# <PROJETO> TEST CASE OUTLINE

Version <x.0>

<dd/mm/yyyy>

# Histórico de Versões

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	<author name=""></author>	<mm dd="" yy=""></mm>	<name></name>	<mm dd="" yy=""></mm>	draft

# Tabela de Conteúdos

1.	. Introdução	. 4
	1.1. Identificador do documento	
	1.2. Âmbito	.4
	1.3. References	.4
	1.4 Glossário	.4
2	Detalhes	. 4
	2.1. Identificador do caso de teste	. 4
	2.2. Objetivo ou requisito/funcionalidade associado	. 4
	2.3. Inputs	.4
	2.6. Estado inicial ou pré-condições	. 4
	2.5. Resultados esperados	. 4
	2.6. Procedimentos especiais ou STEPS	. 4
	2.7. Dependência de outros casos	.4

# 1. Introdução

#### 1.1. Identificador do documento

[Fornecer um identificador único para este documento. Este identificador único será usado por outros documentos para referenciar este de forma unívoca.]

#### 1.2. Âmbito

#### 1.3. References

[Fazer referência a outros documentos importantes para uma melhor compreensão deste. Exemplo: documento de requisitos, Master Test Plan, ...]

#### 1.4 Glossário

### 2. Detalhes

[cada uma das seções abaixo deverá ser repetida para cada test case. Em alternativa poderá usar-se uma tabela. Uma das principais diferenças entre as seções 2.3 e 2.4 do documento Test Case Design e o test case em si, está diretamente relacionado com os dados, ou seja, este documento pretende apresentar os dados de acordo com as descrições definidas no âmbito do documento test case design]

- 2.1. Identificador do caso de teste
- 2.2. Objetivo ou requisito/funcionalidade associado
- 2.3. Inputs
- 2.6. Estado inicial ou pré-condições
- 2.5. Resultados esperados
- 2.6. Procedimentos especiais ou STEPS
- 2.7. Dependência de outros casos

AddAccount(User user)

Test Case ID	B1.1	Dependências	

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

#### Pré-condições / estado inicial Base de dados iniciada e vazia. Procedimento / Passos do user para esta função 1. Chamar método usando user válido como argumento. Requisito/Use Resultados Pass/Fail/Untested Inputs Valores do Inputs Case/ **Esperados Funcionalidade** Adicionar conta à New User user true base de dados ("username", "password")

|--|

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

Base de dados iniciada e com users.

# Procedimento / Passos do user para esta função

1. Chamar método usando user com nome já existente como argumento.

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Adicionar	user	New User	Throws	
conta à base		("username",	exception	
de dados		"password")		

#### LogIn(string username, string password)

Test Case ID	B2.1	Dependências	
--------------	------	--------------	--

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

#### Pré-condições / estado inicial

Base de dados existe e com users

#### Procedimento / Passos do *user* para esta função

1. 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	

Test Case ID	B2.2	Dependências	
		2 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	

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

Base de dados existe e com users

#### Procedimento / Passos do *user* para esta função

1. Chamar método LogIn com combinação username e password inexistente 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")	False	

#### GetUserById(int id)

|--|

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

#### Pré-condições / estado inicial

Base de dados existe e com users

#### Procedimento / Passos do *user* para esta função

1. 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 especifico	Int id	("id")	true	

Test Case ID	B3.2	Dependências	

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

Base de dados existe e com 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 especifico	Int id	("id")	false	

#### GetAllUsers()

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

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

#### Pré-condições / estado inicial

Base de dados existe e com users

### Procedimento / Passos do *user* para esta função

1. Chamar método GetAllUsers.

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Receber todos os users registados			true	

Test Case ID B4.2	Dependências	
-------------------	--------------	--

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

Base de dados existe vazia

# Procedimento / Passos do user para esta função

1. Chamar método GetAllUsers.

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Receber todos os			false	
users registados				

#### EditUser(User user)

|--|

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

# Pré-condições / estado inicial

Base de dados existe e com users

# Procedimento / Passos do user para esta função

1. 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	("user")	true	

Test Case ID B5.2 Dependência	as
-------------------------------	----

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

Base de dados existe e com users

#### Procedimento / Passos do user para esta função

1. Chamar método EditUser com user inexistente na base de dados.

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Editar dados de	user	("user")	false	
user				

#### DeleteUser(int id)

|--|

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

#### Pré-condições / estado inicial

Base de dados existe e com users

#### Procedimento / Passos do user para esta função

1. 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	("id")	true	

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

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

#### Pré-condições / estado inicial

Base de dados existe e com users

#### Procedimento / Passos do user para esta função

1. Chamar método DeleteUser com id correspondente que não corresponde a um user registado na base de dados.

Requisito/Use	Inputs	Valores do Inputs	Resultados	Pass/Fail/Untested
Case/			Esperados	
Funcionalidade				
Eliminar user	id	("id")	false	

|--|

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

Base de dados existe e com users

#### Procedimento / Passos do *user* para esta função

1. 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	id	("id")	true	
Matches de um				
certo user				

	Dependências	B6.2	Test Case ID
--	--------------	------	--------------

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

#### Pré-condições / estado inicial

Base de dados existe e com users

#### Procedimento / Passos do *user* para esta função

1. Chamar método GetMatches 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 todos os	id	("id")	false	
Matches de um				
certo user				

AddBook(Book book);

|--|

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

#### Pré-condições / estado inicial

Base de dados iniciada

#### Procedimento / Passos do user para esta função

1. Chamar método usando book válido 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	

Test Case ID	B7.2	Dependências	
--------------	------	--------------	--

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

# Pré-condições / estado inicial

Base de dados iniciada

# Procedimento / Passos do user para esta função

1. 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	

Test Case ID	B7.3	Dependências		
Data de Execução	Hora de	Tester	Pass/Fail	Resumo da Falha
Data de Execução	Execução	rester	r assy i ali	Nesulilo da l'allia

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	

|--|

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

# 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	

Test Case ID	D/.3	Dependencias		
	I -	I		I
Data de Execução	Hora de	Tester	Pass/Fail	Resumo da Falha
	Execução			

	LACCUÇUO		
			i
			·

Base de dados iniciada

#### Procedimento / Passos do user 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	

# GetBooks();

Test Case ID	B8.1	Dependências	
	DO.1	- cp ca.ca.a	

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

# Pré-condições / estado inicial

Base de dados existe e com books

#### Procedimento / Passos do user para esta função

1. Chamar método GetBooks.

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Receber todos os			true	
books registados				

Test Case ID	B8.2	Dependências		
Data de Execução	Hora de	Tester	Pass/Fail	Resumo da Falha

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

Base de dados existe vazia

# Procedimento / Passos do user para esta função

1. Chamar método GetBooks.

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Receber todos os			false	
books registados				

# GetBookById(int id);

Case ID
---------

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

#### Pré-condições / estado inicial

Base de dados existe e com books

#### Procedimento / Passos do *user* para esta função

1. 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 especifico	Int id	("id")	true	

Test Case ID	B9.2	Dependências	

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

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 especifico	Int id	("id")	false	

# DeleteBook(int id);

st Case ID B10.1 Dependê	as
--------------------------	----

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

#### 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
	: d	/": d"\	+	
Eliminar book	id	("id")	true	

Test Case ID	B10.2	Dependências	

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

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	Inputs	Valores do Inputs	Resultados	Pass/Fail/Untested
Case/			Esperados	
Funcionalidade				
Eliminar book	id	("id")	false	

# EditBook(int id)

|--|

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

#### Pré-condições / estado inicial

Base de dados existe e com books

#### Procedimento / Passos do user para esta função

1. Chamar método EditBook com book registado na base de dados.

Requisito/Use Case/	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Funcionalidade				
Editar dados de	user	("user")	true	
book				

Test Case ID	R11 2	Dependências	
I est case ib	DII.Z	Dependencias	

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

Base de dados existe e com books

#### Procedimento / Passos do *user* 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	

# SubmitReport(Report report);

|--|

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

#### Pré-condições / estado inicial

Base de dados iniciada

#### Procedimento / Passos do *user* para esta função

1. Chamar método 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	

Test Case ID	B12.2	Dependências		
Data de Execução	Hora de	Tester	Pass/Fail	Resumo da Falha

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

Base de dados iniciada

#### Procedimento / Passos do user para esta função

1. Chamar método usando um reporterId inválido como argumento.

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	

# GetReports()

Test Case ID
--------------

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

# Pré-condições / estado inicial

Base de dados existe e com reports

# Procedimento / Passos do user para esta função

1. Chamar método GetReports.

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Receber todos os			true	
reports registados				

Test Case ID	B13.2	Dependências			
Data de Execução	Hora de	Tester	Pass/Fail	Resumo da Falha	
	Execução				
Pré-condições / est	tado inicial				
Base de dados exist	e vazia				
Procedimento / Pa	ssos do <i>user</i> para es	sta função			
1. Chamar método	•	•			
	·				
Requisito/Use	Inputs	Valores do Inputs	Resultados	Pass/Fail/Untested	
Case/		·	Esperados		
Funcionalidade					
Receber todos os			false		
reports registados					
GetReport(int id)					
Test Case ID	B14.1	Dependências			
1000 0000 12					
Data de Execução	Hora de	Tester	Pass/Fail	Resumo da Falha	
_ u.u. u.u	Execução				
	<u> </u>				
Pré-condições / est	ado inicial				
Base de dados exist					
	ssos do <i>user</i> para es	sta função			
	•	•	e a um renort regista	ado na hase de dados	
1. Chamar método GetReport(int Id) com id correspondente a um report registado na base de dados.					
Requisito/Use	Inputs	Valores do Inputs	Resultados	Pass/Fail/Untested	
Case/		1 a. 5 . 6 5 a. 6	Esperados	. acc, ran, chicotca	
Funcionalidade					
	1	///· III)			

("id")

true

Receber report

com um id especifico

Int id

Test Case ID	B14.2		Dependências			
			1			
Data de Execuçã			Tester	Pass/Fa	il	Resumo da Falha
	Execuç	ao				
Pré-condições /	estado inici	al				
Base de dados ex	iste e com	reports				
Procedimento /		•	•			
	do GetRepo	rt(int ld) co	om id que não corres <sub>i</sub>	ponde a u	m report r	egistado na base de
dados.						
Requisito/Use	Inputs		Valores do Inputs	Resulta	dos	Pass/Fail/Untested
Case/	•		<b>,</b>	Esperad	os	
Funcionalidade				-		
Receber report	Int id		("id")	false		
com um id						
especifico	specifico					
SendMessage(	Message	meccan	·e)			
Test Case ID	B17.1	Dependê				
1000 0000 12					I	
Data de	Hora de	Tester			Pass/Fail	Resumo da Falha
Execução	Execuçã					
	0					
Pré-condições /	estado inici	al				
Base de dados ex			 S			
Procedimento /	Passos do ι	<i>iser</i> para e	sta função			
1. Chamar métod	do SendMe	essage() c	om message válido	como a	rgumento	
		1			ı	
Requisito/Use	Inputs	Valores o	do Inputs			o Pass/Fail/Unteste
Case/ Funcionalidad					S	d .s
e					Esperado	<b>15</b>
Enviar	Message	New Mes	ssage		true	
mensagem	O ·		senderId","receiverId	","time"		
dada como		}				
argumento						
Test Case ID	B17.2	Dependê	ncias			
Data de	Horo do	Tostor			Dace /Feil	Posumo do Folha
Execução	Hora de Execuçã	Tester			Pass/Fail	Resumo da Falha
LACCUÇUO	O CACCUÇA					
Pré-condições / estado inicial						
	Base de dados existe com Users válidos  Procedimento / Passos do user para esta função					
Procedimento /	rassos do u	<i>iser</i> para e	sta funçao			

1. Chamar método SendMessage() com message como argumento que corresponde a utilizador inexistente na base de dados.

Requisito/Use Case/ Funcionalidad e	Inputs	Valores do Inputs	Resultado s Esperados	Pass/Fail/Unteste d
Enviar mensagem dada como argumento	Message	New Message {"text","senderId","receiverId","time" }	false	

ChengeStatusByMessageId(int id);

Test Case ID B16.1 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)

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Mudar Status da	Int id	{"id}	true	
Message com o id				
dado				

Test Case ID B1
-----------------

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 Message registado na base de dados.

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Mudar Status da	Int id	{"id}	false	
Message com o id				
dado				