# 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 especais (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

- 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

1. Adicionar uma linha à transação criada no ponto 2 que deverá possuir o produto e a respetiva quantidade desejada.

Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/
Case/		Inputs	Esperados	Untested
Funcionalidade				
Requisito:	*linhaTransacao	linhaTransacao:	true	Pass
Use Case:	(TransactionLine)	BasicLinhaTrans		
Módulo		acao		
Transações				
Funcionalidade:				
addTransactionLi				
ne()				

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	

- 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

1. Adicionar na transação criada uma linha de transação que já se encontra inserida, por exemplo a criada no caso anterior.

Requisito/Use Case/	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Funcionalidade				
Requisito: Use Case: Módulo Transações Funcionalidade: addTransactionLi ne()	*linhaTransacao (TransactionLine)	linhaTransacao: BasicLinhaTrans acao 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

- 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

- 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: addTransactionLi ne()	*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 / est	ado inicial					
Procedimento / Pa	ssos do <i>user</i> para esta função					
1. Adicionar a	uma transação null					
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/		
Case/	·	Inputs	Esperados	Untested		
Funcionalidade			•			
Requisito:	*linhaTransacao	linhaTransacao:	Throw	Pass		
Use Case:	(TransactionLine)	null	RunTimeException			
Módulo	·					
Transações						
Funcionalidade:						
addTransactionLi						
ne()						
Test Case ID	5	Dependências				
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo		
				da Falha		
Pré-condições / est	ado inicial					
Procedimento / Pa	ssos do <i>user</i> para esta função					
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/		
Case/		Inputs	Esperados	Untested		
Funcionalidade						
Requisito:	*linhaTransacao	linhaTransacao:	Erro de Sintaxe	Untested		
Use Case:	(TransactionLine)	"linhaTransacao"				
Módulo	,					
Transações						
Funcionalidade:						
addTransactionLi						
ne()						

## 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

- 1. Criar duas entidades que vão ser o sender e o receiver de uma transação.
- 2. Criar 1 produto
- 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 a encomenda que foi criada anteriormente que ainda não foi adicionada ao sistema

Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/
Case/		Inputs	Esperados	Untested
Funcionalidade				
Requisito:	*encomenda	encomenda:	true	Pass
Adicionar	(Encomenda)	BasicEncomen		
encomenda		da		
Use Case:				
Módulo				
Transações				
Funcionalidade:				
addEncomenda(				
)				

	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	

- 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/	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Funcionalidade				
Requisito:	*encomenda	encomenda:	false	Pass
Adicionar	(Encomenda)	BasicEncomen		
encomenda	,	da		
Use Case:				
Módulo				
Transações				
Funcionalidade:				
addEncomenda(				
)				

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

- 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 IllegalArgumentExcep tion	Pass

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

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

- 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. Enviar uma String, ou seja, enviar como parâmetro um valor inválido.

21 2111101 011110	string, ou seju, criviar como	oarannoero ann van	71 1111 4111 411 411	
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/
Case/		Inputs	Esperados	Untested
Funcionalidade				
Requisito:	*encomenda	encomenda:	Erro de Sintaxe	Untested
Adicionar	(Encomenda)	"encomenda"		
encomenda				
Use Case:				
Módulo				
Transações				
Funcionalidade:				
addEncomenda(				
)				

#### 2.2.2 cancelar Encomenda

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/	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Funcionalidade				
Requisito:	*encomenda	encomenda:	true	Pass
Cancelar	(Encomenda)	BasicEncomen		
encomenda		da		
Use Case:				
Módulo				
Transações				
Funcionalidade:				
cancelarEncome				
nda()				

	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	

- 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.

1. Caricciai a	encomenda que foi chada am	chonnente que al	naa nao ioi aaicionaaa.	
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/
Case/		Inputs	Esperados	Untested
Funcionalidade				
Requisito:	*encomenda	encomenda:	false	Pass
Cancelar	(Encomenda)	BasicEncomen		
encomenda		da		
Use Case:				
Módulo				
Transações				
Funcionalidade:				
cancelarEncome				
nda()				

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	

- 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

1. Caricelai ui	illa elicolliellua liula			
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/
Case/		Inputs	Esperados	Untested
Funcionalidade				
Requisito:	*encomenda	encomenda:	Throw	Pass
Cancelar	(Encomenda)	null	RunTimeException	
encomenda				
Use Case:				
Módulo				
Transações				
Funcionalidade:				
cancelarEncome				
nda( )				

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

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

- 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. Enviar como parâmetro uma string, ou seja, um parâmetro inválido.

1. Liividi coili	o parametro ama string, oa s	eja, ani parametro	iiivaliao.	_
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/
Case/		Inputs	Esperados	Untested
Funcionalidade				
Requisito:	*encomenda	encomenda:	Erro de Sintaxe	Untested
Cancelar	(Encomenda)	"encomenda5"		
encomenda				
Use Case:				
Módulo				
Transações				
Funcionalidade:				
cancelarEncome				
nda( )				

#### 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.

21 2111101 0 101	ara erroerrierraa erraaa qare ja i	0. 00.0.0.0.00.0		
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/
Case/		Inputs	Esperados	Untested
Funcionalidade				
Requisito:	*idencomenda	idencomenda:	Encomenda criada	Pass
Use Case:	(String)	"20220112_01"		
Funcionalidade:		_		
findEncomenda(				
)				

|--|

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	Inputs	Valores do	Resultados	Pass/Fail/
Case/		Inputs	Esperados	Untested
Funcionalidade				
Requisito:	*idencomenda	idencomenda:	null	Pass
Use Case:	(String)	"20220112_100"		
Funcionalidade:				
findEncomenda(				
)				

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	

- 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
				uu i uiiiu

## 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	

- 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 um id de encomenda null.

2. 2	a ac cheomenaa nam			
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/Un
Case/		Inputs	Esperados	tested
Funcionalidade				
Requisito:	*idencomenda	idencomenda:	Throw	Pass
Use Case:	(String)	null	IlegalArgumentExce	
Funcionalidade:			ption	
findEncomenda(				
)				

## 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

- 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/	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Funcionalidade				
Requisito: Calcular o número médio de encomendas por dia Use Case: Módulo Transações Funcionalidade: getNumMedioEn comendasPorDia ()	* encomendas (Iterator <encomenda>)</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	

- 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

- 1. Adicionar uma encomenda
- 2. Enviar o iterador das encomendas

Requisito/Use Case/	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Funcionalidade			1560.000	o inconcu
Requisito: Calcular o número médio de encomendas por dia Use Case: Módulo Transações Funcionalidade: getNumMedioEn comendasPorDia	* encomendas (Iterator <encomenda>)</encomenda>	encomendas: iterador para encomendas.	1	Pass

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

Pré-condições / est	Pré-condições / estado inicial				
Procedimento / Pag	ssos do <i>user</i> para esta função				
	ugar do iterator das encomen				
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: getNumMedioEn comendasPorDia ()	* encomendas (Iterator <encomenda>)</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 / est	ado inicial				
Procedimento / Passos do <i>user</i> para esta função					
1. Enviar no lu	1. Enviar no lugar do Iterator <encomenda> uma string</encomenda>				
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/	
Case/		Inputs	Esperados	Untested	
Funcionalidade					
Requisito:	* encomendas	encomendas:	Erro de Sintaxe	Untested	
Calcular o	(Iterator <encomenda>)</encomenda>	"encomenda"			
número médio					
de encomendas					
por dia					
Use Case:					
Módulo					
Transações					
Funcionalidade:					
getNumMedioEn					
comendasPorDia					

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

## 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

- 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 uma encomenda com o primeiro produto e uma quantidade associada de 5
- 7. Criar uma encomenda com o segundo produto e uma quantidade associada de 1

- 1. Adicionar as duas encomendas
- 2. Enviar o iterador das encomendas

Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/
Case/		Inputs	Esperados	Untested
Funcionalidade				
Requisito:	* encomendas	encomendas:	3	Pass
Calcular número	(Iterator <encomenda>)</encomenda>	iterador para		
médio de		encomendas.		
produtos por				
transação				
Use Case:				
Módulo				
Transações				
Funcionalidade:				
getNumMedioPr				
odutosPorTransa				
cao()				

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	

- 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

- 1. Adicionar as um encomendas ao Gestão Encomendas (considerando que as duas encomendas foram adicionadas no passo anterior)
- 2. Enviar o iterador das encomendas.

nendas: 2.33 or para nendas.	Untested Pass
or para	Pass
iciidus.	

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	

Drá condições / est	ada inicial			
Pré-condições / est	ado iniciai			
Procedimento / Pas	ssos do <i>user</i> para esta função			
1. Enviar no lu	ugar do iterator das encomen	das o valor null.		
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/
Case/		Inputs	Esperados	Untested
Funcionalidade				
Requisito:	* encomendas	encomendas:	Throw	Pass
Calcular número	(Iterator <encomenda>)</encomenda>	null	IllegalArgumentExcepti	
médio de	,		on	
produtos por				
transação				
Use Case:				
Módulo				
Transações				
Funcionalidade:				
getNumMedioPr				
odutosPorTransa				
cao()				

|--|

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

Pré-condições / est	ado inicial			
Procedimento / Pas	ssos do <i>user</i> para esta função			
1. Enviar no lu	ugar do Iterator <encomenda></encomenda>	uma string		
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/
Case/		Inputs	Esperados	Untested
Funcionalidade				
Requisito:	* encomendas	encomendas:	Erro de Sintaxe	Untested
Calcular número	(Iterator <encomenda>)</encomenda>	"encomenda"		
médio de				
produtos por				
transação				
Use Case:				
Módulo				
Transações				
Funcionalidade:				
getNumMedioPr				
odutosPorTransa				
cao()				

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 / est	ado inicial			
Procedimento / Pas	ssos do <i>user</i> para esta função			
1. Enviar um I	terator <encomenda> vazio.</encomenda>			
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/
Case/		Inputs	Esperados	Untested
Funcionalidade				
Requisito:	* encomendas	encomendas:	0	Pass
Calcular número	(Iterator <encomenda>)</encomenda>	iterador para		
médio de		encomendas.		
produtos por				
transação				
Use Case:				
Módulo				
Transações				
Funcionalidade:				
getNumMedioPr				
odutosPorTransa				
cao()				

## 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

- 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

- 1. Adicionar as duas encomendas
- 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: getValorMedioTr ansacoes()	* encomendas (Iterator <encomenda>)</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	

- 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

- 1. Adicionar a encomenda que ainda não foi adicionada.
- 2. Enviar o iterador das encomendas

Requisito/Use Case/	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Funcionalidade				
Requisito:	* encomendas	encomendas:	26.67	Pass
Calcular valor	(Iterator <encomenda>)</encomenda>	iterador para		
médio de		encomendas.		
transações				
Use Case:				
Módulo				
Transações				
Funcionalidade:				
getValorMedioTr				
ansacoes()				

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

Pré-condições / estado inicial					
	<del></del>				
Procedimento / Pas	ssos do <i>user</i> para esta função				
1. Enviar no lu	ugar do iterator das encomen	das o valor null.			
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/	
Case/		Inputs	Esperados	Untested	
Funcionalidade					
Requisito:	* encomendas	encomendas:	Throw	Pass	
Calcular valor	(Iterator <encomenda>)</encomenda>	null	RuntimeException		
médio de					
transações					
Use Case:					
Módulo					
Transações					
Funcionalidade:					
getValorMedioTr					

Test Case ID	4	Dependências	

ansacoes()

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

Pré-condições / estado inicial				
Procedimento / Pas	ssos do <i>user</i> para esta função	)		
1. Enviar no lu	ıgar do Iterator <encomenda></encomenda>	uma string		
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/
Case/		Inputs	Esperados	Untested
Funcionalidade				
Requisito:	* encomendas	encomendas:	Erro de Sintaxe	Untested
Calcular valor	(Iterator <encomenda>)</encomenda>	"encomenda"		
médio de				
transações				
Use Case:				
Módulo				
Transações				
Funcionalidade:				
getValorMedioTr				
ansacoes()				

## 2.3.4 getTotalCustosEnvio

Test Case ID	1	Dependências	
	_	_ 0 0 0 11 0 10 10 10 10 10 10 10 10 10 1	

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

#### Pré-condições / estado inicial

- 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

- 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	Inputs	Valores do	Resultados	Pass/Fail/
Case/		Inputs	Esperados	Untested
Funcionalidade				
Requisito:	* encomendas	encomendas:	66.87	Pass
Calcular o total	(Iterator <encomenda>)</encomenda>	iterador para		
de custos de	* datalnicio (LocalDate)	encomendas.		
envio num	* dataFim (LocalDate)			
determinado		dataInicio:		
período de		16-01-2022		
tempo				
Use Case:		dataFim:		
Módulo		17-01-2022		
Transações		17-01-2022		
Funcionalidade:				
getTotalCustosE				
nvio( )				

Test Case ID	2	Dependências	
icst case ib	l <del>-</del>	Dependencia	

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

- 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

- 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: getTotalCustosE nvio()	* encomendas (Iterator <encomenda>) * dataInicio (LocalDate) * dataFim (LocalDate)</encomenda>	encomendas: iterador para encomendas.  datalnicio: 15-01-2022  dataFim: 16-01-2022	97.19	Pass

Test Case ID	3	Dependências	
iest case ib	3	Dependencias	

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

- 1. Definir o período de tempo de 16-01-2022 a 20-01-2022

Z. Enviar nuli	para as encomendas e definir	o periodo de tem	po de 16-01-2022 a 20-0	1-2022.
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/
Case/		Inputs	Esperados	Untested
Funcionalidade				
Requisito: Calcular o total de custos de envio num determinado período de tempo Use Case: Módulo Transações Funcionalidade: getTotalCustosE nvio()	* encomendas (Iterator <encomenda>) * dataInicio (LocalDate) * dataFim (LocalDate)</encomenda>	encomendas: null datalnicio: 16-01-2022 dataFim: 20-01-2022	Throw IlegalArgumentExce 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	

-----

- 1. Enviar o valor null para a data início
- 2. Enviar as encomendas e definir a data fim como 20-01-2022.

				- /- "/
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/
Case/		Inputs	Esperados	Untested
Funcionalidade				
Requisito:	* encomendas	encomendas:	Throw	Pass
Calcular o total	(Iterator <encomenda>)</encomenda>	iterador para	IlegalArgumentExce	
de custos de	* datalnicio (LocalDate)	encomendas	ption	
envio num	* dataFim (LocalDate)			
determinado		datalnicio: null		
período de				
tempo		dataFim:		
Use Case:		20-01-2022		
Módulo				
Transações				
Funcionalidade:				
getTotalCustosE				
nvio( )				

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 / est				
Procedimento / Pas	ssos do <i>user</i> para esta função			
<ol> <li>Enviar o va</li> </ol>	lor null para a data fim			
2. Enviar as e	ncomendas e definir a data in	ício como 16-01-20	022.	
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/
Case/			_	_
Case/		Inputs	Esperados	Untested
Funcionalidade		Inputs	Esperados	Untested
•	* encomendas	encomendas:	Throw	Untested Pass

Requisito/ Ose	Imputs	valores do	Resultados	Pass/raii/
Case/		Inputs	Esperados	Untested
Funcionalidade				
Requisito:	* encomendas	encomendas:	Throw	Pass
Calcular o total	(Iterator <encomenda>)</encomenda>	iterador para	IlegalArgumentExce	
de custos de	* datalnicio (LocalDate)	encomendas	ption	
envio num	* dataFim (LocalDate)			
determinado		dataInicio:		
período de		16-01-2022		
tempo		10 01 1011		
Use Case:		dataFim:		
Módulo		null		
Transações		l IIuli		
Funcionalidade:				
getTotalCustosE				
nvio( )				

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	

- 1. Enviar as encomendas existentes no sistema de gestão
- 2. Definir o período de tempo de 17-01-2022 a 15-01-2022.

2. Delinii o periodo de tempo de 17 of 2022 à 13 of 2022.				
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/
Case/		Inputs	Esperados	Untested
Funcionalidade				
Requisito:	* encomendas	encomendas:	Throw	Pass
Calcular o total	(Iterator <encomenda>)</encomenda>	iterador para	IlegalArgumentExce	
de custos de	* datalnicio (LocalDate)	encomendas	ption	
envio num	* dataFim (LocalDate)			
determinado	, , ,	dataInicio:		
período de		17-01-2022		
tempo		1, 01 2022		
Use Case:		dataFim:		
Módulo		15-01-2022		
Transações		15-01-2022		
Funcionalidade:				
getTotalCustosE				
nvio( )				

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 / est	ado inicial			
Procedimento / Pas	ssos do <i>user</i> para esta função			
<ol> <li>Enviar null</li> </ol>	para todos os parâmetros	_		
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/
Case/		Inputs	Esperados	Untested
Funcionalidade				
Requisito:	* encomendas	encomendas:	Throw	Pass
Calcular o total	(Iterator <encomenda>)</encomenda>	null	IlegalArgumentExce	
de custos de	* datalnicio (LocalDate)		ption	
envio num	* dataFim (LocalDate)	dataInicio: null		
determinado				
período de		dataFim:		
tempo Use Case:		null		
Módulo				
Transações				
Funcionalidade:				
getTotalCustosE				
nvio( )				

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

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

Pré-condições / est	ado inicial			
Procedimento / Pas	ssos do <i>user</i> para esta função			
1. Enviar o va	lor null para todos os atributo	)S		
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/
Case/		Inputs	Esperados	Untested
Funcionalidade				
Requisito:	* encomendas	encomendas:	Erro de Sintaxe	Untested
Calcular o total	(Iterator <encomenda>)</encomenda>	null		
de custos de	* datalnicio (LocalDate)			
envio num	* dataFim (LocalDate)	dataInicio:		
determinado		null		
período de				
tempo		dataFim:		
Use Case:		null		
Módulo				
Transações				
Funcionalidade:				
getTotalCustosE				
nvio( )				

## 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

- 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 transação 3 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.

- 1. Adicionar as encomendas criadas anteriormente ao sistema de gestão.
- 2. Enviar o iterador para as encomendas adicionadas

Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/
Case/		Inputs	Esperados	Untested
Funcionalidade				
Requisito:	* encomendas	encomendas:	matrix [4][0] = 14.68	Pass
Calcular o valor	(Iterator <encomenda>)</encomenda>	iterador para	matrix [4][1] = 66.13	
médio de vendas	,	encomendas.		
e compras por			matrix [6][0] = 66.13	
distrito			matrix [6][1] = 14.68	
Use Case:				
Módulo			true	
Transações				
Funcionalidade:				
getTotalCustosE				
nvio( )				

|--|

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

- 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 transação 2 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 transação 3 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.

- 1. Adicionar as encomendas criadas anteriormente ao sistema de gestão.
  - Enviar o iterador para as encomendas adicionadas

Requisito/Use	Inputs	Valores do	Resultados Esperados	Pass/Fail/
Case/		Inputs		Untested
Funcionalidade				
Requisito:	* encomendas	encomendas:	matrix [4][0] = 86.34	Pass
Calcular o valor	(Iterator <encomenda>)</encomenda>	iterador para	matrix [4][1] = 66.13	
médio de vendas		encomendas.		
e compras por			matrix [6][0] = 66.13	
distrito			matrix [6][1] = 113.97	
Use Case:				
Módulo			matrix [13][0] = 213.25	
Transações			matrix [13][1] = 158.0	
Funcionalidade:				
getTotalCustosE			true	
nvio( )				

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	

21/01/2022	03:25	Bruno Ferreira	Pass			
Pré-condições / estado inicial						
Procedimento / Pas	ssos do <i>user</i> para esta funçã	ăo				
1. Enviar o va	lor null para o iterador					
Requisito/Use	Inputs	Valores do	Resultados Esperados	Pass/Fail/		
Case/		Inputs		Untested		
Funcionalidade						
Requisito:	* encomendas	encomendas:	Throw	Pass		
Calcular o valor	(Iterator <encomenda>)</encomenda>	null	IlegalArgumentExcepti			
médio de vendas			on			
e compras por						
distrito						
Use Case:						
Módulo						
Transações						
Funcionalidade:						
getTotalCustosE nvio( )						

Test Case ID	4	Dependências						
		-						
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo				
				da Falha				
Pré-condições / est	ado inicial							
	ssos do <i>user</i> para esta funçã	ío						
1. Enviar no lu	ugar do iterador uma string			Enviar no lugar do iterador uma string				
Requisito/Use	Inputs	Valores do	Resultados Esperados	Pass/Fail/				
Case/	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested				
Case/ Funcionalidade	·		Resultados Esperados					
Case/ Funcionalidade Requisito:	* encomendas		Resultados Esperados  Erro de sintaxe					
Case/ Funcionalidade Requisito: Calcular o valor	·	Inputs		Untested				
Case/ Funcionalidade Requisito: Calcular o valor médio de vendas	* encomendas	Inputs encomendas:		Untested				
Case/ Funcionalidade Requisito: Calcular o valor médio de vendas e compras por	* encomendas	Inputs encomendas:		Untested				
Case/ Funcionalidade  Requisito: Calcular o valor médio de vendas e compras por distrito	* encomendas	Inputs encomendas:		Untested				
Case/ Funcionalidade  Requisito: Calcular o valor médio de vendas e compras por distrito Use Case:	* encomendas	Inputs encomendas:		Untested				
Case/ Funcionalidade  Requisito: Calcular o valor médio de vendas e compras por distrito Use Case: Módulo	* encomendas	Inputs encomendas:		Untested				
Case/ Funcionalidade  Requisito: Calcular o valor médio de vendas e compras por distrito Use Case: Módulo Transações	* encomendas	Inputs encomendas:		Untested				
Case/ Funcionalidade  Requisito: Calcular o valor médio de vendas e compras por distrito Use Case: Módulo	* encomendas	Inputs encomendas:		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

- 1. Criar um entidade sender do Aveiro
- 2. Criar um entidade receiver de Porto
- 3. Criar um produtos com o volume de 1.65 e peso de 0.10
- 4. Criar duas transações do tipo envio.
- 5. Adicionar um produto
- 6. Adicionar a linha de transação a transação
- 7. Criar uma encomenda
- 8. Importar tabela de custos

# Procedimento / Passos do user para esta função

1. Enviar a encomenda

Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/
Case/		Inputs	Esperados	Untested
Funcionalidade				
Requisito:	*encomenda	encomenda:	4.25	Pass
Calcular custo de	(Encomenda)	BasicEncomen		
envio de uma		da		
determinada				
encomenda				
Use Case:				
Módulo de				
cálculo de custos				
Funcionalidade:				
calculaCustoEnvi				
o()				

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	

- 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

1. Liiviai a Cii	1. Liivia a encomenda			
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/
Case/		Inputs	Esperados	Untested
Funcionalidade				
Requisito:	*encomenda	encomenda:	20.72	Pass
Calcular custo de	(Encomenda)	BasicEncomen		
envio de uma		da		
determinada				
encomenda				
Use Case:				
Módulo de				
cálculo de custos				
Funcionalidade:				
calculaCustoEnvi				
o()				

Test Case ID 3 Dependê	encias
------------------------	--------

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 / Pas	ssos do <i>user</i> para esta função				
1. Enviar no lu	ıgar da Encomenda o valor nı	اا			
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/	
Case/		Inputs	Esperados	Untested	
Funcionalidade					
Requisito:	*encomenda	encomenda:	Throw	Pass	
Calcular custo de	(Encomenda)	null	IlegalArgumentExce		
envio de uma			ption		
determinada					
encomenda					
Use Case:					
Módulo de					
cálculo de custos					
Funcionalidade:					
calculaCustoEnvi					
o()					

Test Case ID	4	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo
				da Falha

Pré-condições / estado inicial				
Procedimento / Pas	ssos do <i>user</i> para esta função			
1. Enviar no lu	ıgar da Encomenda uma strin	g		_
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/
Case/		Inputs	Esperados	Untested
Funcionalidade				
Requisito:	*encomenda	encomenda:	Erro de Sintaxe	Untested
Calcular custo de	(Encomenda)	"encomenda"		
envio de uma				
determinada				
encomenda				
Use Case:				
Módulo de				
cálculo de custos				
Funcionalidade:				
calculaCustoEnvi				
0()				

# 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

- 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.

- 1. Enviar as encomendas criadas.
- 2. Enviar uma localização para a exportação dos contentores.

Requisito/Use Case/	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/ Untested
Funcionalidade				
Requisito:    Atribuir    encomendas a    contentores    Use Case:    Módulo de    expedição    Funcionalidade:    assignEncomend    as()	* encomendas (Iterator <encomenda>) *path (String)</encomenda>	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	

- 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.

- 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:    assignEncomend    as()	* encomendas (Iterator <encomenda>) *path (String)</encomenda>	encomendas: iterador para encomendas path: "./libs/Content ores.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	

- 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.
- 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.

- 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:    assignEncomend    as()	* encomendas (Iterator <encomenda>) *path (String)</encomenda>	encomendas: iterador das encomendas vazio path: "./libs/Content ores.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 / est	ado inicial			
Procedimento / Pas	ssos do <i>user</i> para esta função			
	ugar das encomendas o valor localização para a exportação			
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/
Cocol	•		_	
Case/		Inputs	Esperados	Untested
Funcionalidade		Inputs	Esperados	Untested
•	* encomendas (Iterator <encomenda>) *path (String)</encomenda>	encomendas: null path: "./libs/Content ores.json"	Throw IlegalArgumentExce ption	Pass

as()

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

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

Pré-condições / estado inicial					
Procedimento / Passos do <i>user</i> para esta função					
1. Enviar no lugar das encomendas uma string					
2. Enviar uma	2. Enviar uma localização para a exportação dos contentores.				
Requisito/Use	Inputs	Valores do	Resultados	Pass/Fail/	
Case/		Inputs	Esperados	Untested	
Funcionalidade					
Requisito:	* encomendas	encomendas:	Erro de sintaxe	Untested	
Atribuir	(Iterator <encomenda>)</encomenda>	"encomendas"			
encomendas a	*path	path:			
contentores	(String)	"./libs/Content			
Use Case:		ores.json"			
Módulo de					
expedição					
Funcionalidade:					
assignEncomend as()					