

<LEI\_ESII2021\_TP2\_GRUPO11>

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](#_Toc92548452)

[1.1. Identificador do documento 5](#_Toc92548453)

[1.2. Âmbito 5](#_Toc92548454)

[1.3. References 5](#_Toc92548455)

[1.4 Glossário 5](#_Toc92548456)

[2. Detalhes Class BasicTransaction 6](#_Toc92548457)

[BasicTransactionBVA01 6](#_Toc92548458)

[BasicTransactionECP01 6](#_Toc92548459)

[addTransactionLineBVA01 7](#_Toc92548460)

[addTransactionLineECP01 8](#_Toc92548461)

[addTransactionLineECP02 8](#_Toc92548462)

[removeTransactionLineECP01 9](#_Toc92548463)

[removeTransactionLineBVA01 9](#_Toc92548464)

[removeTransactionLineBVA02 10](#_Toc92548465)

[removeTransactionLineECP02 10](#_Toc92548466)

[getTransactionLineBVA01 11](#_Toc92548467)

[getTransactionLineECP01 12](#_Toc92548468)

[getTransactionLineECP02 12](#_Toc92548469)

[getTransactionCountECP01 13](#_Toc92548470)

[getTotalValueECP01 13](#_Toc92548471)

[3 Detalhes Class BasicEntity 14](#_Toc92548472)

[BasicEntityBVA01 14](#_Toc92548473)

[BasicEntityBVA02 14](#_Toc92548474)

[BasicEntityECP01 15](#_Toc92548475)

[addTokensBVA01 15](#_Toc92548476)

[addTokensBVA02 16](#_Toc92548477)

[addTokensBVA03 16](#_Toc92548478)

[addTokensECP01 17](#_Toc92548479)

[4.Detalhes Class BasicOrganization 18](#_Toc92548480)

[getBlockBVA01 18](#_Toc92548481)

[getBlockBVA02 18](#_Toc92548482)

[getBlockBVA03 19](#_Toc92548483)

[getBlockECP01 19](#_Toc92548484)

[testRegisterTransactionsInLedgerECP01 20](#_Toc92548485)

[testRegisterTransactionsInLedgerECP02 21](#_Toc92548486)

[getBlockCountECP01 22](#_Toc92548487)

[getBlockCountECP02 23](#_Toc92548488)

[addTransactionBVA01 24](#_Toc92548489)

[addTransactionECP01 24](#_Toc92548490)

[addTransactionECP02 25](#_Toc92548491)

[addTransactionECP03 25](#_Toc92548492)

[addTransactionECP04 26](#_Toc92548493)

[removeTransactionBVA01 27](#_Toc92548494)

[removeTransactionBVA02 27](#_Toc92548495)

[removeTransactionECP01 28](#_Toc92548496)

[removeTransactionECP02 28](#_Toc92548497)

[getTransactionBVA01 29](#_Toc92548498)

[getTransactionECP01 29](#_Toc92548499)

[getTransactionECP02 30](#_Toc92548500)

[isValidEdgerBVA01 30](#_Toc92548501)

[isValidEdgerBVA02 31](#_Toc92548502)

[isValidEdgerBVA03 31](#_Toc92548503)

[IsValidEdgerECP01 32](#_Toc92548504)

[5.Detalhes Class BasicTransactionLine 33](#_Toc92548505)

[BasicTransactionLineBVA01 33](#_Toc92548506)

[BasicTransactionLineBVA02 33](#_Toc92548507)

[BasicTransactionLineBVA03 34](#_Toc92548508)

[BasicTransactionLineBVA04 34](#_Toc92548509)

[BasicTransactionLineBVA05 35](#_Toc92548510)

[BasicTransactionLineBVA06 35](#_Toc92548511)

[testBasicTransactionLineECP01 36](#_Toc92548512)

[6. Summary Tests 37](#_Toc92548513)

# 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/Use 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 pârametro 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 pârametro uma linha válida e verificar o resultado | | | | |
| **Requisito/Use Case/ Funcionalidade** | **Inputs** | **Valores do Inputs** | **Resultados Esperados** | **Pass/Fail/Untested** |
| 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 pârametro uma linha que não está adicionada na transação | | | | |
| **Requisito/Use Case/ Funcionalidade** | **Inputs** | **Valores do Inputs** | **Resultados Esperados** | **Pass/Fail/Untested** |
| 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/Use 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 correponder 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/Use 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/Use 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 *String,* 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 *Token*, é preciso passar um parâmetro do tipo *int*.  Passos:  Invocar método addTokens do objeto do tipo Entity passando -1 como *input*. | | | | |
| **Requisito/Use Case/ Funcionalidade** | **Inputs** | **Valor do Inputs** | **Resultados Esperados** | **Pass/Fail/Untested** |
| Adicionar um Token | *int* | -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 | *int* | 0 | nº *tokens* 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 *input*.  2-Verificar número de transações adicionadas. | | | | |
| **Requisito/Use Case/ Funcionalidade** | **Inputs** | **Valor do Input** | **Resultados Esperados** | **Pass/Fail/Untested** |
| Adicionar um Token | *int* | 2147483647 | nº *tokens* 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 *input*.  2-Verificar número de transações adicionadas. | | | | |
| **Requisito/Use Case/ Funcionalidade** | **Inputs** | **Valores do Inputs** | **Resultados Esperados** | **Pass/Fail/Untested** |
| Adicionar um Token | *int* | *int* = 3 | return nº *tokens* 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 registar). | | | | |
| **Procedimentos** | | | | |
| Para registar 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** |
| Registar transações no *ledger* | basicTransaction | (tabela auxiliar 1) | 1 | PASS |

**Tabela auxiliar 1**

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

## 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 registar transação). | | | | |
| **Procedimentos** | | | | |
| Para registar 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** |
| Registar transações no *ledger* | 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 registar). | | | | |
| **Procedimentos** | | | | |
| Para registar uma transação é preciso adicioná-la ao *ledger*.  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** |
| Registar transações no *ledger* | basicTransaction | (tabela auxiliar 3) | 2 | PASS |

**Tabela auxiliar 3**

|  |
| --- |
| *sender.addTokens(1)* |
| basicTransaction = new BasicTransaction(sender, receiver) |
| basicOrganization.addTransaction(basicTransaction) |
| basicOrganization.registerTransactionsInLedger() |
| getBlockCount() |

## 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 registar). | | | | |
| **Procedimentos** | | | | |
| Para registar uma transação é preciso adicioná-la ao *ledger*.  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 *ledger* | 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 menas uma Transaction adicionada ao *ledger*. | | | | |
| **Procedimentos** | | | | |
| Invocar método addTransaction do objeto do tipo Organization, passando como parâmetro uma transação já existente no *ledger*. | | | | |
| **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 *ledger*.  3-Registar a transação no *ledger*  4-Adicionar novamente a transação ao *ledger* | | | | |
| **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 *ledger*.  2-Registar a transação no *ledger*  3-Adicionar novamente a transação ao *ledger* | | | | |
| **Requisito/Use Case/ Funcionalidade** | **Inputs** | **Valores do Inputs** | **Resultados Esperados** | **Pass/Fail/Untested** |
| Adicionar transação ao ledger | transaction | (tabela auxiliar 5) | *false* | 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 arrray 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 *input* 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 *ledger.* | | | | |
| **Requisito/Use Case/ Funcionalidade** | **Inputs** | **Valores do Inputs** | **Resultados Esperados** | **Pass/Fail/Untested** |
| Remover transação | transaction | transaction= basicTransaction | *true* | 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 *false*. | | | | |
| **Requisito/Use Case/ Funcionalidade** | **Inputs** | **Valores do Inputs** | **Resultados Esperados** | **Pass/Fail/Untested** |
| Remover transação | transaction | basicTransaction = new BasicTransaction(sender, receiver); | *false* | 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 *input*. | | | | |
| **Requisito/Use 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 *ledger*. | | | | |
| **Procedimentos** | | | | |
| Invocar método getTransaction passando como *input* 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 *ledger*. | | | | |
| **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 ínvalido). | | | | |
| **Requisito/Use Case/ Funcionalidade** | **Inputs** | **Valores do Inputs** | **Resultados Esperados** | **Pass/Fail/Untested** |
| Verificar se *ledger* é válido | transaction | basicTransaction | false | PASS |

## isValidEdgerBVA02

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | TC#testIsValidEdger BVA02 | **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.  Registar transação no ledger.  Adicionar nova transação ao ledger.  Registar transação no ledger.  Editar estado da última transação adicionada.  Verificar estado do bloco (deve estar adulterado).  Verificar ledger ( deve ser ínvalido, return false). | | | | |
| **Requisito/Use Case/ Funcionalidade** | **Inputs** | **Valores do Inputs** | **Resultados Esperados** | **Pass/Fail/Untested** |
| Verificar se *ledger* 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 *ledger* 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 ínvalido, return false). | | | | |
| **Requisito/Use Case/ Funcionalidade** | **Inputs** | **Valores do Inputs** | **Resultados Esperados** | **Pass/Fail/Untested** |
| 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 *quantity* 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 *double* 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 | IllegalArgumentException | 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 | IllegalArgumentException | PASS |

## testBasicTransactionLineECP01

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | TC# BasicTransactionLineECP01 | **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 = 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 |