

Qualificação Profissional de Assistente de  
Desenvolvimento de Sistemas

# LÓGICA DE PROGRAMAÇÃO

# **GEEaD - Grupo de Estudos de Educação a Distância**

## **Centro de Educação Tecnológica Paula Souza São Paulo – SP, 2019**

### **Expediente**

GOVERNO DO ESTADO DE SÃO PAULO  
EIXO TECNOLÓGICO DE INFORMAÇÃO E COMUNICAÇÃO  
TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS  
LÓGICA DE PROGRAMAÇÃO

*Autores:*

*Eliana Cristina Nogueira Barion*

*Marcelo Fernando Iguchi*

*Paulo Henrique Mendes Carvalho*

*Rute Akie Utida*

*Revisão Técnica: Kelly Cristina Dall Pozzo*

*Revisão Gramatical:*

*Juçara Maria Montenegro Simonsen Santos*

*Editoração e Diagramação: Flávio Biazim*

---

# AGENDA 3

---

## A LÓGICA APLICADA EM JAVA







## MERGULHANDO NO TEMA...

Java é uma linguagem orientada a objetos<sup>1</sup> que foi desenvolvida nos anos de 1990 pela Sun Microsystems, projetada para ser pequena, simples e portátil a todas as plataformas e sistemas operacionais.

É utilizada para desenvolver aplicativos corporativos, páginas web com conteúdo dinâmico e interativo, aprimorar funcionalidades de servidores www e está cada vez mais sendo utilizada para desenvolvimento de aplicativos móveis para telefones celulares, pagers e PDAs.



Muito provavelmente você já deve ter ouvido falar sobre a linguagem Java. Basta ler uma revista de informática ou matérias sobre desenvolvimento de *softwares* que logo encontra alguma informação sobre Java, dado o sucesso que esta linguagem tem tido no mercado, principalmente pelo fato dos programas escritos em Java poderem ser executados virtualmente em qualquer plataforma e aceitos em qualquer tipo de computador ou outros aparelhos, uma característica marcante da Internet. Por isso, a linguagem Java tem se destacado muito no mercado e aprendê-la é importante para você e para a sua profissão!



***Java é uma Linguagem de Programação Orientada a Objetos, porém para que você compreenda melhor a Lógica utilizando esta Linguagem, iremos utilizar o Console (uma forma estruturada) para exercitar os comandos.***

Para escrevermos um programa desenvolvido em Pseudocódigo de forma que o computador possa compilá-lo<sup>2</sup> e executá-lo, precisamos utilizar uma Linguagem de Programação que nada mais é do que o “idioma” necessário para conversar com o computador. No desenvolvimento de nossos estudos em Lógica de Programação utilizaremos a linguagem de programação Java.

O Java atualmente é uma das linguagens de programação mais utilizadas no mercado de trabalho. É também capaz de fornecer uma portabilidade muito grande, sendo compatível com a maioria dos Sistemas Operacionais disponíveis para usuários gerais.

O mesmo programa escrito com a Linguagem de Programação Java, poderá ser utilizado em um computador com os Sistemas Operacionais Windows, Linux ou Mac OS sem que nenhuma linha do código fonte<sup>3</sup> necessite ser alterada, ganhando tempo no desenvolvimento de novas aplicações.

<sup>1</sup> - Linguagem de Programação Orientada a Objetos (P.O.O) significa que esta Linguagem de Programação utilizará os conceitos de Orientação a Objetos (O.O.) o qual veremos no módulo seguinte com mais profundidade.

<sup>2</sup> - Compilação é a ação de transformar um código amigável escrito com uma Linguagem de Programação em um programa executável baseado em Código de Máquina.

<sup>3</sup> - Código Fonte é o conjunto de instruções criadas pelo programador utilizando uma Linguagem de Programação.

Além disto, Java é uma linguagem de programação que pode ser utilizada para desenvolver páginas da Internet, por meio de um Servidor Web configurado para executar páginas do tipo Java Server Pages (jsp). Com ele também podemos desenvolver aplicativos para celulares que utilizam o Sistema Operacional Android.

Pela sua vasta lista de plataformas suportada, utilizaremos o Java como Linguagem de Programação de apoio no desenvolvimento da Lógica de Programação.



*Você sabia que o Java não é a única linguagem de Programação que funciona em todos os principais Sistemas Operacionais do mercado? Pesquise outras linguagens existentes no mercado e reflita sobre a eficiência delas em relação à Linguagem Java.*

### Conhecendo as ferramentas Java

Para iniciar o estudo, primeiramente é necessário conhecer as diferentes ferramentas para desenvolvimento que o Java oferece (JRE, JVM, JSE, JEE, JME, JDK...) parece uma sopa de letrinha, não é mesmo? Em seguida, com base na aplicação que pretende desenvolver, deve selecionar as tecnologias e ferramentas necessárias para este fim.

É mais ou menos assim: Imagine que você precisa fazer um brigadeiro! Com a receita em mãos, você vai até o supermercado, compra os ingredientes: leite condensado, chocolate em pó, manteiga e chocolate granulado e ainda providencia as ferramentas necessárias para a confecção do doce: panela e colher de pau.

A mesma coisa acontece com o desenvolvimento de um aplicativo em Java! Conhecendo o tipo de aplicação que deseja desenvolver, você seleciona as ferramentas necessárias para tal finalidade. Vamos lá:



Imagem 03: Freepik – Doce de brigadeiro

### Java Virtual Machine – JVM

O que faz com que a portabilidade da linguagem Java seja eficiente é uma aplicação responsável por executar programas desenvolvidos na linguagem. Sua função é simular a um computador permitindo a execução do código fonte, por isto recebe o nome de Máquina Virtual.

Na prática, basta instalar em seu computador a JVM desenvolvida para o Sistema Operacional e você estará pronto para executar programas desenvolvidos em Java.

Imagem 04 Freepik – Afazeres



## O Java Development Kit (JDK) e a Integrated Development Environment (IDE)

Como em breve seremos desenvolvedores e não usuários, também precisaremos instalar em nosso computador o kit para desenvolvimento de programas feitos em Java, o que inclui o compilador<sup>4</sup> da linguagem de programação e a Máquina Virtual (JVM). Sem ele não é possível finalizar um programa desenvolvido em Java, mesmo que você escreva o código fonte completo.

Apesar de já ser possível criar programas apenas com o JDK, utilizaremos uma interface de desenvolvimento integrada (IDE) para auxiliar na codificação, compilação e testes dos nossos

programas. Dentre algumas IDEs de desenvolvimento mais conhecidas do Java estão o NetBeans, o Eclipse, o Blue J e o IntelliJ Idea. Independente da IDE escolhida, os comandos sempre serão os mesmos, logo, se você aprende a programar em Java, conseguirá utilizar qualquer uma das IDEs sem maiores problemas.

Durante o desenvolvimento das atividades utilizaremos a IDE Netbeans, porém você pode utilizar qualquer outra IDE desde que siga fielmente as estruturas listadas neste material.

***Durante os nossos estudos, utilizaremos a distribuição do Java SE (Standard Edition), que é voltada para o Desenvolvimento de Sistemas Desktop. Existem também as Distribuições Java ME (Micro Edition) que é voltada para dispositivos de pequeno porte e Java EE (Enterprise Edition) que é direcionado a aplicações corporativas e Web.***

### ***Colocando a mão na massa...***

Para começar seu primeiro software, é necessário baixar e instalar as ferramentas que você acabou de conhecer.

Logo abaixo inserimos um tutorial passo a passo do Netbeans, mas se você já conhece outra IDE, fique à vontade em escolher a interface de sua preferência.

Na seção Ampliando Horizontes, disponibilizamos um [tutorial de Instalação do IntelliJ Idea](#). Caso prefira trabalhar com essa IDE, acesse esse link.

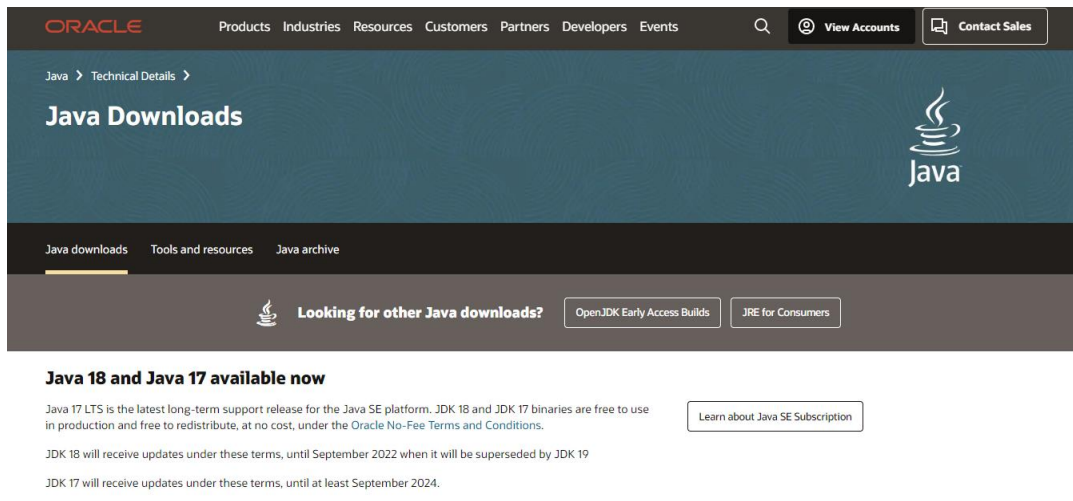
## Tutorial passo a passo do Netbeans...

Vamos conhecer agora o passo a passo para instalação da IDE NETBEANS e o JDK.

A IDE Netbeans é a ferramenta oficial de desenvolvimento Java para Oracle, portanto é possível baixar o kit JDK junto com o Netbeans no site da Oracle.

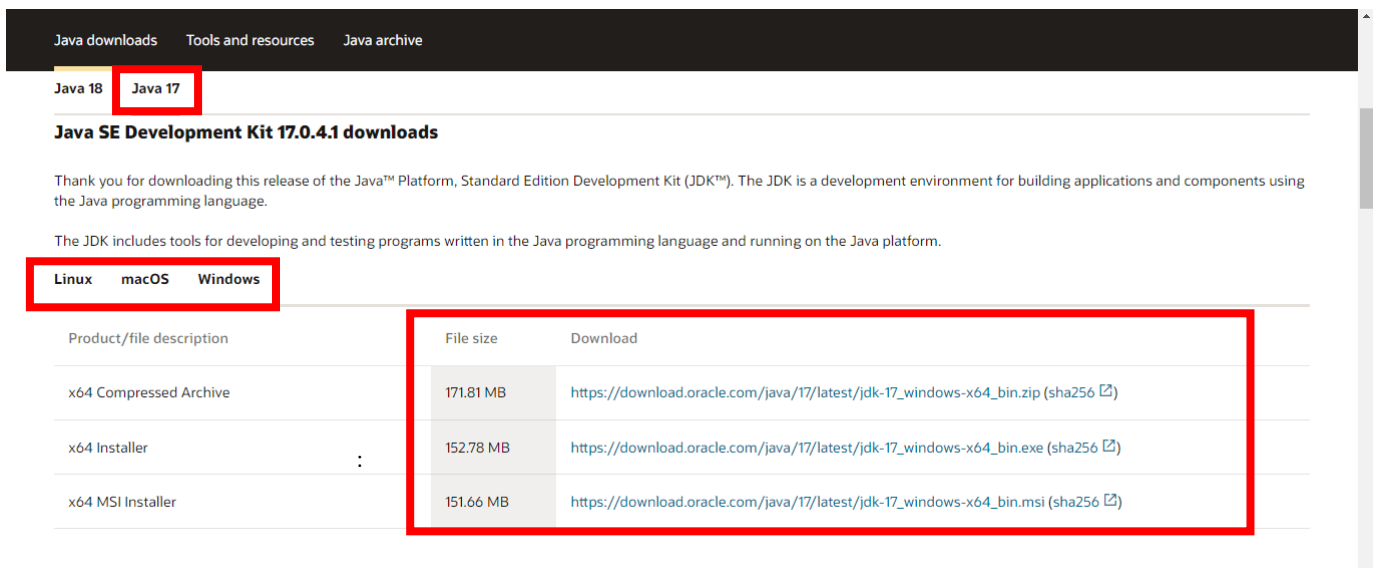
4- Erros de compilação são configurados quando há erros na estrutura do código fonte. Caso este tipo de erro ocorra, não será possível a execução do programa.

Acesse o site: <https://www.oracle.com/java/technologies/downloads/>

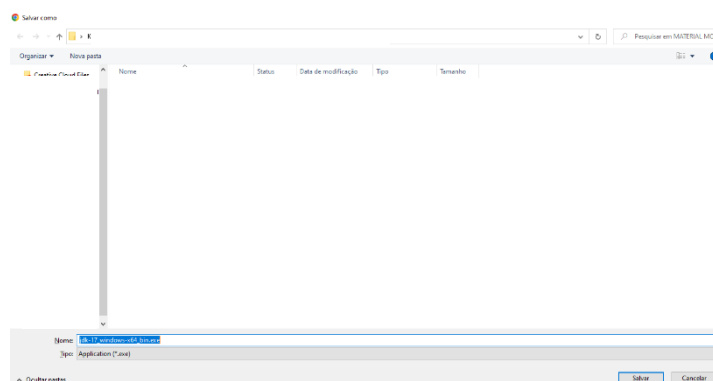


No site oficial da Oracle a versão em destaque para download é versão 18, porém para essa versão já foi informado que será substituída em setembro, portanto baixe a versão 17 ou inferior, dependendo da capacidade de sua máquina.

1. Selecione a versão 17 ou inferior (no final da página), logo após, selecione o seu sistema operacional e escolha o arquivo para Download de acordo com a versão do seu sistema.



2. Ao selecionar a opção de sistema o download dos arquivos se inicia, selecione o local para armazenamento.

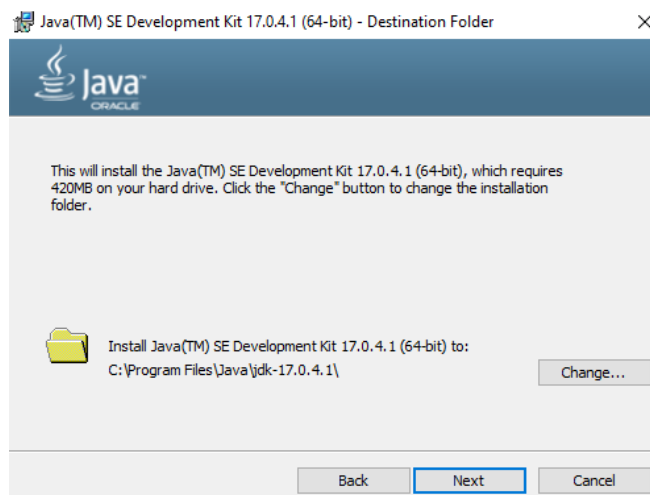


**Muito importante!**  
Se estiver utilizando o Sistema Operacional Windows é fundamental verificar se o sistema é 32 ou 64bits.

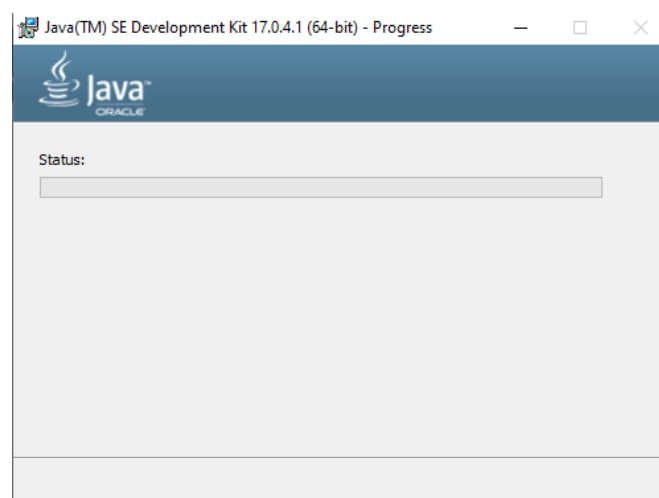
3. Após download, vá até o local onde o arquivo foi armazenado e de um duplo clique sobre o mesmo.
4. Após iniciar o processo de instalação, clique em próximo(Next).



5. Defina o local de instalação, sendo recomendável manter o padrão adicionado de forma automática. Em seguida, clique em Próximo (Next).



6. Aguarde o processo de instalação.



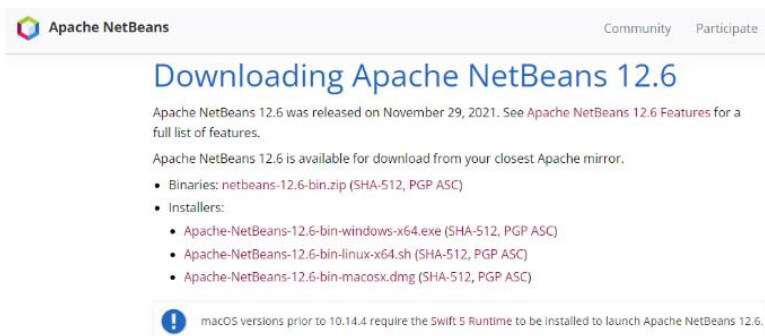
7. Logo após clique no botão “Close” para finalizar o processo de instalação do JDK.





Com isso, o kit JDK já está instalado em seu computador, agora vamos para instalação da IDE. Neste material vamos mostrar a instalação da versão já testada do Netbeans 12.6.

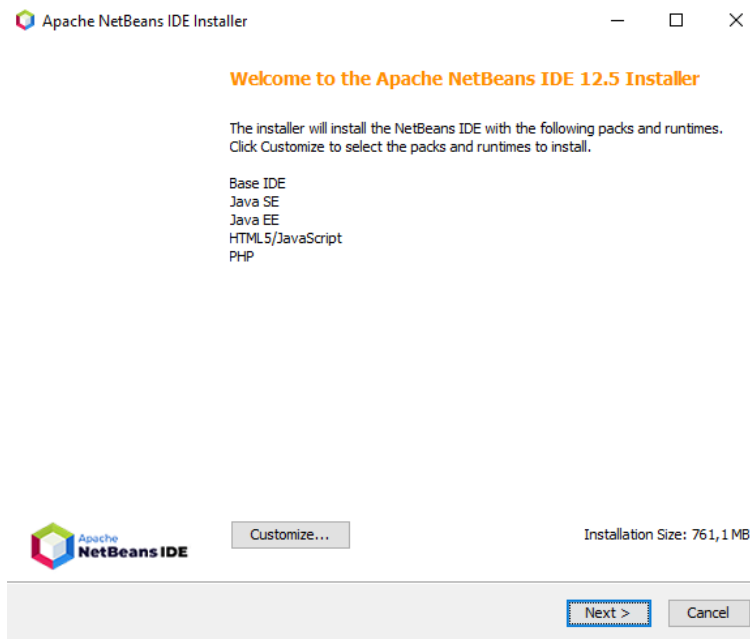
8. Acesse o link: <https://netbeans.apache.org/download/archive/index.html>
9. Selecione versão desejada para download, neste material utilizamos a 12.6.



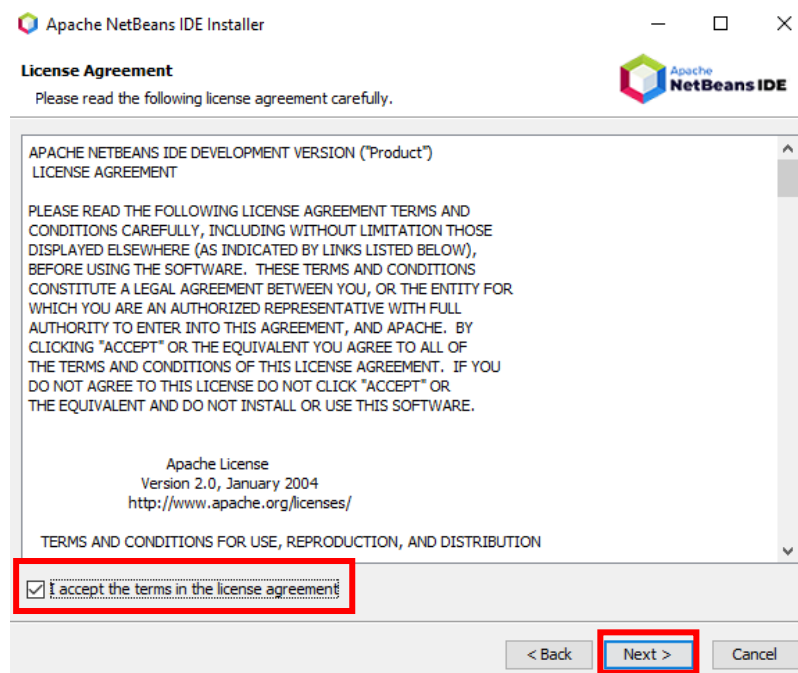
10. Selecione a versão correspondente ao seu Sistema Operacional para iniciar o download.

11. Escolha o local para armazenamento do arquivo, clique em **Salvar**.
12. Após download, vá até o local onde o arquivo foi armazenado e de um duplo clique sobre o mesmo.

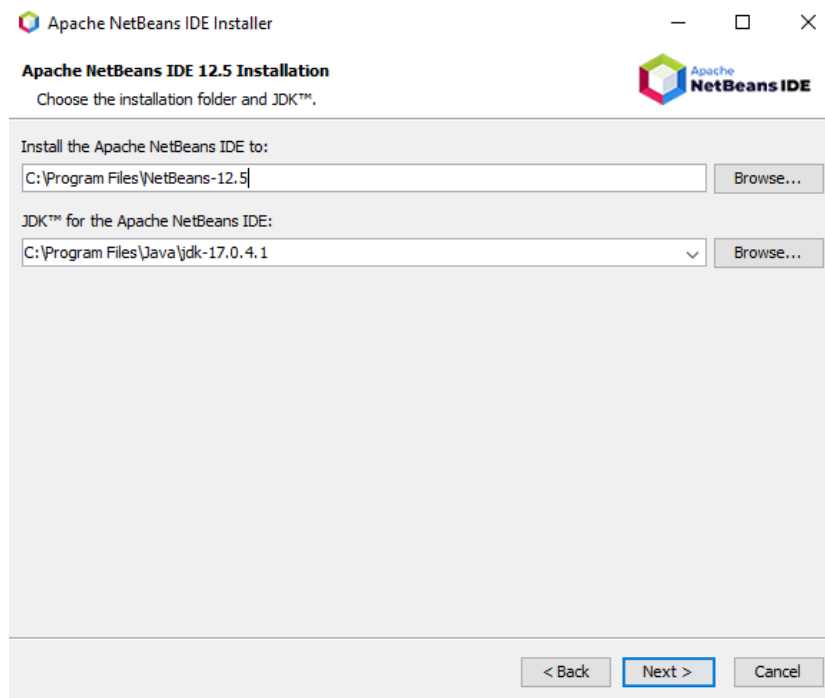
13. Na tela inicial de instalação pressione Próximo (Next).



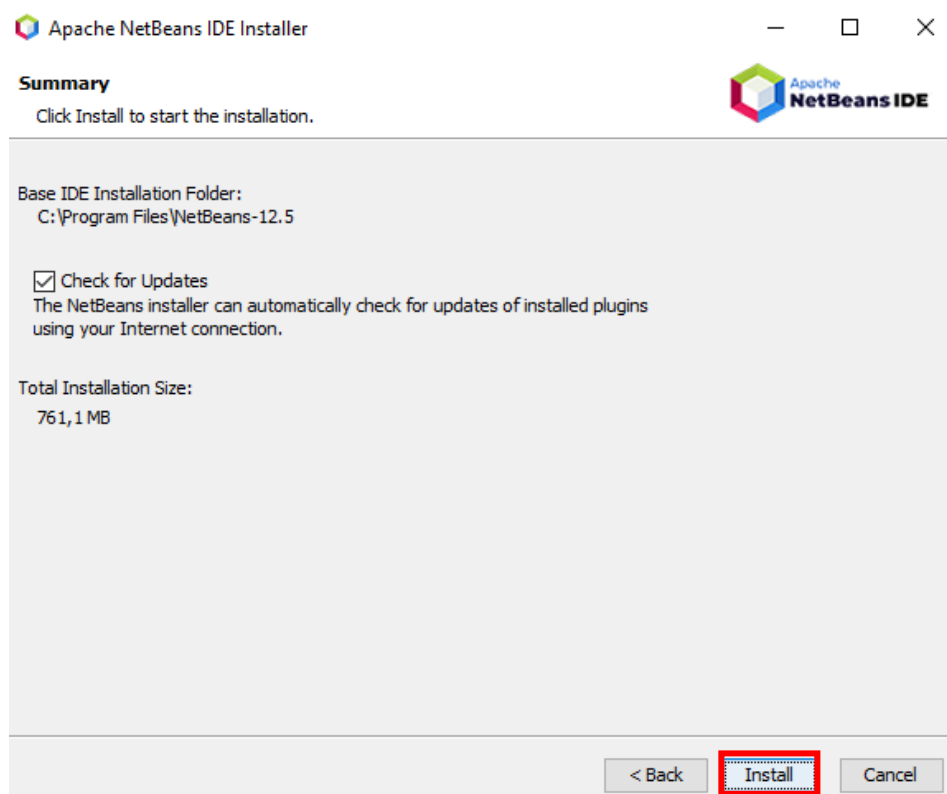
14. Para continuar o processo de instalação aceite os Termos de Uso. Em seguida, clique em Próximo (Next)



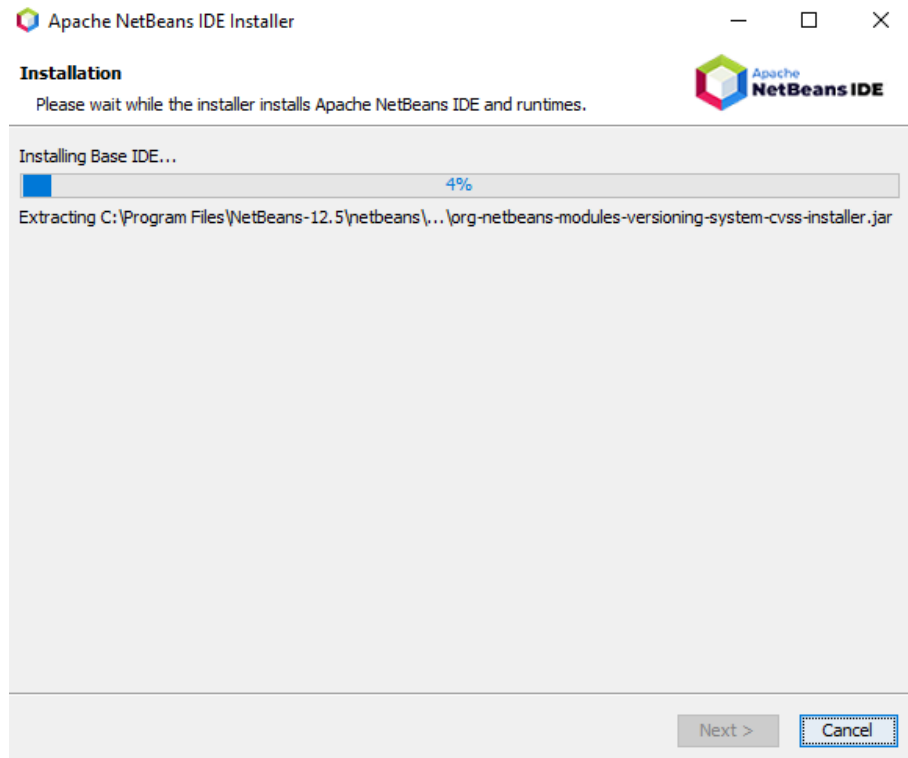
15. Defina o local de instalação, sendo recomendável manter o padrão adicionado de forma automática. Em seguida, clique em Próximo (Next).



16. Em seguida, clique no botão Install.



17. Aguarde o processo de instalação.



Após processo de instalação será exibido um resumo, clique em **Finish**.

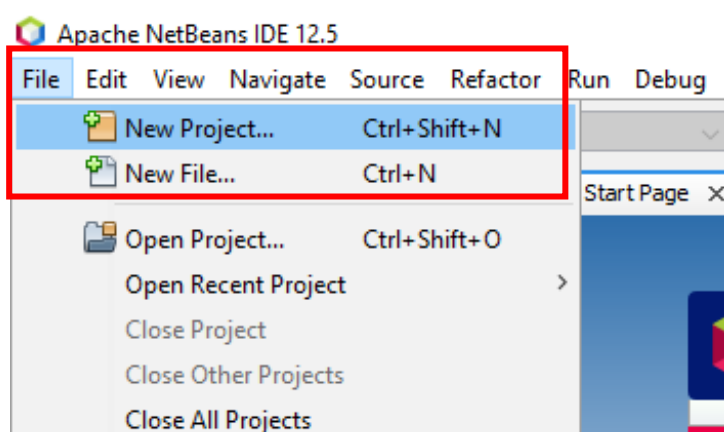
**Importante:** O processo de instalação leva um tempo, pois ele passa por 2 processos, o da instalação do JDK e instalação do NetBeans.

Após instalação, basta localizar e abrir a IDE Netbeans por meio do botão de busca.

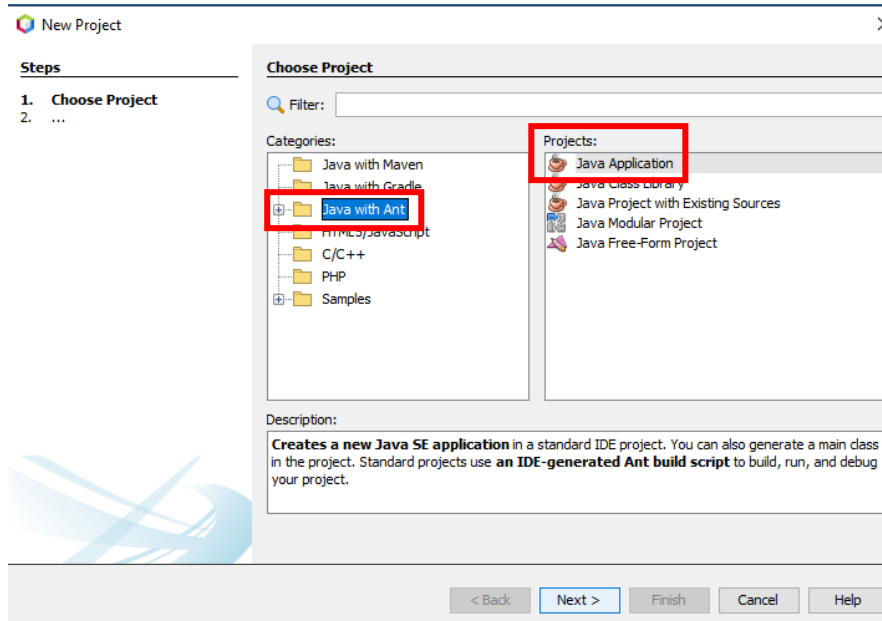
**Importante:** Caso tenha algum problema na abertura do NetBeans, pode haver uma incompatibilidade do seu computador com a versão do JDK instalado, sendo necessário instalar uma versão mais antiga.

Na tela inicial selecione o menu **Arquivo, Novo Projeto** ou utilize o atalho **Novo Projeto** disponível na **Barra de menu**.

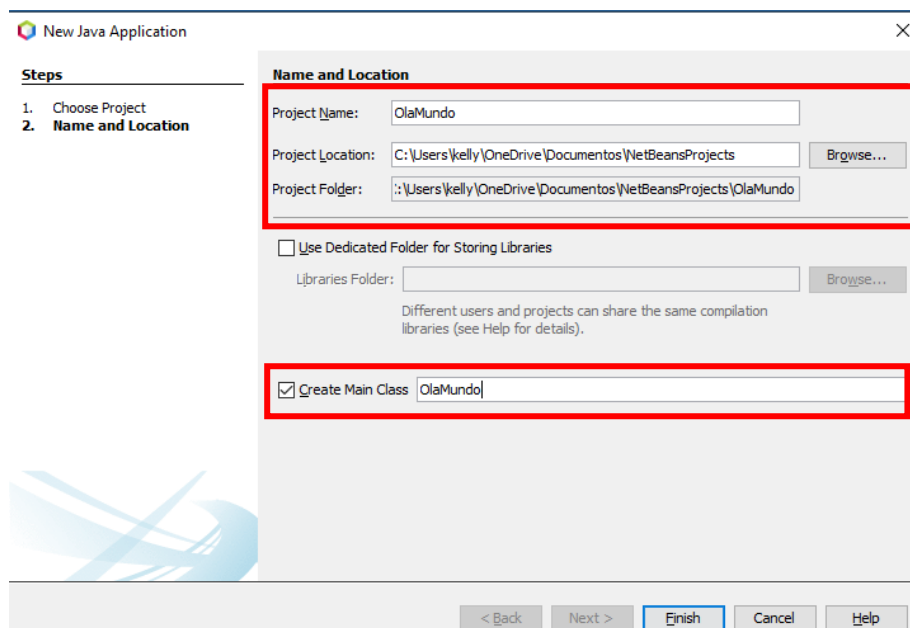
1. Na janela **Novo Projeto**, selecione a opção **File, New Project**.



2. Selecione a categoria **Java with Ant**, em projetos selecione **Java Application**. Clique no botão **Next**.

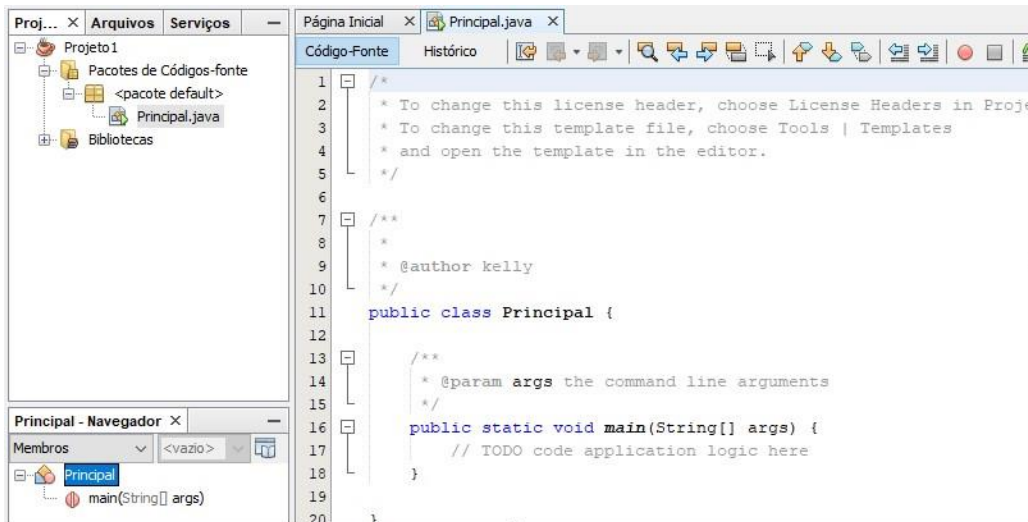


3. Defina um nome e um local para armazenamento do projeto. Defina também um nome para Classe Principal.



4. Clique no botão **Finalizar**.
5. Pronto! Seu projeto já está criado juntamente com a classe Principal.





Agora que você já criou um código fonte em Java, você deve aprender a estrutura básica de um programa. A seguir, temos uma imagem representando um código fonte em Java:

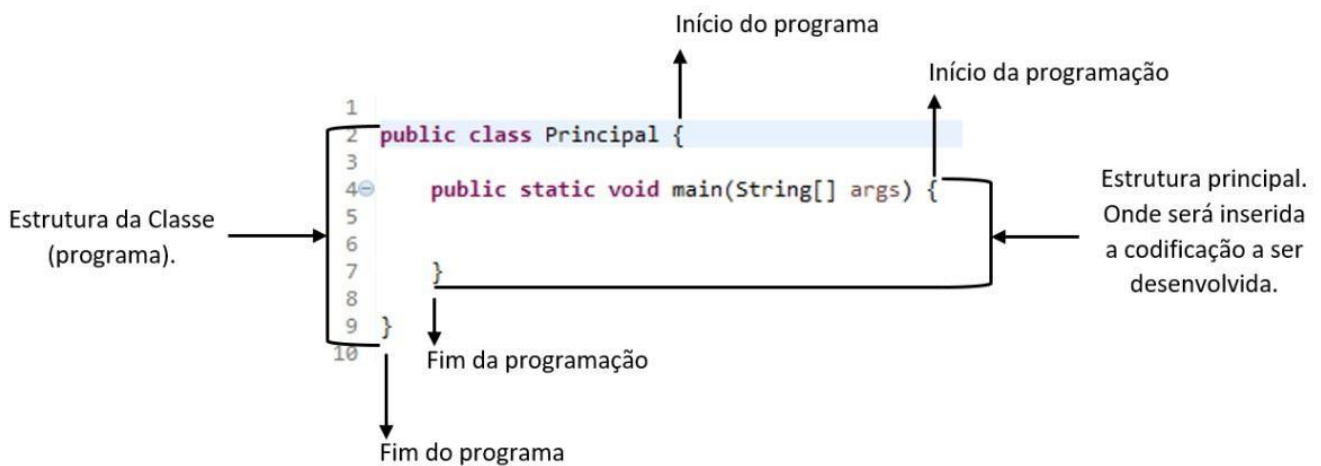


Imagem 21: A imagem representa a estrutura básica de uma classe o símbolo “abre e fecha chaves” significa início e fim de programa, respectivamente. A estrutura de um programa é composta por: public class “nome do Programa”, abre chave, public static void main (String[] args, fecha a chave { da estrutura principal, onde será inserida a codificação a ser desenvolvida e fecha a chave da classe principal.

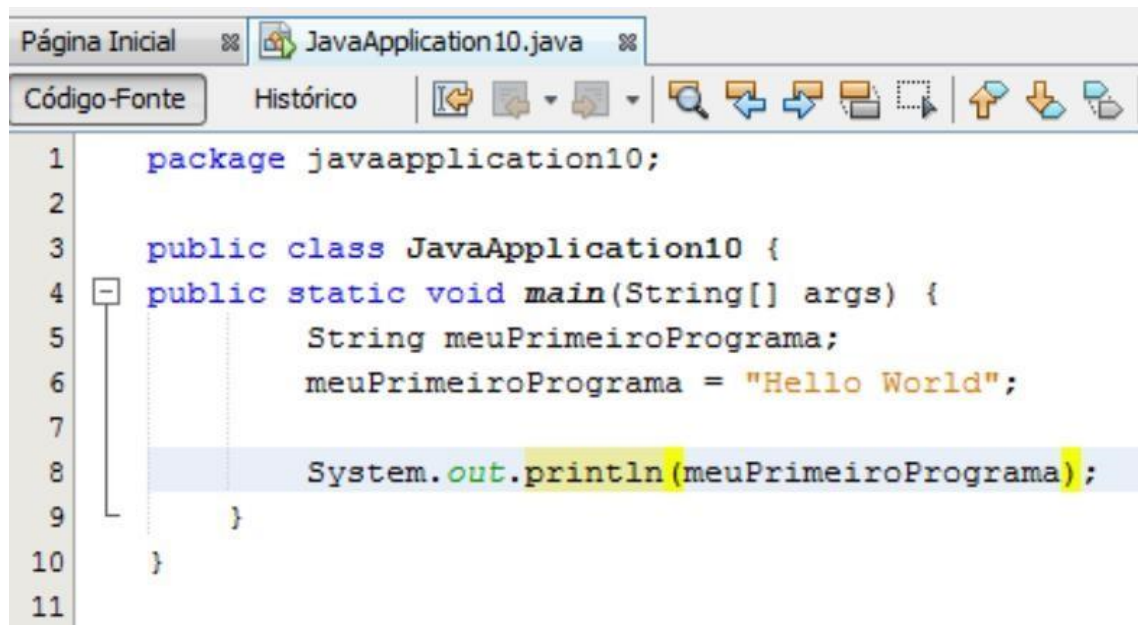
Você pode ver na imagem que o programa em si está dentro de uma classe, essa estrutura é necessária para seu funcionamento, porém, para estudos de lógica de programação não abordaremos os conceitos de Orientação a objetos.

## A estrutura de um programa feito em Java

Os comandos Início e Fim contidos no Pseudocódigo são substituídos por chaves { } em Java, sendo a abertura de Chave { o início e, o fechamento de chave } o final do seu programa. Em algumas estruturas que veremos posteriormente, também utilizaremos marcações de início e fim por meio de chaves.

Nas próximas agendas estudaremos as estruturas e comandos Java. Não se preocupe!

Mas... apenas para ver um pequeno programa rodando em Java, vamos criar o famoso Olá Mundo!



```

1  package javaapplication10;
2
3  public class JavaApplication10 {
4  public static void main(String[] args) {
5      String meuPrimeiroPrograma;
6      meuPrimeiroPrograma = "Hello World";
7
8      System.out.println(meuPrimeiroPrograma);
9  }
10 }
11

```

```

String meuPrimeiroPrograma;
meuPrimeiroPrograma = "Hello World";

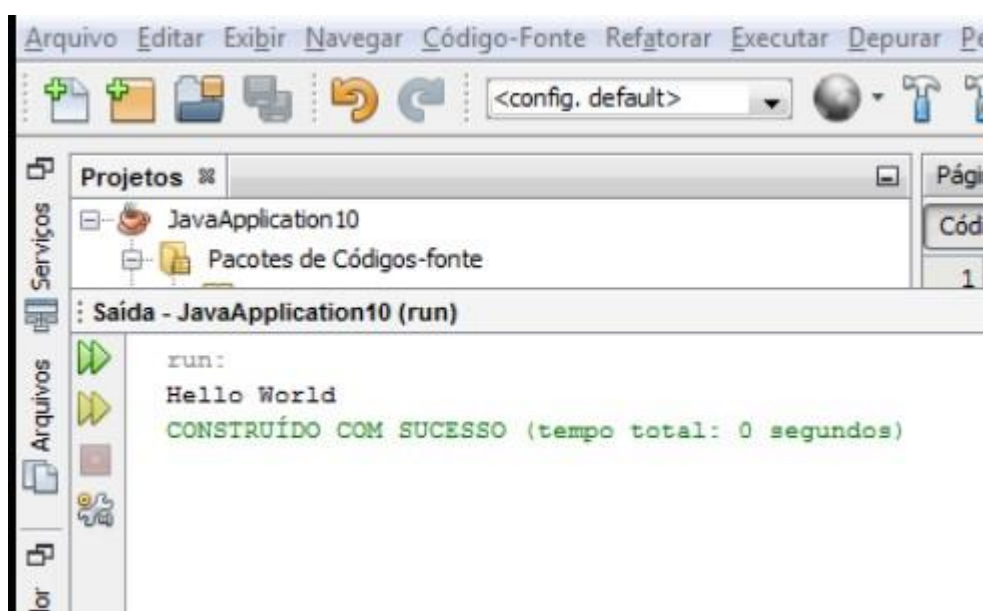
System.out.println(meuPrimeiroPrograma);

```

**Observação:** clique **Alt+Shift+F** para organizar o seu código.

Aperte **F6** ou em **Executar Projeto**.

O programa te mostrará o retorno:



```

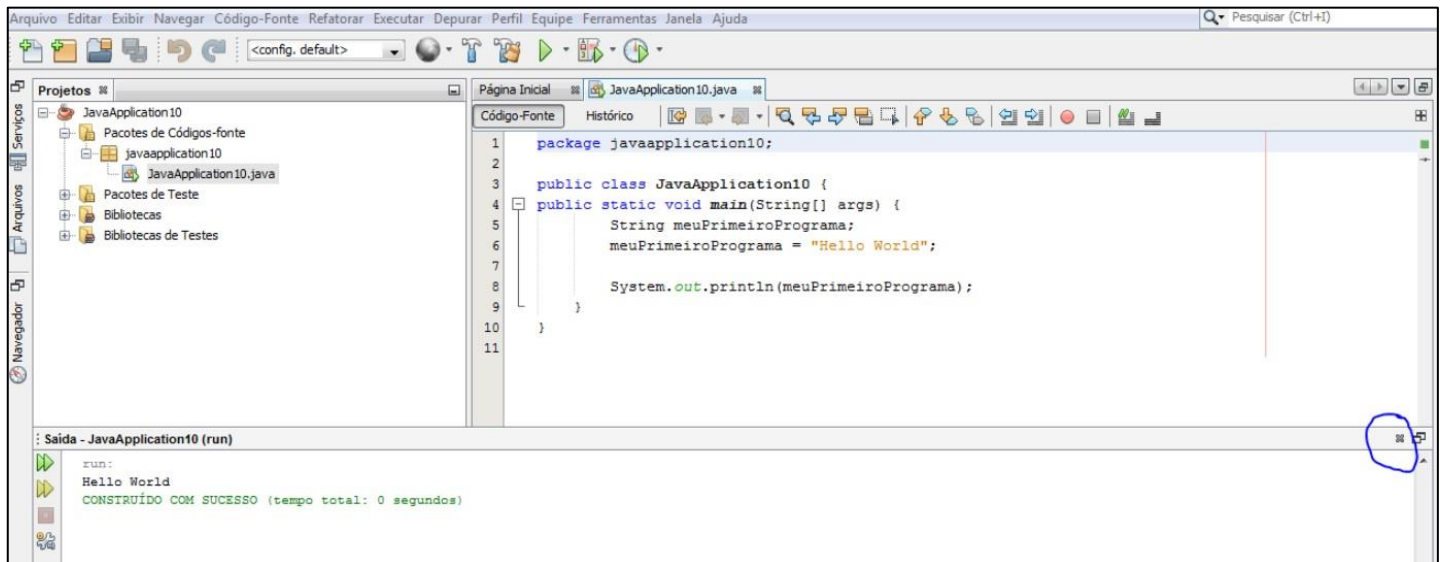
Arquivo  Editar  Exibir  Navegar  Código-Fonte  Refatorar  Executar  Depurar  Pe
<config. default>

Projetos
JavaApplication10
Pacotes de Códigos-fonte

Saída - JavaApplication10 (run)
run:
Hello World
CONSTRUÍDO COM SUCESSO (tempo total: 0 segundos)

```

Arraste a janela ou clique no **X** no canto superior direito para fechar.



(Adaptado de: <https://programacaosemfrescura.blogspot.com/2019/03/meu-primeiro-programa-no-netbeans.html>, acesso em 16/02/2021)

Pronto! Você conseguiu fazer o seu primeiro programa no Netbeans!

Mas você entendeu todas essas codificações?

Se não entendeu, calma! Nas próximas agendas explicaremos tudo detalhadamente.

Até lá!