

[< VOLTAR](#)

# Introdução ao Java

O Java é uma das linguagens orientada a objetos mais usadas no mundo, pelas maiores empresas. Esta aula apresenta a você uma breve introdução ao Java, sua história e motivo de sua singular importância na atualidade. Você também aprenderá aqui sobre como a linguagem evolui ao longo do tempo. Você terminará essa aula com o seu primeiro programa em Java funcionando: O clássico Hello World! (Olá mundo). Bem-vindo(a) ao maravilhoso mundo orientado a objetos do Java!

## NESTE TÓPICO

- > Breve histórico da linguagem
- > Por que o Java é tão popular e multiplataforma
- > Algumas coisas importantes que você precisa saber sobre Java
- > Versionamento do Java
- > Extensões dos arquivos Java
- > Java e Javascript, é a mesma coisa?



## Breve histórico da linguagem



Hoje o Java é uma linguagem de programação que pertence a Oracle, mas nem sempre foi assim. O Java é uma linguagem que se originou em 1991 em um projeto da *Sun Microsystems*.

Tudo começou com James Gosling e um projeto nomeado de *Projeto Green* que visava a criação de tecnologias que pudessem ser vendidas para empresas que fabricavam equipamentos eletrônicos, afim de que seus dispositivos pudessem se comunicar, mesmo sendo diferentes. Essa é a grande sacada do Java: Pode rodar em praticamente qualquer dispositivo, pois não depende da plataforma. O *Projeto Green* criou o \*7 (*Star Seven*), um controle remoto universal com uma tela bastante interativa. O software deste controle foi desenvolvido pela linguagem de programação nomeada de Oak.

Mas foi somente em Maio 1995 que surgiu, de fato, o Java, como uma evolução da linguagem Oak. Ela teve ser renomeada por questões legais na época e a equipe escolheu Java que, segundo Deitel, originou-se em uma cafeteria que servia um café que era originado em Java, a cidade, e o nome pegou. É por isso que o ícone do Java é uma xícara de café



Este é o logotipo atual do Java: Uma xícara de café

Fonte: Gil C / Shutterstock.com

O anúncio foi feito pela *Sun Microsystems* em uma conferência e chamou bastante atenção da área de negócios pois, nesta época, a internet estava começando a ser rentável. O Java surgiu já com a ideia de que se poderia desenvolver aplicativos para diversos equipamentos, com o mesmo código fonte.

Em 2009 a empresa Oracle (na época, muito conhecida pelo seu excelente sistema gerenciador de bancos de dados) adquiriu a *Sun Microsystems* por US\$ 7,4 milhões, tornando-se uma das maiores empresas de tecnologia do mundo.



## Por que o Java é tão popular e multiplataforma

A linguagem de programação Java é uma das mais populares do mundo e sua portabilidade é um de seus pontos mais fortes para que isso tenha acontecido. Portabilidade é a capacidade de se portar um software para diferentes plataformas, com o mínimo de esforço, ou seja, Java é **multiplataforma**.

Diferentemente da maior parte das linguagens de programação que precisam de um compilador específico para cada plataforma, o Java possui um compilador único, ou seja, o Java não é uma linguagem que depende de uma plataforma específica para sua execução.

Os binários compilados da linguagem são executados em uma máquina virtual Java (**JVM** - *Java Virtual Machine*), por isso não depende da plataforma, ou seja, o mesmo código Java pode ser executado em

praticamente qualquer dispositivo com uma JVM. Existem JVMs para quase todas as plataformas atualmente, desde controles remotos e microondas até computadores quânticos.

Java é uma linguagem proprietária da Oracle, mas de uso livre e gratuito, ou seja, você pode criar qualquer tipo de aplicação com Java, desde aplicações de código aberto (*OpenSource*) a aplicações comerciais, mas não temos acesso ao código-fonte do compilador do Java.

Há quem diga que o Java é uma linguagem lenta, justamente por não ser executada diretamente no Sistema Operacional (as instruções do Sistema Operacional são geradas pela JVM), mas a tecnologia Java evoluiu tanto que o processo de execução é quase transparente. Mesmo assim, Java é uma das linguagens mais populares do mundo.

A tabela abaixo ilustra o ranking de utilização das linguagens de programação em 2015, segundo o IEEE. Veja que o Java ficou em primeiro lugar.

Posição #	Linguagem de Programação
1	Java
2	C
3	C++
4	Python
5	C#
6	R
7	PHP
8	JavaScript
9	Ruby
10	Matlab

Ranking das linguagens de programação em  
2015

Isso somente nos leva a perceber a popularidade do Java. A linguagem é utilizada, hoje, pela maior parte das empresas de desenvolvimento e, por isso, merece essa especial atenção.



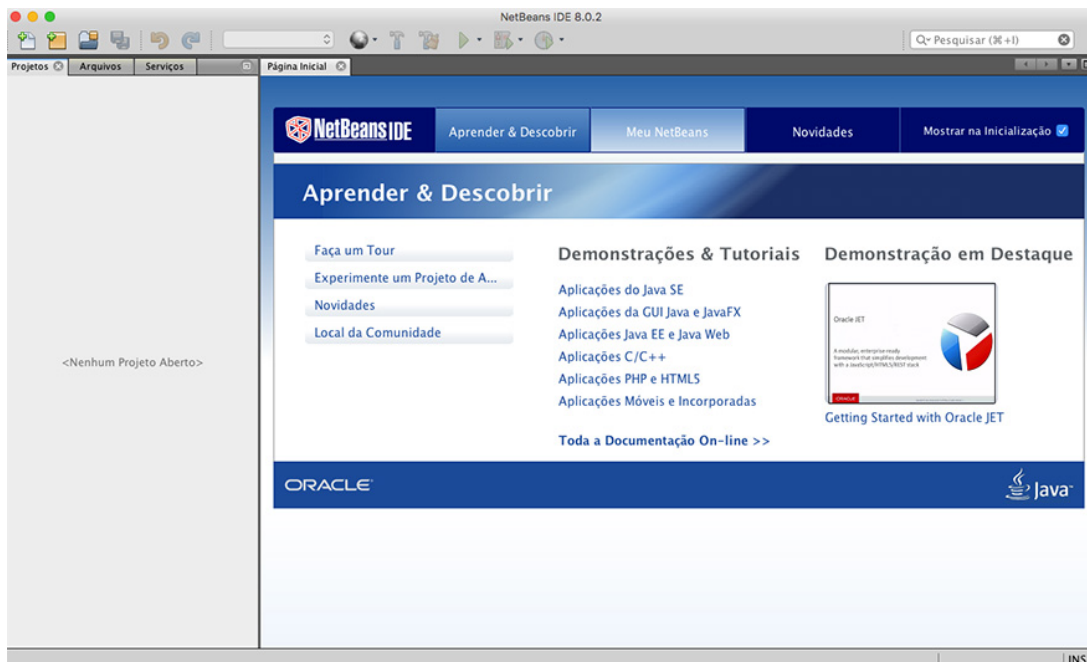
# Algumas coisas importantes que você precisa saber sobre o Java

É muito importante nos ambientarmos com as nomenclaturas muito usadas no mundo Java e suas versões. Vamos ver algumas coisas importantes que todo programador Java deve conhecer muito bem (Teruel, 2015):

- **JRE - Java Runtime Environment (ambiente de execução Java):** A JRE contém as bibliotecas que são responsáveis pela execução das aplicações na JVM. A JRE é composta pela JVM, bibliotecas e componentes necessários para essa execução;
- **JVM - Java Virtual Machine (máquina virtual Java):** Como mencionado acima, a JVM é responsável por executar instruções compiladas em Java. A JVM é um ambiente de computação virtualizado (abstrato) que executa e gerencia os processos Java, a alocação de memória e recursos de CPU etc. É a JVM quem abstrai a execução do código compilado para o equipamento onde será executada a aplicação;
- **JDK - Java Development Kit (kit de desenvolvimento Java):** É o Kit de Desenvolvimento Java que será instalado para desenvolver aplicações em Java, pois o JDK possui todas as ferramentas necessárias para compilação de código Java para a JRE. A JDK possui bibliotecas Java para diferentes tipos de desenvolvimento. A JDK é instalada automaticamente com a versão do Java que você baixar:
  - **JSE - Java Standard Edition (edição padrão do Java):** Essa é a edição essencial do Java e já contém a JDK, JVM e JRE. Trabalharemos com essa edição do Java, que é capaz de criar qualquer aplicação para ser executada em um computador;
  - **JEE - Java Enterprise Edition (edição corporativa do Java):** Nesta edição do Java você pode desenvolver aplicações corporativas de grande porte e que rodam em rede, pois aqui você encontrará bibliotecas para *Web Services*, desenvolvimento WEB, JSP (*Java Server Pages*), JSF (*Java Server Faces*), *JavaBeans* etc.
  - **JME - Java Micro Edition (edição micro do Java):** Esta edição do Java acompanha bibliotecas que permitem o desenvolvimento para sistemas embarcados que possuem uma JVM, como carros, eletrodomésticos, celulares etc., mas não confunda com desenvolvimento para Android ou iOS (Apple), pois estes sistemas operacionais existem ferramentas de desenvolvimento específicas de cada fabricante, por exemplo, para Android a Google disponibiliza a ADT - *Android Development Tools* (ferramentas de desenvolvimento Android) - que possui suas próprias bibliotecas para integração com o Java. A Google possui, ainda, um ambiente próprio para desenvolvimento Android chamado *Android Studio* que utiliza Java.
- **IDE Integrated Development Environment (ambiente de desenvolvimento integrado):** É na IDE onde iremos trabalhar, pois ela lhe fornecerá um enorme conjunto de ferramentas de desenvolvimento que aumentam sua produtividade no processo de desenvolvimento. A IDE possui ferramentas de edição de código, implementa as bibliotecas, compila, executa, gera executáveis, depura código em tempo real etc. Existem diversas IDEs disponíveis no mercado, como:



- Eclipse - <http://www.eclipse.org> (<http://www.eclipse.org/>)
- Netbeans - <http://netbeans.org> (<http://netbeans.org/>) - Para desenvolvimento Java SE, iremos utilizar o NetBeans, uma das mais poderosas IDEs disponíveis e da Oracle, a mesma proprietária do Java
- BlueJ - <http://www.bluej.org> (<http://www.bluej.org/>)
- JCreator - <http://www.jcreator.com/> (<http://www.jcreator.com/>)
- entre diversas outras



Tela principal do Netbeans IDE

## Versionamento do Java

Como qualquer software, o Java evolui com o tempo, ganhando e perdendo funções, melhorias de estabilidade e desempenho, segurança, melhorias no gerenciamento de memória e threads etc. O versionamento da linguagem segue um padrão bastante interessante.

O Java está sempre na versão 1! É isso mesmo, desde 1995 o Java está na versão 1 e as evoluções da linguagem são mencionadas na subversão. Para falar, usa-se normalmente a subseção do Java, por exemplo, se você tem instalada a versão 1.8, então você tem a versão 8 do Java.

Mas há, ainda, uma subversão dentro dessas versões, recém implantada pela Oracle. Em 2013, a Oracle criou um novo esquema de versionamento do Java para que possamos identificar quando estamos trabalhando com uma versão que corrige CPU (Critical Patch Update - atualização crítica) e quando corrige LFR (Limited Feature Release - atualização de funcionalidade). Também serve para facilitar o planejamento e desenvolvimento de correções para futuras versões (Oracle):

- **CPU** - Critical Patch Update: As atualizações criadas com correções de segurança importantes. A versão será sempre ímpar ou múltipla de 5, por exemplo 7u40, 7u52, 6u55.
- **LFR** - Limited Feature Release: As atualizações com correções de performance, funcionalidades e recursos. Aqui as versões são múltiplas de 20 e com final 0, por exemplo 7u60, 7u80, 8u20.

O importante é manter seu Java atualizado! Lembre-se que algumas atualizações retiram ou substituem métodos prontos que usamos. A grande vantagem de usarmos uma IDE para desenvolvimento Java é que ela lhe avisará se as funções que você está utilizando já estão obsoletas (já foram retiradas) ou se ficarão obsoletas na próxima versão, sugerindo outra forma de chamar algum método.

### DICA IMPORTANTE

Para saber qual versão do Java você tem instalado, entre no terminal (prompt de comando) de seu sistema operacional e digite: `java -version`

## Extensões dos arquivos Java



Os arquivos de código fonte Java possuem extensão .java, enquanto os arquivos compilados do Java (*bytecodes*) possuem extensão .class. Há também algumas extensões importantes que devemos conhecer quando programamos em Java SE.

A extensão .jar: Essa extensão refere-se a um pacote de arquivos .class e, normalmente, possui informações de execução do projeto, ou seja, você pode exportar seu projeto compilado para um único arquivo de extensão jar. Este será o seu executável.

## Java e Javascript, é a mesma coisa?

**Muito cuidado** aqui! Java e JavaScript estão muito longe de ser a mesma coisa. O Javascript é uma linguagem que é executada diretamente em navegadores de internet (*Browsers*) enquanto o Java é uma linguagem de alto nível orientada a objetos que requer compilação e é executada dentro da JVM, como visto aqui.

Então não, Java e JavaScript são linguagens totalmente diferentes e com propósitos diferentes. Uma é exclusiva para web (browser) e a outra, embora possa ser utilizada também para desenvolvimento web, é uma linguagem que é executada no servidor da aplicação web e não no browser.

## Certificações Java

A linguagem cresceu tanto e é tão usada no mundo inteiro, que possui seu próprio conjunto de certificações disponível.

A Oracle possui diversas certificações que você pode obter para trabalhar com Java. Segundo a Oracle, a certificação serve para garantir às empresas que contratam esses profissionais, que eles possuem os conhecimentos necessários para trabalhar com determinada tecnologia.

São diversas as certificações oferecidas pela própria Oracle. Você pode obter mais detalhes de certificações Oracle em: <http://education.oracle.com/> (<http://education.oracle.com/>)

## Preparação do ambiente

Para essa disciplina usaremos o Netbeans como IDE, mas antes de instalá-lo, você precisa preparar a máquina para programação em Java. O que é preciso instalar, nessa ordem, para programar em Java SE:

1. **JDK SE** - Java Development Kit para Standard Edition
2. **Netbeans** - A IDE que usaremos para desenvolvimento Java



A JDK pode ser baixada diretamente no site da Oracle, em:

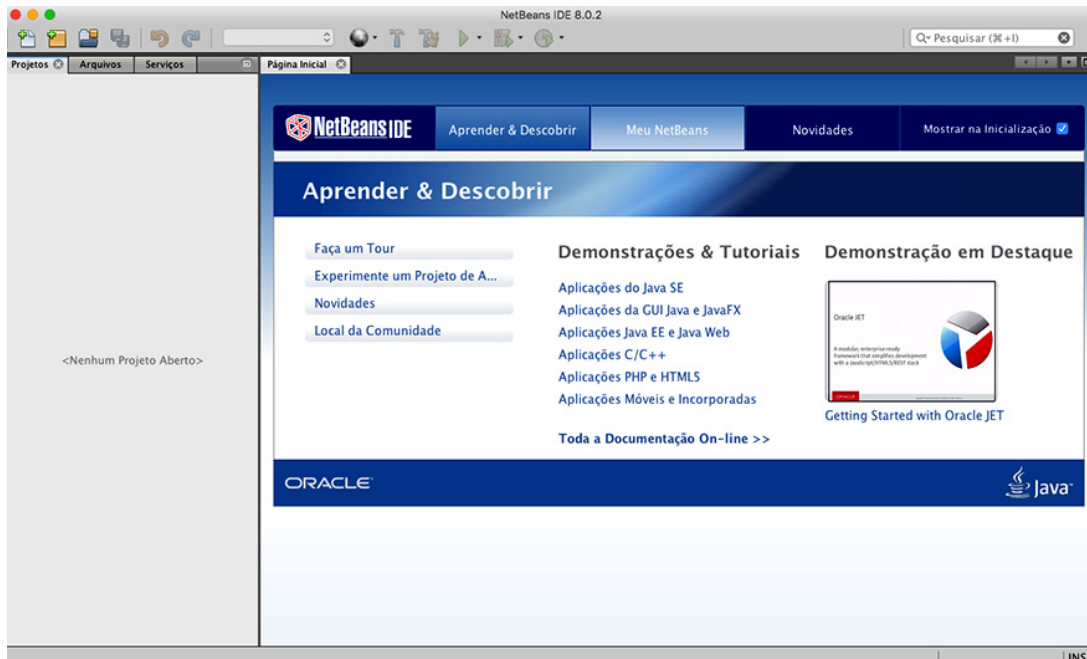
<http://www.oracle.com/technetwork/pt/java/javase/downloads/index.html>  
(<http://www.oracle.com/technetwork/pt/java/javase/downloads/index.html>)

O Netbeans pode ser baixado diretamente de seu site, em [www.netbeans.org](http://www.netbeans.org), embora no site da Oracle, você possa baixar um único instalador para os dois.

Uma vez instalados, seu ambiente está pronto para desenvolvimento Java. Agora sim, a coisa vai começar!

## Meu primeiro programa em Java

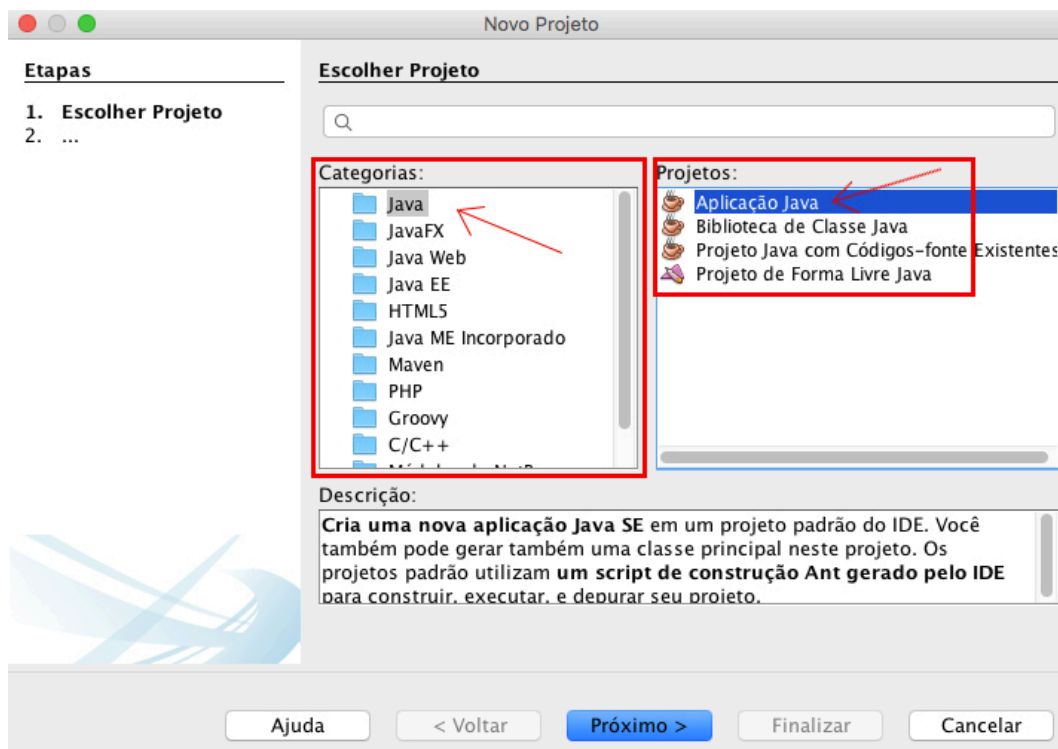
Execute o Netbeans. Na primeira vez que você o abrir, ele estará com essa aparência:



Primeira tela do Netbeans IDE, sendo executado pela primeira vez

É muito importante ambientar-se com o ambiente de programação. Explore todos os cantos do software, intime-se com os menus, veja as teclas de atalho (muito úteis) etc.

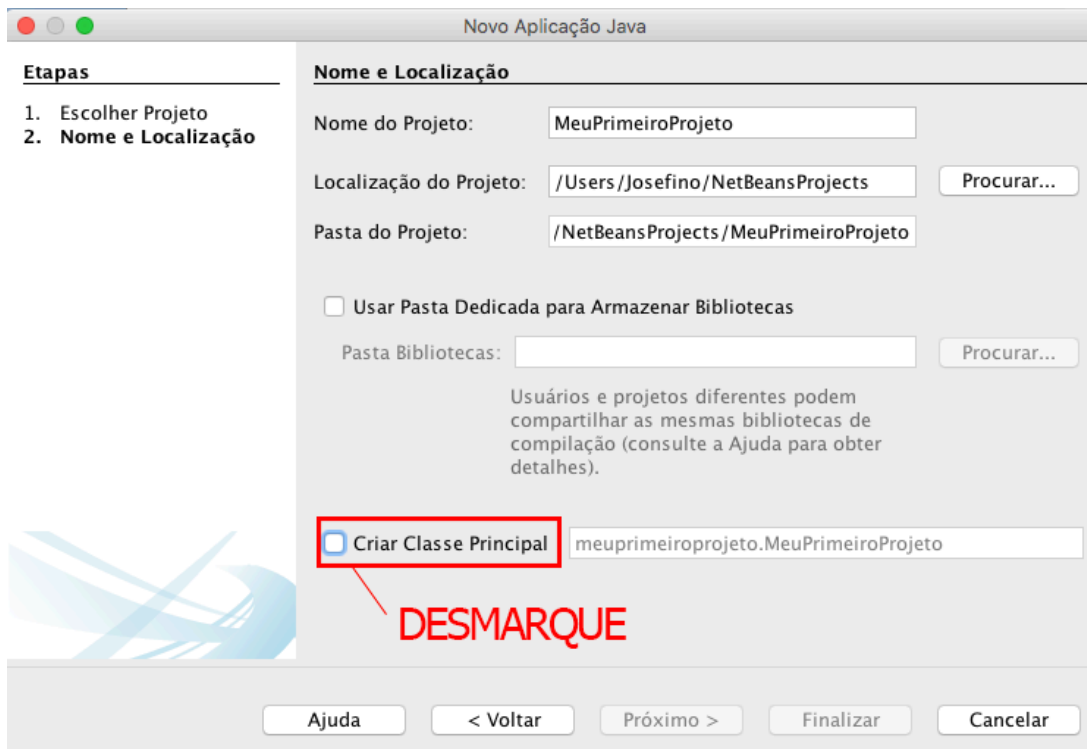
É preciso lembrar que cada aplicação que você for desenvolver é tratada como um projeto. Então crie um novo projeto: Arquivo —> Novo Projeto. Você verá uma tela como a apresentada abaixo:



Criando o projeto Java SE no Netbeans - Primeira etapa



Escolha, em categoria, a opção “Java” e em projetos, Aplicação Java (como mostrado acima). Escolhidas estas opções, clique em “Próximo”. Você verá uma tela como esta:



**Novo Aplicação Java**

**Etapas**

1. Escolher Projeto
2. Nome e Localização

**Nome e Localização**

Nome do Projeto:

Localização do Projeto:

Pasta do Projeto:

☐ Usar Pasta Dedicada para Armazenar Bibliotecas

Pasta Bibliotecas:

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

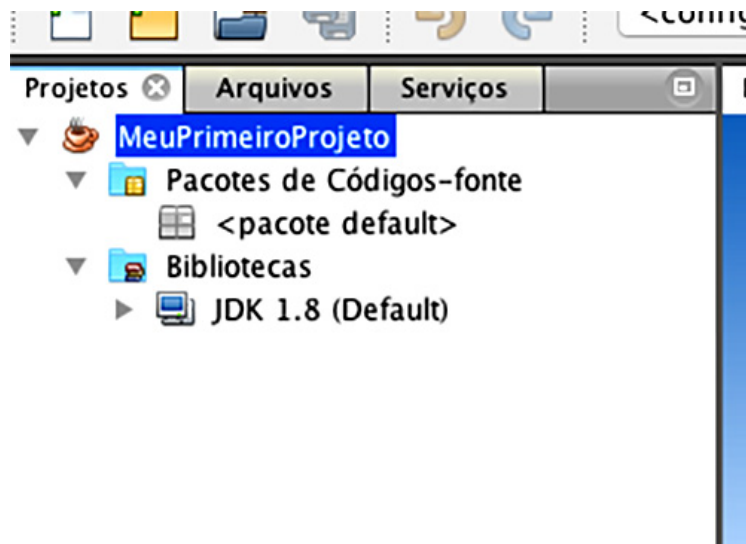
☒ Criar Classe Principal

**DESMARQUE**

Criando o projeto Java SE no Netbeans - segunda etapa

Nessa tela, dê um nome para seu projeto. Não use espaço ou caracteres especiais, defina a localização do seu projeto e desmarque (se estiver marcada) a opção “Criar Classe Principal”. Depois disso, clique em Finalizar. O nome do projeto deve caracteriza-lo, ou seja, evite usar nomes como “projeto1”, “projetoLegal” etc.

Perceba que algo aconteceu em seu NetBeans: O projeto foi criado. Como saber isso? Veja, no quadro superior esquerdo, chamado “Projetos” que ele aparece lá. Expanda as pastas e veja que não há nenhuma classe criada, conforme a imagem abaixo:



Quadro de projetos do Netbeans - Estrutura do projeto

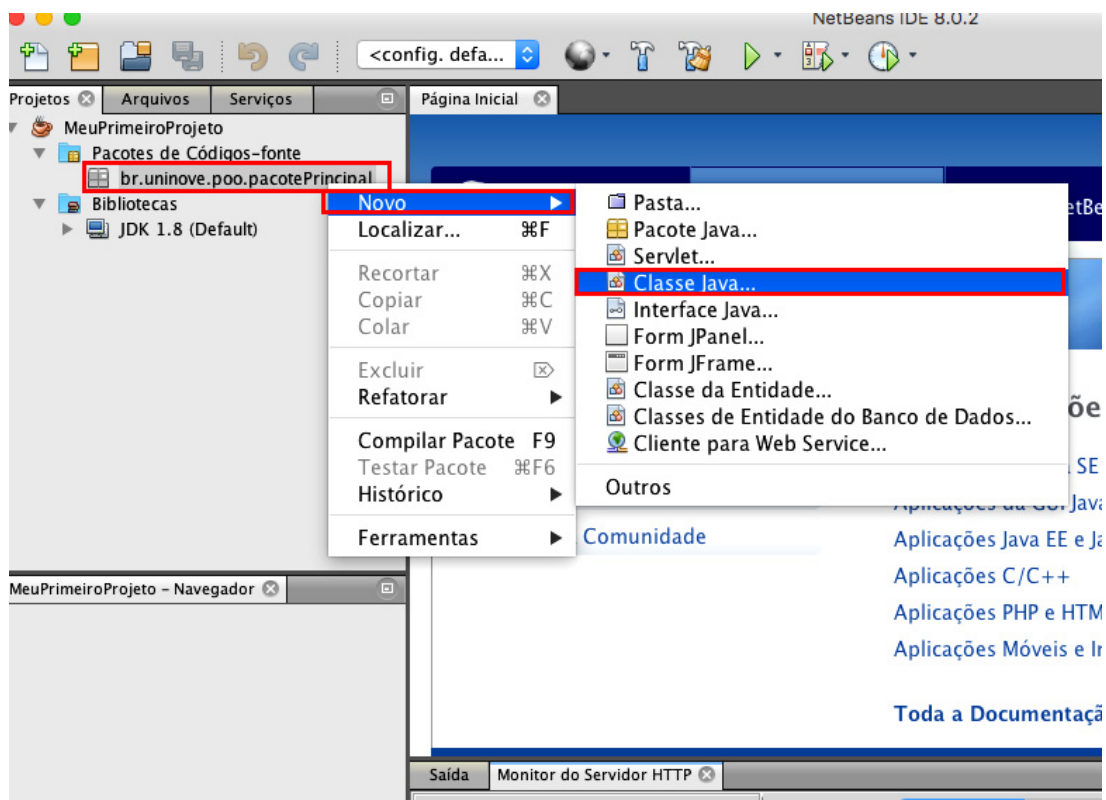
Antes de criar a primeira classe, é preciso criar um pacote (não é obrigatório, mas é muito importante para que você se organize). Projetos grandes possuem muitos pacotes. A divisão dos pacotes ajuda a dividir as responsabilidades de cada conjunto de classes. Um pacote nada mais é (fisicamente) que uma pasta onde as classes ficam.

Para criar o pacote, clique com o botão direito em “Pacotes de código-fonte” e selecione: Novo —> Pacote Java. No assistente que abrir, defina um nome para seu pacote e clique em finalizar.

Existe um padrão internacional de boas práticas de programação para nomenclatura de pacotes. Normalmente utiliza-se algo parecido como `br.com.NomeDaMinhaEmpresa.nomeDoProjeto.NomeDoPacote`, por exemplo, `br.uninove.poo.pacotePrincipal`. Pode parecer estranho, mas você se acostuma.

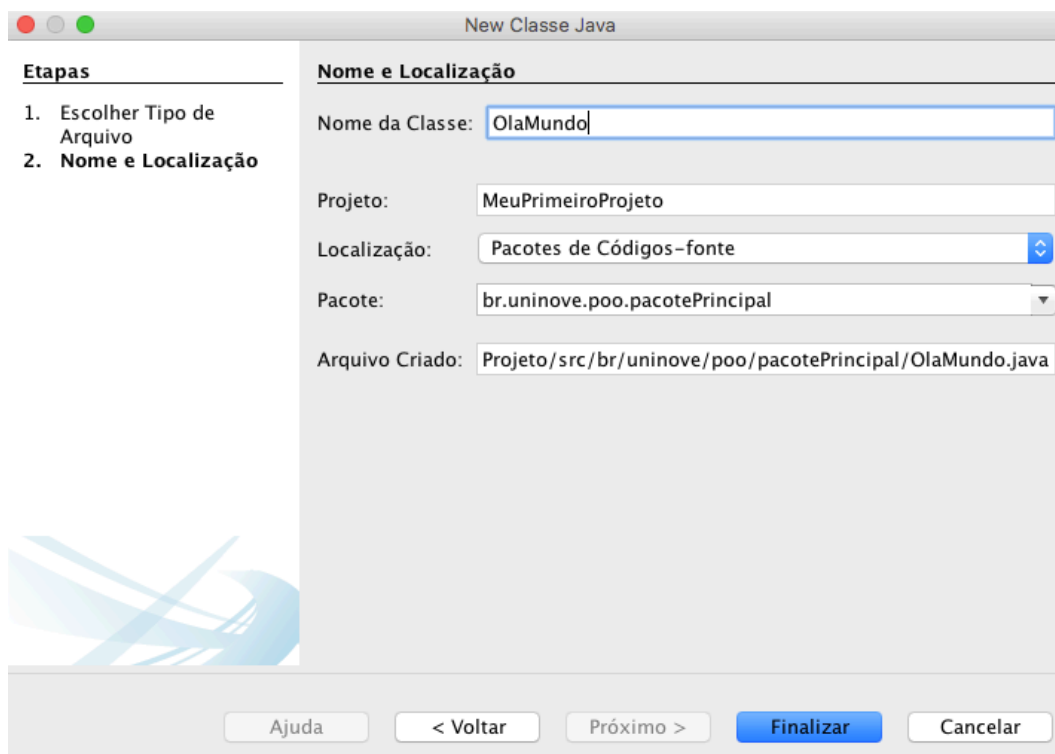
Depois disso, você irá criar, finalmente, sua primeira classe Java. Para isso, clique com o botão direito **no pacote criado** e em seguida, no menu que abrir, clique em Novo —> Classe Java, como pode ser visto na imagem abaixo:





Criando a primeira classe no NetBeans

Na tela que abrir, é preciso dar um nome a sua classe, como visto abaixo:



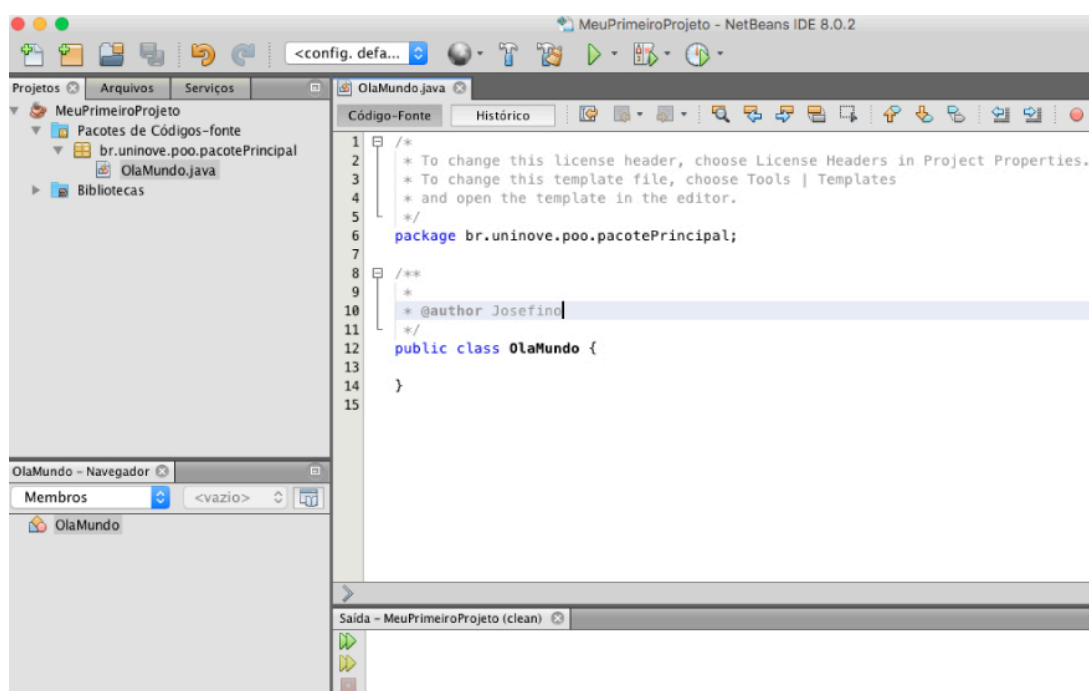
Nomeando a classe no Netbeans

## MUITO IMPORTANTE - BOAS PRÁTICAS DE PROGRAMAÇÃO

Para nomear a classe, **não utilize espaços ou caracteres especiais** (incluindo “ç” e acentos). Em boas práticas de programação devemos nomear a classe com **a primeira letra maiúscula e as demais minúsculas**, ou seja, não use somente letras minúsculas e somente maiúsculas. Exemplos: “Pessoa”; “Endereco”; “Caneta”; “Veiculo” etc.

Na imagem acima, o nome definido da classe é “OlaMundo”. O projeto, a localização dela e o pacote o qual ela será criada, já estão pre-definidos pois clicou-se exatamente sobre o pacote onde deseja-se criar a classe. Note que a classe criada será, fisicamente, um arquivo na pasta onde está localizado seu projeto, visto em “Arquivo criado”. Note que o nome do arquivo é o mesmo que o nome da classe e sua extensão é .java. Note também que o pacote foi dividido em pastas. Feito isso, clique em “Finalizar”.

Feito isso, sua primeira classe está criada. Note que o NetBeans colocou alguns comentários, também:



Classe criada, com sua estrutura pré-definida

Antes de criar o primeiro programa, vamos entender, rapidamente, como funcionam comentários em Java.

Para comentar uma linha, você deve utilizar duas barras “//”. Do ponto em diante onde essas barras foram inseridas o texto é livre. Para criar um bloco de comentário é preciso usar um delimitador para iniciar o bloco e outro para termina-lo. Assim:

```
1.  /**
2.     você pode criar o bloco de comentário
3.     do tamanho que quiser aqui dentro
4.  */
```

Os delimitadores para blocos de comentários são “/\*\*” para abrir o bloco e “\*/” para fechar o bloco.

## DICA

**Evite** usar caracteres especiais e acentos dentro dos comentários

Os comentários criados pelo NetBeans são padrões e podem ser apagados.

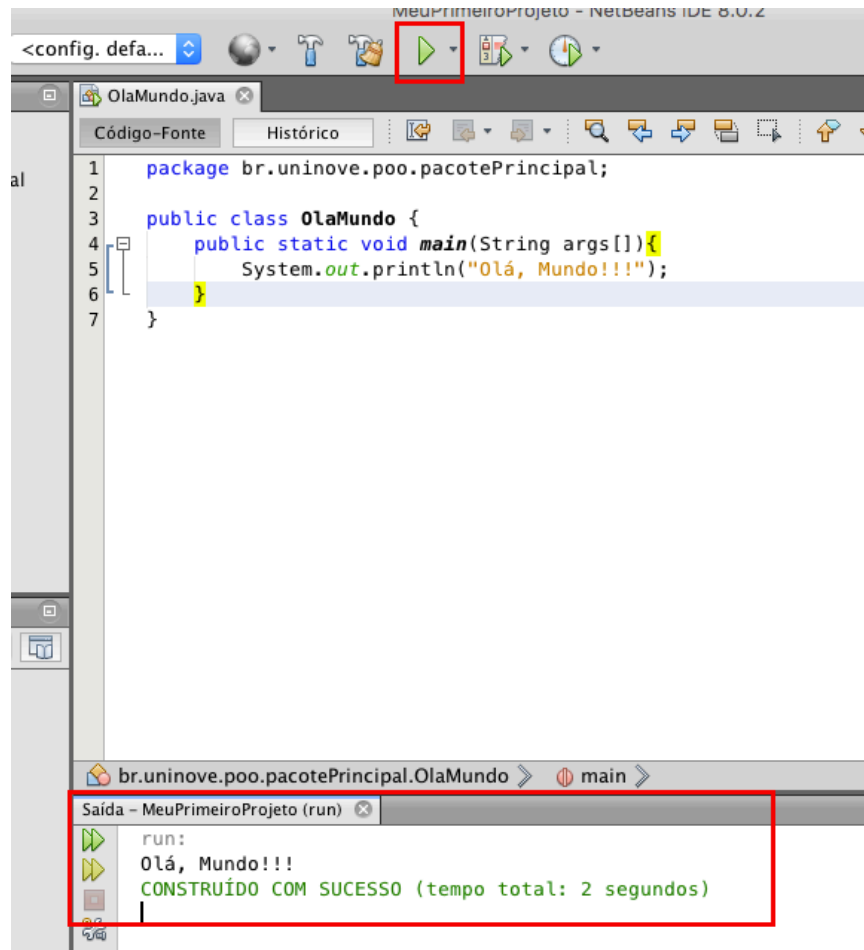
A primeira classe, com o olá mundo, deverá ficar assim. A classe está comentada para melhor entendimento de cada linha:

```
1. package br.uninove.poo.pacotePrincipal; // Obrigatório se a classe está dentro de um
   pacote.
2.
3. public class OlaMundo {                // Definição de classe em Java.
4.     public static void main(String args[]){    // Essa é a definição do método mai
        n.
5.         System.out.println("Olá, Mundo!!!"); // Para imprimir no console
6.     }                                           // fechamento do método
7. }                                              // fechamento da classe
```



O método main é o método que o compilador irá procurar para começar a execução e seu código. É a partir dele que tudo começa, por isso é importante sempre descreve-lo da mesma forma como mostrado acima. Só é possível ter um único método main por projeto.

Feito isso, pode-se executar seu código, clicando na “Seta verde”, chamada “Executar projeto”. O resultado da execução será exibido no console de execução do próprio Netbeans, conforme ilustrado pela imagem abaixo.



Execução do código Olá Mundo

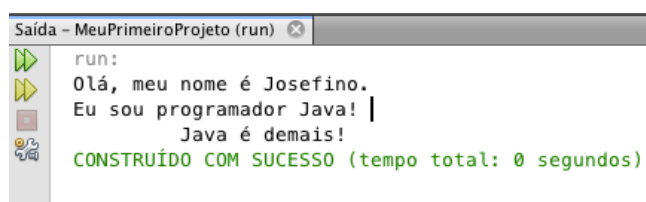


O comando “System.out.println()” é uma chamada ao método “println” que imprime o texto informado e coloca uma quebra de linha ao final. Se você não quer a quebra de linha ao final, utilize o método “print”.

Textos em Java podem ter quebras e tabulações. Para quebrar uma linha dentro do texto (pular linha), utilize o comando “\n”. Para tabular, utilize o comando “\t”. Por exemplo:

1. `//...`
2. `System.out.println("Olá, meu nome é Josefino. \nEu sou programador Java! \n \t Java é demais!");`
3. `//...`

E o resultado será:



Execução do código acima, com quebra de linhas e tabulação

## O FAMOSO PONTO E VÍRGULA - MUITO CUIDADO!

Note que ao final de cada linha é preciso colocar um “;” (ponto e vírgula). Muito cuidado, pois esquecer o “ponto e vírgula” é a causa de muitos erros. O “;” deve ser usado sempre ao final de cada **linha de instrução**, ou seja, não é preciso coloca-lo ao final de abertura de métodos ou classes.

## Resumo dessa aula

Nesta aula você foi apresentado(a) ao importantíssimo mundo Java, onde aprendeu:

- O breve histórico do Java, que começou a se popularizar em 1995
- A evolução da linguagem
- A importância do Java na tecnologia hoje, devido a diversos fatores, como portabilidade, orientação a objetos etc.
- As importantes siglas que compõe o mundo Java: JRE, JDK, JSE, JEE, JME e JVM. Usaremos uma IDE para desenvolver em Java, o NetBeans
- As extensões que envolvem o mundo Java (.java, .class, .jar)
- A diferença entre Java e JavaScript, que não são a mesma coisa
- A preparação do ambiente NetBeans
- Uma primeira aplicação Java: O Hello World



Pronto! Agora você é um programador Java! Hora de começar a ver algumas coisas um pouco mais sofisticadas, mas isso é assunto para outra hora. Tenha certeza que você entendeu a importância dessa nobre linguagem, seu funcionamento e que seu ambiente de desenvolvimento está funcionando corretamente.

## Quiz

Exercício Final

INICIAR ➤

## Referências

Oracle, The History of Java Technology, disponível em <http://www.oracle.com/technetwork/java/javase/overview/javahistory-index-198355.html> (<http://www.oracle.com/technetwork/java/javase/overview/javahistory-index-198355.html>), acessado em 8 de Março de 2016

Oracle, Java Platform Standard Edition 8 Documentation, disponível em <https://docs.oracle.com/javase/8/docs/> (<https://docs.oracle.com/javase/8/docs/>), acessado em 8 de Março de 2016

Cass, S., 2015, The 2015 top ten programming languages, IEEE, disponível em <http://spectrum.ieee.org/computing/software/the-2015-top-ten-programming-languages> (<http://spectrum.ieee.org/computing/software/the-2015-top-ten-programming-languages>), acessado dia 15 de Março de 2016

Deitei P. e Deitel H., 2010, Java : Como programar, 8ª Edição, Pearson Pretice Hall

Teruel, E. C., 2015, Programação Orientada a Objetos com Java - sem mistérios - 1ª Ed., Editora Uninove



Avalie este tópico



Biblioteca  
(<https://www.uninove.br/conhec-a-uninove/biblioteca/sobre-a-biblioteca/apresentacao/>)  
Portal Uninove  
(<http://www.uninove.br>)  
Mapa do Site



Ajuda?  
PRÓXIMO  
(<https://ava.uninove.br/curso/>)  
Introdução à Programação Orientada a Objetos com Java

© Todos os direitos reservados

