

Automatização de Teste com a Ferramenta EclEmma/JaCoCo

Prof. Simone do Rocio Senger de Souza

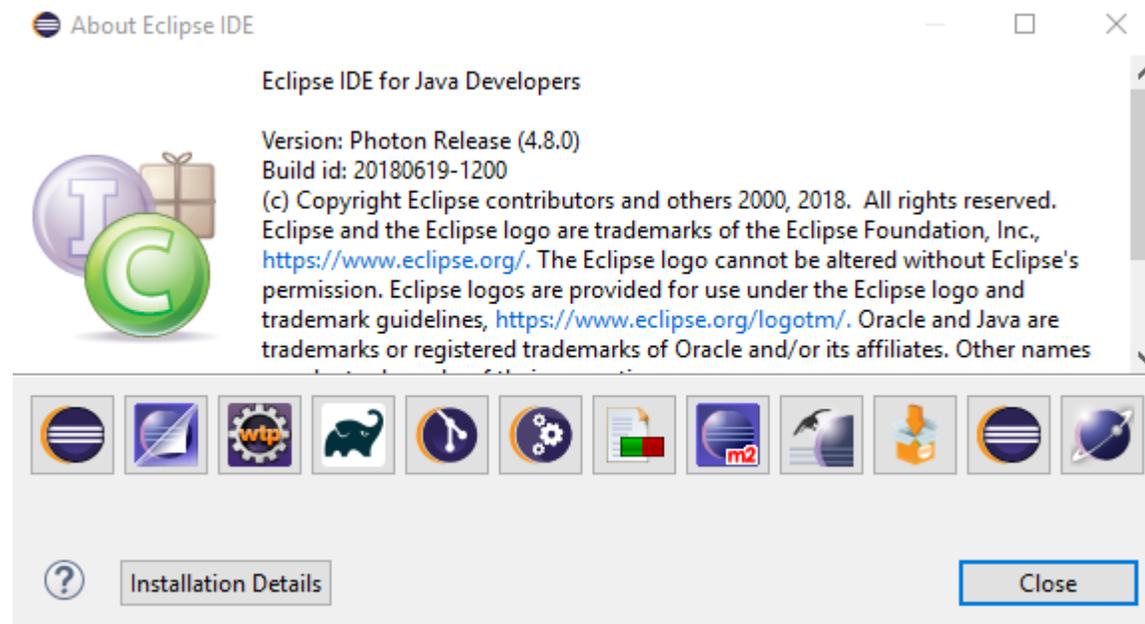
Estagiário PAE: Leo Natan Paschoal

Como instalar?

- Instalar a versão mais recente do Eclipse

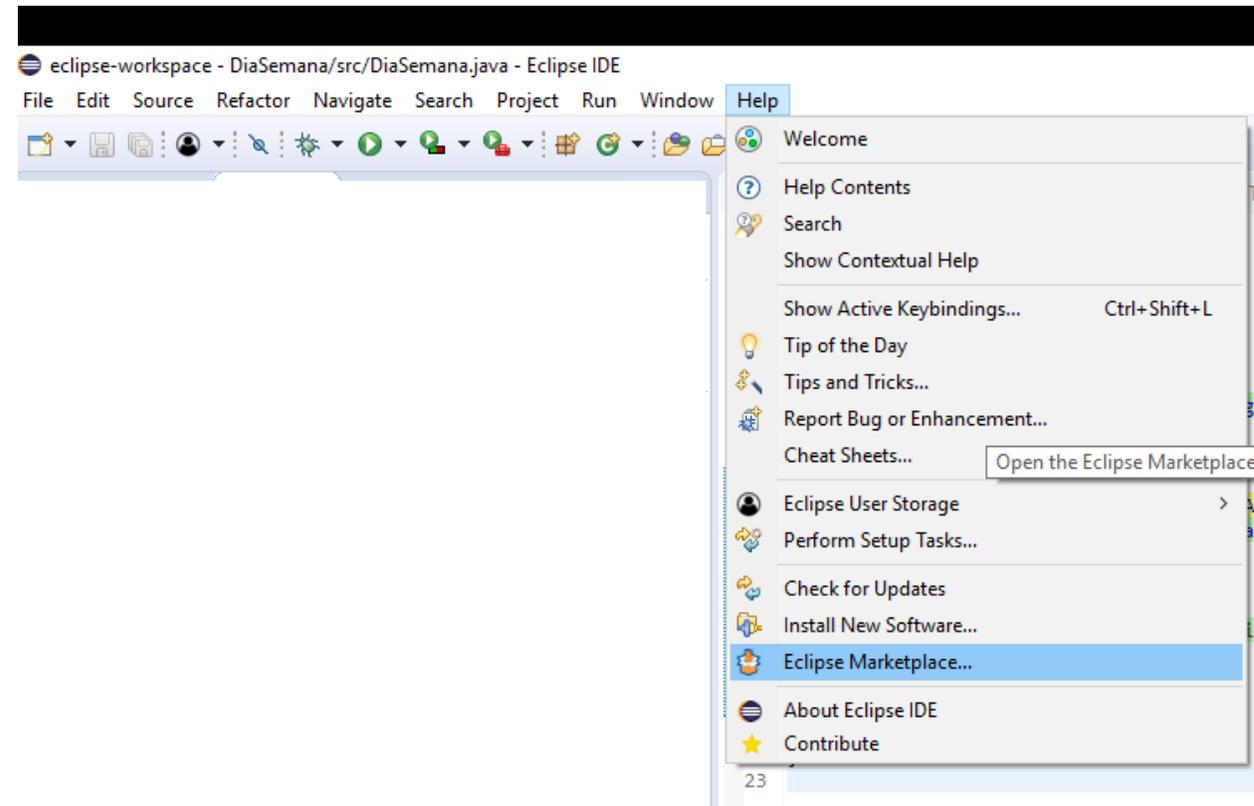
Disponível em: <<https://www.eclipse.org/downloads/>>

Versão escolhida para a aula:



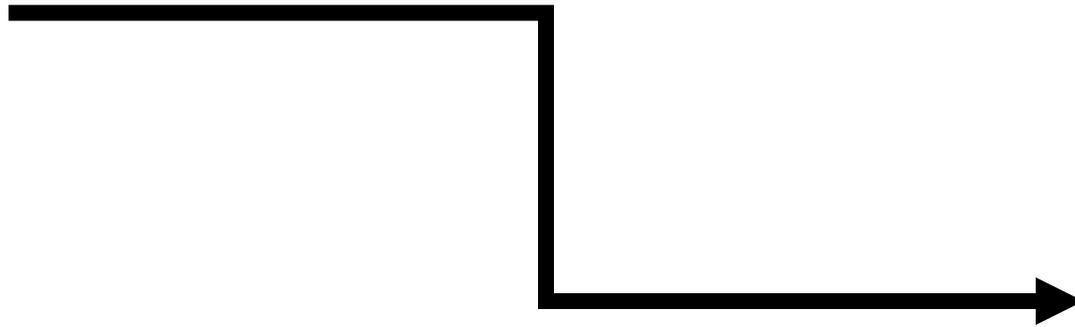
Como instalar?

- Após a Instalação do Eclipse IDE, é necessário localizar a ferramenta EclEmma
- Passos para a instalação:
 - Executar o eclipse IDE;
 - Acessar o menu Help;
 - Escolher a opção “Eclipse Marketplace...”;
 - Veja o exemplo na imagem 



Como instalar?

- Esperar a janela carregar

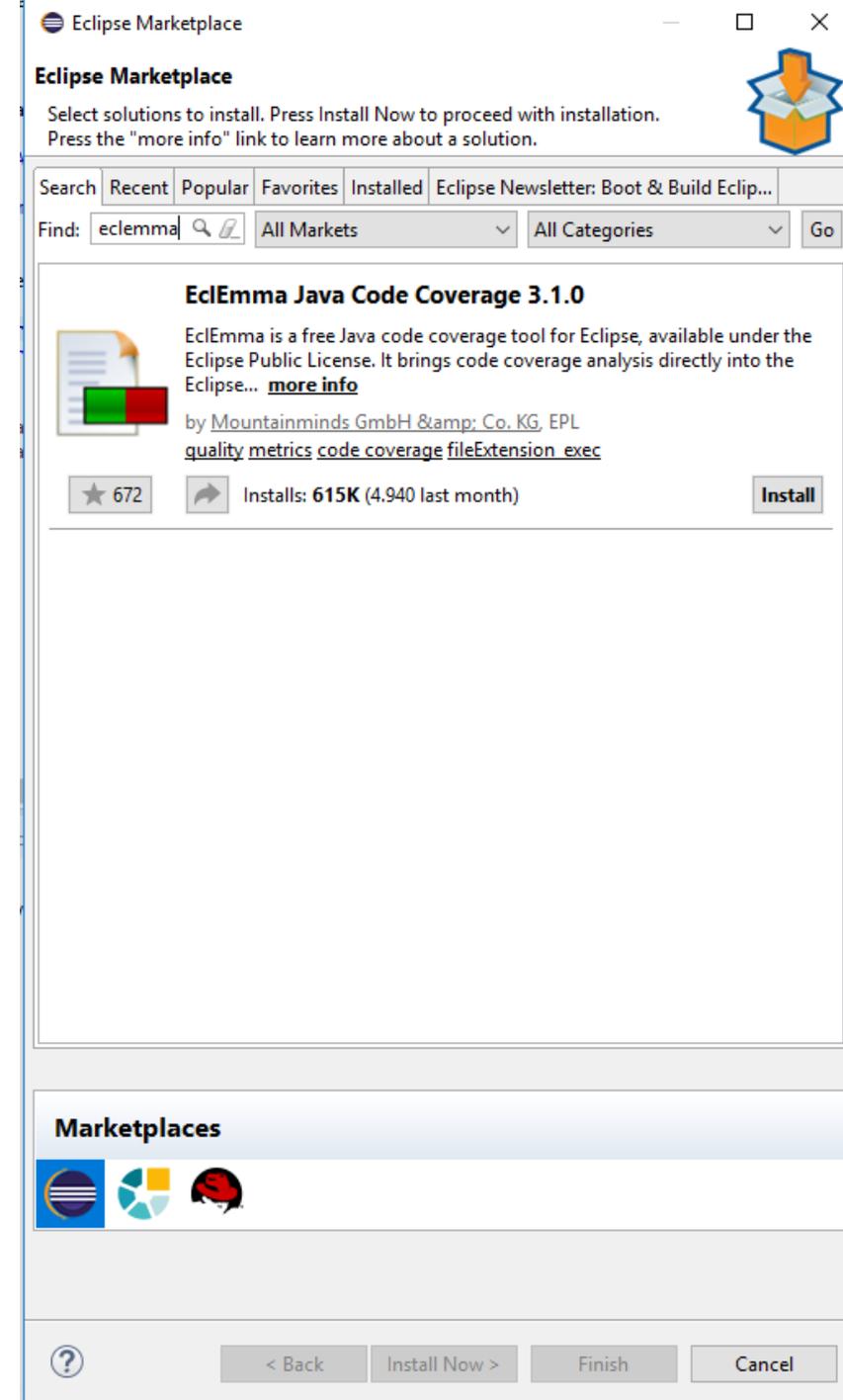
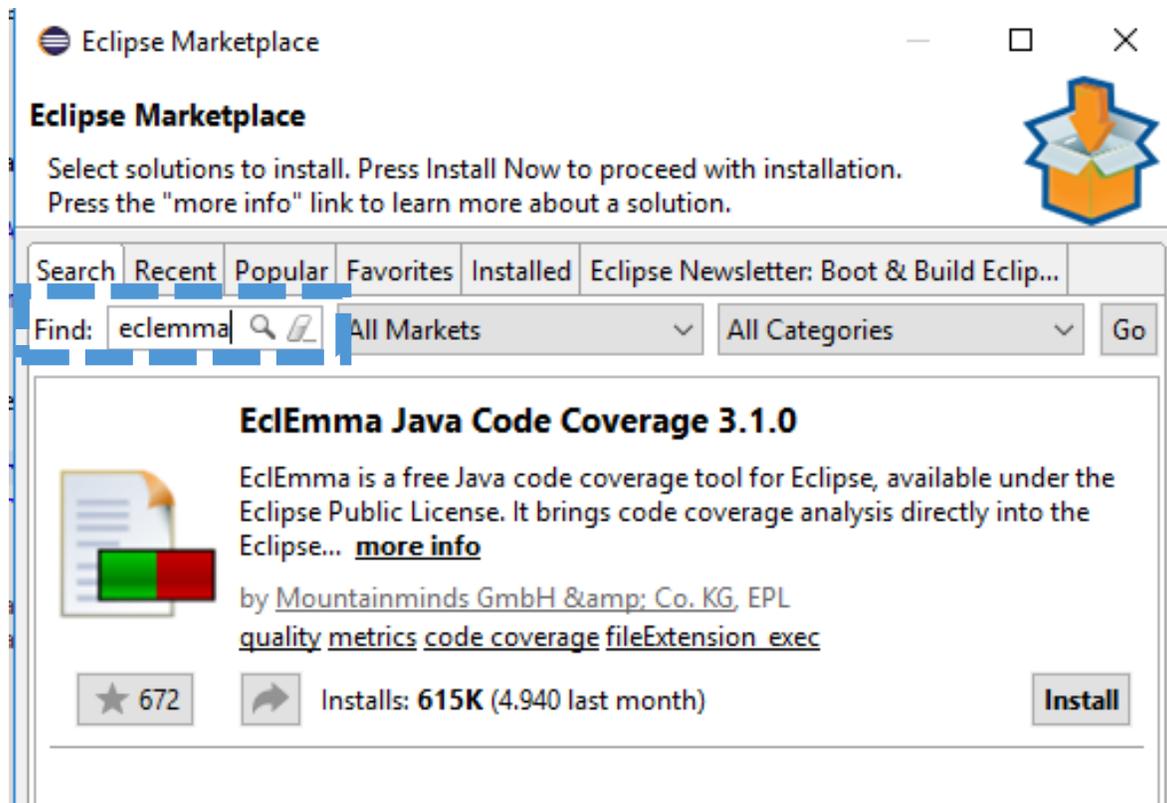
The screenshot shows the Eclipse Marketplace interface. At the top, it says "Eclipse Marketplace" and "Select solutions to install. Press Install Now to proceed with installation. Press the 'more info' link to learn more about a solution." Below this is a search bar and navigation tabs: "Search", "Recent", "Popular", "Favorites", "Installed", and "Eclipse Newsletter: Boot & Build Eclip...". The main content area is titled "Featured" and lists three plugins:

- CodeMix CI 2018.8.15**: Promoted - With CodeMix, unlock a wide array of technologies from Visual Studio Code and add-on extensions built for Code OSS directly inside your Eclipse IDE. First... [more info](#). by Genuitec, LLC, Commercial. VSCoDe editor PHP Python go ... 100 stars, 43,2K installs (9,678 last month). [Install](#)
- Vaadin Plugin for Eclipse 4.0.2**: Promoted - Vaadin Framework is an open source Java UI library for creating rich web user interfaces. Using its component based API developers can create stunning web... [more info](#). by Vaadin Ltd, Apache 2.0. java J2EE web ria java ee ... 194 stars, 194K installs (1,523 last month). [Install](#)
- Trema 0.3.2**: Trema is an Eclipse plugins that allows you to manage your text resources and translations. It allows you to keep all text resources and translations in a single... [more info](#). hv Netretera, MIT.

At the bottom, there is a "Marketplaces" section with icons for different marketplaces. The footer contains a help icon, a "< Back" button, an "Install Now >" button, a "Finish" button, and a "Cancel" button.

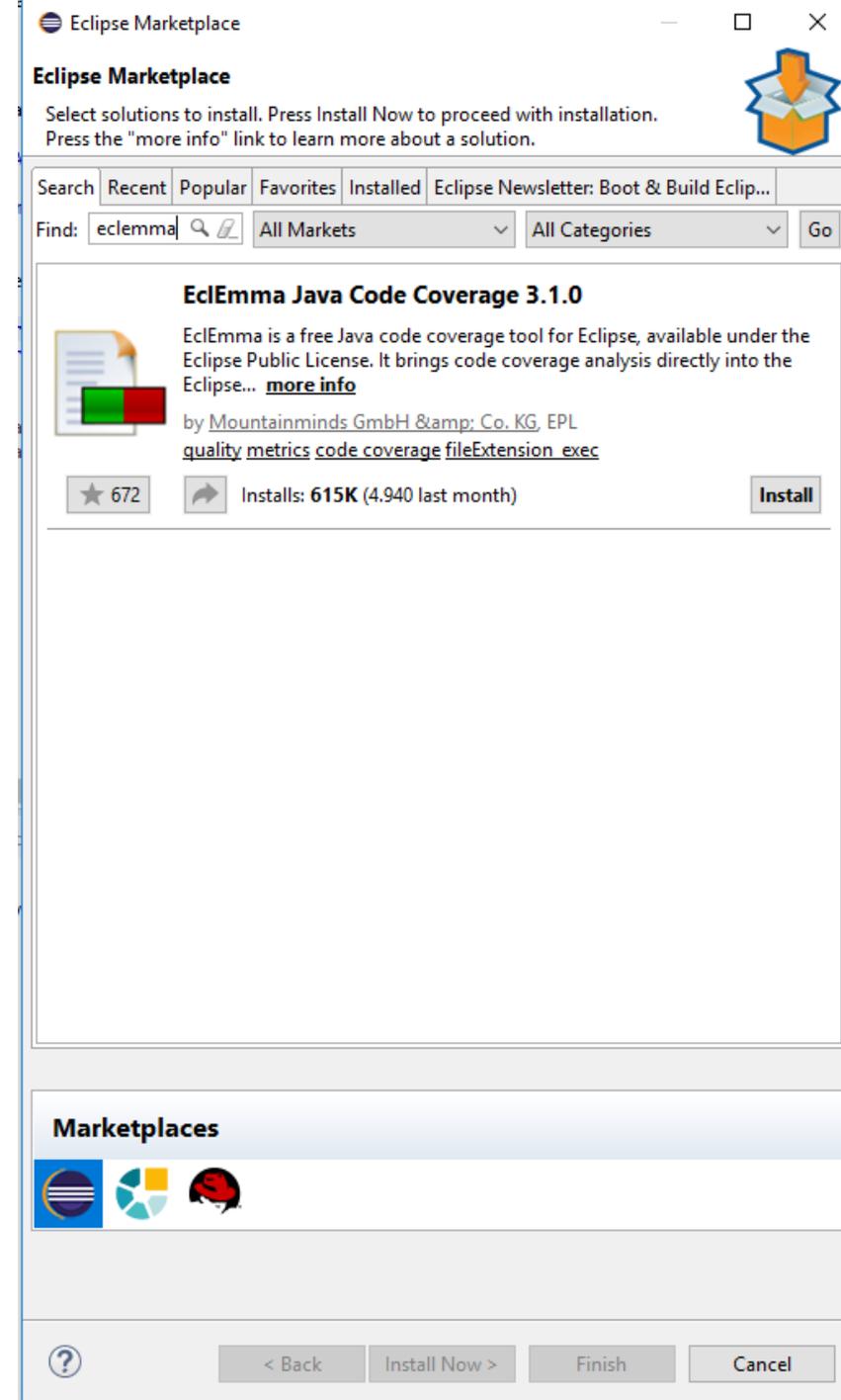
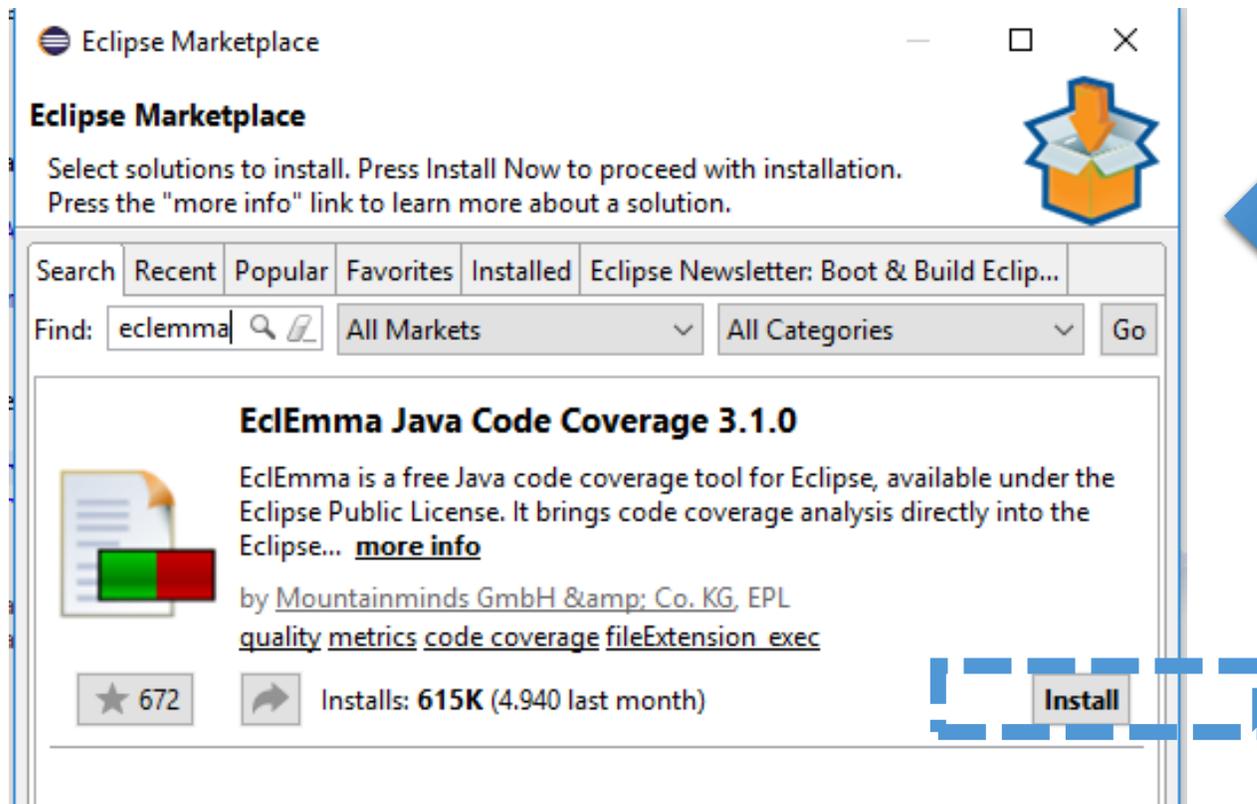
Como instalar?

- Fazer a busca pela ferramenta
 - Procure por EclEmma



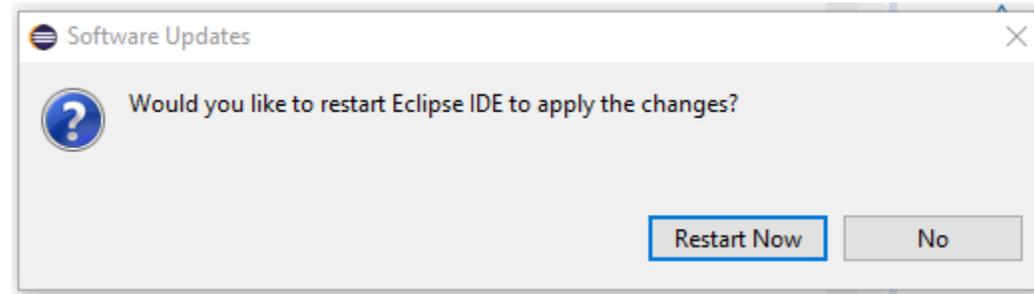
Como instalar?

- Clique em instalar



Como instalar?

- Aguarde até o Eclipse pedir para reiniciar ...



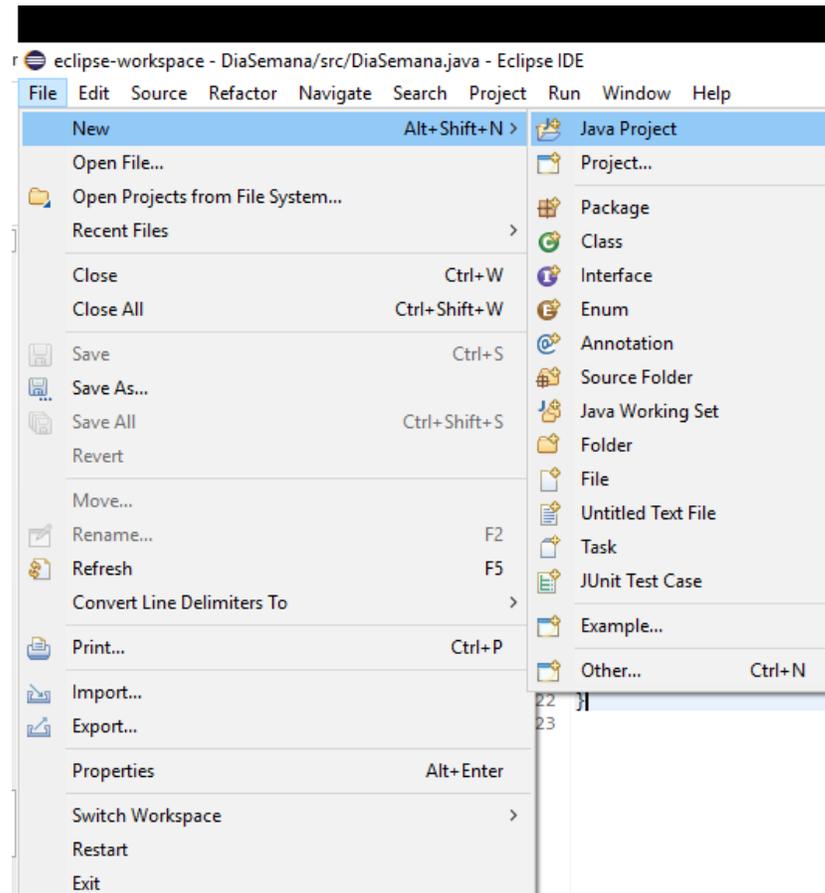
- Cliquei em reiniciar

Pronto!

- A Ferramenta EclEmma/JaCoCo está instalada
- Lembre-se que a ferramenta só funciona em conjunto com o JUnit. Então, ao criar um novo projeto no Eclipse IDE, é necessário adicionar a biblioteca Junit.

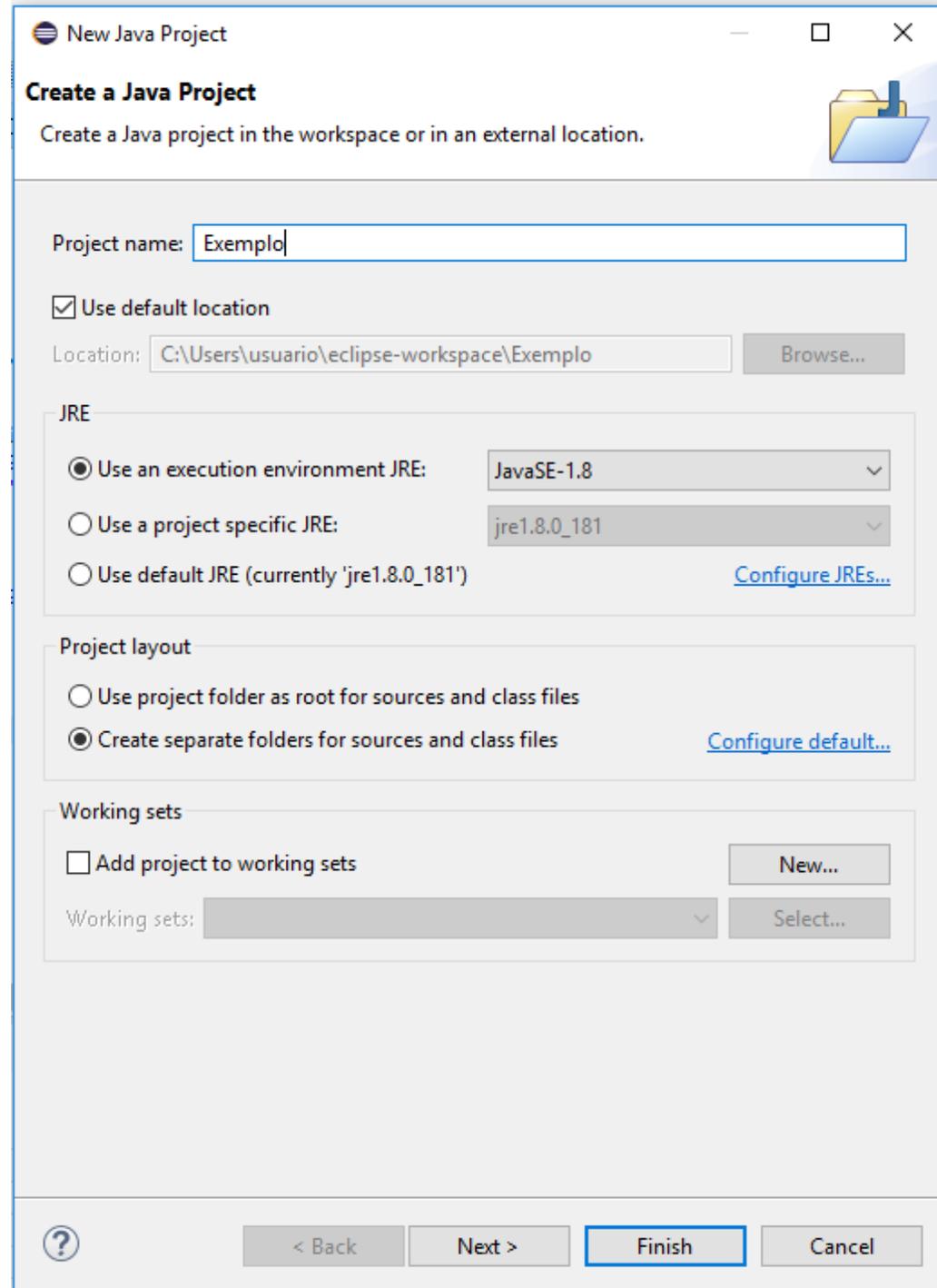
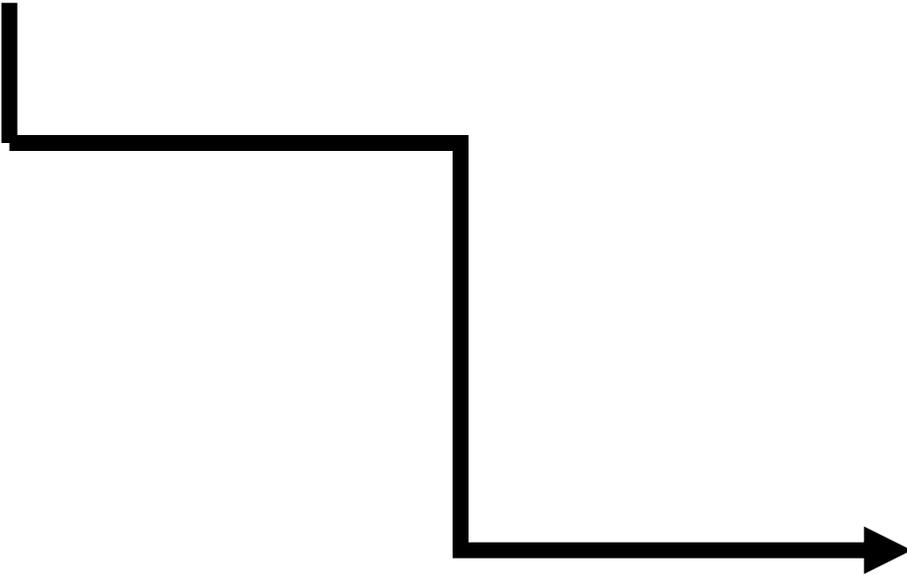
JUnit no Eclipse

- Crie um novo projeto na IDE Eclipse



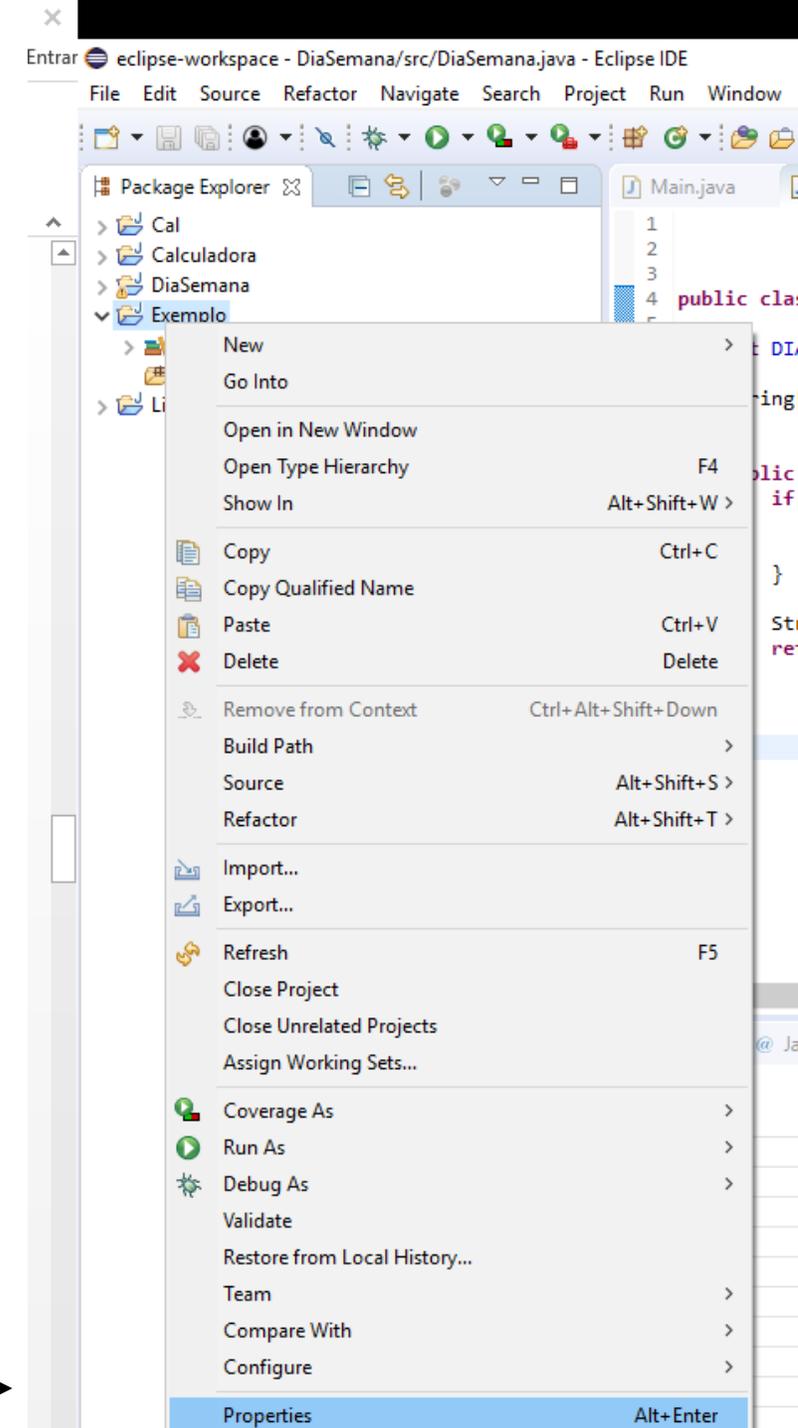
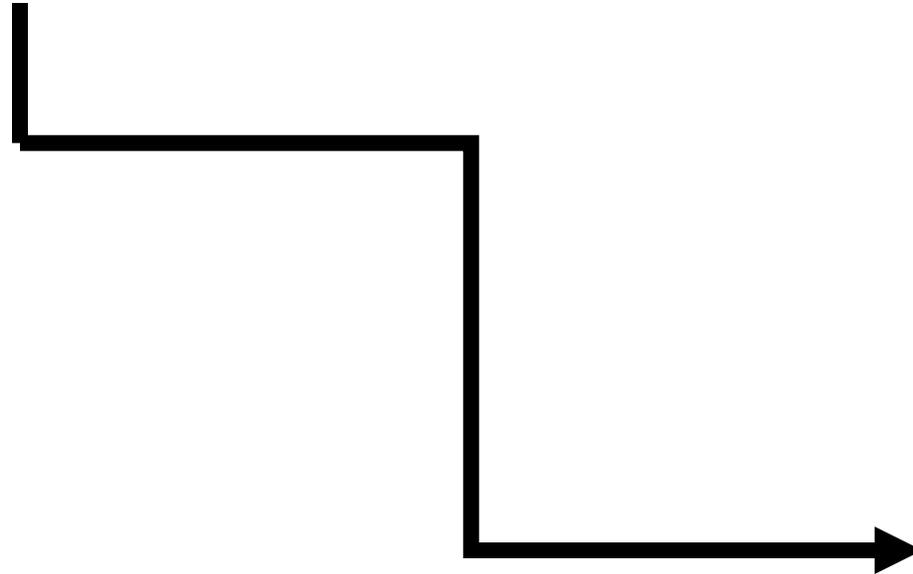
JUnit no Eclipse

- Adicione o nome ao projeto e clique em Finalizar



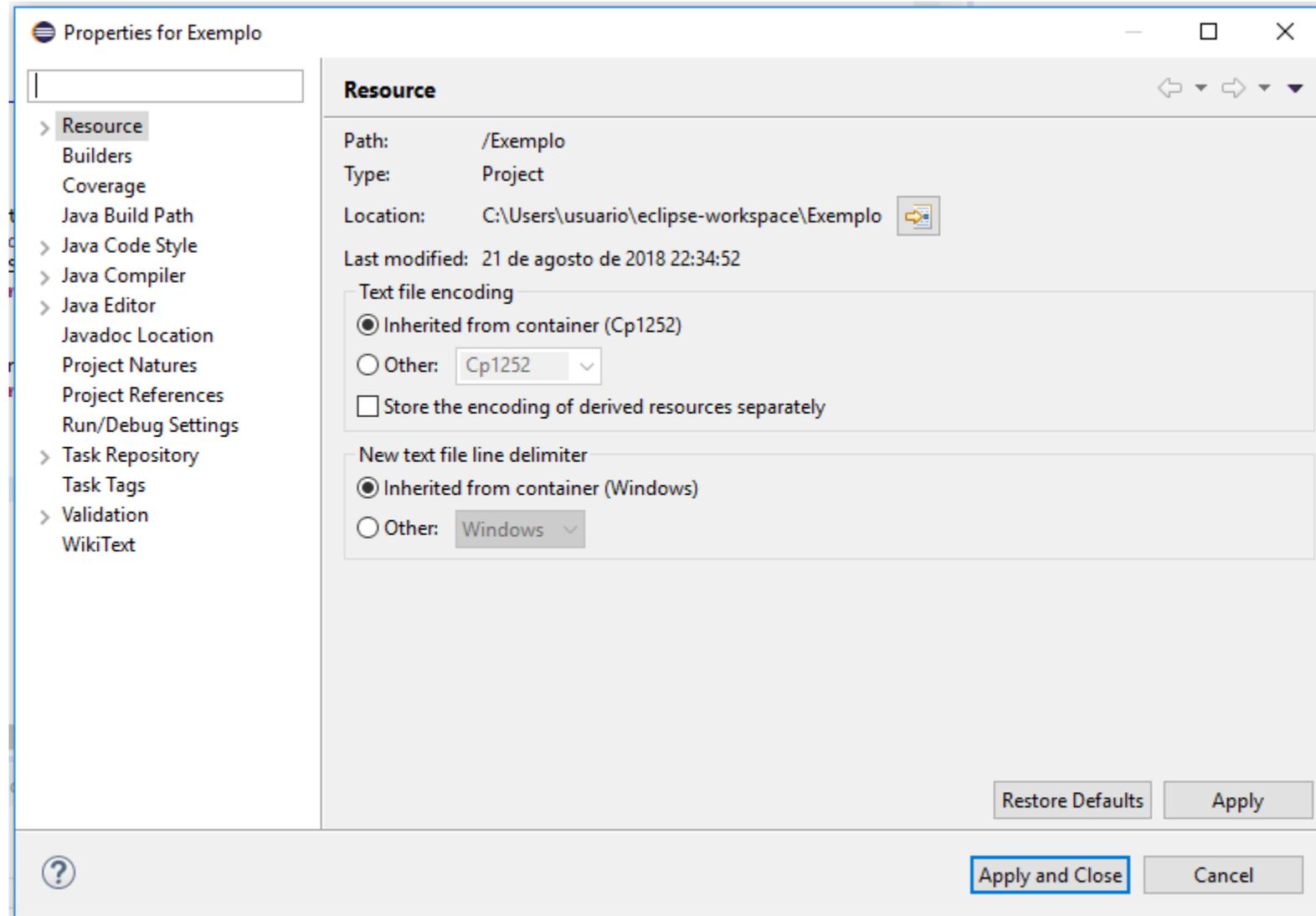
JUnit no Eclipse

- Após a criação:
 - Vá até a lista de pacotes
 - Selecione o projeto;
 - Clique com o botão direito;
 - No menu referente ao projeto, escolha propriedades



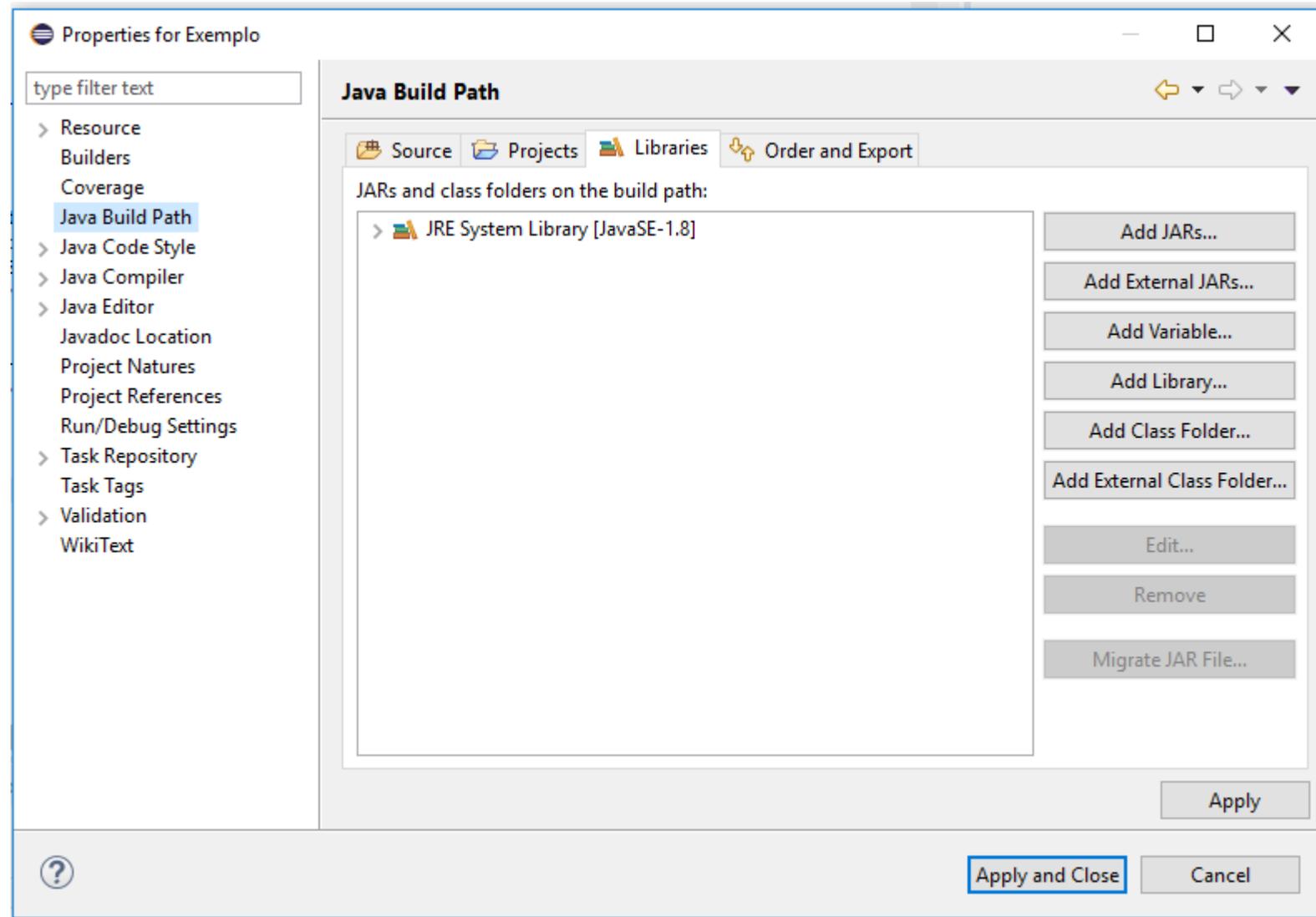
JUnit no Eclipse

- Na janela que será aberta, escolha a opção “Java Build Path”.



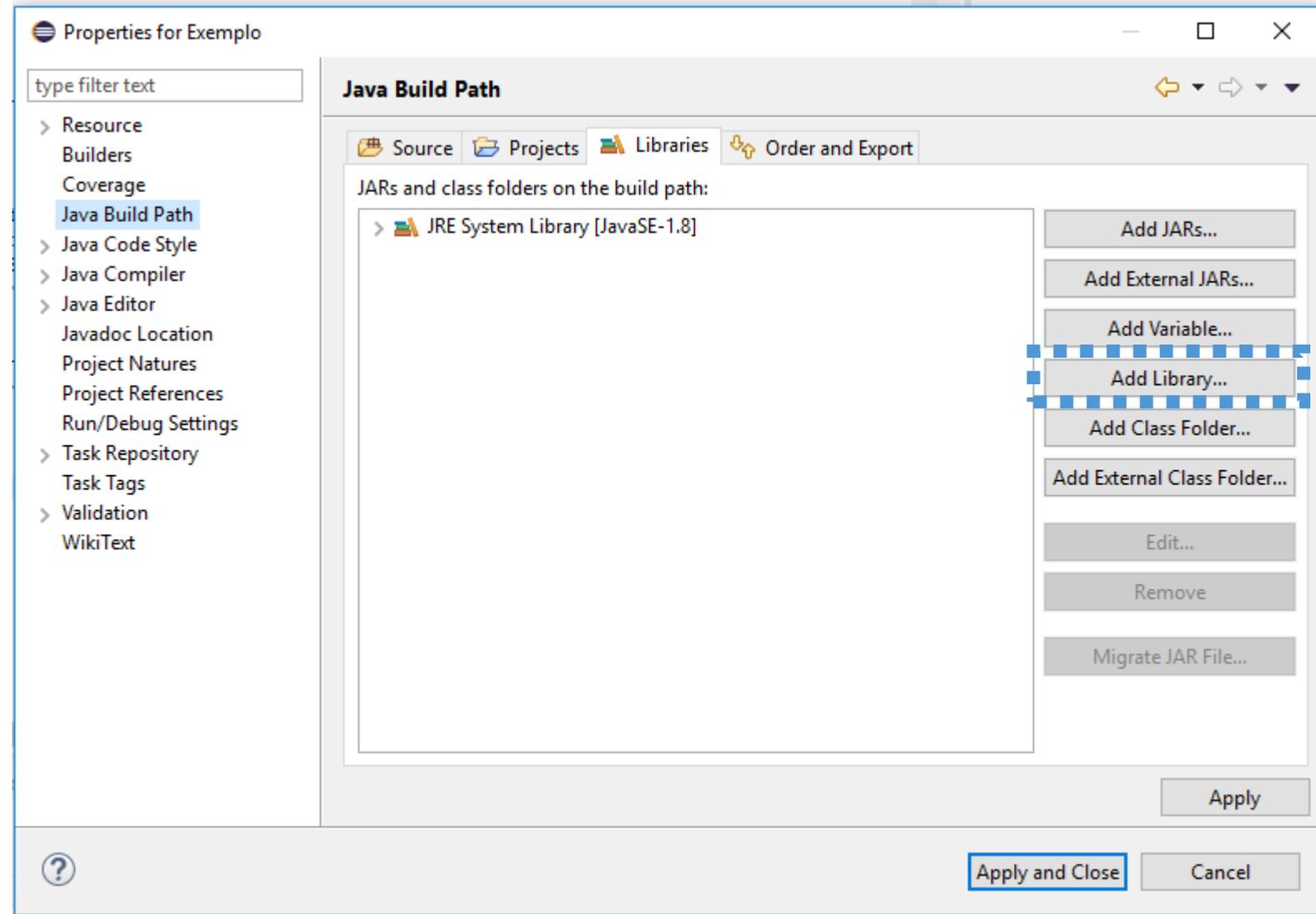
JUnit no Eclipse

- Observe as bibliotecas disponíveis para o projeto



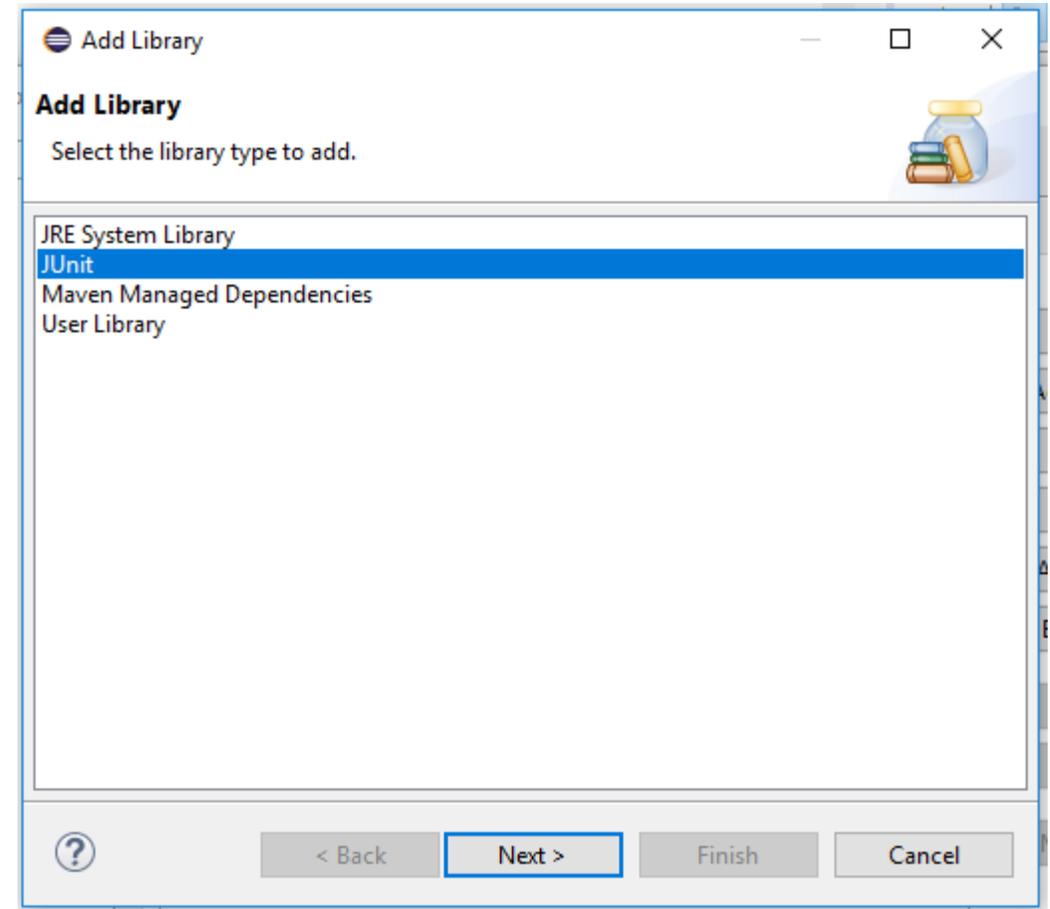
JUnit no Eclipse

- Clique no botão adicionar biblioteca



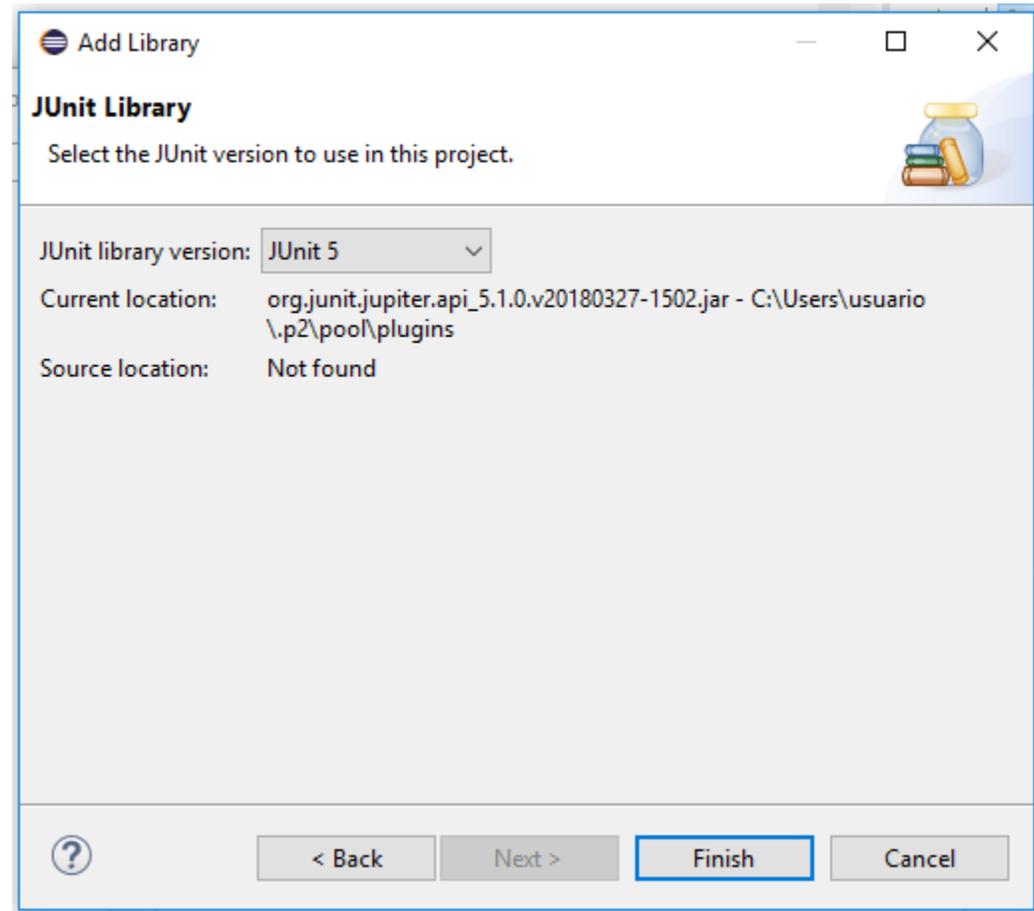
JUnit no Eclipse

- Na nova janela, selecione JUnit e clique em Próximo



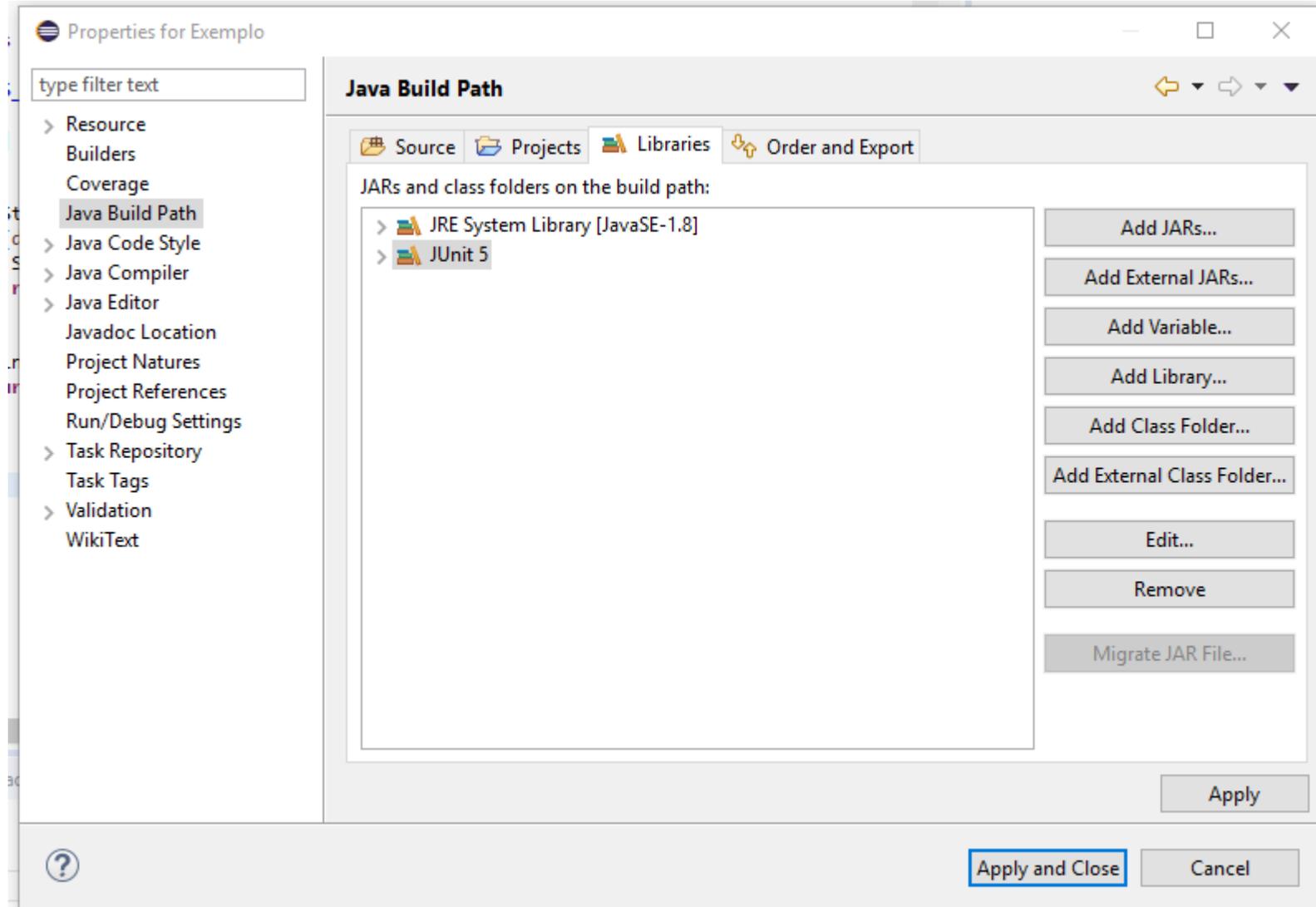
JUnit no Eclipse

- Na nova janela, selecione a versão do JUnit e clique em finalizar.



JUnit no Eclipse

- Por fim, clique em Aplicar e Fechar



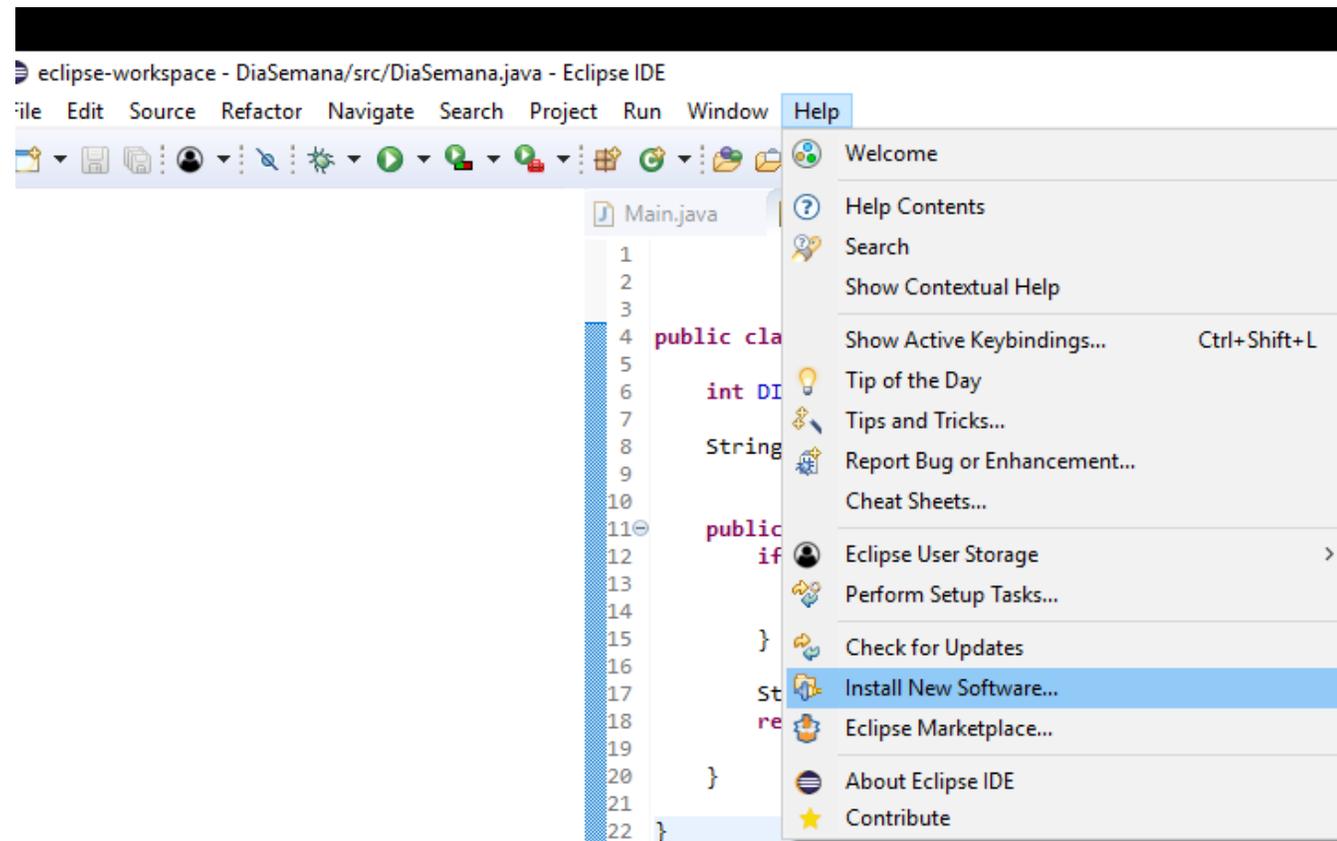
Pronto!

- O JUnit foi adicionado ao projeto!

GFC

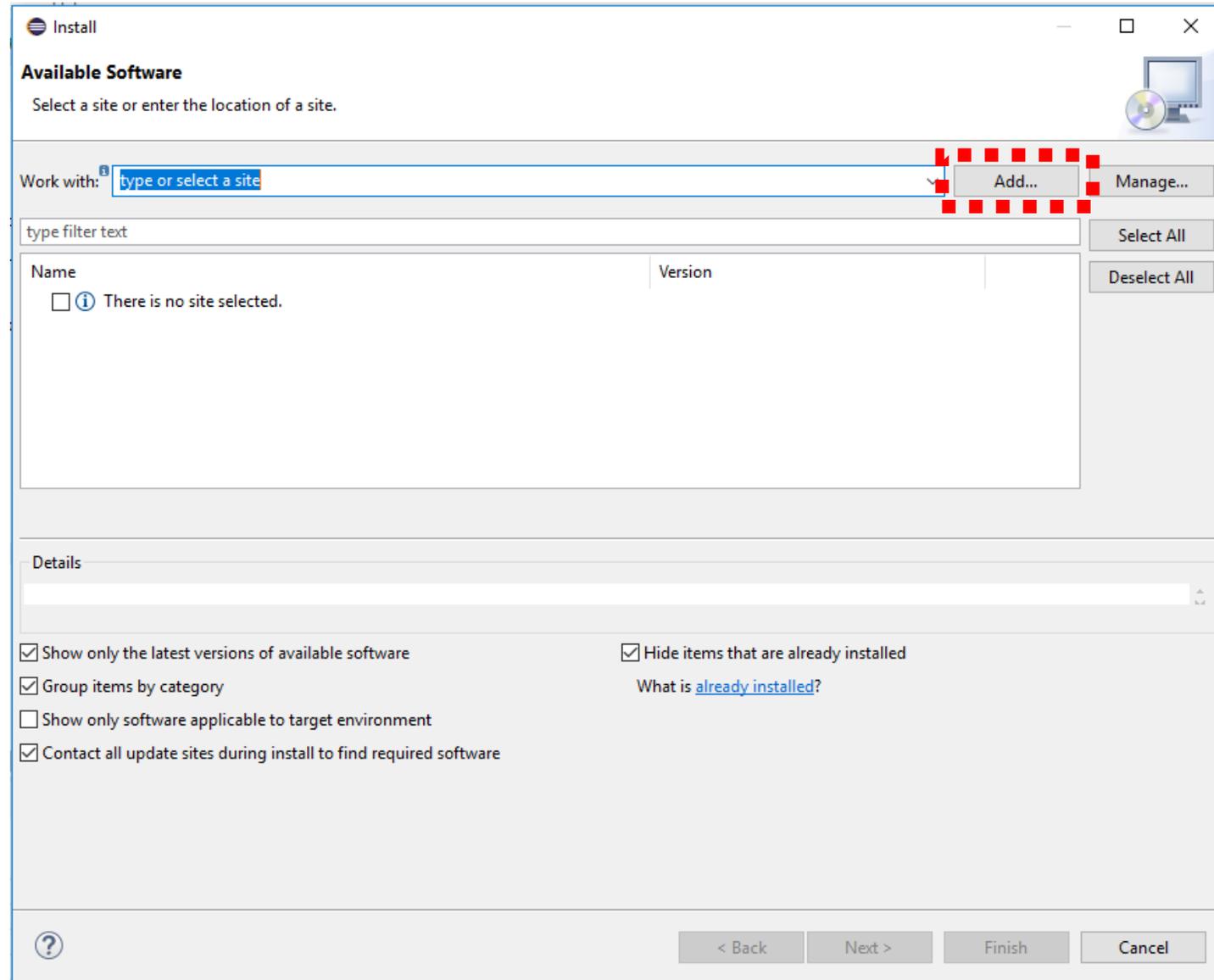
O Eclipse possui um *plugin* para gerar o GFC.... Para tanto:

- Executar o Eclipse
- No menu Help, acessar:
 - Install New Software ...



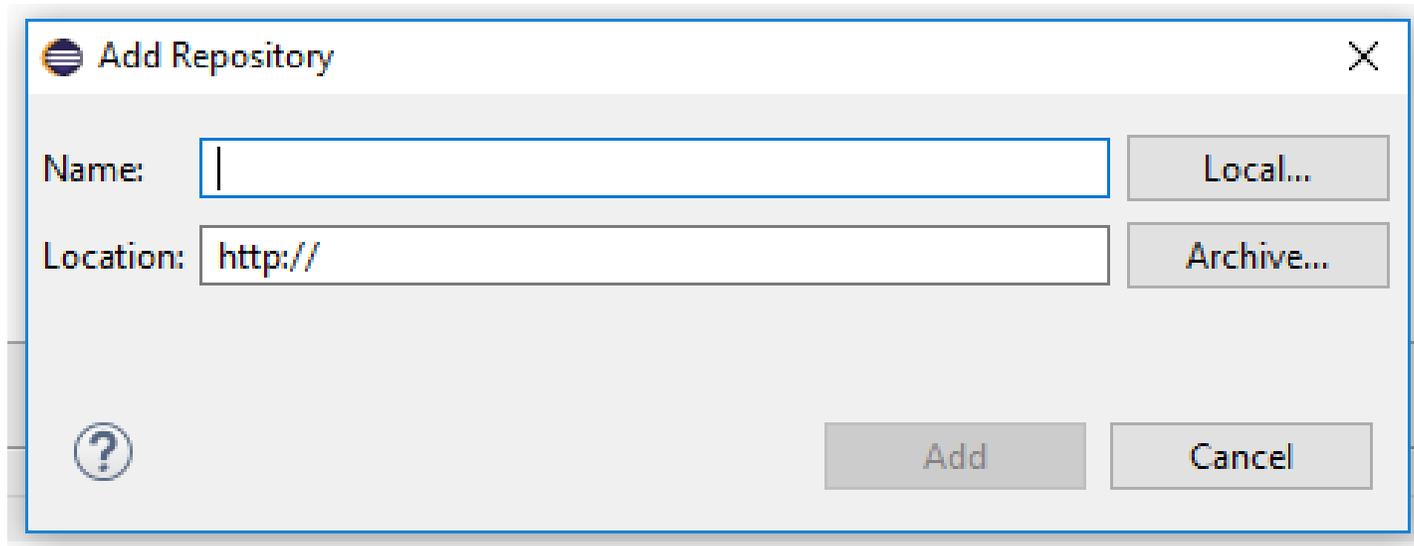
GFC

- Na janela aberta clique em:
Add...



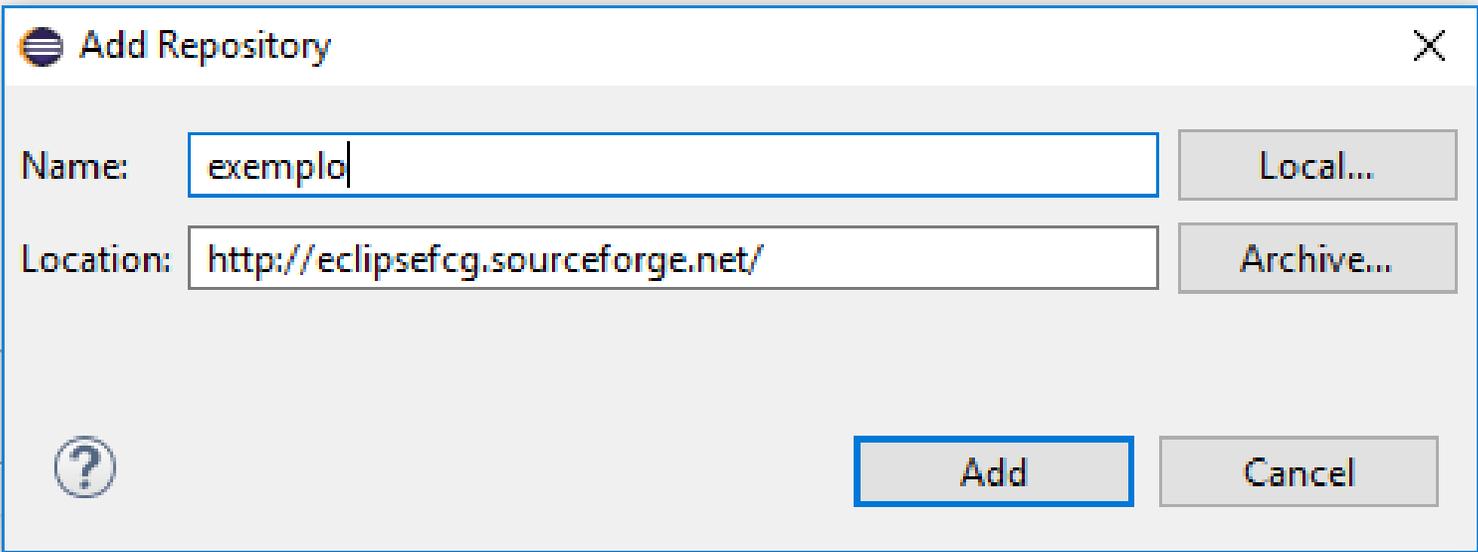
GFC

- Observe que uma nova janela foi aberta:



GFC

- Preencha os campos conforme a figura:

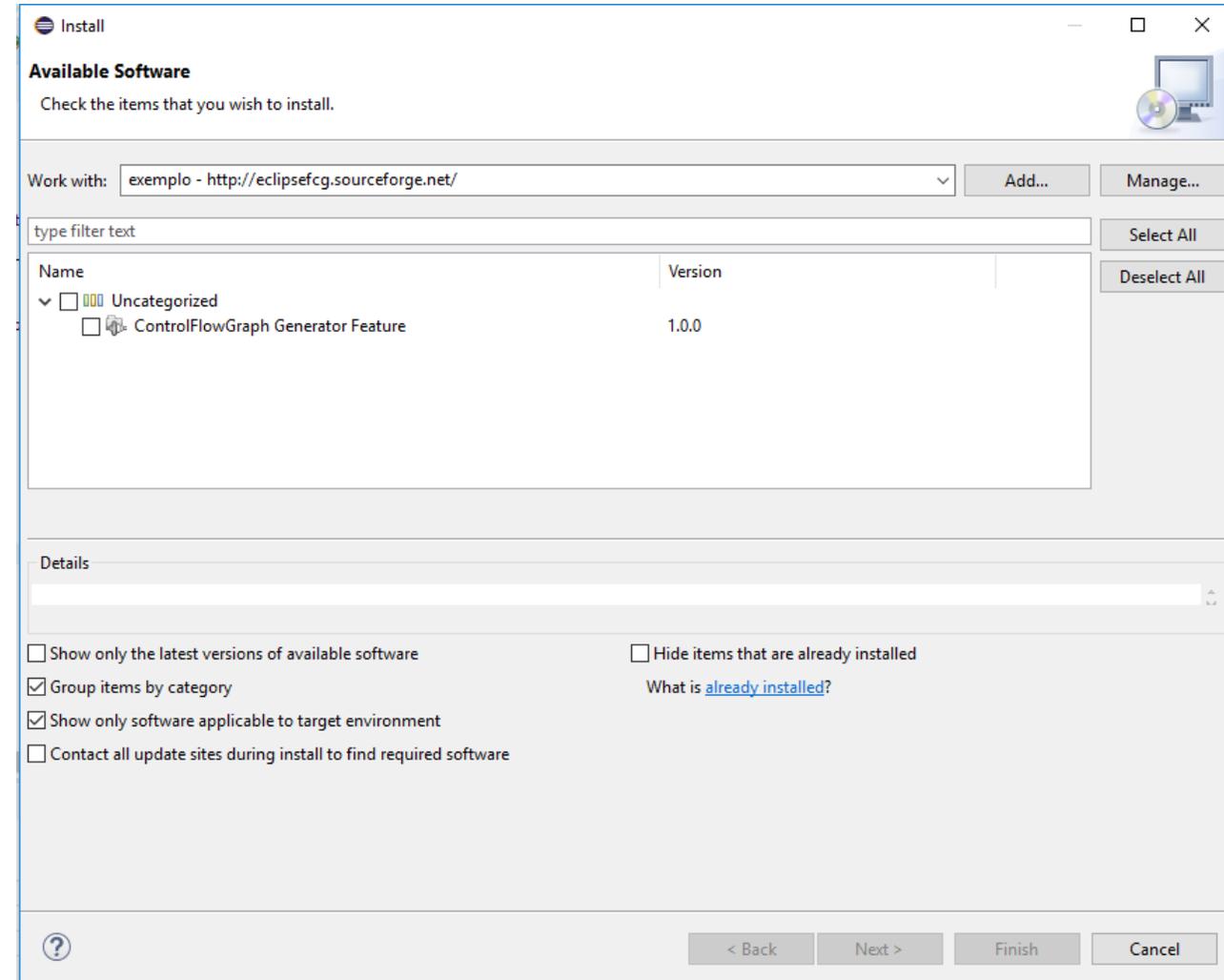


The screenshot shows a dialog box titled "Add Repository" with a close button (X) in the top right corner. The dialog contains two input fields: "Name" with the text "exemplo" and "Location" with the text "http://eclipsefcg.sourceforge.net/". To the right of the "Name" field is a "Local..." button, and to the right of the "Location" field is an "Archive..." button. At the bottom left is a help icon (question mark in a circle). At the bottom right are "Add" and "Cancel" buttons. The "Add" button is highlighted with a blue border.

- Clique em Adicionar

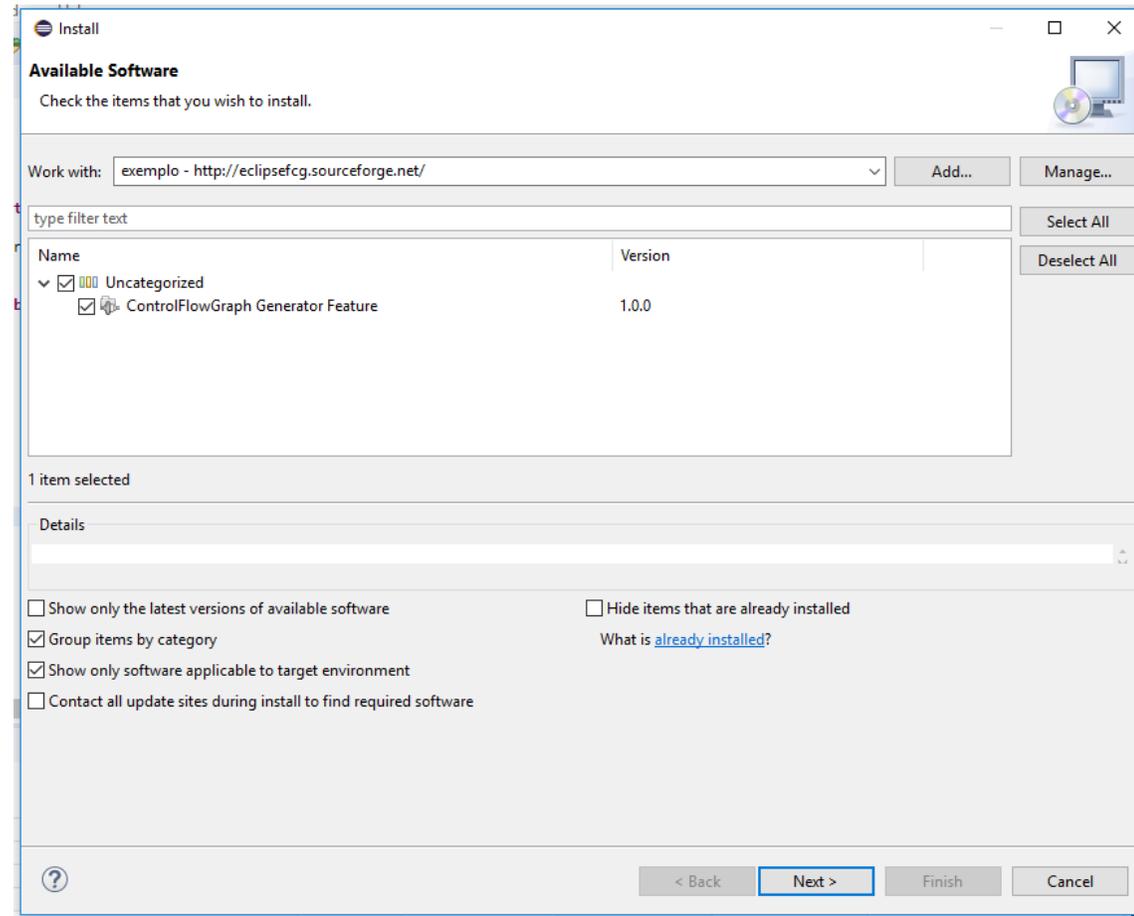
GFC

- Observe que o Eclipse carregou uma ferramenta
- É necessário expandir a categoria
- Posteriormente, selecione:
ControlFlowGraph Generator Feature



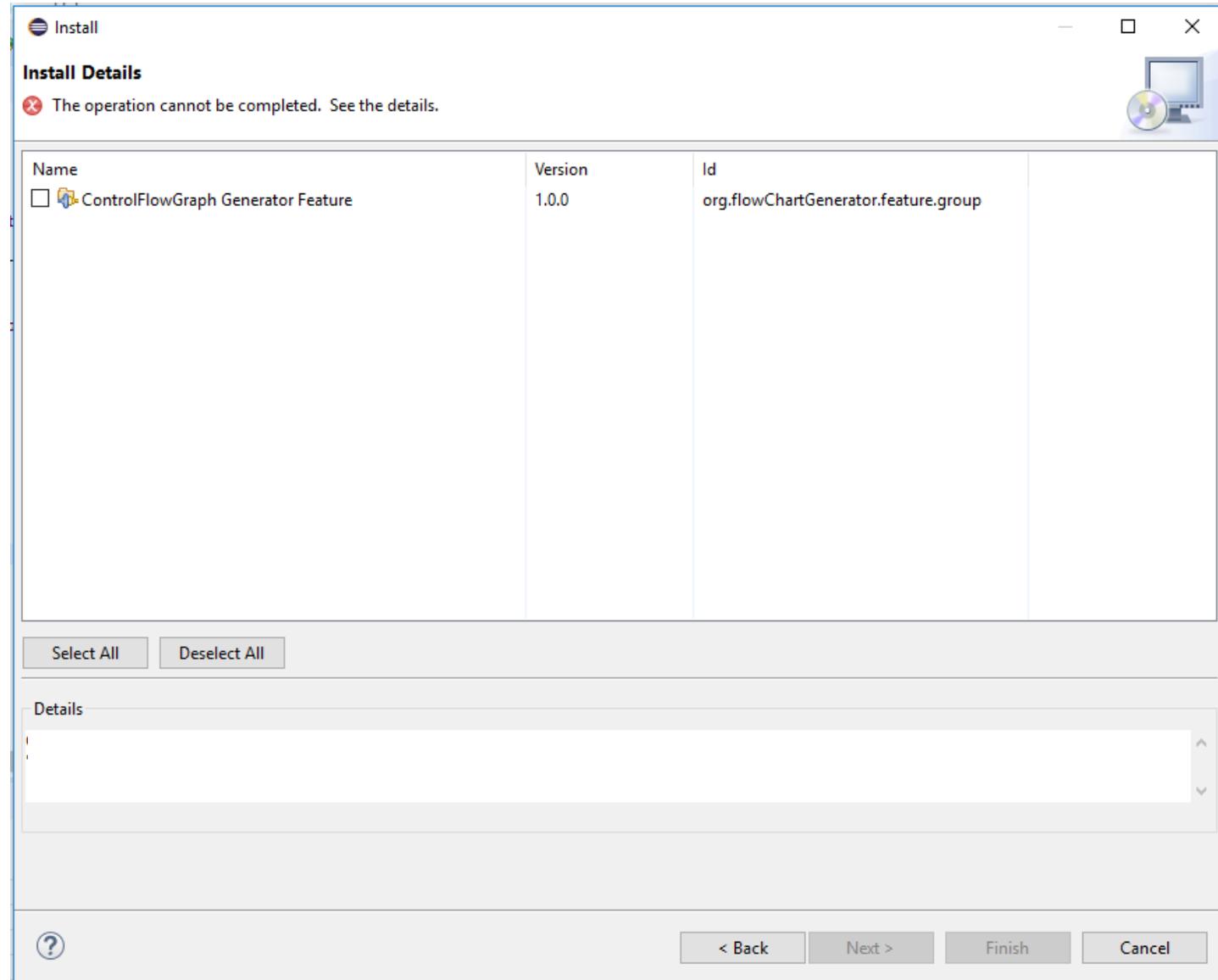
GFC

- Clique em Próximo



GFC

- Selecione novamente o nome da ferramenta e clique em Finalizar e aguarde a instalação



Pronto!

As ferramentas necessárias para a automatização da atividade de teste estão prontas para serem utilizadas!

Exemplo:

Programa: Dia da Semana

- Especificação

Escreva um programa em que o usuário deve informar um número de 1 a 7. A partir disso, o programa deve exibir o nome do dia da semana correspondente. Caso o usuário informar um número fora desse intervalo, o programa deve apresentar uma mensagem de erro.

Programa: Dia da Semana

O programa possui duas classes:

Main.java

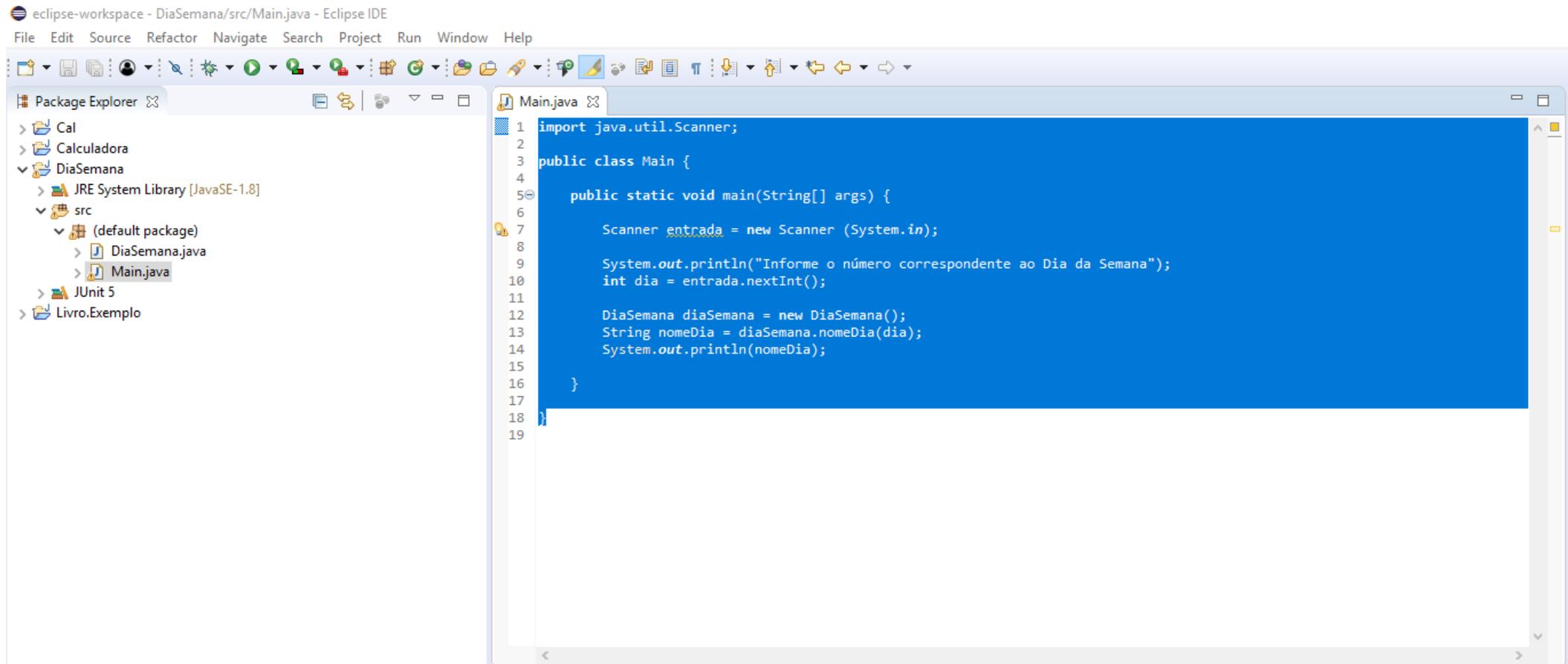
DiaSemana.java

O nosso objetivo é implementar casos de teste para testar o método:

```
public String nomeDia(int dia)
```

Programa: Dia da Semana

- Classe: Main

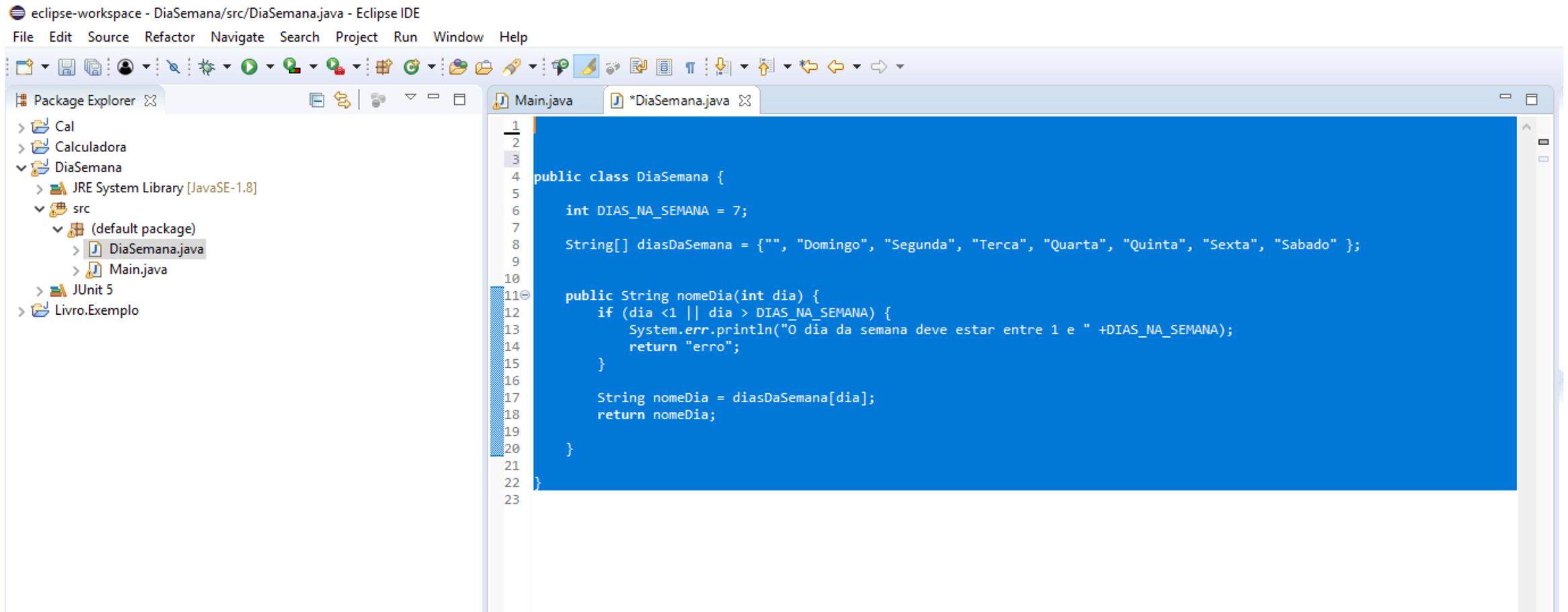


The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays the project structure: Cal, Calculadora, DiaSemana (expanded to show JRE System Library [JavaSE-1.8], src, (default package) containing DiaSemana.java and Main.java, JUnit 5, and Livro.Exemplo). The Main.java editor on the right contains the following code:

```
1 import java.util.Scanner;
2
3 public class Main {
4
5     public static void main(String[] args) {
6
7         Scanner entrada = new Scanner (System.in);
8
9         System.out.println("Informe o número correspondente ao Dia da Semana");
10        int dia = entrada.nextInt();
11
12        DiaSemana diaSemana = new DiaSemana();
13        String nomeDia = diaSemana.nomeDia(dia);
14        System.out.println(nomeDia);
15
16    }
17
18 }
19
```

Programa: Dia da Semana

- Classe: DiaSemana



The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays the project structure: DiaSemana (src) containing DiaSemana.java and Main.java. The main editor window shows the code for DiaSemana.java, which defines a class with a constant for the number of days in a week, an array of day names, and a method to return the name of a day given its index.

```
1
2
3
4 public class DiaSemana {
5
6     int DIAS_NA_SEMANA = 7;
7
8     String[] diasDaSemana = {"", "Domingo", "Segunda", "Terca", "Quarta", "Quinta", "Sexta", "Sabado" };
9
10
11 public String nomeDia(int dia) {
12     if (dia < 1 || dia > DIAS_NA_SEMANA) {
13         System.err.println("O dia da semana deve estar entre 1 e " + DIAS_NA_SEMANA);
14         return "erro";
15     }
16
17     String nomeDia = diasDaSemana[dia];
18     return nomeDia;
19 }
20
21
22
23 }
```

Programa: Dia da Semana

- Primeira coisa a se fazer é construir o GFC

Para tanto:

Selecione a classe que contém o método que irá testar, clique com o botão esquerdo

Programa: Dia da Semana

The screenshot displays the Eclipse IDE interface. The Package Explorer on the left shows a project named 'DiaSemana' with a source folder 'src' containing files 'DiaSemana.java', 'DiaSemanaTest.java', and 'Main.java'. The main editor window shows the code for 'DiaSemana.java':

```
1
2
3
4 public class DiaSemana {
5
6     int DIAS_NA_SEMANA = 7;
7
8     String[] diasDaSemana = {"", "Domingo", "Segunda", "Terca", "Quarta", "Quinta", "Sexta", "Sabado" };
9
10
11 public String nomeDia(int dia) {
12     if (dia < 1 || dia > DIAS_NA_SEMANA) {
13         System.err.println("O dia da semana deve estar entre 1 e " + DIAS_NA_SEMANA);
14         return "erro";
15     }
16
17     String nomeDia = diasDaSemana[dia];
18     return nomeDia;
19
20 }
21
22 }
23
```

The Outline view on the right shows the class structure:

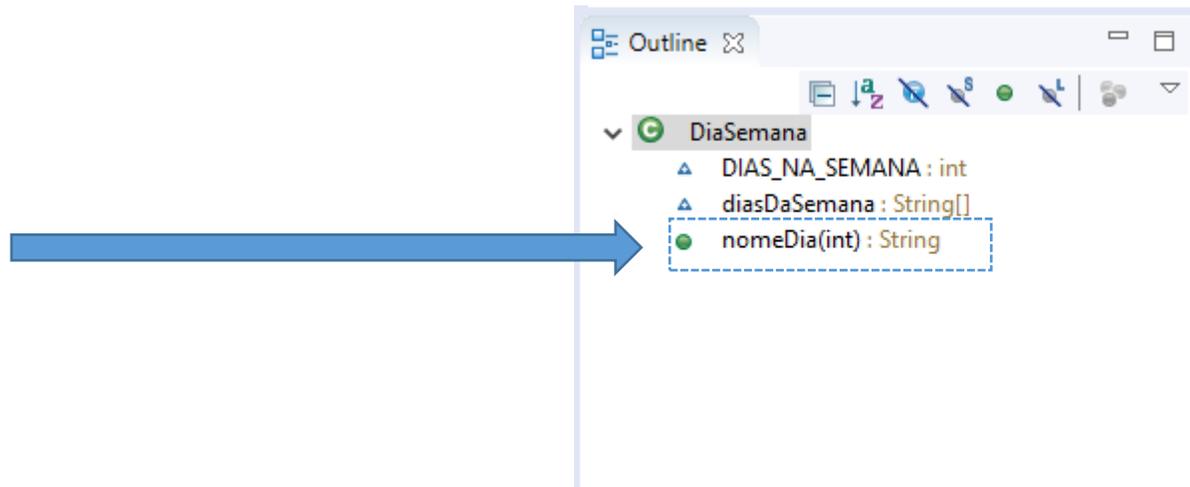
- ▼ DiaSemana
 - ▲ DIAS_NA_SEMANA : int
 - ▲ diasDaSemana : String[]
 - nomeDia(int) : String

The Coverage view at the bottom shows a table with the following columns: Element, Coverage, Covered Instructio, Missed Instructions, and Total Instructions. The table is currently empty.

Writable Smart Insert 11:36

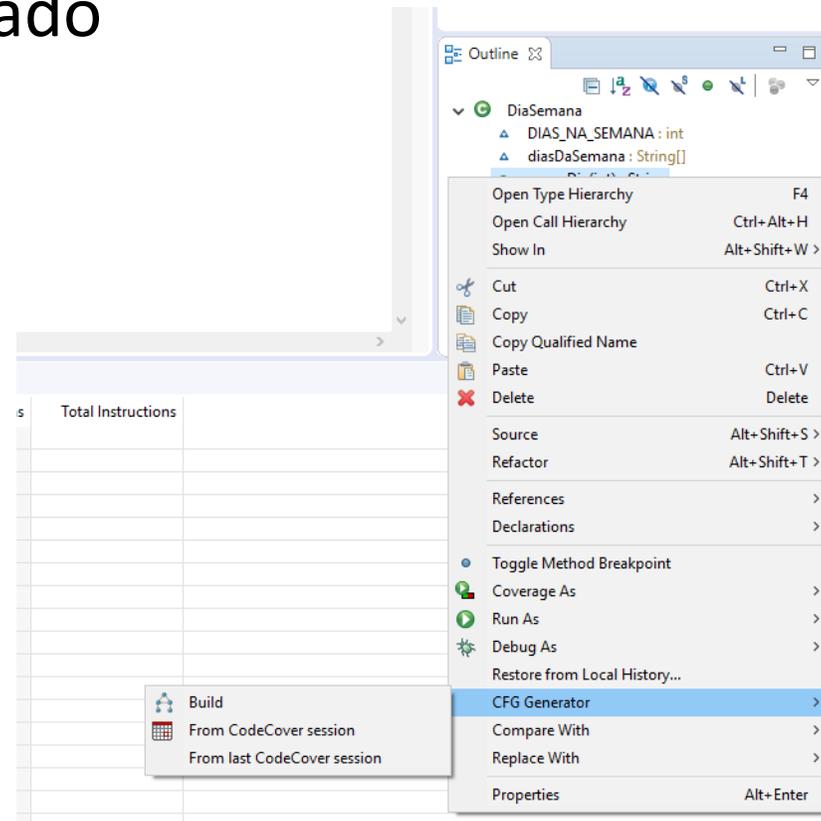
Programa: Dia da Semana

- Encontre a janela “Outline”, observe que ela contém o método que vamos testar. Selecione o método.



Programa: Dia da Semana

- Clique com o botão direito em cima, escolha no menu a opção “CFG Generator e, posteriormente, clique em Build. Em uma nova aba, o grafo será carregado



Programa: Dia da Semana

- GFC

eclipse-workspace - C:\Users\usuario\AppData\Local\Temp\DiaSemana_nomeDia.ff3 - Eclipse IDE

File Edit Navigate Search Project Run Window Help

Package Explorer

- > Cal
- > Calculadora
- > DiaSemana
 - > JRE System Library [JavaSE-1.8]
 - > src
 - > (default package)
 - > DiaSemana.java
 - > DiaSemanaTest.java
 - > Main.java
 - > JUnit 5
 - > cfg
 - > Livro.Exemplo

Main.java *DiaSemana.java DiaSemanaTest.java Flow Chart Viewer

```
graph TD; START([START]) --> Decision{dia < 1 || dia > DIAS_NA_SEMANA}; Decision --> Expression[Expression]; Decision --> nomeDia[nomeDia]; Expression --> END([END]); nomeDia --> END;
```

Task List

Find

Outline

Flow Chart Generator

MacCabe results

5 - 5 + 2 = 2

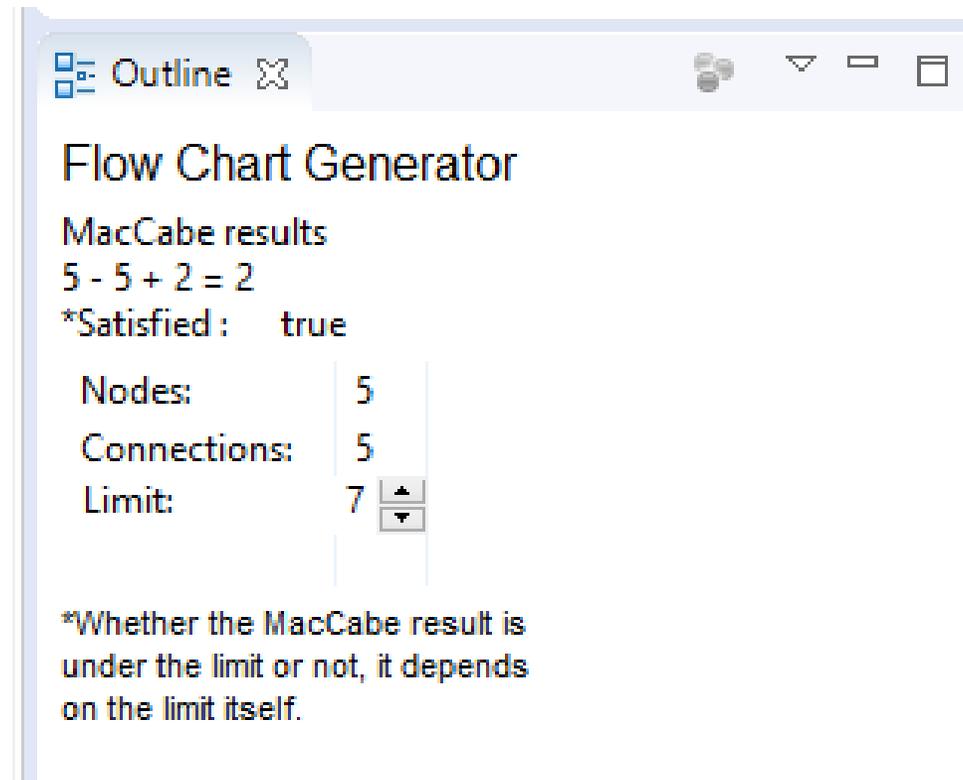
*Satisfied: true

Nodes:	5
Connections:	5
Limit:	7

*Whether the MacCabe result is under the limit or not, it depends on the limit itself.

Programa: Dia da Semana

- Note que a ferramenta calculou a complexidade ciclomática do método:



The screenshot shows a window titled "Outline" with a "Flow Chart Generator" section. It displays the following text:

```
MacCabe results  
5 - 5 + 2 = 2  
*Satisfied: true
```

Nodes:	5
Connections:	5
Limit:	7

*Whether the MacCabe result is under the limit or not, it depends on the limit itself.

Programa: Dia da Semana

- Após criar o grafo, aplique os critérios desejados, projete os casos de teste e os implemente na ferramenta.
- É necessário criar uma classe para os testes.
- A seguir será apresentada uma codificação de casos de teste para o método.

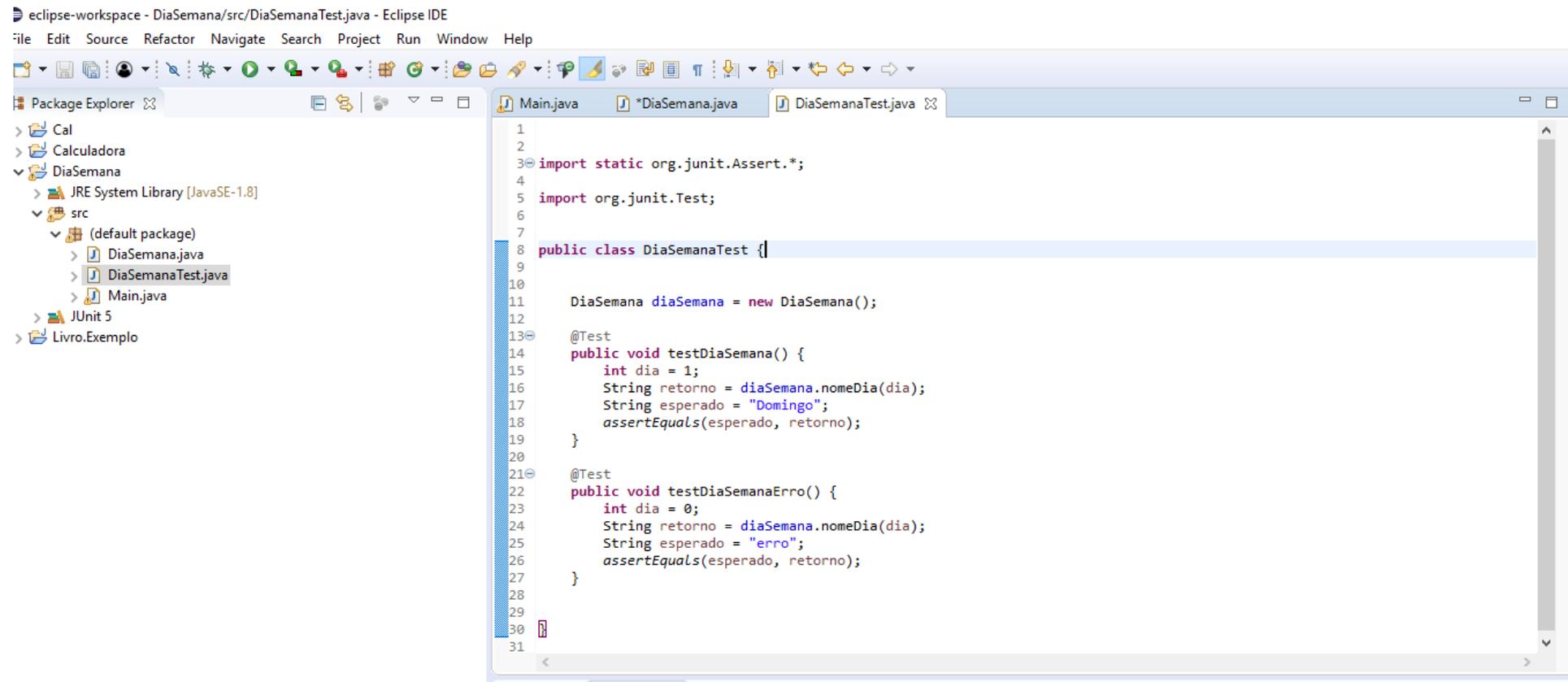


É necessário ter conhecimento sobre o JUnit para entender o funcionamento dos testes. Se você não possui, recomendamos o seguinte material:

<http://dextra.com.br/pt/blog/tutorial-ou-melhor-um-passo-a-passo-explicando-o-funcionamento-do-junit-2/>

Programa: Dia da Semana

- Implementando casos de teste



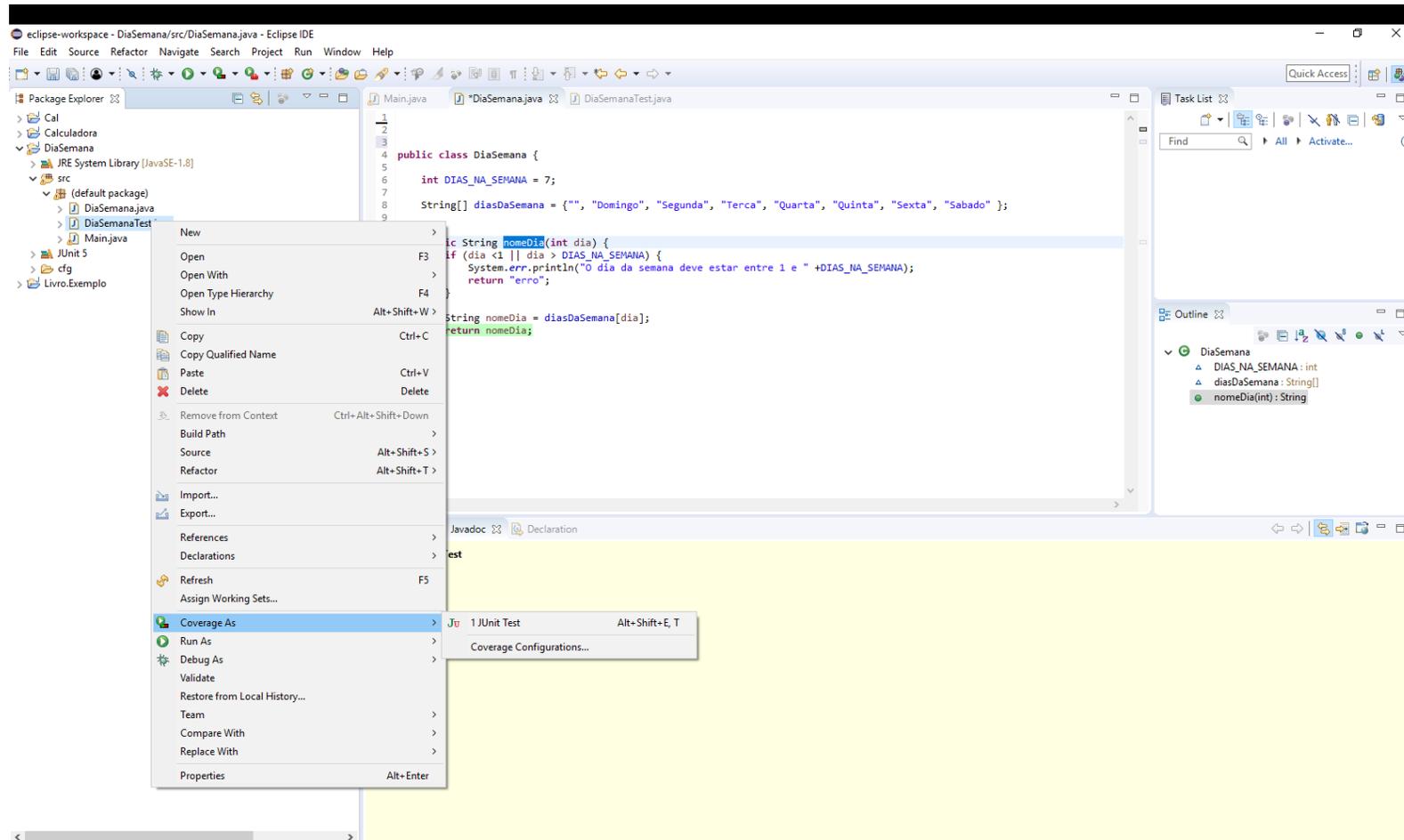
The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays the project structure: Cal, Calculadora, DiaSemana (containing JRE System Library [JavaSE-1.8], src, and sub-packages), JUnit 5, and Livro.Exemplo. The main editor window shows the code for DiaSemanaTest.java. The code includes imports for JUnit Assert and Test, and defines a public class DiaSemanaTest with two test methods: testDiaSemana() and testDiaSemanaErro().

```
1
2
3 import static org.junit.Assert.*;
4
5 import org.junit.Test;
6
7
8 public class DiaSemanaTest {
9
10
11     DiaSemana diaSemana = new DiaSemana();
12
13 @Test
14 public void testDiaSemana() {
15     int dia = 1;
16     String retorno = diaSemana.nomeDia(dia);
17     String esperado = "Domingo";
18     assertEquals(esperado, retorno);
19 }
20
21 @Test
22 public void testDiaSemanaErro() {
23     int dia = 0;
24     String retorno = diaSemana.nomeDia(dia);
25     String esperado = "erro";
26     assertEquals(esperado, retorno);
27 }
28
29
30
31
```

Programa: Dia da Semana

- Com a classe criada, vá até a classe DiaSemanaTest.java e clique com o botão direito. Perceba que agora você possui a opção **Coverage as...** . Selecione a opção **JUnit Test** e espere o resultado.

Programa: Dia da Semana



Programa: Dia da Semana

O código fica colorido!

Fique calmo que está tudo certo. Cada cor significa uma coisa!

Verde: Código Executado

Amarelo: Ponto de decisão

Vermelho: Código não executado

Programa: Dia da Semana

- Preste atenção na aba **Coverage**:

The screenshot shows the Eclipse IDE with the following components:

- Package Explorer:** Shows the project structure with 'DiaSemanaTest' selected.
- JUnit Runner:** Shows 'Finished after 0,141 seconds', 'Runs: 2/2', 'Errors: 0', and 'Failures: 0'.
- Code Editor:** Displays the source code for 'DiaSemana.java'. The code includes a class with a constant for the number of days in a week, an array of day names, and a method to return the name of a day given an index. A test case is visible in the background.
- Coverage View:** A table at the bottom showing the coverage of the code. The 'DiaSemana' class and its test file 'DiaSemanaTest.java' both show 100.0% coverage.

Element	Coverage	Covered Instructio	Missed Instructions	Total Instructions
▼ DiaSemana	79,7 %	102	26	128
▼ src	79,7 %	102	26	128
▼ (default package)	79,7 %	102	26	128
▼ Main.java	0,0 %	0	26	26
> Main	0,0 %	0	26	26
▼ DiaSemana.java	100,0 %	68	0	68
> DiaSemana	100,0 %	68	0	68
> DiaSemanaTest.java	100,0 %	34	0	34

Note que a EclEmma/JaCoCo demonstra que você projetou casos de teste que cobrem 100% do código da classe DiaSemana.java