

Vitor Torres Dantas RM: 88415

Matheus de Oliveira RM: 88430

Desafio 1

The screenshot shows the Eclipse IDE interface. The main editor displays the `CodigoPessoaFisicaTest.java` file. The code includes imports for JUnit 5, a class definition `CodigoPessoaFisicaTest`, and three test methods: `testeValidaCPFInvalido`, `testeValidaCPFValido`, and `formataCPF`. The left sidebar shows the Package Explorer with the test class selected. Below it, the JUnit runner output shows that all three tests passed successfully. The bottom status bar indicates the current time is 8:50 PM on 3/6/2023.

```
1 import static org.junit.jupiter.api.Assertions.assertEquals;
2
3 import org.junit.jupiter.api.Test;
4
5 class CodigoPessoaFisicaTest {
6
7
8     @Test
9     public void testeValidaCPFInvalido() {
10         String valorCpf = "11144477705";
11         Boolean digitoReal = null;
12
13         CodigoPessoaFisica cdg = new CodigoPessoaFisica();
14         Boolean digitoEsperado = false;
15         digitoReal = cdg.validaCPF(valorCpf);
16         assertEquals(digitoEsperado, digitoReal);
17     }
18
19     @Test
20     public void testeValidaCPFValido() {
21         String valorCpf = "37064170817";
22         Boolean digitoReal = null;
23
24         CodigoPessoaFisica cdg = new CodigoPessoaFisica();
25         Boolean digitoEsperado = true;
26         digitoReal = cdg.validaCPF(valorCpf);
27         assertEquals(digitoEsperado, digitoReal);
28     }
29
30     @Test
31     public void formataCPF() {
32         String valorCpf = "362.598.688-47";
33         String cpfEsperado = "36259868847";
34
35         CodigoPessoaFisica cdg = new CodigoPessoaFisica();
36         String cpfReal = cdg.removeCaracteresEspeciais(valorCpf);
37         assertEquals(cpfReal, cpfEsperado);
38     }
39 }
40
```

JUnit Runner Output:

```
Finished after 0.092 seconds
Runs: 3/3 Errors: 0 Failures: 0
CodigoPessoaFisicaTest [Runner: JUnit 5] (0.018 s)
  testeValidaCPFValido() (0.015 s)
  testeValidaCPFInvalido() (0.001 s)
  formataCPF() (0.000 s)
```

Description	Resource	Path	Location	Type
0 items				

Desafio 2

```
ValidadoraAcesso x ValidadoraAcesso LogCpf.java LogCPFTest.java AllTests.java
1 import java.util.Arrays;
2 import java.util.List;
3
4 public class ValidadoraAcesso {
5     private List<String> cpfsAutorizados;
6
7     public ValidadoraAcesso(String[] cpfsAutorizados) {
8         this.cpfsAutorizados = Arrays.asList(cpfsAutorizados);
9     }
10
11     public boolean validarCPF(String cpf) {
12         return cpfsAutorizados.contains(cpf);
13     }
14 }
15
```

```
ValidadoraAcesso ValidadoraAcesso LogCpf.java LogCPFTest.java AllTests.java
1 import org.junit.Assert;
2 import org.junit.Before;
3 import org.junit.Test;
4
5 public class ValidadoraAcessoTeste {
6
7     private ValidadoraAcesso validadora;
8
9     @Before
10     public void setUp() {
11         String[] cpfsAutorizados = {"11111111111", "22222222222", "33333333333"};
12         validadora = new ValidadoraAcesso(cpfsAutorizados);
13     }
14
15     @Test
16     public void testValidarCPF() {
17         Assert.assertTrue(validadora.validarCPF("11111111111"));
18         Assert.assertTrue(validadora.validarCPF("22222222222"));
19         Assert.assertTrue(validadora.validarCPF("33333333333"));
20         Assert.assertFalse(validadora.validarCPF("44444444444"));
21         Assert.assertFalse(validadora.validarCPF("00000000000"));
22     }
23 }
24
25
26
27
```

```

1 public class LogCpf {
2     private ValidadoraAcesso validadoraAcesso;
3
4     public LogCpf(ValidadoraAcesso validadoraAcesso) {
5         this.validadoraAcesso = validadoraAcesso;
6     }
7
8     public String formatarMensagem(String cpf) {
9         if (validadoraAcesso.validarCPF(cpf)) {
10             return "Seu CPF é válido para acesso ao sistema";
11         } else {
12             return "Seu CPF é inválido para acesso ao sistema";
13         }
14     }
15 }

```

```

1 import org.junit.Assert;
2 import org.junit.Before;
3 import org.junit.jupiter.api.Test;
4
5 public class LogCPFTest {
6
7     private ValidadoraAcesso validadora;
8     private LogCpf log;
9
10    @Before
11    public void setUp() {
12        String[] cpfsAutorizados = { "11111111111", "22222222222", "33333333333" };
13        validadora = new ValidadoraAcesso(cpfsAutorizados);
14        log = new LogCpf(validadora);
15    }
16
17    @Test
18    public void testFormatarMensagemValida() {
19        String cpfValido = "11111111111";
20
21        String mensagemValida = "Seu CPF é válido para acesso ao sistema";
22
23        Assert.assertEquals(mensagemValida, log.formatarMensagem(cpfValido));
24    }
25
26    @Test
27    public void testFormatarMensagemInvalida() {
28        String cpfInvalido = "44444444444";
29
30        String mensagemInvalida = "Seu CPF é inválido para acesso ao sistema";
31
32        Assert.assertEquals(mensagemInvalida, log.formatarMensagem(cpfInvalido));
33    }
34 }
35
36
37




```

Finished after 0.087 seconds

Runs: 2/2

✖ Errors: 2

⚙ Failures: 0

- ▼  LogCPFTTest [Runner: JUnit 5] (0.019 s)
 -  testFormatarMensagemValida() (0.017 s)
 -  testFormatarMensagemInvalida() (0.002 s)

≡ Failure Trace



```
J! java.lang.NullPointerException: Cannot invoke "LogCpf.form
  at LogCPFTest.testFormatarMensagemValida(LogCPFTest.jav
  at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
  at java.base/java.util.ArrayList.forEach(ArrayList.java:1511)
```