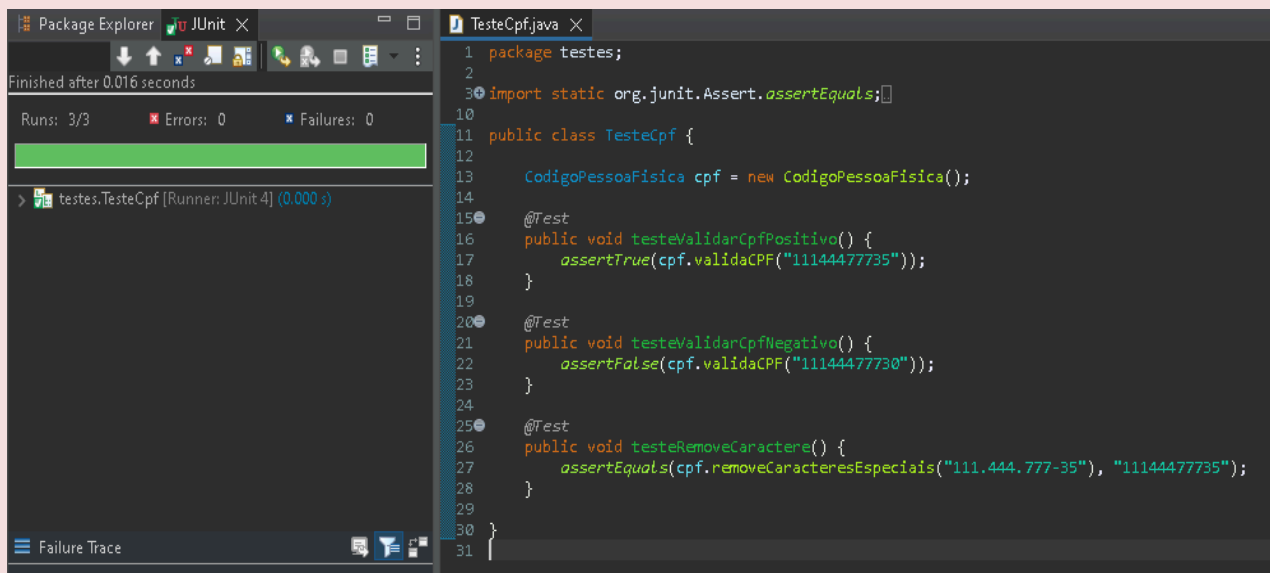


Checkpoint 01- Governança e Melhores Práticas em Desenvolvimento de Sistemas

Gregory Cedraz Araujo - Rm 94404

Anne de Oliveira Rezende - Rm 94580

1- Teste: Método `validaCpf` com retorno `positivo(true)`, logo em seguida um novo teste com retorno `negativo(false)`; Método `removeCaracteresEspeciais` retornando apenas o cpf em números inteiros.



The screenshot displays an IDE interface with two main panels. The left panel shows the 'Package Explorer' and 'JUnit' tabs. It indicates that the test suite 'testes.TesteCpf' has finished after 0.016 seconds, with 3/3 runs, 0 errors, and 0 failures. The right panel shows the source code of the 'TesteCpf.java' file. The code includes package declarations, imports, and three test methods: 'testeValidarCpfPositivo', 'testeValidarCpfNegativo', and 'testeRemoveCaractere'.

```
1 package testes;
2
3 import static org.junit.Assert.assertEquals;
4
10
11 public class TesteCpf {
12
13     CodigoPessoaFisica cpf = new CodigoPessoaFisica();
14
15     @Test
16     public void testeValidarCpfPositivo() {
17         assertTrue(cpf.validaCPF("11144477735"));
18     }
19
20     @Test
21     public void testeValidarCpfNegativo() {
22         assertFalse(cpf.validaCPF("11144477730"));
23     }
24
25     @Test
26     public void testeRemoveCaractere() {
27         assertEquals(cpf.removeCaracteresEspeciais("111.444.777-35"), "11144477735");
28     }
29
30 }
31
```

2- Classe: ValidadoraAcesso com todos os cpfs válidos declarados no array cpfsAutorizados, juntamente com a verificação de acesso dos cpfs. Em seguida os retornos da class testados pelo JUnit.

```
1 package programa;  
2  
3 public class ValidadoraAcesso {  
4  
5     String[] cpfsAutorizados= {"97346812574", "75943468416", "71692037186"};  
6  
7     public boolean cpfValido(String cpf) {  
8  
9         boolean resultado = false;  
10  
11         for (int i = 0; i < cpfsAutorizados.length; i++) {  
12             if(cpf == cpfsAutorizados[i]) {  
13                 resultado = true;  
14             }  
15  
16             else if(cpf != cpfsAutorizados[i] && resultado == false){  
17                 resultado = false;  
18             }  
19         }  
20  
21         return resultado;  
22     }  
23 }  
24 }  
25 }
```

Package Explorer JUnit X

Finished after 0.017 seconds

Runs: 2/2 Errors: 0 Failures: 0

testes.TesteValidadoraAcesso [Runner: JUnit4] (0.000 s)

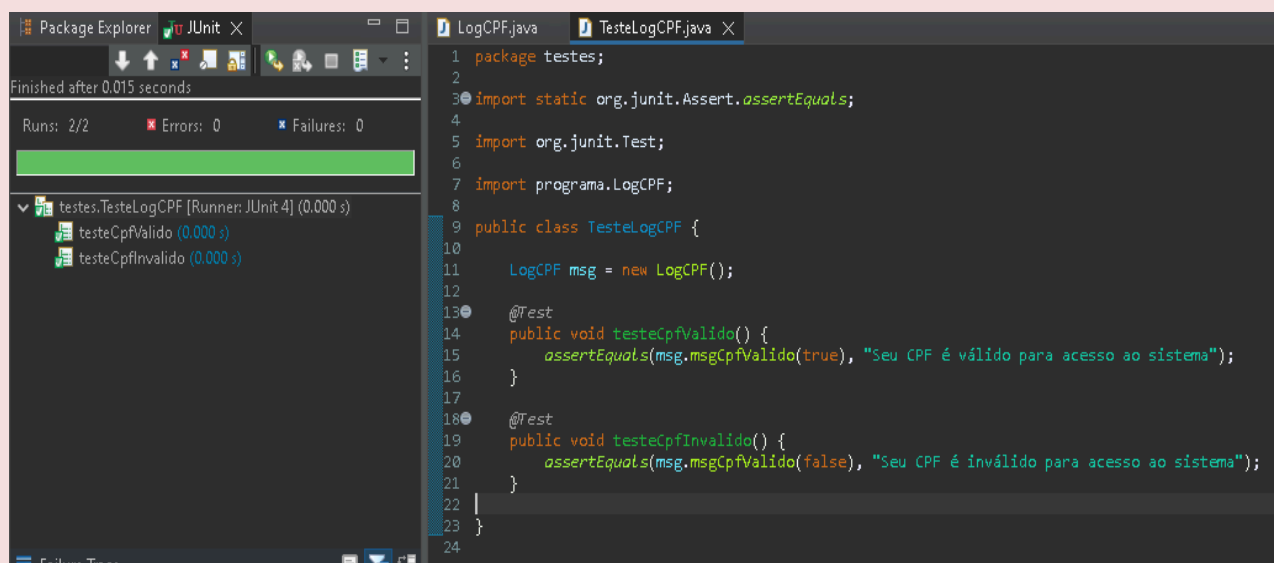
- testeCpfValido (0.000 s)
- testeCpfInvalido (0.000 s)

Failure Trace

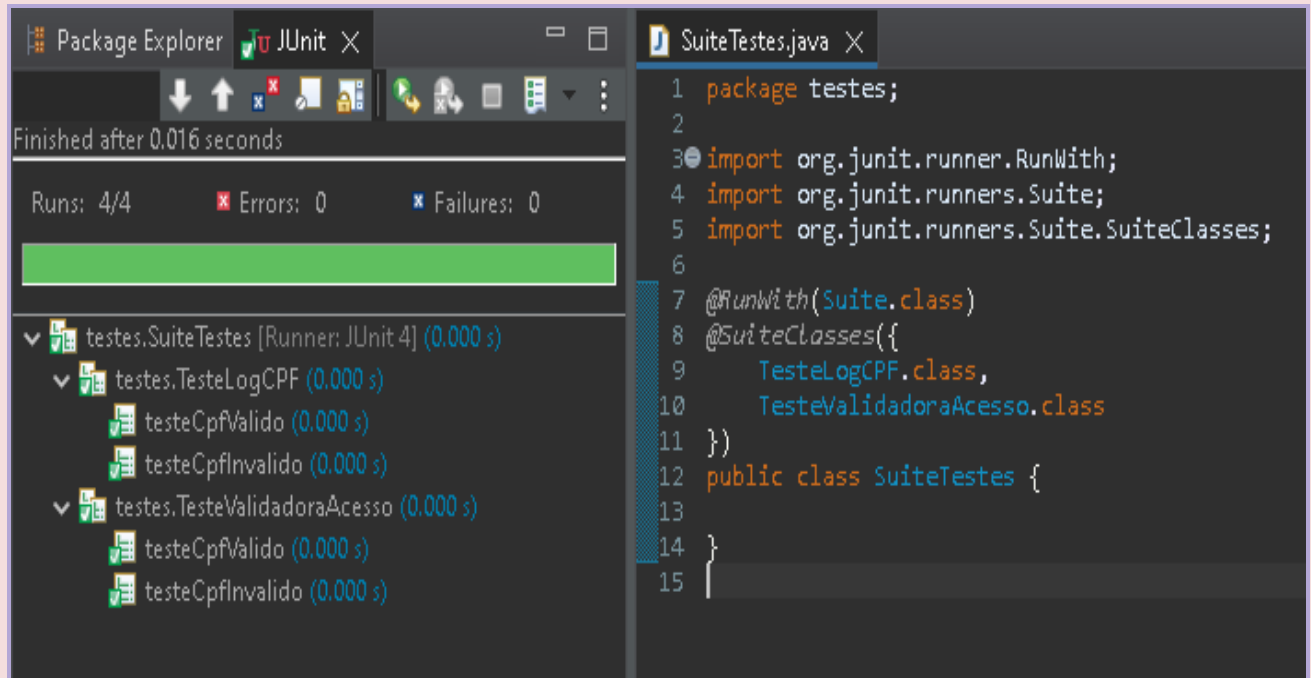
```
1 package testes;  
2  
3 import static org.junit.Assert.assertFalse;  
4 import static org.junit.Assert.assertTrue;  
5  
6 import org.junit.Test;  
7  
8 import programa.ValidadoraAcesso;  
9  
10 public class TesteValidadoraAcesso {  
11  
12     ValidadoraAcesso acesso = new ValidadoraAcesso();  
13  
14     @Test  
15     public void testeCpfValido() {  
16         assertTrue(acesso.cpfValido("97346812574"));  
17     }  
18  
19     @Test  
20     public void testeCpfInvalido() {  
21         assertFalse(acesso.cpfValido("57143945682"));  
22     }  
23 }  
24 }  
25 }
```

3- Classe: LogCpf com uma confirmação em boolean, e retornando a mensagem de validez ou invalidez do acesso. em seguida os testes de acesso feitos no JUnit.

```
1 package programa;
2
3 public class LogCPF {
4
5     public String msgCpfValido(boolean valido) {
6
7         String msg = "";
8
9         if(valido == true) {
10             msg = "Seu CPF é válido para acesso ao sistema";
11         }
12
13         else if(valido == false) {
14             msg = "Seu CPF é inválido para acesso ao sistema";
15         }
16
17         return msg;
18     }
19 }
20
21 }
```



4- Suite: Todas as classes e métodos em um teste.



The screenshot displays an IDE interface with two main panes. The left pane, titled 'Package Explorer', shows a tree structure of test classes and methods. The right pane, titled 'SuiteTestes.java', shows the source code of the test suite.

Package Explorer:

- testes.SuiteTestes [Runner: JUnit 4] (0.000 s)
 - testes.TesteLogCPF (0.000 s)
 - testeCpfValido (0.000 s)
 - testeCpfInvalido (0.000 s)
 - testes.TesteValidadoraAcesso (0.000 s)
 - testeCpfValido (0.000 s)
 - testeCpfInvalido (0.000 s)

SuiteTestes.java:

```
1 package testes;
2
3 import org.junit.runner.RunWith;
4 import org.junit.runners.Suite;
5 import org.junit.runners.Suite.SuiteClasses;
6
7 @RunWith(Suite.class)
8 @SuiteClasses({
9     TesteLogCPF.class,
10    TesteValidadoraAcesso.class
11 })
12 public class SuiteTestes {
13
14 }
15
```