



---

## Laboratório 01 – NetBeans

### 1 Introdução

Os programas em geral são desenvolvidos com a ajuda dos chamados Ambientes Integrados de Desenvolvimento, ou simplesmente **IDE** (*Integrated Development Environment*). Estes ambientes fornecem recursos importantes para auxiliar e facilitar o trabalho do programador, como a identificação de palavras reservadas durante a escrita do código-fonte, detecção de erros de sintaxe, depuração<sup>1</sup> e compilação.

O IDE adotado no curso é o NetBeans (<https://netbeans.apache.org/>). Como em outros IDEs, todo o desenvolvimento de aplicativos (ou programas) é realizado num projeto. Outras opções populares são o Eclipse (<http://www.eclipse.org/>) e o IntelliJ (<https://www.jetbrains.com/pt-br/idea/> – há uma versão gratuita, a *Community Edition*). Na sequência deste texto é apresentada uma introdução à utilização do IDE NetBeans.

O NetBeans, durante algum tempo, foi disponibilizado oficialmente pela Oracle, para download em conjunto com o JDK<sup>2</sup>, a ferramenta básica de programação em Java. A Oracle doou o NetBeans para Fundação Apache em abril de 2019.

O JDK pode ser obtido oficialmente a partir da URL <https://www.oracle.com/br/java/technologies/downloads/>. É importante ler as instruções de instalação específicas para seu sistema operacional, que também se encontram no link.

### 2 Visão geral da criação da primeira aplicação usando o NetBeans

Esta primeira aplicação, chamada `BemVindo.java`, simplesmente mostrará a frase “Sejam bem vindos ao Java!”.

Para criar este programa é necessário:

- **Criar um projeto IDE (*IDE project*)**

Ao criar um projeto IDE, cria-se o ambiente necessário para construir e executar aplicações. O uso de projetos IDE elimina problemas normalmente associados com o compilador `javac` e demais ferramentas. Para construir e executar aplicações basta escolher um menu dentro do IDE.

---

<sup>1</sup>Processo de identificação e remoção de erros, facilitado pelos IDEs.

<sup>2</sup>*Java SE development toolkit*, ou kit de ferramentas para desenvolvimento em Java – Standard Edition. Para desenvolvedores em Java, incluía o JRE completo (ambiente de execução), além de ferramentas para desenvolver, depurar e monitorar aplicações Java

- **Adicionar código para o arquivo-fonte gerado**

Um arquivo-fonte contém código, escrito na linguagem de programação Java. Como parte da criação de um projeto IDE, é gerado automaticamente o esqueleto de um arquivo-fonte, que deverá ser modificado ao adicionar-se a mensagem “Sejam bem vindos ao Java”.

- **Compilar o arquivo-fonte gerando o arquivo com extensão .class**

O IDE inicia o compilador (**javac**), que traduz o arquivo-fonte em instruções que podem ser traduzidas pelo JVM (*JAVA virtual machine*). Estas instruções são conhecidas como *bytecodes*.

- **Executar o programa**

A aplicação poderá ser executada por meio do IDE.

## Criando uma aplicação a partir do NetBeans

A primeira aplicação usando o NetBeans irá mostrar uma mensagem de boas-vindas ao Java na tela.

Passos:

1. **Carregue o IDE NetBeans**<sup>3</sup>
2. **Crie um novo projeto.** Escolha, no NetBeans, **File** → **New Project...**, use o atalho de teclado **Ctrl+Shift+N**, ou ainda, use o ícone correspondente na barra de ferramentas (Figura 1).

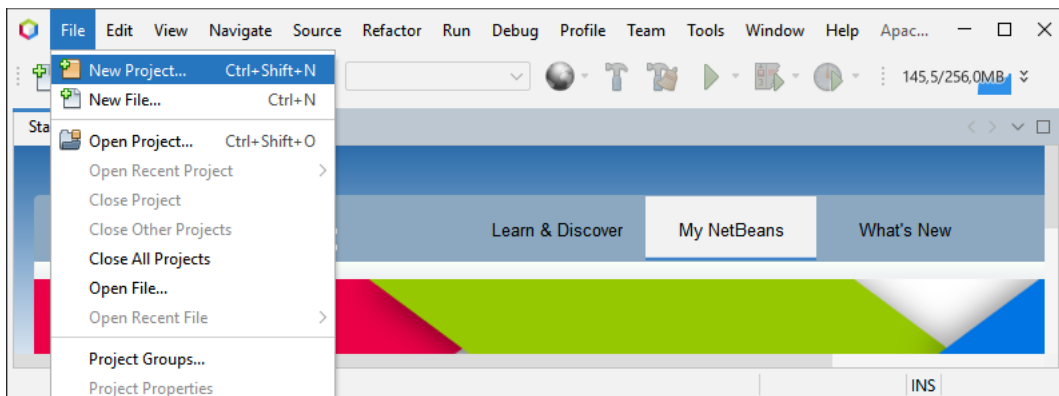


Figura 1: Criando novo projeto

Na janela de novo projeto, expanda a categoria **Java with Maven**<sup>4</sup> e selecione **Java Application** – vide Fig. 2.

3. Após clicar em **Next >**, **preencha o campo Project Name** com **BemvindoJava** – veja Figura 3. Mantenha o restante como está.

<sup>3</sup>Serão usadas figuras da versão 18 da IDE (corrente em 08/2023). Adaptado de <http://docs.oracle.com/javase/tutorial/getStarted/cupojava/netbeans.html> (versão 8.0).

<sup>4</sup>Em versões anteriores do NetBeans, como a 8.0 – quando a IDE ainda era mantida pela Oracle – a categoria era chamada simplesmente **Java**.

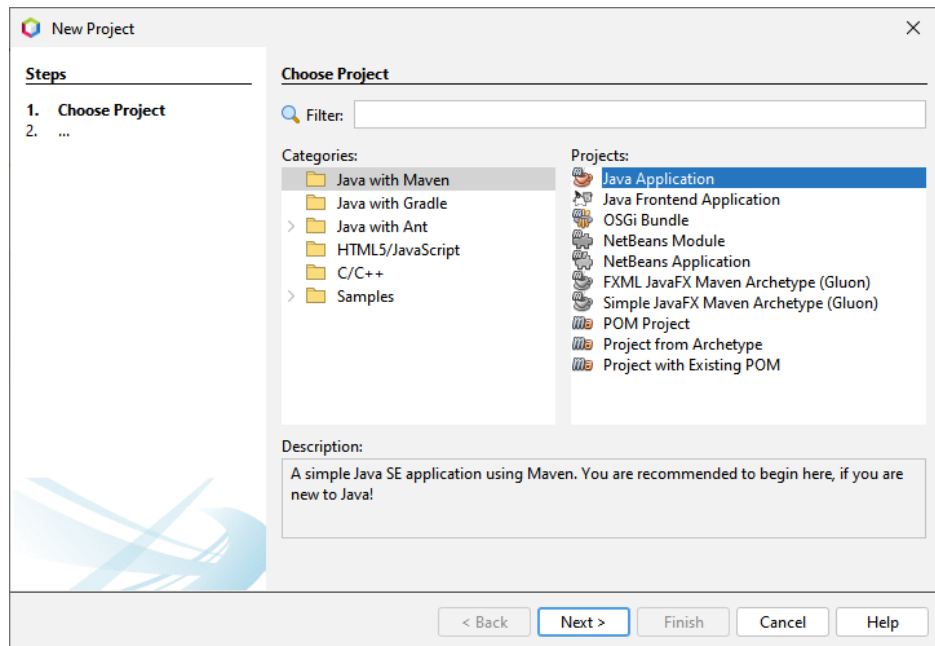


Figura 2: Categoria de Projeto – Java Application

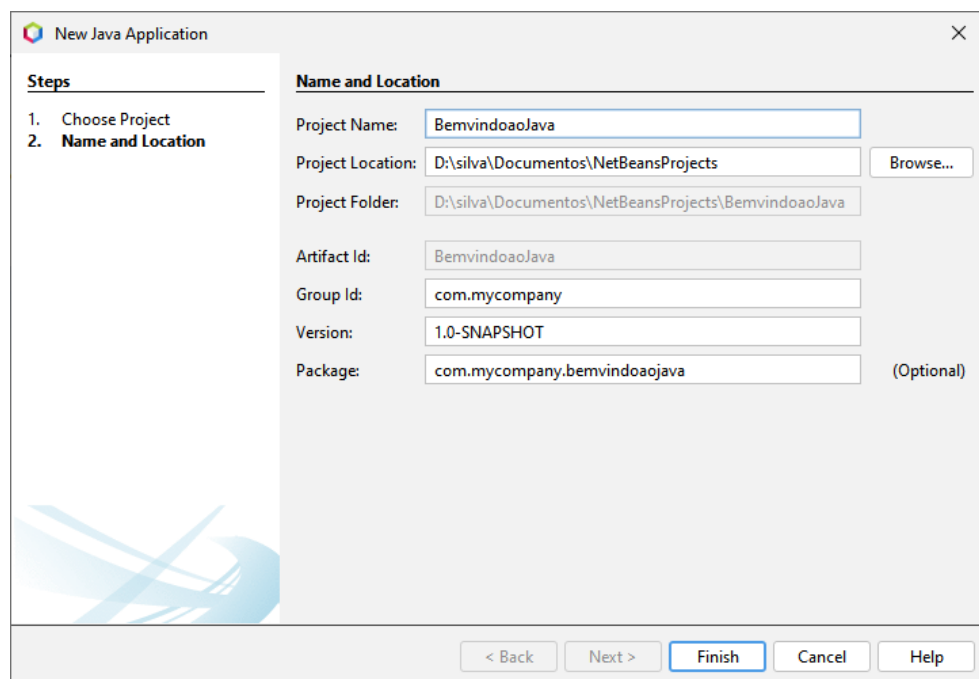


Figura 3: Definindo o nome do projeto

**Nota:** O campo **Project Location:** poderá ser modificado com a localização preferida para a pasta contendo o projeto em desenvolvimento.

#### 4. Clique em Finish.

O projeto é criado automaticamente e aberto no IDE. Alguns dos painéis abertos no IDE são apresentados na Figura 4 e detalhados a seguir:

- O painel **Projects**, que contém uma visualização em árvore dos componentes do projeto, incluindo arquivos fonte, bibliotecas relacionadas com o

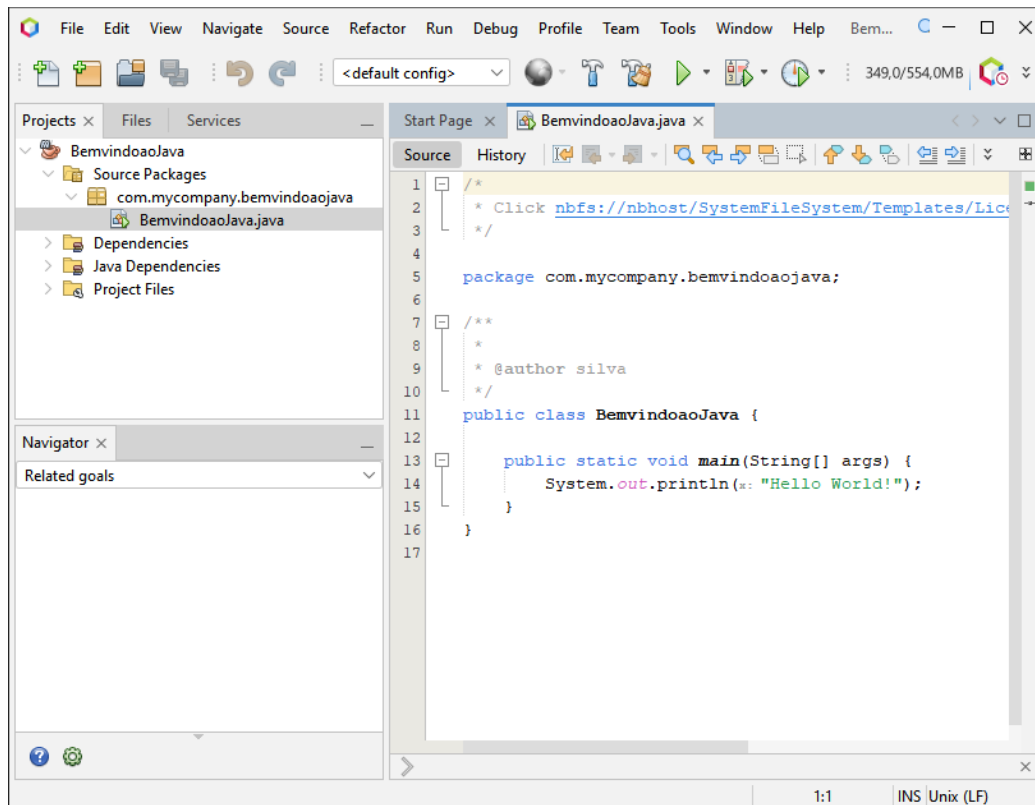


Figura 4: IDE NetBeans com o projeto aberto

código em construção, etc.

- O painel **Source Editor** com um arquivo chamado **BemvindoaoJava.java** – se você alterou o nome como sugerido no item 3 – aberto, ou com o nome da classe principal, seguido da extensão Java.
- O painel **Navigator**, que pode ser usado para a navegação rápida entre elementos internos a uma classe selecionada.

## Adicionar código no arquivo fonte gerado

O IDE automaticamente criou o esqueleto de uma classe. Portanto, pode-se adicionar a mensagem “Sejam bem vindos ao Java!” a este esqueleto de código apenas substituindo a linha de código:

```
System.out.println("Hello World!");
```

pela linha

```
System.out.println("Sejam bem-vindos ao Java!");
```

### Notas importantes:

- Cuidado ao digitar o código, comandos e nomes de arquivos. A linguagem Java é sensível a letras maiúsculas (caixa alta ou *upper case*) e a letras minúsculas (caixa baixa ou *lower case*). Assim, **BemVindos** ≠ **bemvindos**.
- Toda mudança feita no código é salva escolhendo **File** → **Save** do menu principal ou apenas usando o atalho **Ctrl+S**. Salve o seu projeto.

## Compilar o arquivo fonte, gerando o arquivo com a extensão .class

Para compilar o arquivo-fonte, escolha **Run** → **Build Project** do menu principal do IDE, ou apenas tecla F11.

O programa terá sua compilação e execução bem sucedida se a saída terminar com a sentença **BUILD SUCCESS**. Porém, se ocorrer erro, como o de sintaxe, a saída terminará com a sentença **BUILD FAILURE**.

Ao construir o projeto, é gerado o arquivo **BemvindoaoJava.class**, formado por *bytecodes*. Para verificar a localização deste arquivo basta abrir o painel **Files** e expandi-la (algo como

... > target > classes > com > mycompany > bemvindoaojava > BemvindoaoJava.class)

Após a construção do projeto, basta executar o programa como explicado a seguir.

## Executar o programa

No menu principal do IDE escolher **Run** → **Run Project** ou tecla F6.

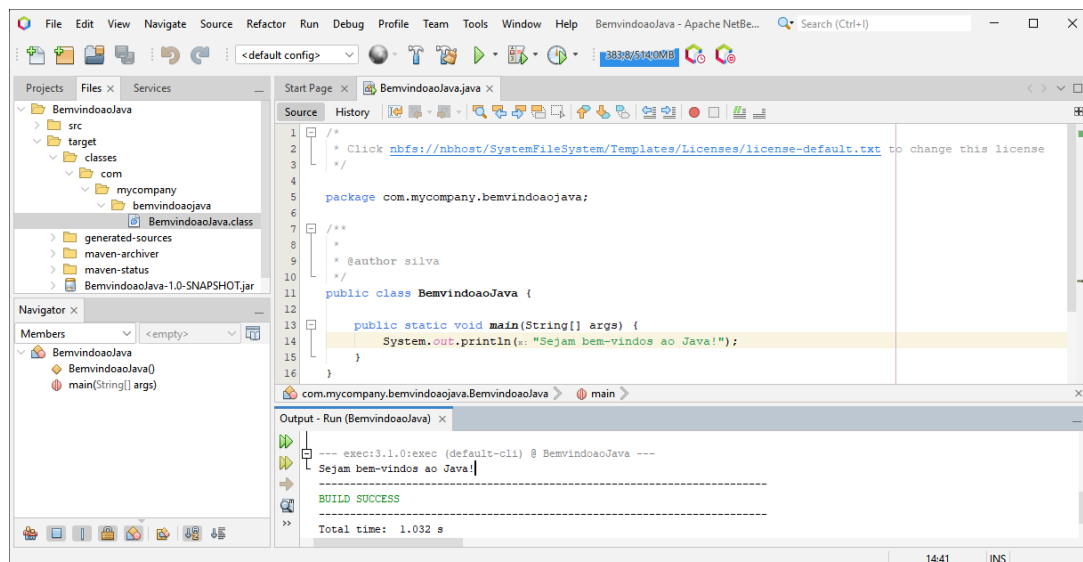


Figura 5: Executando o projeto

## 3 Exercícios

1. Escrever um programa em Java que leia dois números, calcule a soma e imprima o seu resultado na tela.
2. Crie um programa em Java que peça um número real ao usuário e armazene este número na variável de nome  $x$ . Depois peça outro número real e armazene na variável  $y$ . Mostre esses números na tela. Em seguida, troque os valores das variáveis ( $x$  deve receber o valor de  $y$  e  $y$  deve receber o valor de  $x$ ). Mostre na tela os novos valores de  $x$  e  $y$ .

3. Escreva um programa em Java que solicita 7 números reais ao usuário, através de um laço for, e ao final mostra o maior entre os números digitados. Não é necessário armazenar todos os números.
4. Escreva um programa em Java que solicita 10 números reais ao usuário, através de um laço for, e ao final mostra os dois maiores entre os números digitados. Novamente, não é necessário armazenar os números, apenas os dois maiores.
5. Crie um programa em Java que peça um número inicial ao usuário, uma razão e calcule e mostre na tela os 20 primeiros termos de uma P.A.. Não é necessário armazenar todos os números. (Obs.:  $a_n = a_1 + (n - 1) * r$ )
6. Escreva um programa em Java que leia 5 números reais e encontre e mostre na tela o menor deles. Depois, some este valor a todos os números e mostre na tela estes números na mesma linha, separados por vírgula, com exatamente duas casas decimais.

### 3.1 Sugestão de leitura – qual versão utilizar?

<https://www.stackchief.com/blog/Which%20Version%20of%20Java%20Should%20You%20Use%3F>