

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DO RIO GRANDE DO NORTE
CAMPUS JOÃO CÂMARA

LINGUAGENS DE PROGRAMAÇÃO

Nickerson Fonseca Ferreira
nickerson.ferreira@ifrn.edu.br

Introdução

2

- Já vimos como criar os algoritmos e suas formas de representação
 - Linguagem natural
 - Fluxograma
 - Pseudo-código
- Mas como iremos fazer para que o computador entenda nossos algoritmos?

Introdução

3

- Também já vimos que os computadores só entende a linguagem de máquina.
- A linguagem de máquina é a sequência de instruções, representadas por bits.
- Essa sequência de bits é conhecido como código binário.
- O código binário é formado por 0 e 1.

00010001

01010010

00110011

Introdução

4

- Quais os problemas desse tipo de linguagem ?
 - Muito difíceis de programar
 - Cada computador possui seu próprio conjunto de instruções.
- Então foram criadas as linguagens de alto nível.
 - Mais compatíveis com a linguagem humana
 - Exemplo JAVA, Pascal, COBOL, C, etc.
- Normalmente uma única instrução de uma linguagem de alto nível equivale a várias instruções da linguagem de máquina.

Introdução

5

- OK, mas se os computadores só entendem a linguagem de máquina ?? O que acontece com as linguagens de alto nível ??
- Um programa escrito com linguagens de alto nível precisam passar por um processo de tradução antes de serem executados.
- Essa tradução é conhecida como **compilação** ou **interpretação**.

Introdução

6

```
media.c
#include <stdio.h>

int main(void)
{
    float nota1, nota2, media;

    scanf("%f", &nota1);
    scanf("%f", &nota2);

    media = (nota1 + nota2)/2;

    printf("%f", media);

    return 0;
}
```

Programa-fonte



Tradução

Compilador

ou

Interpretador

```
101010100000010100100101000101
10100001010100010010001010000
.01000110001000001010000101001
100110100010101010110011000101
.00000101000001001000100101010
10000101011000010011001011100
.01101001001001001010010101010
101010100000010100100101000101
10100001010100010010001010000
.01000110001000001010000101001
100110100010101010110011000101
.00000101000001001000100101010
10000101011000010011001011100
.01101001001001001010010101010
```

Programa-objeto

A linguagem JAVA

7

- Um programa escrito em JAVA consiste em um ou mais métodos/funções.
- A primeira função que será executada no programa **SEMPRE** será a **main**.
- Cada método/função deve ter:
 - modificador de acesso: informa o nível de “visibilidade” do método, ou seja, quem pode usá-lo.
 - Uma lista de parâmetros (opcional): podemos informar que aquela função recebe valores externos.
 - Um bloco de instruções.

A linguagem JAVA

8

```
package algoritmos;

/**
 *
 * @author ferreira
 */
public class Algoritmos {

    public static void main(String[] args) {
        System.out.println(" E ai, galera ?! ");
    }

}
```


A linguagem JAVA

9

- Cada bloco de instrução é colocado entre chaves { e }.
- Cada linha de comando deve terminar com um ponto-e-vírgula (;).
- Letras maiúsculas e minúsculas NÃO são iguais.
- Para criar os algoritmos utilizando Java, vamos precisar de uma ferramenta chamada Ambiente de Desenvolvimento Integrada (IDE – Integrated Development Environment)



Utilizando o Netbeans

10

The screenshot displays the NetBeans IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. A search bar on the right of the menu bar contains the text "Search (Ctrl+I)". Below the menu bar is a toolbar with various icons for file operations and development actions. The main workspace is divided into two panes. The left pane, titled "Pro...", shows a project explorer with a tree view of the project structure. The right pane, titled "Start Page", displays the NetBeans IDE logo and navigation links: "Learn & Discover", "My NetBeans", "What's New", and "Show On Startup". The "Learn & Discover" section is active, showing a "Take a Tour" button and a list of "Demos & Tutorials" including Java SE Applications, Java and JavaFX GUI Applications, Java EE & Java Web Applications, C/C++ Applications, PHP Applications, and Mobile and Embedded Applications. A "Featured Demo" section shows a preview of an "HTML5Application" project structure with folders for "css", "img", "js", and "lib", and a file named "index.html". The Oracle logo and the Java logo are visible at the bottom of the interface.

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

304.8/376.7MB

Pro... Files Services Favorit... Start Page

AnagramGame8

j-html5-game5-code

- Site Root
 - images
 - js
 - behaviors
 - requestNextAnimationFrame.js
 - sprites.js
 - game.css
 - game.html
 - game.js

JsfJpaCrud

- Web Pages
- Source Packages
 - jpa.controllers
 - jpa.controllers.exceptions
 - jpa.entities
 - jsf
 - jsf.util
 - JsfCrudELResolver.java
 - JsfUtil.java
 - PagingInfo.java
- Libraries
- Configuration Files
- Server Resources
- langtools
- WebApplication

NetBeans IDE

Learn & Discover

My NetBeans

What's New

Show On Startup

Learn & Discover

Take a Tour

Try a Sample Project

What's New

Community Corner

Demos & Tutorials

- Java SE Applications
- Java and JavaFX GUI Applicat
- Java EE & Java Web Applicati
- C/C++ Applications
- PHP Applications
- Mobile and Embedded Applic

All Online Documentation >>

Featured Demo

Projects Files Ser

HTML5Application

- Site Root
 - css
 - img
 - js
 - lib
 - index.html

Working with JavaScript in HT

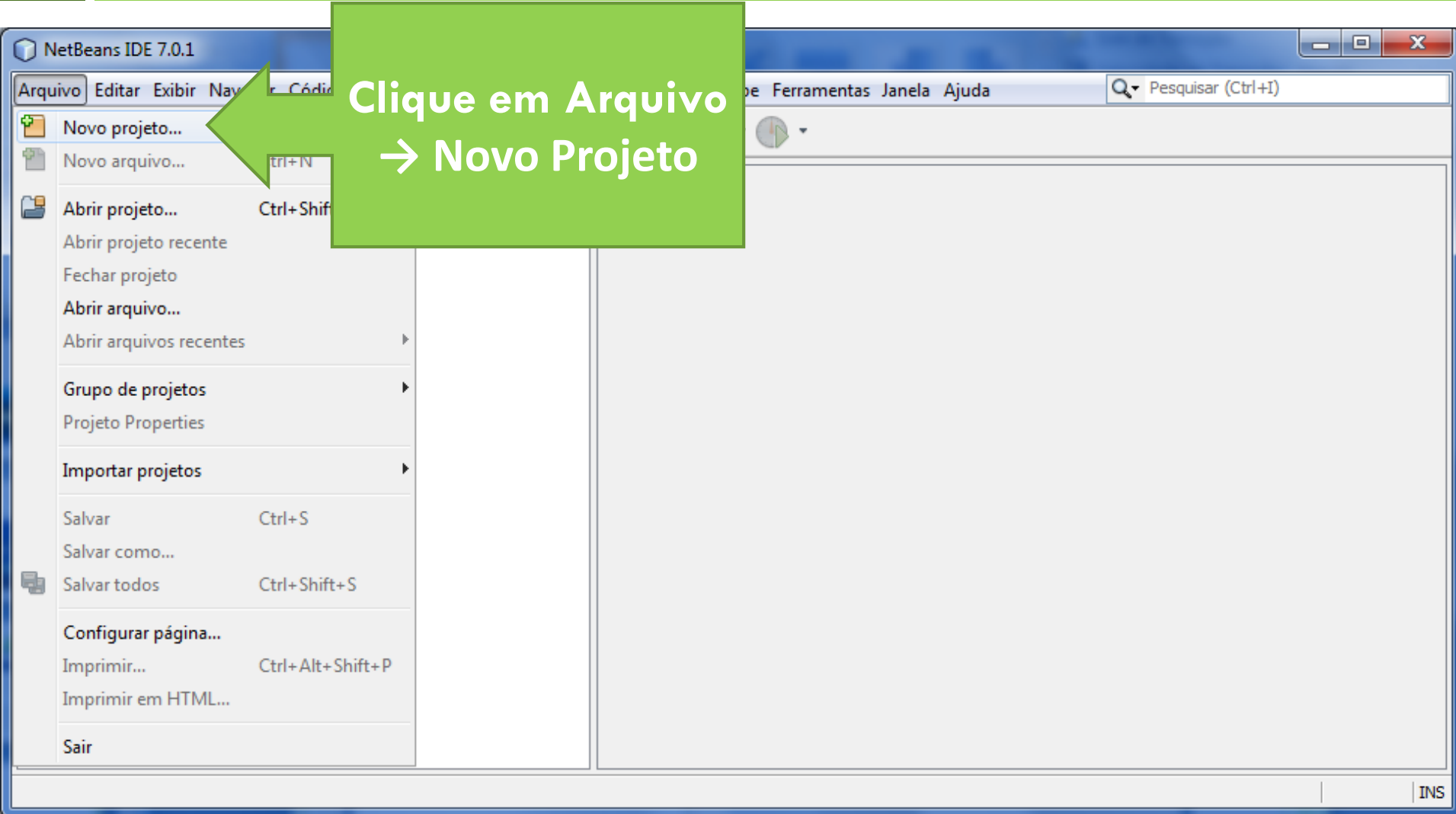
ORACLE

Java

Criando um projeto

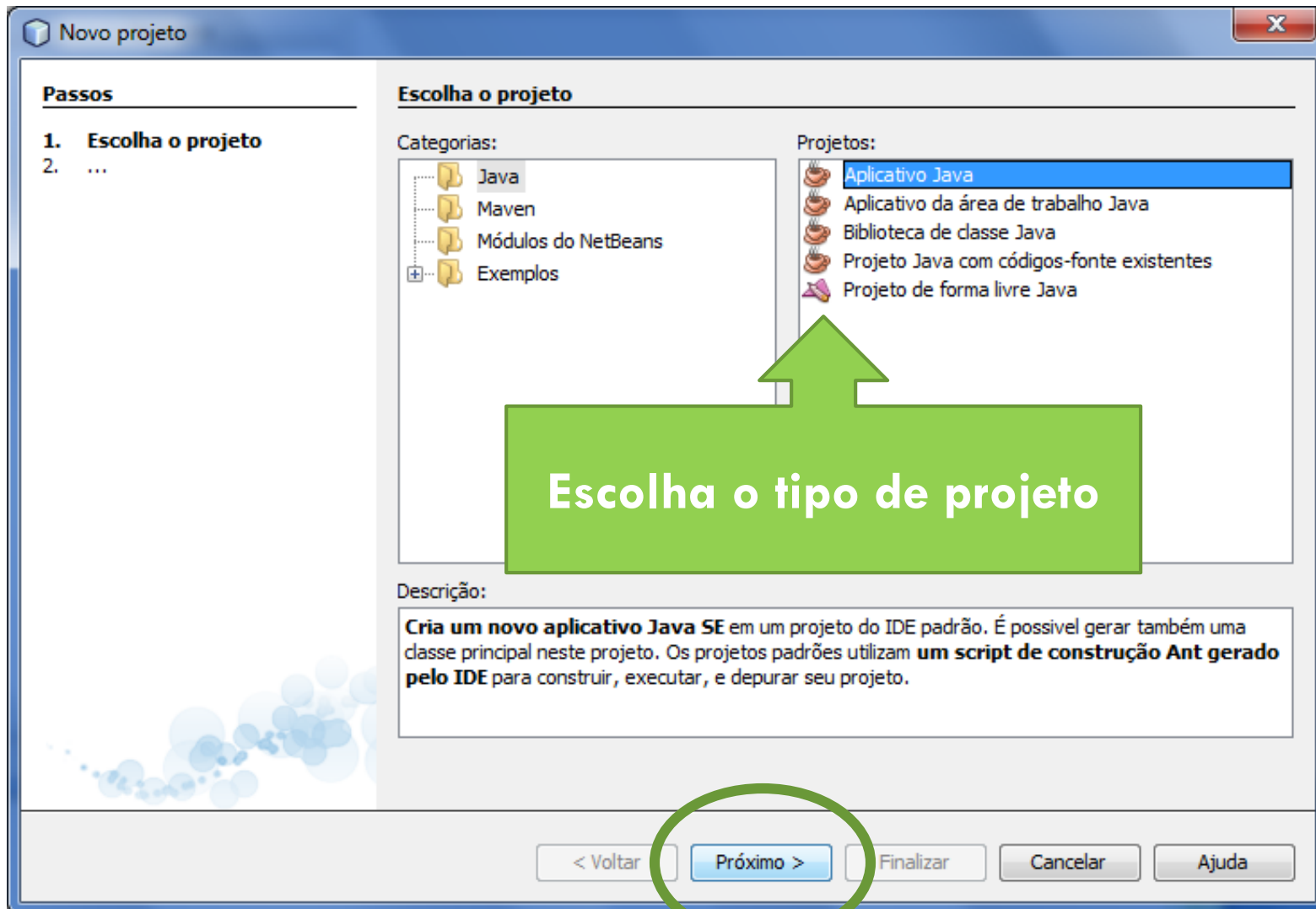
11

Clique em Arquivo
→ Novo Projeto



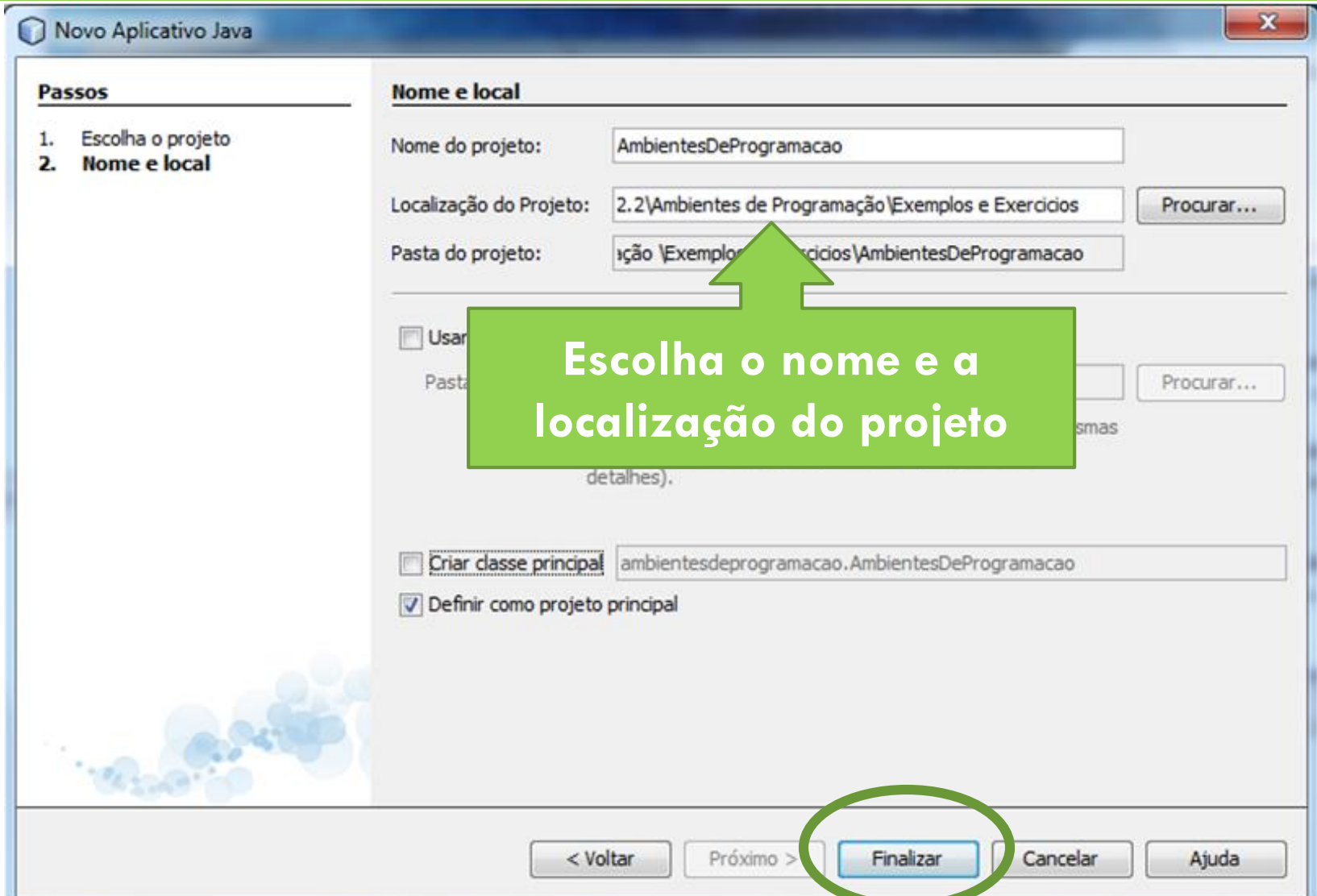
Criando um projeto

12



Criando um projeto

13



Criando um projeto

14

The image shows the NetBeans IDE 7.0.1 interface. The main window displays a project named 'ProjetoJava1'. The 'Arquivos' (Files) view on the left shows the project structure: 'Pacotes de código-fonte' (Source Packages) containing 'projetojava1' and 'ProjetoJava1.java', and 'Bibliotecas' (Libraries). The 'Tarefas' (Tasks) view at the bottom shows a task: 'TODO code application logic here' located in 'ProjetoJava1.../...ava1/ProjetoJava1.java'. A callout circle highlights the 'AmbientesDeProgramacao' (Development Environments) tree view, which shows 'Pacotes de código-fonte' and 'Bibliotecas'.

```
1  /*
2  * To ch
3  * and c
4  */
5  package
6
7  /**
8  *
9  * @author
10 */
11 public class
12
13 /**
14 * @param a
15 */
16 public static void
17 // TODO code
18 }
19
20
```

Descrição	Arquivo	Localização
TODO code application logic here	ProjetoJava1...	...ava1/ProjetoJava1.java

TODO: 1 em todos os projetos abertos

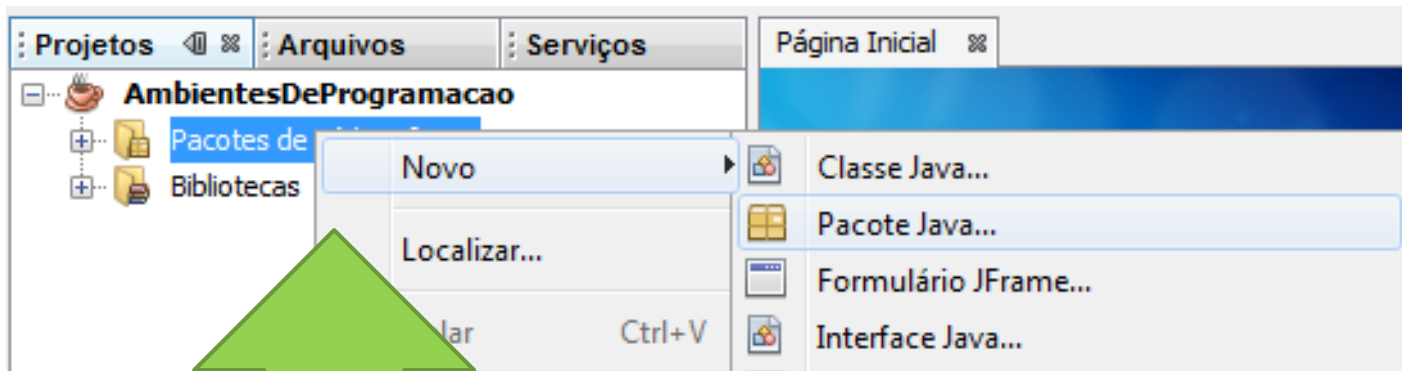
Organização dos projetos

15

- ❑ Um projeto possui uma ou várias classes.
- ❑ Essas classes estão separadas em pacotes, para organizar melhor o projeto.
- ❑ Cada pacote possui classes de uma determinada finalidade. Ex: `br.edu.ifrn.gui`
- ❑ O Netbeans vai criar subpastas para cada nível do pacote. Ex: `NomeProjeto/src/br/edu/ifrn/gui`
- ❑ Por padrão, a nomenclatura dos pacotes utiliza o domínio reverso. Ex: `br.edu.ifrn.aula01`

Criando um pacote

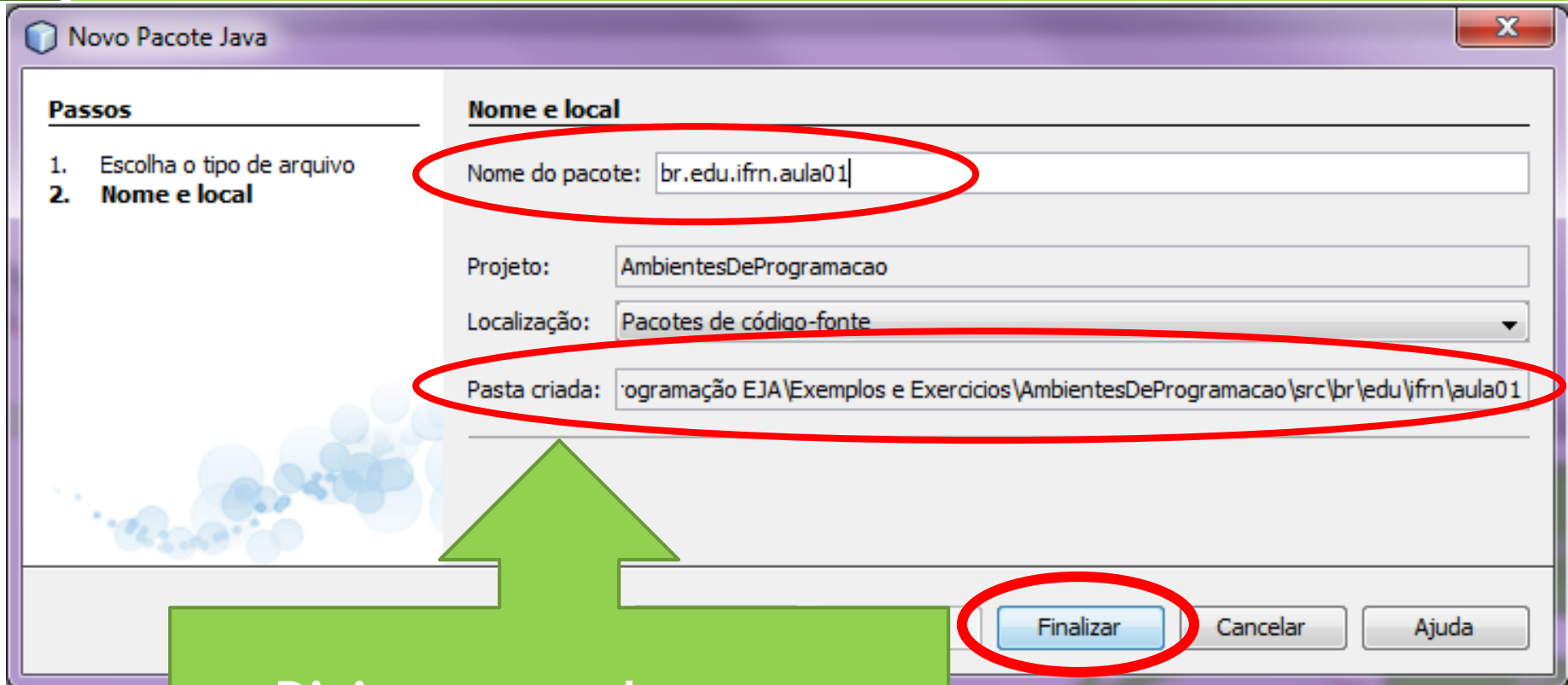
16



**Na aba Projetos, selecione
Pacotes de código-fonte
utilizando o botão direito do
mouse → Novo → Pacote Java**

Criando um pacote

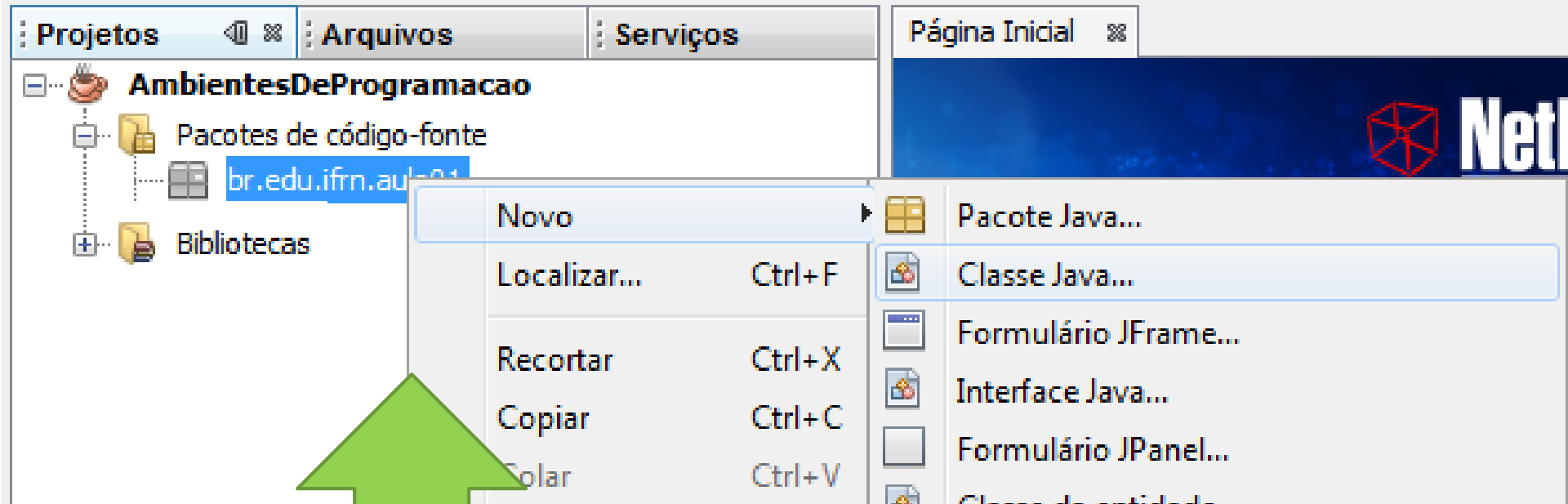
17



**Digite o nome do pacote.
Perceba a pasta que o NetBeans
cria para o seu pacote.**

Criando uma classe

18



Selecione o pacote e clique com o botão direito. Selecione Novo → Classe Java...

Criando uma classe

19

Novo Classe Java

Passos

1. Escolha o tipo de arquivo
2. **Nome e local**

Nome e local

Nome da classe: OlaMundo|

Projeto: AmbientesDeProgramacao

Localização: Pacotes de código fonte

Pacote: br.edu.ifrn.aula01

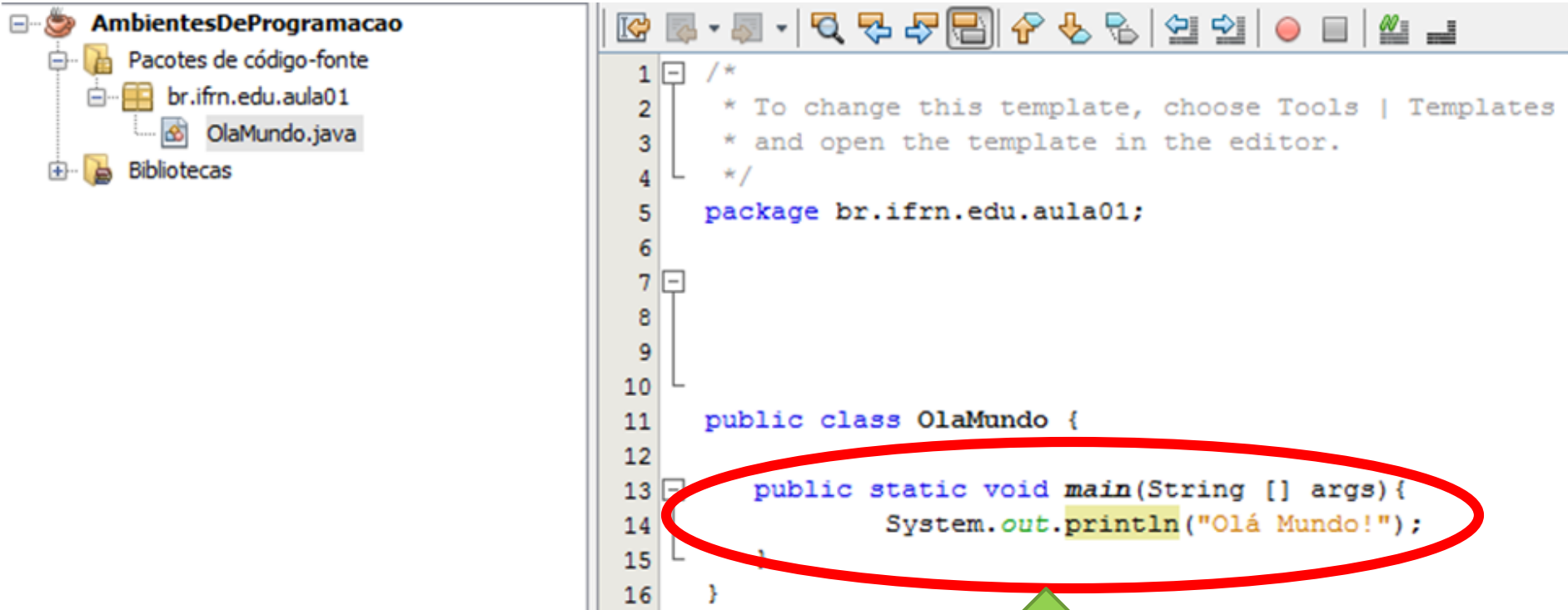
Arquivo criado: xemplos e Exercicios\AmbientesDeProgramacao\src\br\edu\ifrn\aula01\OlaMundo.java

< Voltar Próximo > **Finalizar** Cancelar Ajuda

**Digite o nome da classe.
Perceba o pacote e a pasta onde
a classe será criada.**

Executando a classe criada

20



The screenshot shows an IDE interface. On the left, a project tree under 'AmbientesDeProgramacao' contains 'Pacotes de código-fonte', 'br.ifrn.edu.aula01', and 'Bibliotecas'. Under 'br.ifrn.edu.aula01', the file 'OlaMundo.java' is highlighted. The main editor displays the following Java code:

```
1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5  package br.ifrn.edu.aula01;
6
7
8
9
10
11 public class OlaMundo {
12
13     public static void main(String [] args){
14         System.out.println("Olá Mundo!");
15     }
16 }
```

A red circle highlights the `main` method in lines 13-15. A green arrow points from a text box below to this circle.

Para executar uma classe Java é necessário implementar o método main.