

Engenharia de Software 2 Trabalho pratico 2

Módulo de Custo de Envio / Transações /Expedição

Version <1.4> <20/01/2022>

Realizado por: 8180378-Hugo Silva 8200306-Fábio Costa

Histórico de Versões

Version	Implemented	Revision	Approved	Approval	Reason
#	Ву	Date	Ву	Date	
1.0	HUGO	<2/1/2022>	<hugo, fabio=""></hugo,>	<2/1/2022>	Aprovação estrutura
1.1	HUGO	<4/1/2022>	<hugo,fabio></hugo,fabio>	<4/1/2022>	Detalhes TestStatisticsE xporter
1.2	HUGO	<6/1/2022>	<hugo, fabio=""></hugo,>	<6/1/2022>	Detalhes OrderMetricSta tistics
1.3	Fábio	<14/1/2022>	<hugo, fabio=""></hugo,>	<18/1/2022>	Detalhes Class DistanceTable,
1.4	HUGO	<20/1/2022>	<hugo, fabio=""></hugo,>	<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 Class Orders	9
	testRegisterOrdersInLedgerBVA01	9
	testRegisterOrdersInLedgerBVA02	9
	testRegisterOrdersInLedgerBVA03	. 10
	testRegisterOrdersInLedgerECP01	. 11
	testRegisterOrdersInLedgerECP02	. 12
D	etalhes TestStatistExporter	13
	testCreateFileBVA01	. 13
	testCreateFileBVA02	. 13
	testCreateFileECP01	. 14
	testWriteToFileBVA01	. 14
	testWriteToFileBVA02	. 15
	testWriteToFileBVA03	. 15
	TestWriteToFileECP01	. 16
D	etalhes OrderMetricStatistics	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
D	etalhes Product	32
	testBasicProductBVA08	. 32
	testBasicProductBVA06	. 33
	testBasicProductBVA09	. 33
	testBasicProductBVA10	. 34
D	etalhes 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
D	etalhes ProductsInsideTruck	45
	testProductsInsideTruckBVA01	. 45
	testProductsInsideTruckBVA02	. 45
	testProductsInsideTruckBVA03	. 46
	testProductsInsideTruckBVA04	. 46
	testProductsInsideTruckBVA05	. 47
	testProductsInsideTruckBVA06	. 47

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

D	etalhes 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
D	etalhes Orders	. 84
	getOrderBVA01	84
	GetOrderECP01	84
	GetOrderECP02	85
	addOrderBVA01	85
	addOrderECP01	86
	getOrderShippingCostBVA01	86
	getOrderShippingCostECP01	87
	getOrderShippingCostECP02	88
	testGetOrderIndexBVA01	89
	testGetOrderIndexBVA02	89
	testGetOrderIndexECP01	90
D	etalhes 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#	Dependência
	testRegisterOrdersInLedgerBVA	s
	01	

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

Pré-condições /	Pré-condições / estado inicial						
Tem de ser insta	Tem de ser instanciado 1 objeto do tipo Order						
Procedimentos							
-Invocar métod	o registerOrdersInLedger						
Requisito/Use Inputs Valores do Resultado Pass/Fail/Unto							
Case/		Inputs	s	d			
Funcionalidad	Funcionalidad Esperados						
е							
Registar	Não tem	Não tem	0	PASS			
encomendas							
no <i>ledger</i>							

test Register Orders In Ledger BVA 02

Test Case ID	TC#	Dependência
	testRegisterOrdersInLedgerBVA	S
	01	

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

- -Adicionar 1 token à entidade;
- Criar uma order com a entidade como sender;
- Adicionar produtos à order;
- -Adicionar order à lista de orders para registo;
- -Invocar método registerOrdensInLedger;

Dogwieite /Llee	Immunita	Volence de	Desultada	Dece/Feil/Untesta
Requisito/Use	Inputs	Valores do	Resultado	Pass/Fail/Unteste
Case/		Inputs	S	d
Funcionalidad		•	Ecnorados	
Funcionalidad			Esperados	
е				

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

testRegisterOrdersInLedgerBVA03

Test Case ID	TC#	Dependência
	testRegisterOrdersInLedgerBVA	S
	01	

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/	Inputs	Valores do Inputs	Resultado s	Pass/Fail/Unteste d
Funcionalidad e			Esperados	
Registar 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);

test Register Orders In Ledger ECP 01

Test Case ID TC#		Dependência	
	testRegisterOrdersInLedgerBVA	S	
	01		

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/ Funcionalidad e	Inputs	Valores do Inputs	Resultado s Esperados	Pass/Fail/Unteste d
Registar encomendas no <i>ledger</i>	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);

test Register Orders In Ledger ECP 02

Test Case ID	TC#	Dependência	
	testRegisterOrdersInLedgerBVA	s	
	02		

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

- -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/ Funcionalidad e	Inputs	Valores do Inputs	Resultado s Esperados	Pass/Fail/Unteste d
Registar encomendas no <i>ledger</i>	Não tem	Tabela auxiliar 8	0	PASS

Detalhes TestStatistExporter

testCreateFileBVA01

Test Case ID	TC#	Dependência
	testCreateFileBVA0	s
	2	

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

Pré-condições /	Pré-condições / estado inicial				
Tem de ser insta	Tem de ser instanciado 1 objeto do tipo StatisticExporter				
Procedimentos	Procedimentos				
-Invocar método	-Invocar método createFile.				
Requisito/Use	Inputs	Valores do	Resultados Esperados	Pass/Fail/Unteste	
Case/		Inputs		d	
Funcionalidad					
е					
Criar ficheiro	String path	null	IllegalArgumentExcepti	PASS	
			on		

testCreateFileBVA02

Test Case ID	TC#	Dependência	
	testCreateFileBVA0	s	
	2		

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

Pré-condições /	Pré-condições / estado inicial				
Tem de ser insta	Tem de ser instanciado 1 objeto do tipo StatisticExporter				
Procedimentos	Procedimentos				
-Invocar método	-Invocar método createFile.				
Requisito/Use	Inputs	Valores do	Resultados Esperados	Pass/Fail/Unteste	
Case/		Inputs		d	
Funcionalidad					
е					
Criar ficheiro	String path	"" (empty)	IllegalArgumentExcepti	PASS	
			on		

testCreateFileECP01

Test Case ID	TC#	Dependências	
	testCreateFileECP		
	01		

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

Pré-condições	Pré-condições / estado inicial				
Tem de ser inst	Tem de ser instanciado 1 objeto do tipo StatisticExporter				
Procedimentos	Procedimentos				
-Invocar métod	-Invocar método createFile.				
Requisito/Us Inputs		Valores do Inputs	Resultad	Pass/Fail/Untest	
e Case/			os	ed	
Funcionalida			Esperado		
de			S		
Criar ficheiro String path "filesJson/To					
Criar ficheiro	String path	"filesJson/TestCreateFile/test.js	true	PASS	

testWriteToFileBVA01

Test Case ID	TC#	Dependência
	testWriteToFileBVA	s
	01	

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

Pré-condições /	Pré-condições / estado inicial				
Tem de ser insta	Tem de ser instanciado 1 objeto do tipo StatisticExporter				
Procedimentos					
-Invocar métod	-Invocar método writeFile.				
Requisito/Use	Inputs	Valores do	Resultados Esperados	Pass/Fail/Untest	
Case/		Inputs		ed	
Funcionalidad					
е					
Criar ficheiro	String, JsonObject	null,	IllegalArgumentExcepti	PASS	
	-	jsonObject	on		

testWriteToFileBVA02

Test Case ID	TC#	Dependência
	testWriteToFileBVA	s
	01	

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

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

testWriteToFileBVA03

Test Case	TC#	Dependências	
ID	testWriteToFileB		
	VA01		

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

Pré-condiçõe	ses / estado inicial				
Tem de ser ir	Tem de ser instanciado 1 objeto do tipo StatisticExporter				
Procediment	os				
-Invocar mét	odo writeFile.				
Requisito/	Inputs	Valores do Inputs	Resultados	Pass/Fail/Unt	
Use Case/			Esperados	ested	
Funcionalid					
ade					
Criar	String,	("filesJson/TestCreateFile/t	IllegalArgumentExc	PASS	
ficheiro	JsonObject	est.json", null)	eption		

TestWriteToFileECP01

Test Case ID	TC#	Dependências	
	testWriteToFileBV A01		

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

Duá sandisãos	Pré-condições / estado inicial				
Pre-condições	/ estado iniciai				
Tem de ser ins	tanciado 1 objeto do tip	oo StatisticExporter			
Procedimento	Procedimentos				
-Invocar métod	do writeFile.				
Requisito/Us	Inputs	Valores do Inputs	Resultad	Pass/Fail/Untest	
e Case/			os	ed	
Funcionalida			Esperado		
de			S		
Criar ficheiro	String, JsonObject	("filesJson/TestCreateFile/test.j	true	PASS	
	,	son", jsonObject)			

Detalhes OrderMetricStatistics

test Average Number Products Transaction BVA 01

Test Case	TC#	Dependên	
ID	testAverageNumberProductsTransac	cias	
	tionBVA01		

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

Pré-condiçõe	es / estado inicial				
Tem de ser in	em de ser instanciado : Objeto do tipo Basic Organization				
Procediment	os				
-Invocar mét	odo averageNumberProductsTransactio	n			
Requisito/	/ Inputs Valores do Resultados Pass/Fail/U				
Use Case/		Inputs	Esperados	sted	
Funcionalid					
ade					
Obter	Não tem	Não tem	AverageExcep	PASS	
média de			tion		
produtos					
por					
transação				ļ	

test Average Number Products Transaction ECP 01

Test Case ID	TC#	Dependênc
	testAverageNumberProductsTransactio	ias
	nECP01	

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

- -Adicionar produtos à order;
- -Adicionar encomendas ao objeto do tipo Orders;
- -Registar encomendas no ledger;
- -Invocar método <u>AverageNumberProductsTransaction;</u>

Requisito/U se Case/ Funcionalida de	Inputs	Valores do Inputs	Resultad os Esperado s	Pass/Fail/Untes ted
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

test Average Value Transactions BVA 01

Test Case	TC#	Dependên	
ID	testAverageNumberProductsTransac	cias	
	tionBVA01		

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

Pré-condiçõe	s / estado inicial				
Tem de ser in	Tem de ser instanciado : Objeto do tipo Orders				
Procedimentos					
-Invocar mét	odo averageValueTransactions;				
Requisito/	Inputs	Valores do	Resultados	Pass/Fail/Unte	
Use Case/		Inputs	Esperados	sted	
Funcionalid					
ade					
Obter valor	Não tem		AverageExcep	PASS	
médio das			tion		
transações					

test Average Value Transactions ECP 01

Ī	Test Case ID	TC#	Dependênc
		testAverageNumberProductsTransactio	ias
		nBVA01	

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

- -Adicionar produtos à order;
- -Adicionar encomendas ao objeto do tipo Orders;
- -Registar encomendas no ledger;
- -Invocar método <u>averageValueTransactions;</u>

Requisito/U se Case/ Funcionalida de	Inputs	Valores do Inputs	Resultad os Esperad os	Pass/Fail/Untes ted
Obter valor médio das transações	3 objetos do tipo Order	Valor das transações: T1 = 12 T2 = 6	9	PASS

test Average Value Sales Purchases District BVA 01

Test Case ID	TC#	Dependênc
	testAverageValueSalesPurchases	ias
	District BVA01	

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

Pré-condições	Pré-condições / estado inicial				
Tem de ser ins	Tem de ser instanciado : Objeto do tipo Orders				
Procedimento	Procedimentos				
-Invocar méto	odo averageValueTransactions;				
Requisito/U	Inputs Valores do Resultados Pass/Fail/Untes				
se Case/		Inputs	Esperados	ted	
Funcionalida					
de					
Obter média	Não tem	Não tem	AverageExcept	PASS	
de compras			ion		
e vendas por					
distrito					

test Average Value Sales Purchases District BVA02

Test Case ID	TC#	Dependênc	
	testAverageValuePurchases	ias	
	District BVA01		

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

Pré-condições	s / estado inicial			
Tem de ser ins	stanciado : Objeto do tipo Orders	5		
Procedimento	os			
-Invocar méto	odo averageValueTransactions;			
Requisito/U	Inputs	Valores do	Resultados	Pass/Fail/Unte
se Case/		Inputs	Esperados	sted
Funcionalid				
ade				
Obter média	Não tem	null	IllegalArgumentExce	PASS
de compras			ption	
e vendas				
por distrito				

test Average Value Sales Purchases District ECP 01

	Test Case ID	TC#	Dependênci
l		testAverageValuePurchasesDistrictE	as
		CP01	

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 *ledger*; (Repetir passo anterior e este 10000 vezes)
- -Invocar método averageValueSalesPurchasesDistrict;
- Comparar map criado com map retornado pelo método;

Requisito/Us e Case/ Funcionalida de	Inputs	Valores do Inputs	Resultad os Esperado s	Pass/Fail/Untest ed
Obter média de compras e vendas por distrito	Não tem	(Ver tabela auxliar 6)	Map (tabela auxliar 6)	PASS

test Average Value Sales Purchases District ECP 02

Test Case ID	TC#	Dependênci	
	testAverageValuePurchasesDistrictE	as	
	CP02		

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 ledger; (Repetir passo anterior e este 10000 vezes)
- -Invocar método <u>averageValueSalesPurchasesDistrict;</u>
- Comparar map esperado com map retornado pelo método;

Requisito/Us	Inputs	Valores do	Resultad	Pass/Fail/Untest
e Case/		Inputs	os	ed

Funcionalida			Esperado	
de			S	
Obter média	Não tem	(Ver tabela	map	PASS
de compras e		auxliar 6)		
vendas por				
distrito				

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}}
```

test Average Value Sales Purchases District ECPO3

Test Case ID	TC#	Dependênci
	testAverageValuePurchasesDistrictE	as
	CP02	

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 ledger;
- -Invocar método <u>averageValueSalesPurchasesDistrict;</u>
- Comparar map esperado com map retornado pelo método;

Requisito/Us e Case/ Funcionalida	Inputs	Valores do Inputs	Resultad os Esperado	Pass/Fail/Untest ed
de			S	
Obter média	3 objetos do tipo Order com produtos	Ver tabela 6	Map*	PASS
de compras e	adicionados			
vendas por				
distrito				

^{* {}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}}

test Number Orders By District BVA 01

Test Case ID	TC#	Dependên
	testNumberOrdersByDistric	cias
	tBVA01	

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 insta	nciado: 1 objeto do tipo Ordersl	MetricStatistic	S	
Procedimentos				
-Invocar método	o <u>numberOrderSentReceivedB</u>	yDistrict;		
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/Unte
Case/		Inputs	Esperados	sted
Funcionalidad				
е				
Obter número	ArrayList <block></block>	null	IllegalArgumentExc	PASS
de			eption	
encomendas			-	
enviadas/rece				
bidas por				
distrito				

test Number Orders By District ECP 01

Test Case ID	TC#	Dependênci
	testNumberOrdersByDistrictBV	as
	A01	

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.

- Adicionar produtos às encomendas;
- Adicionar encomendas ao objeto do tipo Orders;
- Registar encomendas no ledger;
- Invocar método number Orders By District;

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultado s Esperado s	Pass/Fail/Untest ed
Obter número de encomendas enviadas/recebid as por distrito	ArrayList <block></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}}

test Standard Deviation Number Product Transactions BVA 01

Test Case	TC#	Dependê
ID	testStandardDeviationNumberProduct	ncias
	TransactionsBVA01	

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

Pré-condiçõ	Pré-condições / estado inicial						
Tem de ser	instanciado 1 objeto do tipo OrdersMetricSt	tatistics					
Procedimer	ntos						
-Invocar me	étodo <u>standard Deviation Number Product T</u>	ransactions					
Requisito	Inputs	Valores	Resultados	Pass/Fail/Un			
/Use		do Inputs	Esperados	tested			
Case/							
Funcionali							
dade							
Obter	ArrayList <block></block>	null	IllegalArgumentE	PASS			
desvio			xception				
padrão do							
número							
de							
produtos							

test Standard Deviation Number Product Transactions ECP 01

Test Case	TC#	Dependên	
ID	testStandardDeviationNumberProductTrans	cias	
	actionsECP01		

Data de Execução	Hora de Execução	Tester	Pass/Fai I	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.

- Adicionar produtos às *orders* instanciadas;
- Invocar método de registo de orders no ledger;
- Invocar método <u>standardDeviationNumberProductTransactions;</u>

Requisito/ Use Case/ Funcionalid ade	Inputs	Valores do Inputs	Resulta dos Esperad os	Pass/Fail/Unt ested
Obter desvio padrão do valor médio de produtos por transação	ArrayList <block></block>	Ver tabela auxliar x	0.82	PASS

test Standard Deviation Transactions ECP 01

Test Case ID	TC#	Dependênci	
	testStandardDeviationTransactionsE	as	
	CP01		

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.

- Adicionar produtos às orders instanciadas;
- Invocar método de registo de orders no ledger;
- Invocar método <u>standardDeviationNumberProductTransactions</u>;

Requisito/Us e Case/ Funcionalida de	Inputs	Valores do Inputs	Resultad os Esperado s	Pass/Fail/Untest ed
Obter desvio padrão do valor médio de transações	3 orders inseridas para registo	Ver tabela auxliar 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();
```

test Standard Deviation Transactions BVA01

Test Case	TC#	Dependên
ID	testStandardDeviationTransacti	cias
	onsBVA01	

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

Pré-condiçõe	es / estado inicial			
Tem de ser ir	stanciado 1 objeto do tipo OrdersM	etricStatistics		
Procediment	os			
-Invocar mét	odo standard Deviation Transaction	sBVA01;		
Requisito/	Inputs	Valores do	Resultados	Pass/Fail/Unt
Use Case/		Inputs	Esperados	ested
Funcionalid				
ade				
Obter	Não tem	null	IllegalArgumentExc	PASS
desvio			eption	
padrão do			·	
valor médio				
de				
transações				

Detalhes Product

testBasicProductBVA08

Test Case ID	TC#	Dependênci
	testBasicProductBVA0	as
	8	

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	5			
-Invocar const	rutor <u>Product</u> ;			
Requisito/Us	Inputs	Valores do	Resultados Esperados	Pass/Fail/Untest
e Case/		Inputs		ed
Funcionalida				
de				
Criar um	Int, String , String,	1,	IllegalArgumentExcepti	PASS
produto	Double,Int,Double,Dou	"Product1",	on	
	ble	"Description		
		", 1, 3, 63.1,		

testBasicProductBVA06

Test Case ID	TC#	Dependênci
	testBasicProductBVA0	as
	6	

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 const	rutor <u>Product</u> ;			
Requisito/Us	Inputs	Valores do	Resultados Esperados	Pass/Fail/Untest
e Case/		Inputs		ed
Funcionalida				
de				
Criar um	Int, String , String,	1,	IllegalArgumentExcepti	PASS
produto	Double,Int,Double,Dou	"Product1",	on	
	ble	"Description		
		", 1, 3, 0,		
		5.2l		

testBasicProductBVA09

Test Case ID	TC#	Dependências
	testBasicProductBVA	
	09	

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

Pré-condições	Pré-condições / estado inicial					
Não tem						
Procedimento	s					
-Invocar const	rutor <u>Product</u> ;					
Requisito/Us	Inputs	Valores do	Resultados Esperados	Pass/Fail/Untest		
e Case/		Inputs		ed		
Funcionalida						
de						
Criar um	Int, String , String,	1,	IllegalArgumentExcept	PASS		
produto	Double,Int,Double,Do	null,Descriptio	ion			
	uble	n", 1, 3, 0, 5.2l				

testBasicProductBVA10

Test Case ID	TC#	Dependênci
	testBasicProductBVA1	as
	0	

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

Pré-condições	Pré-condições / estado inicial				
Não tem					
Procedimentos	Procedimentos				
-Invocar const	rutor <u>Product</u> ;				
Requisito/Us	Inputs	Inputs Valores do Resultados Esperados Pass/Fail/Untest			
e Case/		Inputs		ed	
Funcionalida					
de					
Criar um	Int, String , String,	1,	IllegalArgumentExcepti	PASS	
produto	Double,Int,Double,Dou	"Product1",	on		
	ble	null, 1, 3, 0,			
		5.2l			

Detalhes Truck

TestTruckBVA01

Test Case ID	TC#	Dependência	
	testTruckBVA0	S	
	2		

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

Pré-condições /	Pré-condições / estado inicial				
Não tem					
Procedimentos					
-Invocar constru	itor <u>Truck</u> ;				
Requisito/Use	Inputs	Inputs Valores do Resultados Esperados Pass/Fail/Unteste			
Case/		Inputs		d	
Funcionalidad					
е					
Criar um	String	un	IllegalArgumentExceptio	PASS	
camião			n		

TestTruckBVA02

Test Case ID	TC#	Dependência
	testTruckBVA0	S
	<u>2</u>	

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 constru	itor <u>Truck</u> ;				
Requisito/Use	Inputs	Inputs Valores do Resultados Esperados Pass/Fail/Unteste			
Case/		Inputs		d	
Funcionalidad					
е					
Criar um	String	un	IllegalArgumentExceptio	PASS	
camião			n		

TestTruckECP01

Test Case ID	TC# testTruckBVA02	Dependências		
Data da	Hara da Evacueão	Tostor	Docs/Fail	Posumo do Folho

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	Inputs	Valores do	Resultados	Pass/Fail/Untested	
Case/		Inputs	Esperados		
Funcionalidade					
Criar um	String name	"HUGO"	Instanceof	PASS	
camião			Truck (true)		

TestAddAllOrderInTruckBVA01

Test Case ID	TC#	Dependência
	testTruckBVA0	S
	1	

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

Pré-condições / estado inicial						
Objeto do tipo Truck instanciado;						
Procedimentos	Procedimentos					
-Invocar método <u>addAllOrderInTruck</u> ;						
Requisito/Use	Inputs	Valores do	Resultados Esperados	Pass/Fail/Unteste		
Case/		Inputs		d		
Funcionalidad						
е						
Adicionar	IOrder order	null	IllegalArgumentExceptio	PASS		
produto ao			n			
camião						

TestAddAllOrderInTruckBVA02

Test Case ID	TC# testTruckBVA02	Dependências		
			T	1
Data de	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
Execução				
16 janeiro	17:33h	Hugo	PASS	

Pré-condições /	estado inicial			
Objeto do tipo Ti	ruck instanciado;			
Procedimentos				
-Invocar método	addAllOrderInTruck;			
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/Untested
Case/		Inputs	Esperados	
Funcionalidade				
Adicionar	IOrder order	Orde.getVolum()	False	PASS
produto ao		= 63.1		
camião				

TestAddAllOrderInTruckBVA03

Test Case ID	TC#	Dependências
	testTruckBVA0	
	2	

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 T	ruck instanciado;			
Procedimentos				
-Invocar métod	o <u>addAllOrderInTr</u>	uck;		
Requisito/Use	Inputs	Valores do	Resultados Esperados	Pass/Fail/Unteste
Case/		Inputs		d
Funcionalidad				
е				
Adicionar	IOrder order	Orde.getVolum(IllegalArgumentExceptio	PASS
produto ao) = 0	n	
camião				

testAddAllOrderInTruckProductInTruckECP01

Test Case ID	TC#	Dependências
	testAddAllOrderInTruckProductInTru	
	ckECP01	

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;			
Procedimento	s			
-Invocar méto	odo <u>addAllOrderInTruck</u> ;			
Requisito/U	Inputs	Valores do	Resultad	Pass/Fail/Untes
se Case/		Inputs	os	ted
Funcionalida			Esperad	
de			os	
Adicionar	IOrder order	order.getVolu	true	PASS
produto ao		m() = 45		
camião				

testAddProductInTruckBVA01

Test Case ID	TC#	Dependências	
	testAddProductInTruckBVA0		
	1		

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 T	ruck instanciado;			
Procedimentos				
Invocar método	a <u>ddProductInTruck</u> ;			
Requisito/Use	Inputs	Valores do	Resultado	Pass/Fail/Unteste
Case/		Inputs	s	d
Funcionalidad			Esperados	
е				
Adicionar	String idOrder, String	"idpordutor,"23"	false	PASS
produto ao camião	idProduct, Double volume	, 63.1		

testAddProductInTruckBVA02

Test Case ID	TC#	Dependências
	testAddProductInTruckB	
	VA02	

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

Pré-condições	Pré-condições / estado inicial				
Objeto do tipo	Objeto do tipo Truck instanciado;				
Procedimento)S				
-Invocar méto	odo a <u>ddProductInTruck</u> ;				
Requisito/U se Case/ Funcionalid ade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untes ted	
Adicionar produto ao camião	String idOrder, String idProduct, Double volume	"idpordutor"," 23", 0	IllegalArgumentExce ption	PASS	

testAddProductInTruckBVA03

Test Case ID	TC#	Dependênci
	testAddProductInTruckB	as
	VA03	

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

Pré-condições	Pré-condições / estado inicial					
Objeto do tipo	Truck instanciado;					
Procedimento	s					
-Invocar méto	do a <u>ddProductInTruck</u> ;					
Requisito/Us e Case/ Funcionalida de						
Adicionar um produto a um camião	String idOrder, String idProduct, Double volume					

test Add Product In Truck BVA 04

Test Case ID	TC#	Dependênci
	testAddProductInTruckB	as
	VA04	

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

Pré-condições	Pré-condições / estado inicial			
Objeto do tipo	Truck instanciado;			
Procedimento	s			
-Invocar méto	do a <u>ddProductInTruck</u> ;			
Requisito/Us	Inputs Valores do Resultados Esperados Pass/Fail/Untes			
e Case/		Inputs		ted
Funcionalida				
de				
Adicionar um	String idOrder, String	"ws",null, 5	IllegalArgumentExcep	PASS
produto a	idProduct, Double volume		tion	
um camião				

testAddProductInTruckBVA05

Test Case ID	TC#	Dependênci	
	testAddProductInTruckB	as	
	VA05		

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

Pré-condições	Pré-condições / estado inicial			
Objeto do tipo	Truck instanciado;			
Procedimento	s			
-Invocar méto	do a <u>ddProductInTruck</u> ;			
Requisito/Us e Case/ Funcionalida de	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untes ted
Adicionar um produto a um camião	String idOrder, String idProduct, Double volume	<i>"",</i> "34" , 5	IllegalArgumentExcep tion	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 /	Pré-condições / estado inicial			
Objeto do tipo Ti	Objeto do tipo Truck instanciado;			
Procedimentos	Procedimentos			
-Invocar método	a <u>ddProductInTruck</u> ;			
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/Untested
Case/ Inputs Esperados				
Funcionalidade				
Adicionar um	String idOrder, String	"ws","ola", 5	true	PASS
produto a um	idProduct, Double volume	ws, old, s	truc	1733
camião	.a dadd, Dadde Volume			

testGetProductContainerTruckBVA01

Test Case ID	TC#	Dependên	
	testGetProductContainerTruc	cias	
	kBVA01		

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

Pré-condiçõe	Pré-condições / estado inicial			
Objeto do tipo	Objeto do tipo Truck instanciado;			
Procedimento	os			
-Invocar mét	odo <u>getProductContainerTruck</u> ;			
Requisito/U se Case/ Funcionalid ade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Unte sted
Obter produto de um camião	IProductsInsideTruck iProductsInsideTruck	null	IllegalArgumentExce ption	PASS

testGetProductContainerTruckECP01

Test Case	TC#	Dependências	
ID	testGetProductContainerTruc		
	kECP01		

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 camião;
- -Invocar método <u>getProductContainerTruck(passando</u> como parâmetro o produto adicionado no passo anterior);

Requisito/U se Case/ Funcionalid ade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Unte sted
Obter produto de um camião	IProductsInsideTruck iProductsInsideTruck	iProductsInsideT ruck	ProductInsideT ruck	PASS

testGetProductContainerTruckECP02

Test Case ID	TC#	Dependências
	testGetProductContainerTruckE	
	CP01	

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 getProductContainerTruck; Requisito/Us **Valores do Inputs** Resultad Pass/Fail/Untest Inputs e Case/ os ed **Funcionalida** Esperado de iProductsInsideTrnull PASS Obter IP roducts In side Trucki Products In side Truckproduto de uck um camião

testGetProductListProductsInsideTruckBVA01

Test Case	TC#	Dependê	
ID	testGetProductListProductsInsid	ncias	
	eTruckBVA01		

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

Pré-condiçõ	es / estado inicial					
Objeto do ti	po DistrictTrucks instanciado;					
Procedimentos						
-Invocar mé	-Invocar método getProductListProductsInsideTruck;					
Requisito/	Inputs					
Use Case/		do Inputs	Esperados	ested		
Funcionali						
dade						
Obter	Int index	-1	IndexOutOfBoundsE	PASS		
produto de			xception			
um camião			•			
através de						
um indice						

test Get Product List Products Inside Truck BVA02

Test	TC#	Dependências	
Case ID	testGetProductListProductsI		
	nsideTruckBVA02		

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

Pré-condiç	Pré-condições / estado inicial					
Objeto do t	Objeto do tipo DistrictTrucks instanciado;					
Procedime	Procedimentos					
-Invocar m	étodo getProductListProductsIr	nsideTruck;				
Requisito	Inputs	Valores do Inputs	Resultados	Pass/Fail/U		
/Use			Esperados	ntested		
Case/						
Funciona						
lidade						
Obter	Int index	truck.numberProduc	IndexOutOfBound	PASS		
produto		tInContainer	sException			
de um						
camião						
através						
de um						
indice						

TestGetProductListProductsInsideTruckECP01

Test Case	TC#	Dependências	
ID	testGetProductListProductsIns		
	ideTruckECP01		

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

Pré-condiçõ	Pré-condições / estado inicial					
Objeto do t	Objeto do tipo DistrictTrucks instanciado;					
Procedimentos						
-Invocar me	étodo <u>getProductListProductsInsi</u>	deTruck;				
Requisito	Inputs Valores do Inputs Resultados Pass/Fail/Ur					
/Use			Esperados	tested		
Case/						
Funcionali						
dade						
Obter	Int index	truck.numberProductI	instanceof	PASS		
produto		nContainer	ProductsInsid			
de um			eTruck			
camião						
através de						
um indice						

Detalhes ProductsInsideTruck

testProductsInsideTruckBVA01

Test Case ID	TC#	Dependênci	
	testProductsInsideTruckB	as	
	VA01		

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

Pré-condições	Pré-condições / estado inicial				
Procedimento	S				
-Invocar const	trutor <u>ProductsInsideTruck;</u>				
Requisito/Us	Requisito/Us Inputs Valores do Resultados Esperados Pass/Fail/Untes				
e Case/		Inputs		ted	
Funcionalida					
de					
Criar	String idOrder, String	"", "Hugo",	IllegalArgumentExcep	PASS	
produto para	idProduct, int quantity,	4, 2.2	tion		
ser inserido	double uniqueSize				
dentro do					
camião					

testProductsInsideTruckBVA02

Test Case ID	TC#	Dependênci
	testProductsInsideTruckB	as
	VA02	

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

Pré-condições	Pré-condições / estado inicial				
n !: .					
Procedimento	S				
-Invocar const	rutor <u>ProductsInsideTruck;</u>				
Requisito/Us Inputs Valores do Resultados Esperados Pass/Fail/U					
e Case/		Inputs		ted	
Funcionalida		_			
de					
Criar	String idOrder, String	"Hugo", ""	IllegalArgumentExcep	PASS	
produto para	idProduct, int quantity,	,4, 2.2	tion		
ser inserido	double uniqueSize				
dentro do					
camião					

testProductsInsideTruckBVA03

Test Case ID	TC#	Dependênci	
	testProductsInsideTruckB	as	
	VA03		

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

Pré-condições	Pré-condições / estado inicial				
Procedimento	S				
-Invocar const	trutor <u>ProductsInsideTruck;</u>				
Requisito/Us Inputs Valores do Resultados Esperados Pass/Fail/Ur					
e Case/		Inputs		ted	
Funcionalida					
de					
Criar	String idOrder, String	"Hugo", "	IllegalArgumentExcep	PASS	
produto para	idProduct, int quantity,	ID_PRODUC	tion		
ser inserido	double uniqueSize	T" ,0, 2.2			
dentro do	·				
camião					

testProductsInsideTruckBVA04

Test Case ID	TC#	Dependênci	
	testProductsInsideTruckB	as	
	VA04		

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

Pré-condições	Pré-condições / estado inicial				
Procedimento	Procedimentos				
-Invocar const	rutor <u>ProductsInsideTruck;</u>				
Requisito/Us Inputs Valores do Resultados Esperados Pass/Fail/U					
e Case/		Inputs		ted	
Funcionalida					
de					
Criar	String idOrder, String	"Hugo", "	IllegalArgumentExcep	PASS	
produto para	idProduct, int quantity,	ID_PRODUC	tion		
ser inserido	double uniqueSize	T" ,4, 0			
dentro do					
camião					

testProductsInsideTruckBVA05

Test Case ID	TC#	Dependênci	i
	testProductsInsideTruckB	as	
	VA05		

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

Pré-condições	Pré-condições / estado inicial					
Procedimento	Procedimentos					
-Invocar const	trutor <u>ProductsInsideTruck;</u>					
Requisito/Us	Inputs	Valores do	Resultados Esperados	Pass/Fail/Untes		
e Case/		Inputs		ted		
Funcionalida						
de						
Criar	String idOrder, String	"Hugo",	IllegalArgumentExcep	PASS		
produto para	idProduct, int quantity,	null, ,4, 5.4	tion			
ser inserido	double uniqueSize	', '	C.O.I.			
dentro do	'					
camião						

testProductsInsideTruckBVA06

Test Case ID	TC#	Dependênci
	testProductsInsideTruckB	as
	VA05	

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

Pré-condições	/ estado inicial						
Procedimento	Procedimentos						
-Invocar const	trutor <u>ProductsInsideTruck;</u>						
Requisito/Us	Inputs	Valores do	Resultados Esperados	Pass/Fail/Untes			
e Case/		Inputs		ted			
Funcionalida							
de							
Criar	String idOrder, String	null,	IllegalArgumentExcep	PASS			
produto para	idProduct, int quantity,	"Hugo", ,4,	tion				
ser inserido	double uniqueSize	2.3					
dentro do							
camião							

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 constru	itor <u>ProductsInsideTruck;</u>			
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar produto para ser inserido dentro do camião	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#	Dependênci					
	testGetTruckByIndexB	as					
	V01						

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

Pré-condições	Pré-condições / estado inicial						
Objeto do tipo	Objeto do tipo DistrictTrucks instanciado;						
Procedimento	Procedimentos						
-Invocar méto	do getTruckByIndex;						
Requisito/Us e Case/ Funcionalida de	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untes ted			
Obter um camião através de um indice	Int index	-1	IndexOutOfBoundsExcep tion	PASS			

testGetTruckByIndexBVA02

Test Case	TC#	Dependências		
ID	testGetTruckByInde			
	xBV02			

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

Pré-condiçõe	Pré-condições / estado inicial						
Objeto do tip	Objeto do tipo DistrictTrucks instanciado;						
Procediment	Procedimentos						
-Invocar mét	odo getTruckByIndex;						
Requisito/	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Unte			
Use Case/		sted					
Funcionalid							
ade							
Obter um	Int index	ordersByDistrictBeja	IndexOutOfBoundsExc	PASS			
camião		.size()	eption				
através de			·				
um indice							

test Get Truck By Index ECP 01

Test Case ID	TC#	Dependências	
	testGetTruckByIndexECP01		

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

- Adicionar produto à encomenda;
- Adicionar encomenda à lista de encomendas do objeto DistrictTrucks;
- Agrupar encomendas por camiões, invocando método agroupOrders()

- Invocar método getTruckByIndex;

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

Test Case ID	TC#	Dependênci
	testCreateTruckBVA	as
	01	

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 creatTruck;

Requisito/Us e Case/ Funcionalidad e	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untest ed
Criar um conjunto de camiões	Int numberTrucks	0	IllegalArgumentExcepti on	PASS

testCreateTruckECP01

Test Case ID	TC#	Dependências
	testCreateTrucECP01	

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#	Dependência	
	testCreateTrucECP0	s	
	2		

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

Pré-condições /	Pré-condições / estado inicial					
Objeto do tipo [Objeto do tipo DistrictTrucks instanciado;					
Procedimentos						
-Invocar métod	o <u>creatTruck</u> ;					
Requisito/Use	Inputs	Valores do	Resultados Esperados	Pass/Fail/Unteste		
Case/		Inputs		d		
Funcionalidad						
е						
Criar um	Int numberTrucks	-3	IllegalArgumentExcepti	PASS		
conjunto de			on			
camiões						

testDistrictTrucksBVA01

Test Case ID	TC#	Dependênci
	testDistrictTrucksBVA	as
	01	

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

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

testDistrictTrucksBVA02

Test Case ID	TC#	Dependências	
	testDistrictTrucksBV		
	A02		

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

Pré-condições	Pré-condições / estado inicial			
Procedimento	S			
-Invocar méto	do <u>creatTruck</u> ;			
Requisito/Us	Inputs	Valores do	Resultados Esperados	Pass/Fail/Untes
e Case/		Inputs		ted
Funcionalida				
de				
Criar um	String nameDistrict,	"", 5,	IllegalArgumentExcep	PASS
conjunto de	int numberContainer,	ordersByDistrictB	tion	
camiões	ArrayList <iorder></iorder>	eja		
	ordersByDistrict			

testDistrictTrucksBVA03

Test Case ID	TC#	Dependências
	testDistrictTrucksBV	
	A02	

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

Pré-condições	Pré-condições / estado inicial				
Procedimento	S				
-Invocar méto	do <u>creatTruck</u> ;				
Requisito/Us	Inputs	Valores do	Resultados Esperados	Pass/Fail/Untes	
e Case/		Inputs		ted	
Funcionalida					
de					
Criar um	String nameDistrict,	null, 5,	IllegalArgumentExcep	PASS	
conjunto de	int numberContainer,	ordersByDistrictB	tion		
camiões	ArrayList <iorder></iorder>	eja			
	ordersByDistrict				

testDistrictTrucksECP01

Test Case ID	TC#	Dependências	
	TestDistrictTrucksECP0		
	1		

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

Pré-condições /	Pré-condições / estado inicial				
Procedimentos	Procedimentos				
-Invocar métod	o <u>creatTruck</u> ;				
Requisito/Use Case/ Funcionalidad e	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Unteste d	
Criar um conjunto de camiões	String nameDistrict, int numberContainer, ArrayList <iorder> ordersByDistrict</iorder>	"idProduct", 5, ordersByDistrictBej a	Instance of DistrictTruc k (true)	PASS	

Detalhes GroupAllTrucks

testAddDistrictTrucksBVA01

Test Case ID	TC#	Dependênci
	testAddDistrictTrucksBV	as
	A01	

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

Pré-condições	Pré-condições / estado inicial						
Objeto do tipo	DistrictTrucks instanciado;						
Objeto do tipo GroupAllTrucks;							
Procedimento	s						
-Invocar méto	do <u>addDistrictTrucks</u> ;						
Requisito/Us	Inputs	Valores do	Resultados Esperados	Pass/Fail/Untest			
e Case/		Inputs		ed			
Funcionalida							
de							
Adiciona um conjunto de camiões 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 addDistrictTrucks;

-invocar metodo	o addDistrictTrucks;					
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	IDistrictTrucks iDistrictExpedition	iDistrictExpedition	true	PASS		
todos os resultados do agrupamento						

Detalhes TestExpeditionMethods testGetArrayOrdersByDistrictBVA01

Test Case	TC#	Dependênc
ID	testGetArrayOrdersByDistric	ias
	tBVA01	

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;						
os estados esta						
odo getArrayOrdersByDistrict;						
Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Unte sted			
String district, ArrayList <iorder> listOrders</iorder>	"Porto", null	IllegalArgumentExce ption	PASS			
)	D ExpeditionMethods; USS USS USON DESCRIPTION OF THE PROPERTY	D ExpeditionMethods; Use Dodo getArrayOrdersByDistrict; Inputs Valores do Inputs String district, "Porto",	ExpeditionMethods; String district, Porto", IllegalArgumentExce			

test Get Array Orders By District BVA02

Test Case	TC#	Dependênc
ID	testGetArrayOrdersByDistric	ias
	tBVA02	

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

Pré-condiçõe:	Pré-condições / estado inicial						
Objeto do tipo ExpeditionMethods;							
Procedimento	Procedimentos						
-Invocar méte	odo getArrayOrdersByDistrict;						
Requisito/U	Inputs	Valores do	Resultados	Pass/Fail/Unte			
se Case/		Inputs	Esperados	sted			
Funcionalid							
ade							
Obter lista	String district,	"",	IllegalArgumentExce	PASS			
de	ArrayList <iorder> listOrders</iorder>	listOrders	ption				
encomenda							
s de um							
distrito							

test Get Array Orders By District BVA03

Test Case	TC#	Dependênc
ID	testGetArrayOrdersByDistric	ias
	tBVA02	

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

Pré-condições	Pré-condições / estado inicial				
Objeto do tipo	Objeto do tipo ExpeditionMethods intanciado;				
Procedimento	Procedimentos				
-Invocar méto	odo getArrayOrdersByDistrict;				
Requisito/U	Inputs	Valores do	Resultados	Pass/Fail/Unte	
se Case/	Inputs Esperados sted				
Funcionalid					
ade					
Obter lista	String district,	null,	IllegalArgumentExce	PASS	
de	ArrayList <iorder> listOrders</iorder>	listOrders	ption		
encomenda			·		
s de um					
distrito					

test Get Array Orders By District ECP 01

Test Case ID	TC#	Dependência
	testGetArrayOrdersByDistrictECP	S
	01	

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 getArrayOrdersByDistrict, passando como parâmetro a order do passo anterior;

Requisito/Use	Inputs	Valores do	Resultado	Pass/Fail/Unteste
Case/		Inputs	s	d
Funcionalidad			Esperados	
е				
Obter lista de encomendas	String district, ArrayList <iorder> listOrders</iorder>	"Porto", listOrders	true	PASS
de um distrito				

test Group Orders By Trucks BVA 01

Test Case ID	TC#	Dependênc	
	testGroupOrdersByTrucksB	ias	
	VA01		

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

Pré-condições	Pré-condições / estado inicial				
Objeto do tipo	Objeto do tipo ExpeditionMethods intanciado;				
Procedimento	s				
-Invocar méto	odo groupOrdersByTrucks;				
Requisito/U se Case/ Funcionalida de	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untes ted	
Obter lista de encomendas de um distrito	ArrayList <iorder> listOrders</iorder>	null	IllegalArgumentExce ption	PASS	

testGroupOrdersByTrucksECP01

Test Case ID	TC#	Dependência	
	testGroupOrdersByDistrictECP0	s	
	1		

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/ Funcionalidad e	Inputs	Valores do Inputs	Resultado s Esperados	Pass/Fail/Unteste d
Obter lista de encomendas de um distrito	ArrayList <iorder> listOrders Ver tabela xy</iorder>	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);
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": 0.1000000000000142,
    "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,
      "ld_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": 9.99999974752427E-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"
}
]
```

test Group Orders By Trucks ECP 02

Test 0	Case ID	TC#	Dependência	
		testGroupOrdersByDistrictECP	s	
		02		

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 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/ Funcionalidad e	Inputs	Valores do Inputs	Resultado s Esperados	Pass/Fail/Unteste d
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:

[]
```

test Export Group Orders For File Js on ECP 01

Test Case ID	TC#	Dependênci
testExportGroupOrdersForFileJson		as
	ECP01	

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 ExportGroupOrdersForFileJson;

Requisito/Us e Case/ Funcionalida de	Inputs	Valores do Inputs	Resultad os Esperado s	Pass/Fail/Untest ed
Obter lista de encomendas de um distrito	Ver tabela XYW	Ver tabela XYW	3	PASS

Detalhes Product

ProductBVA01

Dependência	Test Case ID TC#ProductBVA0	Test Case ID	
S	1		

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

Pré-condições / estado inicial

Procedimentos

Invocar contrutor 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 id será igual a -1, que é um input invalido, para os outros parametros serão passados *inputs* validos.

Requisito/Use Case/ Funcionalidad e	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Unteste d
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	IllegalArgumentExceptio n	PASS

ProductBVA02

Test Case ID	TC#ProductBVA0	Dependência
	2	s

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

Pré-condições / estado inicial

Procedimentos

Invocar contrutor 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 "", que é um input invalido, para os outros parametros serão passados inputs validos.

Requisito/Use Case/ Funcionalidad e	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Unteste d
Criar Produto	id, name, description, value, quantity, volume, weight	id = 1, name = "", description = "Description", value = 2, quantity = 3, volume = 4.1, weight = 5.2	IllegalArgumentExceptio n	PASS

Test Case ID	A0 Dependência
	s

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

Pré-condições / estado inicial

Procedimentos

Invocar contrutor 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 description será igual a "" , que é um input invalido, para os outros parametros serão passados inputs validos.

Requisito/Use Case/ Funcionalidad e	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Unteste d
Criar Produto	id, name, description, value, quantity, volume, weight	id = 1, name = "Product1", description = "", value = 2, quantity = 3, volume = 4.1, weight = 5.2	IllegalArgumentExceptio n	PASS

Test Case ID TC#ProductBVA0 Dependê
4 s

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

Pré-condições / estado inicial

Procedimentos

Invocar contrutor 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/ Funcionalidad e	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Unteste d
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	IllegalArgumentExceptio n	PASS

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

Pré-condições / estado inicial

Procedimentos

Invocar contrutor 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 quantity será igual a 0, que é um input invalido, para os outros parametros serão passados inputs validos.

Requisito/Use Case/ Funcionalidad e	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Unteste d
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	IllegalArgumentExceptio n	PASS

68

Test Case ID	TC#ProductBVA0	Dependência
	6	S

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

Pré-condições / estado inicial

Procedimentos

Invocar contrutor 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 volume será igual a 0, que é um input invalido, para os outros parametros serão passados inputs validos.

Requisito/Use Case/ Funcionalidad e	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Unteste d
Criar Produto	id, name, description, value, quantity, volume, weight	id = 1, name = "Product1", description = "Description", value = 1, quantity = 3, volume = 0, weight = 5.2	IllegalArgumentExceptio n	PASS

Test Case ID	TC#ProductBVA0	Dependência
	7	S

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

Pré-condições / estado inicial

Procedimentos

Invocar contrutor 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 weight será igual a 0, que é um input invalido, para os outros parametros serão passados inputs validos.

Requisito/Use	Inputs	Valores do	Resultados Esperados	Pass/Fail/Unteste
Case/		Inputs	·	d
Funcionalidad				
е				
Criar Produto	id, name, description, value, quantity, volume, weight	id = 1, name = "Product1", description = "Description", value = 1, quantity = 3, volume = 4.1, weight = 0	IllegalArgumentExceptio n	PASS

Test Case ID	TC#ProductBVA0	Dependência
	8	s

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

Pré-condições / estado inicial

Procedimentos

Invocar contrutor 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 volume será igual a 63.1, que é um input invalido, para os outros parâmetros serão passados inputs validos.

Requisito/Use Case/ Funcionalidad e	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Unteste d
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	IllegalArgumentExceptio n	PASS

Test Case ID	TC#ProductBVA0 De	ependência
	9 s	

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

Pré-condições / estado inicial

Procedimentos

Invocar contrutor 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 *description* será igual a *null*, pois é um input invalido, para os outros parâmetros serão passados inputs validos.

Requisito/Use Case/ Funcionalidad e	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Unteste d
Criar Produto	id, name, description, value, quantity, volume, weight	id = 1, name = "Product1", description = null, value = 1, quantity = 3, volume = 4.1, weight = 5.2	IllegalArgumentExceptio n	PASS

ProductBVA10

Test Case ID	TC#ProductBVA1	Dependência
	0	s

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

Pré-condições / estado inicial

Procedimentos

Invocar contrutor 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/ Funcionalidad e	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Unteste d
Criar Produto	id, name, description, value, quantity, volume, weight	id = 1, name = null, description = "Description" , value = 1, quantity = 3, volume = 4.1, weight = 5.2	IllegalArgumentExcepti on	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 *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*.

Requisito/Use Case/	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested		
Funcionalidade						
runcionalidade						
Criar Produto	id,	id = 1,	Instanceof	PASS		
	name, description,	name =	Product			
	value, quantity,	"Product1",				
	volume, weight	description =				
	voidine, weight	"Description",				
		value = 1,				
		quantity = 3,				
		volume = 4.1,				
		weight = 5.2				

Detalhes Class Order

OrderBVA01

Test Case ID	TC#OrderBVA0	Dependência
	1	S

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 id será igual a "", que é um input invalido, para os outros parametros serão passados inputs validos.

Requisito/Use Case/ Funcionalidad e	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Unteste d
Criar Order	id, date, sender, receiver	<pre>id = "", date = "2021- 01-01", sender = senderEntity, receiver = receiverEntity</pre>	IllegalArgumentExceptio n	PASS

OrderBVA02

Test Case ID	TC#OrderBVA0	Dependência
	2	s

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 date será igual a "", que é um *input* invalido, para os outros parametros serão passados inputs validos.

Requisito/Use Case/ Funcionalidad e	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Unteste d
Criar Order	id, date, sender, receiver	<pre>id = "1234", date = "", sender = senderEntity, receiver = receiverEntity</pre>	IllegalArgumentExceptio n	PASS

OrderBVA03

Test Case ID	TC#OrderBVA0	Dependência
	3	s

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

Requisito/Use Case/ Funcionalidad e	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Unteste d
Criar Order	id, date, sender, receiver	id = "", date = "2021- 01-01", sender = null, receiver = receiverEntity	IllegalArgumentExceptio n	PASS

OrderBVA04

|--|

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/	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Funcionalidade				
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#getProductBVA	Dependênci
	01	as

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/Us e Case/ Funcionalida de	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untest ed
Obter um Product da Order	Product product	product = null	IllegalArgumentExcepti on	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 exixtir 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 product	product =	Instanceof	PASS
Product da Order		product1	Product	

getProductECP02

Test Case ID	TC#getProductECP02	Dependências		
Data de	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
Execução				
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	product =	null	PASS
Product da Order		product1		

addProductBVA01

Test	Case ID	TC#addProductBVA0	Dependência
		1	S

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/ Funcionalidad e	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Unteste d
Adicionar um Product à Order	product	product = null	IllegalArgumentExceptio n	PASS

addProductECP01

Test Case ID TC#addProductECP01

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 product	product = product1	true	PASS
Product à Order		producti		

addProductECP02

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

Data de	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
Execução				
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

shipping Cost Calculation ECP 01

Test Case ID	TC#shippingCostCalculationECP	Dependência
	01	s

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/ Funcionalidad e	Inputs	Valores do Inputs	Resultado s Esperados	Pass/Fail/Unteste d
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:

(weight + distance) * 0.25 + (0.15 * volume) = (5.2 + 60) * 0.25 + (0.15 * 4.1) = 16.91€

shipping Cost Calculation ECP 02

Test Case ID	TC#shippingCostCalculationECP	Dependência
	02	s

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 reveiver são do mesmo district.

Requisito/Use Case/ Funcionalidad e	Inputs	Valores do Inputs	Resultado s Esperados	Pass/Fail/Unteste d
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#testGetOrderBVA	Dependênci
	01	as

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

Pré-condições	Pré-condições / estado inicial			
Tem de ser inst	anciada uma class Orde	rs válida.		
Procedimentos				
Invocar método	getOrder passando nu	II como <i>input</i> .		
Requisito/Us e Case/ Funcionalida de	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untest ed
Obter order	order	null	IllegalArgumentExcept ion	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 exitir pelo menos uma Order já adicionada à class Orders.

Procedimentos

Invocar método getOrder passando como input 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 *input* uma Orderque 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#testAddOrderBV	Dependênci
	A01	as

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 inst	anciado uma class Orde	rs válida.		
Procedimentos	5			
Invocar método	o addOrder do objeto do	tipo Orders, pa	assando como parâmetro	null.
Requisito/Us	Inputs	Valores do	Resultados Esperados	Pass/Fail/Untest
e Case/		Inputs		ed
Funcionalida				
de				
Adicionar	order	order = null	IllegalArgumentExcept	PASS
order			ion	

addOrderECP01

Test Case ID TC#testAddOrderEC	L 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	Inputs	Valores do	Resultados	Pass/Fail/Untested
Case/		Inputs	Esperados	
Funcionalidade				
Adicionar	order	order = order	true	PASS
order				

getOrderShippingCostBVA01

Test Case	TC#testGetOrderShippingCo	Dependên
ID	stBVA01	cias

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 null como input.

Requisito/U se Case/ Funcionalid ade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Unte sted
Obter o custo de envio de uma encomenda	Order	order = null	IllegalArgumentExc eption	PASS

getOrderShippingCostECP01

Test Case ID	TC#testGetOrderShippingCostECP	Dependência
	01	s

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 input.

Requisito/Use Case/ Funcionalidad e	Inputs	Valores do Inputs	Resultado s Esperados	Pass/Fail/Unteste d
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:

(weight + distance) * 0.25 + (0.15 * volume) = (5.2 + 60) * 0.25 + (0.15 * 4.1) = 16.91€

getOrder Shipping Cost ECP 02

Test Case ID	TC#testGetOrderShippingCostEC	Dependênci
	P02	as

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 *input,* 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/ Funcionalida de	Inputs	Valores do Inputs	Resultado s Esperado	Pass/Fail/Untest ed
Obter o custo de envio de uma encomenda.	Order	order = order	0	PASS

testGetOrderIndexBVA01

Test Case ID	TC#	Dependênci
	testGetOrderIndexBV	as
	A01	

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

Pré-condições	Pré-condições / estado inicial				
Objeto do tipo	Objeto do tipo Orders instanciado;				
Procedimento	s				
- Invocar méto	odo getOrderIndex;				
Requisito/Us	Inputs	Valores do	Resultados Esperados	Pass/Fail/Untest	
e Case/		Inputs		ed	
Funcionalida					
de					
Obter	Int index	-1	IndexOutOfBoundsExcep	PASS	
encomenda			tion		
através de					
um indice					

testGetOrderIndexBVA02

Test Case	TC#	Dependências	
ID	testGetOrderIndex		
	BVA02		

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

Pré-condiçõe	s / estado inicial			
Objeto do tip	o Orders instanciado;			
Procediment	os			
- Invocar mé	todo getOrderIndex;			
Requisito/	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Unte
Use Case/				sted
Funcionalid				
ade				
Obter	Int index	ordersByDistrictBeja	IndexOutOfBoundsExc	PASS
encomenda		.size()	eption	
através de				
um indice				

testGetOrderIndexECP01

Test Case ID	TC#	Dependências	
	testGetOrderIndexECP01		

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 getOrderIndex;

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

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/Us e Case/ Funcionalida de	Inputs	Valores do Input	Resultados Esperados	Pass/Fail/Untest ed
Obter distância entre Districts	districtDistance	null	IllegalArgumentExcep tion	PASS

getDistrictDistanceECP01

Test Case ID	TC#testGetDistrictDistanceECP	Dependências	
	01		

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/ Funcionalidad	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untest ed
е				
Obter	districtDistance	districtDistanc	Instanceof	PASS
distância		e = basic	DistrictDistanc	
entre <i>Districts</i>		districtDistanc	е	
		е		

getDistrictDistanceECP02

Test Case ID	TC#testGetDistrictDistanceECP0	Dependências
	2	

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/ Funcionalidad e	Inputs	Valores do Inputs	Resultado s Esperados	Pass/Fail/Unteste d
Obter distância entre Districts	districtDistance	districtDistanc e = basic districtDistanc e	null	PASS

addDistrictDistanceBVA01

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/U	Inputs	Valores do	Resultados Esperados	Pass/Fail/Untes
se Case/		Inputs		ted
Funcionalid				
ade				
Obter	districtDistance	districtDista	IllegalArgumentExce	PASS
distância		nce = null	ption	
entre				
Districts				

addDistrictDistanceBVA01

Test (Case ID	TC#testAddDistrictDistance	Dependênci
		BVA01	as

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/U se Case/ Funcionalid ade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untes ted
Obter distância entre Districts	districtDistance	districtDista nce = null	IllegalArgumentExce ption	PASS

getDistanceValueFromShippingBVA01

Test Case	TC#testGetDistanceValueFromShi	Dependên	
ID	ppingBVA01	cias	

Data de	Hora de Execução	Tester	Pass/Fail	Resumo da
Execução				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 necessario 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/ Funcionalid ade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Unt ested
Obter distância entre Districts	sender, receiver	sender = null, receiver = receiverEn tity	IllegalArgumentExc eption	PASS

get Distance Value From Shipping BVA02

Test Case	TC#testGetDistanceValueFromShi	Dependên
ID	ppingBVA02	cias

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

Para invocar o método getDistanceValueFromShipping, é necessario passar dois parâmetros do tipo Entity, uma entidade para o envio e outra para a receção.

Neste caso, a senderEntity será enviado um input válido, e a receiverEntity deverá ser passada como null.

Requisito/ Use Case/ Funcionalid ade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Unt ested
Obter distância entre Districts	sender, receiver	sender = senderEnti ty, receiver = null	IllegalArgumentExc eption	PASS

get Distance Value From Shipping ECP 01

Test Case ID	TC#testGetDistanceValueFromShipping	Dependênci
	ECP01	as

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

Pré-condições / estado inicial

Tem de ser instanciado : um objeto do tipo DistanceTable, e pelo menos dois objetos do tipo Entity válidos.

Deverá ser importado o ficheiro *Json*, correspondente às distâncias entre distritos, para o Objeto do tipo DistanceTable instanciado anteriormente.

Procedimentos

Invocar o método getDistanceValueFromShipping, passando dois parâmetros do tipo Entity válidos, que neste caso têm como valor de *District* Porto, para o *sende*r, e Braga, para o *receiver*. 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.

Requisito/Us e Case/ Funcionalida de	Inputs	Valores do Inputs	Resultad os Esperado s	Pass/Fail/Untest ed
Obter distância entre Districts	sender, receiver	sender = senderEntit y, receiver = receiverEnti ty	55	PASS

${\tt getDistanceValueFromShippingECP02}$

Test Case ID	TC#testGetDistanceValueFromShipping	Dependênci
	ECP02	as

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

Pré-condições / estado inicial

Tem de ser instanciado : um objeto do tipo DistanceTable, e pelo menos dois objetos do tipo Entity válidos.

Deverá ser importado o ficheiro *Json*, correspondente às distâncias entre distritos, no entanto não deverá existir distância entre os dois distritos passados como parâmetro.

Procedimentos

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 distânica.

Requisito/Us	Inputs	Valores do	Resultad	Pass/Fail/Untest
e Case/		Inputs	os	ed
Funcionalida			Esperado	
de			S	
Obter	sender, receiver	sender,	-1	PASS
distância		receiver		
entre				
Districts				

Detalhes Class DistrictDistance

DistrictDistanceBVA01

Test Case ID	TC#DistrictDistanceBVA	Dependênci
	01	as

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

Pré-condições ,	/ estado inicial			
Procedimentos				
Invocar constru	itor de DistrictDistance, pas	ssar o parâmetr	o <i>name,</i> que é do tipo Strii	ng, como vazio.
Requisito/Us	Inputs	Valores do	Resultados Esperados	Pass/Fail/Untest
e Case/		Inputs		ed
Funcionalida				
de				
Obter	name	name = " "	IllegalArgumentExcepti	PASS
distância			on	
entre <i>Districts</i>				

DistrictDistanceBVA02

Test Case ID	TC#DistrictDistanceBVA	Dependênci
	02	as

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

Pré-condições ,	/ estado inicial				
Procedimentos	Procedimentos				
Invocar constru	tor de DistrictDistance, pas	ssar o parâmetr	o null		
Requisito/Us e Case/ Funcionalida de	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untest ed	
Obter distância entre <i>Districts</i>	name	null	IllegalArgumentExcepti on	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 *name*, 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	name	name =	Instanceof	PASS
entre Districts		"Porto"	DistrictDistance	

getDistanceBVA01

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

Pré-condições /	Pré-condições / estado inicial					
Tem de ser insta	Tem de ser instanciado um objeto do tipo DistrictDistance.					
Procedimentos						
Invocar método	getDistance do objeto [DistrictDistance (passando null como parâm	etro).		
Requisito/Use	Inputs	Valores do	Resultados Esperados	Pass/Fail/Unteste		
Case/		Inputs		d		
Funcionalidad						
е	e e					
Obter	distance	distance =	IllegalArgumentExceptio	PASS		
distância		null	n			
entre Districts						

getDistanceECP01

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	distance	distance =	Instanceof	PASS
entre Districts		basicDistance	Distance	

DistrictDistanceECP01

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 *name*, 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	name	name =	Instanceof	PASS
entre Districts		"Porto"	DistrictDistance	

getDistanceECP02

Test Case ID	TC#getDistanceECP02	Dependências		
Data de	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
Execução				
12 janeiro	15:56h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de ser insta	nciado um objeto do tipo Dis	strictDistance.		
Procedimentos				
DistrictDistance.	getDistance do objeto Distrio	ctDistance, nesta	situação o <i>input</i> não	
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/Untested
Case/		Inputs	Esperados	
Funcionalidade				
Obter distância	distance	distance =	null	PASS
entre Districts		basicDistance		

addDistanceBVA01

Test Case ID	TC#addDistanceBVA0	Dependência
	1	s

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12 janeiro	14:10h	Fábio	PASS	

Pré-condições /	Pré-condições / estado inicial				
Tem de instanci	ado um objeto do tipo D	istrictDistance.			
Procedimentos					
Adicionar uma D	Distance ao DistrictDistar	nce, passando ur	n parâmetro do tipo null.		
Requisito/Use	Inputs Valores do Resultados Esperados Pass/Fail/Unteste				
Case/		Inputs		d	
Funcionalidad					
е					
Obter	distance	distance =	Throw	PASS	
distância		null	IllegalArgumentExcepti		
entre Districts			on		

addDistanceECP01

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

Data de	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
Execução				
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	Inputs	Valores do	Resultados	Pass/Fail/Untested
Case/		Inputs	Esperados	
Funcionalidade				
Obter distância	distance	distance =	true	PASS
entre Districts		basicDistance		

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

Tes	t Case ID	TC#testDistanceBVA	Dependência
		01	S

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 distanceValue com o valor de -10, e o parâmetro é passado com input válido. Pass/Fail/Untest Requisito/Use Inputs Valores do **Resultados Esperados** Case/ Inputs ed **Funcionalidad** Obter id = "Porto", PASS id, distanceValue IllegalArgumentExcepti distância distanceValu on entre Districts e = -10

DistanceBVA02

Test Case ID	TC#testDistanceBVA	Dependência
	02	s

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

Pré-condições /	estado inicial					
Procedimentos						
O construtor Dispassada vazia.	stance recebe dois parâi	metros: uma Stri	ng e um inteiro, sendo que a	a String deve ser		
Requisito/Use Case/ Funcionalidad e	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untest ed		
Obter distância entre <i>Districts</i>	id, distanceValue	id = "", distanceValu e = 30	IllegalArgumentExcepti on	PASS		

DistanceECP01

|--|

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 Dis	O construtor Distance recebe dois parâmetros: uma String e um inteiro, neste caso serão passados					
para todos os pa	râmetros inputs válidos.					
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/Untested		
Case/		Inputs	Esperados			
Funcionalidade						
Obter distância	id, distanceValue	id = "Porto",	Instanceof	PASS		
entre Districts		distanceValue	Distance			
		= 30				

Detalhes Class ImportConfig

testImportConfigCapacityBVA01

Test Cas	TC#testImportConfigCapacity
	BVA01

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

Pré-condições	s / estado inicial						
Instanciar obj	Instanciar objeto do tipo ImportConfigsTruck;						
Procedimento	Procedimentos						
O método rec	ebe um parâmetro: String						
Requisito/U se Case/	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untes ted			
Funcionalid							
ade							
Obter	String path	null	IllegalArgumentExce	PASS			
distância			ption				
entre							
Districts							

testImportConfigCapacityBVA02

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;					
Procedimento	os				
O método rec	O método recebe um parâmetro: String				
Requisito/U	Inputs	Valores do	Resultados Esperados	Pass/Fail/Untes	
se Case/		Inputs		ted	
Funcionalid					
ade					
Obter	String path	un	IllegalArgumentExce	PASS	
distância			ption		
entre					
Districts					

$\underline{\mathsf{testImportConfigCapacityECP01}}$

Test Case ID	TC#testImportConfigCapacityECP	Dependências
	01	

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	Procedimentos					
O método receb	e um parâmetro: String					
Requisito/Use	Inputs	Valores do	Resultado	Pass/Fail/Unteste		
Case/		Inputs	s	d		
Funcionalidad			Esperados			
е						
Obter	String path	"naoexite.json	63	PASS		
distância		"				
entre Districts						

3. Summary Tests

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