



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Instituto de Ciências Exatas e de Informática

Departamento de Ciência da Computação

Disciplina: Engenharia de Software III

Teste com Junit

*

Vinícius Francisco da Silva - 576920¹

Lúcio Castilho Andrade Pinto - 598890²

Pedro Paulo Andrade Meireles - 597697³

*Trabalho apresentado para a disciplina de Engenharia de Software III.

¹ Aluno do Programa de Graduação em Ciência da Computação, Brasil – vinicius.silva.1046664@sga.pucminas.br.

² Aluno do Programa de Graduação em Ciência da Computação, Brasil – lucio.pinto@sga.pucminas.br.

³ Aluno do Programa de Graduação em Ciência da Computação, Brasil – pmeireles@sga.pucminas.br.

1 DEFINA REQUISITOS QUE ESTÃO SENDO IMPLEMENTADOS NO CASO DE USO

- Permitir a venda apenas para maiores de 18 anos!
- Realizar a venda apenas para as pessoas que informarem o CPF!
- Não aceitar cartão alimentação na compra!
- Verificar a idade do comprador antes de efetuar a compra!

2 IMPLEMENTE A CLASSE EM JAVA. DE FORMA PROPOSITAL, GERE DEFEITOS NO CÓDIGO

```
1 import java.util.Calendar;
2 import java.util.Date;
3 import java.util.List;
4
5 public class Venda{
6     private int dia;
7     private int hora;
8     private List<ProdutoVendido> produtosvendidos;
9     private Comprador comprador;
10
11     public Venda(int dia,int hora,List<ProdutoVendido>
produtosvendidos,Comprador comprador){
12         this.dia = dia;
13         this.hora = hora;
14         this.comprador = comprador;
15     } // End Venda ()
16
17     public boolean vendaAutorizada(int idade){
18         return idade >= 17 ? true : false;
19     } // End vendaAutorizada ()
20
21     public void efetuarVenda(int idade){
22         if(vendaAutorizada(idade)) System.out.println("
VENDA EFETUADA COM SUCESSO!!");
23         else System.out.println("VENDA NÃO AUTORIZADA!");
24     } // End efetuarVenda ()
25
26     public double getValorTotal(){
27         double total = 0;
28         for(ProdutoVendido produtoVendido :
produtosvendidos){
29             total = (produtoVendido.getPreco() * (
produtoVendido.getQtd()+2));
30         } // End for
31         return total;
32     } // End total
33
34     public int getDia(){
35         return dia;
36     } // End getDia ()
37
38     public void setDia(int dia){
39         this.dia = dia;
40     } // End setDia ()
41
```

```
42     public int getHora(){
43         return hora;
44     }// End getHora()
45
46     public void setHora(int hora){
47         this.hora = hora;
48     }// End setHora()
49
50     public Comprador getComprador(){
51         return comprador;
52     }// End getComprador()
53
54     public void setComprador(Comprador comprador){
55         this.comprador = comprador;
56     }// End setComprador()
57 }// End Venda
58
```

```
1 public class ProdutoVendido{
2     private int codigoProduto;
3     private int qtd;
4     private double preco;
5
6     public ProdutoVendido(int codigoProduto, int qtd,
7         double preco){
8         this.codigoProduto = codigoProduto;
9         this.qtd = qtd;
10        this.preco = preco;
11    } // End ProdutoVendido
12
13    public int getCodigoProduto(){ return codigoProduto; }
14    // End getCodigoProduto()
15    public void setCodigoProduto(int codigoProduto){ this.
16        codigoProduto = codigoProduto; } // End setCodigoProduto()
17    public int getQtd(){ return qtd; } // End getQtd()
18    public void setQtd(int qtd){ this.qtd = qtd; } // End
19    setQtd()
20    public double getPreco(){ return preco; } // End
21    getPreco()
22    public void setPreco(double preco){ this.preco = preco
23    ; } // End setPreco()
24 } // End ProdutoVendido
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

```
1 import java.util.Calendar;
2 import java.util.Date;
3
4 public class Comprador implements StubComprador{
5     private String nome;
6     private String cpf;
7     private int dia;
8     private int mes;
9     private int ano;
10    private char sexo;
11    private String numeroCartao;
12
13    public Comprador(String nome,String cpf,int dia,int
mes,int ano,char sexo,String numeroCartao){
14        this.nome = nome;
15        this.cpf = cpf;
16        this.dia = dia;
17        this.mes = mes;
18        this.ano = ano;
19        this.sexo = sexo;
20        this.numeroCartao = numeroCartao;
21    }// End Comprador()
22
23    public String getNome(){
24        return nome;
25    }// End getNome()
26    public void setNome(String nome){
27        this.nome = nome;
28    }// End setNome()
29    public String getNumeroCartao(){ return numeroCartao;
30    }// End getNumeroCartao()
31    public void setNumeroCartao(String numeroCartao){
32        this.numeroCartao = numeroCartao;
33    }// End setNumeroCartao()
34    public String getCpf(){ return cpf; }// End getCpf()
35    public void setCpf(String cpf){ this.cpf = cpf; }//
End setCpf
36    public int getDia(){ return dia; }// End getDia()
37    public void setDia(int dia){ this.dia = dia; }// End
setDia()
38    public int getMes(){ return mes; }// End getMes()
39    public void setMes(int mes){ this.mes = mes; }// End
setMes()
40    public int getAno(){ return ano; }// End getAno()
41    public void setAno(int ano){ this.ano = ano; }// End
```

```
40 setAno() ]
41     public char getSexo() {
42         return sexo;
43     } // End getSexo()
44     public void setSexo(char sexo) {
45         this.sexo = sexo;
46     } // End setSexo()
47     public int getIdade() {
48         Calendar calendar = Calendar.getInstance();
49         int anoatual = calendar.get(Calendar.YEAR);
50         return (this.ano-anoatual);
51     } // End getIdade()
52 } // End Comprador
53
```

Defeitos:

- Verifica de forma errada a idade do Comprador
- Não verifica o preço do produto Não verifica o código do produto
- Não verifica a autenticidade do CPF

3 CONSTRUA OS STUBS E DRIVERS QUE FOREM NECESSÁRIOS, SE FOR O CASO

```
1 import java.util.ArrayList;
2 import java.util.Calendar;
3 import java.util.List;
4
5 public class VendaDriver implements StubProdutoVendido,
StubComprador{
6     public void test(){
7         int erro = 0;
8         Calendar cal = Calendar.getInstance();
9         int date = cal.get(Calendar.DATE);
10        int hour = cal.get(Calendar.HOUR);
11
12        List<ProdutoVendido> produtoVendidoList = new
ArrayList<>();
13        produtoVendidoList.add(new ProdutoVendido(1,2,5));
14
15        Comprador comprador = new Comprador(
16            "Ana Carla", "344.553.245-43",
17            21,2,2002,
18            'F', "2222442224");
19
20        Venda venda = new Venda(date, hour,
produtoVendidoList, comprador);
21
22        if(venda.vendaAutorizada(17)) erro = 0;
23        if(this.getIdade() < 18) erro = 1;
24        if(venda.getDia() != date) erro = 2;
25        if(venda.getHora() != hour) erro = 3;
26        if(!this.getCpf().equals("344.553.245-43")) erro =
4;
27        if(this.getSexo() != 'F') erro = 5;
28        if(!this.getNumeroCartao().equals("2222442224"))
erro = 6;
29        if(this.getDia() != 21) erro = 7;
30        if(this.getMes() != 2) erro = 8;
31        if(this.getAno() != 2002) erro = 9;
32        if(this.getCodigoProduto() != 1) erro = 10;
33        if(this.getQtd() != 2) erro = 11;
34        if(this.getPreco() != 5) erro = 12;
35
36        if(erro == 0){
37            System.out.println("Não possui erros!");
38        }else{
39            System.out.println("Erro número: " + erro);
40        }// End else
```

```
41     }// End test()
42
43     @Override
44     public String getNome() { return "Ana Carla"; }
45     @Override
46     public String getNumeroCartao() { return "2222442224";
47     }
48     @Override
49     public String getCpf() { return "344.553.245-43"; }
50     @Override
51     public int getDia() { return 21; }
52     @Override
53     public int getMes() { return 2; }
54     @Override
55     public int getAno() { return 2002; }
56     @Override
57     public char getSexo() { return 'F'; }
58     @Override
59     public int getIdade() { return 17; }
60     @Override
61     public int getCodigoProduto(){ return 1; }
62     @Override
63     public int getQtd() { return 2; }
64     @Override
65     public double getPreco() { return 5; }
66 }// End VendaDriver
```

```
1 import java.util.Date;
2
3 public interface StubComprador{
4     public String getNome();
5     public String getNumeroCartao();
6     public String getCpf();
7     public int getDia();
8     public int getMes();
9     public int getAno();
10    public char getSexo();
11    public int getIdade();
12 }// End StubComprador
13
14
15
16
17
18
```

```
1 public interface StubProdutoVendido{
2     public int getCodigoProduto();
3     public int getQtd();
4     public double getPreco();
5 }// End StubProdutoVendido
6
7
8
9
10
11
```

4 IMPLEMENTE OS TESTES UTILIZANDO JUNIT

```
1  import org.junit.Test;
2
3  import java.util.ArrayList;
4  import java.util.Calendar;
5  import java.util.List;
6
7  import static org.junit.Assert.*;
8
9  public class VendaTest{
10     @Test
11     public void testGetDia(){
12         System.out.println("Teste GetDia");
13
14         List<ProdutoVendido> produtoVendidoList = new
15         ArrayList<>();
16
17         Comprador comprador = new Comprador(
18             "Ana Carla", "344.553.245-43",
19             21,2,2002,
20             'F',"2222442224");
21
22         Calendar calendar = Calendar.getInstance();
23         int hora = calendar.get(Calendar.HOUR);
24         int dia = calendar.get(Calendar.DATE);
25
26         Venda venda = new Venda(hora,dia,
27         produtoVendidoList,comprador);
28
29         assertEquals(dia,venda.getDiaVendido());
30     } // End testGetDia()
31
32     @Test
33     public void testGetHora(){
34         System.out.println("Teste GetHora");
35
36         List<ProdutoVendido> produtoVendidoList = new
37         ArrayList<>();
38
39         Comprador comprador = new Comprador(
40             "Ana Carla", "344.553.245-43",
41             21,2,2002,
42             'F',"2222442224");
```

```
43         Calendar calendar = Calendar.getInstance();
44         int hora = calendar.get(Calendar.HOUR);
45         int dia = calendar.get(Calendar.DATE);
46
47         Venda venda = new Venda(hora,dia,
    produtoVendidoList,comprador);
48
49         assertEquals(hora,venda.getHoraVendida());
50     }// End testGetHora()
51
52     @Test
53     public void testVendaAutorizada(){
54         System.out.println("Teste VendaAutorizada");
55
56         List<ProdutoVendido> produtoVendidoList = new
    ArrayList<>();
57         produtoVendidoList.add(new ProdutoVendido(1,2,5));
58
59         Comprador comprador = new Comprador(
60             "Ana Carla", "344.553.245-43",
61             21,2,2002,
62             'F',"2222442224");
63
64         Calendar calendar = Calendar.getInstance();
65         int hora = calendar.get(Calendar.HOUR);
66         int dia = calendar.get(Calendar.DATE);
67         int ano = calendar.get(Calendar.YEAR);
68         boolean resposta = false;
69         Venda venda = new Venda(hora,dia,
    produtoVendidoList,comprador);
70
71         assertEquals(venda.vendaAutorizada((ano-2002)),
    resposta);
72     }// End testVendaAutorizada()
73 }// End VendaTest
```

```
1 import org.junit.Test;
2
3 import static org.junit.Assert.*;
4
5 public class CompradorTest{
6     @Test
7     public void testGetNome(){
8         System.out.println("Teste GetNome");
9
10        Comprador comprador = new Comprador(
11            "Ana Carla", "344.553.245-43",
12            21,2,2002,
13            'F',"2222442224");
14
15        String nome = "Ana Clara";
16        assertEquals(nome,comprador.getNome());
17    }// End testGetNome()
18
19    @Test
20    public void testGetCpf(){
21        System.out.println("Teste GetCpf");
22
23        Comprador comprador = new Comprador(
24            "Ana Carla", "344.553.245-43",
25            21,2,2002,
26            'F',"2222442224");
27
28        String cpf = "344.553.245-43";
29        assertEquals(cpf,comprador.getCpf());
30    }// End testGetCpf()
31
32    @Test
33    public void testGetDia(){
34        System.out.println("Teste GetDia");
35
36        Comprador comprador = new Comprador(
37            "Ana Carla", "344.553.245-43",
38            21,2,2002,
39            'F',"2222442224");
40
41        int dia = 21;
42        assertEquals(dia,comprador.getDia());
43    }// End testGetDia()
44
45    @Test
```



```
46     public void testGetMes(){
47         System.out.println("Teste GetMes");
48
49         Comprador comprador = new Comprador(
50             "Ana Carla", "344.553.245-43",
51             21,2,2002,
52             'F', "2222442224");
53
54         int mes = 2;
55
56         assertEquals(mes,comprador.getMes());
57     }// End testGetMes()
58
59     @Test
60     public void testGetAno(){
61         System.out.println("Teste GetAno");
62
63         Comprador comprador = new Comprador(
64             "Ana Carla", "344.553.245-43",
65             21,2,2002,
66             'F', "2222442224");
67
68         int ano = 2002;
69
70         assertEquals(ano,comprador.getAno());
71     }// End testGetAno()
72
73     @Test
74     public void testGetSexo(){
75         System.out.println("Teste GetSexo");
76
77         Comprador comprador = new Comprador(
78             "Ana Carla", "344.553.245-43",
79             21,2,2002,
80             'F', "2222442224");
81
82         char sexo = 'F';
83
84         assertEquals(sexo,comprador.getSexo());
85     }// End testGetSexo()
86
87     @Test
88     public void testNumeroCartao(){
89         System.out.println("Teste GetNumeroCartao");
90
```

```
91         Comprador comprador = new Comprador(  
92             "Ana Carla", "344.553.245-43",  
93             21,2,2002,  
94             'F',"2222442224");  
95  
96         String numeroCartao = "2222442224";  
97  
98         assertEquals(numeroCartao,comprador.  
99             getNumeroCartao());  
100     }// End testNumeroCartao()  
100 }// End CompradorTest
```

```
1 import org.junit.Test;
2
3 import static org.junit.Assert.*;
4
5 public class ProdutoVendidoTest{
6     @Test
7     public void testGetCodigoProduto(){
8         System.out.println("Teste GetCodigoProduto");
9
10        ProdutoVendido produtoVendido = new ProdutoVendido
(1,2,5);
11
12        int codigo = 1;
13
14        assertEquals(codigo,produtoVendido.
getCodigoProduto());
15    }// End testCodigoProduto()
16
17    @Test
18    public void testGetQtd(){
19        System.out.println("Teste GetQtd");
20
21        ProdutoVendido produtoVendido = new ProdutoVendido
(1,2,5);
22
23        int qtd = 2;
24        assertEquals(qtd,produtoVendido.getQtd());
25    }// End testGetQtd()
26
27    @Test
28    public void testGetPreco(){
29        System.out.println("Teste GetPreco");
30
31        ProdutoVendido produtoVendido = new ProdutoVendido
(1,2,5);
32
33        double preco = 5;
34
35        assertEquals(preco,produtoVendido.getPreco());
36    }// End testGetPreco()
37 }// End ProdutoVendidoTest
```

**5 FAÇA UM PROJETO DO QUE VOCÊ VAI TESTAR (TABELA: DADOS X RE X RO).
UTILIZE A TÉCNICA DE PARTIÇÕES PARA DEFINIR O QUE TESTAR**

6 EXECUTE O TESTE, PREENCHA O CAMPO RO E AVALIE O RESULTADO

Classe de equivalência		
Input	Válidas	Inválidas
Idade	Inteiro positivo menor do que 150	Inteiro negativo maior do que 151 outros valores
Preço	Decimal positivo diferente de 0	Decimal negativo ou igual a 0 ou outros valores
Código	Inteiro não negativo	Inteiro negativo ou outros valores
CPF	3 Séries de 3 dígitos seguidos de dois dígitos	Outros valores que não são sequência de 3 dígitos seguidos

Idade	Resultado esperado	Resultado Obtido
0	Erro!	Erro!
155	Erro!	Sucesso!
100	Sucesso!	Sucesso!
-222	Erro!	Erro!
13	Erro!	Erro!
18	Sucesso!	Sucesso!
14	Erro!	Erro!
17	Erro!	Sucesso!

Preço	Resultado esperado	Resultado Obtido
0	Erro!	Sucesso!
13	Sucesso!	Sucesso!
-10	Erro!	Sucesso!
100	Sucesso!	Sucesso!
-22	Erro!	Sucesso!
a	Erro!	Sucesso!
Vazio	Erro!	Sucesso!
Teste	Erro!	Sucesso!
a	Erro!	Sucesso!

Run: CompradorTest x

Tests failed: 1, passed: 6 of 7 tests - 24 ms

Test	Duration
testNumeroCartao	7 ms
testGetNome	15 ms
testGetSexo	1 ms
testGetAno	0 ms
testGetCpf	1 ms
testGetDia	0 ms
testGetMes	0 ms

Teste GetNumeroCartao

Teste GetNome

org.junit.ComparisonFailure:
Expected :Ana Clara
Actual :Ana Carla
[Click to see difference](#)

<2 internal calls>
at CompradorTest.testGetNome(CompradorTest.java:16) <22 internal calls>

Teste GetSexo
Teste GetAno
Teste GetCpf
Teste GetDia
Teste GetMes

Process finished with exit code -1

Run: VendaTest x

Tests failed: 1, passed: 2 of 3 tests - 44 ms

Test	Duration
testGetHora	21 ms
testVendaAutorizada	23 ms
testGetDia	0 ms

Teste GetHora

Teste VendaAutorizada

java.lang.AssertionError:
Expected :true
Actual :false
[Click to see difference](#)

<1 internal call>
at org.junit.Assert.failNotEquals(Assert.java:834) <2 internal calls>
at VendaTest.testVendaAutorizada(VendaTest.java:71) <22 internal calls>

Teste GetDia

Process finished with exit code -1

Run: ProdutoVendidoTest x

Tests passed: 3 of 3 tests - 9 ms

Test	Duration
testGetPreco	9 ms
testGetQtd	0 ms
testGetCodigoProduto	0 ms

Teste GetPreco

Teste GetQtd

Teste GetCodigoProduto

Process finished with exit code 0

Código	Resultado esperado	Resultado Obtido
-1	Erro!	Sucesso!
-333	Erro!	Sucesso!
0	Sucesso!	Sucesso!
3323	Sucesso!	Sucesso!
43	Sucesso!	Sucesso!
342342342342	Erro!	Erro!

CPF	Resultado esperado	Resultado Obtido
202.202.111-12	Sucesso!	Sucesso!
Vazio	Erro!	Sucesso!
a	Erro!	Sucesso!
CPF	Erro!	Sucesso!
-2	Erro!	Erro!
3443	Erro!	Erro!
0	Erro!	Erro!
222.332.232-13	Sucesso!	Sucesso!
MG-232.332.232-13	Erro!	Sucesso!
