



# T2 – Tarefa sobre “Teste de Unidade”

**Acadêmicos:** Gustavo Santos, Henrique Baptista e Pedro Carlo Iglesias

**Disciplina:** Verificação e Validação de Software



# TESTES DE VALOR LIMITE

Variável			Condição		Tipo		T1		T2		T3		T4		T5	
Aditivo com Excesso			x <= 250		on		250									
Tanque Aditivo = 250					off				125							
			x >= 0		on						0					
					off								250			
			typical		in										125	
Aditivo sem Excesso			x >= 251		on		251									
Tanque Aditivo = 250					off				500							
			typical		in						500					
Aditivo Inválido			x <= -1		on		-1									
Tanque Aditivo = 0					off				-500							
Gasolina sem Excesso	x <= 5000	on	5000						Álcool sem Excesso	x <= 1250	on	1250				
Tanque Gasolina = 5000		off		2500					Tanque Álcool = 1250 x2		off		625			
	x >= 0	on			0					x >= 0	on			0		
		off				5000	2500				off				1250	
	typical	in								typical	in					625
Gasolina com Excesso	x >= 5001	on	5001						Álcool com Excesso	x > 1250	on	1251				
Tanque Gasolina = 5000		off		10000					Tanque Álcool = 1250 x2		off		2500			
	typical	in			10000				Álcool Inválido	x < 0	on	-1				
Gasolina Inválida	x <= -1	on	-1						Tanque Álcool = 0		off		-2500			
Tanque Gasolina = 0		off		-5000												
Inicialização Normal		Tanques >= 50%	on	{500, 10000, 1250, 1250}					Encomenda Aceita	x >= 0	on	0				
			off			{250, 5000, 625, 625}			Tanques Chelos		off		500			
Inicialização Sobreaviso		Tanques <= 49%	on	{249, 5000, 625, 625}						x <= 10000	on			10000		
			off			{250, 5000, 624, 625}					off				500	
		Tanques >= 25%	on	{125, 2500, 313, 313}						typical	in					500
			off			{250, 4999, 625, 625}			Encomenda Inválida	x <= -1	on	-1				
Inicialização Emergência		Tanques < 25%	on	{124, 2500, 313, 313}							off	-1000				
			off			{125, 2500, 311, 313}			Encomenda Recusada Situação							
		Tanques >= 0%	on	{0, 0, 0, 0}					Tanques < 25%	x <= 124	on	124				
			off			{125, 2500, 313, 311}			Encomenda Recusada Combustível	x >= 600	on	600				
			off						Tanques < 600		off		1000			

# CASOS DE TESTE: INICIALIZAÇÃO DO C.D

```
@ParameterizedTest
@MethodSource("casosNormalICD")
public void inicializaCDNormal(int[] casosNormal)
{
    cd = new CentroDistribuicao(casosNormal[0],casosNormal[1],casosNormal[2],casosNormal[3]);
    assertEquals("NORMAL",cd.gettSituacao().toString());
}

@ParameterizedTest
@MethodSource("casosSobreavisoICD")
public void inicializaCDSobreaviso(int[] casosSobreaviso)
{
    cd = new CentroDistribuicao(casosSobreaviso[0],casosSobreaviso[1],casosSobreaviso[2],casosSobreaviso[3]);
    assertEquals("SOBREAVISO",cd.gettSituacao().toString());
}
```

# CASOS DE TESTE: INICIALIZAÇÃO DO C.D

```
@ParameterizedTest
@MethodSource("casosEmergenciaICD")
public void inicializaCDEmergencia(int[] casosEmergencia)
{
    cd = new CentroDistribuicao(casosEmergencia[0], casosEmergencia[1] , casosEmergencia[2], casosEmergencia[3]);
    assertEquals("EMERGENCIA",cd.gettSituacao().toString());
}

@ParameterizedTest
@MethodSource("casosInvalidosICD")
public void inicializaCDInvalida(int[] casosInvalidos)
{
    assertThrows(InvalidParameterException.class, () -> new CentroDistribuicao(casosInvalidos[0],
    casosInvalidos[1] , casosInvalidos[2], casosInvalidos[3]) );
}
```

# CASOS DE TESTE: INICIALIZAÇÃO DO C.D

```
public static int[][] casosNormalICD() {  
    return new int[][] { { 500 , 10000, 1250, 1250 }, { 250, 5000, 625, 625 } };  
}  
  
public static int[][] casosSobreavisoICD() {...}  
  
public static int[][] casosEmergenciaICD() {...}  
  
public static int[][] casosInvalidosICD() {...}
```

## CASOS DE TESTE: RECEBE ADITIVO

```
@ParameterizedTest
@ValueSource( ints = {0, 125, 250} )
public void recebeAditivoSemExcesso(int aditivo){
    cd = new CentroDistribuicao(250, 0, 0, 0);
    assertEquals( aditivo, cd.recebeAditivo(aditivo) );
}

@ParameterizedTest
@ValueSource( ints = {251, 500})
public void recebeAditivoComExcesso(int aditivo){
    cd = new CentroDistribuicao(250, 0, 0, 0);
    int aditivoArmazenado = cd.getAditivo();
    assertEquals( CentroDistribuicao.MAX_ADITIVO - aditivoArmazenado , cd.recebeAditivo(aditivo) );
}

@ParameterizedTest
@ValueSource( ints = {-1, -500})
public void recebeAditivoInvalido(int aditivo){
    cd = new CentroDistribuicao(0, 0, 0, 0);
    assertEquals( -1 , cd.recebeAditivo(aditivo) );
}
```

# CASOS DE TESTE: RECEBE GASOLINA

```
@ParameterizedTest
@ValueSource( ints = {0, 2500, 5000} )
public void recebeGasolinaSemExcesso(int gasolina){
    cd = new CentroDistribuicao(0, 5000, 0, 0);
    assertEquals( gasolina, cd.recebeGasolina(gasolina) );
}

@ParameterizedTest
@ValueSource( ints = {5001, 10000})
public void recebeGasolinaComExcesso(int gasolina){
    cd = new CentroDistribuicao(0, 5000, 0, 0);
    int gasolinaArmazenada = cd.gettGasolina();
    assertEquals( CentroDistribuicao.MAX_GASOLINA - gasolinaArmazenada , cd.recebeGasolina(gasolina) );
}

@ParameterizedTest
@ValueSource( ints = {-1, -5000})
public void recebeGasolinaInvalido(int gasolina){
    cd = new CentroDistribuicao(0, 0, 0, 0);
    assertEquals( -1 , cd.recebeGasolina(gasolina) );
}
```

# CASOS DE TESTE: RECEBE ÁLCOOL

```
@ParameterizedTest
@ValueSource( ints = {0, 625, 1250} )
public void recebeAlcoolSemExcesso(int alcool){
    cd = new CentroDistribuicao(0, 0, 625, 625);
    assertEquals( alcool, cd.recebeAlcool(alcool) );
}

@ParameterizedTest
@ValueSource( ints = {1251, 2500})
public void recebeAlcoolComExcesso(int alcool){
    cd = new CentroDistribuicao(0, 0, 625, 625);
    int alcoolArmazenado = cd.getAlcool1()*2;
    assertEquals( CentroDistribuicao.MAX_ALCOOL - alcoolArmazenado , cd.recebeAlcool(alcool) );
}

@ParameterizedTest
@ValueSource( ints = {-1, -2500})
public void recebeAlcoolInvalido(int alcool){
    cd = new CentroDistribuicao(0, 0, 0, 0);
    assertEquals( -1 , cd.recebeAlcool(alcool) );
}
```



# CASOS DE TESTE: ENCOMENDAS DOS POSTOS

```
@ParameterizedTest
@ValueSource( ints = {500, 10000})
public void encomendaAceitado(int combustivel){
    cd = new CentroDistribuicao(500, 10000, 1250, 1250);
    int[] tanquesAtual = cd.encomendaCombustivel(combustivel, TIPOPOSTO.COMUM);
    assertEquals(new int[]{cd.gettAditivo(), cd.gettGasolina(), cd.gettAlcool1(), cd.gettAlcool2()},
        tanquesAtual);
}

@ParameterizedTest
@ValueSource(ints = {-1, -1000})
public void encomendaRecusadoParametro(int combustivel){
    cd = new CentroDistribuicao(0, 0, 0, 0);
    assertEquals(new int[]{-7}, cd.encomendaCombustivel(combustivel, TIPOPOSTO.COMUM));
    assertEquals(new int[]{-7}, cd.encomendaCombustivel(combustivel, TIPOPOSTO.ESTRATEGICO));
}
```

# CASOS DE TESTE: ENCOMENDAS DOS POSTOS

```
@ParameterizedTest
@ValueSource(ints = {1, 10, 100, 1000})
public void encomendaRecusadoSituacao(int combustivel){
    cd = new CentroDistribuicao(120, 2400, 300, 300);
    assertEquals(new int[]{-14}, cd.encomendaCombustivel(combustivel, TIPOPOSTO.COMUM));
}

@ParameterizedTest
@ValueSource(ints = {600, 1000})
public void encomendaRecusadoCombustivel(int combustivel){
    cd = new CentroDistribuicao(500, 200, 1250, 1250);
    assertEquals(new int[]{-21}, cd.encomendaCombustivel(combustivel, TIPOPOSTO.ESTRATEGICO));
}
```

# CODE COVERAGE

## CentroDistribuicao

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
● <a href="#">encomendaCombustivel(int, CentroDistribuicao.TIPOPOSTO)</a>		80%		72%	4	10	4	17	0	1
● <a href="#">defineSituacao()</a>		100%		100%	0	7	0	7	0	1
● <a href="#">consegueCombustivel(int, CentroDistribuicao.TIPOPOSTO)</a>		100%		66%	2	4	0	8	0	1
● <a href="#">recebeAlcool(int)</a>		100%		100%	0	3	0	11	0	1
● <a href="#">CentroDistribuicao(int, int, int, int)</a>		100%		100%	0	9	0	9	0	1
● <a href="#">recebeAditivo(int)</a>		100%		100%	0	3	0	10	0	1
● <a href="#">recebeGasolina(int)</a>		100%		100%	0	3	0	10	0	1
● <a href="#">retiraCombustivel(int, int, int)</a>		100%		n/a	0	1	0	5	0	1
● <a href="#">getAditivo()</a>		100%		n/a	0	1	0	1	0	1
● <a href="#">getGasolina()</a>		100%		n/a	0	1	0	1	0	1
● <a href="#">getAlcool1()</a>		100%		n/a	0	1	0	1	0	1
● <a href="#">getAlcool2()</a>		100%		n/a	0	1	0	1	0	1
● <a href="#">getSituacao()</a>		100%		n/a	0	1	0	1	0	1
Total	20 of 387	94%	7 of 64	89%	6	45	4	82	0	13

# CASOS DE TESTE ADICIONAIS: CODE-COVERAGE

```
@Test
public void encomendaRecusadoSituacaoSobreavisoQuantidadeMenor(){
    cd = new CentroDistribuicao(249, 10000, 1250, 1250);
    assertEquals(new int[]{-14}, cd.encomendaCombustivel(1, TIPOPOSTO.COMUM));
}

@Test
public void encomendaRecusadoSituacaoEmergenciaQuantidadeMenor(){
    cd = new CentroDistribuicao(10, 10000, 1250, 1250);
    assertEquals(new int[]{-14}, cd.encomendaCombustivel(1, TIPOPOSTO.ESTRATEGICO));
}

@Test
public void encomendaAceitadoSituacaoSobreavisoQuantidadeMaiorPostoComum(){
    cd = new CentroDistribuicao(249, 10000, 1250, 1250);
    assertEquals(new int[]{246,9965,1243,1243}, cd.encomendaCombustivel(100, TIPOPOSTO.COMUM));
}
```

# CASOS DE TESTE ADICIONAIS: CODE-COVERAGE

```
@Test
public void encomendaAceitadoSituacaoEmergenciaQuantidadeMaiorPostoComum(){
    cd = new CentroDistribuicao(249, 10000, 1250, 1250);
    assertEquals(new int[]{244,9930,1237,1237}, cd.encomendaCombustivel(100, TIPOPOSTO.ESTRATEGICO));
}

@ParameterizedTest
@MethodSource("casosTanquesVazios")
public void entregaRecusadaTanquesVazios(int[] dados){
    cd = new CentroDistribuicao(dados[0], dados[1], dados[2], dados[3]);
    assertEquals(new int[]{-21}, cd.encomendaCombustivel(1000, TIPOPOSTO.ESTRATEGICO));
}

public static int[][] casosTanquesVazios() {
    return new int[][] { { 0, 10000, 1250, 1250 }, { 500, 0, 1250, 1250 }, { 500, 10000, 0, 100 }, { 500, 10000,
100, 0 } };
}
```

# CODE COVERAGE

## CentroDistribuicao

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
● <a href="#">encomendaCombustivel(int, CentroDistribuicao.TIPOPOSTO)</a>	<div></div>	100%	<div></div>	100%	0	10	0	17	0	1
● <a href="#">defineSituacao()</a>	<div></div>	100%	<div></div>	100%	0	7	0	7	0	1
● <a href="#">consegueCombustivel(int, CentroDistribuicao.TIPOPOSTO)</a>	<div></div>	100%	<div></div>	100%	0	4	0	8	0	1
● <a href="#">recebeAlcool(int)</a>	<div></div>	100%	<div></div>	100%	0	3	0	11	0	1
● <a href="#">CentroDistribuicao(int, int, int, int)</a>	<div></div>	100%	<div></div>	100%	0	9	0	9	0	1
● <a href="#">recebeAditivo(int)</a>	<div></div>	100%	<div></div>	100%	0	3	0	10	0	1
● <a href="#">recebeGasolina(int)</a>	<div></div>	100%	<div></div>	100%	0	3	0	10	0	1
● <a href="#">retiraCombustivel(int, int, int)</a>	<div></div>	100%		n/a	0	1	0	5	0	1
● <a href="#">getAditivo()</a>	<div></div>	100%		n/a	0	1	0	1	0	1
● <a href="#">getGasolina()</a>	<div></div>	100%		n/a	0	1	0	1	0	1
● <a href="#">getAlcool1()</a>	<div></div>	100%		n/a	0	1	0	1	0	1
● <a href="#">getAlcool2()</a>	<div></div>	100%		n/a	0	1	0	1	0	1
● <a href="#">getSituacao()</a>	<div></div>	100%		n/a	0	1	0	1	0	1
Total	0 of 387	100%	0 of 64	100%	0	45	0	82	0	13