

# Introdução ao Java

## Capítulo I

## Preparando nosso ambiente

O que iremos instalar ?

- O Kit de Desenvolvimento Java – jdk
- Um ambiente de desenvolvimento integrado, IDE - NetBeans

## Preparando nosso ambiente

### Baixar a JDK [aqui](#) (Versão 8)

[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: From your Oracle account, select **Subscriptions**, expand **Technology**, and subscribe to **Java**.
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

[JDK 8u201 checksum](#)  
[JDK 8u202 checksum](#)

#### Java SE Development Kit 8u201

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	72.98 MB	<a href="#">jdk-8u201-linux-arm32-vfp-hflt.tar.gz</a>
Linux ARM 64 Hard Float ABI	69.92 MB	<a href="#">jdk-8u201-linux-arm64-vfp-hflt.tar.gz</a>
Linux x86	170.98 MB	<a href="#">jdk-8u201-linux-i586.rpm</a>
Linux x86	185.77 MB	<a href="#">jdk-8u201-linux-i586.tar.gz</a>
Linux x64	168.05 MB	<a href="#">jdk-8u201-linux-x64.rpm</a>
Linux x64	182.93 MB	<a href="#">jdk-8u201-linux-x64.tar.gz</a>
Mac OS X x64	245.92 MB	<a href="#">jdk-8u201-macosx-x64.dmg</a>
Solaris SPARC 64-bit (SVR4 package)	125.33 MB	<a href="#">jdk-8u201-solaris-sparcv9.tar.Z</a>
Solaris SPARC 64-bit	88.31 MB	<a href="#">jdk-8u201-solaris-sparcv9.tar.gz</a>
Solaris x64 (SVR4 package)	133.99 MB	<a href="#">jdk-8u201-solaris-x64.tar.Z</a>
Solaris x64	92.16 MB	<a href="#">jdk-8u201-solaris-x64.tar.gz</a>
Windows x86	197.66 MB	<a href="#">jdk-8u201-windows-i586.exe</a>
Windows x64	207.46 MB	<a href="#">jdk-8u201-windows-x64.exe</a>



- Escolher seu SO
- Aceitar os termos de uso

Preparando nosso ambiente

Será necessário criar uma conta da Oracle

Início de sessão na  
conta Oracle

 Nome de Utilizador 

 Senha 

**Iniciar sessão**

[Precisa de ajuda?](#)

---

**Criar Conta**

© Oracle | Condições de Utilização | Política de Privacidade



Preparando nosso ambiente  
Preencha todos os seus dados pessoais e depois clique em *Criar conta*

## Crie a sua Conta Oracle

Já tem uma Conta Oracle? [Iniciar Sessão](#)

Endereço de e-mail \*

O seu endereço de e-mail é o seu nome de utilizador.

✓ Irá ser-lhe enviada uma mensagem de e-mail de confirmação.

Senha \*

As senhas têm de incluir letras minúsculas e maiúsculas, pelo menos 1 número, um mínimo de 8 caracteres e não podem corresponder a nenhuma parte do seu e-mail.

✓ A senha cumpre os requisitos

Reintroduzir senha \*

✓

☐ Sim, enviem-me comunicações de marketing sobre Produtos, Serviços e Eventos da Oracle.

Pode optar por não participar em nenhuma das comunicações de marketing: [Anular Subscrição](#).

Ao clicar no botão "Criar conta" abaixo, compreende e aceita que a utilização do site da Oracle está sujeita às [Condições de Utilização de Oracle.com](#). Os detalhes adicionais relacionados com a recolha e utilização das suas informações pessoais pela Oracle, incluindo informações sobre o acesso, a retenção, a retificação, a eliminação, a segurança, transferências internacionais e outros tópicos, estão disponíveis na [Política de Privacidade da Oracle](#).



**Criar conta**

Preparando nosso ambiente

Após criar a conta e clicar em iniciar sessão, o download irá começar automaticamente

## Preparando nosso ambiente

Baixar o netbeans em <https://netbeans.org/downloads/8.2/>

### Download o NetBeans IDE 8.2

8.1 | 8.2 | Desenvolvimento | Arquivo

Endereço de email (opcional):

Inscriver-se na newsletter:

☒ Mensal

☐ Semanal

☒ Permito me contatar neste email

Idioma  
do  
IDE:

Português (Bras

Plataforma: Windows

Nota: Tecnologias em cinza não são suportadas para esta plataforma.

### Distribuições para baixar do NetBeans IDE

Tecnologias suportadas *	Java SE	Java EE	HTML5/JavaScript	PHP	C/C++	Tudo
① SDK da plataforma NetBeans	•	•				•
① Java SE	•	•				•
① Java FX	•	•				•
① Java EE		•				•
① Java ME						•
① HTML5/JavaScript		•	•	•		•
① PHP			•	•		•
① C/C++					•	•
① Groovy						•
① Java Card(tm) 3 Connected						•
Servidores embutidos						
① GlassFish Server Open Source Edition 4.1.1		•				•
① Apache Tomcat 8.0.27		•				•

Download

Download

Download x86

Download x86

Download x86

Download

Download x64

Download x64

Download x64

95 MB livre(s)

197 MB livre  
(s)

108 - 112 MB livre  
(s)

108 - 112 MB livre  
(s)

107 - 110 MB livre  
(s)

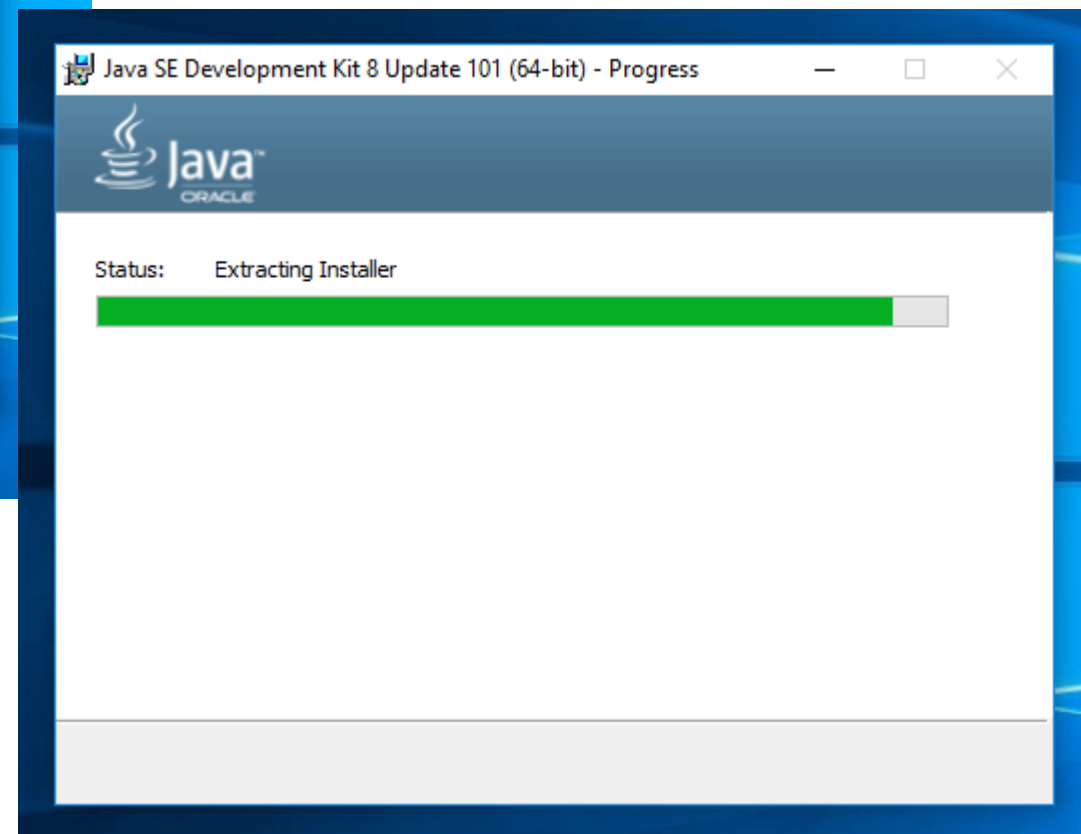
221 MB livre  
(s)

## Preparando no Windows

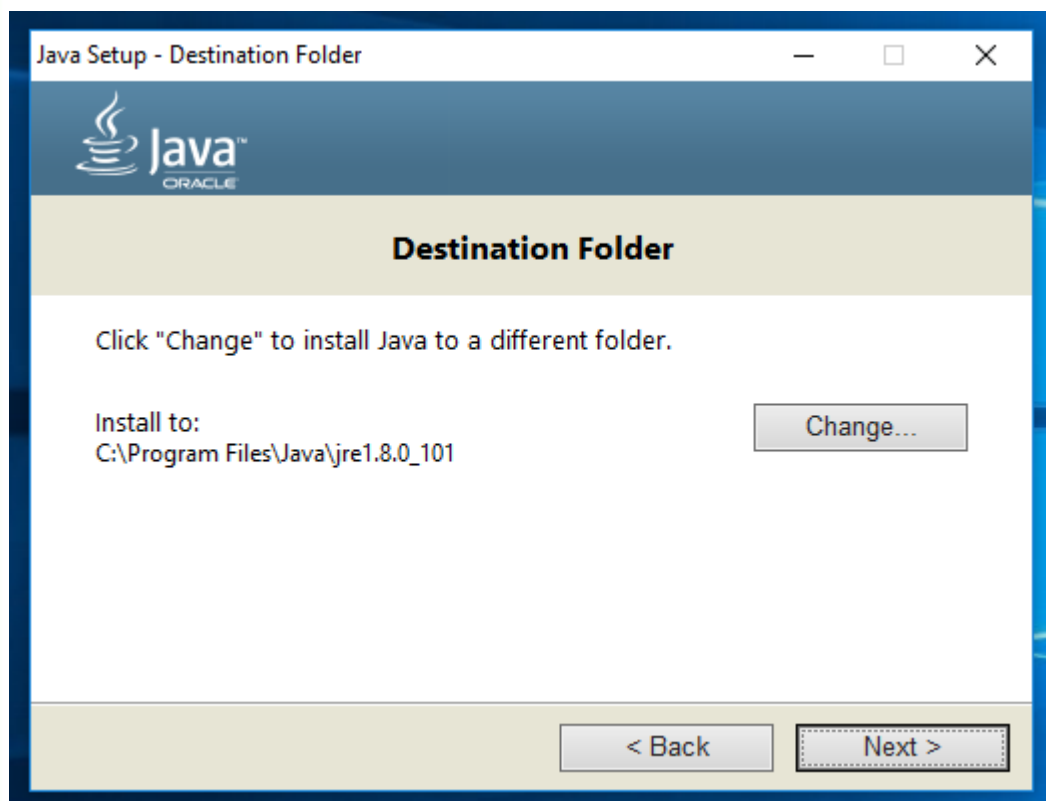




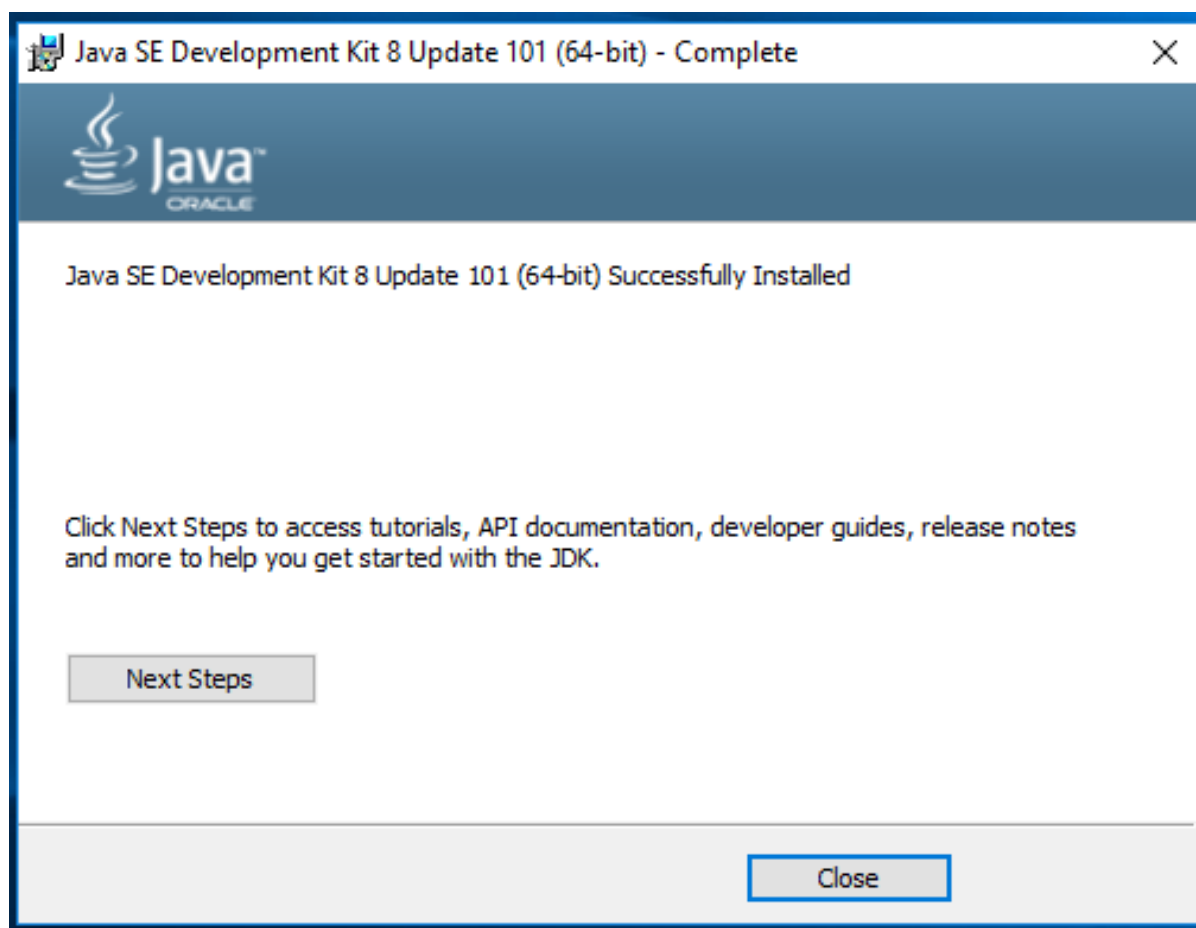
## Instalando a JDK no Windows



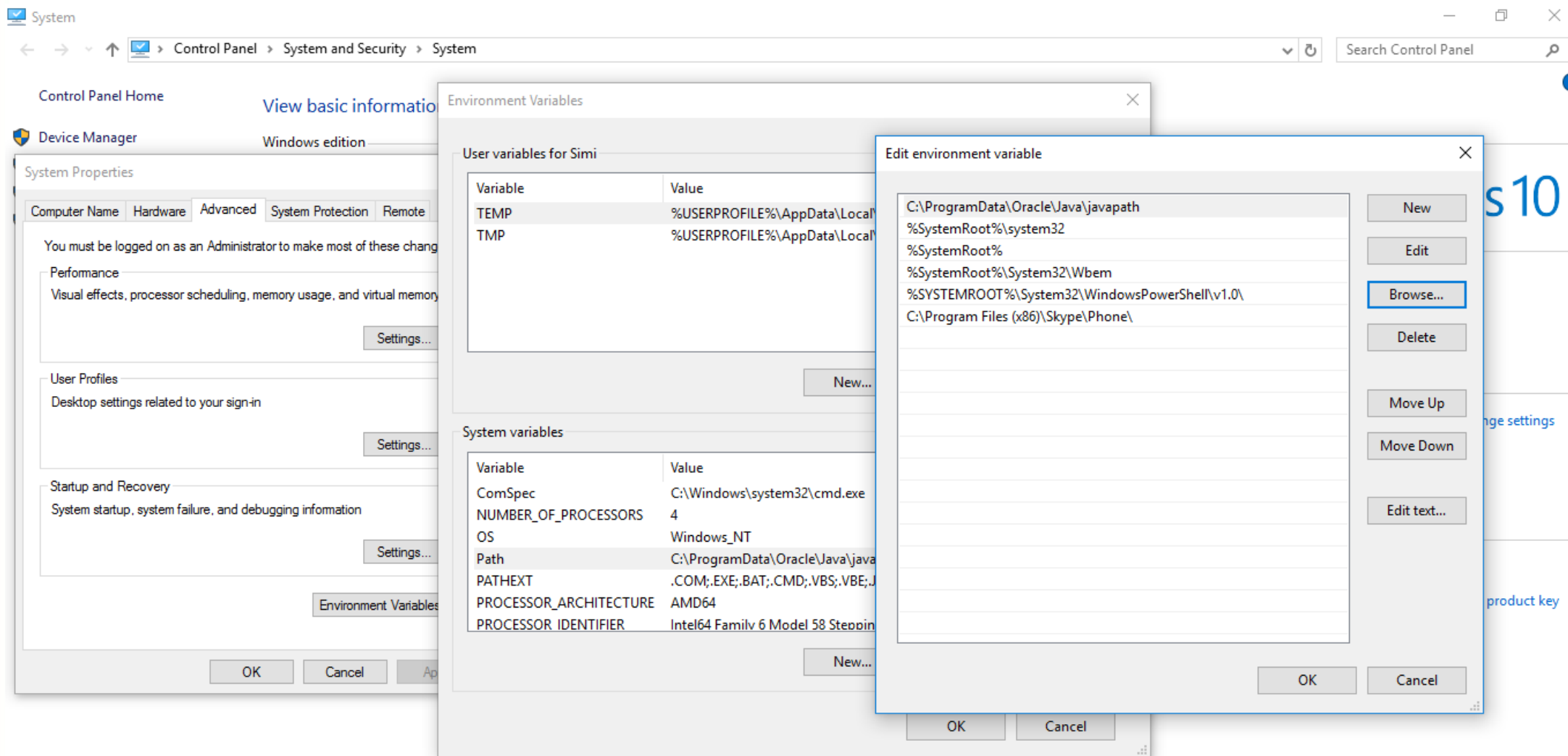
## Instalando a JDK no Windows



## Instalando a JDK no Windows

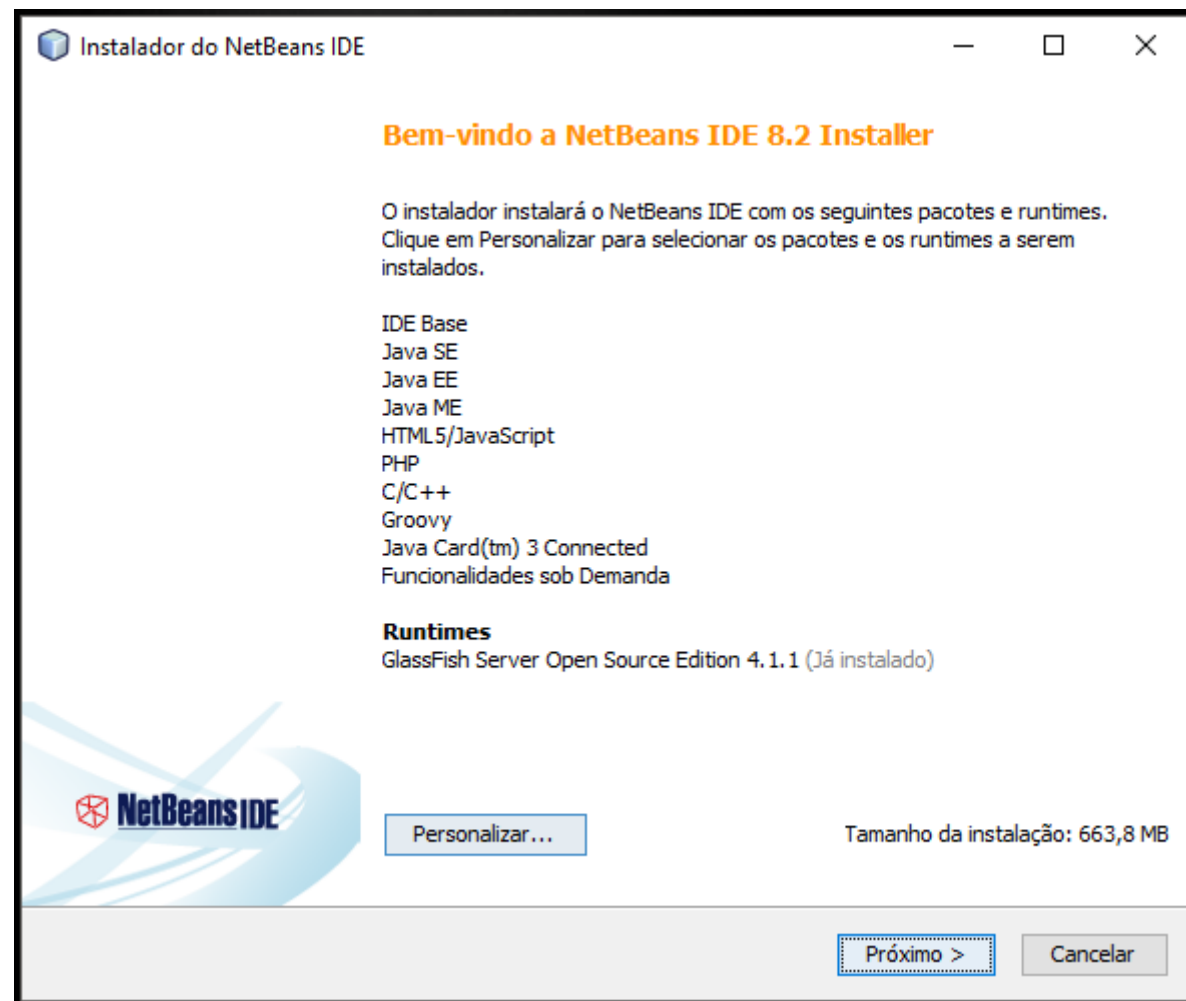


## Instalando a JDK no Windows

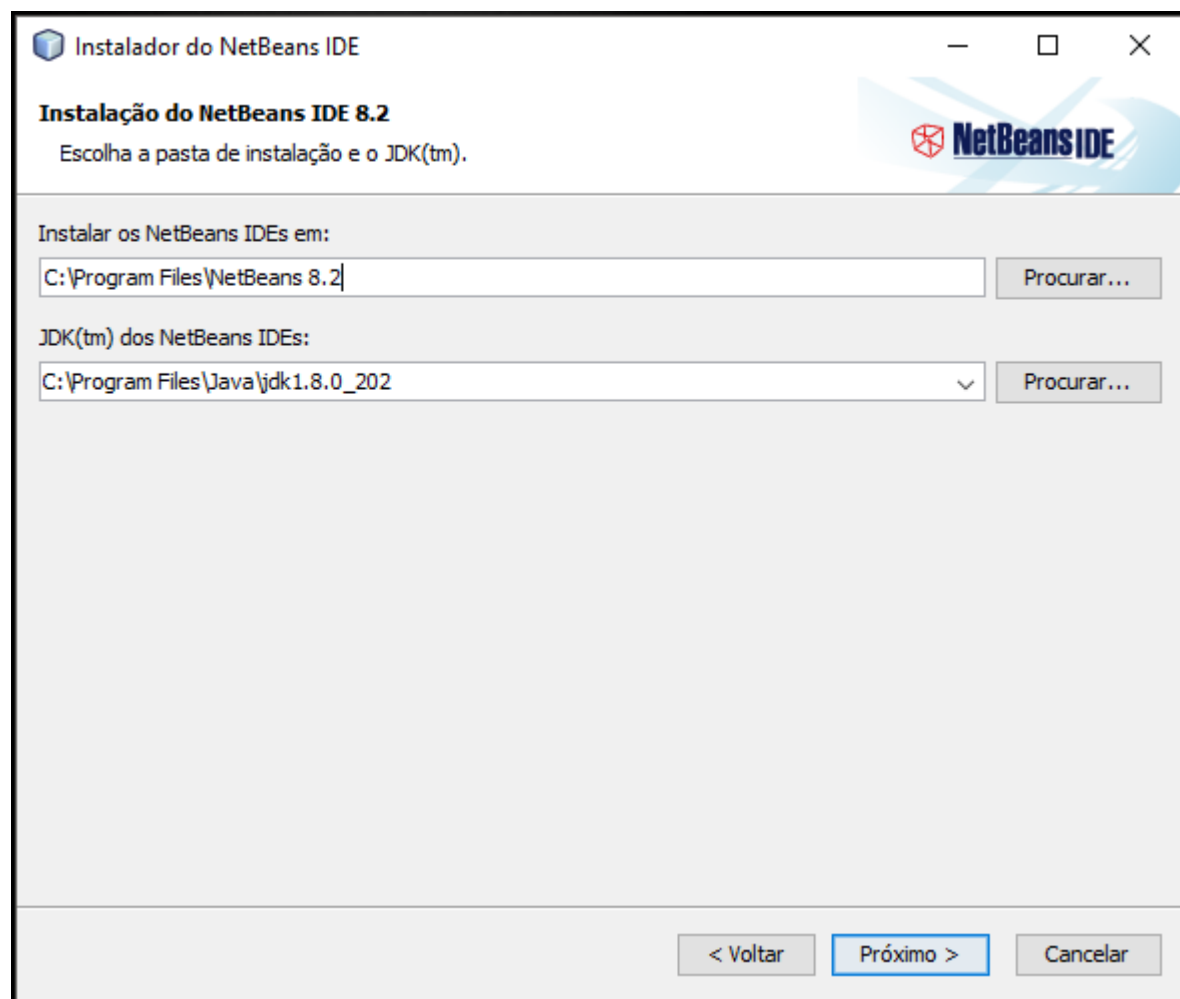
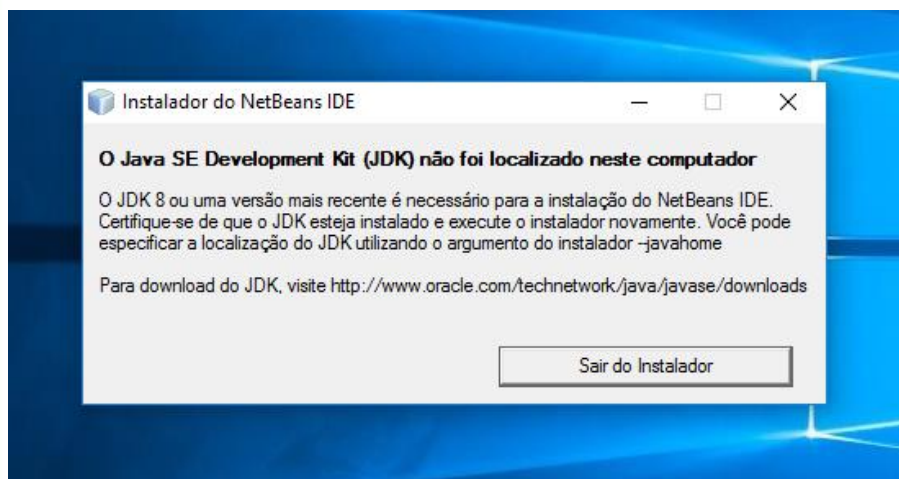


Caso tenha problema com a JAVA\_HOME do Windows vá em: Paine de Controle > Propriedades do Sistema > Variáveis de Ambiente e altere a Path para o diretório de instalação da sua JDK

## Instalando o Netbeans no Windows



## Instalando o Netbeans no Windows



## Preparando no Linux





## Instalando a JDK

Por padrão a versão instalada é a OpenJDK

```
liveuser@localhost:/home/liveuser/Downloads
File Edit View Search Terminal Help
[root@localhost Downloads]# java -version
openjdk version "1.8.0_91"
OpenJDK Runtime Environment (build 1.8.0_91-b14)
OpenJDK 64-Bit Server VM (build 25.91-b14, mixed mode)
[root@localhost Downloads]#
```





## Instalando a JDK

### Instalando o pacote .rpm

```
liveuser@localhost:~/Downloads
File Edit View Search Terminal Help
[liveuser@localhost Downloads]$ ls
jdk-8u101-linux-x64.rpm
[liveuser@localhost Downloads]$ sudo rpm -Uvh jdk-8u101-linux-x64.rpm

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

Preparing...                               ##### [100%]
Updating / installing...
 1:jdk1.8.0_101-2000:1.8.0_101-fcs ##### [100%]
Unpacking JAR files...
  tools.jar...
  plugin.jar...
  javaws.jar...
  deploy.jar...
  rt.jar...
```

-Uvh = desinstala a última versão e atualiza o pacote se necessário exibindo o log



## Instalando a JDK

Configure o alternatives para “java” e “javac”

Java = alternatives --install /usr/bin/java java /usr/java/jdk1.8.0\_101/jre/bin/java 200000

Javac = alternatives --install /usr/bin/javac javac /usr/java/jdk1.8.0\_101/bin/javac 200000



liveuser@localhost:/home/liveuser/Downloads

```
File Edit View Search Terminal Help
[root@localhost Downloads]# alternatives --config java

There are 2 programs which provide 'java'.

  Selection    Command
  -----
*+ 1           /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.91-7.b14.fc24.x86_64/jre/bin/java
   2           /usr/java/jdk1.8.0_101/jre/bin/java

Enter to keep the current selection[+], or type selection number: █
```

## Instalando a JDK

## Configure a JAVA\_HOME

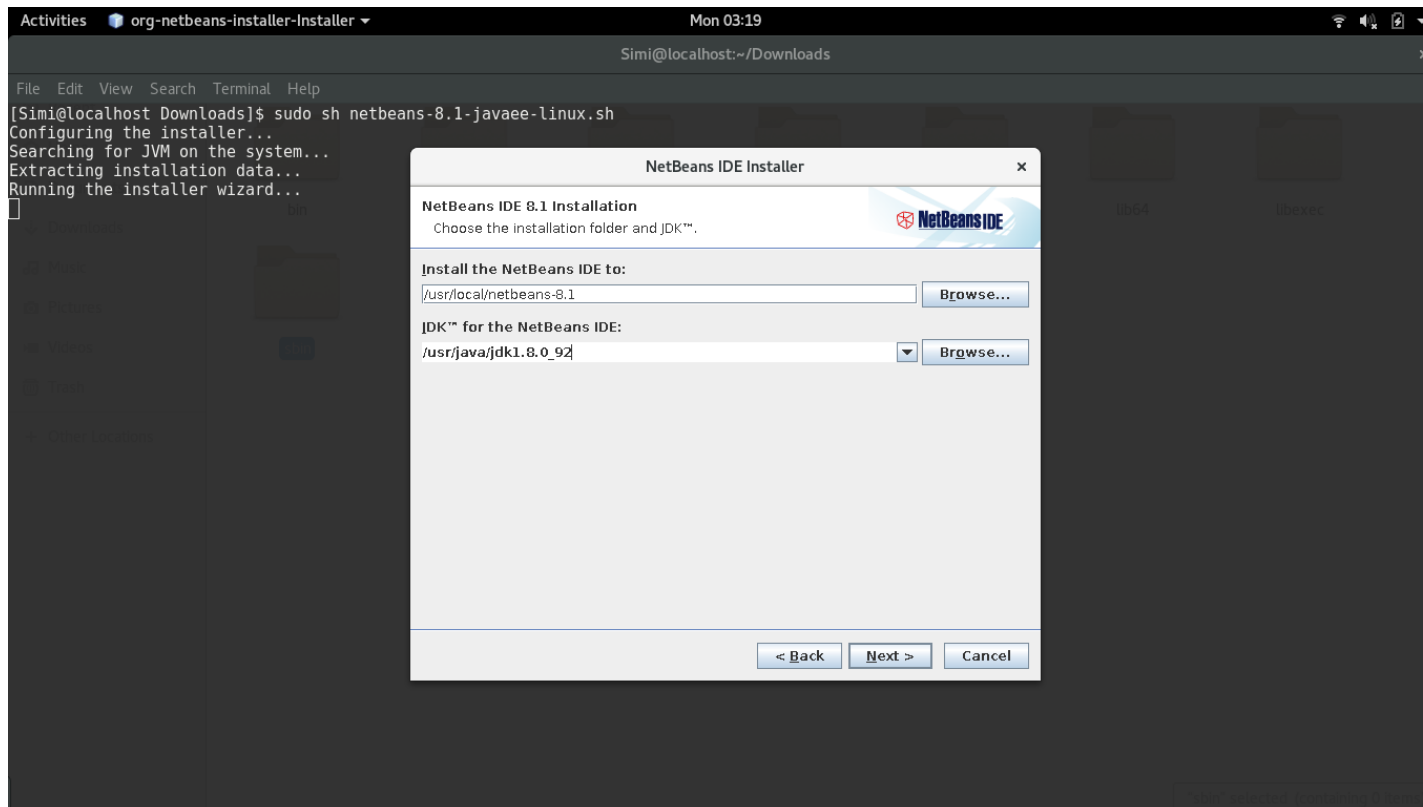


```
Activities Terminal Mon 03:59
Simi@localhost: /usr/java/jre1.8.0_92/bin

File Edit View Search Terminal Help
[Simi@localhost bin]$ cat >> ~/.bashrc <<'EOF'
> export JAVA_HOME=/usr/java/jdk1.8.0_92
> export JAVA_HOME
> export PATH=$PATH:$JAVA_HOME/bin
> EOF
[Simi@localhost bin]$ echo $JAVA_HOME
/usr/java/jdk1.8.0_92
[Simi@localhost bin]$ fig java
alternatives --config javac
export JAVA_HOME=/usr/java/jdk1.8.0_92
```

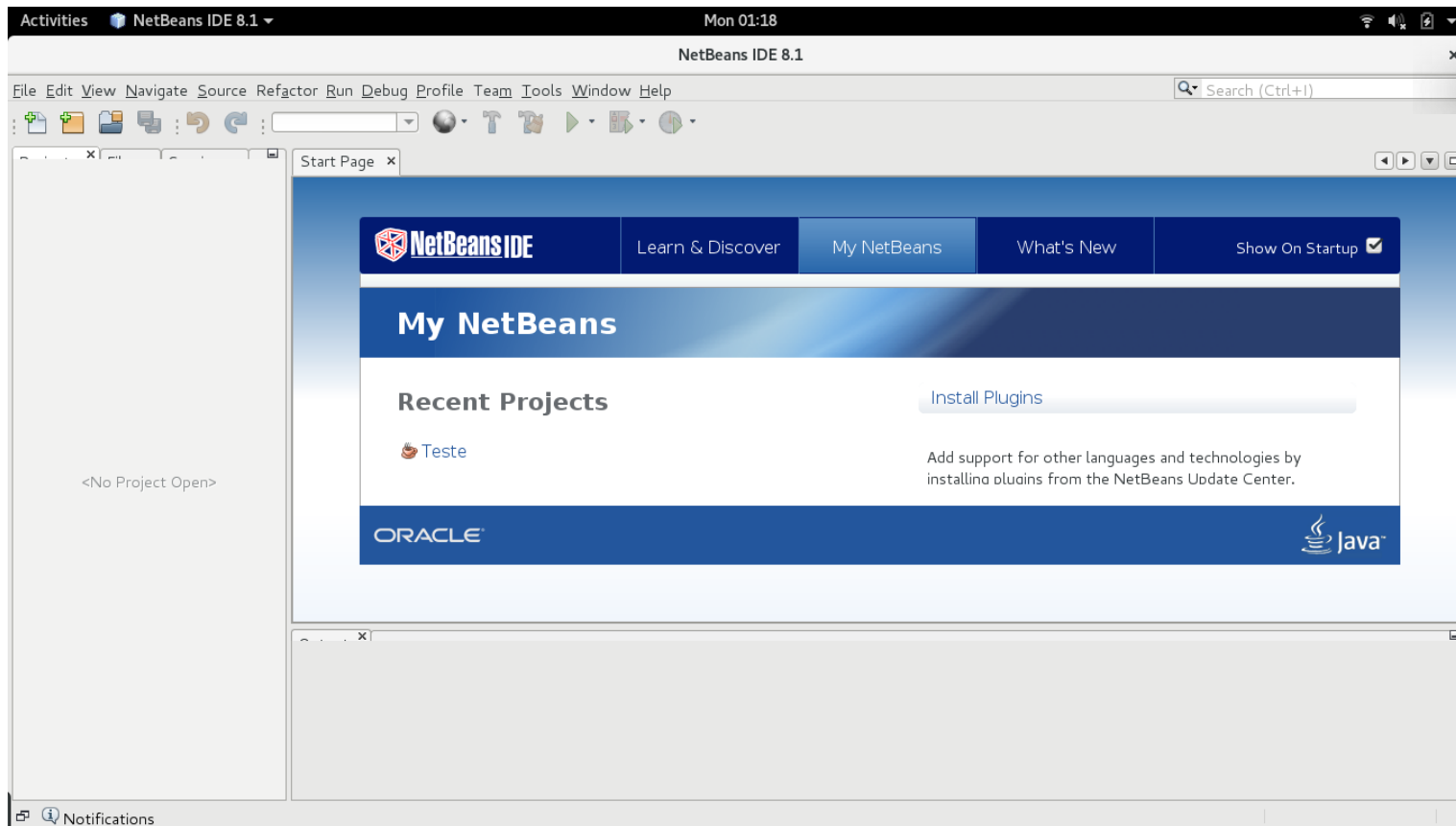
## Instalando o Netbeans no Linux

Execute o script de instalação  
(atenção com as permissões “chmod”)



## Instalando o Netbeans no Linux

Agora é só executar!

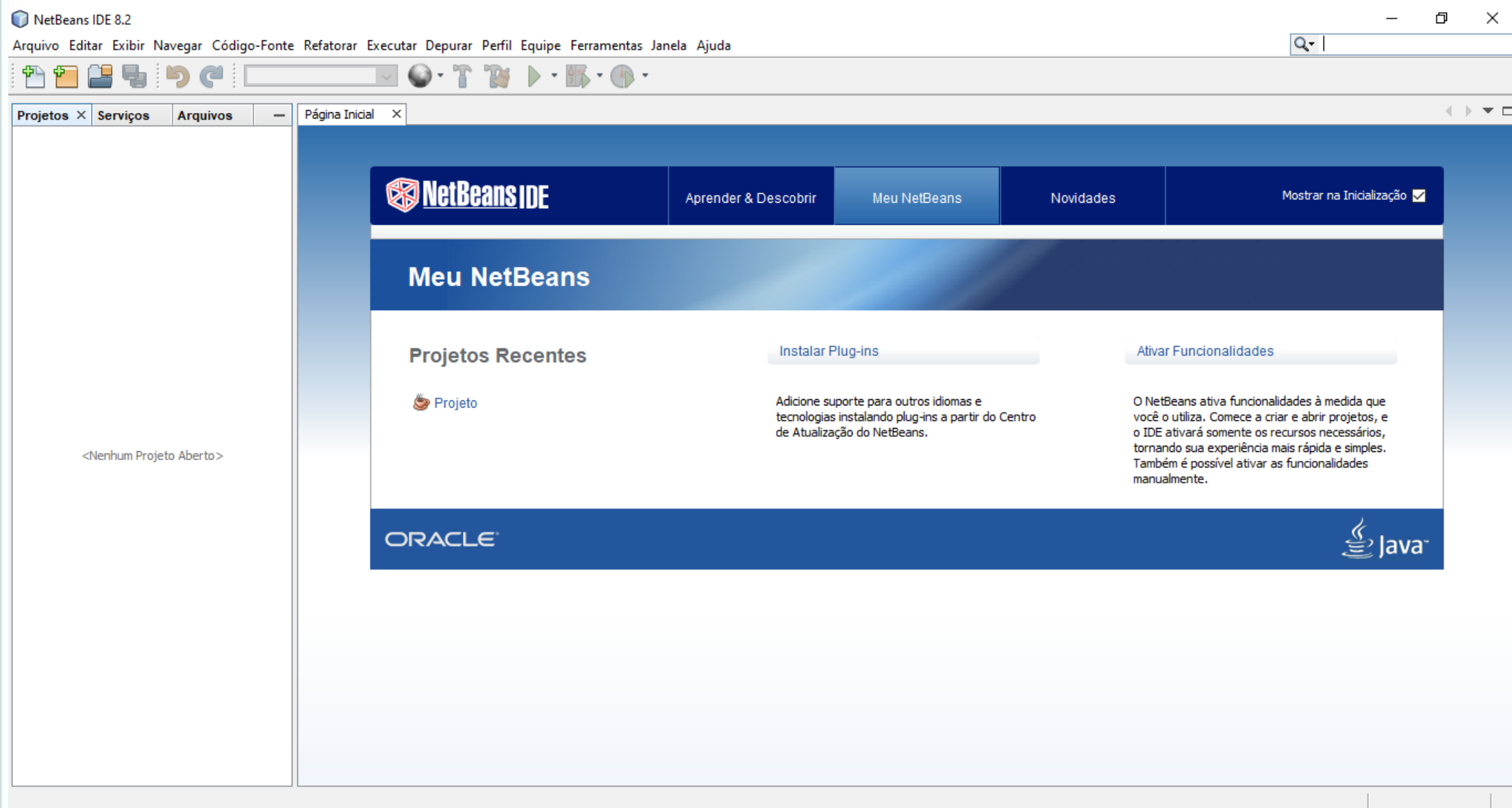


## Conhecendo o Netbeans



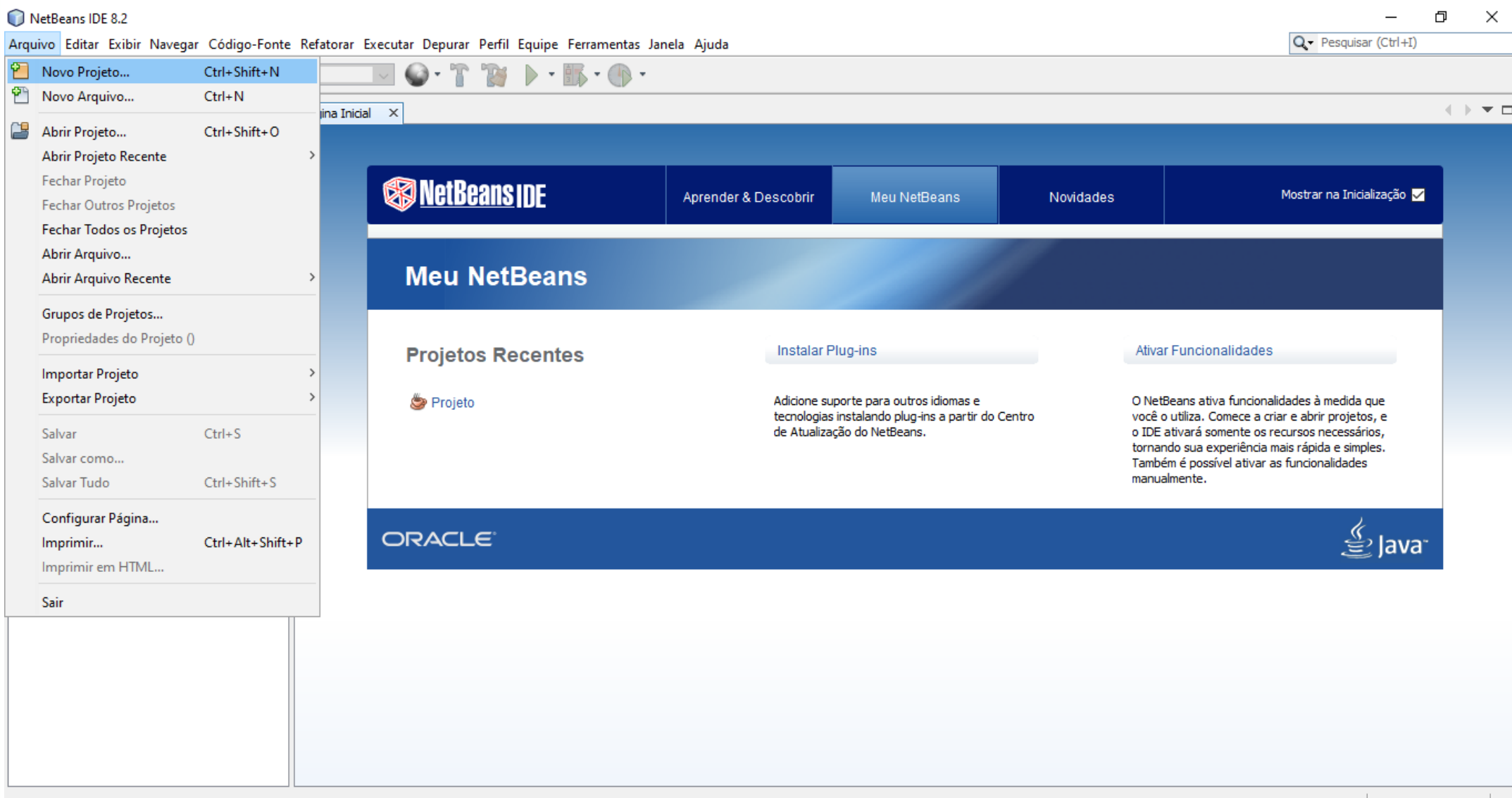
# NetBeans

## Conhecendo o Netbeans



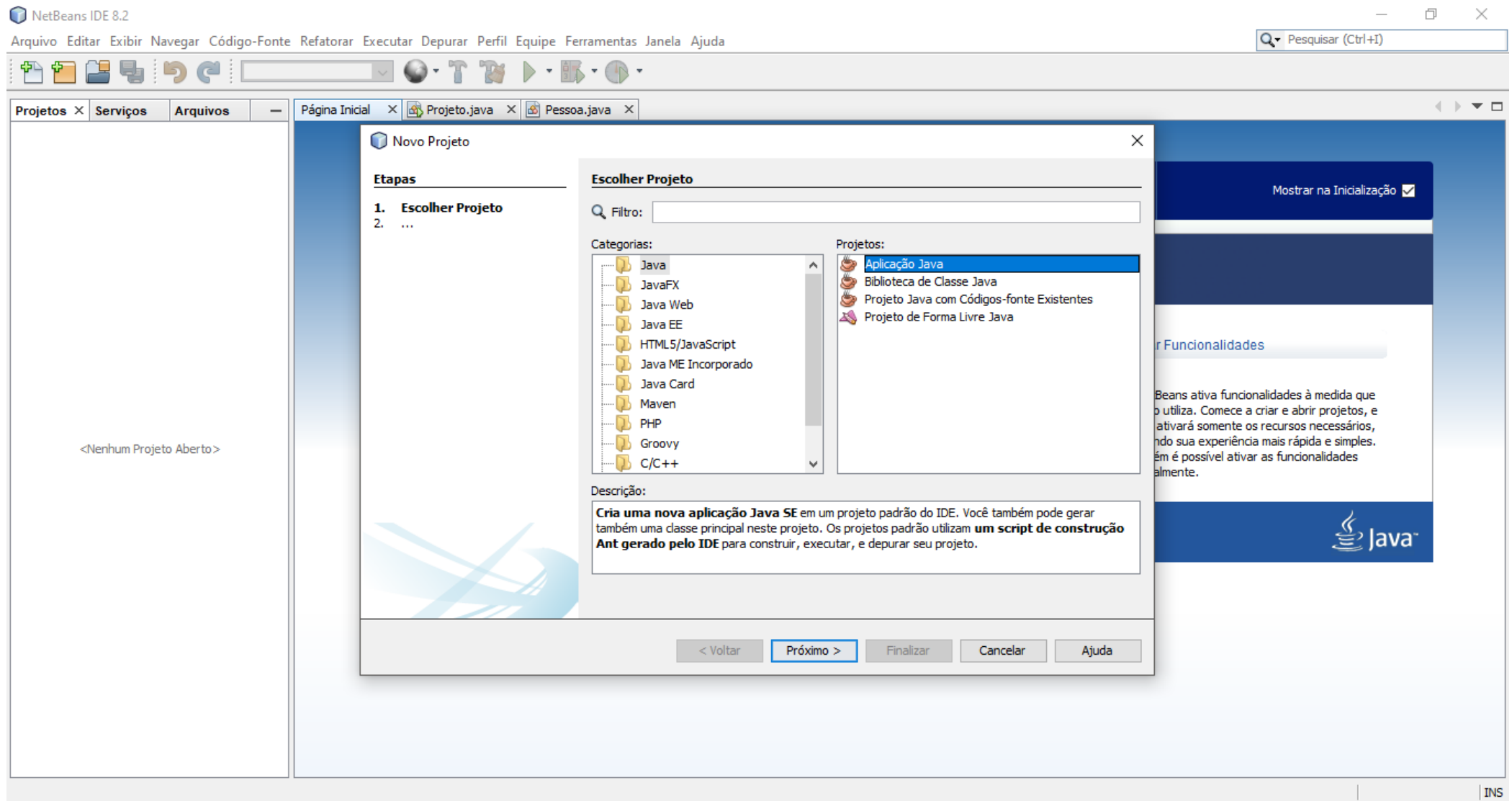


### Criando um novo projeto Java SE





### Criando um novo projeto Java SE



## Criando um novo projeto Java SE

NetBeans IDE 8.2

Arquivo Editar Exibir Navegar Código-Fonte Refatorar Executar Depurar Perfil Equipe Ferramentas Janela Ajuda

Projetos x Serviços Arquivos

Página Inicial x Projeto.java x Pessoa.java x

**Nome e Localização**

Nome do Projeto: HelloWorld

Localização do Projeto: C:\Users\Samuel\Documents\NetBeansProjects Procurar...

Pasta do Projeto: C:\Users\Samuel\Documents\NetBeansProjects\HelloWorld

☐ Usar Pasta Dedicada para Armazenar Bibliotecas

Pasta Bibliotecas: Procurar...

Usuários e projetos diferentes podem compartilhar as mesmas bibliotecas de compilação (consulte a Ajuda para obter detalhes).

☒ Criar Classe Principal helloworld.HelloWorld

< Voltar Próximo > Finalizar Cancelar Ajuda

Nome do projeto

Marcar esta opção

Mostrar na Inicialização ☒

Funcionalidades

NetBeans ativa funcionalidades à medida que o utiliza. Comece a criar e abrir projetos, e ativará somente os recursos necessários, dando sua experiência mais rápida e simples. Também é possível ativar as funcionalidades manualmente.

Java

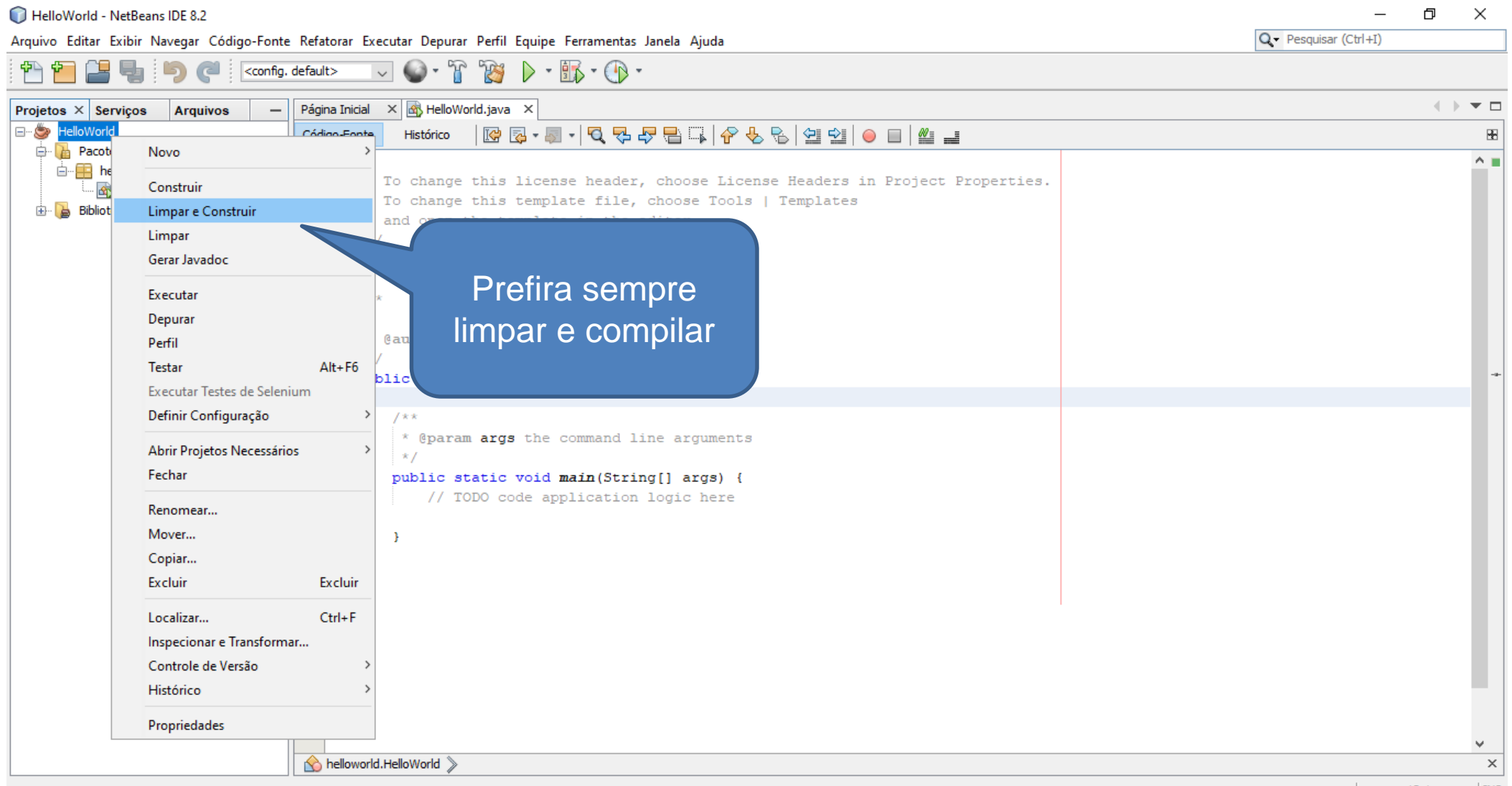
### Criando um novo projeto Java SE

NetBeans IDE 8.2 interface showing a new Java SE project named "HelloWorld". The main editor window displays the "HelloWorld.java" file with the following code structure:

```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package helloworld;
7
8  /**
9   *
10  * @author Samuel
11  */
12  public class HelloWorld {
13
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18          // TODO code application logic here
19      }
20  }
21
22
23
```

A blue callout bubble points to the main method, containing the text: "Escreva o código aqui !!!"

## Compilando o projeto Java SE



## Compilando o projeto Java SE

Para executar, clique aqui ou aperte F6

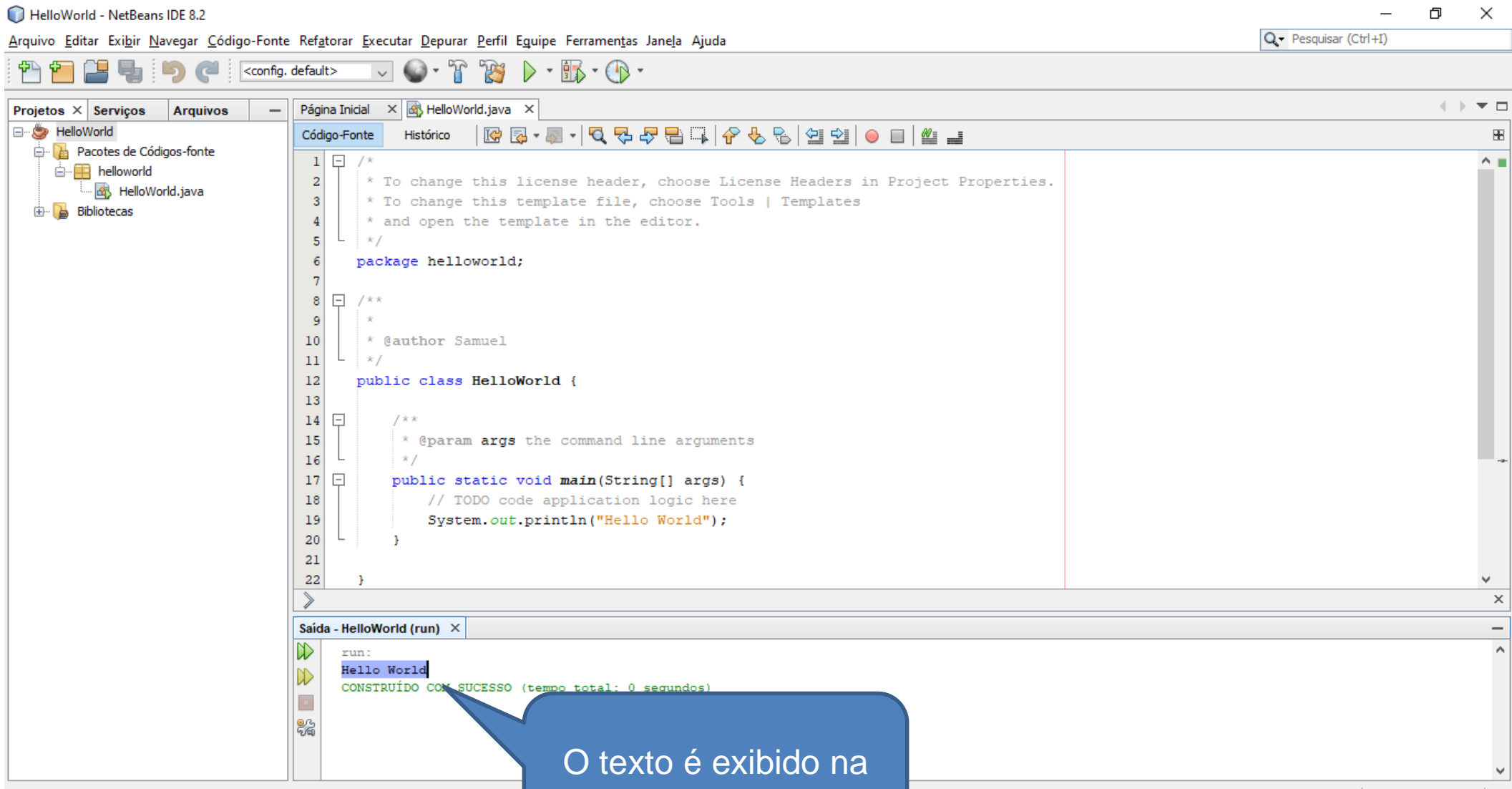
Executar Projeto (HelloWorld) (F6)

```
11  /*
12  public class HelloWorld {
13
14
15      /**
16       * @param args the command line arguments
17       */
18      public static void main(String[] args) {
19          // TODO code application logic here
20          System.out.println(" Hello World ");
21      }
22  }
23
```

helloworld.HelloWorld > main >

19:45 | INS

## Compilando o projeto Java SE



### Utilizando a saída padrão no Java

NetBeans IDE 8.2 interface showing the HelloWorld project. The main editor displays the source code for HelloWorld.java, which includes package, class, and main method declarations. The main method uses `System.out.println` and `System.out.print` to demonstrate standard output.

```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package helloworld;
7
8  /**
9   *
10  * @author Samuel
11  */
12  public class HelloWorld {
13
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18          // TODO code application logic here
19          System.out.println("Imprime o texto e move o cursor para a próxima linha");
20          System.out.print("Imprime o texto e mantém o cursor na mesma linha");
21      }
22  }
```

The bottom pane shows the output of the program, confirming successful execution:

```
run:
Imprime o texto e move o cursor para a próxima linha
Imprime o texto e mantém o cursor na mesma linhaCONSTRUÍDO COM SUCESSO (tempo total: 0 segundos)
```

Construção HelloWorld (run) finalizada.

## Fazendo a mão...

- Crie uma pasta na Área de Trabalho;
- Abra o Notepad (Bloco de Notas);
- Escreva o seguinte programa;

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World");  
    }  
}
```

- Salve o arquivo dentro da pasta criada como HelloWorld.java



## Comandos javac e java (sem pacote)

```
C:\Users\Samuel\Desktop\Aula01>javac -version
javac 1.8.0_202

C:\Users\Samuel\Desktop\Aula01>dir
O volume na unidade C é OS
O Número de Série do Volume é 7819-4E4A

Pasta de C:\Users\Samuel\Desktop\Aula01

31/01/2019  19:34    <DIR>        .
31/01/2019  19:34    <DIR>        ..
31/01/2019  19:33                121 HelloWorld.java
                1 arquivo(s)                121 bytes
                2 pasta(s)  792.846.712.832 bytes disponíveis

C:\Users\Samuel\Desktop\Aula01>javac HelloWorld.java

C:\Users\Samuel\Desktop\Aula01>dir
O volume na unidade C é OS
O Número de Série do Volume é 7819-4E4A

Pasta de C:\Users\Samuel\Desktop\Aula01

31/01/2019  19:36    <DIR>        .
31/01/2019  19:36    <DIR>        ..
31/01/2019  19:36                426 HelloWorld.class
31/01/2019  19:33                121 HelloWorld.java
                2 arquivo(s)                547 bytes
                2 pasta(s)  792.846.712.832 bytes disponíveis

C:\Users\Samuel\Desktop\Aula01>java HelloWorld
Hello, World

C:\Users\Samuel\Desktop\Aula01>
```

## Curiosidade

O que significa `public static void main(String[] args)`?

### `public`

É o modificador de acesso do método. Usando este modificador o método pode ser acessado por qualquer classe dentro (e fora) do projeto.

Outros modificadores são `protected`, `private` ou **sem modificador\***. [Aqui](#) pode-se ler mais sobre os modificadores de acesso do Java.

### `static`

Define o método como **estático**, isso quer dizer que a classe não precisa ser instanciada para chamar este método.

No exemplo, tenho a classe `Cliente` com os métodos (estático) `FazerAlgo()` e (não-estático) `FazerAlgoDois()`, o uso seria assim:

```
Cliente cliente = new Cliente();  
cliente.FazerAlgoDois(); // Este é o método não-estático  
  
Cliente.FazerAlgo(); // Este é o método estático
```

## Curiosidade

### `void`

É tipo de retorno do método. Este tipo de retorno significa vazio/nada, o método não dá retorno nenhum. Os métodos podem retornar qualquer tipo do seu projeto, até mesmo os criados por você.

### `main`

É o nome do método. Todo e qualquer método precisa ter um nome. Os nomes são definidos pelo programador e geralmente seguem algum padrão convencional definido previamente pela linguagem ou pela comunidade, embora isso seja opcional. [No caso do Java, a própria Oracle define estas convenções](#). A convenção referente a nomeação de métodos diz:

Methods should be verbs, in mixed case with the first letter lowercase, with the first letter of each internal word capitalized.

Em tradução livre:

Métodos devem ser verbos, em "mixed case" com a primeira letra minúscula e a primeira letra das palavras internas em letra maiúscula.

No Java (e em outras linguagens também) o `main` é o ponto de entrada da aplicação. É o método que a JRE procura para executar a aplicação. Por isso, em alguns tipos de aplicação (como Swing ou console) é obrigatório tê-lo implementado. É possível ver mais detalhes sobre isso em [Por que é obrigatório implementar "public static void main \(String \[\] args\)"?](#)

## Curiosidade

`(String[] args)`

Define que o método deve receber como parâmetro um array de `String` (nomeado `args`). Nesse caso específico: este parâmetro serve para caso seu programa precise receber algum valor como argumento, isso é muito comum quando o programa é iniciado por outro programa ou pelo terminal (CMD, Shell, Bash, etc.).

Um exemplo muito comum é o `Git`. Quando você digita `git commit` no seu terminal está chamando o `Git` com o parâmetro `commit`. Todas as "strings" que vierem depois do nome do programa serão recebidas pelo mesmo dentro do array (`args`). Geralmente a primeira posição do array é o caminho que a aplicação se encontra.

Resposta retirada do StackOverflow

<https://pt.stackoverflow.com/questions/93048/o-que-significa-public-static-void-mainstring-args>

## Conhecendo JavaDoc para esclarecer duvidas

The screenshot shows the Oracle Java Platform Standard Edition 8 API Specification page. The browser address bar displays the URL <https://docs.oracle.com/javase/8/docs/api/>. The page has a dark blue header with navigation links: OVERVIEW, PACKAGE, CLASS, USE, TREE, DEPRECATED, INDEX, and HELP. Below the header, there are links for PREVIOUS and NEXT, and FRAMES and NO FRAMES. The main content area is titled "Java™ Platform, Standard Edition 8 API Specification" and includes a description: "This document is the API specification for the Java™ Platform, Standard Edition. See: Description". A "Profiles" section lists compact1, compact2, and compact3. A "Packages" section is highlighted, showing a table with two columns: Package and Description. The table lists several packages including java.applet, java.awt, java.awt.color, java.awt.datatransfer, and java.awt.dnd, each with a brief description of its contents.

Java™ Platform Standard Ed. 8

OVERVIEW PACKAGE CLASS USE TREE DEPRECATED INDEX HELP

PREV NEXT FRAMES NO FRAMES

### Java™ Platform, Standard Edition 8 API Specification

This document is the API specification for the Java™ Platform, Standard Edition.  
See: Description

#### Profiles

- compact1
- compact2
- compact3

#### Packages

Package	Description
<b>java.applet</b>	Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context.
<b>java.awt</b>	Contains all of the classes for creating user interfaces and for painting graphics and images.
<b>java.awt.color</b>	Provides classes for color spaces.
<b>java.awt.datatransfer</b>	Provides interfaces and classes for transferring data between and within applications.
<b>java.awt.dnd</b>	Drag and Drop is a direct manipulation gesture found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in the GUI.

<https://docs.oracle.com/javase/8/docs/api/>

## Exercícios

1. Instale o Netbeans
2. Crie um projeto chamado “PrimeiraAula”
3. Faça um programa que imprima a seguinte mensagem:  
“Minha primeira aula de POO com Java”
4. Agora altere sua aplicação para que ela imprima a seguinte mensagem:  
“Minha primeira aula de POO com Java.  
Estamos usando o Netbeans!”
5. **Desafio:** imprima a mesma mensagem do exercício 4 utilizando três comandos de impressão.

**Obrigado!**