

Engenharia de Software 2

Trabalho pratico 2

Módulo de Custo de Envio / Transações /Expedição

<ES2_2021_GRUPO_11>

Version <1.6>

<20/01/2022>

Realizado por:

8180378-Hugo Silva

8200306-Fábio Costa

Histórico de Versões

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Hugo	27-12-2021	Todos	28-12-2021	Adição de análise Funcional
1.1	Fabio	3-01-2022	Todos	3-01-2022	Adição de uses cases
1.2	Hugo	5-01-2022	Todos	5-01-2022	Tests BVA BasicEntity
1.3	Hugo	7-01-2022	Todos	7-01-2022	Todos os tests BVA
1.4	Fabio	13-01-2022	Todos	13-01-2022	Métodos de agrupar encomendas
1.5	Hugo	18-01-2022	Todos	18-01-2022	Indentificação dos testes e tabelas terminadas
1.6	Hugo	20-01-2022	Todos	20-01-2022	Método em falta adicionado

Índice

1.Introdução	6
Identificador do documento	6
Âmbito.....	6
Glossário.....	6
Referências	6
2.Features/Itens a testar	7
3. Detalhes da abordagem aos testes	9
3.1 Método averageValueTransactions	9
Técnica ECP.....	9
Técnica BVA	9
3.2 Método averageNumberProductsTransaction	10
Técnica ECP	10
Técnica BVA	10
3.3 Método averageValueSalesPurchasesDistrict.....	11
Técnica ECP	11
Técnica BVA	12
3.4 Método registerOrdersInLedger	13
Técnica ECP	13
Técnica BVA	15
3.5 Método createFile.....	16
Técnica ECP.....	16
Técnica BVA	16
3.6 Construtor Order	17
Técnica ECP	17
Técnica BVA	17
3.6 Método addProduct	18
Técnica ECP	18
Técnica BVA	19
3.7 Método getProduct	19
Técnica ECP.....	19
Técnica BVA	20
3.8 Método addOrder	20
Técnica ECP.....	20
Técnica BVA	21
3.9 Método getOrder	22

Técnica ECP	22
Técnica BVA	23
3.10 Método numberOrdersByDistrict	23
Técnica ECP	23
Técnica BVA	23
3.11 Método standardDeviationNumberProductTransactions	24
Técnica ECP	24
Técnica BVA	24
3.12 Método standardDeviationTransactions	25
Técnica ECP	25
Técnica BVA	25
3.12 Construtor Product.....	26
Técnica ECP	26
Técnica BVA	26
3.12 Método Create File.....	27
Técnica ECP	27
Técnica BVA	27
3.13 Construtor Truck	28
Técnica ECP	28
Técnica BVA	28
3.13 Método addProductInTruck.....	29
Técnica ECP	29
Técnica BVA	29
3.14 Método addAllOrderInTruck.....	30
Técnica ECP	30
Técnica BVA	30
3.14 Método getProductContainerTruck	31
Técnica ECP	31
Técnica BVA	32
3.15 Construtor DistrictTruck	33
Técnica ECP	33
Técnica BVA	33
3.16 Método creatTruck.....	34
Técnica ECP	34
Técnica BVA	34
3.17 Método getTruckByIndex	35

Técnica ECP	35
Técnica BVA	35
3.18 Construtor ProductsInsideTruck	36
Técnica ECP	36
Técnica BVA	36
3.19 Método groupOrdersByTrucks.....	37
Técnica ECP	37
Técnica BVA	39
3.20 Método getArrayOrdersByDistrict	40
Técnica ECP	40
Técnica BVA	40
3.21 Método getOrderShippingCost	41
Técnica ECP	41
Técnica BVA	41
3.22 Método addDistrictTruck.....	42
Técnica ECP	42
Técnica BVA	42
3.23 Método getOrderIndex.....	43
Técnica ECP	43
Técnica BVA	43
3.24 Método getProductListProductsInsideTruck	44
Técnica ECP	44
Técnica BVA	44
3.25 Método importCostsTableKM.....	45
Técnica ECP	45
Técnica BVA	45
3.26 Construtor DistanceLine	46
Técnica ECP	46
Técnica BVA	46
4. Identificação dos Testes	47
5. Critérios de passagem ou falha das features.....	53

1.Introdução

Identificador do documento

TestCaseSpecificationMóduloTransações.

Âmbito

Este documento refere-se a um relatório de testes desenvolvido para a disciplina de Engenharia de Software II do curso de Engenharia Informática.

O relatório é realizado no âmbito do trabalho prático 2 e o objetivo é testar a *API* desenvolvida para responder às novas funcionalidades.

A estratégia de testes abordada é a estratégia de testes de caixa preta, aplicando técnicas ECP E BVA.

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: *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.

Referências

IEEE Std 829™-2008

2.Features/Itens a testar

Na seguinte tabela são apresentados os métodos que foram identificados para teste.

Item a testar	Descrição	Requisitos	Responsabilidade
Método averageValueTransactions	Método que retorna o valor médio das transações	11	Hugo
Método averageNumberProductsTransaction	Método que retorna o número de produtos médio por transação	9	Hugo
Método averageValueSalesPurchasesDistrict	Método que retorna o número médio de compras e vendas por distrito	11	Hugo
Método registerOrdersInLedger	Método que regista encomendas no ledger	7	Hugo
Método createFile	Método que cria um ficheiro	22	Hugo
Construtor Order	Construtor que cria uma encomenda	16	Hugo
Método addProduct	Método que adiciona um produto à encomenda	16	Fábio
Método getProduct	Método para obter produto de uma encomenda	16	Fábio
Método addOrder	Método que adiciona uma encomenda à lista	16	Fábio
Método getOrder	Método que obtém uma encomenda da lista	16	Fábio
Método numberOrdersByDistrict	Método que obtém número de encomendas por distrito	12	Hugo

Método standardDeviationNumberProductTransactions	Método que obtem o desvio padrão do número de produtos por transação	14	Hugo
Método standardDeviationTransactions	Método que obtem o desvio padrão do valor médio de transações	13	Hugo
Construtor Product	Construtor que permite criar um produto	16	Fábio
getArrayOrdersByDistrict	Método que obtem a lista de encomendas de um distrito	20	Hugo
Construtor Truck	Construtor que permite instanciar um camião	21	Hugo
Método addProductInTruck	Método que permite adicionar um produto num camião	23	Hugo
Método groupOrdersByTrucks	Método que agrupa encomendas,por distrito, a contentores de camiões	19	Hugo
Método getOrderShippingCost	Método que calcula e retorna o custo de envio de uma encomenda	17	Fábio
Método importCostsTableKM	Método que carrega tabela de ficheiro	x	Fábio
Construtor DistanceLine	Método que cria uma linha da tabela	x	Fábio
Método getOrderIndex	Método que retorna uma order dado um index	17	Hugo

3. Detalhes da abordagem aos testes

3.1 Método averageValueTransactions

Para este método o objetivo é testar se o método averageValueTransactions retorna os resultados esperados.

Técnica ECP

Case	Descrição	Crítérios	Classe Válida	Classe Inválida
1	Obter média de transações	Nº inputs	1	!= 1
		Tipo inputs	ArrayList<Block> ledger	!= ArrayList<Block> ledger
		Pré-Condições	Objeto do tipo IOOrders instanciado	
		Restrições Inputs	<ul style="list-style-type: none"> O ledger deve ter pelo menos uma transação registada 	<ul style="list-style-type: none"> Não tem
		Pós-Condições	Return valor medio (double)	Throws IllegalArgumentException
		Resultado Esperado	valor medio das transações do <i>ledger</i> (double)	IllegalArgumentException
		Exemplos	1-addTransaction (transation) 2- addTransaction (transaction2) 3-registerTransactionsInLedger() 4- averageValueTransactions()	getTransaction (null)

Figura 1 Tabela ECP média de transações

Técnica BVA

Para a técnica BVA foi testado correr o método apenas com o bloco genesis criado. Como o bloco genesis não tem transações é esperado como resultado uma exceção.

3.2 Método averageNumberProductsTransaction

Para este método o objetivo é testar se o método averageNumberProductsTransaction retorna os resultados esperados.

Técnica ECP

Case	Descrição	Crítérios	Classe Válida	Classe Inválida
1	Obter média de produtos por transação	Nº inputs	1	!= 1
		Tipo inputs	ArrayList<Block> ledger	!= ArrayList<Block> ledger
		Pré-Condições	Objeto do tipo IStatistics instanciado	
		Restrições Inputs	<ul style="list-style-type: none"> O ledger deve ter pelo menos uma transação registada 	<ul style="list-style-type: none"> Não tem
		Pós-Condições	Return valor medio (double)	Throws IllegalArgumentException
		Resultado Esperado	valor medio das transações do ledger (double)	IllegalArgumentException
		Exemplos	1-addTransaction (transation) 2- addTransaction (transaction2) 3-registerTransactionsInLedger() 4-averageValueTransactions(this.ledger)	averageValueTransactions (null)

Figura 2 Tabela ECP média de produtos por transação

Técnica BVA

Para a técnica BVA foi testado correr o método apenas com o bloco genesis criado e com um input a null. Como o bloco genesis não tem transações é esperado como resultado uma exceção.

Case ID	Inputs	Min	Min blow	MID	MAX	Max above	Special Case
1	ArrayList<Block> ledger	N	N	N	N	N	null
2	ArrayList<Block> ledger	posição 0 do array	N	N	N	N	N

Figura 3 Tabela BVA média de produtos por transação

3.3 Método `averageValueSalesPurchasesDistrict`

Para este método o objetivo é testar se o método `averageValueSalesPurchasesDistrict` retorna a média de vendas e compras por distrito corretamente.

Técnica ECP

Cas e	Descrição	Critérios	Classe Válida	Classe Inválida
1	Obter média de vendas por distrito	Nº inputs	1	!= 1
		Tipo inputs	ArrayList<Block> ledger	!= ArrayList<Block> ledger
		Pré-Condições	Objeto do tipo IStatistics instanciado	
		Restrições Inputs	<ul style="list-style-type: none"> O ledger deve ter pelo menos uma transação registrada 	<ul style="list-style-type: none"> Não tem
		Pós-Condições	Return <i>hashmap</i> com as médias	Throws <code>IllegalArgumentException</code>
		Resultado Esperado	<i>hashmap</i> com as médias	<code>IllegalArgumentException</code>
		Exemplos	1-addTransaction (transation) 2- addTransaction (transaction2) 3-registerTransactionsInLedger() 4- averageValueSalesPurchaseDistrict (this.ledger)	averageValueSalesPurchaseDistrict (null)

Figura 4 Tabela ECP01 método `averageValueSales`

Cas e	Descriç ão	Critérios	Classe Válida	Classe Inválida
2	Obter média com entidad es de envio e compra iguais	Nº inputs	1	!= 1
		Tipo inputs	ArrayList<Block> ledger	!= ArrayList<Block> ledger
		Pré- Condições	Objeto do tipo BasicOrganization instanciado	
		Restrições Inputs	<ul style="list-style-type: none"> O ledger deve ter pelo menos uma transação registada Pelo menos uma das transações deve ter a entidade de envio e compra igual. <u>Exemplo em estado de objeto</u> ledger.size() > 0	<ul style="list-style-type: none"> Não tem
		Pós- Condições	Retorna hashmap com as médias	Throws IllegalArgumentException
		Resultado Esperado	hashmap com as médias	IllegalArgumentException
		Exemplos	1-addTransaction (transation) 2- addTransaction (transaction2) 3-registerTransactionsInLedger() 4- averageValueSalesPurchaseDistrict (this.ledger)	averageValueSalesPurchaseDistrict (null)

Figura 5 Tabela ECP3 método *averageValueSalesPurchasesTransaction*

Para o TC ECP01 o objetivo do teste é verificar o normal comportamento do método.

Para o TC ECP02 o objetivo é testar com um número bastante elevado de transações.

Para o TC ECP03 o objetivo é testar o comportamento do método quando as entidades de envio e recepção são a mesma.

Técnica BVA

Para a técnica BVA foi testado:

ID	Case
1	this.ledger.size() == 1
2	null

No case 1 foi testado invocar método com o ledger apenas com o bloco Genesis criado.

No case 2 foi testado invocar o método com o parâmetro a *null*.

3.4 Método registerOrdersInLedger

Este método tem como objetivo registar encomendas no *ledger*. Cada encomenda representa duas transações (pagamento e envio). Para registar uma encomenda é necessário dois tokens, sendo que estes devem ser adicionados de forma automática.

Técnica ECP

Cas e	Descrição	Critérios	Classe Válida	Classe Inválida
1	Registar encomendas no ledger	Nº inputs	1	!= 1
		Tipo inputs	ArrayList<Order>	!= ArrayList<Order>
		Pré-Condições	Objeto do tipo IOrder instanciado	
		Restrições Inputs	<ul style="list-style-type: none"> O arraylist deve ter pelo menos uma encomenda. A entidade emissora da encomenda deve possuir pelo menos dois tokens Estado da encomenda não pode estar “fechado” <p><u>Exemplo em estado de objeto</u></p> <pre>ArrayList<Order> order.size() > 0; Entity sender.getTokens() >=2; Order order.state != closed;</pre>	<ul style="list-style-type: none"> Não tem
		Pós-Condições	Retorna número transações registadas	Não tem
		Resultado Esperado	número transações registadas	Não tem

Figura 6 Tabela ECP01 registerOrdersInLedger

Para o TC ECP01 o objetivo é verificar o normal funcionamento do método.

Cas e	Descriç ão	Critérios	Classe Válida	Classe Inválida
2	Registar encomendas no ledger que já foram registadas	Nº inputs	1	!= 1
		Tipo inputs	ArrayList<Order>	!= ArrayList<Order>
		Pré- Condições	Objeto do tipo Order instanciado	
		Restrições Inputs	<ul style="list-style-type: none"> O arraylist deve ter uma encomenda. A entidade emissora da encomenda deve possuir pelo menos dois tokens Estado da encomenda deve ser "fechado" <p><u>Exemplo em estado de objeto</u></p> <pre>ArrayList<Order> order.size() > 0; Entity sender.getTokens() >= 2; Order order.state == closed;</pre>	<ul style="list-style-type: none"> Não tem
		Pós- Condições	0	Não tem
		Resultado Esperado	0	Não tem

Figura 7 Tabela ECP02 registerOrdersInLedger

Para este TC ECP02 o objetivo é verificar se o método regista encomendas fechadas, ou seja, encomendas que já estão no *ledger*.

Técnica BVA

Para a técnica BVA foi testado:

ID	Case	Objet Example
1	<i>array</i> de encomendas vazio	<code>basicOrders.size() == 0</code>
2	entidade de envio com 0 <i>tokens</i>	<code>sender.getTokens() == 0</code>
3	entidade de envio com 1 <i>tokens</i>	<code>sender.getTokens() == 1</code>

Figura 8 Tabela BVA *registerOrdersInLedger*

No TC 1 o objetivo é tentar registar um *array* de encomendas vazio.

No TC 2 o objetivo é testar o comportamento do método quando a entidade não possui qualquer *tokens*.

No TC 3 o objetivo é testar um valor limite mínimo dado que para efetuar registo de uma encomenda é preciso ter pelo menos dois *tokens*.

3.5 Método createFile

Técnica ECP

Case	Descrição	Critérios	Classe Válida	Classe Inválida
1	Criar um ficheiro	Nº inputs	1	!= 1
		Tipo inputs	String	!=String
		Pré-Condições	StatisticExporter instanciado	
		Restrições Inputs	!= null && != ""	== null != ""
		Pós-Condições	Cria um ficheiro	Throws IllegalArgumentException
		Resultado Esperado	true	IllegalArgumentException

Figura 9 Tabela ECP creatFile

Técnica BVA

Para a técnica BVA foi testado:

ID	Case
1	null
2	"" (empty)

Figura 10 Tabela BVA creatFile

No TC 1 é passado o input a *null*.

No TC2 é passado o input *empty*.

3.6 Construtor Order

Técnica ECP

Case	Descrição	Critérios	Classe Válida	Classe Inválida
1	Obter média de transações	Nº inputs	4	!=4
		Tipo inputs	String,String,Entity,Entity	!=String, !=String, !=Entity, !=Entity
		Pré-Condições	Não tem	
		Restrições Inputs	id != null && id!= "" && date!= null && date!= "" && sender!= null && receiver != null	id == null id== "" date == null date== "" sender == null receiver == null
		Pós-Condições	Cria uma Order	Throws IllegalArgumentException
		Resultado Esperado	Objeto Order criado	IllegalArgumentException
		Exemplo	BasicOrder("id_1", "2021-01-01", senderEntity, receiverEntity)	BasicOrder("", "2021-01-01", senderEntity, receiverEntity)

Figura 11 Tabela ECP BasicOrder

Técnica BVA

Para a técnica BVA foi testado:

ID	Input	Case
1	id	null
2	id	"" (empty)
3	date	null
4	date	"" (empty)
5	sender	null
6	receiver	null

Figura 12 Tabela BVA BasicOrder

3.6 Método addProduct

Técnica ECP

Case	Descrição	Critérios	Classe Válida	Classe Inválida
1	Adicionar produto à encomenda	Nº inputs	1	!= 1
		Tipo inputs	Product	!= Product
		Pré-Condições	Objeto IOrder Instanciado	
		Restrições Inputs	Objeto do tipo Product não pode estar adicionado na encomenda	Input tem de ser null
		Pós-Condições	Adiciona produto a uma encomenda	Throws IllegalArgumentException
		Resultado Esperado	true	IllegalArgumentException
		Exemplo	basicOrder.addProduct(product1)	basicOrder.addProduct(null)

Figura 13 Tabela ECP01 addProduct

Case	Descrição	Critérios	Classe Válida	Classe Inválida
1	Adicionar produto existente na encomenda	Nº inputs	1	!= 1
		Tipo inputs	Product	!= Product
		Pré-Condições	Objeto IOrder Instanciado	
		Restrições Inputs	Objeto do tipo Product tem de estar na encomenda	Input tem de ser null
		Pós-Condições	Adiciona produto a uma encomenda	Throws IllegalArgumentException
		Resultado Esperado	false	IllegalArgumentException
		Exemplo	basicOrder.addProduct(product1) basicOrder.addProduct(product1)	basicOrder.addProduct(null)

Figura 14 Tabela ECP02 addProduct

Para o ECP 01 foi testado adicionar um produto que não existe na encomenda, o objetivo é testar o método em condições normais.

Para o ECP 02 foi testado adicionar um produto já existente na encomenda, o objetivo é fazer a verificação correta dos produtos existentes e retorna o esperado.

Técnica BVA

Para a técnica BVA foi testado:

ID	Input	Case
1	product	null

Figura 15 Tabela BVA addProduct

3.7 Método getProduct

Técnica ECP

Case	Descrição	Critérios	Classe Válida	Classe Inválida
1	Obter produto de uma encomenda	Nº inputs	1	!= 1
		Tipo inputs	Product	!= Product
		Pré-Condições	Objeto de uma classe que implemente a IOrder Instanciado	
		Restrições Inputs	Objeto do tipo Product deve existir na order	Input tem de ser null
		Pós-Condições	Retorna o produto existente	Throws IllegalArgumentException
		Resultado Esperado	Instance of Product	IllegalArgumentException
		Exemplo	basicOrder.addProduct(product1) basicOrder.getProduct(product1)	basicOrder.addProduct(null)

Figura 16 Tabela ECP01 getProduct

Case	Descrição	Critérios	Classe Válida	Classe Inválida
2	Obter produto não existente na encomenda	Nº inputs	1	!= 1
		Tipo inputs	Product	!= Product
		Pré-Condições	Objeto de uma classe que implemente a IOrder instanciado	
		Restrições Inputs	Objeto do tipo Product não deve existir na order	Input tem de ser null
		Pós-Condições	Retorna null	Throws IllegalArgumentException
		Resultado Esperado	null	IllegalArgumentException
		Exemplo	basicOrder.addProduct(product1) basicOrder.getProduct(product2)	basicOrder.addProduct(null)

Figura 17 Tabela ECP02 getProduct

Técnica BVA

Para a técnica BVA foi testado:

ID	Input	Case
1	product	null

Figura 18 Tabela BVA getProduct

3.8 Método addOrder

Técnica ECP

Case	Descrição	Critérios	Classe Válida	Classe Inválida
1	Adicionar uma encomenda à lista	Nº inputs	1	!= 1
		Tipo inputs	Order	!= Order
		Pré-Condições	O objeto do tipo IOrders e objeto IOrder instanciados	
		Restrições Inputs	O objeto order não pode estar já inserido na lista de encomendas	Input tem de ser null
		Pós-Condições	Adiciona encomenda	Throws IllegalArgumentException
		Resultado Esperado	true	IllegalArgumentException
		Exemplo	basicOrders.addOrder(order)	basicOrders.addOrder(null)

Figura 19 Tabela ECP01 método addOrder

Case	Descrição	Critérios	Classe Válida	Classe Inválida
2	Adicionar uma encomenda já existente na lista	Nº inputs	1	!= 1
		Tipo inputs	Order	!= Order
		Pré-Condições	O objeto do tipo Orders e objeto Order instanciados	
		Restrições Inputs	O objeto order já deve estar na lista de encomendas	<i>Input</i> tem de ser <i>null</i>
		Pós-Condições	Encomenda não adicionada	Throws <code>IllegalArgumentException</code>
		Resultado Esperado	false	<code>IllegalArgumentException</code>
		Exemplo	<code>basicOrders.addOrder(order)</code> <code>basicOrders.addOrder(order)</code>	<code>basicOrders.addOrder(null)</code>

Figura 20 Tabela ECP02 método `addOrder`

Técnica BVA

Para a técnica BVA foi testado:

ID	Input	Case
1	order	<i>null</i>

Figura 21 Tabela BVA `addOrder`

3.9 Método getOrder

Técnica ECP

Case	Descrição	Critérios	Classe Válida	Classe Inválida
1	Obter uma encomenda da lista	Nº inputs	1	!= 1
		Tipo inputs	Order	!= Order
		Pré-Condições	Instâncias de classes que implementam as interfaces IOrders e Order criadas	
		Restrições Inputs	O objeto order no input deve estar na lista de encomendas	Input tem de ser null
		Pós-Condições	Retorna a encomenda	Throws IllegalArgumentException
		Resultado Esperado	Instance of Order	IllegalArgumentException
		Exemplo	basicOrders.addOrder(order) basicOrders.getOrder(order)	basicOrders.getOrder(null)

Figura 22 Tabela ECP01 método getOrder

Case	Descrição	Critérios	Classe Válida	Classe Inválida
2	Obter uma encomenda que não existe na lista	Nº inputs	1	!= 1
		Tipo inputs	Order	!= Order
		Pré-Condições	O objeto do tipo Orders e objeto Order instanciados	
		Restrições Inputs	O objeto order no input não deve estar na lista de encomendas	Input tem de ser null
		Pós-Condições	Retorna null	Throws IllegalArgumentException
		Resultado Esperado	null	IllegalArgumentException
		Exemplo	basicOrders.addOrder(order) basicOrders.getOrder(order2)	basicOrders.getOrder(null)

Figura 23 Tabela ECP 02 método getOrder

Técnica BVA

Para a técnica BVA foi testado:

ID	Input	Case
1	order	null

Figura 24 Tabela BVA getOrder

3.10 Método numberOrdersByDistrict

Técnica ECP

Case	Descrição	CrITÉRIOS	Classe Válida	Classe Inválida
1	Obter número de encomendas por distrito	Nº inputs	1	!= 1
		Tipo inputs	ArrayList<Block> ledger	!= ArrayList<Block> ledger
		Pré-Condições	O objeto, cuja classe implementa a interface IStatistics , instanciado	
		Restrições Inputs	ledger.size() > 1	ledger ==null ledger.size()==1
		Pós-Condições	Retorna <i>HashMap</i> com resultado	Throws <i>IllegalArgumentException</i>
		Resultado Esperado	<i>hashmap</i>	<i>IllegalArgumentException</i>
		Exemplo	statistics .numberOrdersByDistrict(ledger)	statistics .numberOrdersByDistrict(null)

Figura 25 Tabela ECP01 numberOrdersByDistrict

Técnica BVA

Para a técnica BVA foi testado:

Case ID	Inputs	Min	Min blow	MID	MAX	Max above	Special Case
1	ArrayList<Block> ledger	N	N	N	N	N	null
2	ArrayList<Block> ledger	1	N	N	N	N	N

Figura 26 Tabela BVA método numberOrdersByDistrict

No TC BVA 01 foi testado passou o input a null.

No TC BVA 02 foi testado passar como parâmetro o ledger apenas com o bloco genesis criado, o objetivo é testar se o método faz a verificação correta no número de blocos e retorna o esperado.

3.11 Método standardDeviationNumberProductTransactions

Técnica ECP

C a s e	Descrição	Crítérios	Classe Válida	Classe Inválida
1	Obter desvio padrão da média do número de produtos por transação	Nº inputs	1	!= 1
		Tipo inputs	ArrayList<Block> ledger	!= ArrayList<Block> ledger
		Pré-Condições	O objeto que implementa a interface IStatistics instanciado	
		Restrições Inputs	ledger.size() > 1	ledger == null ledger.size() == 1
		Pós-Condições	Retorna desvio padrão do número medio de produtos por transação	Throws IllegalArgumentException
		Resultado Esperado	valor desvio padrão (Double)	IllegalArgumentException
		Exemplo	statistics .standardDeviationNumberProductTransactions (ledger)	statistics .standardDeviationNumberProductTransactions (null)

Figura 27 Tabela ECP01 standardDeviationNumberProductTransactions

Técnica BVA

Para a técnica BVA foi testado:

Case ID	Inputs	Min	Min blow	MID	MAX	Max above	Special Case
1	ArrayList<Block> ledger	N	N	N	N	N	null
2	ArrayList<Block> ledger	1	N	N	N	N	N

Figura 28 Tabela BVA standardDeviationNumberProductTransactions

3.12 Método standardDeviationTransactions

Técnica ECP

C a s e	Descrição	Critérios	Classe Válida	Classe Inválida
1	Obter desvio padrão do valor médio das transações	Nº inputs	1	!= 1
		Tipo inputs	ArrayList<Block> ledger	!= ArrayList<Block> ledger
		Pré-Condições	O objeto que implementa a interface IStatistics instanciado	
		Restrições Inputs	ledger.size() > 1	ledger == null ledger.size() == 1
		Pós-Condições	Retorna desvio padrão valor médio das transações	Throws IllegalArgumentException
		Resultado Esperado	valor desvio padrão (Double)	IllegalArgumentException
		Exemplo	statistics . standardDeviationTransactions (ledger)	statistics . standardDeviationTransactions (null)

Figura 29 Tabela ECP 01 standardDeviationTransactions

Técnica BVA

Para a técnica BVA foi testado:

Case ID	Inputs	Min	Min blow	MID	MAX	Max above	Special Case
1	ArrayList<Block> ledger	N	N	N	N	N	null
2	ArrayList<Block> ledger	1	N	N	N	N	N

Figura 30 Tabela BVA standardDeviationTransactions

3.12 Construtor Product

Técnica ECP

C a s e	Descrição	Critérios	Classe Válida	Classe Inválida
1	Criar um produto	Nº inputs	7	!= 7
		Tipo inputs	Int, String, String, Double, Int, Double, Double	!= Int, String, String, Double, Int, Double, Double
		Pré-Condições	Não tem	
		Restrições Inputs	id >= 0 && name != "" && name != null && quantity > 0 && description != null && description != "" && volume > 0 && volume <= 63 && weight > 0	Id < 0 name == "" name == null description == null description == "" volume <= 0 volume > 63 weight <= 0 quantity < 1
		Pós-Condições	Cria objeto do tipo Product	Throws IllegalArgumentException
		Resultado Esperado	Instance of Product	IllegalArgumentException
		Exemplo	new Product(1, "Product1", "Description", 1, 3, 5.1, 5.2)	new Product(1, "Product1", "Description", 1, 3, -4.1, 5.2)

Figura 31 Tabela ECP01 Construtor Product

Técnica BVA

Para a técnica BVA foi testado:

ID	Input	Case	Value
1	id	Minium limit	-1
2	name	Special Case	""
3	name	Special Case	null
4	description	Special Case	""
5	description	Special Case	null
6	volume	Minium limit	0
7	volume	Max limite	63.1
8	weight	Minium limit	0
9	value	Minium limit	-0.1
10	quantity	Minium limit	0

Figura 32 Tabela BVA Product

3.12 Método Create File

Técnica ECP

C a s e	Descrição	Critérios	Classe Válida	Classe Inválida
1	Criar um ficheiro	Nº inputs	1	!= 1
		Tipo inputs	String path	!=String path
		Pré- Condições	Não tem	
		Restrições Inputs	path != "" && path!= null	path == "" path== null
		Pós- Condições	Cria Ficheiro	Throws IllegalArgumentException
		Resultado Esperado	true	IllegalArgumentException
		Exemplo	creatFile("drivers/drivers.json")	creatFile(null)

Figura 33 Tabela ECP 01 creatFile

Técnica BVA

ID	Input	Case	Value
1	path	Special Case	null
2	path	Special Case	""

Figura 34 Tabela BVA creatFile

3.13 Construtor Truck

Este método permite criar um camião.

Técnica ECP

C a s e	Descrição	Critérios	Classe Válida	Classe Inválida
1	Criar um camião	Nº inputs	1	!= 1
		Tipo inputs	String name	!=String name
		Pré- Condições	Não tem	
		Restrições Inputs	name != "" && name != null	name == "" name == null
		Pós- Condições	Cria um truck	Throws IllegalArgumentException
		Resultado Esperado	Instance of truck (true)	IllegalArgumentException
		Exemplo	Truck("driver")	Truck (null)

Figura 35 Tabela ECP01 construtor Truck

Técnica BVA

ID	Input	Case	Value
1	name	Special Case	null
2	name	Special Case	""

Figura 36 Tabela BVA construtor Truck

3.13 Método addProductInTruck

Este método adiciona um produto dentro do contentor de um camião

Técnica ECP

C a s e	Descrição	Critérios	Classe Válida	Classe Inválida
1	Adicionar um produto a um camião	Nº inputs	3	!= 3
		Tipo inputs	String idOrder , String idProduct ,Double volum	!= String idOrder , String idProduct ,Double volum
		Pré-Condições	Objeto do tipo Truck Instanciado	
		Restrições Inputs	idOrder != "" && idOrder != null && idProduct != null && idProduct != "" && volum > 0 && volum < 63.1	IdOrder == "" IdOrder == null idProduct == null idProduct == "" volum > 63 volum <= 0
		Pós-Condições	Adiciona um produto a um camião	Throws IllegalArgumentException
		Resultado Esperado	true	IllegalArgumentException
		Exemplo	addProductInTruck("id","dd",23.2)	addProductInTruck("id","dd",0)

Figura 37 Tabela ECP01 addProductInTruck

Técnica BVA

ID	Input	Limit	Case	Value
1	IdOrder	x	Special Case	null
2	IdOrder	x	Special Case	""
3	idProduct	x	Special Case	null
4	idProduct	x	Special Case	""
5	volum	Minimum Limit	x	0
6	volum	Maximum limit	x	63.1

Figura 38 Tabela BVA addProductInTruck

3.14 Método addAllOrderInTruck

Este método adiciona uma encomenda completa a um contentor de um camião

Técnica ECP

C a s e	Descrição	Crítérios	Classe Válida	Classe Inválida
1	Adicionar uma encomenda a um camião	Nº inputs	1	!= 1
		Tipo inputs	Order order	!= Order order
		Pré- Condições	Objeto do tipo Truck Instanciado	
		Restrições Inputs	Order != null && order.getVolume() <= 63	Order == null order.getVolume() > 63
		Pós- Condições	Adiciona uma encomenda a um camião	Throws IllegalArgumentException
		Resultado Esperado	true	IllegalArgumentException
		Exemplo	addAllOrderInTruck(order)	addAllOrderInTruck (null)

Figura 39 Tabela ECP01 addAllOrderInTruck

Técnica BVA

ID	Input	Limit	Case	Value
1	volum	Minimum Limit	x	0
2	volum	Maximum limit	x	63.1
3	order	x	Special Case	null

Figura 40 Tabela BVA addAllOrderInTruck

3.14 Método getProductContainerTruck

Este método obtém um produto de um contentor de um camião.

Técnica ECP

C a s e	Descrição	Crítérios	Classe Válida	Classe Inválida
1	Obter produto de um camião	Nº inputs	1	!= 1
		Tipo inputs	ProductsInsideTruck productsInsideTruck	!= ProductsInsideTruck productsInsideTruck
		Pré- Condições	Objeto do tipo Truck Instanciado	
		Restrições Inputs	Produto passado como parâmetro tem de estar inserido no camião	Não tem
		Pós- Condições	Retorna o produto encontrado	Throws IllegalArgumentException
		Resultado Esperado	Instance of ProductsInsideTruck (true)	IllegalArgumentException
		Exemplo	getProductContainerTruck ("id","dd",23.2)	getProductContainerTruck (null)

Figura 41 Tabela ECP01 getProductContainerTruck

C a s e	Descrição	Critérios	Classe Válida	Classe Inválida
2	Obter produto não existente no camião	Nº inputs	1	!= 1
		Tipo inputs	ProductsInsideTruck productsInsideTruck	!= ProductsInsideTruck productsInsideTruck
		Pré- Condições	Objeto do tipo Truck Instanciado	
		Restrições Inputs	Produto passado como parâmetro não pode estar no camião	Não tem
		Pós- Condições	Retorna null	Throws IllegalArgumentException
		Resultado Esperado	null	IllegalArgumentException
		Exemplo	getProductContainerTruck ("id","dd",23.2)	getProductContainerTruck (null)

Figura 42 Tabela ECP02 getProductContainerTruck

Para o ECP01 o objetivo é testar se o método procurando um produto existente no camião.

Para o ECP02 o objetivo é testar se o método retorna o esperado caso não encontre o produto no camião.

Técnica BVA

ID	Input	Limit	Case	Value
1	productsInsideTruck	x	Special Case	null

Figura 43 Tabela BVA getProductContainerTruck

3.15 Construtor DistrictTruck

Este método cria uma estrutura para armazenar um conjunto de camiões para um distrito.

Técnica ECP

C a s e	Descrição	Critérios	Classe Válida	Classe Inválida
1	Criar um conjunto de camiões para um distrito	Nº inputs	1	!= 1
		Tipo inputs	String nameDistrict, int numberContainer, ArrayList<IOrder> ordersByDistrict	!=String nameDistrict, int numberContainer, ArrayList<IOrder> ordersByDistrict
		Pré-Condições	Não tem	
		Restrições Inputs	nameDistrict != null && nameDistrict != "" && numberContainer > 0 && ordersByDistrict != null	nameDistrict == null nameDistrict == "" numberContainer < 1 ordersByDistrict == null
		Pós-Condições	Cria um objeto do tipo DistrictTruck	Throws IllegalArgumentException
		Resultado Esperado	Instance of Truck (true)	IllegalArgumentException
		Exemplo	New DistrictTruck ("id", "dd", 23.2)	New DistrictTruck ("id", "", 23.2)

Figura 44 Tabela ECP01 construtor DistrictTruck

Técnica BVA

ID	Input	Limit	Case	Value
1	nameDistrict	x	Special Case	null
2	numberContainer	Minimum limit	x	0
3	ordersByDistrict	x	Special Case	null

Figura 45 Tabela BVA DistrictTruck

3.16 Método creatTruck

Este método cria um conjunto de camiões.

Técnica ECP

C a s e	Descrição	Crítérios	Classe Válida	Classe Inválida
1	Criar um conjunto de camiões para um distrito	Nº inputs	1	!= 1
		Tipo inputs	int numberTruck	!= int numberTruck
		Pré- Condições	Objeto do tipoTruck instanciado	
		Restrições Inputs	numberTruck > 0	numberTruck < 1
		Pós- Condições	Cria um um número de camiões	Throws IllegalArgumentException
		Resultado Esperado	<i>Instance of Truck (true)</i>	IllegalArgumentException
		Exemplo	creatTruck (3)	creatTruck (0)

Figura 46 Tabela ECP01 creatTruck

Técnica BVA

ID	Input	Limit	Case	Value
1	numberTruck	<i>Minimum limit</i>	<i>X</i>	<i>0</i>

Figura 47 Tabela BVA creatTruck

3.17 Método getTruckByIndex

Este método retorna um camião dado um índice.

Técnica ECP

C a s e	Descrição	Critérios	Classe Válida	Classe Inválida
1	Obter camião dado um índice	Nº inputs	1	!= 1
		Tipo inputs	int numberTruck	!= int numberTruck
		Pré-Condições	Objeto do tipoTruck instanciado	
		Restrições Inputs	numberTruck >= 0	numberTruck < 0
		Pós-Condições	Retorna o camião no index correspondente	Throws IndexOutOfBoundsException
		Resultado Esperado	Instance of Truck (true)	IndexOutOfBoundsException
		Exemplo	getTruckByIndex (3)	getTruckByIndex (-1)

Figura 48 Tabela ECP01 getTruckByIndex

Técnica BVA

ID	Input	Limit	Case	Value
1	numberTruck	Minimum limit	X	-1
2	numberTruck	Maximum limit	X	listTruck.size()

Figura 49 Tabela BVA getTruckByIndex

3.18 Construtor ProductsInsideTruck

Este método cria um produto que estará dentro do contentor do camião.

Técnica ECP

C a s e	Descrição	Critérios	Classe Válida	Classe Inválida
1	Criar um produto que estará dentro do contentor do camião	Nº inputs	4	!= 4
		Tipo inputs	String idOrder, String idProduct, int quantity, double uniqueSize	!= String idOrder, String idProduct, int quantity, double uniqueSize
		Pré-Condições	Objeto do tipoTruck instanciado	
		Restrições Inputs	idOrder!= null && idOrder!= "" && quantity > 0 && uniqueSize > 0 && idProduct != null && idProduct!= ""	idOrder == null idOrder == "" quantity < 1 ordersByDistrict == null idProduct == null idProduct=="" uniqueSize <= 0
		Pós-Condições	Cria um um número de camiões	Throws IllegalArgumentException
		Resultado Esperado	Instance of ProductsInsideTruck (true)	IllegalArgumentException
		Exemplo	ProductsInsideTruck productsInsideTruck = new ProductsInsideTruck("Hugo", "", 2, 34);	ProductsInsideTruck productsInsideTruck = new ProductsInsideTruck("Hugo", "", 0, 34);

Figura 50 Tabela ECP01 ProductsInsideTruck

Técnica BVA

ID	Input	Limit	Case	Value
1	idOrder	x	Special Case	null
2	idOrder	x	Special Case	empty
3	idProduct	x	Special Case	null
4	idProduct	x	Special Case	empty
5	quantity	Minimum limit	x	0
6	uniqueSize	Minimum limit	x	0.0

Figura 51 Tabela BVA ProductsInsideTruck

3.19 Método groupOrdersByTrucks

Este método recebe um conjunto de encomendas e agrupa-as, por distrito, a conjuntos de camiões.

Técnica ECP

C a s e	Descrição	Critérios	Classe Válida	Classe Inválida
1	Agrupar encomendas por distrito em camiões	Nº inputs	1	!= 1
		Tipo inputs	ArrayList<IOrder> listOrders	ArrayList<IOrder> listOrders
		Pré- Condições	Objeto do tipo ExpeditionMethods instanciado	
		Restrições Inputs	listOrders != null	listOrders != null
		Pós- Condições	Agrupa encomendas por distrito	Throws IllegalArgumentException
		Resultado Esperado	JSONArray com os resultados do agrupamento	IllegalArgumentException
		Exemplo	groupOrdersByTrucks(listOrders)	groupOrdersByTrucks(null)

Figura 52 Tabela ECP01 groupOrdersByTrucks

C a s e	Descrição	Critérios	Classe Válida	Classe Inválida
2	Agrupar encomendas com o mesmo distrito de envio e receção	Nº inputs	1	!= 1
		Tipo inputs	ArrayList<IOrder> listOrders	!=ArrayList<IOrder> listOrders
		Pré-Condições	Objeto do tipo ExpeditionMethods instanciado	
		Restrições Inputs	listOrders != null e encomendas devem ter mesmo sender e receiver	listOrders != null
		Pós-Condições	Retorna array vazio	Throws IllegalArgumentException
		Resultado Esperado	[]	IllegalArgumentException
		Exemplo	groupOrdersByTrucks(listOrders)	groupOrdersByTrucks(null)

Figura 53 Tabela ECP02 groupOrdersByTrucks

Técnica BVA

Para a tecnica BVA deve ser testado :

ID	Input	Limit	Case	Value
1	listOrders	x	<i>Special Case</i>	<i>null</i>

Figura 54 Tabela BVA groupOrdersByTrucks

Deve ser também testado:

Camiões

Camião 1 = 63 m³

Camiao 2 = 63 m³

Total: 126m³

Produtos:

Produto 1 : 62 m³

Produto 2 : 62 m³

Produto 3 : 2 m³

Total: 126m³

Embora dois camiões levem 126 m³ serão necessários 3 camiões para a escomenda, dado, que em nenhum existe espaço para o produto 3.

Deve ser estada esta situação (Teste Caixa Branca)

3.20 Método `getArrayOrdersByDistrict`

Este método retorna um *array* com as encomendas um determinado distrito de envio.

Técnica ECP

Case	Descrição	Critérios	Classe Válida	Classe Inválida
1	Obter lista de encomendas de um distrito	Nº inputs	1	!= 1
		Tipo inputs	String district, ArrayList<IOrder> listOrders	!=String district, ArrayList<IOrder> listOrders
		Pré-Condições	Objeto do tipo ExpeditionMethods instanciado	
		Restrições Inputs	district != null && district != "" && listOrders != null	district == null district == "" listOrders == null
		Pós-Condições	Retorna a lista de encomendas	Throws IllegalArgumentException
		Resultado Esperado	Arraylist de encomendas	IllegalArgumentException
		Exemplo	getArrayOrdersByDistrict ("Porto",listOrders)	getArrayOrdersByDistrict ("",listOrders)

Figura 55 Tabela ECP01 `getArrayOrdersByDistrict`

Técnica BVA

ID	Input	Limit	Case	Value
1	district	x	Special Case	""
2	district	x	Special Case	null
3	listOrders	x	Special Case	null

Figura 56 Tabela BVA `getArrayOrdersByDistrict`

3.21 Método getOrderShippingCost

Este método calcula o custo de envio de uma encomenda entre dois distritos.

Técnica ECP

C a s e	Descrição	Crítéri os	Classe Válida	Classe Inválida
1	Obter custo de envio de uma encomenda	Nº inputs	1	!= 1
		Tipo inputs	IOrder order	!= IOrder order
		Pré- Condiç ões	Objeto do tipo Orders instanciado	
		Restriç ões Inputs	order != null	order == null
		Pós- Condiç ões	Retorna o custo de envio da encomenda	Throws IllegalArgumentException
		Result ado Esper ado	<i>Custo de envio (double)</i>	IllegalArgumentException
		Exem plo	getOrderShippingCost(order)	getOrderShippingCost(null)

Figura 57 Tabela ECP getOrderShippingCost

Técnica BVA

ID	Input	Limit	Case	Value
1	order	x	<i>Special Case</i>	<i>null</i>

Figura 58 Tabela BVA getOrderShippingCost

3.22 Método addDistrictTruck

Este método adiciona um conjunto de camiões de agrupados para um distrito à lista que armazena todos

Técnica ECP

C a s e	Descrição	Critéri os	Classe Válida	Classe Inválida
1	Obter custo de envio de uma encomenda	Nº inputs	1	!= 1
		Tipo inputs	IDistrictTrucks iDistrictExpedition	!= IDistrictTrucks iDistrictExpedition
		Pré- Condiç ões	Objeto do tipo GroupAllTrucksinstanciado	
		Restriç ões Inputs	iDistrictExpedition!= null	iDistrictExpedition == null
		Pós- Condiç ões	adiciona um conjunto de camiões à lista	Throws IllegalArgumentException
		Result ado Esper ado	true	IllegalArgumentException
		Exem plo	addDistrictTruck (order)	addDistrictTruck (null)

Figura 59 Tabela ECP addDistrictTruck

Técnica BVA

ID	Input	Limit	Case	Value
1	iDistrictExpedition	x	Special Case	null

Figura 60 Tabela BVA addDistrictTruck

3.23 Método getOrderIndex

Este método retorna uma order dado um índice.

Técnica ECP

C a s e	Descrição	Crítérios	Classe Válida	Classe Inválida
1	Obter camião dado um índice	Nº inputs	1	!= 1
		Tipo inputs	int index	!= int index
		Pré- Condições	Objeto do tipo Truck instanciado	
		Restrições Inputs	numberTruck >= 0 && listOrders.size() < 0	numberTruck < 0 listOrders.size() >= 0
		Pós- Condições	Retorna a order no index correspondente	Throws IndexOutOfBoundsException
		Resultado Esperado	Instance of Order (true)	IndexOutOfBoundsException
		Exemplo	getOrderIndex (3)	getOrderIndex (-1)

Figura 61 Tabela ECP01 getOrderIndex

Técnica BVA

ID	Input	Limit	Case	Value
1	index	Minimum limit	X	-1
2	index	Maximum limit	X	listTruck.size()

Figura 62 Tabela BVA getOrderIndex

3.24 Método `getProductListProductsInsideTruck`

Este método obtém um produto de um contentor de um camião.

Técnica ECP

C a s e	Descrição	Critérios	Classe Válida	Classe Inválida
1	Obter produto de um camião através de um índice	Nº inputs	1	!= 1
		Tipo inputs	Int index	!= Int index
		Pré-Condições	Objeto do tipo Truck Instanciado, com produtos adicionados no container <code>truck.numberProductInContainer > 0</code>	
		Restrições Inputs	<code>index >= 0 && index < truck.numberProductInContainer()</code>	<code>index < 0 && index >= truck.numberProductInContainer()</code>
		Pós-Condições	Retorna o produto encontrado	Throws <code>IndexOutOfBoundsException</code>
		Resultado Esperado	<i>Instance of</i> <code>ProductsInsideTruck</code> (true)	<code>IndexOutOfBoundsException</code>
		Exemplo	<code>getProductListProductsInsideTruck(0)</code>	<code>getProductListProductsInsideTruck(-1)</code>

Figura 63 Tabela ECP `getProductListProductsInsideTruck`

Técnica BVA

ID	Input	Limit	Case	Value
1	index	<i>Minimum limit</i>	X	-1
2	index	<i>Maximum limit</i>	X	<code>truck.numberProductInContainer()</code>

Figura 64 Tabela BVA `getProductListProductsInsideTruck`

3.25 Método importCostsTableKM

Técnica ECP

C a s e	Descrição	CrITÉRIOS	Classe Válida	Classe Inválida
1	Importar tabela das distâncias	Nº inputs	1	!= 1
		Tipo inputs	String path	!=String path
		Pré- Condições	ImportDistanceTableKM instanciado	
		Restrições Inputs	path != "" && path!= null	path == "" path== null
		Pós- Condições	Importa tabela	Throws IllegalArgumentException
		Resultado Esperado	Valores das distância entre distritos	IllegalArgumentException
		Exemplo	importCostsTableKm("drivers/drivers.json")	importCostsTableKm (null)

Figura 65 Tabela ECP importCostsTableKM

Técnica BVA

ID	Input	Case	Value
1	path	<i>Special Case</i>	<i>null</i>
2	path	<i>Special Case</i>	<i>""</i>

Figura 66 Tabela BVA importCostsTableKM

Nota: Ficheiro deve corresponder a um formato válido para a estrutura que se pretende carregar.

3.26 Construtor DistanceLine

Técnica ECP

C a s e	Descrição	CrITÉRIOS	Classe Válida	Classe Inválida
1	Importar tabela das distâncias	Nº inputs	1	!= 1
		Tipo inputs	String name, int value	!= String name, int value
		Pré- Condições	Não tem	
		Restrições Inputs	name!= "" && name!= null && value >= 0	name == "" name == null value < 0
		Pós- Condições	Cria uma linha da tabela distâncias	Throws IllegalArgumentException
		Resultado Esperado	Objeto do tipo DistanceLine criado	IllegalArgumentException
		Exemplo	DistanceLine ("Porto",4)	DistanceLine (null,5)

Figura 67 Tabela ECP DistanceLine

Técnica BVA

ID	Input	Case	Value
1	name	Special Case	null
2	name	Special Case	""
3	value	Limit minimum	-1

Figura 68 Tabela BVA DistanceLine

4. Identificação dos Testes

Método averageNumberProductsTransaction

Nome	Tecnica
testAverageValueTransactionsECP01	ECP
testAverageValueTransactionsBVA01	BVA

Método averageValueTransactions

Nome	Tecnica
testAverageNumberProductsTransactionECP01	ECP
testAverageNumberProductsTransactionBVA01	BVA

Método averageValueSalesPurchasesDistrict

Nome	Tecnica
testAverageValueSalesPurchasesDistrict ECP01	ECP
testAverageValueSalesPurchasesDistrict ECP02	ECP
testAverageValueSalesPurchasesDistrict BVA01	BVA
testAverageValueSalesPurchasesDistrict BVA02	BVA

Método registerOrdersInLedger

Nome	Tecnica
testRegisterOrdersInLedgerBVA01	BVA
testRegisterOrdersInLedgerBVA02	BVA
testRegisterOrdersInLedgerBVA03	BVA
testRegisterOrdersInLedger ECP01	ECP
testRegisterOrdersInLedger ECP02	ECP

Método creatFile

Nome	Tecnica
testCreateFileBVA01	BVA
testCreateFileBVA02	BVA
TestCreateFileECP01	ECP

Método addProduct

Nome	Tecnica
testAddOrderBVA01	BVA
testAddProductECP01	ECP
testAddProductECP02	ECP

Método standardDeviationTransaction

Nome	Tecnica
testStandardDeviationTransactions BVA01	BVA
testStandardDeviationTransactionsECP01	ECP

Método standardDeviationNumberProductTransactions

Nome	Tecnica
testStandardDeviationNumberProductTransactionsBVA01	BVA
testStandardDeviationNumberProductTransactionsECP01	ECP

Método numberOrdersByDistrict

Nome	Tecnica
testNumberOrdersByDistrictBVA01	BVA
testNumberOrdersByDistrictECP01	ECP
testNumberOrdersByDistrictECP02	ECP

Construtor Product

Nome	Tecnica
testBasicProductBVA01	BVA
testBasicProductBVA02	BVA
testBasicProductBVA03	BVA
testBasicProductBVA04	BVA
testBasicProductBVA05	BVA
testBasicProductBVA06	BVA
testBasicProductBVA07	BVA
testBasicProductBVA08	BVA
testBasicProductBVA09	BVA
testBasicProductBVA10	BVA
testBasicProductECP01	ECP

Construtor Truck

Nome	Tecnica
testTruckBVA01	BVA
testTruckBVA02	BVA
testTruckECP01	ECP

Método addProductInTruck

Nome	Tecnica
testAddProductInTruckBVA01	BVA
testAddProductInTruckBVA02	BVA
testAddProductInTruckBVA03	BVA
testAddProductInTruckBVA04	BVA

testAddProductInTruckBVA05	BVA
TestAddProductInTruckECP01	ECP

Método addAllOrderInTruck

Nome	Tecnica
testAddAllOrderInTruckBVA01	BVA
testAddAllOrderInTruckBVA02	BVA
testAddAllOrderInTruckBVA03	BVA
testAddAllOrderInTruckECP01	ECP

Método getProductContainerTruck

Nome	Tecnica
testGetProductContainerTruckBVA01	BVA
testGetProductContainerTruckECP01	ECP
testGetProductContainerTruckECP02	ECP

Método getTruckByIndex

Nome	Tecnica
testGetTruckByIndexBV01	BVA
testGetTruckByIndexBV02	BVA
testGetTruckByIndexECP01	ECP

Construtor DistrictTrucks

Nome	Tecnica
testDistrictTrucksBVA01	BVA
testDistrictTrucksBVA02	BVA
testDistrictTrucksECP01	ECP

Método CreateTruck

Nome	Tecnica
testCreateTruckBVA01	BVA
testCreateTruckECP01	ECP

Método getOrderShippingCost

Nome	Tecnica
testGetOrderShippingCostBVA01	BVA
testGetOrderShippingCostECP01	ECP
testGetOrderShippingCostECP02	ECP

Método getOrder

Nome	Tecnica
testGetOrderBVA01	BVA
TestGetOrderECP01	ECP
TestGetOrderECP02	ECP

Construtor ProductsInsideTruck

Nome	Tecnica
testProductsInsideTruckBVA01	BVA
testProductsInsideTruckBVA02	BVA
testProductsInsideTruckBVA03	BVA
testProductsInsideTruckBVA04	BVA
testProductsInsideTruckBVA05	BVA
testProductsInsideTruckBVA06	BVA
testProductsInsideTruckECP01	ECP

Método getArrayOrdersByDistrict

Nome	Tecnica
testGetArrayOrdersByDistrictBVA01	BVA
testGetArrayOrdersByDistrictBVA02	BVA
testGetArrayOrdersByDistrictBVA03	BVA
testGetArrayOrdersByDistrictECP01	ECP

Método testGroupOrders

Nome	Tecnica
testGroupOrdersECP01	ECP
testGroupOrdersECP02	ECP
testGroupOrdersByTrucksBVA01	BVA

Método addDistrictTrucks

Nome	Tecnica
testAddDistrictTrucksBVA01	BVA
testAddDistrictTrucksECP01	ECP

Método importOrders

Nome	Tecnica
testImportOrdersBVA01	BVA
testImportOrdersBVA02	BVA
testImportOrdersECP01	ECP
testImportOrdersECP02	ECP

Método exportGroupOrdersForFileJson

Nome	Tecnica
testExportGroupOrdersForFileJsonECP01	ECP
testExportGroupOrdersForFileJsonECP02	ECP

Método AddDistrictTrucks

Nome	Tecnica
testAddDistrictTrucksBVA01	BVA
testAddDistrictTrucksECP01	ECP

Método getOrderIndex

Nome	Tecnica
testGetOrderIndexBVA01	BVA
testGetOrderIndexBVA02	BVA
testGetOrderIndexECP01	ECP

Método getProductListProductsInsideTruck

Nome	Tecnica
testGetProductListProductsInsideTruckBVA01	BVA
testGetProductListProductsInsideTruckBVA02	BVA
TestGetProductListProductsInsideTruckECP01	ECP

Método importCostsTableKM

Nome	Tecnica
testImportOrdersBVA01	BVA
testImportOrdersBVA02	BVA
testImportOrdersECP01	ECP
testImportOrdersECP02	ECP

Construtor DistanceLine

Nome	Tecnica
testBasicDistanceBVA01	BVA
testBasicDistanceBVA01	BVA
testBasicDistanceECP01	ECP

Construtor DistrictDistance

Nome	Tecnica
testDistrictDistanceBVA01	BVA
testDistrictDistanceECP01	ECP

Método getDistance

Nome	Tecnica
testGetDistanceBVA01	BVA
testGetDistanceECP01	ECP
testGetDistanceECP02	ECP

Método AddDistance

Nome	Tecnica
testGetDistanceBVA01	BVA
testGetDistanceECP01	ECP
testGetDistanceECP02	ECP

Método importConfigsTruck

Nome	Tecnica
testImportConfigCapacityBVA01	BVA
testImportConfigCapacityECP01	ECP

5. Critérios de passagem ou falha das features

Classe	Nome	Passagem(Resultado obtido)	Falha(Resultado obtido)
OS	testAverageNumberProductsTransactionBVA01	AverageException	!= AverageException
OS	testAverageNumberProductsTransactionECP01	18.33	!= 18.33
OS	testAverageValueTransactionsECP01	9	!= 9
OS	testAverageValueTransactionsBVA01	AverageException	!= AverageException
OS	testAverageValueSalesPurchasesDistrictECP01	map	!=map
OS	testAverageValueSalesPurchasesDistrictECP02	map	!=map
OS	testRegisterOrdersInLedgerBVA01	0	!=0
OS	testRegisterOrdersInLedgerBVA02	2	!=2
OS	testRegisterOrdersInLedgerBVA03	2	!=2
OS	testRegisterOrdersInLedger ECP01	2	!=2
OS	testRegisterOrdersInLedger ECP02	0	!=0
OS	testGetOrderBVA01	IllegalArgumentException	!= IllegalArgumentException
OS	testGetOrderECP01	Instance of Order (true)	!=Instance of Order (true)
OS	testGetOrderECP02	null	!=null
OS	testAddOrderBVA01	IllegalArgumentException	!= IllegalArgumentException
OS	testAddProductECP01	true	!= true
OS	testAddProductECP02	false	!=false
OS	testStandardDeviationTransactions BVA01	IllegalArgumentException	!= IllegalArgumentException
OS	testStandardDeviationTransactionsECP01	24.53	!=24.53
OS	testNumberOrdersByDistrictBVA01	IllegalArgumentException	!= IllegalArgumentException
OS	testNumberOrdersByDistrictECP01	map	!=map
OS	testNumberOrdersByDistrictECP02	map	!=map
OS	testStandardDeviationNumberProductTransactionsBVA01	IllegalArgumentException	!= IllegalArgumentException
OS	testStandardDeviationNumberProductTransactionsECP01	0.82	!=0.82
SE	testCreateFileBVA01	IllegalArgumentException	!= IllegalArgumentException
SE	testCreateFileBVA02	IllegalArgumentException	!= IllegalArgumentException
SE	TestCreateFileECP01	true	!= true
BO	testAddProductECP02	false	!=false
BO	testAddProductECP01	true	!= true
BO	testGetProductECP02	null	!=null

BO	testGetProductECP01	<i>Instance of Product (true)</i>	<i>Instance of Product (true)</i>
BO	testGetProductBVA01	IllegalArgumentException	!= IllegalArgumentException
BO	testBasicOrderBVA01	IllegalArgumentException	!= IllegalArgumentException
BO	testBasicOrderBVA02	IllegalArgumentException	!= IllegalArgumentException
BO	testBasicOrderBVA03	IllegalArgumentException	!= IllegalArgumentException
BO	testBasicOrderBVA04	IllegalArgumentException	!= IllegalArgumentException
BO	testBasicOrderECP01	<i>Instance of Order (true)</i>	<i>!=Instance of Order (true)</i>
BO	shippingCostCalculationECP01	16.91	!=16.91
BO	shippingCostCalculationECP02	0	!=0
BO	testGetOrderIndexBVA01	IllegalArgumentException	!= IllegalArgumentException
BO	testGetOrderIndexBVA02	IllegalArgumentException	!= IllegalArgumentException
BO	testGetOrderIndexECP01	<i>Instance of Order (true)</i>	<i>!=Instance of Order (true)</i>
P	testBasicProductBVA01	IllegalArgumentException	!= IllegalArgumentException
P	testBasicProductBVA02	IllegalArgumentException	!= IllegalArgumentException
P	testBasicProductBVA03	IllegalArgumentException	!= IllegalArgumentException
P	testBasicProductBVA04	IllegalArgumentException	!= IllegalArgumentException
P	testBasicProductBVA05	IllegalArgumentException	!= IllegalArgumentException
P	testBasicProductBVA06	IllegalArgumentException	!= IllegalArgumentException
P	testBasicProductBVA07	IllegalArgumentException	!= IllegalArgumentException
P	testBasicProductBVA08	IllegalArgumentException	!= IllegalArgumentException
P	testBasicProductBVA09	IllegalArgumentException	!= IllegalArgumentException
P	testBasicProductBVA10	IllegalArgumentException	!= IllegalArgumentException
T	testTruckBVA01	IllegalArgumentException	!= IllegalArgumentException
T	testTruckBVA02	IllegalArgumentException	!= IllegalArgumentException
T	testTruckECP01	<i>Instace of Truck (true)</i>	<i>!= Instace of Truck (true)</i>

T	testAddProductInTruckBVA01	IllegalArgumentException	!= IllegalArgumentException
T	testAddProductInTruckBVA02	IllegalArgumentException	!= IllegalArgumentException
T	testAddProductInTruckBVA03	IllegalArgumentException	!= IllegalArgumentException
T	testAddProductInTruckBVA04	IllegalArgumentException	!= IllegalArgumentException
T	testAddProductInTruckBVA05	IllegalArgumentException	!= IllegalArgumentException
T	testAddProductInTruckECP01	true	!= true
T	testAddAllOrderInTruckBVA01	IllegalArgumentException	!= IllegalArgumentException
T	testAddAllOrderInTruckBVA02	false	!= false
T	testAddAllOrderInTruckBVA03	IllegalArgumentException	!= IllegalArgumentException
T	testAddAllOrderInTruckECP01	true	!= true
T	testGetProductContainerTruckBVA01	IllegalArgumentException	!= IllegalArgumentException
T	testGetProductContainerTruckECP01	Instance of ProductsInsideTruck (true)	!= Instance of ProductsInsideTruck (true)
T	testGetProductListProductsInsideTruckBVA01	IndexOutOfBoundsException	!=IndexOutOfBoundsException
T	testGetProductListProductsInsideTruckBVA02	IndexOutOfBoundsException	!=IndexOutOfBoundsException
T	TestGetProductListProductsInsideTruckECP01	Instance of ProductsInsideTruck (true)	!= Instance of ProductsInsideTruck (true)
T	testGetProductContainerTruckECP02	null	!= null
DT	testCreateTruckBVA01	IllegalArgumentException	!= IllegalArgumentException
DT	testCreateTruckECP01	3	!=3
DT	testCreateTruckECP02	IllegalArgumentException	!= IllegalArgumentException
DT	testDistrictTrucksBVA01	IllegalArgumentException	!= IllegalArgumentException
DT	testDistrictTrucksBVA02	IllegalArgumentException	!= IllegalArgumentException
DT	testDistrictTrucksECP01	Instance of Truck (true)	!=Instance of Truck (true)
DT	testGetTruckByIndexBV01	IndexOutOfBoundsException	!=IndexOutOfBoundsException
DT	testGetTruckByIndexBV02	IndexOutOfBoundsException	!=IndexOutOfBoundsException
DT	testGetTruckByIndexECP01	Instance of Truck (true)	!=Instance of Truck (true)
PT	testProductsInsideTruckBVA01	IllegalArgumentException	!= IllegalArgumentException
PT	testProductsInsideTruckBVA02	IllegalArgumentException	!= IllegalArgumentException
PT	testProductsInsideTruckBVA03	IllegalArgumentException	!= IllegalArgumentException
PT	testProductsInsideTruckBVA04	IllegalArgumentException	!= IllegalArgumentException

PT	testProductsInsideTruckBVA05	IllegalArgumentException	!= IllegalArgumentException
PT	testProductsInsideTruckBVA06	IllegalArgumentException	!= IllegalArgumentException
PT	testProductsInsideTruckECP01	<i>Instance of ProductsInsideTruck (true)</i>	<i>!=Instance of ProductsInsideTruck (true)</i>
EM	testGroupOrdersECP01	<i>jsonarray.size > 0</i>	<i>!= jsonarray.size > 0</i>
EM	testGroupOrdersECP02	<i>[]</i>	<i>!=[]</i>
EM	testGroupOrdersByTrucksBVA01	IllegalArgumentException	!= IllegalArgumentException
EM	testGetArrayOrdersByDistrictBVA01	IllegalArgumentException	!= IllegalArgumentException
EM	testGetArrayOrdersByDistrictBVA02	IllegalArgumentException	!= IllegalArgumentException
EM	testGetArrayOrdersByDistrictBVA03	IllegalArgumentException	!= IllegalArgumentException
EM	testGetArrayOrdersByDistrictECP01	2	!=2
EM	testExportGroupOrdersForFileJsonECP01	true	!= true
EM	testExportGroupOrdersForFileJsonECP02	false	!= false
IO	testImportOrdersBVA01	IllegalArgumentException	!= IllegalArgumentException
IO	testImportOrdersBVA02	IllegalArgumentException	!= IllegalArgumentException
IO	testImportOrdersECP01	null	!= null
IO	testImportOrdersECP02	true	!= true
GO	testAddDistrictTrucksBVA01	IllegalArgumentException	!= IllegalArgumentException
GO	testAddDistrictTrucksECP01	true	!= true
DD	testDistrictDistanceBVA01	IllegalArgumentException	!= IllegalArgumentException
DD	testDistrictDistanceECP01	<i>Instance of DistricDistance (true)</i>	<i>!=Instance of DistricDistance (true)</i>
DD	testGetDistanceBVA01	IllegalArgumentException	!= IllegalArgumentException
DD	testGetDistanceECP01	true	!= true
DD	testGetDistanceECP02	false	!= false

Legendas Classes:

T -> Truck;

IO->ImportOrdes;

EM->Expedition Methods;

PT->Product Inside Truck;

DT->District Trucks;

GO->GroupAllTrucks;

BO-> Order;

DD -> DistrictDistance

OS->OrdersMetricStatistics;

P-> Product;