

TRABALHO PRÁTICO 2 - ESII

TEST CASE OUTLINE

Version 6.0
21/01/2022

Histórico de Versões

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Bruno Ferreira Gonçalo Oliveira Jorge Correia Nuno Castro	13-01-2022	Bruno Ferreira Gonçalo Oliveira Jorge Correia Nuno Castro	14-01-2022	Análise testes caixa preta para o módulo transações
2.0	Bruno Ferreira Gonçalo Oliveira Jorge Correia Nuno Castro	15-01-2022	Bruno Ferreira Gonçalo Oliveira Jorge Correia Nuno Castro	15-01-2022	Adição dos teste white box para o módulo transações
3.0	Bruno Ferreira Gonçalo Oliveira Jorge Correia Nuno Castro	15-01-2022	Bruno Ferreira Gonçalo Oliveira Jorge Correia Nuno Castro	15-01-2022	Execução dos Testes que falharam, após a correção do código
4.0	Bruno Ferreira Gonçalo Oliveira Jorge Correia Nuno Castro	17-01-2022	Bruno Ferreira Gonçalo Oliveira Jorge Correia Nuno Castro	17-01-2022	Análise testes caixa preta para o módulo de custos de envio
5.0	Bruno Ferreira Gonçalo Oliveira Jorge Correia Nuno Castro	18-01-2022	Bruno Ferreira Gonçalo Oliveira Jorge Correia Nuno Castro	18-01-2022	Análise testes caixa branca para o módulo de custos de envio
6.0	Bruno Ferreira Gonçalo Oliveira Jorge Correia Nuno Castro	21-01-2022	Bruno Ferreira Gonçalo Oliveira Jorge Correia Nuno Castro	21-01-2022	Análise testes caixa preta para o módulo de expedições

Índice

1. Introdução	4
1.1. Identificador do documento	4
1.2. Âmbito	4
1.3. Repositório	4
1.4. Referências	4
1.5 Glossário	4
2. Detalhes	6
2.1 Classe BasicTransactionEncomenda	6
2.1.1 addTransactionLine	6
2.2 Classe BasicGestaoEncomenda	9
2.2.1 addEncomenda	9
2.2.2 cancelarEncomenda	12
2.2.3 findEncomenda	16
2.3 Classe Estatísticas	19
2.3.1 getNumMedioEncomendasPorDia	19
2.3.2 getNumMedioProdutosPorTransacao	23
2.3.3 getValorMedioTransacoes	28
2.3.4 getTotalCustosEnvio	31
2.3.5 getValorMedioVendasComprasPorDistrito	39
2.4 Classe BasicCustos	43
2.4.1 calculaCustoEnvio	43
2.5 Classe Expedicoes	46
2.5.1 assignEncomendas	46

1. Introdução

1.1. Identificador do documento

TestCase_Outline_Grupo1213

1.2. Âmbito

Este relatório de Test Case Outline está a ser realizado no âmbito do Trabalho Prático 2 da disciplina de Engenharia de Software II em que, o objetivo deste mesmo relatório é essencialmente apresentar e comparar os resultados esperados dos casos de teste, especificados no Test Case Specification, com os resultados obtidos, bem como, todos os procedimentos a realizar para a testagem.

Tal como indicado no relatório Test Case Specification, ao longo do desenvolvimento deste módulo foram selecionados os métodos identificados como cruciais para o bom funcionamento da aplicação e foram devidamente testados através de técnicas de Equivalence Class Partitioning e Boundary Value Analysis.

1.3. Repositório

A hiperligação para o repositório onde foi realizado o trabalho é a seguinte:

<https://gitlab.estg.ipp.pt/esii.grupo1213/esii-grupo1213-tp2>

1.4. Referências

Como referência para a realização deste documento foram usados documentos como:

- Slides disponibilizados no Moodle da Unidade Curricular
- IEEE Standard for Software Unit Testing

1.5 Glossário

Testes de software - Um teste de software é um software que executa outro software, validando se os resultados são os esperados (teste de estado) ou se executa a sequência de eventos esperado (teste de comportamento).

Resumidamente, os testes de software permitem aos programadores verificar se a lógica do programa desenvolvido está de acordo com os requisitos.

A execução automática de testes permite identificar “bugs” resultantes de mudanças no código fonte.

ECP - Técnica destinada a reduzir o número de testes necessários dividindo o domínio de entrada (ou saída) em classes de dados em que os casos de teste podem ser derivados

para cada operação, o “tester” deve identificar as classes de equivalência dos argumentos e os estados dos objetos

BVA - Técnica baseada na observação de bugs que ocorrem frequentemente em valores fronteira. É focada em testar valores especiais (null, 0, etc) e limites do domínio de entrada (ou saída) imediatamente acima e abaixo (além de ou em vez de valores intermédios)

JUnit - Framework open-source para realizar testes unitários para código em Java. Permite a execução de testes de forma:

- Fácil
- Regular
- Fiável

Contém várias funcionalidades para testing entre elas a capacidade de testar cada componente de um programa de forma independente do resto do programa

Gradle - É uma ferramenta que permite integrar e automatizar várias tarefas relacionadas processo de desenvolvimento software em várias linguagens de programação.

O Gradle determina quais os componentes do projeto que estão atualizados, evitando a recompilação de todo o projeto.

2. Detalhes

2.1 Classe BasicTransactionEncomenda

2.1.1 addTransactionLine

Test Case ID	1	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12/01/2022	22:53	Nuno Castro	Pass	

Pré-condições / estado inicial				
<ol style="list-style-type: none">1. Criar duas entidades que vão ser o sender e o receiver de uma transação.2. Criar uma transação do tipo Envio com as duas entidades criadas anteriormente.3. Criar um produto				
Procedimento / Passos do user para esta função				
<ol style="list-style-type: none">1. Adicionar uma linha à transação criada no ponto 2 que deverá possuir o produto e a respetiva quantidade desejada.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: -- Use Case: Módulo Transações Funcionalidade: addTransactionLine() ne()	*linhaTransacao (TransactionLine)	linhaTransacao: BasicLinhaTransacao	true	Pass

Test Case ID	2	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12/01/2022	22:53	Nuno Castro	Pass	

Pré-condições / estado inicial				
<ol style="list-style-type: none"> 1. Criar duas entidades que vão ser o sender e o receiver de uma transação. 2. Criar uma transação do tipo Envio com as duas entidades criadas anteriormente. 3. Criar um produto 4. Adicionar uma linha de transação com o produto criado anteriormente e adicioná-la à transação. 				
Procedimento / Passos do user para esta função				
<ol style="list-style-type: none"> 1. Adicionar na transação criada uma linha de transação que já se encontra inserida, por exemplo a criada no caso anterior. 				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: -- Use Case: Módulo Transações Funcionalidade: addTransactionLine()	*linhaTransacao (TransactionLine)	linhaTransacao: BasicLinhaTransacao w	false	Pass

Test Case ID	3	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12/01/2022	22:53	Nuno Castro	Pass	

Pré-condições / estado inicial				
<ol style="list-style-type: none"> 1. Criar duas entidades que vão ser o sender e o receiver de uma transação. 2. Criar uma transação do tipo Envio com as duas entidades criadas anteriormente. 3. Criar um produto 				
Procedimento / Passos do user para esta função				
<ol style="list-style-type: none"> 1. Criar uma linha de transação a partir da classe TransactionLine, ou seja, cujo método addTransactionLine apenas possui uma descrição do produto, quantidade e preço unitário- 2. Adicionar essa linha de transação à transação 				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: -- Use Case: Módulo Transações Funcionalidade: addTransactionLine()	*linhaTransacao (TransactionLine)	linhaTransacao: TransactionLine	Throw RunTimeException	Pass

Test Case ID	4	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12/01/2022	22:53	Nuno Castro	Pass	

Pré-condições / estado inicial				

Procedimento / Passos do user para esta função				
1. Adicionar a uma transação null				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: -- Use Case: Módulo Transações Funcionalidade: addTransactionLine()	*linhaTransacao (TransactionLine)	linhaTransacao: null	Throw RunTimeException	Pass

Test Case ID	5	Dependências	
--------------	---	--------------	--

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

Pré-condições / estado inicial				

Procedimento / Passos do user para esta função				

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: -- Use Case: Módulo Transações Funcionalidade: addTransactionLine()	*linhaTransacao (TransactionLine)	linhaTransacao: "linhaTransacao"	Erro de Sintaxe	Untested

2.2 Classe BasicGestaoEncomenda

2.2.1 addEncomenda

Test Case ID	1	Dependências	findEncomenda()
--------------	---	--------------	-----------------

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12/01/2022	22:51	Bruno Ferreira	Pass	

Pré-condições / estado inicial				
<ol style="list-style-type: none">1. Criar duas entidades que vão ser o sender e o receiver de uma transação.2. Criar 1 produto3. Criar uma transação do tipo envio.4. Adicionar produtos à transação5. Criar uma encomenda				
Procedimento / Passos do user para esta função				
<ol style="list-style-type: none">1. Adicionar a encomenda que foi criada anteriormente que ainda não foi adicionada ao sistema				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Adicionar encomenda Use Case: Módulo Transações Funcionalidade: addEncomenda()	*encomenda (Encomenda)	encomenda: BasicEncomenda	true	Pass

Test Case ID	2	Dependências	findEncomenda()
--------------	---	--------------	-----------------

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12/01/2022	22:51	Bruno Ferreira	Pass	

Pré-condições / estado inicial				
<ol style="list-style-type: none"> 1. Criar duas entidades que vão ser o sender e o receiver de uma transação. 2. Criar três produtos 3. Criar uma transação do tipo envio. 4. Adicionar produtos à transação 5. Criar uma encomenda 6. Adicionar a encomenda ao sistema de gestão. 				
Procedimento / Passos do user para esta função				
1. Adicionar a encomenda criada, sendo que esta já foi adicionada no sistema.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Adicionar encomenda Use Case: Módulo Transações Funcionalidade: addEncomenda()	*encomenda (Encomenda)	encomenda: BasicEncomenda	false	Pass

Test Case ID	3	Dependências	findEncomenda()
--------------	---	--------------	-----------------

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12/01/2022	22:51	Bruno Ferreira	Pass	

Pré-condições / estado inicial				
<ol style="list-style-type: none"> 1. Criar duas entidades que vão ser o sender e o receiver de uma transação. 2. Criar três produtos 3. Criar uma transação do tipo envio. 4. Adicionar produtos à transação 5. Criar uma encomenda 				
Procedimento / Passos do user para esta função				
1. Adicionar uma encomenda null				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Adicionar encomenda Use Case: Módulo Transações Funcionalidade: addEncomenda()	*encomenda (Encomenda)	encomenda: null	Throw IllegalArgumentException	Pass

Test Case ID	4	Dependências	findEncomenda()
---------------------	---	---------------------	-----------------

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

Pré-condições / estado inicial
<ol style="list-style-type: none"> 1. Criar duas entidades que vão ser o sender e o receiver de uma transação. 2. Criar três produtos 3. Criar uma transação do tipo envio. 4. Adicionar produtos à transação 5. Criar uma encomenda

Procedimento / Passos do user para esta função
<ol style="list-style-type: none"> 1. Enviar uma String, ou seja, enviar como parâmetro um valor inválido.

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Adicionar encomenda Use Case: Módulo Transações Funcionalidade: addEncomenda()	*encomenda (Encomenda)	encomenda: "encomenda"	Erro de Sintaxe	Untested

2.2.2 cancelarEncomenda

Test Case ID	1	Dependências	findEncomenda()	
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12/01/2022	23:07	Jorge Correia	Pass	
Pré-condições / estado inicial				
1. Criar duas entidades que vão ser o sender e o receiver de uma transação. 2. Criar três produtos 3. Criar uma transação do tipo envio. 4. Adicionar produtos à transação 5. Criar uma encomenda 6. Adicionar encomenda ao sistema de gestão				
Procedimento / Passos do user para esta função				
1. Cancelar a encomenda criada anteriormente que já foi adicionada ao sistema.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Cancelar encomenda Use Case: Módulo Transações Funcionalidade: cancelarEncomenda()	*encomenda (Encomenda)	encomenda: BasicEncomenda	true	Pass

Test Case ID	2	Dependências	findEncomenda()
--------------	---	--------------	-----------------

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12/01/2022	23:07	Jorge Correia	Pass	

Pré-condições / estado inicial				
1. Criar duas entidades que vão ser o sender e o receiver de uma transação. 2. Criar três produtos 3. Criar uma transação do tipo envio. 4. Adicionar produtos à transação 5. Criar uma encomenda				
Procedimento / Passos do user para esta função				
1. Cancelar a encomenda que foi criada anteriormente que ainda não foi adicionada.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Cancelar encomenda Use Case: Módulo Transações Funcionalidade: cancelarEncomenda()	*encomenda (Encomenda)	encomenda: BasicEncomenda	false	Pass

Test Case ID	3	Dependências	findEncomenda()
---------------------	---	---------------------	-----------------

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12/01/2022	23:07	Jorge Correia	Pass	

Pré-condições / estado inicial
<ol style="list-style-type: none"> 1. Criar duas entidades que vão ser o sender e o receiver de uma transação. 2. Criar três produtos 3. Criar uma transação do tipo envio. 4. Adicionar produtos à transação 5. Criar uma encomenda

Procedimento / Passos do user para esta função
1. Cancelar uma encomenda nula

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Cancelar encomenda Use Case: Módulo Transações Funcionalidade: cancelarEncomenda()	*encomenda (Encomenda)	encomenda: null	Throw RuntimeException	Pass

Test Case ID	4	Dependências	findEncomenda()
---------------------	---	---------------------	-----------------

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

Pré-condições / estado inicial
<ol style="list-style-type: none"> 1. Criar duas entidades que vão ser o sender e o receiver de uma transação. 2. Criar três produtos 3. Criar uma transação do tipo envio. 4. Adicionar produtos à transação 5. Criar uma encomenda

Procedimento / Passos do user para esta função
<ol style="list-style-type: none"> 1. Enviar como parâmetro uma string, ou seja, um parâmetro inválido.

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Cancelar encomenda Use Case: Módulo Transações Funcionalidade: cancelarEncomenda() 	*encomenda (Encomenda)	encomenda: “encomenda5”	Erro de Sintaxe	Untested

2.2.3 findEncomenda

Test Case ID	1	Dependências	
---------------------	---	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12/01/2022	23:19	Gonçalo Oliveira	Pass	

Pré-condições / estado inicial				
1. Criar duas entidades que vão ser o sender e o receiver de uma transação. 2. Criar dois produtos 3. Criar uma transação do tipo envio. 4. Adicionar produtos à transação 5. Criar uma encomenda 6. Adicionar essa encomenda ao sistema de gestão.				
Procedimento / Passos do user para esta função				
1. Enviar o id da encomenda criada que já foi adicionada.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: -- Use Case: -- Funcionalidade: findEncomenda()	*idencomenda (String)	idencomenda: "20220112_01"	Encomenda criada	Pass

Test Case ID	2	Dependências	
---------------------	---	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12/01/2022	23:19	Gonçalo Oliveira	Pass	

Pré-condições / estado inicial				
1. Criar duas entidades que vão ser o sender e o receiver de uma transação. 2. Criar dois produtos 3. Criar uma transação do tipo envio. 4. Adicionar produtos à transação 5. Criar uma encomenda				
Procedimento / Passos do user para esta função				
1. Enviar o id da encomenda criada que ainda não foi adicionada.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: -- Use Case: -- Funcionalidade: findEncomenda()	*idencomenda (String)	idencomenda: "20220112_100"	null	Pass

Test Case ID	3	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12/01/2022	23:19	Gonçalo Oliveira	Pass	

Pré-condições / estado inicial				
1. Criar duas entidades que vão ser o sender e o receiver de uma transação. 2. Criar dois produtos 3. Criar uma transação do tipo envio. 4. Adicionar produtos à transação 5. Criar uma encomenda				
Procedimento / Passos do user para esta função				
1. Enviar como parâmetro uma String vazia				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: -- Use Case: -- Funcionalidade: findEncomenda()	*idencomenda (String)	idencomenda: ""	null	Pass

Test Case ID	4	Dependências	
--------------	---	--------------	--

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

Pré-condições / estado inicial				
1. Criar duas entidades que vão ser o sender e o receiver de uma transação. 2. Criar dois produtos 3. Criar uma transação do tipo envio. 4. Adicionar produtos à transação 5. Criar uma encomenda				
Procedimento / Passos do user para esta função				
1. Enviar como parâmetro um inteiro.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: -- Use Case: -- Funcionalidade: findEncomenda()	*idencomenda (String)	idencomenda: 123	Erro de Sintaxe	Untested

Test Case ID	5	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12/01/2022	23:19	Gonçalo Oliveira	Pass	

Pré-condições / estado inicial				
<ol style="list-style-type: none"> 1. Criar duas entidades que vão ser o sender e o receiver de uma transação. 2. Criar dois produtos 3. Criar uma transação do tipo envio. 4. Adicionar produtos à transação 5. Criar uma encomenda 				
Procedimento / Passos do user para esta função				
<ol style="list-style-type: none"> 1. Enviar um id de encomenda null. 				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Un tested
Requisito: -- Use Case: -- Funcionalidade: findEncomenda()	*idcomenda (String)	idcomenda: null	Throw IllegalArgumentException	Pass

2.3 Classe Estatísticas

2.3.1 getNumMedioEncomendasPorDia

Test Case ID	1	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
15/01/2022	02:29	Gonçalo Oliveira	Pass	

Pré-condições / estado inicial

1. Criar duas entidades que vão ser o sender e o receiver de uma transação.
2. Criar dois produtos
3. Criar duas transações do tipo envio.
4. Adicionar um produto a cada linha de transação
5. Adicionar uma linha de transação a cada transação
6. Criar duas encomendas

Procedimento / Passos do user para esta função

1. Criar uma encomenda com com a data: 2022-01-02
2. Adicionar as três encomendas
3. Enviar o iterador das encomendas

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Calcular o número médio de encomendas por dia Use Case: Módulo Transações Funcionalidade: getNumMedioEncomendasPorDia() ()	* encomendas (Iterator<Encomenda>)	encomendas: iterador para encomendas.	0.21	Pass

Test Case ID	2	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
15/01/2022	02:29	Gonçalo Oliveira	Pass	

Pré-condições / estado inicial				
<ol style="list-style-type: none"> 1. Criar duas entidades que vão ser o sender e o receiver de uma transação. 2. Criar um produto 3. Criar uma transação do tipo envio. 4. Adicionar o produto à linha de transação 5. Adicionar a linha de transação à transação 6. Criar uma encomenda 				
Procedimento / Passos do user para esta função				
<ol style="list-style-type: none"> 1. Adicionar uma encomenda 2. Enviar o iterador das encomendas 				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Calcular o número médio de encomendas por dia Use Case: Módulo Transações Funcionalidade: getNumMedioEncomendasPorDia() ()	* encomendas (Iterator<Encomenda>)	encomendas: iterador para encomendas.	1	Pass

Test Case ID	3	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12/01/2022	23:26	Gonçalo Oliveira	Pass	

Pré-condições / estado inicial				

Procedimento / Passos do user para esta função				
3. Enviar no lugar do iterator das encomendas o valor null.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Calcular o número médio de encomendas por dia Use Case: Módulo Transações Funcionalidade: getNumMedioEncomendasPorDia() ()	* encomendas (Iterator<Encomenda>)	encomendas: null	Throw RuntimeException	Pass

Test Case ID	4	Dependências	
--------------	---	--------------	--

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

Pré-condições / estado inicial				

Procedimento / Passos do user para esta função				
1. Enviar no lugar do Iterator<Encomenda> uma string				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Calcular o número médio de encomendas por dia Use Case: Módulo Transações Funcionalidade: getNumMedioEncomendasPorDia() 	* encomendas (Iterator<Encomenda>)	encomendas: "encomenda"	Erro de Sintaxe	Untested

Test Case ID	5	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
15/01/2022	02:00	Gonçalo Oliveira	Pass	

Pré-condições / estado inicial				

Procedimento / Passos do user para esta função				
1. Enviar um Iterator<Encomenda> vazio.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Calcular o número médio de encomendas por dia Use Case: Módulo Transações Funcionalidade: getNumMedioEncomendasPorDia() 	* encomendas (Iterator<Encomenda>)	encomendas: iterador para encomendas.	0	Pass

2.3.2 getNumMedioProdutosPorTransacao

Test Case ID	1	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12/01/2022	23:34	Bruno Ferreira	Pass	
Pré-condições / estado inicial				
<div>1. Criar duas entidades que vão ser o sender e o receiver de uma transação.</div> <div>2. Criar dois produtos</div> <div>3. Criar duas transações do tipo envio.</div> <div>4. Adicionar um produto a cada linha de transação</div> <div>5. Adicionar uma linha de transação a cada transação</div> <div>6. Criar uma encomenda com o primeiro produto e uma quantidade associada de 5</div> <div>7. Criar uma encomenda com o segundo produto e uma quantidade associada de 1</div>				
Procedimento / Passos do user para esta função				
<div>1. Adicionar as duas encomendas</div> <div>2. Enviar o iterador das encomendas</div>				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Calcular número médio de produtos por transação Use Case: Módulo Transações Funcionalidade: getNumMedioProdutosPorTransacao()	* encomendas (Iterator<Encomenda>)	encomendas: iterador para encomendas.	3	Pass

Test Case ID	2	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12/01/2022	23:34	Bruno Ferreira	Pass	

Pré-condições / estado inicial				
<ol style="list-style-type: none"> 1. Criar duas entidades que vão ser o sender e o receiver de uma transação. 2. Criar três produtos 3. Criar três transações do tipo envio. 4. Adicionar um produto a cada linha de transação 5. Adicionar um linha de transação a cada transação 6. Criar uma nova encomendas com quantidade de 5 produtor 7. Criar uma nova encomendas com quantidade de 1 produtor 8. Criar uma nova encomendas com quantidade de 1 produtor 				
Procedimento / Passos do user para esta função				
<ol style="list-style-type: none"> 1. Adicionar as um encomendas ao Gestão Encomendas (considerando que as duas encomendas foram adicionadas no passo anterior) 2. Enviar o iterador das encomendas. 				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Calcular número médio de produtos por transação Use Case: Módulo Transações Funcionalidade: getNumMedioProdutosPorTransacao()	* encomendas (Iterator<Encomenda>)	encomendas: iterador para encomendas.	2.33	Pass

Test Case ID	3	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12/01/2022	23:34	Bruno Ferreira	Pass	

Pré-condições / estado inicial				

Procedimento / Passos do user para esta função				
1. Enviar no lugar do iterator das encomendas o valor null.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Calcular número médio de produtos por transação Use Case: Módulo Transações Funcionalidade: getNumMedioProdutosPorTransacao()	* encomendas (Iterator<Encomenda>)	encomendas: null	Throw IllegalArgumentExcepti on	Pass

Test Case ID	4	Dependências	
--------------	---	--------------	--

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

Pré-condições / estado inicial				

Procedimento / Passos do user para esta função				
1. Enviar no lugar do Iterator<Encomenda> uma string				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Calcular número médio de produtos por transação Use Case: Módulo Transações Funcionalidade: getNumMedioProdutosPorTransacao()	* encomendas (Iterator<Encomenda>)	encomendas: "encomenda"	Erro de Sintaxe	Untested

Test Case ID	5	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
15/01/2022	02:00	Gonçalo Oliveira	Pass	

Pré-condições / estado inicial				

Procedimento / Passos do user para esta função				
1. Enviar um Iterator<Encomenda> vazio.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Calcular número médio de produtos por transação Use Case: Módulo Transações Funcionalidade: getNumMedioProdutosPorTransacao()	* encomendas (Iterator<Encomenda>)	encomendas: iterador para encomendas.	0	Pass

2.3.3 getValorMedioTransacoes

Test Case ID	1	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12/01/2022	23:45	Bruno Ferreira	Pass	
Pré-condições / estado inicial				
<div>1. Criar duas entidades que vão ser o sender e o receiver de uma transação.</div> <div>2. Criar dois produtos</div> <div>3. Criar duas transações do tipo envio.</div> <div>4. Adicionar um produto a cada linha de transação</div> <div>5. Adicionar uma linha de transação a cada transação</div> <div>6. Criar duas encomendas</div>				
Procedimento / Passos do user para esta função				
<div>1. Adicionar as duas encomendas</div> <div>2. Enviar o iterador das encomendas</div>				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Calcular valor médio de transações Use Case: Módulo Transações Funcionalidade: getValorMedioTransacoes()	* encomendas (Iterator<Encomenda>)	encomendas: iterador para encomendas.	15	Pass

Test Case ID	2	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12/01/2022	23:45	Bruno Ferreira	Pass	

Pré-condições / estado inicial				
<ol style="list-style-type: none"> 1. Criar duas entidades que vão ser o sender e o receiver de uma transação. 2. Criar dois produtos 3. Criar duas transações do tipo envio. 4. Adicionar um produto a cada linha de transação 5. Adicionar uma linha de transação a cada transação 6. Criar três encomendas 7. Adicionar apenas duas encomendas das que foram adicionadas anteriormente 				
Procedimento / Passos do user para esta função				
<ol style="list-style-type: none"> 1. Adicionar a encomenda que ainda não foi adicionada. 2. Enviar o iterador das encomendas 				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Calcular valor médio de transações Use Case: Módulo Transações Funcionalidade: getValorMedioTransacoes()	* encomendas (Iterator<Encomenda>)	encomendas: iterador para encomendas.	26.67	Pass

Test Case ID	3	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12/01/2022	23:45	Bruno Ferreira	Pass	

Pré-condições / estado inicial				

Procedimento / Passos do user para esta função				
1. Enviar no lugar do iterator das encomendas o valor null.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Calcular valor médio de transações Use Case: Módulo Transações Funcionalidade: getValorMedioTransacoes()	* encomendas (Iterator<Encomenda>)	encomendas: null	Throw RuntimeException	Pass

Test Case ID	4	Dependências	
--------------	---	--------------	--

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

Pré-condições / estado inicial				

Procedimento / Passos do user para esta função				
1. Enviar no lugar do Iterator<Encomenda> uma string				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Calcular valor médio de transações Use Case: Módulo Transações Funcionalidade: getValorMedioTransacoes()	* encomendas (Iterator<Encomenda>)	encomendas: "encomenda"	Erro de Sintaxe	Untested

2.3.4 getTotalCustosEnvio

Test Case ID	1	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
17/01/2022	17:20	Gonçalo Oliveira	Pass	

Pré-condições / estado inicial				
<ol style="list-style-type: none"> 1. Criar duas entidades que vão ser o sender e o receiver de uma transação. 2. Criar um produto com volume 5 e peso 2.5 3. Criar uma transação 4. Adicionar uma linha de transação com o produto criado e quantidade 1 5. Criar uma encomenda com a transação que foi criada anteriormente e definir a data de criação como 16-01-2022 6. Criar outro produto com volume 1.65 e peso 0.5 7. Criar uma transação 8. Adicionar uma linha de transação com o produto criado e quantidade 2 9. Criar uma encomenda com a transação que foi criada anteriormente e definir a data de criação como 17-01-2022 10. Criar uma transação 11. Adicionar uma linha de transação com o primeiro produto criado e quantidade 1 12. Adicionar uma linha de transação com o segundo produto criado e quantidade 1 13. Criar uma encomenda com a transação que foi criada anteriormente e definir a data de criação como 15-01-2022 				
Procedimento / Passos do user para esta função				
<ol style="list-style-type: none"> 1. Adicionar as três encomendas criadas ao sistema de gestão. 2. Enviar as encomendas e definir o período de tempo de 16-01-2022 a 17-01-2022. 				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Calcular o total de custos de envio num determinado período de tempo Use Case: Módulo Transações Funcionalidade: getTotalCustosEnvio()	* encomendas (Iterator<Encomenda>) * dataInicio (LocalDate) * dataFim (LocalDate)	encomendas: iterador para encomendas. dataInicio: 16-01-2022 dataFim: 17-01-2022	66.87	Pass

Test Case ID	2	Dependências	
---------------------	---	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
17/01/2022	17:20	Gonçalo Oliveira	Pass	

Pré-condições / estado inicial
<ol style="list-style-type: none"> 1. Criar duas entidades que vão ser o sender e o receiver de uma transação. 2. Criar um produto com volume 0.41 e peso 0.10 3. Criar um produto com volume 5 e peso 2.5 4. Criar uma transação 5. Adicionar uma linha de transação com o primeiro produto criado e quantidade 1 6. Adicionar uma linha de transação com o segundo produto criado e quantidade 1 7. Criar uma encomenda com a transação que foi criada anteriormente e definir a data de criação como 16-01-2022

Procedimento / Passos do user para esta função
<ol style="list-style-type: none"> 1. Adicionar as duas encomendas criadas ao sistema de gestão. 2. Enviar as encomendas e definir o período de tempo de 15-01-2022 a 16-01-2022.

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Calcular o total de custos de envio num determinado período de tempo Use Case: Módulo Transações Funcionalidade: getTotalCustosEnvio()	* encomendas (Iterator<Encomenda>) * dataInicio (LocalDate) * dataFim (LocalDate)	encomendas: iterador para encomendas. dataInicio: 15-01-2022 dataFim: 16-01-2022	97.19	Pass

Test Case ID	3	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
17/01/2022	17:20	Gonçalo Oliveira	Pass	

Pré-condições / estado inicial				

Procedimento / Passos do user para esta função				
1. Definir o período de tempo de 16-01-2022 a 20-01-2022 2. Enviar null para as encomendas e definir o período de tempo de 16-01-2022 a 20-01-2022.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Calcular o total de custos de envio num determinado período de tempo Use Case: Módulo Transações Funcionalidade: getTotalCustosEnvio()	* encomendas (Iterator<Encomenda> * dataInicio (LocalDate) * dataFim (LocalDate)	encomendas: null dataInicio: 16-01-2022 dataFim: 20-01-2022	Throw IllegalArgumentExce ption	Pass

Test Case ID	4	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
17/01/2022	17:20	Gonçalo Oliveira	Pass	

Pré-condições / estado inicial

Procedimento / Passos do user para esta função
1. Enviar o valor null para a data início 2. Enviar as encomendas e definir a data fim como 20-01-2022.

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Calcular o total de custos de envio num determinado período de tempo Use Case: Módulo Transações Funcionalidade: getTotalCustosEnvio()	* encomendas (Iterator<Encomenda>) * dataInicio (LocalDate) * dataFim (LocalDate)	encomendas: iterador para encomendas dataInicio: null dataFim: 20-01-2022	Throw IllegalArgumentException	Pass

Test Case ID	5	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
17/01/2022	17:20	Gonçalo Oliveira	Pass	

Pré-condições / estado inicial
--

Procedimento / Passos do user para esta função
<ol style="list-style-type: none"> 1. Enviar o valor null para a data fim 2. Enviar as encomendas e definir a data início como 16-01-2022.

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Calcular o total de custos de envio num determinado período de tempo Use Case: Módulo Transações Funcionalidade: getTotalCustosEnvio()	* encomendas (Iterator<Encomenda>) * dataInicio (LocalDate) * dataFim (LocalDate)	encomendas: iterador para encomendas dataInicio: 16-01-2022 dataFim: null	Throw IllegalArgumentException	Pass

Test Case ID	6	Dependências	
---------------------	---	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
17/01/2022	17:20	Gonçalo Oliveira	Pass	

Pré-condições / estado inicial
--

Procedimento / Passos do user para esta função
<ol style="list-style-type: none"> 1. Enviar as encomendas existentes no sistema de gestão 2. Definir o período de tempo de 17-01-2022 a 15-01-2022.

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Calcular o total de custos de envio num determinado período de tempo Use Case: Módulo Transações Funcionalidade: getTotalCustosEnvio()	* encomendas (Iterator<Encomenda>) * dataInicio (LocalDate) * dataFim (LocalDate)	encomendas: iterador para encomendas dataInicio: 17-01-2022 dataFim: 15-01-2022	Throw IllegalArgumentExce ption	Pass

Test Case ID	7	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
17/01/2022	17:20	Gonçalo Oliveira	Pass	

Pré-condições / estado inicial
--

Procedimento / Passos do user para esta função
1. Enviar null para todos os parâmetros

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Calcular o total de custos de envio num determinado período de tempo Use Case: Módulo Transações Funcionalidade: getTotalCustosEnvio()	* encomendas (Iterator<Encomenda>) * dataInicio (LocalDate) * dataFim (LocalDate)	encomendas: null dataInicio: null dataFim: null	Throw IllegalArgumentException	Pass

Test Case ID	8	Dependências	
--------------	---	--------------	--

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

Pré-condições / estado inicial

Procedimento / Passos do user para esta função
--

1. Enviar o valor null para todos os atributos

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Calcular o total de custos de envio num determinado período de tempo Use Case: Módulo Transações Funcionalidade: getTotalCustosEnvio()	* encomendas (Iterator<Encomenda>) * dataInicio (LocalDate) * dataFim (LocalDate)	encomendas: null dataInicio: null dataFim: null	Erro de Sintaxe	Untested

2.3.5 getValorMedioVendasComprasPorDistrito

Test Case ID	1	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
21/01/2022	03:25	Bruno Ferreira	Pass	

Pré-condições / estado inicial				
<ol style="list-style-type: none"> 1. Criar um entidade de Aveiro 2. Criar um entidade de Porto 3. Criar produto1 com preço unitário 1, volume 0.41 e peso 0.10. 4. Criar produto2 com preço unitário 25, volume 5 e peso 2.5. 5. Criar produto3 com preço unitário 50, volume 1.65 e peso 0.5. 6. Criar uma transação1 em que o sender é a entidade de Aveiro e o receiver é de Porto. 7. Adicionar à transação criada anteriormente uma linha de transação que possui o produto 1 com quantidade 5. 8. Criar uma transação2 em que o sender é a entidade de Porto e o receiver é de Aveiro. 9. Adicionar à transação criada anteriormente uma linha de transação que possui o produto 2 com quantidade 1. 10. Criar uma transação3 em que o sender é a entidade de Porto e o receiver é de Aveiro. 11. Adicionar à transação criada anteriormente uma linha de transação que possui o produto 3 com quantidade 1. 12. Criar uma encomenda com as transações criadas anteriormente. 				
Procedimento / Passos do user para esta função				
<ol style="list-style-type: none"> 1. Adicionar as encomendas criadas anteriormente ao sistema de gestão. 2. Enviar o iterador para as encomendas adicionadas 				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Calcular o valor médio de vendas e compras por distrito Use Case: Módulo Transações Funcionalidade: getTotalCustosEnvio()	* encomendas (Iterator<Encomenda>)	encomendas: iterador para encomendas.	$matrix[4][0] = 14.68$ $matrix[4][1] = 66.13$ $matrix[6][0] = 66.13$ $matrix[6][1] = 14.68$ true	Pass

Test Case ID	2	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
21/01/2022	03:25	Bruno Ferreira	Pass	

Pré-condições / estado inicial				
<ol style="list-style-type: none"> 1. Criar um entidade de Aveiro 2. Criar um entidade de Portalegre 3. Criar um entidade de Porto 4. Criar produto1 com preço unitário 1, volume 0.41 e peso 0.10. 5. Criar produto2 com preço unitário 25, volume 5 e peso 2.5. 6. Criar produto3 com preço unitário 50, volume 1.65 e peso 0.5. 7. Criar uma transação1 em que o sender é a entidade de Aveiro e o receiver é de Porto. 8. Adicionar à transação criada anteriormente uma linha de transação que possui o produto 1 com quantidade 5. 9. Criar uma transação2 em que o sender é a entidade de Porto e o receiver é de Aveiro. 10. Adicionar à transação criada anteriormente uma linha de transação que possui o produto 2 com quantidade 1. 11. Criar uma transação3 em que o sender é a entidade de Porto e o receiver é de Aveiro. 12. Adicionar à transação criada anteriormente uma linha de transação que possui o produto 3 com quantidade 1. 13. Criar uma transação4 em que o sender é a entidade de Portalegre e o receiver é de Porto. 14. Adicionar à transação criada anteriormente uma linha de transação que possui o produto 2 com quantidade 1. 15. Criar uma transação5 em que o sender é a entidade de Aveiro e o receiver é de Portalegre. 16. Adicionar à transação criada anteriormente uma linha de transação que possui o produto 3 com quantidade 2. 17. Criar uma encomenda com as transações criadas anteriormente. 				
Procedimento / Passos do user para esta função				
<ol style="list-style-type: none"> 1. Adicionar as encomendas criadas anteriormente ao sistema de gestão. 2. Enviar o iterator para as encomendas adicionadas 				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Calcular o valor médio de vendas e compras por distrito Use Case: Módulo Transações Funcionalidade: getTotalCustosEnvio()	* encomendas (Iterator<Encomenda>)	encomendas: iterator para encomendas.	$matrix [4][0] = 86.34$ $matrix [4][1] = 66.13$ $matrix [6][0] = 66.13$ $matrix [6][1] = 113.97$ $matrix [13][0] = 213.25$ $matrix [13][1] = 158.0$ true	Pass

Test Case ID	3	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
21/01/2022	03:25	Bruno Ferreira	Pass	

Pré-condições / estado inicial				
--				
Procedimento / Passos do user para esta função				
1. Enviar o valor null para o iterador				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Calcular o valor médio de vendas e compras por distrito Use Case: Módulo Transações Funcionalidade: getTotalCustosEnvio()	* encomendas (Iterator<Encomenda>)	encomendas: null	Throw IllegalArgumentException	Pass

Test Case ID	4	Dependências	
--------------	---	--------------	--

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

Pré-condições / estado inicial
<p>--</p>

Procedimento / Passos do user para esta função
--

1. Enviar no lugar do iterador uma string

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Calcular o valor médio de vendas e compras por distrito Use Case: Módulo Transações Funcionalidade: getTotalCustosEnvio()	* encomendas (Iterator<Encomenda>)	encomendas: "encomendas"	Erro de sintaxe	Untested

2.4 Classe BasicCustos

2.4.1 calculaCustoEnvio

Test Case ID	1	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
17/01/2022	17:30	Jorge Correia	Pass	

Pré-condições / estado inicial				
<ol style="list-style-type: none">1. Criar um entidade sender do Aveiro2. Criar um entidade receiver de Porto3. Criar um produtos com o volume de 1.65 e peso de 0.104. Criar duas transações do tipo envio.5. Adicionar um produto6. Adicionar a linha de transação a transação7. Criar uma encomenda8. Importar tabela de custos				
Procedimento / Passos do user para esta função				
<ol style="list-style-type: none">1. Enviar a encomenda				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Calcular custo de envio de uma determinada encomenda Use Case: Módulo de cálculo de custos Funcionalidade: calculaCustoEnvio()	*encomenda (Encomenda)	encomenda: BasicEncomenda	4.25	Pass

Test Case ID	2	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
17/01/2022	17:30	Jorge Correia	Pass	

Pré-condições / estado inicial				
<ol style="list-style-type: none"> 1. Criar um entidade sender de Aveiro 2. Criar um entidade receiver de Porto 3. Criar um produtos com o volume de 0.25 e peso de 0.50 4. Criar um produtos com o volume de 0.15 e peso de 0.10 5. Criar duas transações do tipo envio. 6. Adicionar um produto 7. Adicionar a linha de transação a transação 8. Criar uma encomenda 9. Importar tabela de custos 				
Procedimento / Passos do user para esta função				
1. Enviar a encomenda				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Calcular custo de envio de uma determinada encomenda Use Case: Módulo de cálculo de custos Funcionalidade: calculaCustoEnvio()	*encomenda (Encomenda)	encomenda: BasicEncomenda	20.72	Pass

Test Case ID	3	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
17/01/2022	17:30	Jorge Correia	Pass	

Pré-condições / estado inicial				

Procedimento / Passos do user para esta função				
1. Enviar no lugar da Encomenda o valor null				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Calcular custo de envio de uma determinada encomenda Use Case: Módulo de cálculo de custos Funcionalidade: calculaCustoEnvio()	*encomenda (Encomenda)	encomenda: null	Throw IllegalArgumentExce ption	Pass

Test Case ID	4	Dependências	
--------------	---	--------------	--

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

Pré-condições / estado inicial				

Procedimento / Passos do user para esta função				
1. Enviar no lugar da Encomenda uma string				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Calcular custo de envio de uma determinada encomenda Use Case: Módulo de cálculo de custos Funcionalidade: calculaCustoEnvio()	*encomenda (Encomenda)	encomenda: "encomenda"	Erro de Sintaxe	Untested

2.5 Classe Expedicoes

2.5.1 assignEncomendas

Test Case ID	1	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
21/01/2022	03:45	Bruno Ferreira	Pass	

Pré-condições / estado inicial				
<ol style="list-style-type: none"> 1. Criar um entidade de Aveiro 2. Criar um entidade de Portalegre 3. Criar um entidade de Porto 4. Criar produtos 5. Criar uma transação em que o sender é a entidade de Aveiro e o receiver é de Portalegre. 6. Criar uma encomenda com a transação criada anteriormente. 7. Criar uma transação em que o sender é a entidade de Portalegre e o receiver é a entidade de Aveiro. 8. Criar uma encomenda com a transação criada anteriormente. 9. Criar uma transação em que o sender é a entidade de Aveiro e o receiver é a entidade do Porto. 10. Criar uma encomenda com a transação criada anteriormente. 11. Criar uma transação em que o sender é a entidade do Porto e o receiver é a entidade de Aveiro. 12. Criar uma encomenda com a transação criada anteriormente. 13. Criar uma transação em que o sender é a entidade de Aveiro e o receiver é a entidade de Portalegre. 14. Criar uma encomenda com a transação criada anteriormente. 15. Adicionar todas as encomendas ao sistema de gestão. 				
Procedimento / Passos do user para esta função				
<ol style="list-style-type: none"> 1. Enviar as encomendas criadas. 2. Enviar uma localização para a exportação dos contentores. 				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Atribuir encomendas a contentores Use Case: Módulo de expedição Funcionalidade: assignEncomendas()	* encomendas (Iterator<Encomenda>) *path (String)	encomendas: iterador para encomendas path: " "	false	Pass

Test Case ID	2	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
21/01/2022	03:45	Bruno Ferreira	Pass	

Pré-condições / estado inicial				
<ol style="list-style-type: none"> 1. Criar um entidade de Aveiro 2. Criar um entidade de Portalegre 3. Criar um entidade de Porto 4. Criar produtos 5. Criar uma transação em que o sender é a entidade de Aveiro e o receiver é de Portalegre. 6. Criar uma encomenda com a transação criada anteriormente. 7. Criar uma transação em que o sender é a entidade de Portalegre e o receiver é a entidade de Aveiro. 8. Criar uma encomenda com a transação criada anteriormente. 9. Criar uma transação em que o sender é a entidade de Aveiro e o receiver é a entidade do Porto. 10. Criar uma encomenda com a transação criada anteriormente. 11. Criar uma transação em que o sender é a entidade do Porto e o receiver é a entidade de Aveiro. 12. Criar uma encomenda com a transação criada anteriormente. 13. Criar uma transação em que o sender é a entidade de Aveiro e o receiver é a entidade de Portalegre. 14. Criar uma encomenda com a transação criada anteriormente. 15. Adicionar todas as encomendas ao sistema de gestão. 				
Procedimento / Passos do user para esta função				
<ol style="list-style-type: none"> 1. Enviar as encomendas criadas. 2. Enviar uma localização para a exportação dos contentores. 				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Atribuir encomendas a contentores Use Case: Módulo de expedição Funcionalidade: assignEncomendas()	* encomendas (Iterator<Encomenda>) * path (String)	encomendas: iterador para encomendas path: "./libs/Contentores.json"	true	Pass

Test Case ID	3	Dependências	
--------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
21/01/2022	03:45	Bruno Ferreira	Pass	

Pré-condições / estado inicial				
<ol style="list-style-type: none"> 1. Criar um entidade de Aveiro 2. Criar um entidade de Portalegre 3. Criar um entidade de Porto 4. Criar produtos 5. Criar uma transação em que o sender é a entidade de Aveiro e o receiver é de Portalegre. 6. Criar uma encomenda com a transação criada anteriormente. 7. Criar uma transação em que o sender é a entidade de Portalegre e o receiver é a entidade de Aveiro. 8. Criar uma encomenda com a transação criada anteriormente. 9. Criar uma transação em que o sender é a entidade de Aveiro e o receiver é a entidade do Porto. 10. Criar uma encomenda com a transação criada anteriormente. 11. Criar uma transação em que o sender é a entidade do Porto e o receiver é a entidade de Aveiro. 12. Criar uma encomenda com a transação criada anteriormente. 13. Criar uma transação em que o sender é a entidade de Aveiro e o receiver é a entidade de Portalegre. 14. Criar uma encomenda com a transação criada anteriormente. 15. Adicionar todas as encomendas ao sistema de gestão. 				
Procedimento / Passos do user para esta função				
<ol style="list-style-type: none"> 1. Enviar as encomendas vazias. 2. Enviar uma localização para a exportação dos contentores. 				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Atribuir encomendas a contentores Use Case: Módulo de expedição Funcionalidade: assignEncomendas()	* encomendas (Iterator<Encomenda>) * path (String)	encomendas: iterador das encomendas vazio path: "./libs/Contentores.json"	false	Pass

TestCase ID	4	Dependências	
-------------	---	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
21/01/2022	03:45	Bruno Ferreira	Pass	

Pré-condições / estado inicial				

Procedimento / Passos do user para esta função				
1. Enviar no lugar das encomendas o valor null 2. Enviar uma localização para a exportação dos contentores.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Atribuir encomendas a contentores Use Case: Módulo de expedição Funcionalidade: assignEncomendas()	* encomendas (Iterator<Encomenda>) *path (String)	encomendas: null path: "../libs/Contentores.json"	Throw IllegalArgumentExce ption	Pass

Test Case ID	5	Dependências	
--------------	---	--------------	--

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

Pré-condições / estado inicial

Procedimento / Passos do user para esta função
<ol style="list-style-type: none"> 1. Enviar no lugar das encomendas uma string 2. Enviar uma localização para a exportação dos contentores.

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Requisito: Atribuir encomendas a contentores Use Case: Módulo de expedição Funcionalidade: assignEncomendas()	* encomendas (Iterator<Encomenda>) *path (String)	encomendas: "encomendas" path: "./libs/Contentores.json"	Erro de sintaxe	Untested