

Engenharia de Software 2

Trabalho pratico 2

Módulo de Custo de Envio / Transações /Expedição

<LEI_ESII2021_TP2_GRUP011>

Test Case outline

Version <1.4>
<20/01/2022>

Realizado por:
8180378-Hugo Silva
8200306-Fábio Costa

Histórico de Versões

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	HUGO	<2/1/2022>	<Hugo, Fabio>	<2/1/2022>	Aprovação estrutura
1.1	HUGO	<4/1/2022>	<Hugo,Fabio>	<4/1/2022>	Detalhes TestStatisticsE xporter
1.2	HUGO	<6/1/2022>	<Hugo, Fabio>	<6/1/2022>	Detalhes OrderMetricSta tistics
1.3	Fábio	<14/1/2022>	<Hugo, Fabio>	<18/1/2022>	Detalhes Class DistanceTable,
1.4	HUGO	<20/1/2022>	<Hugo, Fabio>	<20/1/2022>	Correções e versão final

Tabela de Conteúdos

1. Introdução.....	8
1.1. Identificador do documento	8
1.2. Âmbito	8
1.3. References	8
1.4 Glossário	8
2. Detalhes <i>Class Orders</i>.....	9
testRegisterOrdersInLedgerBVA01	9
testRegisterOrdersInLedgerBVA02	9
testRegisterOrdersInLedgerBVA03	10
testRegisterOrdersInLedgerECP01	11
testRegisterOrdersInLedgerECP02	12
<i>Detalhes TestStatistExporter</i>	13
testCreateFileBVA01	13
testCreateFileBVA02	13
testCreateFileECP01.....	14
testWriteToFileBVA01	14
testWriteToFileBVA02	15
testWriteToFileBVA03	15
TestWriteToFileECP01	16
<i>Detalhes OrderMetricStatistics</i>.....	17
testAverageNumberProductsTransactionBVA01	17
testAverageNumberProductsTransactionECP01	18
testAverageValueTransactionsBVA01.....	19
testAverageValueTransactionsECP01	20
testAverageValueSalesPurchasesDistrictBVA01	21
testAverageValueSalesPurchasesDistrictBVA02	21
testAverageValueSalesPurchasesDistrictECP01	22
testAverageValueSalesPurchasesDistrictECP02	22
testAverageValueSalesPurchasesDistrictECP03	24
testNumberOrdersByDistrictBVA01.....	25
testNumberOrdersByDistrictECP01	26
testStandardDeviationNumberProductTransactionsBVA01	27
testStandardDeviationNumberProductTransactionsECP01	28

testStandardDeviationTransactionsECP01.....	29
Tabela Auxiliar X.....	30
testStandardDeviationTransactionsBVA01.....	31
Detalhes Product	32
testBasicProductBVA08	32
testBasicProductBVA06	33
testBasicProductBVA09	33
testBasicProductBVA10	34
Detalhes Truck.....	35
TestTruckBVA01	35
TestTruckBVA02	35
TestTruckECP01	36
TestAddAllOrderInTruckBVA01	36
TestAddAllOrderInTruckBVA02	37
TestAddAllOrderInTruckBVA03	37
testAddAllOrderInTruckProductInTruckECP01.....	38
testAddProductInTruckBVA01	38
testAddProductInTruckBVA02	39
testAddProductInTruckBVA03	39
testAddProductInTruckBVA04	40
testAddProductInTruckBVA05	40
testAddProductInTruckECP01	41
testGetProductContainerTruckBVA01	41
testGetProductContainerTruckECP01	42
testGetProductContainerTruckECP02	42
testGetProductListProductsInsideTruckBVA01	43
testGetProductListProductsInsideTruckBVA02	43
TestGetProductListProductsInsideTruckECP01	44
Detalhes ProductsInsideTruck.....	45
testProductsInsideTruckBVA01	45
testProductsInsideTruckBVA02	45
testProductsInsideTruckBVA03	46
testProductsInsideTruckBVA04	46
testProductsInsideTruckBVA05	47
testProductsInsideTruckBVA06	47

testProductsInsideTruckECP01	48
Detalhes DistrictTrucks	49
testGetTruckByIndexBVA01.....	49
testGetTruckByIndexBVA02.....	49
testGetTruckByIndexECP01	49
testCreateTruckBVA01.....	50
testCreateTruckECP01	51
testCreateTruckECP02	52
testDistrictTrucksBVA01	52
testDistrictTrucksBVA02	53
testDistrictTrucksBVA03	53
testDistrictTrucksECP01	54
Detalhes GroupAllTrucks	55
testAddDistrictTrucksBVA01.....	55
TestAddDistrictTrucksECP01	56
Detalhes TestExpeditionMethods	57
testGetArrayOrdersByDistrictBVA01	57
testGetArrayOrdersByDistrictBVA02	58
testGetArrayOrdersByDistrictBVA03	58
testGetArrayOrdersByDistrictECP01	59
testGroupOrdersByTrucksBVA01.....	59
testGroupOrdersByTrucksECP01.....	60
testGroupOrdersByTrucksECP02.....	63
Detalhes Product	65
ProductBVA01	65
ProductBVA02	65
ProductBVA03	66
ProductBVA04	67
ProductBVA05	68
ProductBVA06	69
ProductBVA07	70
ProductBVA08	71
ProductBVA09	72
ProductBVA10	73
ProductECP01	74

Detalhes Class Order	75
OrderBVA01.....	75
OrderBVA02.....	76
OrderBVA03.....	77
OrderBVA04.....	78
OrderECP01	78
getProductBVA01	79
getProductECP01	79
getProductECP02	80
addProductBVA01	80
addProductECP01	81
addProductECP02	81
shippingCostCalculationECP01.....	82
shippingCostCalculationECP02.....	83
Detalhes Orders.....	84
getOrderBVA01	84
GetOrderECP01.....	84
GetOrderECP02.....	85
addOrderBVA01.....	85
addOrderECP01	86
getOrderShippingCostBVA01.....	86
getOrderShippingCostECP01.....	87
getOrderShippingCostECP02.....	88
testGetOrderIndexBVA01	89
testGetOrderIndexBVA02	89
testGetOrderIndexECP01	90
Detalhes Class DistanceTable	91
getDistrictDistanceBVA01.....	91
getDistrictDistanceECP01	91
getDistrictDistanceECP02	92
addDistrictDistanceBVA01.....	92
addDistrictDistanceBVA01.....	93
getDistanceValueFromShippingBVA01	93
getDistanceValueFromShippingBVA02	94
getDistanceValueFromShippingECP01	95

getDistanceValueFromShippingECP02	96
Detalhes Class DistrictDistance	97
DistrictDistanceBVA01	97
DistrictDistanceBVA02	97
DistrictDistanceECP01	98
getDistanceBVA01	98
getDistanceECP01	99
DistrictDistanceECP01	99
getDistanceECP02	100
addDistanceBVA01	100
addDistanceECP01	101
addDistanceECP02	101
Detalhes Class Distance	102
DistanceBVA01	102
DistanceBVA02	102
DistanceECP01	103
3. Summary Tests	105

1. Introdução

1.1. Identificador do documento

TestCaseOutlineTrabalhoB

1.2. Âmbito

Este documento foi realizado para o trabalho 2 de Engenharia de Software II que tem como objetivo realização de atividades de garantia de qualidade de Software "SQA", em concreto a atividade realização de testes caixa preta (BlackBox Tests).

Para a execução dos testes foram usadas duas técnicas, a técnica "ECP" e técnica "BVA".

No decorrer do documento serão apresentados os procedimentos usados, os resultados obtidos de cada Test-Case bem como a data e a hora a que foram realizados.

O objetivo principal dos testes de software realizados neste projeto é verificar se os métodos têm o comportamento esperado.

1.3. References

IEEE Std 829™-2008

1.4 Glossário

BVA – Boundary Value Analysis

ECP – Equivalence Class Partitioning

TC – Teste Case

BVA - Esta Técnica foca-se na análise dos limites do domínio, ou seja, limites superiores e inferiores dos valores fronteira. Além disso, focam-se também em testar valores especiais (por exemplo: 0, *NULL*).

ECP - Esta Técnica foca-se em dividir os dados de input de uma classe de software em conjuntos de dados equivalentes, esses conjuntos podem ser usados para criar casos de teste.

TC-Grupo de condições utilizadas para testar software.

2. Detalhes Class Orders

testRegisterOrdersInLedgerBVA01

Test Case ID	TC# testRegisterOrdersInLedgerBVA01	Dependências	
--------------	--	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
8 janeiro	16:13h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado 1 objeto do tipo Order				
Procedimentos				
-Invocar método registerOrdersInLedger				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestado
Registrar encomendas no <i>ledger</i>	Não tem	Não tem	0	PASS

testRegisterOrdersInLedgerBVA02

Test Case ID	TC# testRegisterOrdersInLedgerBVA01	Dependências	
--------------	--	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
8 janeiro	16:13h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado 1 objeto do tipo Orders , um do tipo Entitye 4 objetos do tipo Product				
Procedimentos				
-Adicionar 1 token à entidade; - Criar uma order com a entidade como <i>sender</i> ; - Adicionar produtos à order; -Adicionar order à lista de orders para registo; -Invocar método registerOrdensInLedger;				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestado

Registrar encomendas no <i>ledger</i>	Não tem	Tabela auxiliar 7	2	PASS
---------------------------------------	---------	-------------------	---	------

testRegisterOrdersInLedgerBVA03

Test Case ID	TC# testRegisterOrdersInLedgerBVA01	Dependências	
--------------	--	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
8 janeiro	16:13h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado 1 objeto do tipo Orders , um do tipo Entitye 4 objetos do tipo Product				
Procedimentos				
- Criar uma order com a entidade com 0 tokens; - Adicionar produtos à order; -Adicionar order à lista de Orders para registo; -Invocar método registerOrdensInLedger;				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestado
Registrar encomendas no <i>ledger</i>	Não tem	Tabela auxiliar 7	2	PASS

Tabela auxiliar 7

```

BasicOrders basicOrders = new BasicOrders();
Entity entitySender = new BasicEntity("Entity-Sender", District.AVEIRO);
entitySender.addTokens(1); apenas usado no testRegisterOrdersInLedgerBVA03
BasicOrder basicOrder = new BasicOrder("123_222", "2021-2-21", entitySender, receiverEntity);
basicOrder.addProduct(product);
basicOrder.addProduct(product2);
basicOrder.addProduct(product3);
basicOrder.addProduct(product4);
basicOrders.addOrder(basicOrder);

```

testRegisterOrdersInLedgerECP01

Test Case ID	TC# testRegisterOrdersInLedgerBVA01	Dependências	
--------------	--	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
8 janeiro	16:13h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado 1 objeto do tipo Orders , um do tipo Entitye 4 objetos do tipo Product				
Procedimentos				
-Adicionar 1 token à entidade; - Criar uma order com a entidade com 2 tokens; - Adicionar produtos à order; -Adicionar order à lista de Orders para registo; -Invocar método registerOrdensInLedger;				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestado
Registar encomendas no ledger	Não tem	Tabela auxiliar 8	2	PASS

Tabela auxiliar 8

```

BasicOrders basicOrders = new BasicOrders();
Entity entitySender = new BasicEntity("Entity-Sender", District.AVEIRO);
entitySender.addTokens(2);
BasicOrder basicOrder = new BasicOrder("123_222", "2021-2-21", entitySender, receiverEntity);
basicOrder.addProduct(product);
basicOrder.addProduct(product2);
basicOrder.addProduct(product3);
basicOrder.addProduct(product4);
basicOrders.addOrder(basicOrder);

```

testRegisterOrdersInLedgerECP02

Test Case ID	TC# testRegisterOrdersInLedgerBVA 02	Dependências	
--------------	--	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
8 janeiro	16:13h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado 1 objeto do tipo Orders , um do tipo Entitye 4 objetos do tipo Product				
Procedimentos				
-Adicionar 1 token à entidade; - Criar uma order com a entidade com 2 tokens; - Adicionar produtos à order; -Adicionar order à lista de Orders para registo; -Invocar método registerOrdensInLedger; -Invocar novamente o método e verificar retorno;				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestado
Registar encomendas no <i>ledger</i>	Não tem	Tabela auxiliar 8	0	PASS

Detalhes TestStatistExporter

testCreateFileBVA01

Test Case ID	TC# testCreateFileBVA02	Dependências	
--------------	----------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
9 janeiro	00:36h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado 1 objeto do tipo StatisticExporter				
Procedimentos				
-Invocar método createFile.				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestado
Criar ficheiro	String path	null	IllegalArgumentException	PASS

testCreateFileBVA02

Test Case ID	TC# testCreateFileBVA02	Dependências	
--------------	----------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
9 janeiro	00:36h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado 1 objeto do tipo StatisticExporter				
Procedimentos				
-Invocar método createFile.				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestado
Criar ficheiro	String path	"" (empty)	IllegalArgumentException	PASS

testCreateFileECP01

Test Case ID	TC#	Dependências	
	testCreateFileECP01		

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
9 janeiro	00:36h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado 1 objeto do tipo StatisticExporter				
Procedimentos				
-Invocar método createFile.				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar ficheiro	String path	"filesJson/TestCreateFile/test.json"	true	PASS

testWriteToFileBVA01

Test Case ID	TC#	Dependência s	
	testWriteToFileBVA01		

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
9 janeiro	00:36h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado 1 objeto do tipo StatisticExporter				
Procedimentos				
-Invocar método writeFile.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar ficheiro	String, JsonObject	null, jsonObject	IllegalArgumentException	PASS

testWriteToFileBVA02

Test Case ID	TC# testWriteToFileBVA01	Dependências	
--------------	-----------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
9 janeiro	00:36h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado 1 objeto do tipo StatisticExporter				
Procedimentos				
-Invocar método writeFile.				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar ficheiro	String, JsonObject	""(empty), jsonObject	IllegalArgumentException	PASS

testWriteToFileBVA03

Test Case ID	TC# testWriteToFileBVA01	Dependências	
--------------	-----------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
9 janeiro	00:36h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado 1 objeto do tipo StatisticExporter				
Procedimentos				
-Invocar método writeFile.				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar ficheiro	String, JsonObject	("filesJson/TestCreateFile/test.json", null)	IllegalArgumentException	PASS

TestWriteToFileECP01

Test Case ID	TC# testWriteToFileBV A01	Dependências	
--------------	---------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
9 janeiro	00:36h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado 1 objeto do tipo StatisticExporter				
Procedimentos				
-Invocar método writeFile.				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar ficheiro	String, JsonObject	("filesJson/TestCreateFile/test.json", jsonObject)	true	PASS

Detalhes OrderMetricStatistics

testAverageNumberProductsTransactionBVA01

Test Case ID	TC# testAverageNumberProductsTransactionBVA01	Dependências	
---------------------	--	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	03:55h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado : Objeto do tipo Basic Organization				
Procedimentos				
-Invocar método averageNumberProductsTransaction				
Requisito/ Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter média de produtos por transação	Não tem	Não tem	AverageException	PASS

testAverageNumberProductsTransactionECP01

Test Case ID	TC# testAverageNumberProductsTransactionECP01	Dependências	
--------------	--	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	03:55h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado : Objeto do tipo Orders, 6 do tipo Product e 3 do tipo Order				
Procedimentos				
-Adicionar produtos à order; -Adicionar encomendas ao objeto do tipo Orders; -Registar encomendas no <i>ledger</i> ; -Invocar método <i>AverageNumberProductsTransaction</i> ;				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter média de produtos por transação	3 objetos to tipo <i>Order</i> contendo produtos	Número de produtos por transação: T1 = 30 T2 = 20 T3 = 5	18.33	PASS

testAverageValueTransactionsBVA01

Test Case ID	TC#	Dependências	
	testAverageNumberProductsTransactionBVA01		

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	03:55h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado : Objeto do tipo Orders				
Procedimentos				
-Invocar método averageValueTransactions;				
Requisito/ Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Un- tested
Obter valor médio das transações	Não tem		AverageException	PASS

testAverageValueTransactionsECP01

Test Case ID	TC# testAverageNumberProductsTransactionBVA01	Dependências	
--------------	--	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	03:55h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado : Objeto do tipo Basic Organization, 2 objetos do tipo BasicTransaction e 6 do tipo TransactionLine				
Procedimentos				
-Adicionar produtos à order; -Adicionar encomendas ao objeto do tipo Orders; -Registar encomendas no <i>ledger</i> ; -Invocar método <u>averageValueTransactions</u> ;				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter valor médio das transações	3 objetos do tipo Order	Valor das transações: T1 = 12 T2 = 6	9	PASS

testAverageValueSalesPurchasesDistrictBVA01

Test Case ID	TC#	Dependências	
	testAverageValueSalesPurchases District BVA01		

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
6 janeiro	22:35h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado : Objeto do tipo Orders				
Procedimentos				
-Invocar método <u>averageValueTransactions;</u>				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter média de compras e vendas por distrito	Não tem	Não tem	AverageException	PASS

testAverageValueSalesPurchasesDistrictBVA02

Test Case ID	TC#	Dependências	
	testAverageValuePurchases District BVA01		

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
6 janeiro	22:40h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado : Objeto do tipo Orders				
Procedimentos				
-Invocar método <u>averageValueTransactions;</u>				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter média de compras e vendas por distrito	Não tem	null	IllegalArgumentExcepcion	PASS

testAverageValueSalesPurchasesDistrictECP01

Test Case ID	TC# testAverageValuePurchasesDistrictECP01	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
5 janeiro	22:40h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado : 4 Entidades , 3 transações e 3 linhas de transações				
Procedimentos				
-Adicionar produtos à order; -Adicionar encomendas ao objeto do tipo Orders; -Registar encomendas no <i>ledger</i> ; (Repetir passo anterior e este 10000 vezes) -Invocar método <i>averageValueSalesPurchasesDistrict</i> ; - Comparar <i>map</i> criado com <i>map</i> retornado pelo método;				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter média de compras e vendas por distrito	Não tem	(Ver tabela auxiliar 6)	Map (tabela auxiliar 6)	PASS

testAverageValueSalesPurchasesDistrictECP02

Test Case ID	TC# testAverageValuePurchasesDistrictECP02	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
5 janeiro	22:40h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado : Objeto do tipo Orders, 6 do tipo Product e 4 do tipo Order				
Procedimentos				
Adicionar produtos à order; -Adicionar encomendas ao objeto do tipo Orders; -Registar encomendas no <i>ledger</i> ; (Repetir passo anterior e este 10000 vezes) -Invocar método <i>averageValueSalesPurchasesDistrict</i> ; - Comparar <i>map</i> esperado com <i>map</i> retornado pelo método;				
Requisito/Us e Case/	Inputs	Valores do Inputs	Resultados	Pass/Fail/Untested

Funcionalidade			Esperados	
Obter média de compras e vendas por distrito	Não tem	(Ver tabela auxiliar 6)	map	PASS

Tabela auxiliar 6

Inputs: this.product = new Product(1, "nome", "produto fixe", 25, 2, 5.4, 7); this.product2 = new Product(2, "nome", "produto top", 5.0, 1, 6.4, 4); this.product3 = new Product(3, "nome", "produto reacondiciona", 5.0, 2, 7.4, 4); this.product4 = new Product(4, "nome", "produto frágil", 2.0, 1, 8.4, 5); this.product5 = new Product(5, "nome", "produto informático", 2.0, 1, 8.4, 5); this.product6 = new Product(6, "nome", "produto unico", 2.0, 1, 8.4, 5);
this.basicOrder1 = new Order("1", "2021-2-2", basicEntity1, basicEntity2); this.basicOrder2 = new Order("2", "2021-2-2", basicEntity3, basicEntity4); this.basicOrder3 = new Order("3", "2021-2-2", basicEntity3, basicEntity4); this.basicOrder4 = new Order("3", "2021-2-2", basicEntity3, basicEntity4);
basicOrder1.addProduct(product); basicOrder1.addProduct(product2); basicOrder2.addProduct(product2); basicOrder3.addProduct(product3); basicOrder4.addProduct(product2);
basicOrders.registerOrdersInLedger(); basicOrders.addOrder(basicOrder1); basicOrders.registerOrdersInLedger();
Resultados Esperados: {Beja={Venda=55.0, Compra=0.0}, Porto={Venda=0.0, Compra=6.67}, Braga={Venda=6.67, Compra=0.0}, Aveiro={Venda=0.0, Compra=55.0}}

testAverageValueSalesPurchasesDistrictECP03

Test Case ID	TC# testAverageValuePurchasesDistrictECP02	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
5 janeiro	22:40h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado : Objeto do tipo Orders, 6 do tipo Product e 3 do tipo Order (uma delas deve ter a mesma entidade de emissão e receção)				
Procedimentos				
Adicionar produtos à order; -Adicionar encomendas ao objeto do tipo Orders; -Registar encomendas no <i>ledger</i> ; -Invocar método <u>averageValueSalesPurchasesDistrict</u> ; - Comparar <i>map</i> esperado com <i>map</i> retornado pelo método;				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter média de compras e vendas por distrito	3 objetos do tipo Order com produtos adicionados	<u>Ver tabela 6</u>	Map*	PASS

* {Porto={Venda=0.0, Compra=5.0}, Beja={Venda=55.0, Compra=0.0}, Braga={Venda=6.67, Compra=10.0}, Aveiro={Venda=0.0, Compra=55.0}}

testNumberOrdersByDistrictBVA01

Test Case ID	TC# testNumberOrdersByDistrictBVA01	Dependências	
--------------	--	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12 janeiro	20:13h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado: 1 objeto do tipo OrdersMetricStatistics				
Procedimentos				
-Invocar método <u>numberOrderSentReceivedByDistrict;</u>				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter número de encomendas enviadas/recebidas por distrito	ArrayList<Block>	null	IllegalArgumentException	PASS

testNumberOrdersByDistrictECP01

Test Case ID	TC#	Dependências	
	testNumberOrdersByDistrictBVA01		

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12 janeiro	20:13h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado: 1 objeto do tipo Orders e 3 do tipo Order.				
Procedimentos				
<ul style="list-style-type: none"> - Adicionar produtos às encomendas; - Adicionar encomendas ao objeto do tipo Orders; - Registrar encomendas no <i>ledger</i>; - Invocar método <code>numberOrdersByDistrict</code>; 				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter número de encomendas enviadas/recebidas por distrito	ArrayList<Block>	Tabela Auxiliar X	*	PASS

* {Porto={recebidas=3, Enviadas=1}, Beja={recebidas=0, Enviadas=1}, Braga={Recebidas=0, Enviadas=1}, Aveiro={Recebidas=1, Enviadas=1}}

testStandardDeviationNumberProductTransactionsBVA01

Test Case ID	TC# testStandardDeviationNumberProductTransactionsBVA01	Dependências	
---------------------	--	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12 janeiro	16:13h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado 1 objeto do tipo OrdersMetricStatistics				
Procedimentos				
-Invocar método <u>standardDeviationNumberProductTransactions</u>				
Requisito /Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Un tested
Obter desvio padrão do número de produtos	ArrayList<Block>	null	IllegalArgumentException	PASS

testStandardDeviationNumberProductTransactionsECP01

Test Case ID	TC# testStandardDeviationNumberProductTransactionsECP01	Dependências	
---------------------	--	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12 janeiro	16:13h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado 1 objeto do tipo Orders e 3 do tipo Order.				
Procedimentos				
<ul style="list-style-type: none"> - Adicionar produtos às <i>orders</i> instanciadas; - Invocar método de registo de <i>orders</i> no <i>ledger</i>; - Invocar método <u>standardDeviationNumberProductTransactions</u>; 				
Requisito/ Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter desvio padrão do valor médio de produtos por transação	ArrayList<Block>	Ver tabela auxiliar x	0.82	PASS

testStandardDeviationTransactionsECP01

Test Case ID	TC# testStandardDeviationTransactionsECP01	Dependências	
---------------------	---	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12 janeiro	16:13h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado 1 objeto do tipo Orders e 3 do tipo Order.				
Procedimentos				
<ul style="list-style-type: none"> - Adicionar produtos às <i>orders</i> instanciadas; - Invocar método de registo de <i>orders</i> no <i>ledger</i>; - Invocar método <u>standardDeviationNumberProductTransactions</u>; 				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter desvio padrão do valor médio de transações	3 orders inseridas para registo	Ver tabela auxiliar x	24.53	PASS

Tabela Auxiliar X

```
this.product = new Product(1, "nome", "produto fixe", 25, 2, 5.4, 7);  
this.product2 = new Product(2, "nome", "produto top", 5.0, 1, 6.4, 4);  
this.product3 = new Product(3, "nome", "produto reacondiciona", 5.0, 2, 7.4, 4);  
this.product4 = new Product(4, "nome", "produto frágil", 2.0, 1, 8.4, 5);  
this.product5 = new Product(5, "nome", "produto informático", 2.0, 1, 8.4, 5);  
this.product6 = new Product(6, "nome", "produto unico", 2.0, 1, 8.4, 5);
```

- basicOrder1.addProduct(product);
- basicOrder1.addProduct(product2);
- basicOrder2.addProduct(product4);
- basicOrder2.addProduct(product5);
- basicOrder3.addProduct(product6);
- basicOrders.addOrder(basicOrder1);
- basicOrders.addOrder(basicOrder2);
- basicOrders.addOrder(basicOrder3);
-

```
basicOrders.registerOrdersInLedger();
```

testStandardDeviationTransactionsBVA01

Test Case ID	TC# testStandardDeviationTransactionsBVA01	Dependências	
---------------------	---	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12 janeiro	16:13h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado 1 objeto do tipo OrdersMetricStatistics				
Procedimentos				
-Invocar método <u>standardDeviationTransactionsBVA01</u> ;				
Requisito/ Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter desvio padrão do valor médio de transações	Não tem	null	IllegalArgumentException	PASS

Detalhes Product

testBasicProductBVA08

Test Case ID	TC# testBasicProductBVA08	Dependências	
--------------	------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
15 janeiro	16:13h	Hugo	PASS	

Pré-condições / estado inicial				
Não tem				
Procedimentos				
-Invocar construtor <u>Product</u> ;				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar um produto	Int, String , String, Double,Int,Double,Double	1, "Product1", "Description", 1, 3, 63.1, 5.2l	IllegalArgumentException	PASS

testBasicProductBVA06

Test Case ID	TC# testBasicProductBVA06	Dependências	
--------------	------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
15 janeiro	16:13h	Hugo	PASS	

Pré-condições / estado inicial				
Não tem				
Procedimentos				
-Invocar construtor <u>Product</u> ;				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar um produto	Int, String , String, Double,Int,Double,Double	1, "Product1", "Description", 1, 3, 0, 5.2l	IllegalArgumentExcepti on	PASS

testBasicProductBVA09

Test Case ID	TC# testBasicProductBVA09	Dependências	
--------------	------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
15 janeiro	16:45	Hugo	PASS	

Pré-condições / estado inicial				
Não tem				
Procedimentos				
-Invocar construtor <u>Product</u> ;				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar um produto	Int, String , String, Double,Int,Double,Double	1, null,Description", 1, 3, 0, 5.2l	IllegalArgumentExcept ion	PASS

testBasicProductBVA10

Test Case ID	TC# testBasicProductBVA10	Dependências	
--------------	------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
15 janeiro	16:33h	Hugo	PASS	

Pré-condições / estado inicial				
Não tem				
Procedimentos				
-Invocar construtor <u>Product</u> ;				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar um produto	Int, String , String, Double,Int,Double,Double	1, "Product1", null, 1, 3, 0, 5.2l	IllegalArgumentException	PASS

Detalhes Truck

TestTruckBVA01

Test Case ID	TC# testTruckBVA02	Dependências	
--------------	-----------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
15 janeiro	16:33h	Hugo	PASS	

Pré-condições / estado inicial				
Não tem				
Procedimentos				
-Invocar construtor <u>Truck</u> ;				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestado
Criar um caminhão	String	""	IllegalArgumentException	PASS

TestTruckBVA02

Test Case ID	TC# <u>testTruckBVA02</u>	Dependências	
--------------	------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	16:33h	Hugo	PASS	

Pré-condições / estado inicial				
Não tem				
Procedimentos				
-Invocar construtor <u>Truck</u> ;				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestado
Criar um caminhão	String	""	IllegalArgumentException	PASS

TestTruckECP01

Test Case ID	TC# <u>testTruckBVA02</u>	Dependências	
--------------	---------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	16:33h	Hugo	PASS	

Pré-condições / estado inicial				
Não tem				
Procedimentos				
-Invocar construtor <u>Truck</u> ;				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar um caminhão	String name	"HUGO"	Instanceof Truck (true)	PASS

TestAddAllOrderInTruckBVA01

Test Case ID	TC# testTruckBVA01	Dependências	
--------------	-----------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	16:33h	Hugo	PASS	

Pré-condições / estado inicial				
Objeto do tipo Truck instanciado;				
Procedimentos				
-Invocar método <u>addAllOrderInTruck</u> ;				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar produto ao caminhão	IOrder order	null	IllegalArgumentException	PASS

TestAddAllOrderInTruckBVA02

Test Case ID	TC# testTruckBVA02	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	17:33h	Hugo	PASS	
Pré-condições / estado inicial				
Objeto do tipo Truck instanciado;				
Procedimentos				
-Invocar método <u>addAllOrderInTruck</u> ;				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar produto ao caminhão	IOrder order	Orde.getVolum() = 63.1	False	PASS

TestAddAllOrderInTruckBVA03

Test Case ID	TC# testTruckBVA02	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	17:33h	Hugo	PASS	
Pré-condições / estado inicial				
Objeto do tipo Truck instanciado;				
Procedimentos				
-Invocar método <u>addAllOrderInTruck</u> ;				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar produto ao caminhão	IOrder order	Orde.getVolum() = 0	IllegalArgumentExceptio n	PASS

testAddAllOrderInTruckProductInTruckECP01

Test Case ID	TC#	Dependências	
	testAddAllOrderInTruckProductInTruckECP01		

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	17:53h	Hugo	PASS	

Pré-condições / estado inicial				
Objeto do tipo Truck instanciado;				
Procedimentos				
-Invocar método <u>addAllOrderInTruck</u> ;				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar produto ao caminhão	IOrder order	order.getVolume() = 45	true	PASS

testAddProductInTruckBVA01

Test Case ID	TC#	Dependências	
	testAddProductInTruckBVA01		

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	19:53h	Hugo	PASS	

Pré-condições / estado inicial				
Objeto do tipo Truck instanciado;				
Procedimentos				
Invocar método <u>addProductInTruck</u> ;				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar produto ao caminhão	String idOrder, String idProduct, Double volume	"idpordutor,"23", 63.1	false	PASS

testAddProductInTruckBVA02

Test Case ID	TC#	Dependências	
	testAddProductInTruckBVA02		

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	17:08h	Hugo	PASS	

Pré-condições / estado inicial				
Objeto do tipo Truck instanciado;				
Procedimentos				
-Invocar método addProductInTruck;				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar produto ao caminhão	String idOrder, String idProduct, Double volume	"idpordutor","23", 0	IllegalArgumentExcepcion	PASS

testAddProductInTruckBVA03

Test Case ID	TC#	Dependências	
	testAddProductInTruckBVA03		

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	21:45h	Hugo	PASS	

Pré-condições / estado inicial				
Objeto do tipo Truck instanciado;				
Procedimentos				
-Invocar método addProductInTruck;				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar um produto a um caminhão	String idOrder, String idProduct, Double volume	null,"23", 5	IllegalArgumentExcepcion	PASS

testAddProductInTruckBVA04

Test Case ID	TC#	Dependências	
	testAddProductInTruckBVA04		

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	18:53h	Hugo	PASS	

Pré-condições / estado inicial				
Objeto do tipo Truck instanciado;				
Procedimentos				
-Invocar método addProductInTruck;				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar um produto a um caminhão	String idOrder, String idProduct, Double volume	"ws", null, 5	IllegalArgumentException	PASS

testAddProductInTruckBVA05

Test Case ID	TC#	Dependências	
	testAddProductInTruckBVA05		

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
19 janeiro	16:38h	Hugo	PASS	

Pré-condições / estado inicial				
Objeto do tipo Truck instanciado;				
Procedimentos				
-Invocar método addProductInTruck;				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar um produto a um caminhão	String idOrder, String idProduct, Double volume	"" , "34" , 5	IllegalArgumentException	PASS

testAddProductInTruckECP01

Test Case ID	TC#	Dependências		
	testAddProductInTruckECP01			
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	18:53h	Hugo	PASS	
Pré-condições / estado inicial				
Objeto do tipo Truck instanciado;				
Procedimentos				
-Invocar método <u>addProductInTruck</u> ;				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar um produto a um caminhão	String idOrder, String idProduct, Double volume	"ws","ola", 5	true	PASS

testGetProductContainerTruckBVA01

Test Case ID	TC#	Dependências		
	testGetProductContainerTruckBVA01			
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	19:53h	Hugo	PASS	
Pré-condições / estado inicial				
Objeto do tipo Truck instanciado;				
Procedimentos				
-Invocar método <u>getProductContainerTruck</u> ;				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter produto de um caminhão	IProductsInsideTruck iProductsInsideTruck	null	IllegalArgumentException	PASS

testGetProductContainerTruckECP01

Test Case ID	TC#	Dependências	
	testGetProductContainerTruckECP01		

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	20:53h	Hugo	PASS	

Pré-condições / estado inicial				
Objeto do tipo Truck instanciado; Objeto do tipo ProductInsideTruck instanciado;				
Procedimentos				
-Adicionar produto no caminhão; -Invocar método <code>getProductContainerTruck</code> (passando como parâmetro o produto adicionado no passo anterior);				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter produto de um caminhão	IProductsInsideTruck iProductsInsideTruck	iProductsInsideTruck	ProductInsideTruck	PASS

testGetProductContainerTruckECP02

Test Case ID	TC#	Dependências	
	testGetProductContainerTruckECP02		

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	20:53h	Hugo	PASS	

Pré-condições / estado inicial				
Objeto do tipo Truck instanciado; Objeto do tipo ProductInsideTruck instanciado;				
Procedimentos				
-Invocar método <code>getProductContainerTruck</code> ;				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter produto de um caminhão	IProductsInsideTruck iProductsInsideTruck	iProductsInsideTruck	null	PASS

testGetProductListProductsInsideTruckBVA01

Test Case ID	TC#	Dependências	
	testGetProductListProductsInsideTruckBVA01		

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
20 janeiro	11:33h	Hugo	PASS	

Pré-condições / estado inicial				
Objeto do tipo DistrictTrucks instanciado;				
Procedimentos				
-Invocar método <u>getProductListProductsInsideTruck</u> ;				
Requisito/ Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter produto de um caminhão através de um índice	Int index	-1	IndexOutOfBoundsException	PASS

testGetProductListProductsInsideTruckBVA02

Test Case ID	TC#	Dependências	
	testGetProductListProductsInsideTruckBVA02		

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
20 janeiro	11:33h	Hugo	PASS	

Pré-condições / estado inicial				
Objeto do tipo DistrictTrucks instanciado;				
Procedimentos				
-Invocar método <u>getProductListProductsInsideTruck</u> ;				
Requisito/ Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter produto de um caminhão através de um índice	Int index	truck.numberProductsInContainer	IndexOutOfBoundsException	PASS

TestGetProductListProductsInsideTruckECP01

Test Case ID	TC# testGetProductListProductsInsideTruckECP01	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
20 janeiro	11:33h	Hugo	PASS	

Pré-condições / estado inicial				
Objeto do tipo DistrictTrucks instanciado;				
Procedimentos				
-Invocar método <u>getProductListProductsInsideTruck</u> ;				
Requisito /Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Un tested
Obter produto de um caminhão através de um índice	Int index	truck.numberProductInContainer	instanceof ProductsInsideTruck	PASS

Detalhes ProductsInsideTruck

testProductsInsideTruckBVA01

Test Case ID	TC# testProductsInsideTruckBVA01	Dependências	
--------------	-------------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	20:53h	Hugo	PASS	

Pré-condições / estado inicial				
Procedimentos				
-Invocar construtor <u>ProductsInsideTruck</u> ;				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar produto para ser inserido dentro do caminhão	String idOrder, String idProduct, int quantity, double uniqueSize	"" , "Hugo" , 4, 2.2	IllegalArgumentExcep tion	PASS

testProductsInsideTruckBVA02

Test Case ID	TC# testProductsInsideTruckBVA02	Dependências	
--------------	-------------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	20:53h	Hugo	PASS	

Pré-condições / estado inicial				
Procedimentos				
-Invocar construtor <u>ProductsInsideTruck</u> ;				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar produto para ser inserido dentro do caminhão	String idOrder, String idProduct, int quantity, double uniqueSize	"Hugo", "" ,4, 2.2	IllegalArgumentExcep tion	PASS

testProductsInsideTruckBVA03

Test Case ID	TC# testProductsInsideTruckBVA03	Dependências	
--------------	-------------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	20:43h	Hugo	PASS	

Pré-condições / estado inicial				
Procedimentos				
-Invocar construtor <u>ProductsInsideTruck</u> ;				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar produto para ser inserido dentro do caminhão	String idOrder, String idProduct, int quantity, double uniqueSize	"Hugo", "ID_PRODUC T", 0, 2.2	IllegalArgumentExcep tion	PASS

testProductsInsideTruckBVA04

Test Case ID	TC# testProductsInsideTruckBVA04	Dependências	
--------------	-------------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	23:13h	Hugo	PASS	

Pré-condições / estado inicial				
Procedimentos				
-Invocar construtor <u>ProductsInsideTruck</u> ;				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar produto para ser inserido dentro do caminhão	String idOrder, String idProduct, int quantity, double uniqueSize	"Hugo", "ID_PRODUC T", 4, 0	IllegalArgumentExcep tion	PASS

testProductsInsideTruckBVA05

Test Case ID	TC# testProductsInsideTruckBVA05	Dependências	
--------------	-------------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	22:13h	Hugo	PASS	

Pré-condições / estado inicial				
Procedimentos				
-Invocar construtor <u>ProductsInsideTruck</u> ;				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar produto para ser inserido dentro do caminhão	String idOrder, String idProduct, int quantity, double uniqueSize	"Hugo", null, ,4, 5.4	IllegalArgumentException	PASS

testProductsInsideTruckBVA06

Test Case ID	TC# testProductsInsideTruckBVA05	Dependências	
--------------	-------------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	22:13h	Hugo	PASS	

Pré-condições / estado inicial				
Procedimentos				
-Invocar construtor <u>ProductsInsideTruck</u> ;				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar produto para ser inserido dentro do caminhão	String idOrder, String idProduct, int quantity, double uniqueSize	null , "Hugo", ,4, 2.3	IllegalArgumentException	PASS

testProductsInsideTruckECP01

Test Case ID	TC#	Dependências	
	testProductsInsideTruckECP01		

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	22:18h	Hugo	PASS	

Pré-condições / estado inicial				
Procedimentos				
-Invocar construtor <u>ProductsInsideTruck</u> ;				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar produto para ser inserido dentro do caminhão	String idOrder, String idProduct, int quantity, double uniqueSize	"id" , "Hugo", ,4, 2.3	Instance of Product (true)	PASS

Detalhes DistrictTrucks

testGetTruckByIndexBVA01

Test Case ID	TC# testGetTruckByIndexBV01	Dependências	
--------------	--------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	22:33h	Hugo	PASS	

Pré-condições / estado inicial				
Objeto do tipo DistrictTrucks instanciado;				
Procedimentos				
-Invocar método <u>getTruckByIndex</u> ;				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter um caminhão através de um índice	Int index	-1	IndexOutOfBoundsException	PASS

testGetTruckByIndexBVA02

Test Case ID	TC# testGetTruckByIndexBV02	Dependências	
--------------	--------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	21:19h	Hugo	PASS	

Pré-condições / estado inicial				
Objeto do tipo DistrictTrucks instanciado;				
Procedimentos				
-Invocar método <u>getTruckByIndex</u> ;				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter um caminhão através de um índice	Int index	ordersByDistrictBeja.size()	IndexOutOfBoundsException	PASS

testGetTruckByIndexECP01

Test Case ID	TC# testGetTruckByIndexECP01	Dependências	
---------------------	---------------------------------	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	21:19h	Hugo	PASS	

Pré-condições / estado inicial				
Objeto do tipo DistrictTrucks instanciado; Objeto do tipo Order instanciado; Objeto do tipo Product instanciado;				
Procedimentos				
<ul style="list-style-type: none"> - Adicionar produto à encomenda; - Adicionar encomenda à lista de encomendas do objeto DistrictTrucks; - Agrupar encomendas por camiões, invocando método <u>agroupOrders()</u> - Invocar método <u>getTruckByIndex</u>; 				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter um camião através de um indice	Int index	0	Instance of Truck	PASS

testCreateTruckBVA01

Test Case ID	TC# testCreateTruckBVA01	Dependências	
---------------------	-----------------------------	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	21:13h	Hugo	PASS	

Pré-condições / estado inicial				
Objeto do tipo DistrictTrucks instanciado;				
Procedimentos				
-Invocar método <u>creatTruck</u> ;				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar um conjunto de camiões	Int numberTrucks	0	IllegalArgumentException	PASS

testCreateTruckECP01

Test Case ID	TC# testCreateTrucECP01	Dependências	
---------------------	----------------------------	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	21:09h	Hugo	PASS	

Pré-condições / estado inicial				
Objeto do tipo DistrictTrucks instanciado;				
Procedimentos				
-Invocar método <u>creatTruck</u> ;				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar um conjunto de camiões	Int numberTrucks	3	3	PASS

testCreateTruckECP02

Test Case ID	TC#	Dependências	
	testCreateTrucECP02		

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	21:59h	Hugo	PASS	

Pré-condições / estado inicial				
Objeto do tipo DistrictTrucks instanciado;				
Procedimentos				
-Invocar método <code>creatTruck</code> ;				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untest ed
Criar um conjunto de camiões	Int numberTrucks	-3	IllegalArgumentExcepti on	PASS

testDistrictTrucksBVA01

Test Case ID	TC#	Dependências	
	testDistrictTrucksBVA01		

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	21:59h	Hugo	PASS	

Pré-condições / estado inicial				
Procedimentos				
-Invocar método <code>creatTruck</code> ;				
Requisito/Us e Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untest ed
Criar um conjunto de camiões	String nameDistrict, int numberContainer, ArrayList<IOrder> ordersByDistrict	"Porto", 3, null	IllegalArgumentExcepti on	PASS

testDistrictTrucksBVA02

Test Case ID	TC# testDistrictTrucksBVA02	Dependências	
--------------	--------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	22:29h	Hugo	PASS	

Pré-condições / estado inicial				
Procedimentos				
-Invocar método <u>creatTruck</u> ;				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar um conjunto de camiões	String nameDistrict, int numberContainer, ArrayList<IOrder> ordersByDistrict	"" , 5, ordersByDistrictBeja	IllegalArgumentExcep tion	PASS

testDistrictTrucksBVA03

Test Case ID	TC# testDistrictTrucksBVA02	Dependências	
--------------	--------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	22:29h	Hugo	PASS	

Pré-condições / estado inicial				
Procedimentos				
-Invocar método <u>creatTruck</u> ;				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar um conjunto de camiões	String nameDistrict, int numberContainer, ArrayList<IOrder> ordersByDistrict	null, 5, ordersByDistrictBeja	IllegalArgumentExcep tion	PASS

testDistrictTrucksECP01

Test Case ID	TC# TestDistrictTrucksECP01	Dependências	
--------------	--------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	22:25h	Hugo	PASS	

Pré-condições / estado inicial				
Procedimentos				
-Invocar método <u>creatTruck</u> ;				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestado
Criar um conjunto de camiões	String nameDistrict, int numberContainer, ArrayList<IOrder> ordersByDistrict	"idProduct", 5, ordersByDistrictBeja	Instance of DistrictTruck (true)	PASS

Detalhes GroupAllTrucks

testAddDistrictTrucksBVA01

Test Case ID	TC# testAddDistrictTrucksBVA01	Dependências	
---------------------	-----------------------------------	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	22:25h	Hugo	PASS	

Pré-condições / estado inicial				
Objeto do tipo DistrictTrucks instanciado; Objeto do tipo GroupAllTrucks;				
Procedimentos				
-Invocar método <u>addDistrictTrucks</u> ;				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adiciona um conjunto de caminhões de um distrito à lista responsável por armazenar todos os resultados do agrupamento	IDistrictTrucks iDistrictExpedition	null	IllegalArgumentExcep tion	PASS

TestAddDistrictTrucksECP01

Test Case ID	TC#	Dependências	
	TestDistrictTrucksECP01		

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	22:25h	Hugo	PASS	

Pré-condições / estado inicial				
Objeto do tipo DistrictTrucks instanciado; Objeto do tipo GroupAllTrucks;				
Procedimentos				
-Invocar método <u>addDistrictTrucks</u> ;				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adiciona um conjunto de camiões de um distrito à lista responsável por armazenar todos os resultados do agrupamento	IDistrictTrucks iDistrictExpedition	iDistrictExpedition	true	PASS

Detalhes TestExpeditionMethods

testGetArrayOrdersByDistrictBVA01

Test Case ID	TC# testGetArrayOrdersByDistrictBVA01	Dependências	
--------------	--	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	22:37h	Hugo	PASS	

Pré-condições / estado inicial				
Objeto do tipo ExpeditionMethods;				
Procedimentos				
-Invocar método <u>getArrayOrdersByDistrict</u> ;				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Unexpected
Obter lista de encomendas de um distrito	String district, ArrayList<IOrder> listOrders	"Porto", null	IllegalArgumentException	PASS

testGetArrayOrdersByDistrictBVA02

Test Case ID	TC# testGetArrayOrdersByDistrictBVA02	Dependências	
---------------------	--	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	22:37h	Hugo	PASS	

Pré-condições / estado inicial				
Objeto do tipo ExpeditionMethods;				
Procedimentos				
-Invocar método <u>getArrayOrdersByDistrict</u> ;				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter lista de encomendas de um distrito	String district, ArrayList<IOrder> listOrders	"" listOrders	IllegalArgumentExce ption	PASS

testGetArrayOrdersByDistrictBVA03

Test Case ID	TC# testGetArrayOrdersByDistrictBVA02	Dependências	
---------------------	--	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	22:37h	Hugo	PASS	

Pré-condições / estado inicial				
Objeto do tipo ExpeditionMethods intanciado;				
Procedimentos				
-Invocar método <u>getArrayOrdersByDistrict</u> ;				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter lista de encomendas de um distrito	String district, ArrayList<IOrder> listOrders	null, listOrders	IllegalArgumentExce ption	PASS

testGetArrayOrdersByDistrictECP01

Test Case ID	TC# testGetArrayOrdersByDistrictECP01	Dependências	
--------------	--	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	22:37h	Hugo	PASS	

Pré-condições / estado inicial				
- Objeto do tipo ExpeditionMethods instanciado; - Objeto do tipo Order instanciado; - Dois objetos do tipo Product instanciado;				
Procedimentos				
- Adicionar produtos à order; - Invocar método <u>getArrayOrdersByDistrict</u> , passando como parâmetro a order do passo anterior;				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter lista de encomendas de um distrito	String district, ArrayList<IOrder> listOrders	"Porto", listOrders	true	PASS

testGroupOrdersByTrucksBVA01

Test Case ID	TC# testGroupOrdersByTrucksBVA01	Dependências	
--------------	-------------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	22:37h	Hugo	PASS	

Pré-condições / estado inicial				
Objeto do tipo ExpeditionMethods intanciado;				
Procedimentos				
-Invocar método <u>groupOrdersByTrucks</u> ;				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter lista de encomendas de um distrito	ArrayList<IOrder> listOrders	null	IllegalArgumentExce ption	PASS

testGroupOrdersByTrucksECP01

Test Case ID	TC#	Dependências	
	testGroupOrdersByDistrictECP01		

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	23:37h	Hugo	PASS	

Pré-condições / estado inicial				
- Objeto do tipo ExpeditionMethods instanciado; - 3 Objetos do tipo Order instanciados; - 1 Objeto do tipo Orders instanciado; - 6 objetos do tipo Product instanciado;				
Procedimentos				
- Adicionar produtos às orders; - Adicionar orders ao objeto do tipo Orders; - Criar um objeto com o resultado esperado; - Invocar método <u>groupOrdersByTrucks</u> , passando como parâmetro a lista e comparar com o objeto esperado				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestado
Obter lista de encomendas de um distrito	ArrayList<IOrder> listOrders Ver tabela xy	Ver tabela xy	Ver tabela xy	PASS

Tabela auxiliar XY

```

this.product = new Product(1, "nome", "produto fixe", 25, 2, 31, 7);
this.product2 = new Product(2, "nome", "produto top", 5.0, 1, 62.9, 4);
this.product3 = new Product(3, "nome", "produto reacondiciona", 5.0, 6, 1, 4);
this.product4 = new Product(4, "nome", "produto frágil", 2.0, 1, 2, 5);
this.product5 = new Product(5, "nome", "produto informático", 2.0, 1, 0.12, 5);
this.product6 = new Product(6, "nome", "produto unico", 2.0, 1, 62.999999, 5);

this.basicOrder1 = new Order("20210101_01", "2021-2-2", basicEntity1, basicEntity2);
this.basicOrder2 = new Order("20210101_02", "2021-2-2", basicEntity1, basicEntity4);
this.basicOrder3 = new Order("3", "2021-2-2", basicEntity3, basicEntity3);
this.basicOrder4 = new Order("3", "2021-2-2", basicEntity4, basicEntity1);

basicOrder1.addProduct(product);
basicOrder1.addProduct(product2);
basicOrder2.addProduct(product3);
basicOrder2.addProduct(product4);

```

```
basicOrder2.addProduct(product5);
basicOrder4.addProduct(product5);
basicOrder4.addProduct(product6);
```

```
basicOrders.addOrder(basicOrder1);
basicOrders.addOrder(basicOrder2);
basicOrders.addOrder(basicOrder4);
```

Resultado Esperado:

```
[
  {
    "Trucks": [
      {
        "AvaibleCapacity": 0.100000000000000142,
        "TruckName": "Truck1",
        "ProductInContainer": [
          {
            "Unit_Volume": 62.9,
            "Id_Order": "20210101_01",
            "Id_Product": "2",
            "Quantity": 1
          }
        ]
      },
      {
        "AvaibleCapacity": 1.0,
        "TruckName": "Truck2",
        "ProductInContainer": [
          {
            "Unit_Volume": 31.0,
            "Id_Order": "20210101_01",
            "Id_Product": "1",
            "Quantity": 2
          }
        ]
      },
      {
        "AvaibleCapacity": 54.88,
        "TruckName": "Truck3",
        "ProductInContainer": [
          {
            "Unit_Volume": 1.0,
            "Id_Order": "20210101_02",
            "Id_Product": "3",
            "Quantity": 6
          },
          {
            "Unit_Volume": 2.0,
            "Id_Order": "20210101_02",
            "Id_Product": "4",

```

```

        "Quantity": 1
      },
      {
        "Unit_Volume": 0.12,
        "Id_Order": "20210101_02",
        "Id_Product": "5",
        "Quantity": 1
      }
    ]
  },
  "NumberTrucks": 3,
  "District": "Beja"
},
{
  "Trucks": [
    {
      "AvaibleCapacity": 9.999999974752427E-7,
      "TruckName": "Truck1",
      "ProductInContainer": [
        {
          "Unit_Volume": 62.999999,
          "Id_Order": "3",
          "Id_Product": "6",
          "Quantity": 1
        }
      ]
    },
    {
      "AvaibleCapacity": 62.88,
      "TruckName": "Truck2",
      "ProductInContainer": [
        {
          "Unit_Volume": 0.12,
          "Id_Order": "3",
          "Id_Product": "5",
          "Quantity": 1
        }
      ]
    }
  ]
},
  "NumberTrucks": 2,
  "District": "Porto"
}
]

```

testGroupOrdersByTrucksECP02

Test Case ID	TC#	Dependências	
	testGroupOrdersByDistrictECP02		

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	23:57h	Hugo	PASS	

Pré-condições / estado inicial				
<ul style="list-style-type: none"> - Objeto do tipo ExpeditionMethods instanciado; - 1 Objetos do tipo Order instanciados (Deve ter mesmo emissor e o mesmo receptor); - 1 Objeto do tipo Orders instanciado; - 2 objetos do tipo Product instanciado; 				
Procedimentos				
<ul style="list-style-type: none"> - Adicionar produtos às orders; - Invocar método <u>groupOrdersByTrucks</u>, passando como parâmetro a encomenda que tem o mesmo sender e o mesmo receptor; - Comparar resultado (deve ser retornado um objeto vazio) 				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestado
Obter lista de encomendas de um distrito	Ver tabela YX	Ver tabela YX	[]	PASS

Tabela auxiliar YX

```

this.product3 = new Product(3, "nome", "produto reacondiciona", 5.0, 6, 1, 4);

this.basicOrder1 = new Order("20210101_01", "2021-2-2", basicEntity1, basicEntity2);
this.basicOrder2 = new Order("20210101_02", "2021-2-2", basicEntity1, basicEntity4);
this.basicOrder3 = new Order("3", "2021-2-2", basicEntity3, basicEntity3);

basicOrder3.addProduct(product);
basicOrder3.addProduct(product);
basicOrder3.addProduct(product3);
basicOrders.addOrder(basicOrder3);

Resultado Esperado:
[]

```

testExportGroupOrdersForFileJsonECP01

Test Case ID	TC# testExportGroupOrdersForFileJson ECP01	Dependências	
--------------	--	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	23:57h	Hugo	PASS	

Pré-condições / estado inicial				
- Objeto do tipo ExpeditionMethods instanciado; - 1 Objetos do tipo Order instanciados (Deve ter mesmo emisor e o mesmo receptor); - 1 Objeto do tipo Orders instanciado; - 2 objetos do tipo Product instanciado;				
Procedimentos				
- Adicionar produtos às order de como a que o volume total seja 126m ³ ; - Invocar método <u>ExportGroupOrdersForFileJson</u> ;				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter lista de encomendas de um distrito	Ver tabela XYW	Ver tabela XYW	3	PASS

Detalhes Product

ProductBVA01

Test Case ID	TC#ProductBVA01	Dependências	
--------------	-----------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
04 janeiro	19h	Fábio	PASS	

Pré-condições / estado inicial				
Procedimentos				
Invocar construtor da class Product, passar sete parâmetros, parametro id do tipo <i>int</i> , name do tipo <i>String</i> , description do tipo <i>String</i> , value do tipo <i>double</i> , quantity do tipo <i>int</i> , volume value do tipo <i>double</i> , e weight do tipo <i>double</i> . O parâmetro id será igual a -1, que é um input invalido, para os outros parametros serão passados inputs validos.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestado
Criar Produto	id, name, description, value, quantity, volume, weight	id = -1, name = "Product1", description = "Description", value = 2, quantity = 3, volume = 4.1, weight = 5.2	IllegalArgumentException	PASS

ProductBVA02

Test Case ID	TC#ProductBVA02	Dependências	
--------------	-----------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
04 janeiro	19h	Fábio	PASS	

Pré-condições / estado inicial				
Procedimentos				
Invocar construtor da class Product, passar sete parâmetros, parametro id do tipo <i>int</i> , name do tipo <i>String</i> , description do tipo <i>String</i> , value do tipo <i>double</i> , quantity do tipo <i>int</i> , volume value do tipo <i>double</i> , e weight do tipo <i>double</i> . O parâmetro name será igual a "", que é um input invalido, para os outros parametros serão passados inputs validos.				

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestado
Criar Produto	id, name, description, value, quantity, volume, weight	id = 1, name = "", description = "Description", value = 2, quantity = 3, volume = 4.1, weight = 5.2	IllegalArgumentException	PASS

ProductBVA03

Test Case ID	TC#ProductBVA03	Dependências	
--------------	-----------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
04 janeiro	19h	Fábio	PASS	

Pré-condições / estado inicial				
Procedimentos				
<p>Invocar construtor da class Product, passar sete parâmetros, parametro id do tipo <i>int</i>, name do tipo <i>String</i>, description do tipo <i>String</i>, value do tipo <i>double</i>, quantity do tipo <i>int</i>, volume value do tipo <i>double</i>, e weight do tipo <i>double</i>.</p> <p>O parâmetro description será igual a "", que é um input invalido, para os outros parametros serão passados inputs validos.</p>				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestado
Criar Produto	id, name, description, value, quantity, volume, weight	id = 1, name = "Product1", description = "", value = 2, quantity = 3, volume = 4.1, weight = 5.2	IllegalArgumentException	PASS

ProductBVA04

Test Case ID	TC#ProductBVA04	Dependências	
--------------	-----------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
04 janeiro	19h	Fábio	PASS	

Pré-condições / estado inicial

Procedimentos

Invocar construtor da class Product, passar sete parâmetros, parametro id do tipo *int*, name do tipo *String*, description do tipo *String*, value do tipo *double*, quantity do tipo *int*, volume value do tipo *double*, e weight do tipo *double*.

Sendo que o parâmetro value será igual a -0.1, que é um *input* invalido, para os outros parametros serão passados inputs validos.

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestado
Criar Produto	id, name, description, value, quantity, volume, weight	id = 1, name = "Product1", description = "Description", value = -0.1, quantity = 3, volume = 4.1, weight = 5.2	IllegalArgumentException	PASS

ProductBVA05

Test Case ID	TC#ProductBVA05	Dependências	
--------------	-----------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
04 janeiro	19h	Fábio	PASS	

Pré-condições / estado inicial				
Procedimentos				
<p>Invocar construtor da class Product, passar sete parâmetros, parametro id do tipo <i>int</i>, name do tipo <i>String</i>, description do tipo <i>String</i>, value do tipo <i>double</i>, quantity do tipo <i>int</i>, volume value do tipo <i>double</i>, e weight do tipo <i>double</i>.</p> <p>O parâmetro quantity será igual a 0, que é um input invalido, para os outros parametros serão passados inputs validos.</p>				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestado
Criar Produto	id, name, description, value, quantity, volume, weight	id = 1, name = "Product1", description = "Description", value = 1, quantity = 0, volume = 4.1, weight = 5.2	IllegalArgumentException	PASS

ProductBVA06

Test Case ID	TC#ProductBVA06	Dependências	
--------------	-----------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
04 janeiro	19h	Fábio	PASS	

Pré-condições / estado inicial				
Procedimentos				
<p>Invocar construtor da class Product, passar sete parâmetros, parametro id do tipo <i>int</i>, name do tipo <i>String</i>, description do tipo <i>String</i>, value do tipo <i>double</i>, quantity do tipo <i>int</i>, volume value do tipo <i>double</i>, e weight do tipo <i>double</i>.</p> <p>O parâmetro volume será igual a 0, que é um input invalido, para os outros parametros serão passados inputs validos.</p>				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestado
Criar Produto	id, name, description, value, quantity, volume, weight	id = 1, name = "Product1", description = "Description", value = 1, quantity = 3, volume = 0, weight = 5.2	IllegalArgumentException	PASS

ProductBVA07

Test Case ID	TC#ProductBVA07	Dependências	
--------------	-----------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
04 janeiro	19h	Fábio	PASS	

Pré-condições / estado inicial				
Procedimentos				
<p>Invocar construtor da class Product, passar sete parâmetros, parametro id do tipo <i>int</i>, name do tipo <i>String</i>, description do tipo <i>String</i>, value do tipo <i>double</i>, quantity do tipo <i>int</i>, volume value do tipo <i>double</i>, e weight do tipo <i>double</i>.</p> <p>O parâmetro weight será igual a 0, que é um input invalido, para os outros parametros serão passados inputs validos.</p>				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestado
Criar Produto	id, name, description, value, quantity, volume, weight	id = 1, name = "Product1", description = "Description", value = 1, quantity = 3, volume = 4.1, weight = 0	IllegalArgumentException	PASS

ProductBVA08

Test Case ID	TC#ProductBVA08	Dependências	
--------------	-----------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
04 janeiro	19:10h	Fábio	PASS	

Pré-condições / estado inicial

Procedimentos
<p>Invocar construtor da class Product, passar sete parâmetros, parametro id do tipo <i>int</i>, name do tipo <i>String</i>, description do tipo <i>String</i>, value do tipo <i>double</i>, quantity do tipo <i>int</i>, volume value do tipo <i>double</i>, e weight do tipo <i>double</i>.</p> <p>O parâmetro volume será igual a 63.1, que é um input invalido, para os outros parâmetros serão passados inputs validos.</p>

Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestado
Criar Produto	id, name, description, value, quantity, volume, weight	id = 1, name = "Product1", description = "Description", value = 1, quantity = 3, volume = 63.1, weight = 5.2	IllegalArgumentException	PASS

ProductBVA09

Test Case ID	TC#ProductBVA09	Dependências	
--------------	-----------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
04 janeiro	19h	Fábio	PASS	

Pré-condições / estado inicial				
Procedimentos				
<p>Invocar construtor da class Product, passar sete parâmetros, parametro id do tipo <i>int</i>, name do tipo <i>String</i>, description do tipo <i>String</i>, value do tipo <i>double</i>, quantity do tipo <i>int</i>, volume value do tipo <i>double</i>, e weight do tipo <i>double</i>.</p> <p>O parâmetro <i>description</i> será igual a <i>null</i>, pois é um input invalido, para os outros parâmetros serão passados inputs validos.</p>				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestado
Criar Produto	id, name, description, value, quantity, volume, weight	id = 1, name = "Product1", description = <i>null</i> , value = 1, quantity = 3, volume = 4.1, weight = 5.2	IllegalArgumentException	PASS

ProductBVA10

Test Case ID	TC#ProductBVA10	Dependências	
--------------	-----------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
04 janeiro	19h	Fábio	PASS	

Pré-condições / estado inicial

--

Procedimentos

Invocar construtor da class Product, passar sete parâmetros, parametro id do tipo *int*, name do tipo *String*, description do tipo *String*, value do tipo *double*, quantity do tipo *int*, volume value do tipo *double*, e weight do tipo *double*.
O parâmetro name será igual a *null*, que é um input invalido, para os outros parâmetros serão passados inputs validos.

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestado
Criar Produto	id, name, description, value, quantity, volume, weight	id = 1, name = <i>null</i> , description = "Description", value = 1, quantity = 3, volume = 4.1, weight = 5.2	IllegalArgumentException	PASS

ProductECP01

Test Case ID	TC#ProductECP01	Dependências	
---------------------	-----------------	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
04 janeiro	19h	Fábio	PASS	

Pré-condições / estado inicial				
Procedimentos				
Para instanciar um produto é preciso passar sete parâmetros, id do tipo <i>int</i> , name do tipo <i>String</i> , description do tipo <i>String</i> , value do tipo <i>double</i> , quantity do tipo <i>int</i> , volume value do tipo <i>double</i> , e weight do tipo <i>double</i> .				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar Produto	id, name, description, value, quantity, volume, weight	id = 1, name = "Product1", description = "Description", value = 1, quantity = 3, volume = 4.1, weight = 5.2	Instanceof Product	PASS

Detalhes Class Order

OrderBVA01

Test Case ID	TC#OrderBVA01	Dependências	
--------------	---------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
04 janeiro	20h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado: duas entidades válidas, uma para o envio da transação e outra para a receção da encomenda.				
Procedimentos				
Invocar construtor passar dois parâmetros do tipo <i>String</i> (id e date), dois parâmetros do tipo <i>Entity</i> (sender e receiver). O parâmetro id será igual a "", que é um input invalido, para os outros parametros serão passados inputs validos.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestado
Criar Order	id, date, sender, receiver	id = "", date = "2021-01-01", sender = senderEntity, receiver = receiverEntity	IllegalArgumentException	PASS

OrderBVA02

Test Case ID	TC#OrderBVA02	Dependências	
--------------	---------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
04 janeiro	20h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado: duas entidades válidas, uma para o envio da transação e outra para a receção da encomenda.				
Procedimentos				
<p>Invocar construtor passar dois parâmetros do tipo <i>String</i> (id e date), dois parâmetros do tipo <i>Entity</i> (sender e receiver).</p> <p>O parâmetro date será igual a <i>""</i>, que é um <i>input</i> invalido, para os outros parametros serão passados inputs validos.</p>				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestado
Criar Order	id, date, sender, receiver	id = "1234" , date = "" , sender = senderEntity, receiver = receiverEntity	IllegalArgumentExceptio n	PASS

OrderBVA03

Test Case ID	TC#OrderBVA03	Dependências	
--------------	---------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
04 janeiro	20h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado: duas entidades válidas, uma para o envio da transação e outra para a receção da encomenda.				
Procedimentos				
<p>Invocar construtor passar dois parâmetros do tipo <i>String</i> (id e date), dois parâmetros do tipo <i>Entity</i> (sender e receiver).</p> <p>O parâmetro sender será igual a <i>null</i>, que é um <i>input</i> invalido, para os outros parametros serão passados inputs validos.</p>				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestado
Criar Order	id, date, sender, receiver	id = "", date = "2021-01-01", sender = null, receiver = receiverEntity	IllegalArgumentException	PASS

OrderBVA04

Test Case ID	TC#OrderBVA04	Dependências	
--------------	---------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
04 janeiro	20h	Fábio	PASS	

Pré-condições / estado inicial

Tem de ser instanciado: duas entidades válidas, uma para o envio da transação e outra para a receção da encomenda.

Procedimentos

Invocar construtor passar dois parâmetros do tipo *String* (id e date), dois parâmetros do tipo Entity (sender e receiver).

O parâmetro receiver será igual a *null*, que é um input invalido, para os outros parametros serão passados inputs validos.

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar Order	id, date, sender, receiver	id = "", date = "2021-01-01", sender = senderEntity, receiver = null	IllegalArgumentException	PASS

OrderECP01

Test Case ID	TC#OrderECP01	Dependências	
--------------	---------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
04 janeiro	20h	Fábio	PASS	

Pré-condições / estado inicial

Tem de ser instanciado: duas entidades válidas, uma para o envio da transação e outra para a receção da encomenda.

Procedimentos

Invocar construtor passar dois parâmetros do tipo *String* (id e date), dois parâmetros do tipo Entity (sender e receiver), todos eles válidos.

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar Order	id, date, sender, receiver	id = "1234", date = "2021-01-01", sender = senderEntity, receiver = receiverEntity	Instance of Order (true)	PASS

getProductBVA01

Test Case ID	TC#getProductBVA01	Dependências	
--------------	--------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
04 janeiro	20h	Fábio	PASS	

Pré-condições / estado inicial				
Terá de existir dois objetos do tipo Entity criados. Terá de existir um objeto do tipo Order criado.				
Procedimentos				
Invocar método getProduct do objeto Order (passando null como parâmetro).				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter um Product da Order	Product product	product = null	IllegalArgumentException	PASS

getProductECP01

Test Case ID	TC#getProductECP01	Dependências	
--------------	--------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
04 janeiro	20h	Fábio	PASS	

Pré-condições / estado inicial				
Terá de existir dois objetos do tipo Entity criados. Terá de existir um objeto do tipo Order criado. Terá de existir pelo menos um Product já adicionado à Order.				
Procedimentos				
Invocar método getProduct do objeto Order, sendo que o input deve corresponder a um Product existente na Order.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter um Product da Order	Product product	product = product1	InstanceOf Product	PASS

getProductECP02

Test Case ID	TC#getProductECP02	Dependências	
--------------	--------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
04 janeiro	20h	Fábio	PASS	

Pré-condições / estado inicial				
Terá de existir dois objetos do tipo Entity criados. Terá de existir um objeto do tipo Order criado.				
Procedimentos				
Invocar método getProduct do objeto Order. Neste caso o input não deve existir na transação.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter um Product da Order	product	product = product1	null	PASS

addProductBVA01

Test Case ID	TC#addProductBVA01	Dependências	
--------------	--------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
04 janeiro	20h	Fábio	PASS	

Pré-condições / estado inicial				
Terá de existir dois objetos do tipo Entity criados. Terá de existir um objeto do tipo Order criado.				
Procedimentos				
Adicionar Product à Order, passando um parâmetro do tipo null.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar um Product à Order	product	product = null	IllegalArgumentException	PASS

addProductECP01

Test Case ID	TC#addProductECP01	Dependências	
---------------------	--------------------	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
04 janeiro	20h	Fábio	PASS	

Pré-condições / estado inicial				
Terá de existir dois objetos do tipo Entity criados. Terá de existir um objeto do tipo Order criado.				
Procedimentos				
Adicionar Product à Order, passando um Product válido.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar um Product à Order	Product product	product = product1	true	PASS

addProductECP02

Test Case ID	TC#addProductECP02	Dependências	
---------------------	--------------------	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
04 janeiro	20h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de ser instanciados dois objetos do tipo Entity; Tem de ser instanciado um objeto do tipo Order; Tem de existir pelo menos um objeto do tipo Product já adicionado à Order;				
Procedimentos				
Adicionar um Product válido à Order que já existe.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar um Product à Order	Product product	product = product1	false	PASS

shippingCostCalculationECP01

Test Case ID	TC#shippingCostCalculationECP01	Dependências	
--------------	---------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
04 janeiro	16:00h	Fábio	PASS	

Pré-condições / estado inicial				
Têm de ser instanciados: duas entidades válidas, um objeto do tipo Order e um objeto do tipo Product.				
Procedimentos				
-Adicionar produto à order; -Invocar o método shippingCostCalculation do Objeto order.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestado
Obter valor do custo de envio de uma Order.	Ver tabela	Ver tabela	16.91*	PASS

*

Produto adicionado:

`IProduct product = new Product (1, "Product1", "Description, 2, 1, 4.1, 5.1);`

Como a senderEntity tem como district Viana do Castelo e a receiverEntity em como district Braga, a distancia entre as duas é igual a 60km.

Assim sendo o calculo é o seguinte:

$(\text{weight} + \text{distance}) * 0.25 + (0.15 * \text{volume}) = (5.2 + 60) * 0.25 + (0.15 * 4.1) = 16.91\text{€}$

shippingCostCalculationECP02

Test Case ID	TC#shippingCostCalculationECP02	Dependências	
--------------	---------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
14 janeiro	16:00h	Fábio	PASS	

Pré-condições / estado inicial				
Têm de ser instanciadas pelo menos duas entidades válidas, estas entidades deverão ter o mesmo valor de distrito, uma Order válida e existir pelo menos um Product já adicionado à order.				
Procedimentos				
Invocar o metodo shippingCostCalculation do Objeto order, o valor de envio deverá ser 0 pois a encomenda não tem custos de envio quando o sender e o receiver são do mesmo district.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestado
Obter valor do custo de envio de uma Order.	Entity sender, Entity receiver	Entity sender, Entity receiver	0	PASS

Detalhes Orders

getOrderBVA01

Test Case ID	TC#testGetOrderBVA01	Dependências	
--------------	----------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
05 janeiro	19:00h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de ser instanciada uma class Orders válida.				
Procedimentos				
Invocar método getOrder passando null como <i>input</i> .				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter order	order	null	IllegalArgumentException	PASS

GetOrderECP01

Test Case ID	TC#testGetOrderECP01	Dependências	
--------------	----------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
05 janeiro	19:00h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado : um objeto do tipo Orders e dois objetos do tipo Entity . Tem de ser instanciada uma Order. Terá de existir pelo menos uma Order já adicionada à class Orders.				
Procedimentos				
Invocar método getOrder passando como <i>input</i> uma Order já adicionada.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter order	order	order = order	Instanceof Order	PASS

GetOrderECP02

Test Case ID	TC#testGetOrderECP02	Dependências	
---------------------	----------------------	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
05 janeiro	19:00h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado : um objeto do tipo Orders e dois objetos do tipo Entity . Tem de ser instanciada uma Order.				
Procedimentos				
Invocar método getOrder passando como <i>input</i> uma Order que não tenha sido adicionada à class Orders.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter order	order	order = order	null	PASS

addOrderBVA01

Test Case ID	TC#testAddOrderBVA01	Dependências	
---------------------	----------------------	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
05 janeiro	19:15h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado uma class Orders válida.				
Procedimentos				
Invocar método addOrder do objeto do tipo Orders, passando como parâmetro null.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar order	order	order = null	IllegalArgumentException	PASS

addOrderECP01

Test Case ID	TC#testAddOrderECP01	Dependências	
---------------------	----------------------	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
05 janeiro	19:15h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado: um objeto do tipo Orders, dois objetos do tipo Entity e um objeto do tipo Order.				
Procedimentos				
Para ser adicionado uma Order, é preciso passar um parâmetro do tipo Order.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar order	order	order = order	true	PASS

getOrderShippingCostBVA01

Test Case ID	TC#testGetOrderShippingCostBVA01	Dependências	
---------------------	----------------------------------	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
14 janeiro	15:45h	Fábio	PASS	

Pré-condições / estado inicial				
Terá de ser instanciado, dois objetos do tipo Entity. Terá de ser instanciada pelo menos um objeto do tipo Order, e outro do tipo Product.				
Procedimentos				
Invocar método getOrderShippingCost passando <i>null</i> como <i>input</i> .				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter o custo de envio de uma encomenda.	Order	order = null	IllegalArgumentException	PASS

getOrderShippingCostECP01

Test Case ID	TC#testGetOrderShippingCostECP01	Dependências	
---------------------	----------------------------------	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
14 janeiro	15:55h	Fábio	PASS	

Pré-condições / estado inicial				
Terá de ser instanciado, dois objetos do tipo Entity. Terá de ser instanciado pelo menos um objeto do tipo Order, e outro do tipo Product. Terá de ser inserido pelo menos um Product na class Order instanciada.				
Procedimentos				
Invocar método getOrderShippingCost passando o objeto do tipo Order como <i>input</i> .				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestado
Obter o custo de envio de uma encomenda.	Order	order = order*	16.91	PASS

*

Produto adicionado:

`IProduct product = new Product (1, "Product1", "Description, 2, 1, 4.1, 5.1);`

Como a senderEntity tem como district Viana do Castelo e a receiverEntity em como district Braga, a distancia entre as duas é igual a 60km.

Assim sendo o calculo é o seguinte:

$(\text{weight} + \text{distance}) * 0.25 + (0.15 * \text{volume}) = (5.2 + 60) * 0.25 + (0.15 * 4.1) = 16.91\text{€}$

getOrderShippingCostECP02

Test Case ID	TC#testGetOrderShippingCostECP02	Dependências	
---------------------	----------------------------------	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
14 janeiro	15:55h	Fábio	PASS	

Pré-condições / estado inicial				
Terá de ser instanciado, dois objetos do tipo Entity, que tenham o mesmo valor de district. Terá de ser instanciado pelo menos um objeto do tipo Order, e outro do tipo Product. Terá de ser inserido pelo menos um Product na class Order instanciada.				
Procedimentos				
Invocar método getOrderShippingCost passando o objeto do tipo Order como <i>input</i> , como o sender e o receiver da order são do mesmo distrito o valor do custo de envio deverá ser igual a 0.				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter o custo de envio de uma encomenda.	Order	order = order	0	PASS

testGetOrderIndexBVA01

Test Case ID	TC# testGetOrderIndexBVA01	Dependências	
--------------	-------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
19 janeiro	22:33h	Hugo	PASS	

Pré-condições / estado inicial				
Objeto do tipo Orders instanciado;				
Procedimentos				
- Invocar método <u>getOrderIndex</u> ;				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter encomenda através de um índice	Int index	-1	IndexOutOfBoundsException	PASS

testGetOrderIndexBVA02

Test Case ID	TC# testGetOrderIndexBVA02	Dependências	
--------------	-------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
16 janeiro	21:19h	Hugo	PASS	

Pré-condições / estado inicial				
Objeto do tipo Orders instanciado;				
Procedimentos				
- Invocar método <u>getOrderIndex</u> ;				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter encomenda através de um índice	Int index	ordersByDistrictBeja.size()	IndexOutOfBoundsException	PASS

testGetOrderIndexECP01

Test Case ID	TC# testGetOrderIndexECP01	Dependências	
--------------	-------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
19 janeiro	21:19h	Hugo	PASS	

Pré-condições / estado inicial				
Objeto do tipo Orders instanciado; Objeto do tipo Order instanciado; Objeto do tipo Product instanciado;				
Procedimentos				
- Invocar método <u>getOrderIndex</u> ;				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter encomenda através de um indice	Int index	0	Instance of Order	PASS

Detalhes Class DistanceTable

getDistrictDistanceBVA01

Test Case ID	TC#testDistrictDistanceBVA01	Dependências	
--------------	------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
13 janeiro	19:05h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado uma DistanceTable válida.				
Procedimentos				
Invocar o método getDistrictDistance passando null como input.				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Input	Resultados Esperados	Pass/Fail/Untested
Obter distância entre <i>Districts</i>	districtDistance	null	IllegalArgumentException	PASS

getDistrictDistanceECP01

Test Case ID	TC#testGetDistrictDistanceECP01	Dependências	
--------------	---------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
13 janeiro	19:10h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado : um objeto do tipo DistanceTable. Tem de ser instanciada um DistrictDistance. Tem de ser inserido pelo menos um DistrictDistance na DistanceTable.				
Procedimentos				
Invocar método getDistrictDistance passando como input um DistrictDistance que existe no DistanceTable.				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter distância entre <i>Districts</i>	districtDistance	districtDistance = basic districtDistance	InstanceOf DistrictDistance	PASS

getDistrictDistanceECP02

Test Case ID	TC#testGetDistrictDistanceECP02	Dependências	
--------------	---------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
13 janeiro	19:15h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado : um objeto do tipo DistanceTable. Tem de ser instanciada um DistrictDistance.				
Procedimentos				
Invocar método getDistrictDistance passando como input um DistrictDistance, que neste caso, não deverá existir no DistanceTable.				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestad
Obter distância entre Districts	districtDistance	districtDistance = basic districtDistance	null	PASS

addDistrictDistanceBVA01

Test Case ID	TC#testAddDistrictDistanceBVA01	Dependências	
--------------	---------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
13 janeiro	19:20h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado : um objeto do tipo DistanceTable.				
Procedimentos				
Invocar método addDistrictDistance passando como input um DistrictDistance, passando o parâmetro como null.				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter distância entre Districts	districtDistance	districtDistance = null	IllegalArgumentException	PASS

addDistrictDistanceBVA01

Test Case ID	TC#testAddDistrictDistanceBVA01	Dependências	
--------------	---------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
13 janeiro	19:20h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado : um objeto do tipo DistanceTable.				
Procedimentos				
Invocar método addDistrictDistance passando como input um DistrictDistance, passando o parâmetro como null.				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter distância entre Districts	districtDistance	districtDistance = null	IllegalArgumentException	PASS

getDistanceValueFromShippingBVA01

Test Case ID	TC#testGetDistanceValueFromShippingBVA01	Dependências	
--------------	--	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
13 janeiro	19:55h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado : um objeto do tipo DistanceTable, e pelo menos uma Entity válida.				
Procedimentos				
Para invocar o método getDistanceValueFromShipping, é necessário passar dois parâmetros do tipo Entity, uma entidade para o envio e outra para a receção. Neste caso, a senderEntity, deverá ser passada como null, e a receiverEntity será enviado um input válido.				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter distância entre Districts	sender, receiver	sender = null, receiver = receiverEntity	IllegalArgumentException	PASS

getDistanceValueFromShippingBVA02

Test Case ID	TC#testGetDistanceValueFromShippingBVA02	Dependências	
--------------	--	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
13 janeiro	19:58h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado : um objeto do tipo DistanceTable, e pelo menos uma Entity válida.				
Procedimentos				
<p>Para invocar o método getDistanceValueFromShipping, é necessario passar dois parâmetros do tipo Entity, uma entidade para o envio e outra para a receção.</p> <p>Neste caso, a senderEntity será enviado um input válido, e a receiverEntity deverá ser passada como null.</p>				
Requisito/ Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter distância entre Districts	sender, receiver	sender = senderEntity, receiver = null	IllegalArgumentException	PASS

getDistanceValueFromShippingECP01

Test Case ID	TC#testGetDistanceValueFromShippingECP01	Dependências	
---------------------	--	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
13 janeiro	20:03h	Fábio	PASS	

Pré-condições / estado inicial				
<p>Tem de ser instanciado : um objeto do tipo DistanceTable, e pelo menos dois objetos do tipo Entity válidos.</p> <p>Deverá ser importado o ficheiro <i>Json</i>, correspondente às distâncias entre distritos, para o Objeto do tipo DistanceTable instanciado anteriormente.</p>				
Procedimentos				
<p>Invocar o método getDistanceValueFromShipping, passando dois parâmetros do tipo Entity válidos, que neste caso têm como valor de <i>District</i> Porto, para o <i>sender</i>, e Braga, para o <i>receiver</i>. Como a distância, segundo a tabela de distâncias, de ambos os distritos é de 55km, é de esperar que o metodo retorne esse mesmo valor.</p>				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter distância entre <i>Districts</i>	sender, receiver	sender = senderEntity, receiver = receiverEntity	55	PASS

getDistanceValueFromShippingECP02

Test Case ID	TC#testGetDistanceValueFromShippingECP02	Dependências	
--------------	--	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
13 janeiro	20:08h	Fábio	PASS	

Pré-condições / estado inicial				
<p>Tem de ser instanciado : um objeto do tipo DistanceTable, e pelo menos dois objetos do tipo Entity válidos.</p> <p>Deverá ser importado o ficheiro <i>Json</i>, correspondente às distâncias entre distritos, no entanto não deverá existir distância entre os dois distritos passados como parâmetro.</p>				
Procedimentos				
<p>Invocar o método getDistanceValueFromShipping, passando dois parâmetros do tipo Entity válidos. Assim sendo é de esperar que o metodo retorne o valor -1, dado que não vai encontrar a distância.</p>				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter distância entre <i>Districts</i>	sender, receiver	sender, receiver	-1	PASS

Detalhes Class DistrictDistance

DistrictDistanceBVA01

Test Case ID	TC#DistrictDistanceBVA01	Dependências	
--------------	--------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12 janeiro	15:05h	Fábio	PASS	

Pré-condições / estado inicial				
Procedimentos				
Invocar construtor de DistrictDistance, passar o parâmetro <i>name</i> , que é do tipo String, como vazio.				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter distância entre <i>Districts</i>	name	name = " "	IllegalArgumentException	PASS

DistrictDistanceBVA02

Test Case ID	TC#DistrictDistanceBVA02	Dependências	
--------------	--------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12 janeiro	15:35h	Fábio	PASS	

Pré-condições / estado inicial				
Procedimentos				
Invocar construtor de DistrictDistance, passar o parâmetro null				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter distância entre <i>Districts</i>	name	null	IllegalArgumentException	PASS

DistrictDistanceECP01

Test Case ID	TC#DistrictDistanceECP01	Dependências	
--------------	--------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12 janeiro	15:07h	Fábio	PASS	

Pré-condições / estado inicial				
Procedimentos				
Invocar mais uma vez o construtor de DistrictDistance, passar o parâmetro <i>name</i> , que é do tipo String, desta vez com um input válido.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter distância entre <i>Districts</i>	name	name = "Porto"	Instanceof DistrictDistance	PASS

getDistanceBVA01

Test Case ID	TC#getDistanceBVA01	Dependências	
--------------	---------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12 janeiro	15:50h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado um objeto do tipo DistrictDistance.				
Procedimentos				
Invocar método getDistance do objeto DistrictDistance (passando null como parâmetro).				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter distância entre <i>Districts</i>	distance	distance = null	IllegalArgumentException	PASS

getDistanceECP01

Test Case ID	TC#getDistanceECP01	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12 janeiro	15:55h	Fábio	PASS	
Pré-condições / estado inicial				
Tem de ser instanciado um objeto do tipo DistrictDistance. Terá de existir pelo menos uma Distance já adicionada ao DistrictDistance.				
Procedimentos				
Invocar método getDistance do objeto DistrictDistance, neste caso o input deverá corresponder a uma Distance existente no DistrictDistance.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter distância entre <i>Districts</i>	distance	distance = basicDistance	Instanceof Distance	PASS

DistrictDistanceECP01

Test Case ID	TC#DistrictDistanceECP01	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12 janeiro	15:07h	Fábio	PASS	
Pré-condições / estado inicial				
Procedimentos				
Invocar mais uma vez o construtor de DistrictDistance, passar o parâmetro <i>name</i> , que é do tipo String, desta vez com um input válido.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter distância entre <i>Districts</i>	name	name = "Porto"	Instanceof DistrictDistance	PASS

getDistanceECP02

Test Case ID	TC#getDistanceECP02	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12 janeiro	15:56h	Fábio	PASS	
Pré-condições / estado inicial				
Tem de ser instanciado um objeto do tipo DistrictDistance.				
Procedimentos				
Invocar método getDistance do objeto DistrictDistance, nesta situação o <i>input</i> não deverá existir no DistrictDistance.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter distância entre <i>Districts</i>	distance	distance = basicDistance	null	PASS

addDistanceBVA01

Test Case ID	TC#addDistanceBVA01	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12 janeiro	14:10h	Fábio	PASS	
Pré-condições / estado inicial				
Tem de instanciado um objeto do tipo DistrictDistance.				
Procedimentos				
Adicionar uma Distance ao DistrictDistance, passando um parâmetro do tipo null.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestead
Obter distância entre <i>Districts</i>	distance	distance = null	Throw IllegalArgumentExcepti on	PASS

addDistanceECP01

Test Case ID	TC#addDistanceECP01	Dependências	
--------------	---------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12 janeiro	14:15h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de instanciado um objeto do tipo DistrictDistance. Tem de ser instanciado tambem um objeto do tipo Distance.				
Procedimentos				
Adicionar uma Distance ao DistrictDistance, passando um parâmetro do tipo Distance, válido, e que não seja igual a nenhum já existente no DistanceDistrict, caso tenha alguma Distance adicionada.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter distância entre <i>Districts</i>	distance	distance = basicDistance	true	PASS

addDistanceECP02

Test Case ID	TC#addDistanceECP02	Dependências	
--------------	---------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12 janeiro	14:20h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de instanciado um objeto do tipo DistrictDistance. Tem de ser instanciado tambem um objeto do tipo Distance, este deverá também ser adicionado ao DistrictDistance.				
Procedimentos				
Adicionar uma Distance ao DistrictDistance, passando um parâmetro do tipo Distance, válido, e que neste caso deverá ser o mesmo que já foi adicionado ao DistrictDistance anteriormente.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter distância entre <i>Districts</i>	distance	distance = basicDistance	false	PASS

Detalhes Class Distance

DistanceBVA01

Test Case ID	TC#testDistanceBVA01	Dependências	
--------------	----------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12 janeiro	14:40h	Fábio	PASS	

Pré-condições / estado inicial				
Procedimentos				
O construtor Distance recebe dois parâmetros: uma String e um inteiro, nesta situação é passado o parâmetro <i>distanceValue</i> com o valor de -10, e o parâmetro é passado com input válido.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter distância entre <i>Districts</i>	id, distanceValue	id = "Porto", distanceValue = -10	IllegalArgumentException	PASS

DistanceBVA02

Test Case ID	TC#testDistanceBVA02	Dependências	
--------------	----------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12 janeiro	14:42h	Fábio	PASS	

Pré-condições / estado inicial				
Procedimentos				
O construtor Distance recebe dois parâmetros: uma String e um inteiro, sendo que a String deve ser passada vazia.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter distância entre <i>Districts</i>	id, distanceValue	id = "", distanceValue = 30	IllegalArgumentException	PASS

DistanceECP01

Test Case ID	TC#testDistanceECP01	Dependências	
---------------------	----------------------	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12 janeiro	14:45h	Fábio	PASS	

Pré-condições / estado inicial

Procedimentos

O construtor Distance recebe dois parâmetros: uma String e um inteiro, neste caso serão passados para todos os parâmetros inputs válidos.

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter distância entre <i>Districts</i>	id, distanceValue	id = "Porto", distanceValue = 30	Instanceof Distance	PASS

Detalhes Class ImportConfig

testImportConfigCapacityBVA01

Test Case ID	TC#testImportConfigCapacityBVA01	Dependências	
--------------	----------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
20 janeiro	14:45h	Hugo	PASS	

Pré-condições / estado inicial				
Instanciar objeto do tipo ImportConfigsTruck;				
Procedimentos				
O método recebe um parâmetro: String				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter distância entre <i>Districts</i>	String path	null	IllegalArgumentException	PASS

testImportConfigCapacityBVA02

Test Case ID	TC#testImportConfigCapacityBVA01	Dependências	
--------------	----------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
20 janeiro	14:55h	Hugo	PASS	

Pré-condições / estado inicial				
Instanciar objeto do tipo ImportConfigsTruck;				
Procedimentos				
O método recebe um parâmetro: String				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter distância entre <i>Districts</i>	String path	""	IllegalArgumentException	PASS

testImportConfigCapacityECP01

Test Case ID	TC#testImportConfigCapacityECP01	Dependências	
--------------	----------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
20 janeiro	14:55h	Hugo	PASS	

Pré-condições / estado inicial				
Instanciar objeto do tipo ImportConfigsTruck;				
Procedimentos				
O método recebe um parâmetro: String				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestado
Obter distância entre <i>Districts</i>	String path	"naoexite.json"	63	PASS

3. Summary Tests

Tests cases planeados	Tests Cases executados	TC que passaram	TC que falharam	TC não testados
139	139	139	0	0