

REBOOK

TEST CASE OUTLINE

Version 1.0

07/12/2020

Histórico de Versões

Versão #	Implementado por	Data da Revisão	Aprovado por	Data de Aprovação	Razão
0.1	João Leocádio	24/11/2020	Paulo Brito, Manuel Garrido	24/11/2020	Primeiro Documento
0.2	João Leocádio	26/11/2020	Paulo Brito, Manuel Garrido	26/11/2020	Introdução
0.3	Manuel Garrido	01/12/2020	João Leocádio, Paulo Brito	01/12/2020	Adicionado algumas tabelas
0.4	João Leocádio	02/12/2020	Manuel Garrido, Paulo Brito	02/12/2020	Adicionado algumas tabelas
0.5	Paulo Brito	03/12/2020	João Leocádio, Manuel Garrido	03/12/2020	Adicionado algumas tabelas
0.6	João Leocádio	07/12/2020	Manuel Garrido, Paulo Brito	08/12/2020	Revisão da ortografia

Índice

1. Introdução.....	4
1.1. Identificador do documento	4
1.2. Âmbito.....	4
1.3. References.....	4
1.4 Glossário.....	4
2. Detalhes.....	5
AddAccount (User user)	5
UserShouldBeAddedTest	5
2.2 Login(string username, string password)	6
2.3 GetUserById(int id).....	7
2.4 GetAllUsers().....	8
2.5 EditUser(User user)	9
2.6 DeleteUser(int id).....	10
2.7 GetMatches(int id)	11
2.8 AddBookToFavoritesTests(int userId, int bookId)	12
2.9 RemoveBookFromFavoritesTests(int userId, int bookId)	13
2.10 GetAllFavouritesTests(int userId)	15
2.11 Rate(int raterId, int ratedId, int rating).....	16
3.1 AddBook(Book book).....	20
3.2 GetBooksTests()	23
3.3 GetBookById(int id);	24
3.3 DeleteBook(int id);	25
3.4 EditBook(int id)	27
4.1 SubmitReportTests(Report report);.....	28
4.2 GetAllReports()	30
4.3 GetReportByUserId(int id);	31
5.1 SendMessage(Message message)	33
5.2 ChangeStatusByMessageId(int id);	35
5.3 GetMessages()	36
5.4 GetMessagesByUserId(int id);.....	37

1. Introdução

1.1. Identificador do documento

O identificador único deste documento é p seguinte: TestCaseOutline#01.

1.2. Âmbito

O software proposto e desenvolvido, tem como principal objetivo promover a troca de livros entre várias pessoas, ou seja “dar uma nova vida” a livros que nunca mais usamos e/ou estão encostados na prateleira.

Por outro lado, o proposito deste documento insere-se no planeamento de um conjunto de casos de teste seguindo uma abordagem de “*black-box testing*” controlando desta forma o comportamento do programa e dos componentes que definem os vários conjuntos de Use Cases.

Deste modo, vão ser testados vários métodos das seguintes classes: MessageRepo, ReportRepo, UserRepo e BookRepo.

1.3. References

GitLab do Projeto: <http://gitlab.estg.ipp.pt/8180695/lds-13-rebook.git>

1.4 Glossário

Equivalence Class Partitioning – ECP

Boundary Value Analysis - BVA

2. Detalhes

A abordagem utilizada para o conjunto de casos de teste implementados seguiu a norma de “black box testing” utilizando as técnicas de ECP e BVA.

Os resultados dos casos de teste implementados foram visualizados no terminal do IntelliJRider utilizando o comando.

AddAccount (User user)

UserShouldBeAddedTest

Test Case ID	B1.1	Dependências	
--------------	------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
17/11/2020	18:03	Manuel Garrido	Pass	

Pré-condições / estado inicial				
Base de dados iniciada e vazia.				
Procedimento / Passos do user para esta função				
Chamar método usando user válido como argumento.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar conta à base de dados	user	New User (“username”, “password”)	Utilizador adicionado à base de dados	Pass

UsernameAlreadyExistsTest

Test Case ID	B1.2	Dependências	
--------------	------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
17/11/2020	18:04	Manuel Garrido	Pass	

Pré-condições / estado inicial				
Base de dados iniciada e com um user de username = “username”.				
Procedimento / Passos do user para esta função				
Chamar método usando user com nome já existente como argumento. ECP2				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar conta à base de dados	user	New User (“username”, “password”)	Excepção	Pass

2.2 Login(string username, string password)

ShouldBeTrueTest

Test Case ID	B2.1	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
17/11/2020	18:04	Manuel Garrido	Pass	
Pré-condições / estado inicial				
Base de dados existe e com users				
Procedimento / Passos do user para esta função				
Chamar método LogIn com combinação username e password existente na base de dados como argumentos.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Autenticação de conta	String username, string password	("username","password")	True	Pass

ShouldBeFalseTests

Test Case ID	B2.2	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
17/11/2020	18:05	Manuel Garrido	Pass	
Pré-condições / estado inicial				
Base de dados existe e sem users				
Procedimento / Passos do user para esta função				
Chamar método Login				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Autenticação de conta	String username, string password	("username","password")	False	Pass

2.3 GetUserById(int id)

UserExistsTest

Test Case ID	B3.1	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
17/11/2020	18:06	Manuel Garrido	Pass	
Pré-condições / estado inicial				
Base de dados existe e com users				
Procedimento / Passos do user para esta função				
Chamar método GetUserById com id correspondente a um user registado na base de dados.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Receber user com um id específico	Int id	1	Instancia de User	Pass

UserDoesNotExistTest

Test Case ID	B3.2	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
17/11/2020	18:08	Manuel Garrido	Pass	
Pré-condições / estado inicial				
Base de dados existe e sem users				
Procedimento / Passos do user para esta função				
1. Chamar método GetUserById com id que não correspondente a um user registado na base de dados.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Receber user com um id específico	Int id	1	Excepção	Pass

2.4 GetAllUsers()

UsersReturnedTest

Test Case ID	B4.1	Dependências	
--------------	------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
17/11/2020	18:08	Manuel Garrido	Pass	

Pré-condições / estado inicial				
Base de dados existe e com users				
Procedimento / Passos do <i>user</i> para esta função				
Inserem-se 5 Users na bd Chamar método GetAllUsers.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Receber todos os users registados			Lista com 5 instancias de user	Pass

2.5 EditUser(User user)

UserEditedTest()

Test Case ID	B5.1	Dependências	
--------------	------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
17/11/2020	18:08	Manuel Garrido	Pass	

Pré-condições / estado inicial				
Base de dados existe e com users				
Procedimento / Passos do user para esta função				
Chamar método EditUser com user registado na base de dados.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Editar dados de user	User com todos os campos preenchidos	("user")	User inicial com os dados actualizados	Pass

UserDoesNotExist

Test Case ID	B5.2	Dependências	
--------------	------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
17/11/2020	18:08	Manuel Garrido	Pass	

Pré-condições / estado inicial				
Base de dados existe sem users				
Procedimento / Passos do user para esta função				
Chamar método EditUser				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Editar dados de user	User com todos os campos preenchidos	("user")	Excepção	Pass

2.6 DeleteUser(int id)

UserIsDeletedTest

Test Case ID	B6.1	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
17/11/2020	18:09	Manuel Garrido	Pass	
Pré-condições / estado inicial				
Base de dados existe e com users				
Procedimento / Passos do user para esta função				
Chamar método DeleteUser com id correspondente a um user registado na base de dados.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Eliminar user	id	1	user	Pass

UserDoesNotExistTest

Test Case ID	B6.2	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
17/11/2020	18:09	Manuel Garrido	Pass	
Pré-condições / estado inicial				
Base de dados existe sem users				
Procedimento / Passos do user para esta função				
Chamar método DeleteUser com id que não corresponde a um user registado na base de dados.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Eliminar user	id	1	Excepção	Pass

2.7 GetMatches(int id)

ShouldBeMatchTest

Test Case ID	B7.1	Dependências	
--------------	------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
02/12/2020	20:00	Manuel Garrido	Pass	

Pré-condições / estado inicial				
Base de dados existe e com users que partilham uma match				
Procedimento / Passos do <i>user</i> para esta função				
Chamar método GetMatches com id correspondente a um user registado na base de dados.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Receber todos os Matches de um certo user	id	1	Instancia de Match	Pass

BookAndUserInDb

Test Case ID	B8.1	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
3/12/2020	21:30	João Leocádio	Pass	
Pré-condições / estado inicial				
Base de dados existe. Inserir Book e dois Users na base de dados Atribuir book a user 2				
Procedimento / Passos do user para esta função				
Chamar método AddBookToFavourites com userId e bookId que corresponde a um user registado na base de dados e um book registado na base de dados, respetivamente.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar um livro à lista de favoritos	id	1	Livro adicionado a favoritos	Pass

2.8 AddBookToFavoritesTests(int userId, int bookId)

BookNotFoundTest

Test Case ID	B8.2	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
25/11/2020	16:31	Manuel Garrido	Pass	
Pré-condições / estado inicial				
Base de dados existe com users				
Procedimento / Passos do user para esta função				
Chamar método AddBookToFavourites com userId e bookId correspondente a um user registado na base de dados mas o bookId não existe				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar um livro à lista de favoritos	id	1	Excepção	Pass

UserNotFoundTest

Test Case ID	B8.3	Dependências	
---------------------	------	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
25/11/2020	16:31	Manuel Garrido	Pass	

Pré-condições / estado inicial				
Base de dados existe e com books				
Procedimento / Passos do user para esta função				
Chamar método AddBookToFavourites com userId e bookId correspondente a um book registado na base de dados mas o userId não existe				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Remover um livro da lista de favoritos	id	1	Excepção	Pass

2.9 RemoveBookFromFavoritesTests(int userId, int bookId)

BookAndUserInDbBookInFavoritesTest

Test Case ID	B9.1	Dependências	
---------------------	------	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
03/12/2020	22:59	João Leocádio	Pass	

Pré-condições / estado inicial				
Base de dados existe. User e Book existentes na base de dados				
Procedimento / Passos do user para esta função				
Chamar método RemoveBookFromFavourites com userId e bookId correspondente a um user registado na base de dados e um bookId registado, respetivamente.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Remover um livro da lista de favoritos	id	1,1	Livro removido dos favoritos	Pass

BookNotFoundTest

Test Case ID	B9.2	Dependências	
--------------	------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
25/11/2020	16:31	Manuel Garrido	Pass	

Pré-condições / estado inicial				
Base de dados existe contendo users mas não books				
Procedimento / Passos do <i>user</i> para esta função				
Chamar método RemoveBookFromFavourites com userId correspondente a um user registado na base de dados				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Remover um livro da lista de favoritos	UserId, bookId	1,1	Excepção	Pass

UserNotFoundTest()

Test Case ID	B9.3	Dependências	
--------------	------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
25/11/2020	16:32	Manuel Garrido	Pass	

Pré-condições / estado inicial				
Base de dados existe contendo books				
Procedimento / Passos do <i>user</i> para esta função				
Chamar método RemoveBookFromFavourites com userId e bookId correspondente a um book registado na base de dados mas o userId não existe				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Remover um livro da lista de favoritos	UserId, bookId	1,1	Excepção	Pass

BookNotInFavourites

Test Case ID	B9.4	Dependências	
--------------	------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
25/11/2020	16:32	Manuel Garrido	Pass	

Pré-condições / estado inicial				
Base de dados existe e com books				
Procedimento / Passos do user para esta função				
Chamar método RemoveBookFromFavourites com userId e bookId correspondente a um user registado na base de dados e um bookId registado, respetivamente. BookId não está na lista de favoritos.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Remover um livro da lista de favoritos	UserId, BookId	1,1	Excepção	Pass

2.10 GetAllFavouritesTests(int userId)

UserExistsInDbTest

Test Case ID	B10.1	Dependências	
--------------	-------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
03/12/2020	23.04	João Leocádio	Pass	

Pré-condições / estado inicial				
Base de dados existe e com users				
Procedimento / Passos do user para esta função				
Chamar método GetAllFavourites com user registado na base de dados.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Ver lista de favoritos	userId	1	Lista de favoritos com um elemento	Pass

UserDoesNotExistInDbTest

Test Case ID	B10.2	Dependências	
--------------	-------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
25/11/2020	16:37	Manuel Garrido	Pass	

Pré-condições / estado inicial				
Base de dados existe e com users				
Procedimento / Passos do user para esta função				
Chamar método GetAllFavourites com user não registado na base de dados.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Ver lista de favoritos	userId	1	Excepção	Pass

2.11 Rate(int raterId, int ratedId, int rating)

ShouldWorkTest()

Test Case ID	B11.1	Dependências	
--------------	-------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
30/11/2020	14:30	Manuel Garrido	Pass	

Pré-condições / estado inicial				
Base de dados existe e com users				
Procedimento / Passos do user para esta função				
Chamar método RateUser com raterId, ratedId e rating válidos e existentes na base de dados.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Avaliar Utilizador	RaterId, ratedId, rating	1,2,3	true	Pass

RaterDoesNotExistTest

Test Case ID	B11.2	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
30/11/2020	14:30	Manuel Garrido	Pass	
Pré-condições / estado inicial				
Base de dados existe e com users				
Procedimento / Passos do user para esta função				
Chamar método RateUser com raterId inexistente e inválido.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Avaliar Utilizador	RaterId, ratedId, rating	2,1,3	Excepção	Pass

RatedDoesNotExistTest

Test Case ID	B11.3	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
30/11/2020	14:30	Manuel Garrido	Pass	
Pré-condições / estado inicial				
Base de dados existe e com users				
Procedimento / Passos do user para esta função				
Chamar método RateUser com ratedId inexistente e inválido.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Avaliar Utilizador	RaterId, ratedId, rating	1,2,3	Excepção	Pass

UserAlreadyRatedTest

Test Case ID	B11.4	Dependências	
---------------------	-------	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
30/11/2020	14:30	Manuel Garrido	Pass	

Pré-condições / estado inicial				
Base de dados existe e com users				
Procedimento / Passos do <i>user</i> para esta função				
Chamar método RateUser quando o utilizador avaliado já foi avaliado pelo utilizador que avalia.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Avaliar Utilizador	RaterId, ratedId,rating	1,2,3	Excepção	Pass

RatingBiggerthan5Test

Test Case ID	B11.5	Dependências	
---------------------	-------	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
30/11/2020	14:30	Manuel Garrido	Pass	

Pré-condições / estado inicial				
Base de dados existe e com users				
Procedimento / Passos do user para esta função				
Chamar método RateUser e o rating é maior que 5.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Avaliar Utilizador	RaterId, ratedId,rating	1,2,6	Excepção	Pass

RatingBiggerthan0Test

Test Case ID	B11.5	Dependências	
---------------------	-------	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
30/11/2020	14:30	Manuel Garrido	Pass	

Pré-condições / estado inicial				
Base de dados existe e com users				
Procedimento / Passos do user para esta função				
Chamar método RateUser e o rating é menor que 0.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Avaliar Utilizador	raterId, ratedId,rating	1,2,-1	Excepção	Pass

3.1 AddBook(Book book)

BookValidAndUserExistsTest()

Test Case ID	B12.1	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
23/11/2020	11:00	Paulo Brito	Pass	
Pré-condições / estado inicial				
Base de dados iniciada				
Procedimento / Passos do <i>user</i> para esta função				
Chamar método usando book válido como argumento. Campos Name, Author, State e Photo devidamente preenchidos e válidos.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar book à base de dados.	book	New Book (“name”, “Author”)	Book adicionado à DB	Pass

BookHasNoNameTest()

Test Case ID	B12.2	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
23/11/2020	11:00	Paulo Brito	Pass	
Pré-condições / estado inicial				
Base de dados iniciada				
Procedimento / Passos do <i>user</i> para esta função				
Chamar método usando book sem o campo name preenchido como argumento.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar book à base de dados.	book	New Book (“Author”, “State”)	true	

BookHasNoAuthorTest()

Test Case ID	B12.3	Dependências	
--------------	-------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
23/11/2020	11:00	Paulo Brito	Pass	

Pré-condições / estado inicial				
Base de dados iniciada				
Procedimento / Passos do user para esta função				
1. Chamar método usando book sem o campo Author preenchido como argumento.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar book à base de dados.	book	New Book ("name", "State", "Photo")	true	

BookHasNoStateTest()

Test Case ID	B12.4	Dependências	
--------------	-------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
23/11/2020	11:00	Paulo Brito	Pass	

Pré-condições / estado inicial				
Base de dados iniciada				
Procedimento / Passos do user para esta função				
1. Chamar método usando book sem o campo State preenchido como argumento.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar book à base de dados.	book	New Book ("name", "Author", "Photo")	true	

BookHasNoPhotosTest()

Test Case ID	B12.5	Dependências	
--------------	-------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
23/11/2020	11:00	Paulo Brito	Pass	

Pré-condições / estado inicial				
Base de dados iniciada				
Procedimento / Passos do <i>user</i> para esta função				
1. Chamar método usando book sem o campo Photo preenchido como argumento.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar book à base de dados.	book	New Book ("name", "Author", "State")	true	

UserDoesNotExistTest()

Test Case ID	B12.6	Dependências	
--------------	-------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
23/11/2020	11:00	Paulo Brito	Pass	

Pré-condições / estado inicial				
Base de dados iniciada				
Procedimento / Passos do <i>user</i> para esta função				
1. Chamar método usando book sem o campo user preenchido como argumento.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar book à base de dados.	book	New Book ("name", "Author", "State")	true	

3.2 GetBooksTests()

BooksReturnedTests()

Test Case ID	B13.1	Dependências	
--------------	-------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
23/11/2020	11:05	Paulo Brito	Pass	

Pré-condições / estado inicial				
Base de dados existe e com books				
Procedimento / Passos do <i>user</i> para esta função				
1. Chamar método GetBooks e o número de books na base de dados > 0.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Receber todos os books registados			true	

3.3 GetBookById(int id);

BookExistTest()

Test Case ID	B14.1	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
23/11/2020	11:05	Paulo Brito	Pass	
Pré-condições / estado inicial				
Base de dados existe e com books				
Procedimento / Passos do user para esta função				
Chamar método GetBook(int Id) com id correspondente a um book registado na base de dados.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Receber book com um id específico	Int id	("id")	true	

BookDoesNotExistTest()

Test Case ID	B14.2	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
23/11/2020	11:05	Paulo Brito	Pass	
Pré-condições / estado inicial				
Base de dados existe e com books				
Procedimento / Passos do user para esta função				
1. Chamar método GetBook com id que não correspondente a um book registado na base de dados.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Receber book com um id específico	Int id	("id")	false	

3.3 DeleteBook(int id);

BookExistsUserExistsAndHasBookInListTest()

Test Case ID	B15.1	Dependências	
--------------	-------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
23/11/2020	11:10	Paulo Brito	Pass	

Pré-condições / estado inicial				
Base de dados existe e com books				
Procedimento / Passos do user para esta função				
1. Chamar método DeleteBook com id correspondente a um book registado na base de dados.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Eliminar book	id	("id")	true	

BookDoesNotExistTest

Test Case ID	B15.2	Dependências	
--------------	-------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
23/11/2020	11:10	Paulo Brito	Pass	

Pré-condições / estado inicial				
Base de dados existe e com users				
Procedimento / Passos do user para esta função				
1. Chamar método DeleteBook com id correspondente que não corresponde a um book registado na base de dados.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Eliminar book	id	("id")	false	

UserDoesNotExistTest()

Test Case ID	B15.3	Dependências	
--------------	-------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
23/11/2020	11:10	Paulo Brito	Pass	

Pré-condições / estado inicial				
Base de dados existe e com users				
Procedimento / Passos do user para esta função				
1. Chamar método DeleteBook com um utilizador inexistente na base de dados.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Eliminar book	id	("id")	false	

UserDoesNotHaveBookInList

Test Case ID	B15.4	Dependências	
--------------	-------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
23/11/2020	11:10	Paulo Brito	Pass	

Pré-condições / estado inicial				
Base de dados existe e com users				
Procedimento / Passos do user para esta função				
1. Chamar método DeleteBook quando esse mesmo book não existe na lista do utilizador.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Eliminar book	id	("id")	false	

3.4 EditBook(int id)

BookEditedTest()

Test Case ID	B16.1	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
23/11/2020	11:10	Paulo Brito	Pass	
Pré-condições / estado inicial				
Base de dados existe e com books				
Procedimento / Passos do <i>user</i> para esta função				
1. Chamar método EditBook com book registado na base de dados.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Editar dados de book	user	("user")	true	

BookDoesNotExistTest()

Test Case ID	B16.2	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
23/11/2020	11:10	Paulo Brito	Pass	
Pré-condições / estado inicial				
Base de dados existe e com books				
Procedimento / Passos do <i>user</i> para esta função				
1. Chamar método EditBook com book inexistente na base de dados.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Editar dados de book	user	("user")	false	

4.1 SubmitReportTests(Report report);

ReportAndUserInDbTest

Test Case ID	B17.1	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
24/11/2020	12:00	João Leocádio	Pass	
Pré-condições / estado inicial				
Base de dados iniciada				
Procedimento / Passos do user para esta função				
1. Chamar método SubmitReport usando um report válido como argumento.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Submeter Report na base de dados	report	New Report ("reporterId", "reportedId", "Text");	true	Pass

TextNotFoundTest

Test Case ID	B17.2	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
24/11/2020	12:00	João Leocádio	Pass	
Pré-condições / estado inicial				
Base de dados iniciada				
Procedimento / Passos do user para esta função				
1. Chamar método SubmitReport sem o campo Text devidamente preenchido.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Submeter Report na base de dados	report	New Report ("reportedId", "Text");	false	Pass

ReporterNotFoundTest

Test Case ID	B17.3	Dependências	
---------------------	-------	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
24/11/2020	12:00	João Leocádio	Pass	

Pré-condições / estado inicial				
Base de dados iniciada				
Procedimento / Passos do user para esta função				
1. Chamar método SubmitReport e o utilizador que faz o report não existe na base de dados.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Submeter Report na base de dados	report	New Report ("reportedId", "Text");	false	Pass

ReportedNotFoundTest

Test Case ID	B17.4	Dependências	
---------------------	-------	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
24/11/2020	12:00	João Leocádio	Pass	

Pré-condições / estado inicial				
Base de dados iniciada				
Procedimento / Passos do user para esta função				
1. Chamar método SubmitReport e o utilizador reportado não existe na base de dados.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Submeter Report na base de dados	report	New Report ("reportedId", "Text");	false	Pass

4.2 GetAllReports()

ShouldReturnReportsTest

Test Case ID	B18.1	Dependências	
--------------	-------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
24/11/2020	12:00	João Leocádio	Pass	

Pré-condições / estado inicial				
Base de dados existe e com reports				
Procedimento / Passos do <i>user</i> para esta função				
1. Chamar método GetAllReports. O número de reports na base de dados tem que ser maior que zero.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Receber todos os reports registados			true	Pass

4.3 GetReportByUserId(int id);

ReportAndUserInDbTest

Test Case ID	B19.1	Dependências	
--------------	-------	--------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
24/11/2020	12:00	João Leocádio	Pass	

Pré-condições / estado inicial				
Base de dados existe, com reports e users inseridos				
Procedimento / Passos do <i>user</i> para esta função				
Chamar método GetReporByUserId(int Id) com id correspondente a um report registado na base de dados. Este método permite adicionar um report à lista de reports de um determinado utilizador.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar report à lista de reports	Int id	1	true	Pass

ReportedNotFoundTest

Test Case ID	B19.2	Dependências	
---------------------	-------	---------------------	--

Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
24/11/2020	12:00	João Leocádio	Pass	

Pré-condições / estado inicial				
Base de dados existe e com reports				
Procedimento / Passos do <i>user</i> para esta função				
Chamar método GetReporByUserId(int Id) com id que corresponde a um report mas o utilizador que reporta não existe na base de dados.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Receber report com um id específico	Int id	1	Excepção	Pass

5.1 SendMessage (Message message)

ShouldSendMessageTest

Test Case ID	B20.1	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
Pré-condições / estado inicial				
Base de dados existe com Users válidos				
Procedimento / Passos do user para esta função				
1. Chamar método SendMessage() com message válido como argumento.				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Enviar mensagem dada como argumento	Message	New Message {“text”, “senderId”, “receiverId”, “time”}	true	

ReceiverIdNotFoundTest

Test Case ID	B20.2	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
Pré-condições / estado inicial				
Base de dados existe com Users válidos				
Procedimento / Passos do user para esta função				
1. Chamar método SendMessage() e o utilizador que recebe não existe na base de dados.				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Enviar mensagem dada como argumento	Message	New Message {“text”, “senderId”, “receiverId”, “time”}	false	Pass

TestBTextEmpty

Test Case ID	B20.3	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
Pré-condições / estado inicial				
Base de dados existe com Users válidos				
Procedimento / Passos do user para esta função				
1. Chamar método SendMessage() e o utilizador que envia a mensagem não existe na base de dados.				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Enviar mensagem dada como argumento	Message	New Message {“text”, “senderId”, “receiverId”, “time”}	false	Pass

SenderIdNotFoundTest

Test Case ID	B20.4	Dependências		
Data de Execução	Hora de Execução	Tester	Pass/Fail	Resumo da Falha
Pré-condições / estado inicial				
Base de dados existe com Users válidos				
Procedimento / Passos do user para esta função				
1. Chamar método SendMessage() passando o text a null como argumento.				
Requisito/Use Case/Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Enviar mensagem dada como argumento	Message	New Message {“text”, “senderId”, “receiverId”, “time”}	Exceção	Pass

5.2 ChangeStatusByMessageId(int id);

TestB16_1()

Test Case ID	B21.1	Dependências	
--------------	-------	--------------	--

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

Pré-condições / estado inicial				
Base de dados existe e com "messages"				
Procedimento / Passos do user para esta função				
1. Chamar método ChangeStatusByMessageId(int id) para mudar o status da mensagem.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Mudar Status da Message com o id dado	Int id	{"id"}	true	

TestB16_2()

Test Case ID	B21.2	Dependências	
--------------	-------	--------------	--

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

Pré-condições / estado inicial				
Base de dados existe vazia				
Procedimento / Passos do user para esta função				
1. Chamar método ChangeStatusByMessageId(int id) com id que não corresponde a um utilizador que recebe a mensagem.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Mudar Status da Message com o id dado	Int id	{"id"}	Exceção	Pass

5.3 GetMessages()

ShouldWorkTest

Test Case ID	B22.1	Dependências	
--------------	-------	--------------	--

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

Pré-condições / estado inicial				
Base de dados existe vazia				
Procedimento / Passos do user para esta função				
1. Chamar método GetMessages(message) para retornar a lista de mensagens.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter Mensagens	Int id	{“id”}	true	Pass

5.4 GetMessagesByUserId(int id);

ShouldWorkTest

Test Case ID	B23.1	Dependências	
--------------	-------	--------------	--

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

Pré-condições / estado inicial				
Base de dados existe vazia				
Procedimento / Passos do user para esta função				
Chamar método GetMessagesByUserId(receiver.id) para listar todas as mensagens de um utilizador.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter Mensagens por id	Int id	{"id"}	true	Pass

MessagesDatabaseEmptyTest

Test Case ID	B23.2	Dependências	
--------------	-------	--------------	--

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

Pré-condições / estado inicial				
Base de dados existe vazia				
Procedimento / Passos do user para esta função				
1. Chamar método GetMessagesByUserId(receiver.id) sendo que não existem mensagens na base de dados.				
Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Obter Mensagens por id	Int id	{"id"}	Exceção	Pass