

<LEI_ESII2021_TP2_GRUP011>

TEST CASE OUTLINE

Version <1.1>

Realizado por:
8180378-Hugo Silva
8200306-Fábio Costa

Histórico de Versões

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	HUGO	<8/12/2021>	<Hugo,Fabio>	<8/12/2021>	Aprovação estrutura
1.1	HUGO	<8/1/2022>	<Hugo,Fabio>	<8/1/2022>	Versão Final

Tabela de Conteúdos

1. Introdução.....	5
1.1. Identificador do documento	5
1.2. Âmbito	5
1.3. References	5
1.4 Glossário	5
2. Detalhes <i>Class BasicTransaction</i>	6
BasicTransactionBVA01	6
BasicTransactionECP01	6
addTransactionLineBVA01.....	7
addTransactionLineECP01.....	8
addTransactionLineECP02.....	8
removeTransactionLineECP01	9
removeTransactionLineBVA01.....	9
removeTransactionLineBVA02.....	10
removeTransactionLineECP02	10
getTransactionLineBVA01.....	11
getTransactionLineECP01	12
getTransactionLineECP02	12
getTransactionCountECP01	13
getTotalValueECP01	13
3 Detalhes <i>Class BasicEntity</i>	14
BasicEntityBVA01	14
BasicEntityBVA02	14
BasicEntityECP01	15
addTokensBVA01.....	15
addTokensBVA02.....	16
addTokensBVA03.....	16
addTokensECP01	17
4.Detalhes <i>Class BasicOrganization</i>.....	18
getBlockBVA01	18
getBlockBVA02	18
getBlockBVA03	19
getBlockECP01	19
testRegisterTransactionsInLedgerECP01.....	20

testRegisterTransactionsInLedgerECP02.....	21
getBlockCountECP01	22
getBlockCountECP02	23
addTransactionBVA01	24
addTransactionECP01	24
addTransactionECP02	25
addTransactionECP03	25
addTransactionECP04	26
removeTransactionBVA01	27
removeTransactionBVA02	27
removeTransactionECP01	28
removeTransactionECP02	28
getTransactionBVA01	29
getTransactionECP01	29
getTransactionECP02	30
isValidEdgerBVA01	30
isValidEdgerBVA02	31
isValidEdgerBVA03	31
IsValidEdgerECP01	32
5.Detalhes Class BasicTransactionLine	33
BasicTransactionLineBVA01.....	33
BasicTransactionLineBVA02.....	33
BasicTransactionLineBVA03.....	34
BasicTransactionLineBVA04.....	34
BasicTransactionLineBVA05.....	35
BasicTransactionLineBVA06.....	35
testBasicTransactionLineECP01	36
6. Summary Tests	37

1. Introdução

1.1. Identificador do documento

TestCaseOutlineLedgerAPI

1.2. Âmbito

Este documento foi realizado para o trabalho 1 de Engenharia de Software II que tem como objetivo realização de atividades de garantia de qualidade de Software "SQA", em concreto a atividade realização de testes caixa preta (BlackBox Tests).

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

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

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

Os testes descritos foram planeados de acordo com os requisitos funcionais previamente identificados.

1.3. References

IEEE Std 829™-2008

1.4 Glossário

BVA – *Boundary Value Analysis*

ECP – *Equivalence Class Partitioning*

TC – *Teste Case*

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

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

TC-Grupo de condições utilizadas para testar software.

2. Detalhes Class BasicTransaction

BasicTransactionBVA01

Test Case ID	TC#BasicTransactionBVA01	Dependências	
--------------	--------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
31 dezembro	19h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado: duas entidades, uma para o envio da transação e outra para a receção.				
Procedimentos				
Invocar construtor passar dois parâmetros do tipo Entity.				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar Transação	sender, receiver	sender = null, receiver = null	IllegalArgumentException	PASS

BasicTransactionECP01

Test Case ID	TC#BasicTransactionECP01	Dependências	
--------------	--------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
31 dezembro	19h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado: duas entidades válidas				
Procedimentos				
Invocar construtor passando dois parâmetros do tipo Entity				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar Transação	sender, receiver	sender = receiverEntity, receiver = receiverEntity	InstanceOf BasicTransaction	PASS

addTransactionLineBVA01

Test Case ID	TC#addTransactionLineBVA01	Dependências	
---------------------	----------------------------	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
31 dezembro	19h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de instanciado dois objetos do tipo Entity.				
Tem de instanciado um objeto do tipo Transaction.				
Procedimentos				
Adicionar linha à transação, passando um parâmetro do tipo null.				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar uma linha à transação	transactionLine	null	Throw IllegalArgumentException	PASS

addTransactionLineECP01

Test Case ID	TC#addTransactionLineECP01	Dependências	
--------------	----------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
31 dezembro	19h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de instanciado dois objetos do tipo Entity. Tem de instanciado um objeto do tipo Transaction.				
Procedimentos				
Invocar método addTransactionLine passando uma linha válida para registo.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar uma linha à transação	transactionLine	transactionLine = basicTransactioLine*	true	PASS

addTransactionLineECP02

Test Case ID	TC#addTransactionLineECP02	Dependências	
--------------	----------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
31 dezembro	19h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de instanciado dois objetos do tipo Entity. Tem de instanciado um objeto do tipo Transaction. Tem de ser inserido pelo menos uma TransactionLine*.				
Procedimentos				
Para ser adicionada uma linha à transação, é preciso passar um parâmetro do tipo TransactionLine. Adicionar linha à transação. Volta a adicionar linha na transação e verificar o resultado.				
Requisito/Use Case/ Funcionalidade	Input	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar uma linha à transação	transactionLine	transactionLine = basicTransactioLine*	false	PASS

*BasicTransactionLine basicTransactionLine = new BasicTransactionLine("linha1", 1, 2)

removeTransactionLineECP01

Test Case ID	TC#removeTransactionLineECP01	Dependências	
--------------	-------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
31 dezembro	19h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de instanciado dois objetos do tipo Entity. Tem de instanciado um objeto do tipo Transaction.				
Procedimentos				
1-Adicionar linha à transação. 2-Remover transação adicionada no passo 1.				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Remover uma linha à transação	transactionLine	transactionLine = basicTransactionLine	true	PASS

removeTransactionLineBVA01

Test Case ID	TC#removeTransactionLineBVA01	Dependências	
--------------	-------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
31 dezembro	19h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de instanciado dois objetos do tipo Entity. Tem de instanciado um objeto do tipo Transaction.				
Procedimentos				
Para ser removida uma linha à transação, é preciso passar um parâmetro do tipo TransactionLine. Invocar método removeTransaction passando como parâmetro null e verificar o resultado.				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Remover uma linha à transação	transactionLine	null	IllegalArgumentException	PASS

removeTransactionLineBVA02

Test Case ID	TC#removeTransactionLineBVA02	Dependências	
--------------	-------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
31 dezembro	19h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de instanciado: dois objetos do tipo Entity, um do tipo Transaction e um do tipo TransactionLine . O array de linhas deve estar vazio.				
Procedimentos				
Invocar método removeTransaction passando como pãmetro uma linha válida e verificar o resultado				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untest ed
Remover uma linha à transação	transactionLine	transactionLine	false	PASS

removeTransactionLineECP02

Test Case ID	TC#removeTransactionLineECP02	Dependências	
--------------	-------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
31 dezembro	19h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de instanciado dois objetos do tipo Entity. Tem de instanciado um objeto do tipo Transaction criado.				
Procedimentos				
Invocar método removeTransaction passando como pãmetro uma linha que não está adicionada na transação				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untest ed
Remover uma linha à transação	transactionLine	transactionLine = basicTransactionLine	false	PASS

Objeto das tabelas: BasicTransactionLine basicTransactionLine = new BasicTransactionLine("linha1", 1, 2)

getTransactionLineBVA01

Test Case ID	TC#getTransactionLineBVA01	Dependências	
--------------	----------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
31 dezembro	19h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado um objeto do tipo Transaction.				
Procedimentos				
Invocar método getTransactionLine do objeto Transaction (passando null como parâmetro).				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter uma linha da transação	transactionLine	null	UnHashableException	PASS

getTransactionLineECP01

Test Case ID	TC#getTransactionLineECP01	Dependências	
--------------	----------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
31 dezembro	19h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de existir dois objetos do tipo Entity criados Terá de existir também um objeto do tipo Transaction criado. Terá de existir pelo menos uma TransactionLine já adicionada à Transaction.				
Procedimentos				
Invocar método getTransactionLine do objeto Transaction, sendo que o input deve corresponder a uma linha existente na transação.				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter uma linha da transação	transactionLine	transactionLine = basicTransactionLine	instanceof TransactionLine	PASS

getTransactionLineECP02

Test Case ID	TC#getTransactionLineECP02	Dependências	
--------------	----------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
31 dezembro	19h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de existir dois objetos do tipo Entity criados. Terá de existir também um objeto do tipo do Transaction.				
Procedimentos				
Para ser procurada uma linha da transação, é preciso passar um parâmetro do tipo TransactionLine. Neste caso o input não deve existir na transação.				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter uma linha da transação	transactionLine	transactionLine = basicTransactionLine	null	PASS

getTransactionCountECP01

Test Case ID	TC#testGetTransactionCountECP01	Dependências	
--------------	---------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	20:00h	Fábio	PASS	

Pré-condições / estado inicial				
Têm de ser instanciadas duas entidades válidas, uma Transaction válida e três Transactionline.				
Procedimentos				
Criar 3 linhas para adicionar a transação. Inserir as três linhas na transação. Verificar número de linhas da transação.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter valor de uma transação	Tipo TransactionLine	(tabela auxiliar)	3	PASS

getTotalValueECP01

Test Case ID	TC#getTransactionCountECP01	Dependências	
--------------	-----------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	20:00h	Fábio	PASS	

Pré-condições / estado inicial				
Têm de ser instanciadas pelo menos duas entidades válidas, uma Transaction válida e três Transactionline.				
Procedimentos				
Criar 3 linhas para adicionar a transação Inserir as três linhas na transação. Verificar valor total da transação (deverá ser igual à soma de todas as transações).				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter valor de uma transação	Tipo TransactionLine	(tabela auxiliar)	6.0	PASS

Tabela auxiliar

BasicTransactionLine basicTransactionLine1 = new BasicTransactionLine("linha1", 1, 2.0);
BasicTransactionLine basicTransactionLine2 = new BasicTransactionLine("linha2", 1, 2.0);
BasicTransactionLine basicTransactionLine3 = new BasicTransactionLine("linha3", 1, 2.0);

3 Detalhes Class BasicEntity

BasicEntityBVA01

Test Case ID	TC#testBasicEntityBVA01	Dependências	
--------------	-------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	18h	Fábio	PASS	

Pré-condições / estado inicial				
Procedimentos				
Invocar método BasicEntity passando um input a null e o outro vazio.				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar Entidade	name, district	name = "", district = null	IllegalArgumentException	PASS

BasicEntityBVA02

Test Case ID	TC#testBasicEntityBVA02	Dependências	
--------------	-------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	18h	Fábio	PASS	

Pré-condições / estado inicial				
Procedimentos				
Invocar método BasicEntity com os dois inputs a null.				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar Entidade	name, district	name = null, district = null	IllegalArgumentException	PASS

BasicEntityECP01

Test Case ID	TC#BasicEntityECP01	Dependências	
---------------------	---------------------	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	18h	Fábio	PASS	

Pré-condições / estado inicial				
Procedimentos				
Para instanciar uma entidade é preciso passar dois parâmetros , parâmetro name que é do tipo <i>String</i> , e o parâmetro district que é do tipo District.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar Entidade	name, district	name ="Entidade1", district = District.BEJA	Instanceof BasicEntity	PASS

addTokensBVA01

Test Case ID	TC#testAddTokensBVA01	Dependências	
---------------------	-----------------------	---------------------	--

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

Pré-condições / estado inicial				
Tem de instanciado um objeto Entity.				
Procedimentos				
Para ser adicionado um <i>Token</i> , é preciso passar um parâmetro do tipo <i>int</i> . Passos: Invocar método addTokens do objeto do tipo Entity passando -1 como <i>input</i> .				
Requisito/Us e Case/ Funcionalidade	Inputs	Valor do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar um Token	<i>int</i>	-1	Throw IllegalArgumentException	PASS

addTokensBVA02

Test Case ID	TC#testAddTokensBVA02	Dependências	
---------------------	-----------------------	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	19:25h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de instanciado um objeto Entity. Tem de ser obtido o número de tokens da entidade.				
Procedimentos				
1-Invocar método addTokens do objeto do tipo Entity passando 0 como input. 2-Verificar número de transações adicionadas.				
Requisito/Use Case/ Funcionalidade	Inputs	Valor do Input	Resultados Esperados	Pass/Fail/Untested
Adicionar um Token	<i>int</i>	0	nº <i>tokens</i> da Entity	PASS

addTokensBVA03

Test Case ID	TC#testAddTokensBVA03	Dependências	
---------------------	-----------------------	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	9:25h	Hugo	PASS	

Pré-condições / estado inicial				
Instanciar um objeto do tipo Entity.				
Procedimentos				
1-Invocar método addTokens do objeto do tipo Entity passando 2147483647 como <i>input</i> . 2-Verificar número de transações adicionadas.				
Requisito/Use Case/ Funcionalidade	Inputs	Valor do Input	Resultados Esperados	Pass/Fail/Untested
Adicionar um Token	<i>int</i>	2147483647	nº <i>tokens</i> da Entity	PASS

addTokensECP01

Test Case ID	TC#testAddTokensECP01	Dependências	
--------------	-----------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	19:30h	Hugo	PASS	

Pré-condições / estado inicial				
Instanciar um objeto do tipo Entity.				
Procedimentos				
1-Invocar método addTokens do objeto do tipo Entity passando 3 como <i>input</i> . 2-Verificar número de transações adicionadas.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar um Token	<i>int</i>	<i>int</i> = 3	return nº <i>tokens</i> da entidade	PASS

4.Detalhes Class BasicOrganization

getBlockBVA01

Test Case ID	TC#testGetBlockBVA01	Dependências	
--------------	----------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	19:45h	Hugo	FAIL	

Pré-condições / estado inicial				
Tem de ser instanciado uma Organization.				
Procedimentos				
1-Invocar método getBlock e passar parâmetro. 2-Verificar retorno recebido.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Input	Resultados Esperados	Pass/Fail/Untested
Obter Block	int	0	Instanceof Block	FAIL

getBlockBVA02

Test Case ID	TC#testGetBlockBVA02	Dependências	

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	19:45h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de ser criada uma Organization válida.				
Procedimentos				
1-Invocar método getBlock e passar parâmetro com o valor de getBlockCount. 2-Verificar retorno recebido.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Input	Resultados Esperados	Pass/Fail/Untested
Obter Block	int	getBlockCount()	IndexOutOfBoundsException	PASS

getBlockBVA03

Test Case ID	TC#testGetBlockBVA03	Dependências	
--------------	----------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	19:45h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado um objeto do tipo Organization.				
Procedimentos				
1-Invocar método getBlock e passar parâmetro -1 2-Verificar retorno recebido.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter Block	int	int = -1	IndexOutOfBoundsException	PASS

getBlockECP01

Test Case ID	TC#testGetBlockECP01	Dependências	
--------------	----------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12 dezembro	16:55h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado um objeto do tipo Organization. Tem de ser instanciada uma transação e duas entidades em que a entidade de envio tem 2 tokens para registo.				
Procedimentos				
1-Adicionar e registar uma transação. 2-Repetir passo 1. 3- Invocar método getBlock, passando 1 como parâmetro.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter Block	int	1	Instanceof Block	PASS

testRegisterTransactionsInLedgerECP01

Test Case ID	TC# testRegisterTransactionsInLedgerECP01	Dependências	
--------------	--	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	10:45h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado: um objeto do tipo Organization e uma transação (em que o sender tem tokens para registrar).				
Procedimentos				
Para registrar uma transação é preciso adicioná-la ao ledger. Depois de adicionada é invocado o método registerTransactionInLedger que a regista.				
Requisito/ Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Registrar transações no ledger	basicTransaction	(tabela auxiliar 1)	1	PASS

Tabela auxiliar 1

<code>sender.addTokens(1)</code>
<code>basicTransaction = new BasicTransaction(sender, receiver)</code>
<code>basicOrganization.addTransaction(basicTransaction)</code>
<code>basicOrganization.registerTransactionsInLedger()</code>

testRegisterTransactionsInLedgerECP02

Test Case ID	TC# testRegisterTransactionsInLedgerECP02	Dependências	
--------------	--	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	10:49h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado: um objeto do tipo Organization e uma transação (em que o sender não possui tokens para registrar transação).				
Procedimentos				
Para registrar uma transação é preciso adicioná-la ao ledger. Depois de adicionada é invocado o método registerTransactionInLedger.				
Requisito/ Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Registrar transações no <i>ledger</i>	basicTransaction	(tabela auxiliar 2)	0	PASS

Tabela auxiliar 2

basicTransaction = new BasicTransaction(sender, receiver)
basicOrganization.addTransaction(basicTransaction)
basicOrganization.registerTransactionsInLedger()

getBlockCountECP01

Test Case ID	TC# testGetBlockCountECP01	Dependências	
--------------	----------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	10:45h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado: um objeto do tipo Organization e uma transação (em que o sender tem tokens para registrar).				
Procedimentos				
Para registrar uma transação é preciso adicioná-la ao <i>ledger</i> . Depois de adicionada é invocado o metodo registerTransactionInLedger que deve criar um bloco novo com a transação registada.				
Requisito/ Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Registrar transações no <i>ledger</i>	basicTransaction	(tabela auxiliar 3)	2	PASS

Tabela auxiliar 3

<code>sender.addTokens(1)</code>
<code>basicTransaction = new BasicTransaction(sender, receiver)</code>
<code>basicOrganization.addTransaction(basicTransaction)</code>
<code>basicOrganization.registerTransactionsInLedger()</code>
<code>getBlockCount()</code>

getBlockCountECP02

Test Case ID	TC# testGetBlockCountECP02	Dependências	
--------------	----------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	10:55h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado: um objeto do tipo Organization e uma transação (em que o sender não tem tokens para registrar).				
Procedimentos				
Para registar uma transação é preciso adicioná-la ao <i>ledger</i> . Depois de adicionada é invocado o método registerTransactionInLedger que não deve criar nenhum bloco novo pois a transação não tem condições para ser registada.				
Requisito/ Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Registar transações no <i>ledger</i>	basicTransaction	(tabela auxiliar 4)	1	PASS

Tabela auxiliar 4

basicTransaction = new BasicTransaction(sender, receiver)
basicOrganization.addTransaction(basicTransaction)
basicOrganization.registerTransactionsInLedger()
getBlockCount()

addTransactionBVA01

Test Case ID	TC#testAddTransactionBVA01	Dependências	
--------------	----------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	19:35h	Fábio	PASS	

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

addTransactionECP01

Test Case ID	TC#testAddTransactionECP01	Dependências	
--------------	----------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
6 dezembro	19:35h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado: um objeto do tipo Organization, um objeto do tipo Entity e um objeto do tipo BasicTransaction.				
Procedimentos				
Para ser adicionado uma Transaction, é preciso passar um parâmetro do tipo Transaction.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar transação	transaction	transaction = basicTransaction	true	PASS

Objeto “basicTransaction”: basicTransaction = new BasicTransaction(sender, receiver);

addTransactionECP02

Test Case ID	TC#testAddTransactionECP02	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	19:35h	Fábio	PASS	
Pré-condições / estado inicial				
Tem de ser instanciado: um objeto do tipo Organization, um objeto do tipo Entity e um objeto do tipo BasicTransaction. Já deverá existir pelo menos uma Transaction adicionada ao <i>ledger</i> .				
Procedimentos				
Invocar método addTransaction do objeto do tipo Organization, passando como parâmetro uma transação já existente no <i>ledger</i> .				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar transação	transaction	transaction = basicTransaction	false	PASS

addTransactionECP03

Test Case ID	TC#testAddTransactionECP03	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
12 dezembro	03:30h	Hugo	PASS	
Pré-condições / estado inicial				
Tem de ser instanciado: um objeto do tipo Organization, um objeto do tipo Entity e um objeto do tipo BasicTransaction.				
Procedimentos				
1-Adicionar tokens à entidade de envio da transação 2-Adicionar uma transação válida para registo ao <i>ledger</i> . 3-Registar a transação no <i>ledger</i> 4-Adicionar novamente a transação ao <i>ledger</i>				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar transação	transaction	(tabela auxiliar 5)	true	PASS

Tabela auxiliar 5

sender.addTokens(1)
basicTransaction = new BasicTransaction(sender, receiver)
basicOrganization.addTransaction(basicTransaction)
basicOrganization.registerTransactionsInLedger()
basicOrganization.addTransaction(basicTransaction)

addTransactionECP04

Test Case ID	TC#testAddTransactionECP04	Dependências	
---------------------	----------------------------	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	12:30h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado: um objeto do tipo Organization, um objeto do tipo Entity e um objeto do tipo BasicTransaction.				
Procedimentos				
1-Adicionar uma transação inválida para registo ao <i>ledger</i> . 2-Registar a transação no <i>ledger</i> 3-Adicionar novamente a transação ao <i>ledger</i>				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar transação ao ledger	transaction	(tabela auxiliar 5)	<i>false</i>	PASS

removeTransactionBVA01

Test Case ID	TC#testRemoveTransactionBVA01	Dependências	
--------------	-------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	23:40h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de ser instanciada uma Organization válida.				
Procedimentos				
Para ser removida uma Transaction, é preciso passar um parâmetro do tipo Transaction. Neste teste é passado o input null.				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Remover transação	transaction	null	Throw IllegalArgumentException	PASS

removeTransactionBVA02

Test Case ID	TC#testRemoveTransactionBVA02	Dependências	
--------------	-------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
8 dezembro	00:00h	Hugo	PASS	

Pré-condições / estado inicial				
Tem de ser instanciada uma Organization válida. O array de transações tem de estar vazio				
Procedimentos				
Para ser removida uma Transaction, é preciso passar um parâmetro do tipo Transaction. Neste teste é passado como <i>input</i> uma transação válida.				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Remover transação	transaction	transaction	false	PASS

removeTransactionECP01

Test Case ID	TC#testRemoveTransactionECP01	Dependências	
--------------	-------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	19:35h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado um objeto Organization. Tem de ser instanciado duas Entities. Tem de ser instanciado uma BasicTransaction. Já deverá existir pelo menos uma transação no ledger à Organization.				
Procedimentos				
Para ser removida uma Transaction, é preciso passar um parâmetro do tipo Transaction, que neste caso deverá ser igual a uma transação já adicionada no <i>ledger</i> .				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untest ed
Remover transação	transaction	transaction = basicTransaction	<i>true</i>	PASS

removeTransactionECP02

Test Case ID	TC#testRemoveTransactionECP02	Dependências	
--------------	-------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	19:45h	Fábio	PASS	

Pré-condições / estado inicial				
Tem de ser instanciado: um objeto Organization, duas Entities e uma BasicTransaction.				
Procedimentos				
Para ser removida uma Transaction, é preciso passar um parâmetro do tipo Transaction, que neste caso não deverá ser igual a uma transação já adicionada (caso exista alguma), como tal não deverá ser removida nenhuma transação e o método deverá retornar <i>false</i> .				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untest ed
Remover transação	transaction	basicTransaction = new BasicTransaction(sender, receiver);	<i>false</i>	PASS

getTransactionBVA01

Test Case ID	TC#testGetTransactionBVA01	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	19:45h	Fábio	PASS	
Pré-condições / estado inicial				
Tem de ser instanciada uma Organization válida.				
Procedimentos				
Invocar método getTransaction passando null como <i>input</i> .				
Requisito/Us e Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter transação	transaction	null	Throw IllegalArgumentException	PASS

getTransactionECP01

Test Case ID	TC#testGetTransactionECP01	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	19:55h	Fábio	PASS	
Pré-condições / estado inicial				
Tem de ser instanciado : um objeto do tipo Organization e dois objetos do tipo Entity . Tem de ser instanciada uma BasicTransaction. Tem de ser inserida pelo menos uma Transaction no <i>ledger</i> .				
Procedimentos				
Invocar método getTransaction passando como <i>input</i> uma transação que existe no livro.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter transação de um ledger	transaction	transaction = basicTransaction *	Instanceof Basic Transaction	PASS

* Objeto "**basicTransaction**": basicTransaction = new BasicTransaction(sender, receiver);

getTransactionECP02

Test Case ID	TC#testGetTransactionECP02	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	19:35h	Fábio	PASS	
Pré-condições / estado inicial				
Tem de ser instanciado : um objeto do tipo Organization e dois objetos do tipo Entity .				
Tem de ser instanciada uma BasicTransaction.				
Procedimentos				
Para ser obtida uma transação , é necessário passar um parâmetro do tipo Transaction, que neste caso não deverá corresponder a uma transação adicionada no <i>ledger</i> .				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter transação	transaction	transaction = basicTransaction	*	PASS

*documentação não especifica o resultado esperado para este método caso não seja encontrada a transação passada no *input*.

isValidEdgerBVA01

Test Case ID	TC#testIsValidEdger BVA01	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	03:35h	Hugo	PASS	
Pré-condições / estado inicial				
Tem de ser instanciado : um objeto do tipo BasicTransaction e um objeto do tipo BasicTransactionLine.				
Procedimentos				
Inserir linha na transação. Adicionar transação ao ledger. Registar transação no ledger. Editar estado da transação. Verificar estado do bloco (deve estar adulterado). Verificar ledger (deve ser inválido).				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Verificar se <i>ledger</i> é válido	transaction	basicTransaction	false	PASS

isValidEdgerBVA02

Test Case ID	TC#testIsValidEdgerBVA02	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	03:35h	Hugo	PASS	
Pré-condições / estado inicial				
Tem de ser instanciado : dois objetos do tipo BasicTransaction e um objeto do tipo BasicTransactionLine.				
Procedimentos				
Inserir linha na transação. Adicionar transação ao ledger. Registrar transação no ledger. Adicionar nova transação ao ledger. Registrar transação no ledger. Editar estado da última transação adicionada. Verificar estado do bloco (deve estar adulterado). Verificar ledger (deve ser inválido, return false).				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Verificar se <i>ledger</i> é válido	transaction	basicTransaction	false	PASS

isValidEdgerBVA03

Test Case ID	TC#testIsValidEdgerBVA03	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	03:35h	Hugo	PASS	
Pré-condições / estado inicial				
Tem de ser instanciado : dois objetos do tipo BasicTransaction e um objeto do tipo BasicTransactionLine.				
Procedimentos				
Verificar ledger (deve ser válido, return true).				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Verificar se <i>ledger</i> é válido	transaction	basicTransaction*	true	PASS

* Objeto "**basicTransaction**": basicTransaction = new BasicTransaction(sender, receiver);

IsValidEdgerECP01

Test Case ID	TC#testIsValidEdgerECP01	Dependências	
---------------------	--------------------------	---------------------	--

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

Pré-condições / estado inicial				
Tem de ser instanciado : dois objetos do tipo BasicTransaction e um objeto do tipo BasicTransactionLine.				
Procedimentos				
Adicionar e registar transações ao ledger (3 iterações). Adulterar a transação número 2 (que está inserida no bloco 2); Verificar ledger (deve ser inválido, return false).				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untestado
Verificar se ledger é válido	transaction	basicTransaction!=null	false	PASS

5.Detalhes Class BasicTransactionLine

BasicTransactionLineBVA01

Test Case ID	TC#testBasicTransactionLineBVA01	Dependências	
--------------	----------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	13:47h	Hugo	PASS	

Pré-condições / estado inicial				
Procedimentos				
O Construtor BasicTransactionLine recebe três parâmetros: uma String, um inteiro e um double, sendo que a String deve ser passada vazia.				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar uma linha para adicionar à transação	(String itemDescription, int quantity, double unitPrice)	itemDescription = "" quantity = 1 unitPrice = 3	IllegalArgumentException	PASS

BasicTransactionLineBVA02

Test Case ID	TC#testBasicTransactionLineBVA02	Dependências	
--------------	----------------------------------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
13 dezembro	13:57h	Hugo	PASS	

Pré-condições / estado inicial				
Procedimentos				
Neste teste é passado o parâmetro <i>quantity</i> com o valor de -1 e os outros inputs válidos.				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar uma linha para adicionar à transação	(String itemDescription, int quantity, double unitPrice)	itemDescription = "transacao" quantity = -1 unitPrice = 3	IllegalArgumentException	PASS

BasicTransactionLineBVA03

Test Case ID	TC#testBasicTransactionLineBVA03	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	14:11h	Hugo	PASS	
Pré-condições / estado inicial				
Procedimentos				
O Construtor BasicTransactionLine recebe três parâmetros: uma String, um inteiro e um double sendo que o valor <i>double</i> tem o valor -0,1				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar uma linha para adicionar à transação	(String itemDescription, int quantity, double unitPrice)	itemDescription = "transacao " quantity = 1 unitPrice = -0,1	IllegalArgumentException	PASS

BasicTransactionLineBVA04

Test Case ID	TC#testBasicTransactionLineBVA04	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
13 dezembro	14:44h	Hugo	PASS	
Pré-condições / estado inicial				
Procedimentos				
O Construtor BasicTransactionLine recebe três parâmetros: uma String, um inteiro e um double. Neste teste deve ser passado o parâmetro itemDescription com o valor null e os outros inputs válidos.				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar uma linha para adicionar à transação	(String itemDescription, int quantity, double unitPrice)	itemDescription = null quantity = 1 unitPrice = 3	IllegalArgumentException	PASS

BasicTransactionLineBVA05

Test Case ID	TC#testBasicTransactionLineBVA05	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	14:30h	Hugo	PASS	
Pré-condições / estado inicial				
Procedimentos				
Neste teste deve ser passado o parâmetro quantity com o valor 0 e os outros inputs válidos.				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar uma linha para adicionar à transação	(String itemDescription, int quantity, double unitPrice)	itemDescription = "transacao " quantity = 0 unitPrice = 1	IllegalArgumentExcepcion	PASS

BasicTransactionLineBVA06

Test Case ID	TC#testBasicTransactionLineBVA06	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	14:39h	Hugo	PASS	
Pré-condições / estado inicial				
Procedimentos				
O Construtor BasicTransactionLine recebe três parâmetros: uma String, um inteiro e um double				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar uma linha para adicionar à transação	(String itemDescription, int quantity, double unitPrice)	itemDescription = "transacao " quantity = 1 unitPrice = 0	IllegalArgumentExcepcion	PASS

testBasicTransactionLineECP01

Test Case ID	TC#	Dependências	
	BasicTransactionLineECP01		

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
2 janeiro	14:39h	Hugo	PASS	

Pré-condições / estado inicial				
Procedimentos				
O Construtor BasicTransactionLine recebe três parâmetros: uma String, um inteiro e um double				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Criar uma linha para adicionar à transação	(String itemDescription, int quantity, double unitPrice)	itemDescription = "transacao " quantity = 4 unitPrice = 1	true	PASS

6. Summary Tests

Tests cases planeados	Tests Cases executados	TC que passaram	TC que falharam	TC não testados
52	52	52	0	0