

# Preparação do ambiente

Eclipse & JUnit



## Exercício Testes Unitários ▼

Arquivos anexados:

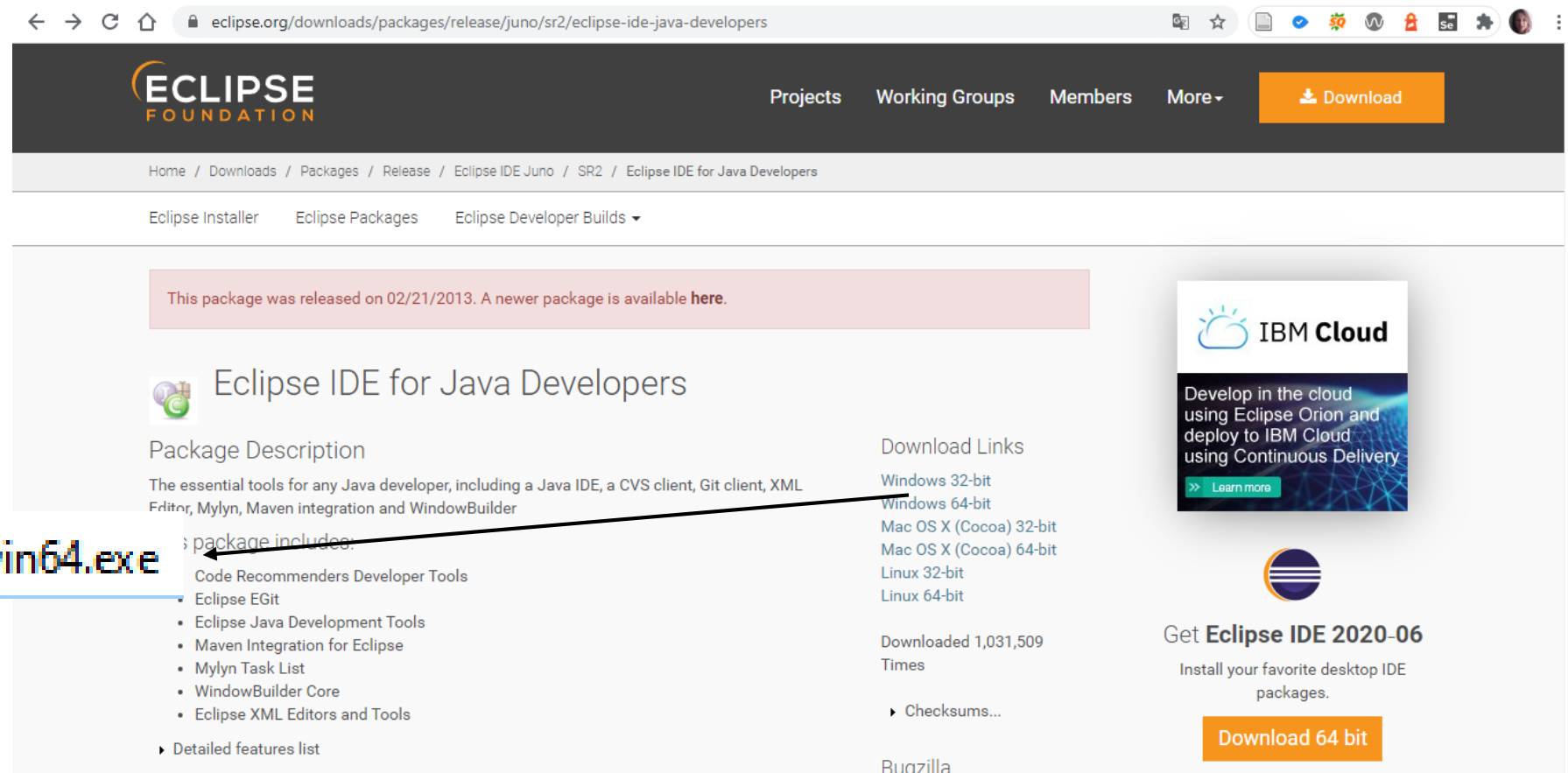
-  SistemaBancario.zip ▼ (15,169 KB)
-  hamcrest-core-1.3.jar ▼ (43,969 KB)
-  junit-4.12.jar ▼ (307,551 KB)
-  Instruções Teste de Unidade com JUnit.pdf ▼ (46,167 KB)
-  Tutorial.pdf ▼ (899,096 KB)

Para a elaboração deste exercício leia o documento em anexo **Instruções Teste de Unidade com JUnit.pdf**.

# Teste Unitário & Eclipse

## 1. Instalar Eclipse IDE for Java Developers

<https://www.eclipse.org/downloads/packages/release/juno/sr2/eclipse-ide-java-developers>



The screenshot shows the Eclipse Foundation website's download page for Eclipse IDE for Java Developers. The page includes a navigation bar with links to Projects, Working Groups, Members, and a Download button. A breadcrumb trail indicates the path: Home / Downloads / Packages / Release / Eclipse IDE Juno / SR2 / Eclipse IDE for Java Developers. Below this, there are links for Eclipse Installer, Eclipse Packages, and Eclipse Developer Builds. A red banner states: "This package was released on 02/21/2013. A newer package is available [here](#)." The main content area is divided into two columns. The left column, titled "Eclipse IDE for Java Developers", contains a "Package Description" section that lists the tools included: Code Recommenders Developer Tools, Eclipse EGit, Eclipse Java Development Tools, Maven Integration for Eclipse, Mylyn Task List, WindowBuilder Core, and Eclipse XML Editors and Tools. A link to "Detailed features list" is also provided. The right column, titled "Download Links", lists various operating system and architecture combinations: Windows 32-bit, Windows 64-bit, Mac OS X (Cocoa) 32-bit, Mac OS X (Cocoa) 64-bit, Linux 32-bit, and Linux 64-bit. It also shows that the package has been downloaded 1,031,509 times and provides a link to "Checksums...". At the bottom right, there is a section for "Get Eclipse IDE 2020-06" with a "Download 64 bit" button. An IBM Cloud advertisement is also visible on the right side of the page.

**ECLIPSE FOUNDATION** Projects Working Groups Members More [Download](#)

Home / Downloads / Packages / Release / Eclipse IDE Juno / SR2 / Eclipse IDE for Java Developers

[Eclipse Installer](#) [Eclipse Packages](#) [Eclipse Developer Builds](#)

This package was released on 02/21/2013. A newer package is available [here](#).

### Eclipse IDE for Java Developers

**Package Description**

The essential tools for any Java developer, including a Java IDE, a CVS client, Git client, XML Editor, Mylyn, Maven integration and WindowBuilder

**package includes:**

- Code Recommenders Developer Tools
- Eclipse EGit
- Eclipse Java Development Tools
- Maven Integration for Eclipse
- Mylyn Task List
- WindowBuilder Core
- Eclipse XML Editors and Tools

[Detailed features list](#)

**Download Links**

- Windows 32-bit
- Windows 64-bit
- Mac OS X (Cocoa) 32-bit
- Mac OS X (Cocoa) 64-bit
- Linux 32-bit
- Linux 64-bit

Downloaded 1,031,509 Times

[Checksums...](#)

Buqzilla

**IBM Cloud**

Develop in the cloud using Eclipse Orion and deploy to IBM Cloud using Continuous Delivery

[Learn more](#)

**Get Eclipse IDE 2020-06**

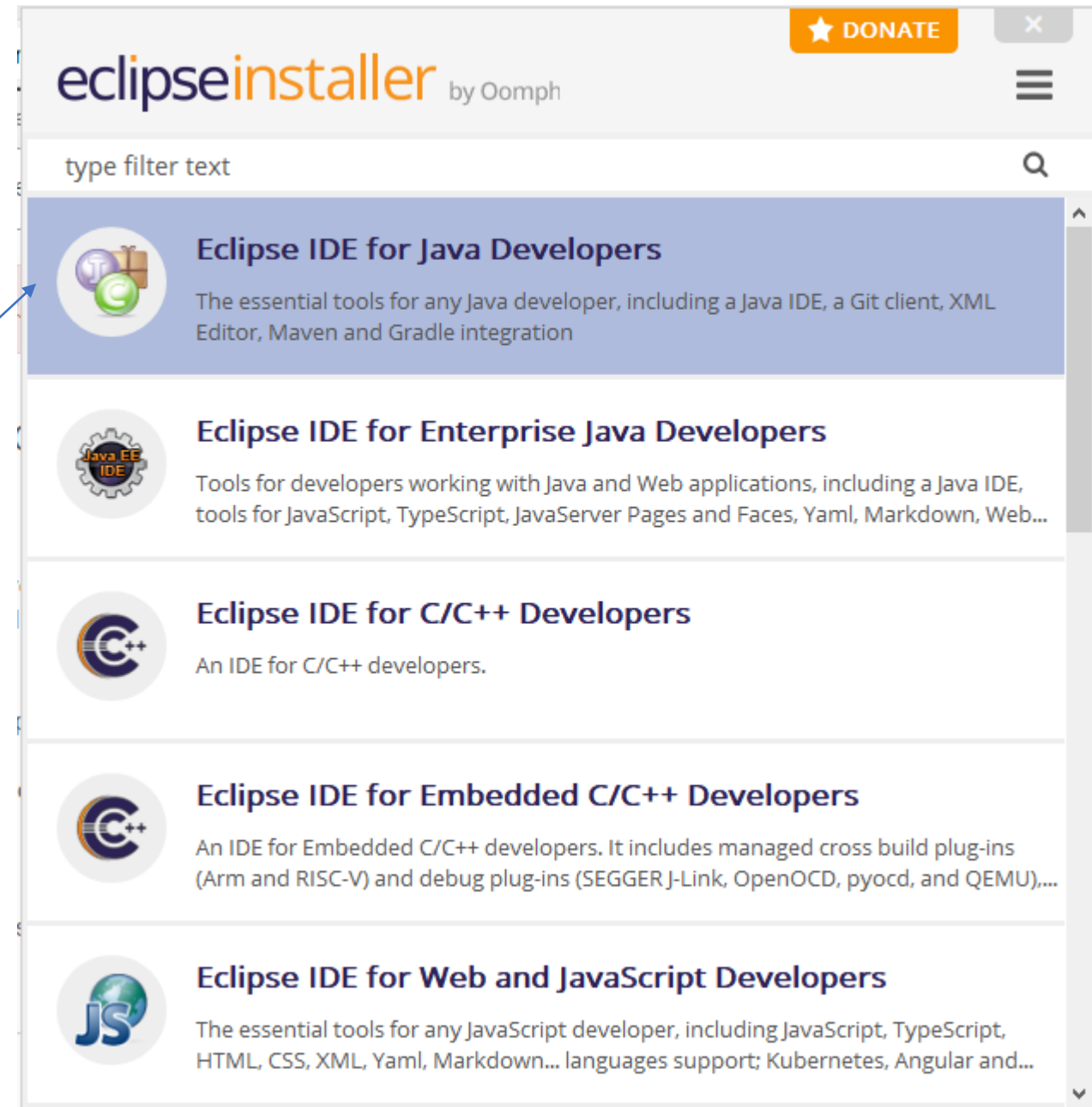
Install your favorite desktop IDE packages.

[Download 64 bit](#)



eclipse-inst-jre-win64.exe

- Eclipse IDE for Java Developers





## Eclipse IDE for Java Developers

[details](#)

The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Maven and Gradle integration.

Java 11+ VM

://download.eclipse.org/justj/jres/15/updates/release/15.0.2



Installation Folder

C:\Users\claud\eclipse\java-2020-12



create start menu entry



create desktop shortcut



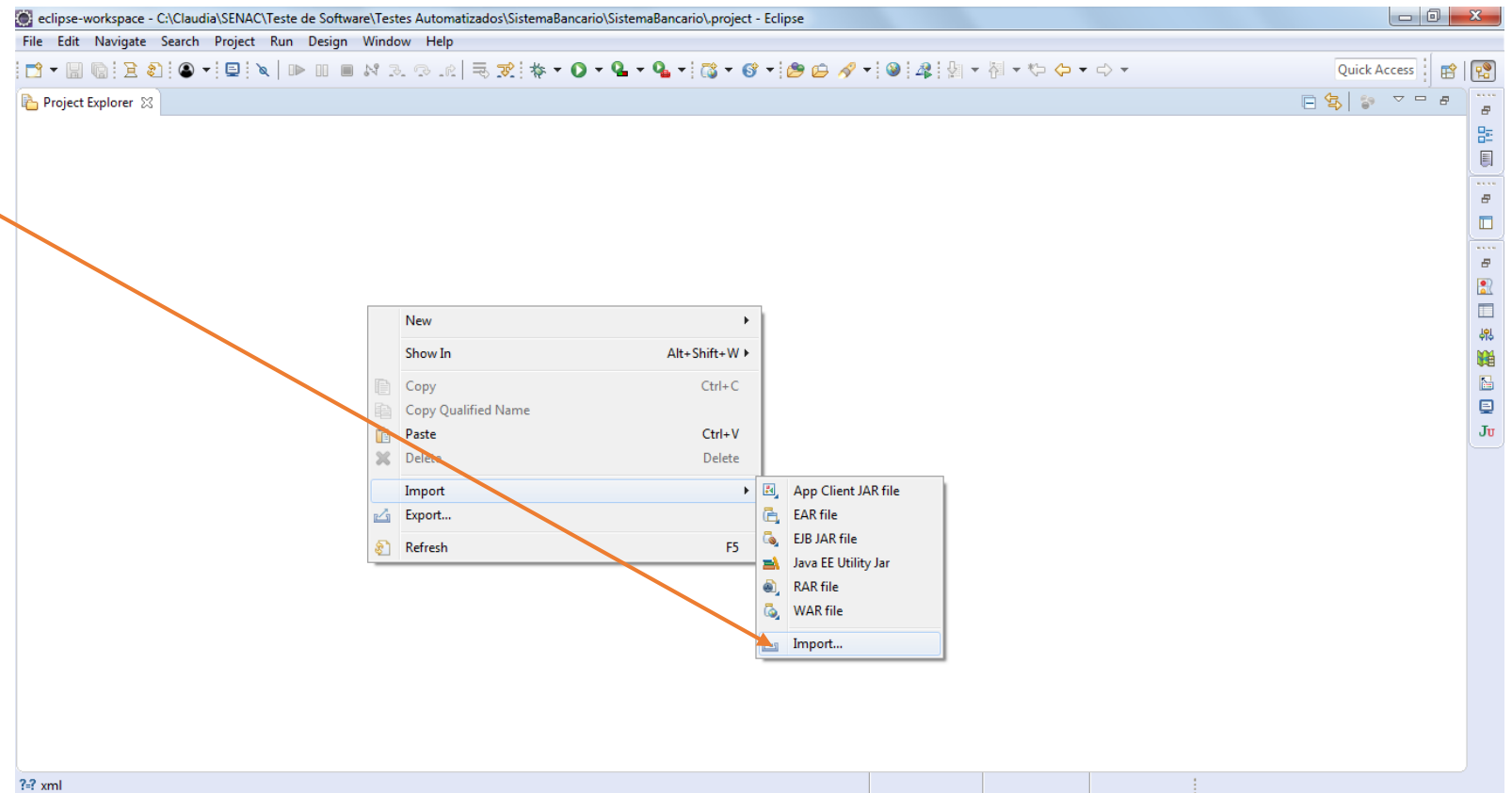
↓ INSTALL

← BACK

# Teste Unitário

- Depois de instalado, acesse o Eclipse e **importe o projeto Sistema Bancário\***, disponibilizado no exercício.

**File\Import**



\*O arquivo está compactado na pasta SistemaBancario.zip, precisa descompactá-lo para utilização do projeto e das bibliotecas.

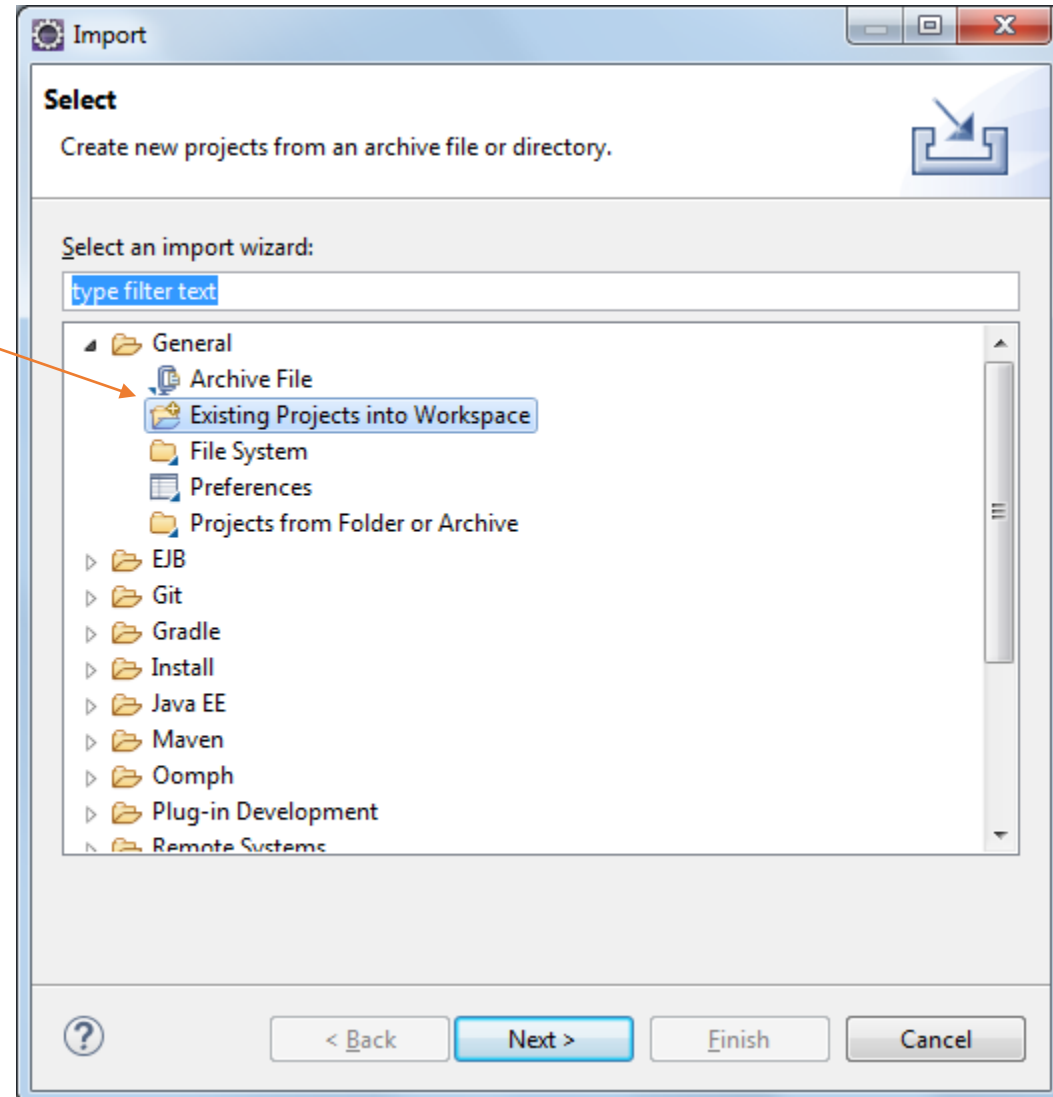
# Teste Unitário

2. Importar o projeto Sistema Bancário

Selecione a opção:

**General**

**Existing Projects into Workspace**

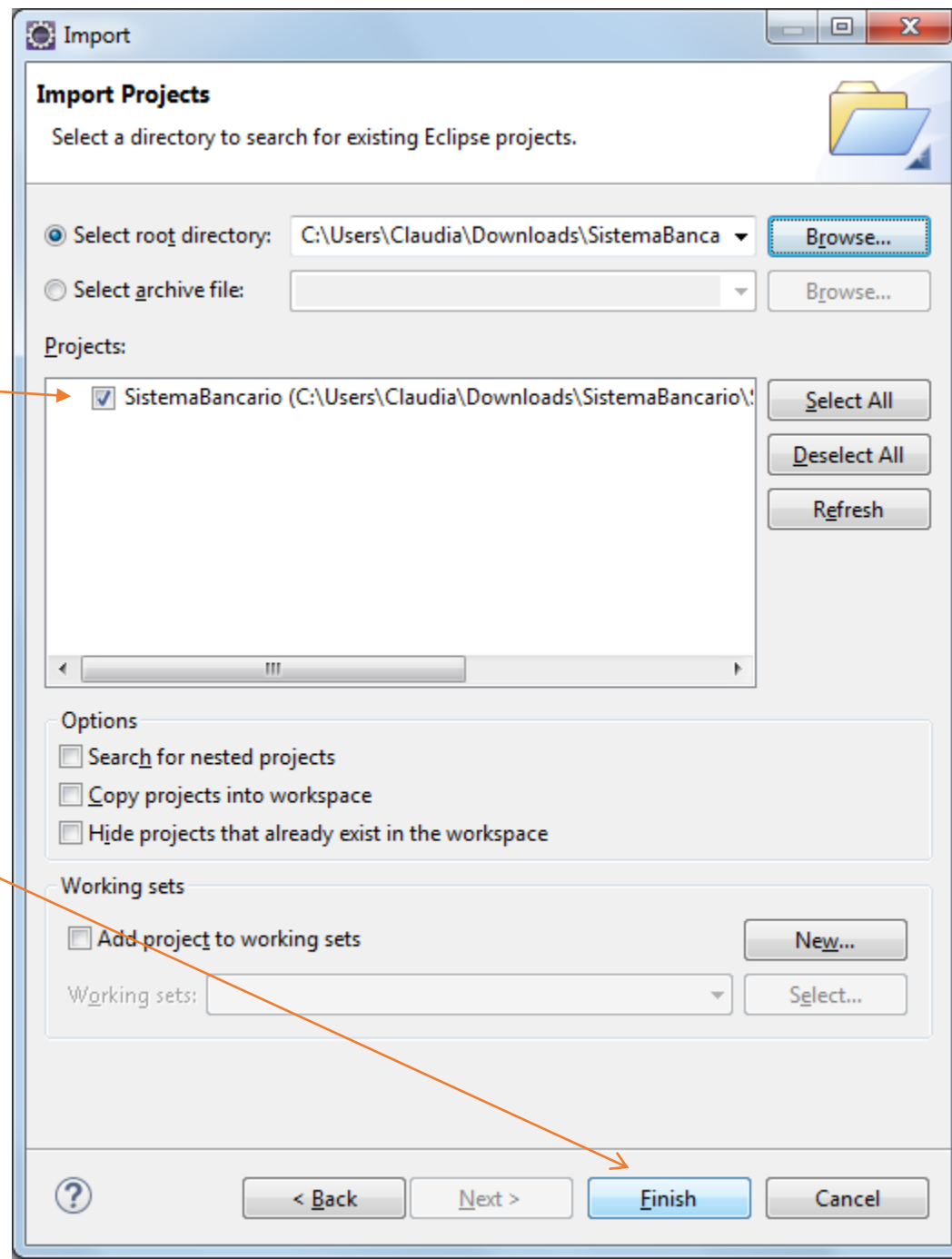


# Teste Unitário

## 2. Importar o projeto Sistema Bancário

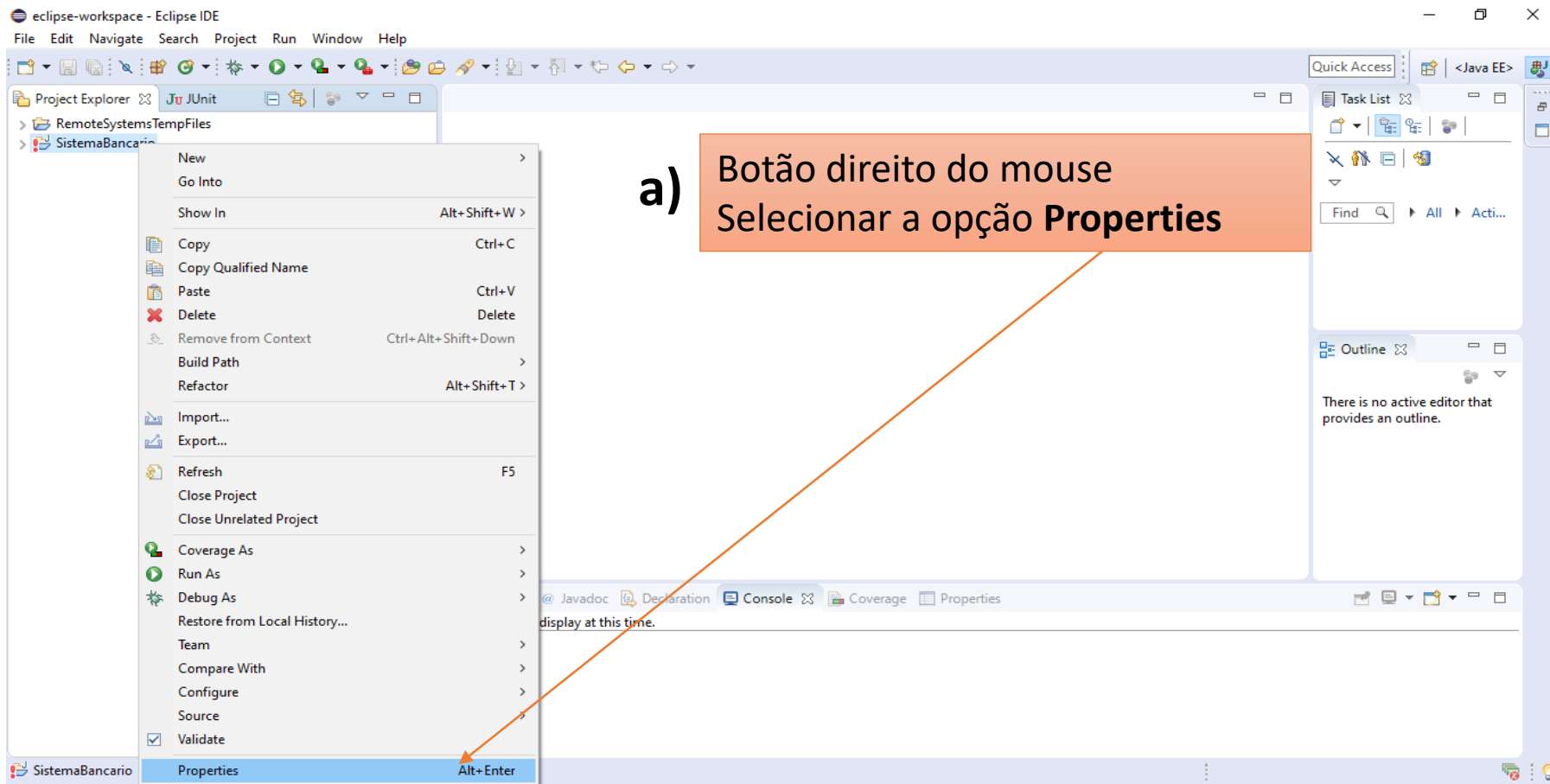
Após selecionar o arquivo finalizar a importação:

**Finish**



# Teste Unitário

3. Importar as bibliotecas **Junit** e **hamcrest**, seguindo os passos **a**, **b**, **c**, **d**:





eclipse-workspace - Eclipse IDE

File Edit Navigate Search Project Run Window Help

Project Explorer JUnit

- RemoteSystemsTempFiles
- SistemaBancario

Properties for SistemaBancario

2 build path entries are missing.

Source Projects Libraries Order and Export Module Dependencies

JARs and class folders on the build path:

- hamcrest-core-1.3.jar - D:\Eclipse + JUnit (missing)
- junit-4.12.jar - D:\Eclipse + JUnit (missing)
- JRE System Library [JavaSE-1.8]

Add JARs...  
Add External JARs...  
Add Variable...  
Add Library...  
Add Class Folder...  
Add External Class Folder...  
Edit...  
Remove  
Migrate JAR File...

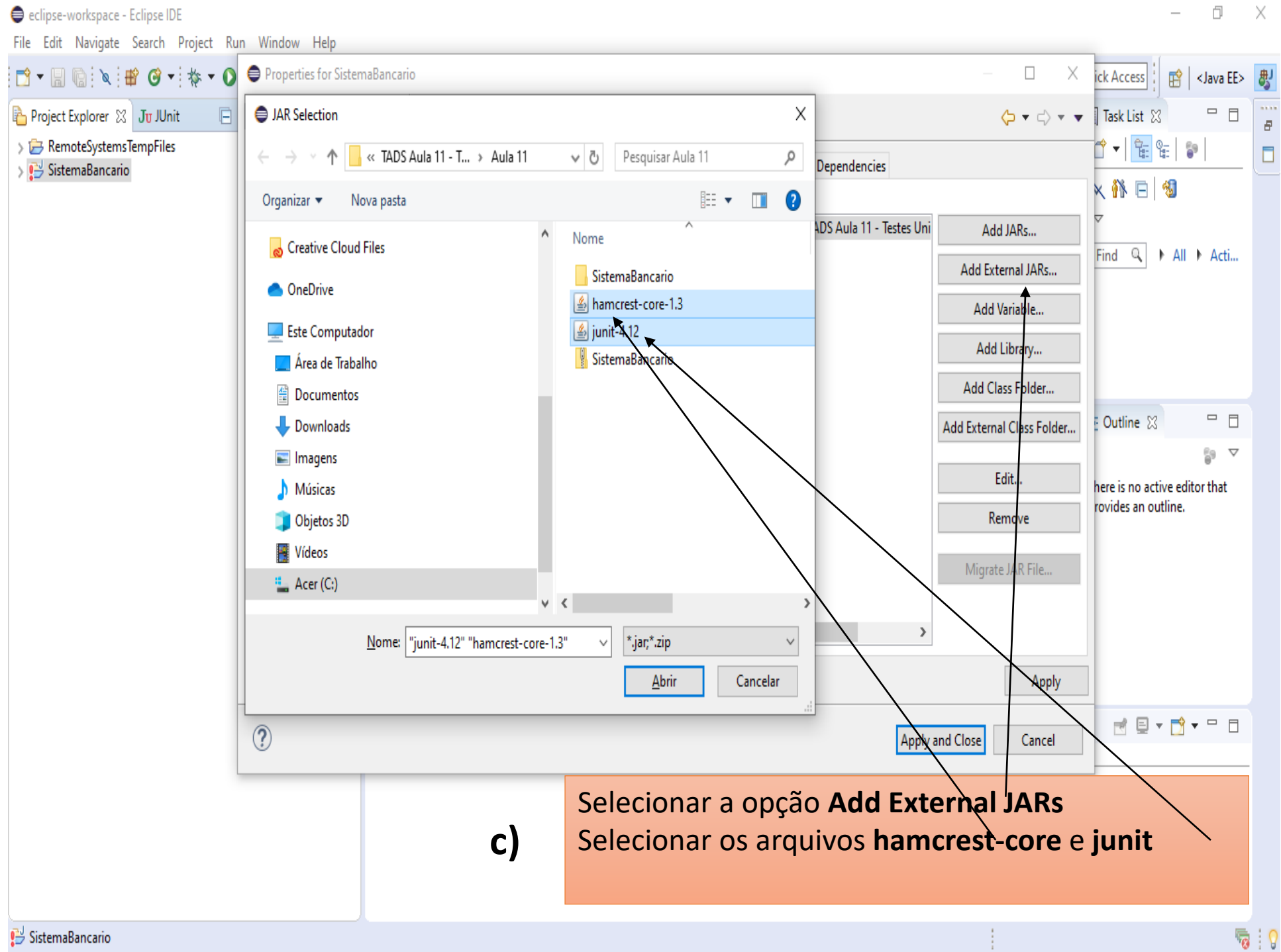
Apply

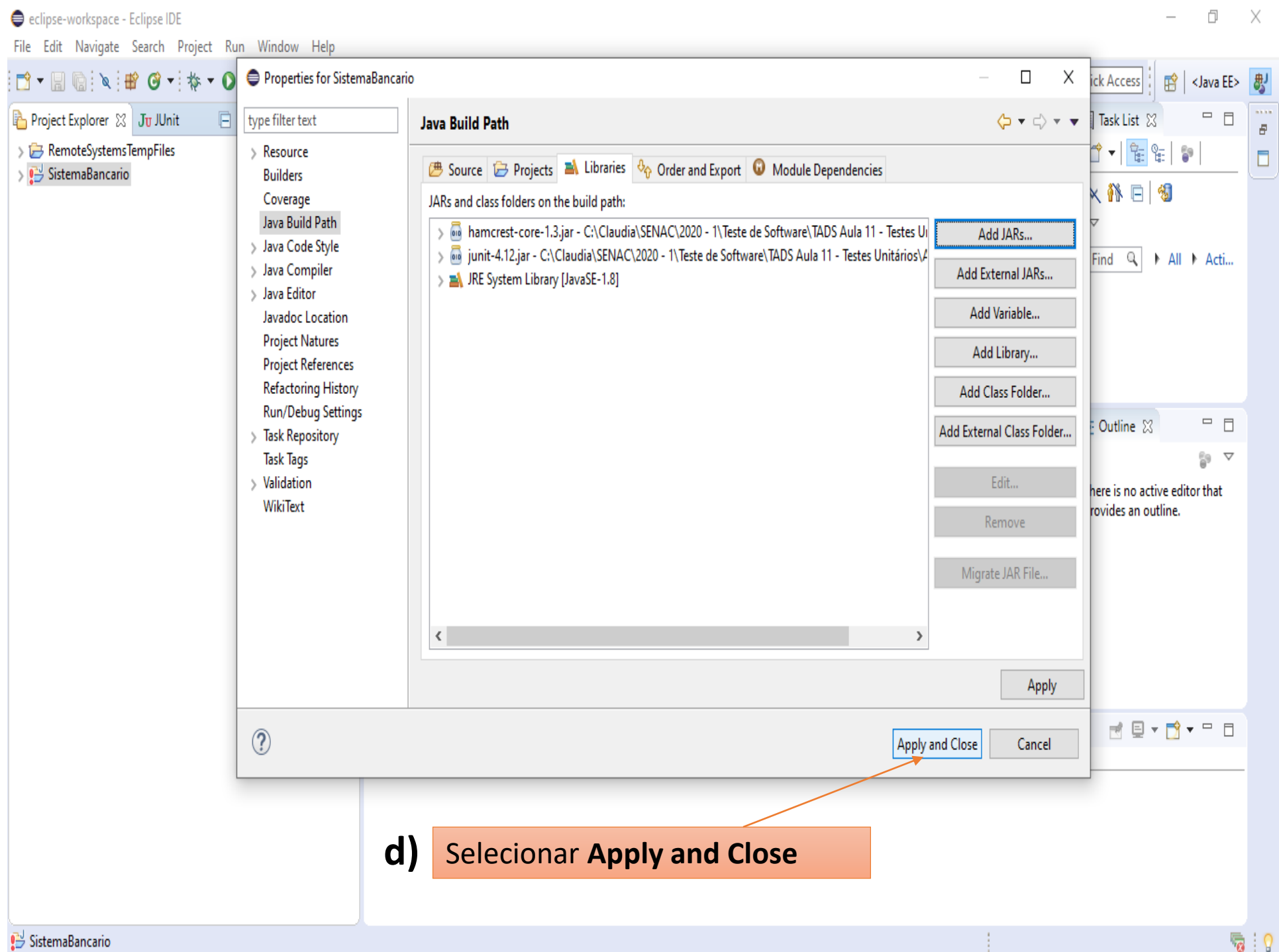
Apply and Close Cancel

Java Build Path

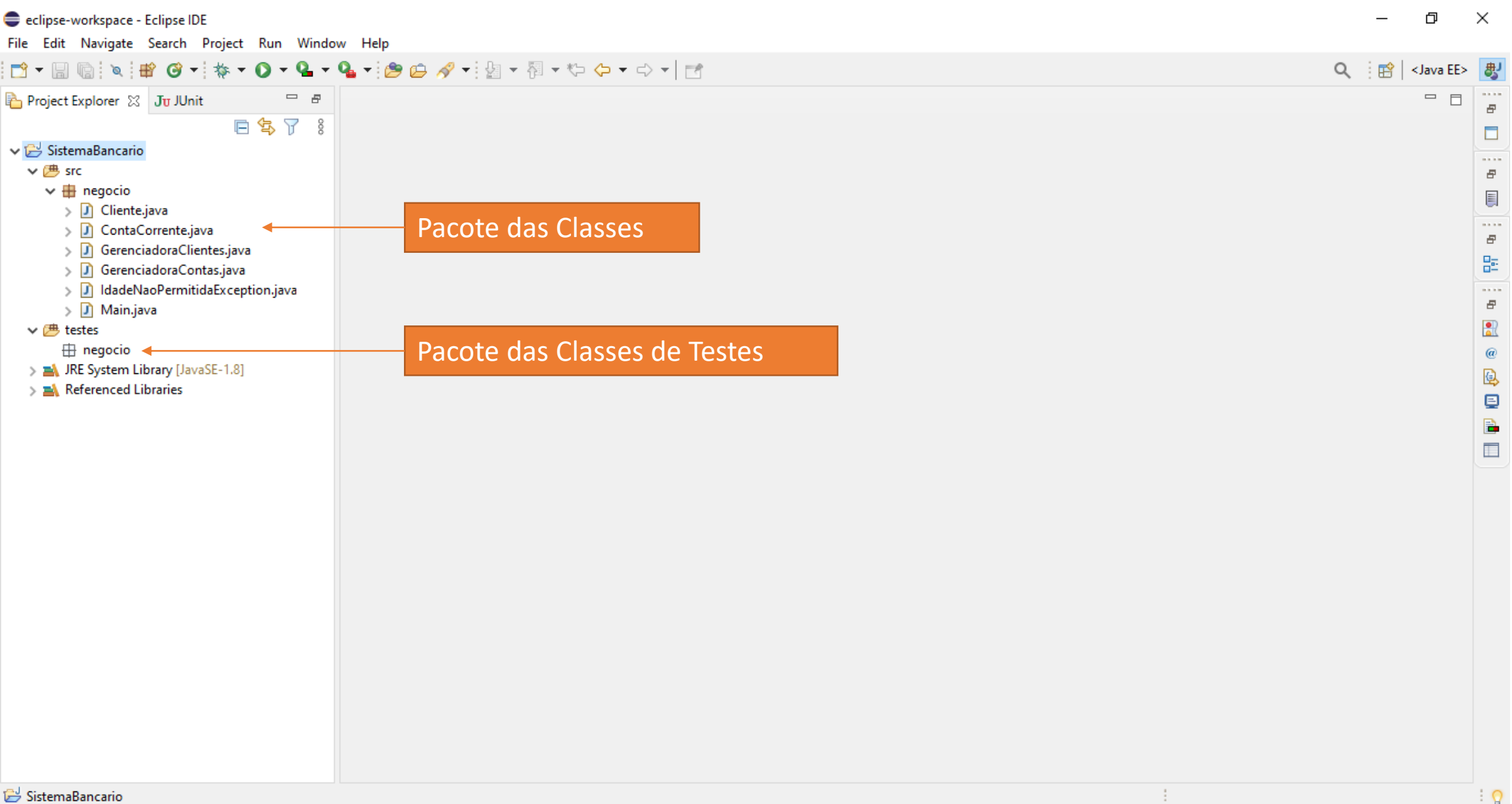
Libraries

b) Selecionar opção Java Build Path  
Pasta Libraries





## 4. Criar as classes de testes no Pacote das Classes de Testes



```
1 package negocio;
2
3 import static org.junit.Assert.*;
4 import static org.hamcrest.CoreMatchers.*;
5
6 import java.util.ArrayList;
7 import java.util.List;
8
9 import org.junit.After;
10 import org.junit.Before;
11 import org.junit.Test;
12
13
14
15 /**
16  * Classe de teste criada para garantir o funcionamento das principais operações
17  * sobre clientes, realizadas pela classe GerenciadoraClientes.
18  *
19  */
20 public class GerenciadoraClientesTest {
21
22     private GerenciadoraClientes gerenciarClientes;
23     private int idClientel = 1;
24     private int idCliente2 = 2;
25
26
27     @Before
28     public void setUp() {
```



Project Explorer JUnit

- ▼ SistemaBancario
  - ▼ src
    - ▼ negocio
      - Cliente.java
      - ContaCorrente.java
      - GerenciadoraClientes.java
      - GerenciadoraContas.java
      - IdadeNaoPermitidaException.java
      - Main.java
  - ▼ testes
    - ▼ negocio
      - GerenciadoraClientesTest.java
      - GerenciadoraContasTest.java
      - TodosOsTestes.java
  - JRE System Library [JavaSE-1.8]
  - Referenced Libraries

\*GerenciadoraClientesTest.java GerenciadoraContasTest.java TodosOsTestes.java

```
11 import org.junit.After;
12 import org.junit.Before;
13 import org.junit.Test;
14
15 /**
16  * ALUNO:
17  * DATA:
18  */
19
20 public class GerenciadoraClientesTest {
21
22     private GerenciadoraClientes gerClientes;
23     private int idCliente01 = 1;
24     private int idCliente02 = 2;
25
26     @Before
27     public void setUp() {
28
29         /* ===== Montagem do cenário ===== */
30
31         // criando alguns clientes
32         Cliente cliente01 = new Cliente(idCliente01, "João", 31, "joao@gmail.com", 1, true);
33         Cliente cliente02 = new Cliente(idCliente02, "Maria", 34, "maria@gmail.com", 1, true);
34
35         // inserindo os clientes criados na lista de clientes do banco
36         List<Cliente> clientesDoBanco = new ArrayList<>();
37         clientesDoBanco.add(cliente01);
38         clientesDoBanco.add(cliente02);
39
40         gerClientes = new GerenciadoraClientes(clientesDoBanco);
41
42     }
43
44     @After
45     public void tearDown() {
46         gerClientes.limpa();
47     }
48
49 }
```



Project Explorer JUnit

- ▼ SistemaBancario
  - ▼ src
    - ▼ negocio
      - > Cliente.java
      - > ContaCorrente.java
      - > GerenciadoraClientes.java
      - > GerenciadoraContas.java
      - > IdadeNaoPermitidaException.java
      - > Main.java
  - ▼ testes
    - ▼ negocio
      - > GerenciadoraClientesTest.java
      - > GerenciadoraContasTest.java
      - > TodosOsTestes.java
  - > JRE System Library [JavaSE-1.8]
  - > Referenced Libraries

\*GerenciadoraClientesTest.java GerenciadoraContasTest.java TodosOsTestes.java

```
46     gerClientes.limpa();
47
48 }
49
50 /**
51  * Teste básico da consulta de um cliente a partir do seu ID.
52  *
53  */
54
55 @Test
56 public void testPesquisaCliente() {
57
58     /* ===== Execução ===== */
59     Cliente cliente = gerClientes.pesquisaCliente(idCliente01);
60
61     /* ===== Verificações ===== */
62     assertThat(cliente.getId(), is(idCliente01));
63
64 }
65
66 /**
67  * Teste básico da remoção de um cliente a partir do seu ID.
68  *
69  */
70 @Test
71 public void testRemoveCliente() {
72
73     /* ===== Execução ===== */
74     boolean clienteRemovido = gerClientes.removeCliente(idCliente02);
75
76     /* ===== Verificações ===== */
77     assertThat(clienteRemovido, is(true));
78     assertThat(gerClientes.getClientesDoBanco().size(), is(1));
79     assertNull(gerClientes.pesquisaCliente(idCliente02));
80
81 }
82
83 }
84
```

```
1 package negocio;
2
3 import static org.junit.Assert.*;
4 import static org.hamcrest.CoreMatchers.*;
5
6 import java.util.ArrayList;
7 import java.util.List;
8
9 import org.junit.Test;
10
11
12
13 /**
14  * Classe de teste criada para garantir o funcionamento das principais operações
15  * sobre contas, realizadas pela classe GerenciadoraContas.
16  *
17  */
18 public class GerenciadoraContasTest {
19
20     private GerenciadoraContas gerenciadoraContas;
21
22
23     /**
24      * Validação de transferência entre contas.
25      * Cenário: Saldo suficiente em ambas as contas
26      */
27
28     @Test
29     public void testTransfereValor() {
30
```





Project Explorer JUnit

SistemaBancario

- src
  - negocio
    - Cliente.java
    - ContaCorrente.java
    - GerenciadoraClientes.java
    - GerenciadoraContas.java
    - IdadeNaoPermitidaException.java
    - Main.java
- testes
  - negocio
    - GerenciadoraClientesTest.java
    - GerenciadoraContasTest.java
    - TodosOsTestes.java
- JRE System Library [JavaSE-1.8]
- Referenced Libraries

Gerenciador... Gerenciador... TodosOsTest... Gerenciador... Cliente.java ContaCorren... Main.java

```
12 /**
13  * ALUNO:
14  * DATA:
15  */
16 public class GerenciadoraContasTest {
17
18     private GerenciadoraContas gerContas;
19
20     /**
21      * Teste básico da transferência de um valor da conta de um cliente para outro,
22      * estando ambos os clientes ativos e havendo saldo suficiente para tal transferência
23      * ocorrer com sucesso.
24      */
25     @Test
26     public void testTransfereValor() {
27
28         /* ===== Montagem do cenário ===== */
29
30         // criando alguns clientes
31         int idConta01 = 1;
32         int idConta02 = 2;
33         ContaCorrente conta01 = new ContaCorrente(idConta01, 200, true);
34         ContaCorrente conta02 = new ContaCorrente(idConta02, 0, true);
35
36         // inserindo os clientes criados na lista de clientes do banco
37         List<ContaCorrente> contasDoBanco = new ArrayList<>();
38         contasDoBanco.add(conta01);
39         contasDoBanco.add(conta02);
40
41         gerContas = new GerenciadoraContas(contasDoBanco);
42
43         /* ===== Execução ===== */
44         boolean sucesso = gerContas.transfereValor(idConta01, 100, idConta02);
45
46         /* ===== Verificações ===== */
47         assertTrue(sucesso);
48         assertEquals(conta02.getSaldo(), is(100.0));
49         assertEquals(conta01.getSaldo(), is(100.0));
50     }
```



Project Explorer JUnit

- ▼ SistemaBancario
  - ▼ src
    - ▼ negocio
      - > Cliente.java
      - > ContaCorrente.java
      - > GerenciadoraClientes.java
      - > GerenciadoraContas.java
      - > IdadeNaoPermitidaException.java
      - > Main.java
    - ▼ testes
      - ▼ negocio
        - > GerenciadoraClientesTest.java
        - > GerenciadoraContasTest.java
        - > TodosOsTestes.java
    - > JRE System Library [JavaSE-1.8]
    - > Referenced Libraries

Gerenciador...

Gerenciador...

TodosOsTest...

Gerenciador...

Cliente.java

ContaCorren...

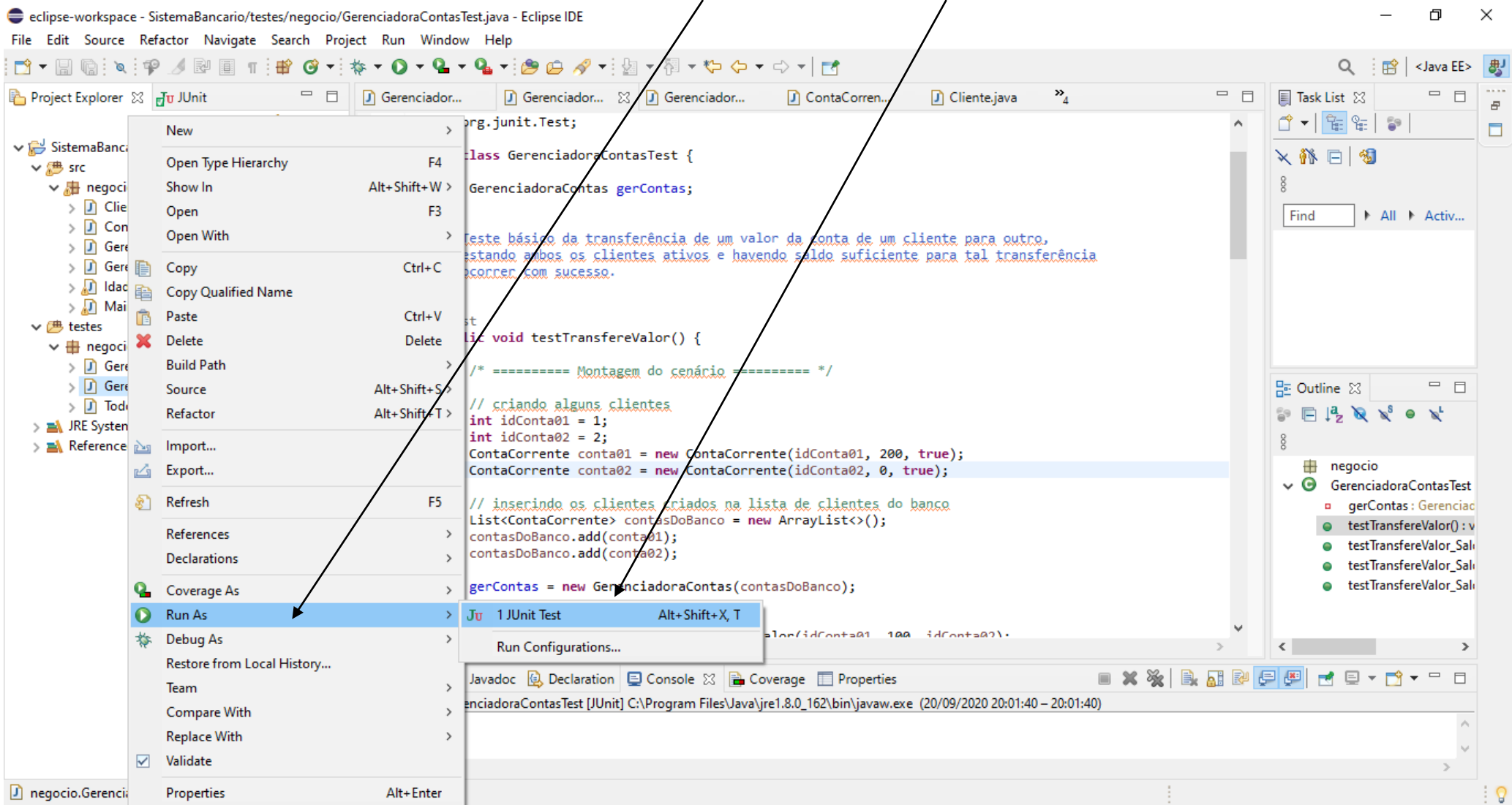
Main.java

»2

&lt;Java EE&gt;

```
1 package negocio;
2
3 import org.junit.runner.RunWith;
4
5
6
7
8 @RunWith(Suite.class)
9 @SuiteClasses({ GerenciadoraClientesTest.class, GerenciadoraContasTest.class})
10 public class TodosOsTestes {
11
12
13 }
14
15
16
17
18
19
20
21
22
23
24
```

## 5. Para rodar o teste utilize a opção: Run As JUnit Test



## 6. Rodando o Teste: verifique se o teste passou

The screenshot shows the Eclipse IDE interface with the following components:

- Top Bar:** eclipse-workspace - SistemaBancario/testes/negocio/GerenciadoraContasTest.java - Eclipse IDE
- Menu Bar:** File Edit Source Refactor Navigate Search Project Run Window Help
- Project Explorer:** Shows the project structure with 'negocio.GerenciadoraContasTest' expanded, listing four test methods: testTransfereValor\_SaldoInsuficiente (0,000 s), testTransfereValor\_SaldoNegativo (0,001 s), testTransfereValor (0,000 s), and testTransfereValor\_SaldoNegativoParaNegativo (0,000 s).
- JUnit Runner:** A green progress bar indicates the test finished after 0,029 seconds. The status shows 'Runs: 4/4', 'Errors: 0', and 'Failures: 0'. An arrow points from the title '6. Rodando o Teste: verifique se o teste passou' to this green bar.
- Editor:** Displays the source code of GerenciadoraContasTest.java, including imports, class definition, and test methods. The code is as follows:

```
7 import org.junit.Test;
8
9 public class GerenciadoraContasTest {
10
11     private GerenciadoraContas gerContas;
12
13     /**
14      * Teste básico da transferência de um valor da conta de um cliente para outro,
15      * estando ambos os clientes ativos e havendo saldo suficiente para tal transferência
16      * ocorrer com sucesso.
17      */
18     @Test
19     public void testTransfereValor() {
20
21         /* ===== Montagem do cenário ===== */
22
23         // criando alguns clientes
24         int idConta01 = 1;
25         int idConta02 = 2;
26         ContaCorrente conta01 = new ContaCorrente(idConta01, 200, true);
27         ContaCorrente conta02 = new ContaCorrente(idConta02, 0, true);
28
29         // inserindo os clientes criados na lista de clientes do banco
30         List<ContaCorrente> contasDoBanco = new ArrayList<>();
31         contasDoBanco.add(conta01);
32         contasDoBanco.add(conta02);
33
34         gerContas = new GerenciadoraContas(contasDoBanco);
35
36         /* ===== Execução ===== */
37         boolean sucesso = gerContas.transfereValor(idConta01, 100, idConta02);
38     }
```
- Task List:** Shows the list of tasks, including 'Find', 'All', and 'Act...'.
- Outline:** Shows the class hierarchy, including 'negocio' and 'GerenciadoraContasTest'.
- Bottom Bar:** Shows the status bar with 'terminated' and the command 'GerenciadoraContasTest [JUnit] C:\Program Files\Java\jre1.8.0\_162\bin\javaw.exe (20/09/2020 20:03:23 - 20:03:24)'.