HOUSEM8 - LDS

TEST CASE OUTLINE

Version 1.7

05/01/2021

Histórico de Versões

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	*Jorge Moreira*	*07/12/2020*	*Jorge Moreira*	*07/12/2020*	Adaptação do documento para o projeto
1.1	*Samuel Cunha*	*07/12/2020*	*Samuel Cunha*	*08/12/2020*	Adicionadas todas as secções para o começo da descrição dos cenários de testes
1.2	*Jéssica Coelho*	*08/12/2020*	*Jorge Moreira*	*08/12/2020*	Adicionados casos de teste para alguns componentes
1.3	*Miguel Costa*	*08/12/2020*	*Jorge Moreira*	*08/12/2020*	Adição de casos de testes para componentes em falta.
1.4	*Jorge Moreira*	*08/12/2020*	*Jorge Moreira*	*08/12/2020*	Finalização dos componentes UserDao, EmployerDao, MateDao, JobDao e WorkDao
1.5	*Miguel Costa*	*24/12/2020*	*Miguel Costa*	*24/12/2020*	Correções de testes relacionados com a pesquisa de trabalho
1.6	*Miguel Costa*	*27/12/2020*	*Miguel Costa*	*27/12/2020*	Correções de testes relacionados com o trabalho.
1.7	*Jéssica Coelho*	*05/01/2021*	*Jéssica Coelho*	*05/01/2021*	Adicionado teste para a componente de atualizar palavra passe.

Tabela de Conteúdos

1. Introdução *5*

1.1. Identificador do documento 5

1.2. Âmbito 5

1.3. Referencias 5

1.4 Glossário 5

2. Detalhes *6*

2.1 Test Cases do componente *UserDAO* 6

2.1.1 Método UpdatePassword 6

2.1.2 Método FindUserByEmail 7

2.1.3 Método FindById 8

2.2 Test Cases do componente *EmployerDAO* 10

2.2.1 Método Create 10

2.2.2 Método Update 11

2.2.3 Método FavoriteList 11

2.2.4 Método AddFavorite 12

2.2.5 Método RemoveFavorite 12

2.2.6 Método FindEmployerById 13

2.3 Test Cases do componente *MateDAO* 14

2.3.1 Método Create 14

2.3.2 Método CategoriesList 15

2.3.3 Método AddCategory 15

2.3.4 Método RemoveCategory 16

2.3.5 Método Update 16

2.3.6 Método FindMateById 17

2.3.7 Método PendingJobsList 17

2.3.8 Método GetMates 19

2.3.9 Método IgnoreJobPost 23

2.4 Test Cases do componente *JobDAO* 24

2.4.1 Método AddPayment 24

2.4.2 Método RemovePayment 24

2.4.3 Método GetJobs 25

2.4.4 Método MakeOfferOnJob 31

2.4.5 Método Create 32

2.4.6 Método Delete 33

2.4.7 Método FindById 33

2.4.8 Método GetEmployerPosts 34

2.4.9 Método UpdatePostDetails 34

2.5 Test Cases do componente *WorkDAO* 36

2.5.1 Método MarkJobAsDone 36

2.5.2 Método Create 38

2.5.3 Método FindById 40

2.6 Test Cases do componente *LoginDAO* 41

2.6.1 Método Authenticate 41

2.7 Test Cases do componente *ReportDAO* 44

2.7.1 Método ReportUser 44

2.8 Test Cases do componente *ReviewEmployerDAO* 45

2.8.1 Método ReviewEmployer 45

2.9 Test Cases do componente *ReviewMateDAO* 46
2.9.1 Método ReviewMate 46
2.10 Test Cases do componente *DistanceHelper* 47
2.10.1 Método GetAddressFromCoordinates 47
2.11 Test Cases do componente *PaymentDAO* 48
2.11.1 Método GetInvoiceByID 48
2.11.2 Método MakePayment 49
2.11.3 Método ConfirmPayment 52
2.12 TestCases do componente ChatDAO 54
2.12.1 Método AddMessage 54
2.12.2 Método CreateChat 55
2.12.3 Método GetMessageList 56
2.12.4 Método GetMessageList 57
2.12.5 Método GetChats 58

1. Introdução

1.1. Identificador do documento

TCO08122020V1.4

Descrição do identificador do projeto:
TCO - TestCaseOutline
12-Numero do mês(dois dígitos seguidos dos dois dígitos iniciais)
07-Dia do mês(dois dígitos iniciais)
2020-Ano de criação(quatro dígitos finais)
V1.0-Numero de versão do projeto(dois caracteres sendo “V” o identificador de versão através de sigla seguido do numero de versão “1” e numero de alteração “.4”)

1.2. Âmbito

Este documento encontra-se enquadrado no âmbito do documento Test Design Specification, referido na secção Referências deste documento. Porém este documento apresenta um objetivo um pouco diferente, ou seja, visa a apresentar os dados de acordo com as descrições definidas no âmbito do documento Test Design Specification.

1.3. Referencias

Lista de referências relevantes para este test case outline:

a) Test Design Specification.
b) https://gitlab.estg.ipp.pt/8160297/lds-housem8

1.4 Glossário

JM – Jorge Moreira
MC – Marcelo Carvalho
SC – Samuel Cunha
JC – Jéssica Coelho
MMC – Miguel Martins Costa

2. Detalhes

2.1 Test Cases do componente *UserDAO*

2.1.1 Método UpdatePassword

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	16:16	JC	Pass	N/A
Pré-condições / estado inicial				
Tem de estar instanciado algum tipo de utilizador				
Procedimento / Passos do *user* para esta função				
1. O utilizador regista-se na plataforma. 2. O utilizador faz a alteração da sua palavra passe, introduzindo a antiga palavra passe corretamente. 3. O utilizador verifica se foi retornado True.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Atualizar palavra passe	Objeto de PasswordUpdateModel, Id do utilizador	newPass, Id	True	Pass

Test Case ID	TC2	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
05/01/21	22:22	JC	Pass	N/A
Pré-condições / estado inicial				
Tem de estar instanciado algum tipo de utilizador				
Procedimento / Passos do *user* para esta função				
1. O utilizador regista-se na plataforma. 2. O utilizador faz a alteração da sua palavra passe introduzindo a antiga palavra passe incorretamente. 3. O utilizador verifica se foi retornado False.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**

Atualizar palavra passe	Objeto de PasswordUpdateModel, Id do utilizador	newPass, Id	False	Pass

2.1.2 Método FindUserByEmail

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	18:36	SC	Pass	N/A
Pré-condições / estado inicial				
Tem de estar instanciado algum tipo de utilizador				
Procedimento / Passos do *user* para esta função				
1. O utilizador regista-se na plataforma. 2. O utilizador verifica se as suas informações retornadas pelo método correspondem ao pedido 3. Retorna o utilizador pretendido				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Encontrar utilizador	String com um email	email	Utilizador pretendido (True)	Pass

Test Case ID	TC2	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	18:36	SC	Pass	N/A
Pré-condições / estado inicial				
Tem de estar instanciado algum tipo de utilizador				
Procedimento / Passos do *user* para esta função				
1. O utilizador regista-se na plataforma. 2. O utilizador verifica se as suas informações retornadas pelo método correspondem ao pedido 3. Retorna nulo				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Encontrar utilizador	String com um email	email	Nulo	Pass

2.1.3 Método FindById

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	20:52	JM	Pass	N/A
Pré-condições / estado inicial				
Tem de estar instanciado um utilizador. O utilizador instanciado tem de estar criado na base de dados.				
Procedimento / Passos do *user* para esta função				
1. Um utilizador regista-se na plataforma. 2. Esse utilizador tenta procurar outro utilizador específico. 3. A api retorna o utilizador encontrado 4. O Utilizador consegue ver os detalhes da sua procura.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Encontrar utilizador	Id de um User	id	User	Pass

Test Case ID	TC2	**Dependências**	---	
Classes de Equivalência Válidas	EC3			
Classes de Equivalência Inválidas	EC4			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	20:55	JM	Pass	N/A
Pré-condições / estado inicial				
Tem de estar instanciado um utilizador. O utilizador instanciado tem de estar criado na base de dados.				
Procedimento / Passos do *user* para esta função				
5. Um utilizador regista-se na plataforma. 6. Esse utilizador tenta procurar outro utilizador específico. 7. A api retorna uma resposta de utilizador não encontrado.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**

Encontrar utilizador	Id de um User	Id	Null	Pass

2.2 Test Cases do componente *EmployerDAO*

2.2.1 Método Create

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	21:09	JM	Pass	N/A
Pré-condições / estado inicial				
Tem de ser possível qualquer utilizador não registado utilizar a rota que usa este método				
Procedimento / Passos do *user* para esta função				
1. O utilizador regista os seus dados 2. O registo e feito na área de employer 3. A api executa a rota com o método Create 4. O utilizador employer é registado no sistema				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Criar um *Employer*	Objeto do tipo Employer	employer	Employer criado na BD, Employer criado é retornado	Pass

Test Case ID	TC2	**Dependências**	---	
Classes de Equivalência Válidas	EC3			
Classes de Equivalência Inválidas	EC4			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	21:09	JM	Pass	N/A
Pré-condições / estado inicial				
Tem de ser possível qualquer utilizador não registado utilizar a rota que usa este método				
Procedimento / Passos do *user* para esta função				
1. O utilizador regista os seus dados exceto a descrição 2. O registo e feito na área de employer 3. A api executa a rota com o método Create O utilizador employer é registado no sistema				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Criar um *Employer*	Objeto do tipo Employer	Employer sem descrição	Pass	N/A

2.2.2 Método Update

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	16:25	JC	Pass	N/A
Pré-condições / estado inicial				
Tem de estar instanciado um Employer				
Procedimento / Passos do *user* para esta função				
1. O Employer regista-se na plataforma. 2. O Employer adiciona as atualizações dos seus dados pessoais. 3. O Employer verifica se os dados devolvidos pelo método são iguais àqueles que inseriu.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Atualizar Employer	Objeto de Employer, Id do Employer	returned, Id	True	Pass

2.2.3 Método FavoriteList

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	16:30	JC	Pass	N/A
Pré-condições / estado inicial				
Têm de estar instanciados um Employer e um Mate				
Procedimento / Passos do *user* para esta função				
1. O Employer regista-se na plataforma. 2. O Mate regista-se na plataforma 3. O Employer adiciona o Mate na sua lista de favoritos 4. O utilizador verifica se consegue aceder à sua lista de Mates favoritos				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Visualizar Lista de Mates Favoritos	Id do Employer	Id	True	Pass

2.2.4 Método AddFavorite

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	16:37	JC	Pass	N/A
Pré-condições / estado inicial				
Têm de estar instanciados um Employer e um Mate				
Procedimento / Passos do *user* para esta função				
1. O Employer regista-se na plataforma. 2. O Mate regista-se na plataforma 3. O Employer adiciona o Mate na sua lista de favoritos 4. O utilizador verifica se o Mate que adicionou se encontra na sua lista de Mates favoritos				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Adicionar Mate à lista de favoritos	Id do Employer, Id do Mate	Id, Id	True	Pass

2.2.5 Método RemoveFavorite

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	16:39	JC	Pass	N/A
Pré-condições / estado inicial				
Têm de estar instanciados um Employer e um Mate				
Procedimento / Passos do *user* para esta função				
1. O Employer regista-se na plataforma. 2. O Mate regista-se na plataforma 3. O Employer adiciona o Mate na sua lista de favoritos 4. O Employer remove esse mesmo Mate da sua lista de favoritos 5. O utilizador verifica se o Mate que removeu não se encontra na sua lista de Mates favoritos				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Remover Mate da lista de favoritos	Id do Employer, Id do Mate	Id, Id	True	Pass

2.2.6 Método FindEmployerById

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	16:41	JC	Pass	N/A
Pré-condições / estado inicial				
Tem de estar instanciado um Employer				
Procedimento / Passos do *user* para esta função				
1. O Employer regista-se na plataforma, adicionando uma descrição no registo. 2. O Employer verifica se as suas informações retornadas pelo método correspondem com aquelas que inseriu ao fazer o registo				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Obter Employer através do ID	Id do Employer	Id	True	Pass

Test Case ID	TC2	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	16:44	JC	Pass	N/A
Pré-condições / estado inicial				
Tem de estar instanciado um Employer				
Procedimento / Passos do *user* para esta função				
1. O Employer regista-se na plataforma, sem adicionar uma descrição no registo. 2. O Employer verifica se as suas informações retornadas pelo método correspondem com aquelas que inseriu ao fazer o registo				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Obter Employer através do ID	Id do Employer	Id	True	Pass

2.3 Test Cases do componente *MateDAO*

2.3.1 Método Create

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	21:20	JM	Pass	N/A
Pré-condições / estado inicial				
Tem de ser possível qualquer utilizador não registado utilizar a rota que usa este método				
Procedimento / Passos do *user* para esta função				
1. O utilizador regista os seus dados 2. O registo e feito na área de mates 3. A api executa a rota com o método Create 4. O utilizador mate é registado no sistema				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Criar um Mate	Objeto do tipo Mate	mate	Mate criado na BD, mate criado é retornado	Pass

Test Case ID	TC2	**Dependências**	---	
Classes de Equivalência Válidas	EC3			
Classes de Equivalência Inválidas	EC4			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	21:20	JM	Pass	N/A
Pré-condições / estado inicial				
Tem de ser possível qualquer utilizador não registado utilizar a rota que usa este método				
Procedimento / Passos do *user* para esta função				
1. O utilizador regista os seus dados exceto a descrição 2. O registo e feito na área de mates 3. A api executa a rota com o método Create 4. O utilizador mate é registado no sistema				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Criar um Mate	Objeto do tipo Mate	Mate sem descrição	Pass	N/A

2.3.2 Método CategoriesList

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	16:47	JC	Pass	N/A
Pré-condições / estado inicial				
Tem de estar instanciado um Mate				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma 2. O Mate verifica se as categorias que inseriu durante o registo se encontram na sua lista de categorias de trabalho				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Listar categorias	Id do Mate	Id	True	Pass

2.3.3 Método AddCategory

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	16:53	JC	Pass	N/A
Pré-condições / estado inicial				
Têm de estar instanciados um Employer e um Mate				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma 2. O Mate adiciona mais categorias à sua lista de categorias de trabalho 3. O Mate verifica se as categorias que adicionou se encontram na sua lista de categorias de trabalho				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Adicionar categorias à lista de categorias	Id do Mate, Array de objetos de Categories Model	Id, categories	True	Pass

2.3.4 Método RemoveCategory

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	16:58	JC	Pass	N/A
Pré-condições / estado inicial				
Tem de estar instanciado um Mate				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma 2. O Mate remove uma das categorias que adicionou no seu registo da sua lista de categorias de trabalho 3. O Mate verifica se a categoria que removeu já não se encontra na sua lista de categorias de trabalho				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Remover categoria à lista de categorias	Id do Mate, Objeto de Categories Model	Id, category	True	Pass

2.3.5 Método Update

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	17:01	JC	Pass	N/A
Pré-condições / estado inicial				
Tem de estar instanciado um Mate				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma. 2. O Mate adiciona as atualizações dos seus dados pessoais. 3. O Mate verifica se os dados devolvidos pelo método são iguais àqueles que inseriu.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Atualizar Mate	Objeto de Mate, Id do Mate	returned, Id	True	Pass

2.3.6 Método FindMateById

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	17:05	JC	Pass	N/A
Pré-condições / estado inicial				
Tem de estar instanciado um Mate				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma, adicionando uma descrição no registo. 2. O Mate verifica se as suas informações retornadas pelo método correspondem com aquelas que inseriu ao fazer o registo				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Obter Mate através do ID	Id do Mate	Id	True	Pass

Test Case ID	TC2	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	17:06	JC	Pass	N/A
Pré-condições / estado inicial				
Tem de estar instanciado um Mate				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma, sem adicionar uma descrição no registo. 2. O Mate verifica se as suas informações retornadas pelo método correspondem com aquelas que inseriu ao fazer o registo				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Obter Mate através do ID	Id do Mate	Id	True	Pass

2.3.7 Método PendingJobsList

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	17:08	JC	Pass	N/A
Pré-condições / estado inicial				
Têm de estar instanciados um Mate e um Employer				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma 2. O Employer regista-se na plataforma 3. O Employer cria uma publicação de trabalho 4. O Employer atribui um Mate e a data de trabalho à publicação 5. O Mate verifica se esse trabalho se encontra na sua lista de trabalhos pendentes, comparando o título do trabalho ao qual foi atribuído com o título recebido no método				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Listar trabalhos pendentes	Id do Mate	Id	True	Pass

2.3.8 Método GetMates

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	20:26	JM	Pass	N/A
Pré-condições / estado inicial				
Existir mates registados na base de dados				
Procedimento / Passos do *user* para esta função				
1. O Employer Seleciona a pesquisa de Mates 2. Preenche todos os filtros 3. A api devolve uma lista de mates consoante os filtros 4. A lista de mates é apresentada				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Obter lista de *Mates* usando filtros. Pesquisar por mates sem autenticação	Categorias, Address, Rank, Distancia , Rating	Categorias, Address, Rank, int Distancia , int Rating	Lista de Mates	Pass

Test Case ID	TC2	**Dependências**	---	
Classes de Equivalência Válidas	EC3			
Classes de Equivalência Inválidas	EC4			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	20:26	JM	Pass	N/A
Pré-condições / estado inicial				
Existir mates registados na base de dados				
Procedimento / Passos do *user* para esta função				
1. O Employer Seleciona a pesquisa de Mates 2. Preenche todos os filtros de categorias, address e rank 3. A api devolve uma lista de mates consoante os filtros 4. A lista de mates é apresentada				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Obter lista de *Mates* usando	Categorias, Address, Rank,	Categorias, Address, Rank	Lista de Mates	Pass

filtros. Pesquisar por mates sem autenticação				

Test Case ID	TC3	**Dependências**	---	
Classes de Equivalência Válidas	EC5			
Classes de Equivalência Inválidas	EC6			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	20:26	JM	Pass	N/A
Pré-condições / estado inicial				
Existir mates registados na base de dados				
Procedimento / Passos do *user* para esta função				
1. O Employer Seleciona a pesquisa de Mates 2. Preenche todos os filtros de categorias, address, rank e distancia 3. A api devolve uma lista de mates consoante os filtros 4. A lista de mates é apresentada				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Obter lista de *Mates* usando filtros. Pesquisar por mates sem autenticação	Categorias, Address, Rank, distancia	Categorias, Address, Rank, distancia	Lista de Mates	Pass

Test Case ID	TC4	**Dependências**	---	
Classes de Equivalência Válidas	EC7			
Classes de Equivalência Inválidas	EC8			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	20:26	JM	Pass	N/A
Pré-condições / estado inicial				
Existir mates registados na base de dados				
Procedimento / Passos do *user* para esta função				
1. O Employer Seleciona a pesquisa de Mates 2. Preenche as categorias com uma só e o Address 3. A api devolve uma lista de mates consoante os filtros 4. A lista de mates é apresentada				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Obter lista de *Mates* usando filtros. Pesquisar por mates sem autenticação	Categorias, Address	Categorias, Address	Lista de Mates	Pass

Test Case ID	TC5	**Dependências**	---	
Classes de Equivalência Válidas	EC9			
Classes de Equivalência Inválidas	EC10			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	20:29	JM	Pass	N/A
Pré-condições / estado inicial				
Existir mates registados na base de dados				
Procedimento / Passos do *user* para esta função				
1. O Employer Seleciona a pesquisa de Mates 2. Preenche as categorias com várias, e o Address 3. A api devolve uma lista de mates consoante os filtros 4. A lista de mates é apresentada				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Obter lista de *Mates* usando filtros. Pesquisar por	Categorias, Address	Categorias, Address	Lista de Mates	Pass

mates sem autenticação				

Test Case ID	TC6	**Dependências**	---	
Classes de Equivalência Válidas	EC11			
Classes de Equivalência Inválidas	EC12			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	20:29	JM	Pass	N/A
Pré-condições / estado inicial				
Existir mates registados na base de dados				
Procedimento / Passos do *user* para esta função				
1. O Employer Seleciona a pesquisa de Mates 2. Preenche os filtros todos menos o Address 3. A api devolve uma lista de mates consoante os filtros 4. A lista de mates é apresentada				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Obter lista de *Mates* usando filtros. Pesquisar por mates sem autenticação	Categorias, Rank, distancia, Rating	Categorias, Rank, distancia, Rating	Lista de Mates	Pass

2.3.9 Método IgnoreJobPost

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	20:06	MMC	Pass	N/A
Pré-condições / estado inicial				
Têm de estar instanciados um Mate, um Employer e um Job				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma 2. O Employer regista-se na plataforma 3. O Employer cria uma publicação de trabalho 4. O Mate ignora o trabalho. 5. Pesquisa por trabalho				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Ignorar Trabalho	Id do Mate, Id do Job	Id do Mate, Id do Job	True e trabalho não existe mais no seu feed	Pass

Test Case ID	TC2	**Dependências**	---	
Classes de Equivalência Válidas	EC3			
Classes de Equivalência Inválidas	EC4			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	20:09	MMC	Pass	N/A
Pré-condições / estado inicial				
Têm de estar instanciados um Mate, um Employer e um Job				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma 2. O Employer regista-se na plataforma 3. O Employer cria uma publicação de trabalho 4. O Mate ignora o trabalho que não existe.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Ignorar Trabalho	Id do Mate, Id do Job	Id do Mate, Id do Job	False	Pass

2.4 Test Cases do componente *JobDAO*

2.4.1 Método AddPayment

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	17:13	JC	Pass	N/A
Pré-condições / estado inicial				
Tem de estar instanciado um Employer				
Procedimento / Passos do *user* para esta função				
1. O Employer regista-se na plataforma 2. O Employer cria uma publicação de trabalho 3. O Employer adiciona um método de pagamento à publicação de trabalho 4. O Employer verifica se o método de pagamento que adicionou está presente na publicação de trabalho				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Adicionar Métodos de Pagamento	Id do JobPost, Array de Objetos de PaymentModel	Id, payments	True	Pass

2.4.2 Método RemovePayment

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	17:16	JC	Pass	N/A
Pré-condições / estado inicial				
Tem de estar instanciado um Employer				
Procedimento / Passos do *user* para esta função				
1. O Employer regista-se na plataforma 2. O Employer cria uma publicação de trabalho 3. O Employer remove um método de pagamento que inseriu durante a criação da publicação do trabalho 4. O Employer verifica se o método de pagamento que removeu não se encontra na sua publicação de trabalho				

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Remover Método de Pagamento	Id do JobPost, Objeto de PaymentModel	Id, paymentModel	True	Pass

2.4.3 Método GetJobs

Test Case ID	TC1	Dependências	---	
Classes de Equivalência Válidas	EC13			
Classes de Equivalência Inválidas	EC14			
Data de Execução	Hora de Execução	*Tester*	Pass/Fail	Resumo da Falha
08/12/20	20:18	MMC	Pass	N/A
Pré-condições / estado inicial				
Têm de estar instanciados um Mate, um Employer e um Job				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma 2. O Employer regista-se na plataforma 3. O Employer cria uma publicação de trabalho 4. O Mate pesquisa por trabalho.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Procurar Trabalho, Modificar Filtros	Categorias	Categorias	Lista de Jobs	Pass

Test Case ID	TC2	**Dependências**	---	
Classes de Equivalência Válidas	EC3			
Classes de Equivalência Inválidas	EC4			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	20:29	MMC	Pass	N/A
Pré-condições / estado inicial				
Têm de estar instanciados um Mate, um Employer e um Job				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma 2. O Employer regista-se na plataforma 3. O Employer cria uma publicação de trabalho 4. O Mate pesquisa por trabalho.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Procurar Trabalho, Modificar Filtros	Categorias, Endereço	Categorias, Endereço	-	Pass

Test Case ID	TC3	**Dependências**	---	
Classes de Equivalência Válidas	EC15			
Classes de Equivalência Inválidas	EC16			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	20:32	MMC	Pass	N/A
Pré-condições / estado inicial				
Têm de estar instanciados um Mate, um Employer e um Job				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma 2. O Employer regista-se na plataforma 3. O Employer cria uma publicação de trabalho 4. O Mate pesquisa por trabalho.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Procurar Trabalho, Modificar Filtros	Categorias, Distância	Categorias, Distância	Lista de Jobs	Pass

Test Case ID	TC4	**Dependências**	---	
Classes de Equivalência Válidas	EC5			
Classes de Equivalência Inválidas	EC6			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	20:33	MMC	Pass	N/A
Pré-condições / estado inicial				
Têm de estar instanciados um Mate, um Employer e um Job				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma 2. O Employer regista-se na plataforma 3. O Employer cria uma publicação de trabalho 4. O Mate pesquisa por trabalho.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Procurar Trabalho, Modificar Filtros	Categorias, Endereço, Distância	Categorias, Endereço, Distância	Lista de Jobs	Pass

Test Case ID	TC5	**Dependências**	---	
Classes de Equivalência Válidas	EC17			
Classes de Equivalência Inválidas	EC18			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	20:41	MMC	Pass	N/A
Pré-condições / estado inicial				
Têm de estar instanciados um Mate, um Employer e um Job				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma 2. O Employer regista-se na plataforma 3. O Employer cria uma publicação de trabalho 4. O Mate pesquisa por trabalho.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Procurar Trabalho, Modificar Filtros	-	-	Lista de Jobs	Pass

Test Case ID	TC6	**Dependências**	---	
Classes de Equivalência Válidas	EC7			
Classes de Equivalência Inválidas	EC8			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	20:47	MMC	Pass	N/A
Pré-condições / estado inicial				
Têm de estar instanciados um Mate, um Employer e um Job				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma 2. O Employer regista-se na plataforma 3. O Employer cria uma publicação de trabalho 4. O Mate pesquisa por trabalho.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Procurar Trabalho, Modificar Filtros	Categorias, Rating	Categorias, Rating	Lista de Jobs	Pass

Test Case ID	TC7	**Dependências**	---	
Classes de Equivalência Válidas	EC9			
Classes de Equivalência Inválidas	EC10			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	20:48	MMC	Pass	N/A
Pré-condições / estado inicial				
Têm de estar instanciados um Mate, um Employer e um Job				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma 2. O Employer regista-se na plataforma 3. O Employer cria uma publicação de trabalho 4. O Mate pesquisa por trabalho.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Procurar Trabalho, Modificar Filtros	Uma Categoria, Rating	Uma Categoria, Rating	Lista de Jobs	Pass

Test Case ID	TC8	**Dependências**	---	
Classes de Equivalência Válidas	EC9			
Classes de Equivalência Inválidas	EC10			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	20:51	MMC	Pass	N/A
Pré-condições / estado inicial				
Têm de estar instanciados um Mate, um Employer e um Job				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma 2. O Employer regista-se na plataforma 3. O Employer cria uma publicação de trabalho 4. O Mate pesquisa por trabalho.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Procurar Trabalho, Modificar Filtros	Uma Categoria, Rating Mais Alto que o do Cliente	Uma Categoria, Rating Mais Alto que o do Cliente	-	Pass

Test Case ID	TC9	**Dependências**	---	
Classes de Equivalência Válidas	EC11			
Classes de Equivalência Inválidas	EC12			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	20:51	MMC	Pass	N/A
Pré-condições / estado inicial				
Têm de estar instanciados um Mate, um Employer e um Job				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma 2. O Employer regista-se na plataforma 3. O Employer cria uma publicação de trabalho 4. O Mate pesquisa por trabalho.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Procurar Trabalho, Modificar Filtros	Duas Categorias	Duas Categorias	Lista de Jobs	Pass

Test Case ID	TC10	**Dependências**	---	
Classes de Equivalência Válidas	EC19			
Classes de Equivalência Inválidas	EC20			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	20:51	MMC	Pass	N/A
Pré-condições / estado inicial				
Têm de estar instanciados um Mate, um Employer e um Job				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma 2. O Employer regista-se na plataforma 3. O Employer cria uma publicação de trabalho 4. O Mate pesquisa por trabalho.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Procurar Trabalho, Modificar Filtros	Categorias, Morada Inválida	Categorias, Morada Inválida	Lista Vazia	Pass

2.4.4 Método MakeOfferOnJob

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	21:08	MMC	Pass	N/A
Pré-condições / estado inicial				
Têm de estar instanciados um Mate, um Employer e um Job				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma 2. O Employer regista-se na plataforma 3. O Employer cria uma publicação de trabalho 4. O Mate realiza oferta de preço ao trabalho.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Realizar oferta de preço ao Trabalho	Oferta, ID do Mate	Oferta, ID do Mate	Oferta	Pass

Test Case ID	TC2	**Dependências**	---	
Classes de Equivalência Válidas	EC3			
Classes de Equivalência Inválidas	EC4			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	21:09	MMC	Pass	N/A
Pré-condições / estado inicial				
Têm de estar instanciados um Mate, um Employer e um Job				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma 2. O Employer regista-se na plataforma 3. O Employer cria uma publicação de trabalho 4. O Mate realiza oferta de preço ao trabalho.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Realizar oferta de preço ao Trabalho	Oferta sem preço, ID do Mate	Oferta sem preço, ID do Mate	Oferta	Pass

2.4.5 Método Create

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	22:09	JM	Pass	N/A
Pré-condições / estado inicial				
O utilizar tem de ter sessão iniciada O utilizador tem de ser um Employer				
Procedimento / Passos do *user* para esta função				
1. O Employer preenche o formulário de criar um post 2. Este não faz o upload de imagens 3. A api guarda na base de dados um post sem imagens 4. O Employer vê o post criado.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
criar um *JobPost*	JobPost	JobPost	JobPost, Base de dados com o Jobpost	Pass

Test Case ID	TC2	**Dependências**	---	
Classes de Equivalência Válidas	EC3			
Classes de Equivalência Inválidas	EC4			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	22:09	JM	Pass	N/A
Pré-condições / estado inicial				
O utilizar tem de ter sessão iniciada O utilizador tem de ser um Employer				
Procedimento / Passos do *user* para esta função				
1. O Employer preenche o formulário de criar um post 2. Este faz o upload de imagens 3. A api guarda na base de dados um post com o diretório de cada imagem na api 4. O Employer vê o post criado com imagens.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
criar um *JobPost*	JobPost	JobPost	JobPost, Base de dados com o Jobpost	Pass

2.4.6 Método Delete

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	22:19	JM	Pass	N/A
Pré-condições / estado inicial				
O utilizar tem de ter sessão iniciada O utilizador tem de ser um Employer				
Procedimento / Passos do *user* para esta função				
1. O Employer seleciona um post par apagar 2. O Employer apaga o post 3. A api usa o método de Delete 4. O método delete retorna true e o post é apagado 5. O Employer deixa de ver o post				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Apagar um *JobPost*	Id do post	id	true	Pass

2.4.7 Método FindById

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	22:19	JM	Pass	N/A
Pré-condições / estado inicial				
O utilizar tem de ter sessão iniciada				
Procedimento / Passos do *user* para esta função				
1. O Utilizador Procura um JobPost 2. O JobPost é selecionado 3. A api usa o método FindByID 4. Os detalhes do JobPost são retornados				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Encontrar um *JobPost por Id*	Id do post	id	JobPost	Pass

2.4.8 Método GetEmployerPosts

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	22:29	JM	Pass	N/A
Pré-condições / estado inicial				
O utilizar tem de ter sessão iniciada O Utilizador é um Employer				
Procedimento / Passos do *user* para esta função				
1. O Employer abre a lista de JobPosts que tem 2. A api usa o método GetEmployerPosts com o Id do Employer 3. É enviada a lista de JobPosts para o Employer 4. O Employer vê a Lista de JobPosts que possui				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Encontrar JobPosts de um Employer	Id do Employer	id	Lista de JobPost	Pass

2.4.9 Método UpdatePostDetails

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	22:29	JM	Pass	N/A
Pré-condições / estado inicial				
O utilizar tem de ter sessão iniciada O Utilizador é um Employer				
Procedimento / Passos do *user* para esta função				
1. O Employer abre a lista de JobPosts que tem 2. Seleciona um JobPost para editar 3. Após editar, guarda os dados 4. A api usa o método UpdatePostDetails 5. O Post é atualizado na base de dados 6. O Utilizador ve o Post com os dados novos				

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Update Jobposts	JobPost	JobPost	JobPost, e JobPost atualizado na Base de dados	Pass

2.5 Test Cases do componente *WorkDAO*

2.5.1 Método MarkJobAsDone

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	21:10	MMC	Pass	N/A
Pré-condições / estado inicial				
Têm de estar instanciados um Mate, um Employer e um Job				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma 2. O Employer regista-se na plataforma 3. O Employer cria uma publicação de trabalho 4. O Mate realizar oferta ao trabalho. 5. O Employer aceita a oferta e marca o trabalho. 6. O Employer marca o trabalho como concluído.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Marcar trabalho como concluído	JobID, UserID	JobID, UserID	True	Pass

Test Case ID	TC2	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	21:11	MMC	Pass	N/A
Pré-condições / estado inicial				
Têm de estar instanciados um Mate, um Employer e um Job				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma 2. O Employer regista-se na plataforma 3. O Employer cria uma publicação de trabalho 4. O Mate realiza oferta ao trabalho. 5. O Employer aceita a oferta e marca o trabalho. 6. O Mate marca o trabalho como concluído.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**

Marcar trabalho como concluído	JobID, UserID	JobID, UserID	True	Pass

Test Case ID	TC3	**Dependências**	---	
Classes de Equivalência Válidas	EC3			
Classes de Equivalência Inválidas	EC4			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	21:48	MMC	Pass	N/A
Pré-condições / estado inicial				
Têm de estar instanciados um Mate, um Employer e um Job				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma 2. O Employer regista-se na plataforma 3. O Employer cria uma publicação de trabalho 4. O Mate realiza oferta ao trabalho. 5. O Employer aceita a oferta e marcar o trabalho. 6. O Mate ou Employer marca o trabalho como concluído.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Marcar trabalho como concluído	JobID inválido, UserID	JobID, UserID	Exception	Pass

Test Case ID	TC4	**Dependências**	---	
Classes de Equivalência Válidas	EC5			
Classes de Equivalência Inválidas	EC6			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	21:51	MMC	Pass	N/A
Pré-condições / estado inicial				
Têm de estar instanciados um Mate, um Employer e um Job				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma 2. O Employer regista-se na plataforma 3. O Employer cria uma publicação de trabalho 4. O Mate realiza oferta ao trabalho. 5. O Employer aceita a oferta e marcar o trabalho. 6. O Mate ou Employer marca o trabalho como concluído.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Marcar trabalho como concluído	JobID, UserID inválido	JobID, UserID inválido	Exception	Pass

2.5.2 Método Create

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	23:00	JM	Pass	N/A
Pré-condições / estado inicial				
O Utilizador tem de estar registado e com sessão iniciada O utilizador é um Employer				
Procedimento / Passos do *user* para esta função				
1. O Employer aceita que o trabalho seja feito pelo Mate 2. O Employer associa o trabalho ao mate 3. A api usa o método Create para criar um Job com o Mate, Employer e JobPost associados				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Criar trabalhos com mate, post e employer associados	Job, Id	Job, Id	Job, Job adicionado a base de dados, e JobPost marcado como Done	Pass

Test Case ID	TC2	**Dependências**	---	
Classes de Equivalência Válidas	EC3			
Classes de Equivalência Inválidas	EC4			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	23:10	JM	Pass	N/A
Pré-condições / estado inicial				
O Utilizador tem de estar registado e com sessão iniciada O utilizador é um Employer				
Procedimento / Passos do *user* para esta função				
1. O Employer tenta adicionar um mate que já não existe a um Job 2. A api usa o método Create 3. É retornada uma mensagem que informa que o Mate não existe				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Criar trabalhos com mate, post e employer associados	Job, Id	Job, Id	Mate não existe	Pass

Test Case ID	TC3	**Dependências**	---	
Classes de Equivalência Válidas	EC5			
Classes de Equivalência Inválidas	EC6			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	23:20	JM	Pass	N/A
Pré-condições / estado inicial				
O Utilizador tem de estar registado e com sessão iniciada O utilizador é um Employer				
Procedimento / Passos do *user* para esta função				
1. O Employer tenta adicionar JobPost que já não existe a um Job 2. A api usa o método Create 3. É retornada uma mensagem que informa que o JobPost não existe				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Criar trabalhos com mate, post e employer associados	Job, Id	Job, Id	JobPost não existe	Pass

2.5.3 Método FindById

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	23:30	JM	Pass	N/A
Pré-condições / estado inicial				
O Utilizador tem de estar registado e com sessão iniciada				
Procedimento / Passos do *user* para esta função				
1. O utilizador tenta aceder a um Job 2. A api executa o método FindById 3. A informação do Job é retornada ao utilizador				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Encontrar trabalhos por Id	Id de Job	Id	Job	

Test Case ID	TC2	**Dependências**	---	
Classes de Equivalência Válidas	EC3			
Classes de Equivalência Inválidas	EC4			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	23:30	JM	Pass	N/A
Pré-condições / estado inicial				
O Utilizador tem de estar registado e com sessão iniciada				
Procedimento / Passos do *user* para esta função				
1. O utilizador tenta aceder a um Job 2. A api executa o método FindById 3. O método retorna null 4. É enviada uma mensagem de Job não existente				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Encontrar trabalhos por Id	Id de Job	Id	Null / Job não existe	Pass

2.6 Test Cases do componente *LoginDAO*

2.6.1 Método Authenticate

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	18:53	SC	Pass	N/A
Pré-condições / estado inicial				
Tem de estar instanciado um employer				
Procedimento / Passos do *user* para esta função				
1. O Employer regista-se na plataforma 2. O Employer tenta fazer login na aplicação 3. O Employer entra na aplicação				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Autenticar utilizador	String com email, string com password	Email, password	True	Pass

Test Case ID	TC2	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	18:55	SC	Pass	N/A
Pré-condições / estado inicial				
Tem de estar instanciado um employer				
Procedimento / Passos do *user* para esta função				
1. O Employer regista-se na plataforma 2. O Employer tenta fazer login na aplicação 3. O Employer não consegue entrar na aplicação				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Autenticar utilizador	String com email, string com password	Email, password	False	Pass

Test Case ID	TC3	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	18:57	SC	Pass	N/A
Pré-condições / estado inicial				
Tem de estar instanciado um mate				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma 2. O Mate tenta fazer login na aplicação 3. O Mate entra na aplicação				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Autenticar utilizador	String com email, string com password	Email, password	True	Pass

Test Case ID	TC4	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	18:59	SC	Pass	N/A
Pré-condições / estado inicial				
Tem de estar instanciado um mate				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma 2. O Mate tenta fazer login na aplicação 3. O Mate não consegue entrar na aplicação				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Autenticar utilizador	String com email, string com password	Email, password	False	Pass

Test Case ID	TC5	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	18:59	SC	Pass	N/A
Pré-condições / estado inicial				
Tem de estar instanciado algum tipo de utilizador				
Procedimento / Passos do *user* para esta função				
1. O utilizador regista-se na plataforma. 2. O utilizador verifica se as suas informações retornadas pelo método correspondem ao pedido 3. Retorna o utilizador pretendido				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Encontrar utilizador	String com um email	email	Utilizador pretendido (True)	Pass

Test Case ID	TC6	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	19:00	SC	Pass	N/A
Pré-condições / estado inicial				
Tem de estar instanciado algum tipo de utilizador				
Procedimento / Passos do *user* para esta função				
1. O utilizador regista-se na plataforma. 2. O utilizador verifica se as suas informações retornadas pelo método correspondem ao pedido 3. Retorna nulo				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Encontrar utilizador	String com um email	email	Nulo	Pass

2.7 Test Cases do componente *ReportDAO*

2.7.1 Método ReportUser

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	19:05	SC	Pass	N/A
Pré-condições / estado inicial				
Tem de estar instanciado dois utilizadores				
Procedimento / Passos do *user* para esta função				
1. O utilizador regista-se na plataforma. 2. O utilizador faz login na aplicação. 3. Escolhe o utilizador que pretende reportar. 4. Preenche os dados do *report*. 5. Envia o report.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Reportar utilizador	Dois inteiros com o id do utilizador que irá reportar e do utilizador a ser reportado, Objeto de Report	Id, user, model	True	Pass

2.8 Test Cases do componente *ReviewEmployerDAO*

2.8.1 Método ReviewEmployer

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	19:10	SC	Pass	N/A
Pré-condições / estado inicial				
Tem de estar instanciado pelo menos um mate e um employer				
Procedimento / Passos do *user* para esta função				
1. O mate regista-se na plataforma. 2. O mate faz login na aplicação. 3. Escolhe o employer que pretende dar uma pontuação. 4. Preenche os dados do Review. 5. Envia o Review.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Avaliar employer	Um inteiro com o id do employer, objeto de Review	employer, model	True	Pass

2.9 Test Cases do componente *ReviewMateDAO*

2.9.1 Método ReviewMate

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	19:15	SC	Pass	N/A
Pré-condições / estado inicial				
Tem de estar instanciado pelo menos um mate e um employer				
Procedimento / Passos do *user* para esta função				
1. O employer regista-se na plataforma. 2. O employer faz login na aplicação. 3. Escolhe o mate que pretende fazer review. 4. Preenche os dados do MateReview. 5. Envia o MateReview.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Avaliar mate	Um inteiro com o id do mate, objeto de MateReview	mate, model	True	Pass

2.10 Test Cases do componente *DistanceHelper*

2.10.1 Método GetAddressFromCoordinates

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	22:00	MMC	Pass	N/A
Pré-condições / estado inicial				
-				
Procedimento / Passos do *user* para esta função				
1. O User utiliza o GPS para se localizar.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Utilização da Localização	Latitude e Longitude	Latitude e Longitude	Endereço	Pass

Test Case ID	TC2	**Dependências**	---	
Classes de Equivalência Válidas	EC3			
Classes de Equivalência Inválidas	EC4			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	22:05	MMC	Pass	N/A
Pré-condições / estado inicial				
-				
Procedimento / Passos do *user* para esta função				
1. O user utiliza o GPS para se localizar.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Utilização da Localização	Latitude e Longitude inválidos	-1000000, 1000000	Exception	Pass

2.11 Test Cases do componente *PaymentDAO*

2.11.1 Método GetInvoiceByID

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	22:20	MMC	Pass	N/A
Pré-condições / estado inicial				
Existir um trabalho já realizado.				
Procedimento / Passos do *user* para esta função				
1. O User tenta aceder a uma fatura de um trabalho				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Efetuar o Pagamento	UserID, JobID	UserID, JobID	Invoice	Pass

Test Case ID	TC2	**Dependências**	---	
Classes de Equivalência Válidas	EC3			
Classes de Equivalência Inválidas	EC4			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	22:25	MMC	Pass	N/A
Pré-condições / estado inicial				
Existir um trabalho já realizado.				
Procedimento / Passos do *user* para esta função				
1. O User tenta aceder a uma fatura de um trabalho				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Efetuar o Pagamento	UserID, JobID inválido	UserID, JobID	Exception	Pass

Test Case ID	TC3	**Dependências**	---	
Classes de Equivalência Válidas	EC5			
Classes de Equivalência Inválidas	EC6			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	22:30	MMC	Pass	N/A
Pré-condições / estado inicial				
Existir um trabalho já realizado.				
Procedimento / Passos do *user* para esta função				
1. O User tenta aceder a uma fatura de um trabalho				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Efetuar o Pagamento	UserID inválido, JobID	UserID, JobID	Exception	Pass

2.11.2 Método MakePayment

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	22:32	MMC	Pass	N/A
Pré-condições / estado inicial				
-				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma 2. O Employer regista-se na plataforma 3. O Employer cria uma publicação de trabalho 4. O Mate realiza oferta ao trabalho. 5. O Employer aceita a oferta e marca o trabalho. 6. O Mate marca o trabalho como concluído. 7. O Employer marca o trabalho como concluído. 8. O Employer realiza o pagamento.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Efetuar o Pagamento	Invoice, JobID, EmployerID	Invoice, JobID, EmployerID	Invoice	Pass

Test Case ID	TC2	**Dependências**	---	
Classes de Equivalência Válidas	EC3			
Classes de Equivalência Inválidas	EC4			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	22:46	MMC	Pass	N/A
Pré-condições / estado inicial				
-				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma 2. O Employer regista-se na plataforma 3. O Employer cria uma publicação de trabalho 4. O Mate realiza oferta ao trabalho. 5. O Employer aceita a oferta e marca o trabalho. 6. O Mate marca o trabalho como concluído. 7. O Employer marca o trabalho como concluído. 8. O Employer realiza o pagamento.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Efetuar o Pagamento	Invoice, JobID, EmployerID inválido	Invoice, JobID, EmployerID inválido	Exception	Pass

Test Case ID	TC3	**Dependências**	---	
Classes de Equivalência Válidas	EC5			
Classes de Equivalência Inválidas	EC6			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	22:48	MMC	Pass	N/A
Pré-condições / estado inicial				
-				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma 2. O Employer regista-se na plataforma 3. O Employer cria uma publicação de trabalho 4. O Mate realiza oferta ao trabalho. 5. O Employer aceita a oferta e marca o trabalho. 6. O Mate marca o trabalho como concluído. 7. O Employer marca o trabalho como concluído. 8. O Employer realiza o pagamento.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**

Efetuar o Pagamento	Invoice, JobID inválido, EmployerID	Invoice, JobID inválido, EmployerID	Exception	Pass

2.11.3 Método ConfirmPayment

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	22:56	MMC	Pass	N/A
Pré-condições / estado inicial				
-				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma 2. O Employer regista-se na plataforma 3. O Employer cria uma publicação de trabalho 4. O Mate realiza oferta ao trabalho. 5. O Employer aceita a oferta e marca o trabalho. 6. O Mate marca o trabalho como concluído. 7. O Employer marca o trabalho como concluído. 8. O Employer realiza o pagamento. 9. O Mate confirma o pagamento.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Efetuar o Pagamento	JobID, MateID	JobID, MateID	True	Pass

Test Case ID	TC2	**Dependências**	---	
Classes de Equivalência Válidas	EC3			
Classes de Equivalência Inválidas	EC4			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	22:59	MMC	Pass	N/A
Pré-condições / estado inicial				
-				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma 2. O Employer regista-se na plataforma 3. O Employer cria uma publicação de trabalho 4. O Mate realiza oferta ao trabalho. 5. O Employer aceita a oferta e marca o trabalho. 6. O Mate marca o trabalho como concluído. 7. O Employer marca o trabalho como concluído. 8. O Employer realiza o pagamento. 9. O Mate confirma o pagamento.				

Requisito/Use Case/ Funcionalidade	Inputs	Valores do Inputs	Resultados Esperados	Pass/Fail/Untested
Efetuar o Pagamento	JobID inválido, MateID	JobID inválido, MateID	Exception	Pass

Test Case ID	TC3	Dependências	---	
Classes de Equivalência Válidas	EC5			
Classes de Equivalência Inválidas	EC6			
Data de Execução	Hora de Execução	*Tester*	Pass/Fail	Resumo da Falha
08/12/20	23:00	MMC	Pass	N/A
Pré-condições / estado inicial				
-				
Procedimento / Passos do *user* para esta função				
1. O Mate regista-se na plataforma 2. O Employer regista-se na plataforma 3. O Employer cria uma publicação de trabalho 4. O Mate realiza oferta ao trabalho. 5. O Employer aceita a oferta e marca o trabalho. 6. O Mate marca o trabalho como concluído. 7. O Employer marca o trabalho como concluído. 8. O Employer realiza o pagamento. 9. O Mate confirma o pagamento.				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Efetuar o Pagamento	JobID, MateID inválido	JobID, MateID inválido	Exception	Pass

2.12 TestCases do componente ChatDAO

2.12.1 Método AddMessage

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	16:00	MC	Pass	N/A
Pré-condições / estado inicial				
Tem de estar instanciado um Employer Tem de estar instanciado um Mate Tem de estar instanciado um Chat				
Procedimento / Passos do *user* para esta função				
1. O Employer da Match com o Mate 2. O Employer envia uma mensagem ao Mate 3. O Mate recebe a mensagem				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Utilizar um chat para negociar com um cliente	Id do Chat	Id	True	Pass

2.12.2 Método CreateChat

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	17:00	MC	Pass	N/A
Pré-condições / estado inicial				
Tem de estar instanciado um Employer Tem de estar instanciado um Mate				
Procedimento / Passos do *user* para esta função				
1. O Employer da Match com o Mate 2. O Employer acede a lista de Match e verifica que foi adicionado uma nova conversa				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Utilizar um chat para negociar com um cliente	Chat	model	True	Pass

2.12.3 Método GetMessageList

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	17:00	MC	Pass	N/A
Pré-condições / estado inicial				
Tem de estar instanciado um Employer Tem de estar instanciado um Mate Tem de estar instanciado um Chat				
Procedimento / Passos do *user* para esta função				
1. O Employer da Match com o Mate 2. O Employer acede a lista de Match e verifica que foi adicionado uma nova conversa 3. O Employer acede á conversa e verifica que todas as mensagens da conversa foram carregadas				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Utilizar um chat para negociar com um cliente	Id do Chat	Id	True	Pass

2.12.4 Método GetMessageList

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	17:00	MC	Pass	N/A
Pré-condições / estado inicial				
Tem de estar instanciado um Employer Tem de estar instanciado um Mate Tem de estar instanciado um Chat				
Procedimento / Passos do *user* para esta função				
1. O Employer da Match com o Mate 2. O Employer acede a lista de Match e verifica que foi adicionado uma nova conversa 3. O Employer acede á conversa e verifica que todas as mensagens da conversa foram carregadas				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Utilizar um chat para negociar com um cliente	Id do Chat	Id	True	Pass

2.12.5 Método GetChats

Test Case ID	TC1	**Dependências**	---	
Classes de Equivalência Válidas	EC1			
Classes de Equivalência Inválidas	EC2			
Data de Execução	**Hora de Execução**	***Tester***	**Pass/Fail**	**Resumo da Falha**
08/12/20	18:00	MC	Pass	N/A
Pré-condições / estado inicial				
Tem de estar instanciado um Employer Tem de estar instanciado um Mate Tem de estar instanciado um Chat				
Procedimento / Passos do *user* para esta função				
1. O Employer da Match com o Mate 2. O Employer acede a lista de Match e verifica que foi adicionado uma nova conversa 3. O Employer acede á conversa e verifica que todas as mensagens da conversa foram carregadas				
Requisito/Use Case/ Funcionalidade	**Inputs**	**Valores do Inputs**	**Resultados Esperados**	**Pass/Fail/Untested**
Utilizar um chat para negociar com um cliente	Id do User	Id	True	Pass