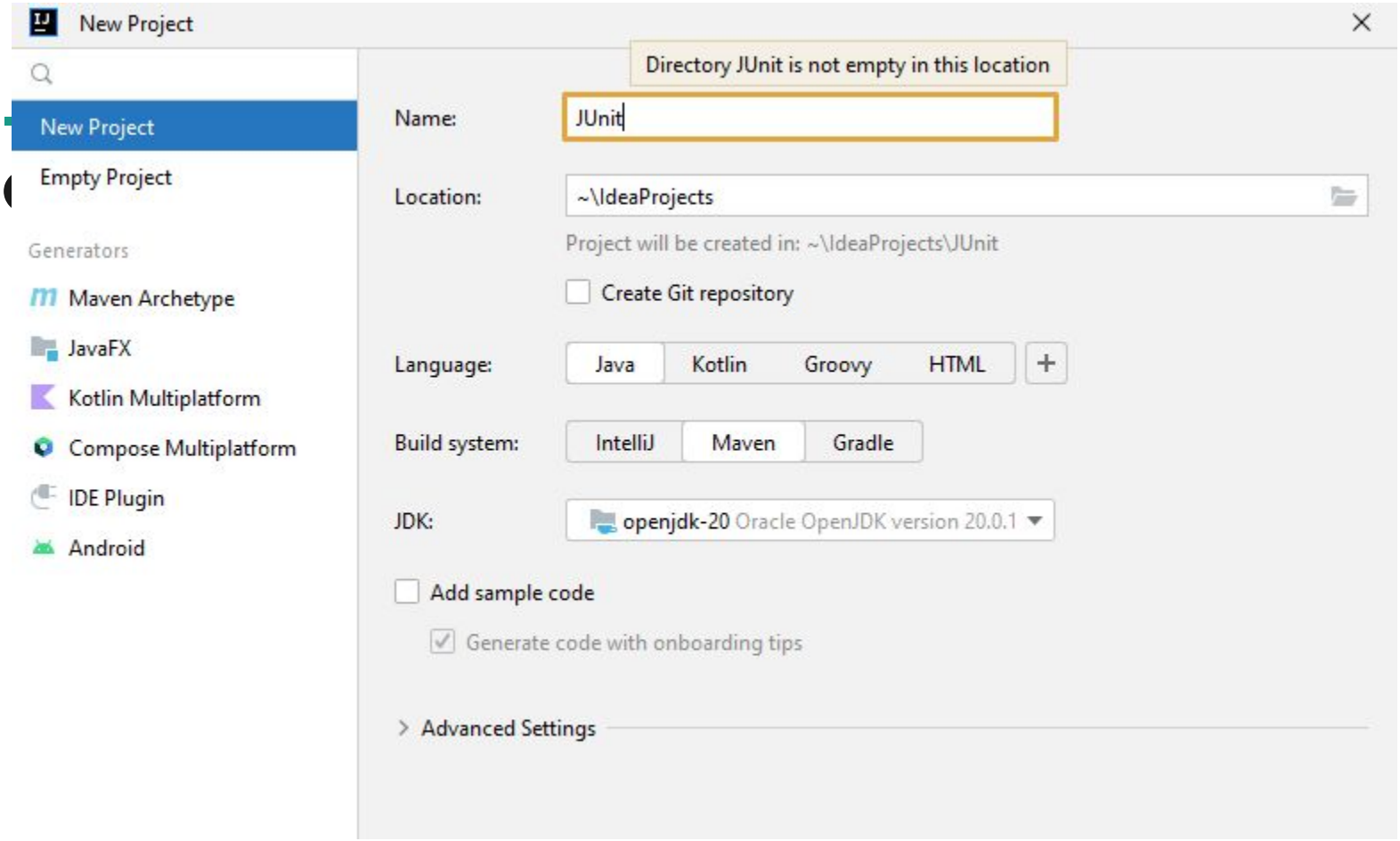
Teste Unitário Junit

Herysson R. Figueiredo
herysson.figueiredo@ufn.edu.br


New Project
New Project
Empty Project
Generators
Maven Archetype
JavaFX
Kotlin Multiplatform
Compose Multiplatform
IDE Plugin
Android
Directory JUnit is not empty in this location
Name:
JUnit
Location:
~\IdeaProjects
Project will be created in: ~\IdeaProjects\JUnit
Create Git repository
Language:
Java
Kotlin
Groovy
HTML
Build system:
IntelliJ
Maven
Gradle
JDK:
openjdk-20 Oracle OpenJDK version 20.0.1
Add sample code
Generate code with onboarding tips
Advanced Settings

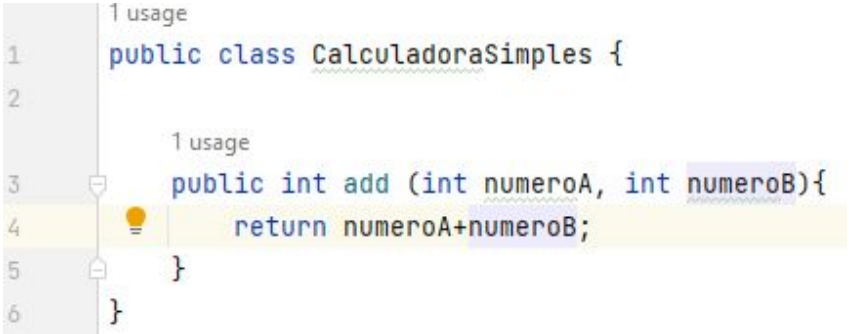
Dependências

Localize o pom.xml e adicione a dependência ao lado.

```xml
<dependencies>
    <dependency>
        <groupId>org.junit.jupiter</groupId>
        <artifactId>junit-jupiter</artifactId>
        <version>5.10.0</version>
    </dependency>
</dependencies>
```

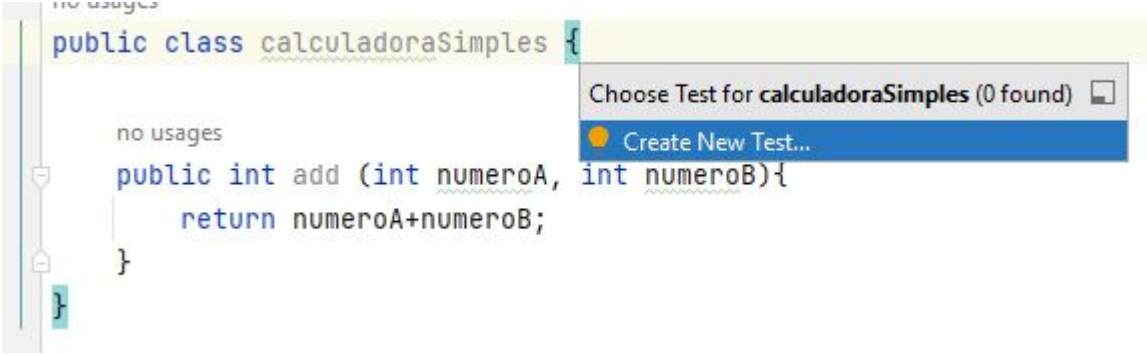
Classe

Crie uma classe que realize a soma de 2 valores inteiros e retorne o resultado

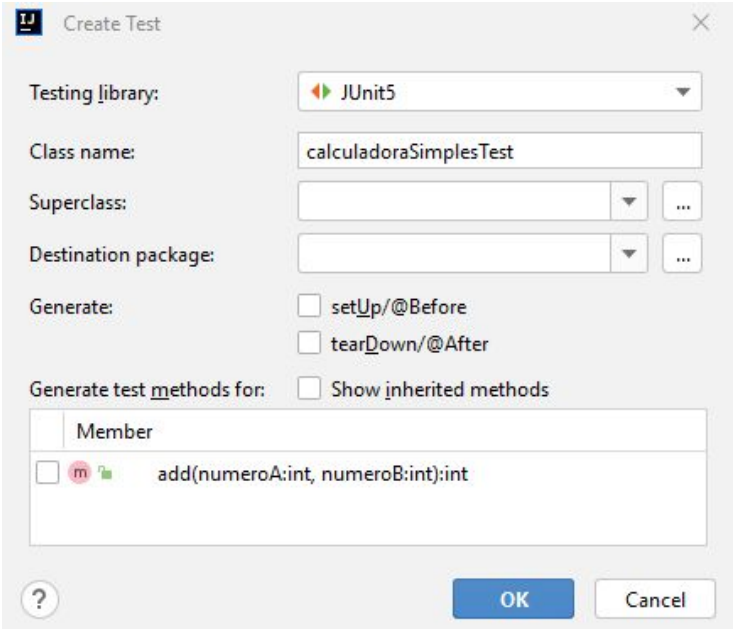


Classe de Teste

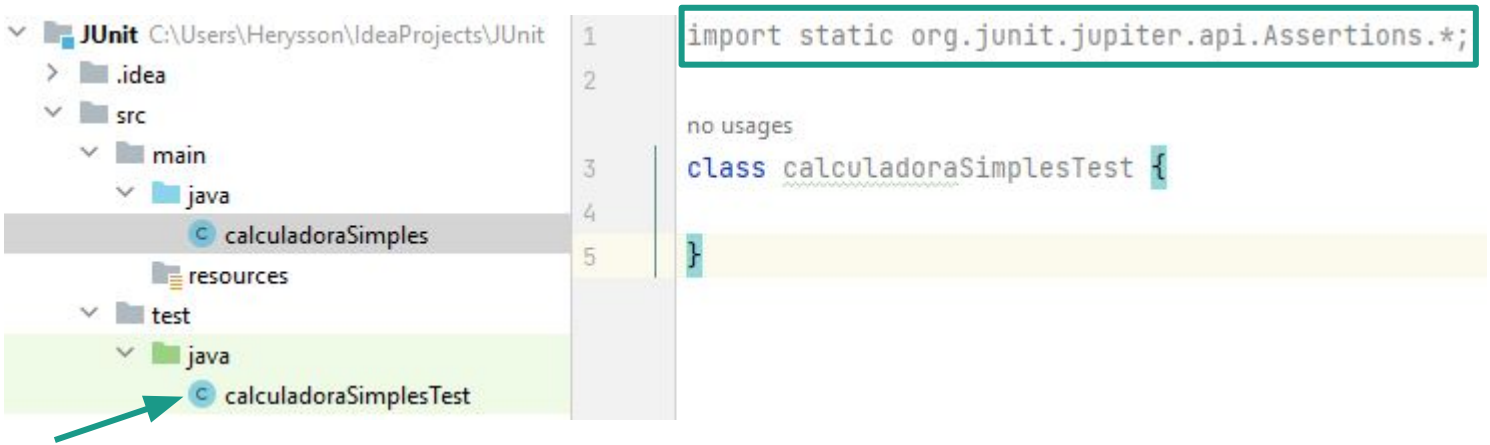
Pressione **ctrl+shift+t** para criação de um teste



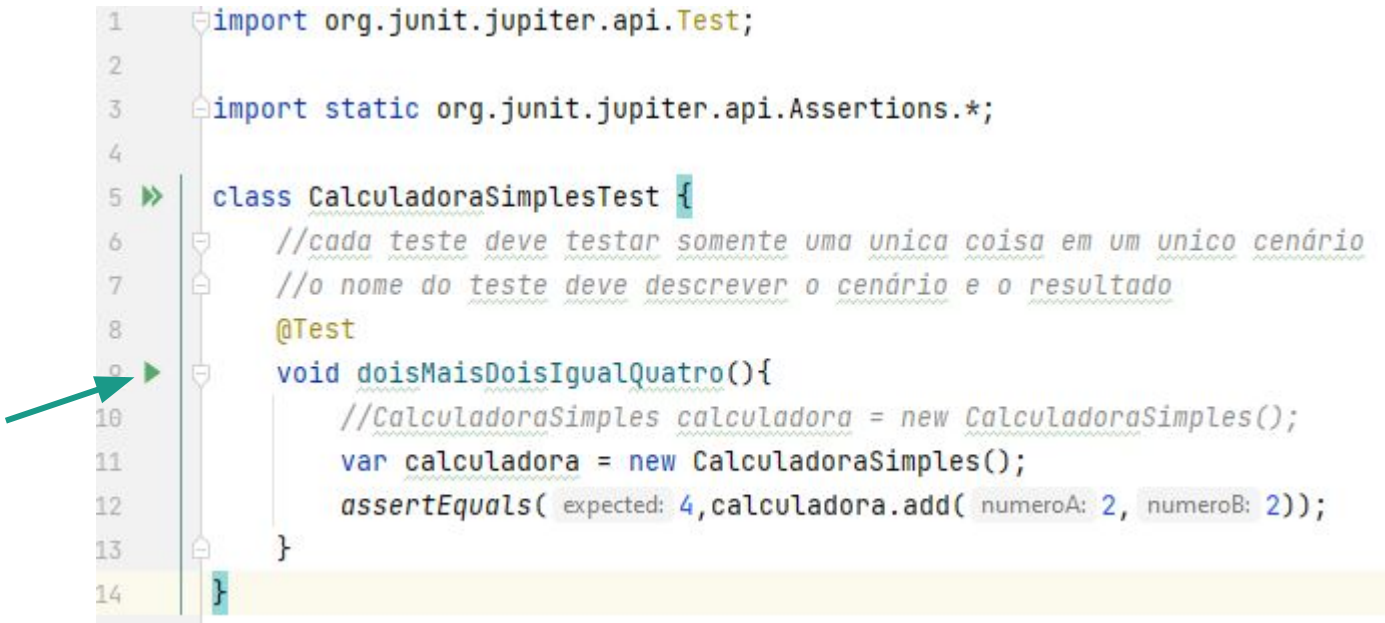
Classe de Teste



Classe de Teste

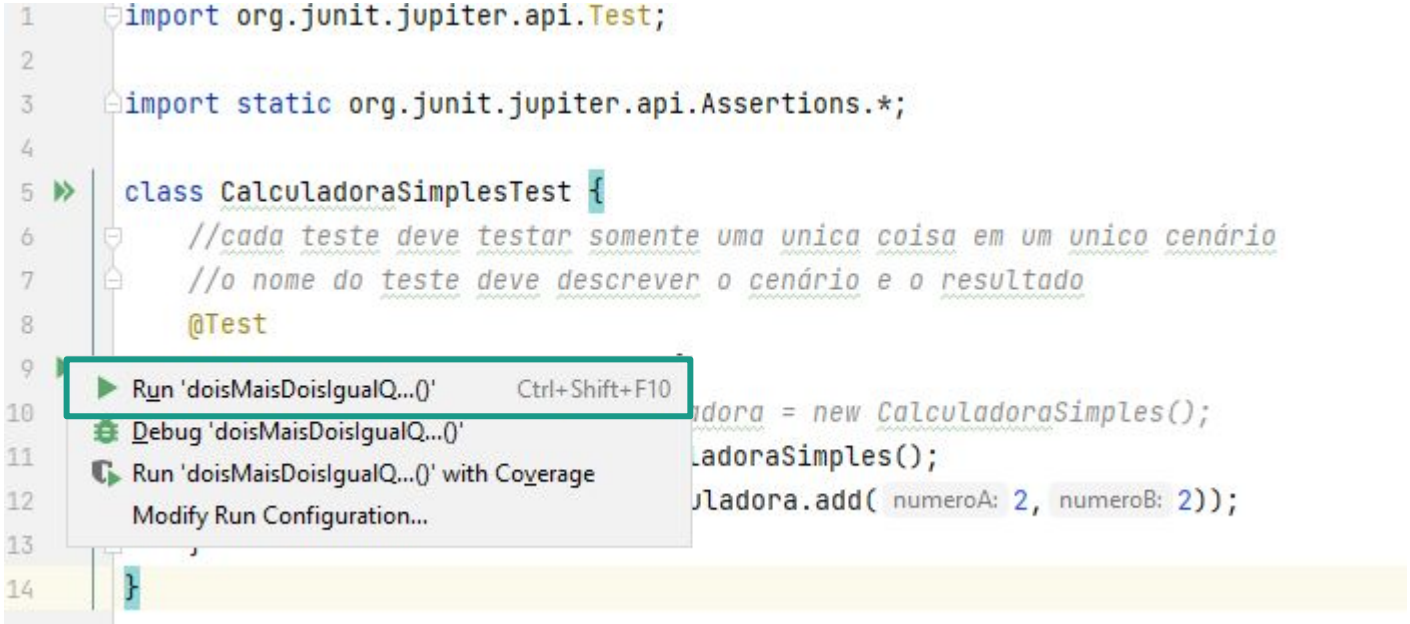


Classe de Teste

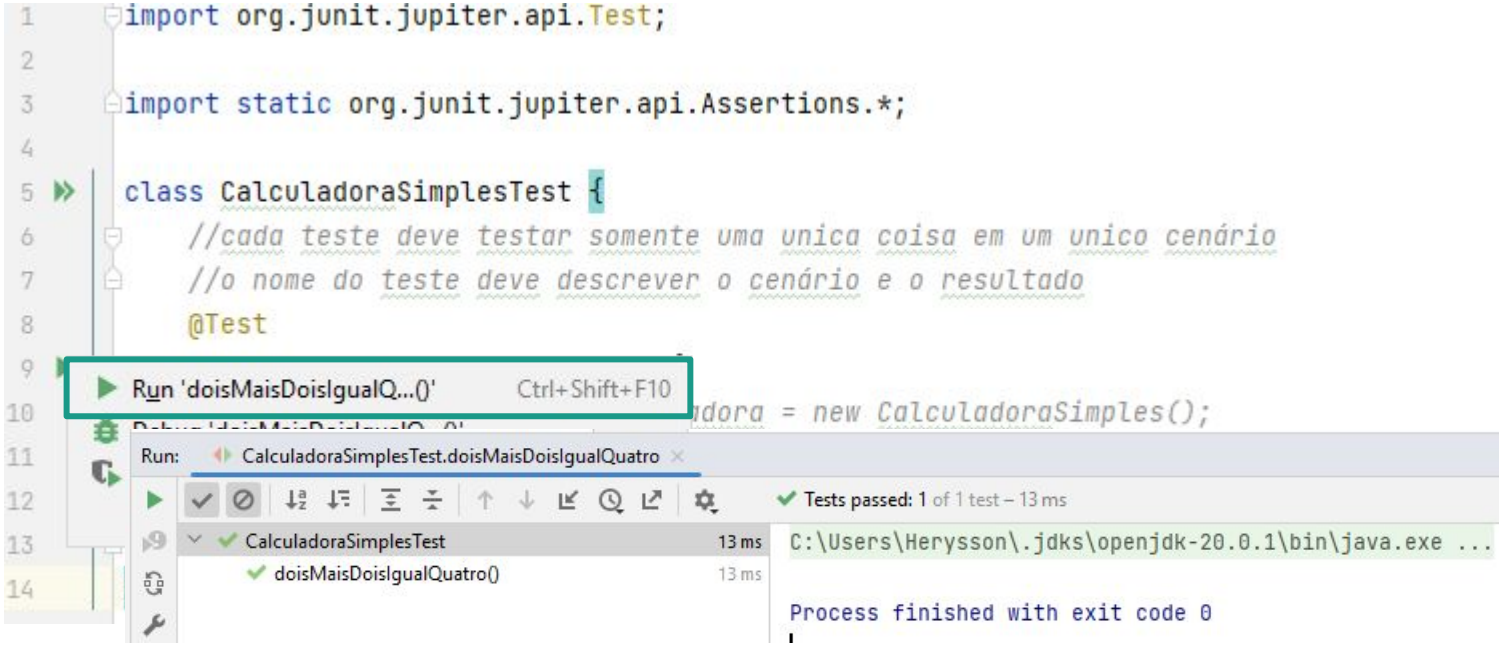


```java
import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.*;

class CalculadoraSimplesTest {
    //cada teste deve testar somente uma unica coisa em um unico cenário
    //o nome do teste deve descrever o cenário e o resultado
    @Test
    void doisMaisDoisIgualQuatro(){
        //CalculadoraSimples calculadora = new CalculadoraSimples();
        var calculadora = new CalculadoraSimples();
        assertEquals( expected: 4,calculadora.add( numeroA: 2, numeroB: 2));
    }
}
```

Classe de Teste



Classe de Teste

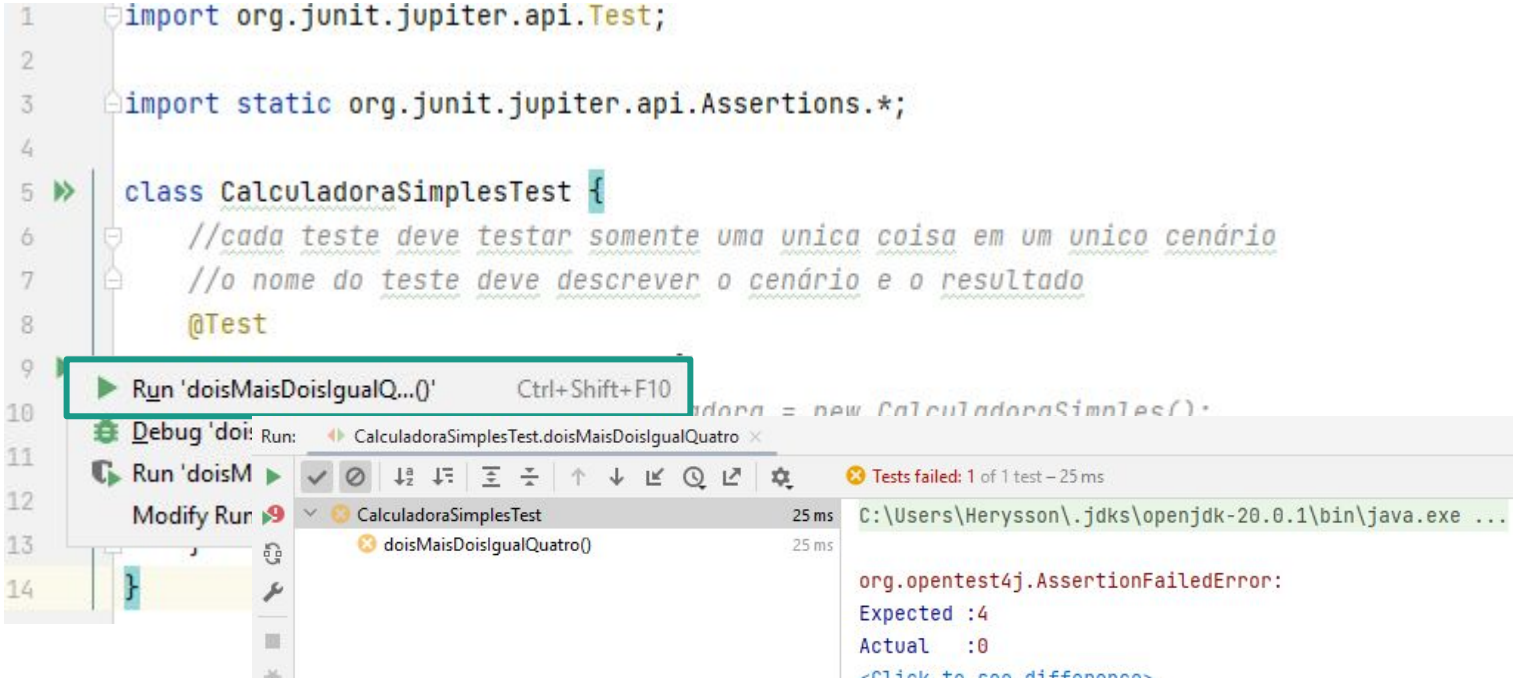


Classe

Alterando a classe CalculadoraSimples

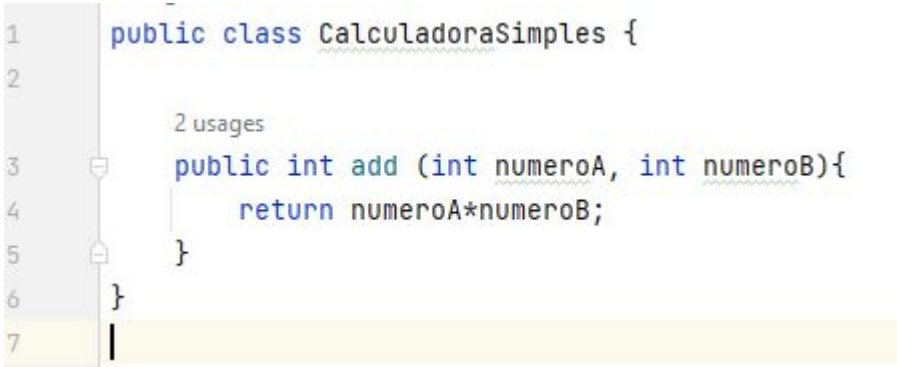
```
1  public class CalculadoraSimples {
2
       1 usage
3      public int add (int numeroA, int numeroB){
4          return numeroA-numeroB;
5      }
6  }
```

Classe de Teste

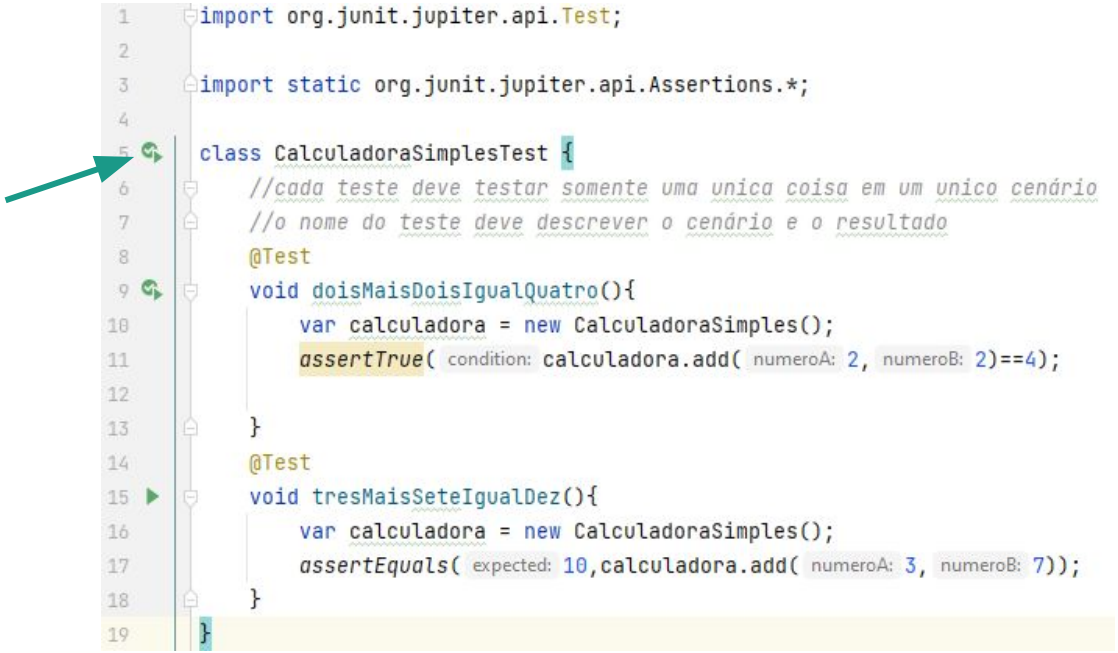


Classe

Alterando a classe CalculadoraSimples

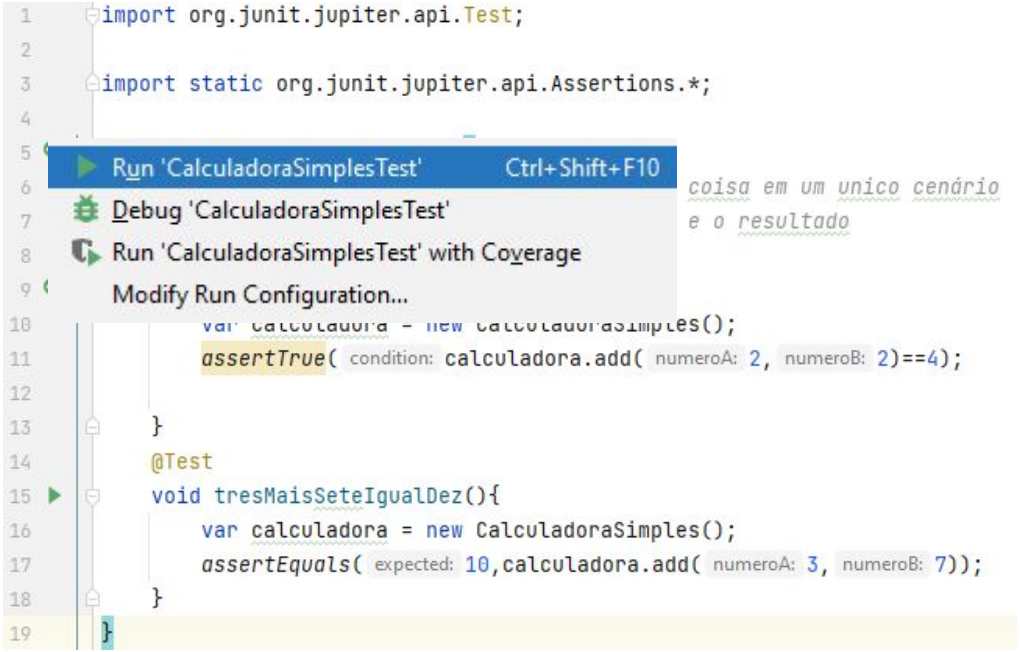


Classe de Teste



```java
import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.*;

class CalculadoraSimplesTest {
    //cada teste deve testar somente uma unica coisa em um unico cenário
    //o nome do teste deve descrever o cenário e o resultado
    @Test
    void doisMaisDoisIgualQuatro(){
        var calculadora = new CalculadoraSimples();
        assertTrue( condition: calculadora.add( numeroA: 2, numeroB: 2)==4);

    }
    @Test
    void tresMaisSeteIgualDez(){
        var calculadora = new CalculadoraSimples();
        assertEquals( expected: 10,calculadora.add( numeroA: 3, numeroB: 7));
    }
}
```

Classe de Teste

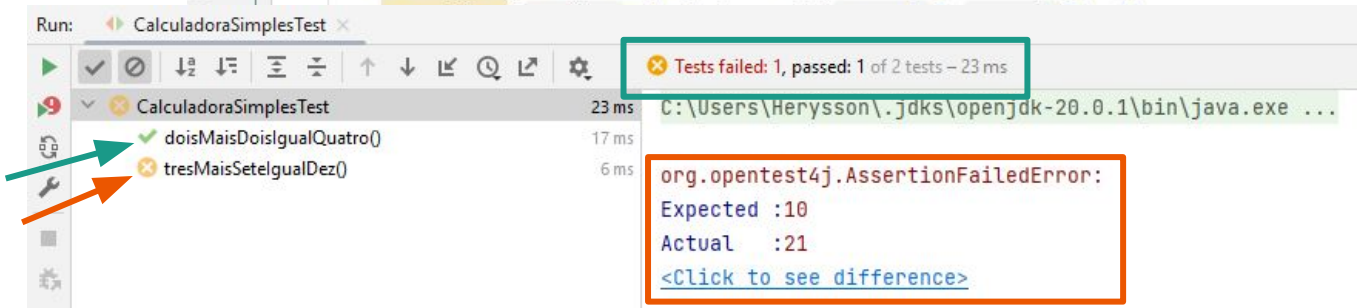


Classe de Teste

```java
import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.*;

class CalculadoraSimplesTest {
    //cada teste deve testar somente uma unica coisa em um unico cenário
    //o nome do teste deve descrever o cenário e o resultado
    @Test
    void doisMaisDoisIgualQuatro(){
        var calculadora = new CalculadoraSimples();
        assertTrue( condition: calculadora.add( numeroA: 2, numeroB: 2)==4);

    }
    @Test
    void tresMaisSeteIgualDez(){
        var calculadora = new CalculadoraSimples();
        assertEquals( expected: 10,calculadora.add( numeroA: 3, numeroB: 7));
    }
}
```

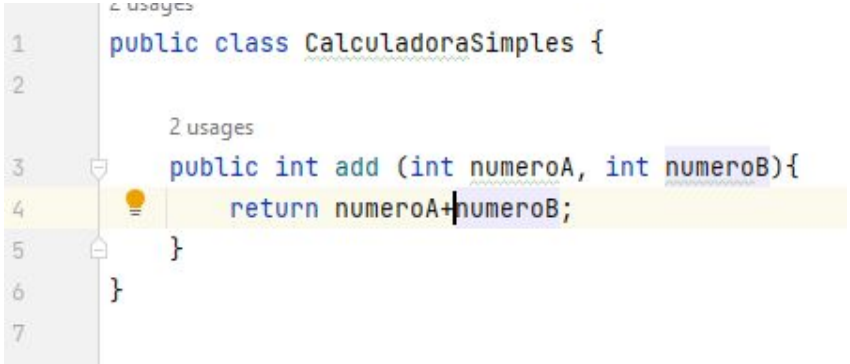
Classe de Teste

```java
import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.*;

class CalculadoraSimplesTest {
    //cada teste deve testar somente uma unica coisa em um unico cenário
    //o nome do teste deve descrever o cenário e o resultado
    @Test
    void doisMaisDoisIgualQuatro(){
        var calculadora = new CalculadoraSimples();
```



Classe

Alterando a classe CalculadoraSimples

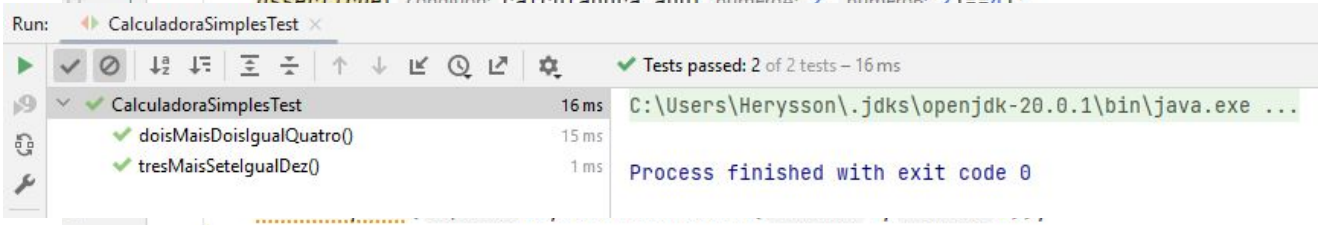


Classe de Teste

```java
import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.*;

class CalculadoraSimplesTest {
    //cada teste deve testar somente uma unica coisa em um unico cenário
    //o nome do teste deve descrever o cenário e o resultado
    @Test
    void doisMaisDoisIgualQuatro(){
        var calculadora = new CalculadoraSimples();
        assertTrue( condition: calculadora.add( numeroA: 2, numeroB: 2)==4);

    }
    @Test
    void tresMaisSeteIgualDez(){
        var calculadora = new CalculadoraSimples();
        assertEquals( expected: 10,calculadora.add( numeroA: 3, numeroB: 7));
    }
}
```

Classe de Teste

```java
import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.*;

class CalculadoraSimplesTest {
    //cada teste deve testar somente uma unica coisa em um unico cenário
    //o nome do teste deve descrever o cenário e o resultado
    @Test
    void doisMaisDoisIgualQuatro(){
        var calculadora = new CalculadoraSimples();
        assertTrue( condition: calculadora.add( numeroA: 2, numeroB: 2)==4);
    }
}
```

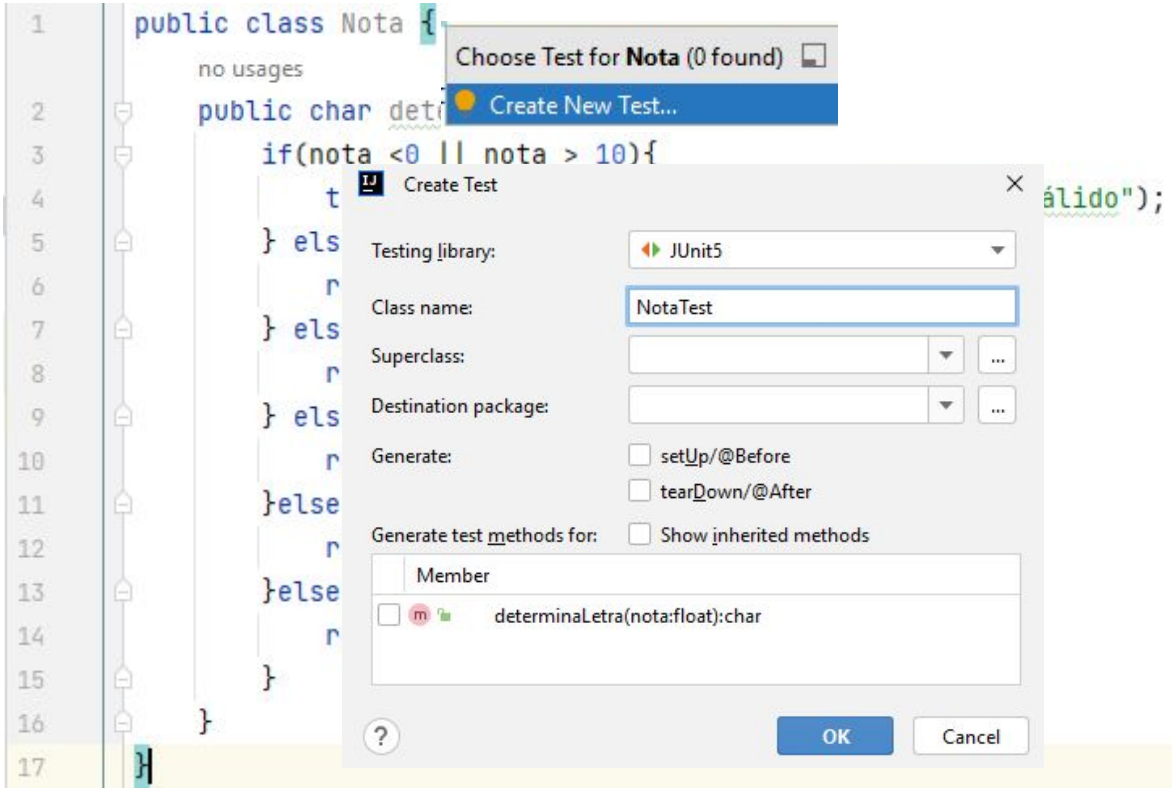


Classe : Nota

```java
public class Nota {
    no usages
    public char determinaLetra(float nota){
        if(nota <0 || nota > 10){
            throw new IllegalArgumentException("Valor inválido");
        } else if (nota < 60){
            return 'F';
        } else if (nota < 70) {
            return 'D';
        } else if (nota <80) {
            return 'C';
        }else if (nota<90){
            return 'B';
        }else {
            return 'A';
        }
    }
}
```

Classe : Nota

Ctrl+Shift+t



Classe de Teste

```
1  import org.junit.jupiter.api.Test;
2
3  import static org.junit.jupiter.api.Assertions.*;
4
5  class NotaTest {
6      @Test
7      void cinquentaENoveRetornaF(){
```

Classe : Nota

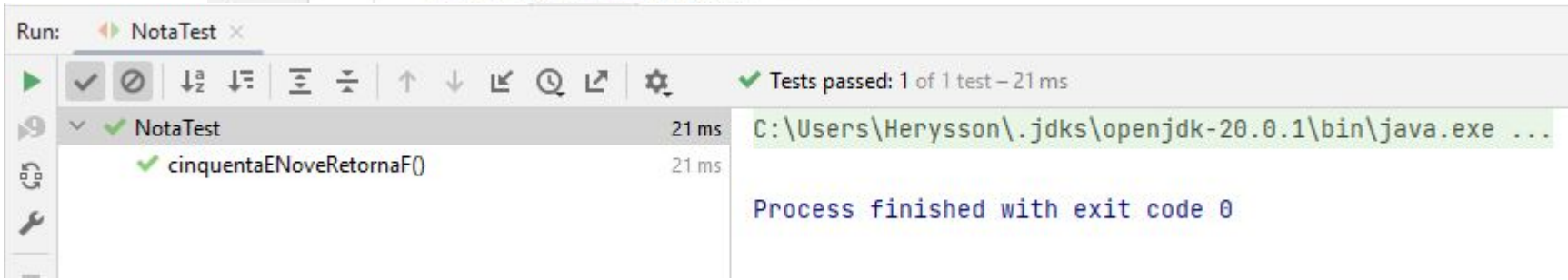
```java
public class Nota {
    no usages
    public char determinaLetra(float nota){
        if(nota <0 || nota > 10){
            throw new IllegalArgumentException("Valor inválido");
        } else if (nota < 60){
            return 'F';
        } else if (nota < 70) {
            return 'D';
        } else if (nota <80) {
            return 'C';
        }else if (nota<90){
            return 'B';
        }else {
            return 'A';
        }
    }
}
```

Classe : Nota

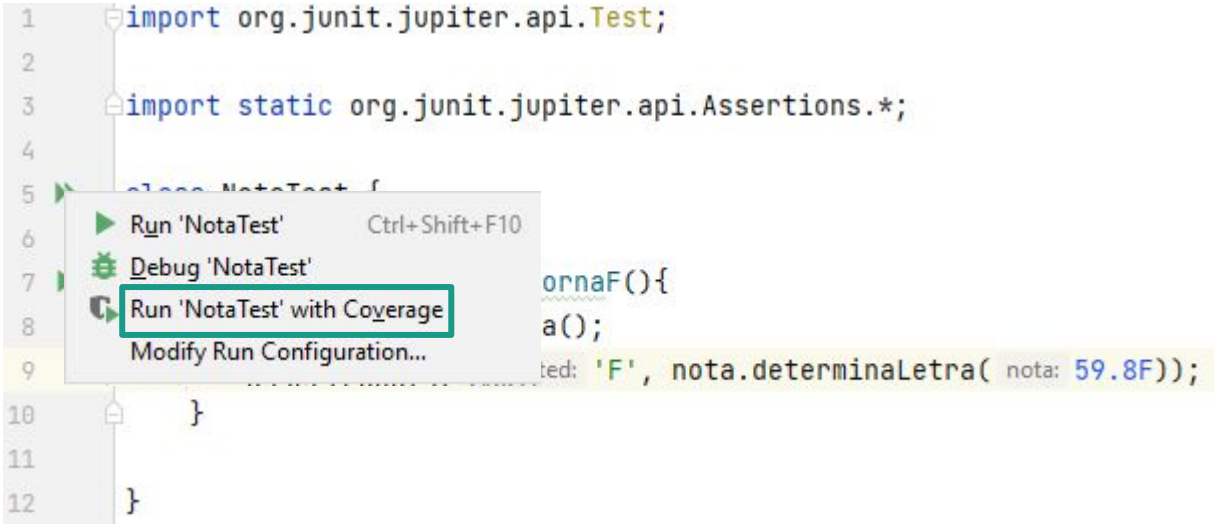
```java
public class Nota {
    1 usage
    public char determinaLetra(float nota){
        if(nota <0 || nota > 100){
            throw new IllegalArgumentException("Valor inválido");
        } else if (nota < 60){
            return 'F';
        } else if (nota < 70) {
            return 'D';
        } else if (nota <80) {
            return 'C';
        }else if (nota<90){
            return 'B';
        }else {
            return 'A';
        }
    }
}
```

Classe de Teste

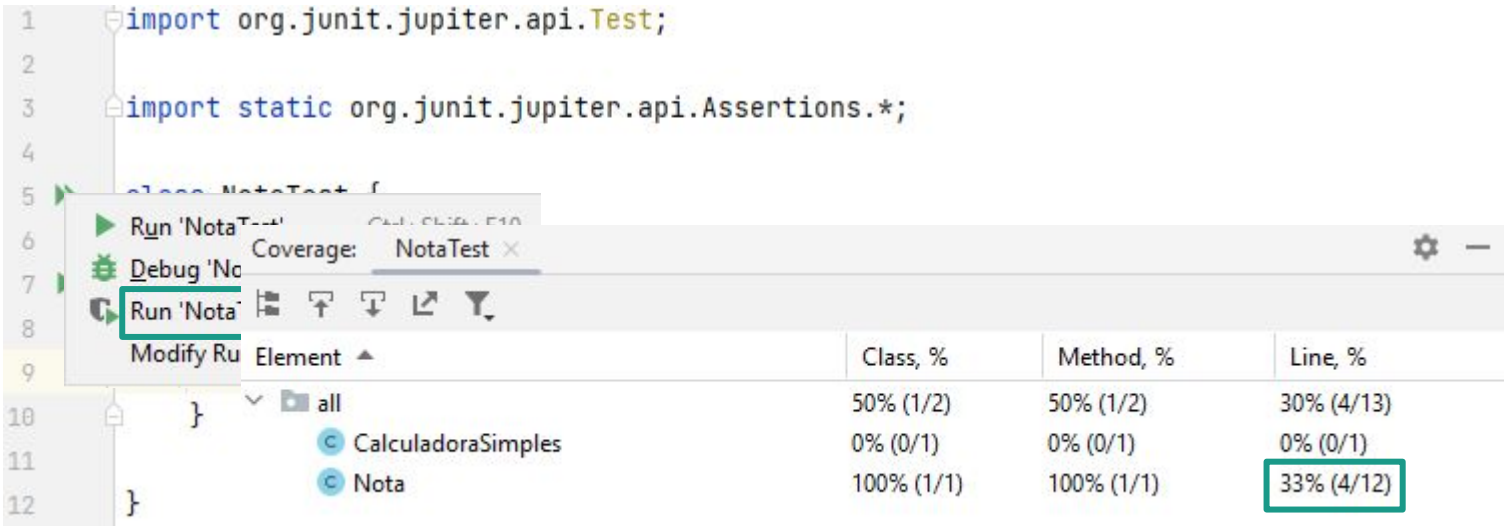
```java
import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.*;

class NotaTest {
    @Test
    void cinquentaENoveRetornaF(){
```



Classe de Teste



Classe de Teste



Classe de Teste

Construa teste para 100% de cobertura.

```
1   public class Nota {
        1 usage
2       public char determinaLetra(float nota){
3           if(nota <0 || nota > 100){
4               throw new IllegalArgumentException("Valor inválido");
5           } else if (nota < 60){
6               return 'F';
7           } else if (nota < 70) {
8               return 'D';
9           } else if (nota <80) {
10              return 'C';
11          }else if (nota<90){
12              return 'B';
13          }else {
14              return 'A';
15          }
16      }
17  }
```

Referências

BRAGA, Pedro Henrique Cacique. Teste de Software. Pearson Education do Brasil. São Paulo. 2016. Disponível na Biblioteca Virtual. DELAMARO, Márcio;

MALDONADO, José Carlos, JINO, Mario. Introdução ao teste de software. Elsevier. Rio de Janeiro. 2007.

PRESSMAN, Roger S. Engenharia de software. 5. Ed. Rio de Janeiro: McGraw Hill, 2002.