
```

Escrever, compilar e executar um programa em Java.

O objetivo destes exercícios é promover uma experiência completa desde a instalação, configuração e criação de um simples programa em Java que será executado através da linha de comando do seu sistema operacional.

Se você já programou em linguagens como C, C++, verá que há pequenas diferenças.

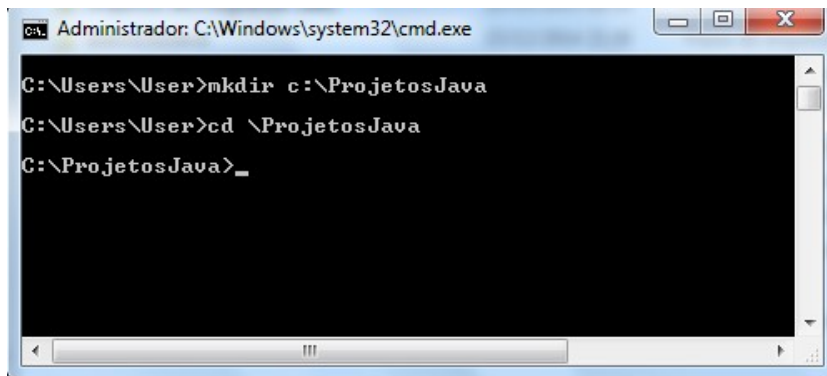
1. Criar o programa Java **HelloWorld**.

2. Compilar o programa através do compilador **javac**
3. Executar o programa através do interpretador **java**

Criar e executar o programa Java HelloWorld, usando o comando *javac* e *java*

1. Crie um diretório em seu sistema operacional onde você colocará seus programas que escrever em Java.
Acesso o menu iniciar, em **Pesquisar programas e arquivos** digite **cmd** e tecla enter.

- C:\Users\User> **mkdir c:\ProjetosJava**
- C:\Users\User> **cd \ProjetosJava**



OBS: Você também pode criar o diretório informado acessando o ícone **Meu Computador**, **Disco Local (C:)**, clique com o botão direito do mouse e clique na opção **novo** depois clique em **pasta**. Um diretório será criado, informe o nome **ProjetosJava**.

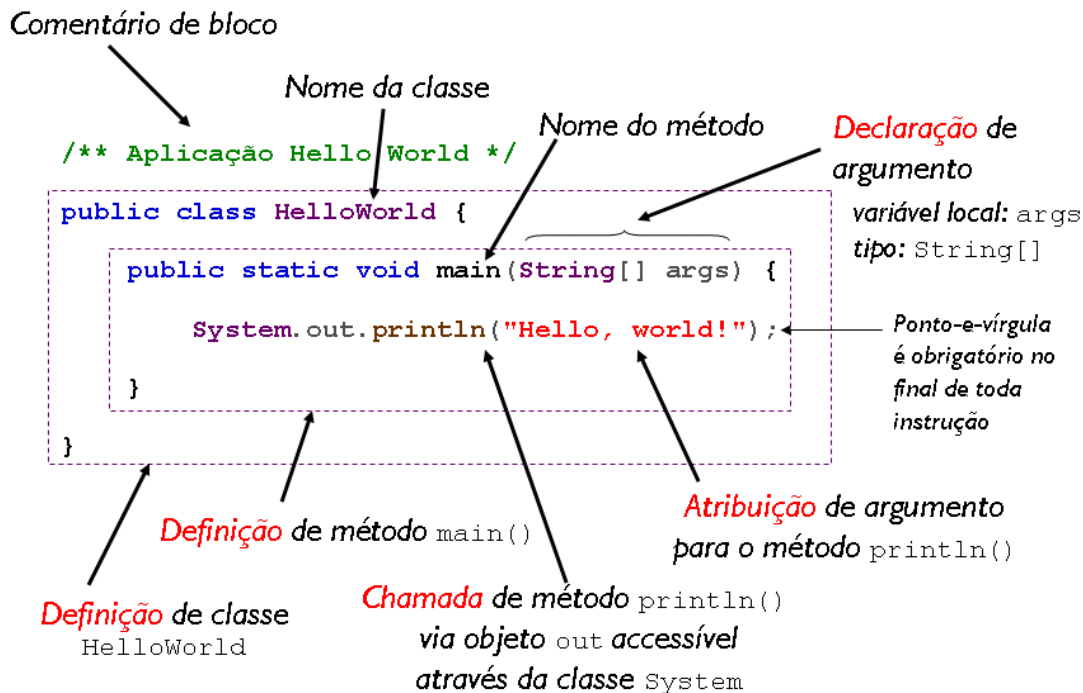
2. Escreva o código **HelloWorld.java** usando o editor de sua preferência, podendo ser o Textpad, Notepad ou Notepad++ no Windows; ou vi, gedit no Linux. Veja o gabarito do programa na Listagem-0.1 abaixo.

Nestes primeiros exemplos é interessante que você digite e tenha experiência com alguns erros compilação.
Após criar o arquivo, você deve salvá-lo no diretório **"ProjetosJava"** com o nome **HelloWorld.java**

```
1 public class HelloWorld {
2     /**
3      *Aplicação Hello World
4      */
5     public static void main( String[] args ){
6
7         // Imprime cadeia de caracteres "Hello world" na tela
8         System.out.println("Hello, world");
9     }
10 }
11
```

Listagem -0.1: HelloWorld.java

Anatomia do Código:



3. Com o Prompt de Comando aberto, iremos compilar o nosso programa através do compilador **javac**.

O compilador **javac** foi instalado com seu **JDK**, ele está no diretório **%JAVA_HOME%\bin** (Windows) ou **\$JAVA_HOME/bin** (Linux). O resultado da compilação será a criação do arquivo **HelloWorld.class**.

> c:\ProjetosJava> javac HelloWorld.java

Problema #1: se você receber a mensagem de erro abaixo

> HelloWorld.java

'javac' is not recognized as an internal or external command, operable program or batch file

Isto significa que **%JAVA_HOME%\bin** para Windows ou **\$JAVA_HOME/bin** para Linux, não está configurada corretamente. Você pode tentar colocando **C:\Program Files\Java\jdk1.8.0_40\bin\javac HelloWorld.java** (para Windows) se quiser continuar sem configurar as variáveis de ambiente.

Solução: Sugiro que você reveja a 6ª etapa do item 1.3 (para Windows) ou a 8ª etapa do item 1.4 (para Linux).

4. Certifique-se de que o arquivo **HelloWorld.class** foi criado. O arquivo **HelloWorld.class** contém a representação **bytecode** da classe **HelloWorld**.

5. Rode o programa HelloWorld usando o comando **java**.

O comando **java** inicia a **Máquina Virtual Java** e executa o programa **HelloWorld**.

>c:\ProjetosJava> java HelloWorld

Problema #2: Se você receber a mensagem de erro

>java HelloWorld

Exception in thread "main" java.lang.UnsupportedClassVersionError: HelloWorld (Unsupported major.minor version 49.0)

É possível que você tenha compilado seu programa com uma versão diferente da que esteja tentando executá-lo.

Solução: certifique que sua variável de ambiente **PATH** esteja corretamente configurada para o diretório de instalação da versão apropriada do seu **JDK**.

Sugiro que você reveja a 6ª etapa do item 1.3 (para Windows) ou a 8ª etapa do item 1.4 (para Linux).

6. Modifique **HelloWorld.java** como mostra a **Listagem-0.2** abaixo.

```
1 public class HelloWorld {
2     /**
3      * Comentário:
4      * Primeiro programa em Java
5      */
6     public static void main( String[] args ){
7         // também é um comentário
8         // Imprime cadeia de caracteres "Hello world" na tela
9         System.out.println("Olá Java");
10        //modifique para imprimir mais
11        System.out.println("Meu primeiro programa Java");
12    }
13 }
14
15
```

Listagem-0.2: HelloWorld.java modificado

7. Compile e execute o programa. Observe a nova mensagem mostrada.

```
>c:\ProjetosJava> javac HelloWorld.java
```

```
>c:\ProjetosJava> java HelloWorld
```

Olá Java

Meu primeiro programa Java

Usar o compilador Java, comando *javac*

Neste passo você aprenderá como usar algumas opções do compilador *javac*.

1. Mostrar as opções do compilador *javac* usando a opção **-help**:

```
> javac -help
```

Usage: javac <options> <source files>

where possible options include:

- g* Generate all debugging info
- g:none* Generate no debugging info
- g:{lines,vars,source}* Generate only some debugging info
- nowarn* Generate no warnings
- verbose* Output messages about what the compiler is doing
- deprecation* Output source locations where deprecated APIs are used
- classpath <path>* Specify where to find user class files
- cp <path>* Specify where to find user class files
- sourcepath <path>* Specify where to find input source files
- bootclasspath <path>* Override location of bootstrap class files
- extdirs <dirs>* Override location of installed extensions
- endorseddirs <dirs>* Override location of endorsed standards path
- d <directory>* Specify where to place generated class files
- encoding <encoding>* Specify character encoding used by source files
- source <release>* Provide source compatibility with specified release
- target <release>* Generate class files for specific VM version
- version* Version information
- help* Print a synopsis of standard options
- X* Print a synopsis of nonstandard options
- J<flag>* Pass <flag> directly to the runtime system

2. Execute o comando *javac* com a opção **-verbose**

```
>javac -verbose HelloWorld.java
[parsing started HelloWorld.java]
[parsing completed 516ms]
[search path for source files: [..]]
[search path for class files: [C:\Program Files\Java\jdk1.7.0_25\jre\lib\rt.jar,C:\Program Files\Java\jdk1.7.0_25\jre\lib\jsse.jar,
C:\Program Files\Java\jdk1.7.0_25\jre\lib\jce.jar, C:\Program Files\Java\jdk1.7.0_25\jre\lib\charsets.jar,C:\Program
Files\Java\jdk1.7.0_25\jre\lib\ext\dnsns.jar, C:\Program Files\Java\jdk1.7.0_25\jre\lib\ext\localedata.jar, C:\Program
Files\Java\jdk1.7.0_25\jre\lib\ext\sunjc_provider.jar, C:\Program Files\Java\jdk1.7.0_25\jre\lib\ext\sunpkcs11.jar, .]]
[loading C:\Program Files\Java\jdk1.7.0_25\jre\lib\rt.jar(java/lang/Object.class)]
[loading C:\Program Files\Java\jdk1.7.0_25\jre\lib\rt.jar(java/lang/String.class)]
[checking HelloWorld]
[loading C:\Program Files\Java\jdk1.7.0_25\jre\lib\rt.jar(java/lang/System.class)]
[loading C:\Program Files\Java\jdk1.7.0_25\jre\lib\rt.jar(java/io/PrintStream.class)]
[loading C:\Program Files\Java\jdk1.7.0_25\jre\lib\rt.jar(java/io/FilterOutputStream.class)]
[loading C:\Program Files\Java\jdk1.7.0_25\jre\lib\rt.jar(java/io/OutputStream.class)]
[wrote HelloWorld.class]
[total 1141ms]
```

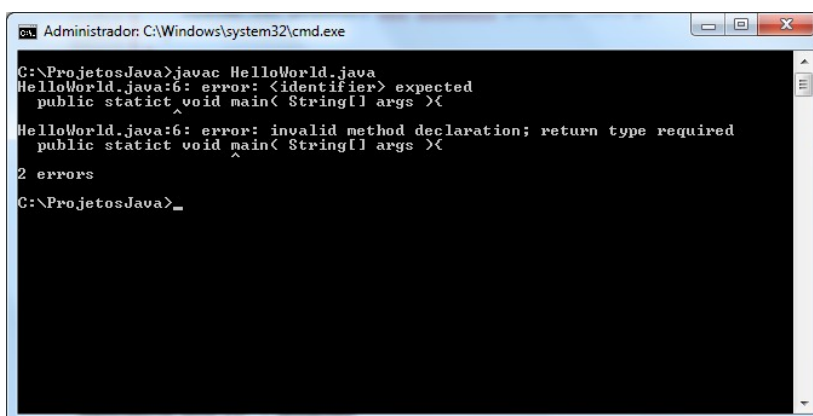
3. Experiência com erros de compilação.

- Modifique o programa **HelloWorld.java** como mostra a **Listagem-0.3** abaixo. A mudança causa um erro de sintaxe, a palavra-chave **static** está incorretamente digitada como **statict**.

```
1 public class HelloWorld {
2     /**
3      * Comentário:
4      * Primeiro programa em Java
5      */
6     public statict void main( String[] args ){
7         // também e um comentario
8         // Imprime cadeia de caracteres na tela
9         System.out.println("Meu primeiro program Java");
10    }
11 }
12
13
14
```

Listagem -0.3: Código contendo erro de sintaxe

- Compile HelloWorld.java com javac.
- Observem os erros



```
C:\ProjetosJava>javac HelloWorld.java
HelloWorld.java:6: error: <identifier> expected
    public statict void main( String[] args ){
           ^
HelloWorld.java:6: error: invalid method declaration; return type required
    public statict void main( String[] args ){
           ^
2 errors
C:\ProjetosJava>
```

- Remova o erro de sintaxe que você introduziu no código e o compile e execute novamente.

Usar o interpretador Java, comando *java*

Neste passo você aprenderá como usar algumas opções do interpretador **java**. Você pode pensar no comando como sua **Java Virtual Machine**.

1. Mostrar o uso das informações de ajuda do comando **java** com a opção **-help** ou **-?**.

```
>java -help
Usage: java [-options] class [args...]
        (to execute a class)
or java [-options] -jar jarfile [args...]
        (to execute a jar file)

where options include:
-client    to select the "client" VM
-server    to select the "server" VM
-hotspot   is a synonym for the "client" VM [deprecated]
           The default VM is client.

-cp <class search path of directories and zip/jar files>
-classpath <class search path of directories and zip/jar files>
           A ; separated list of directories, JAR archives,
           and ZIP archives to search for class files.
-D<name>=<value>
           set a system property
-verbose[:class|gc|jni]
           enable verbose output
-version   print product version and exit
-version:<value>
           require the specified version to run
-showversion print product version and continue
-jre-restrict-search | -jre-no-restrict-search
           include/exclude user private JREs in the version search
-? -help   print this help message
-X         print help on non-standard options
-ea[:<packagename>...]:<classname>]
-enableassertions[:<packagename>...]:<classname>]
           enable assertions
-da[:<packagename>...]:<classname>]
-disableassertions[:<packagename>...]:<classname>]
           disable assertions
-esa | -enablesystemassertions
           enable system assertions
-dsa | -disablesystemassertions
           disable system assertions
-agentlib:<libname>[=<options>]
           load native agent library <libname>, e.g. -agentlib:hprof
           see also, -agentlib:jdwp=help and -agentlib:hprof=help
-agentpath:<pathname>[=<options>]
           load native agent library by full pathname
-javaagent:<jarpath>[=<options>]
           load Java programming language agent, see java.lang.instrument
```

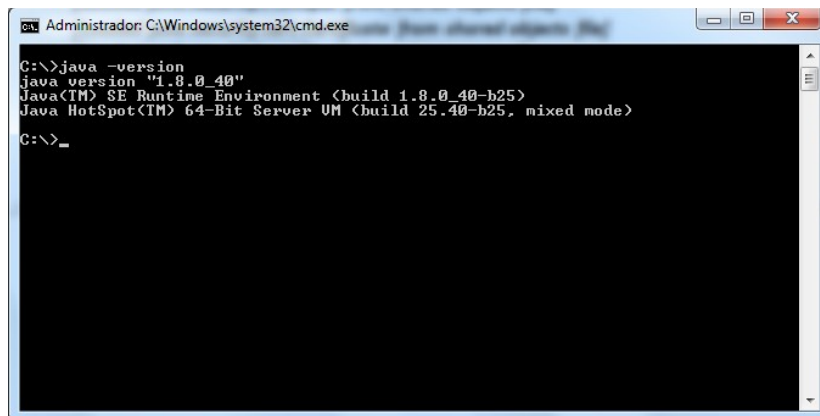
2. Execute o comando **java** com a opção **-verbose**

```
>java -verbose HelloWorld
[Opened C:\Program Files\Java\jre1.7.0_25\lib\rt.jar]
[Opened C:\Program Files\Java\jre1.7.0_25\lib\jsse.jar]
[Opened C:\Program Files\Java\jre1.7.0_25\lib\jce.jar]
[Opened C:\Program Files\Java\jre1.7.0_25\lib\charsets.jar]
[Loaded java.lang.Object from shared objects file]
[Loaded java.io.Serializable from shared objects file]
[Loaded java.lang.Comparable from shared objects file]
[Loaded java.lang.CharSequence from shared objects file]
[Loaded java.lang.String from shared objects file]
[Loaded java.lang.reflect.GenericDeclaration from shared objects file]
[Loaded java.lang.reflect.Type from shared objects file]
[Loaded java.lang.reflect.AnnotatedElement from shared objects file]
```



```
[Loaded java.lang.Class from shared objects file]
[Loaded java.lang.Cloneable from shared objects file]
[Loaded java.lang.ClassLoader from shared objects file]
[Loaded java.lang.System from shared objects file]
[Loaded java.lang.Throwable from shared objects file]
[Loaded java.lang.Error from shared objects file]
[Loaded java.lang.ThreadDeath from shared objects file]
[Loaded java.lang.Exception from shared objects file]
[Loaded java.lang.RuntimeException from shared objects file]
[Loaded java.security.ProtectionDomain from shared objects file]
[Loaded java.security.AccessControlContext from shared objects file]
[Loaded java.lang.ClassNotFoundException from shared objects file]
[Loaded java.lang.LinkageError from shared objects file]
[Loaded java.lang.NoClassDefFoundError from shared objects file]
[Loaded java.lang.ClassCastException from shared objects file]
[Loaded java.lang.ArrayStoreException from shared objects file]
[Loaded java.lang.VirtualMachineError from shared objects file]
....
[Loaded sun.net.www.URLConnection from shared objects file]
[Loaded sun.net.www.protocol.file.FileURLConnection from shared objects file]
[Loaded java.net.ContentHandler from shared objects file]
[Loaded java.net.UnknownContentHandler from shared objects file]
[Loaded sun.net.www.MessageHeader from shared objects file]
[Loaded java.io.FilePermission from shared objects file]
[Loaded java.io.FilePermission$1 from shared objects file]
[Loaded java.io.FilePermissionCollection from shared objects file]
[Loaded java.security.AllPermission from shared objects file]
[Loaded java.security.UnresolvedPermission from shared objects file]
[Loaded java.security.BasicPermissionCollection from shared objects file]
[Loaded java.security.Principal from shared objects file]
[Loaded java.security.cert.Certificate from shared objects file]
[Loaded HelloWorld from file:/C:/myjavaprograms/]
This is my first Java program!
[Loaded java.lang.Shutdown from shared objects file]
[Loaded java.lang.Shutdown$Lock from shared objects file]
```

3. Execute o comando **java** com a opção **-version**. Observe que o número da versão do JDK instalada é mostrado.



```
Administrador: C:\Windows\system32\cmd.exe
C:\>java -version
java version "1.8.0_40"
Java(TM) SE Runtime Environment (build 1.8.0_40-b25)
Java HotSpot(TM) 64-Bit Server VM (build 25.40-b25, mixed mode)
C:\>_
```