

UAT Test Plan for Feira&Office Grupo 2

Índice

1. Scope	3
2. Testing team	4
3. Milestones and deliverables	5
4. Features to be tested per Sprint.....	7

1. Scope

- Funcionalidade de Login (Validação de e-mail)

O objetivo do teste de funcionalidade de Login é descobrir se a funcionalidade de login, tem as devidas validações, nomeadamente, a validação do e-mail.

Exemplo de testes efetuados:

E-mail sem domínio;

E-mail sem @;

E-mail com @ seguido de carater especial;

E-mail com duas @;

E-mail sem parte inicial.

2. Testing team

Name	Responsibilities
David Fernandes	UAT Coordenador
Asaph Alves	Developer & Tester
Tiago Marquez	Developer & Tester
Pedro Valente	Developer & Tester

3. Milestones and deliverables

Resultados:

Relativamente aos e-mails testados na função de Login, registaram todos os erros nos e-mails inválidos e tivemos sucesso nos e-mails corretos.

Design & testing process

Para a utilização do sistema de teste, utilizámos um novo package de nome “testes unitários”, recorrendo à biblioteca Junit, sendo que cada função estará numa classe com o respetivo teste.

Para efetuar o teste de Login, criámos dois métodos com os nomes, testEnderecoDeEmailValido e testEnderecoDeEmailValido o `assertFalse` e o `assertTrue` da referida biblioteca.

Futuramente teremos que efetuar testes unitários na função `adicionarFornecedor`, para validar a inserção de dados e confirmar a gravação dos mesmos na base de dados.

Após esta validação da função `adicionarFornecedor`, teremos seguidamente de estruturar testes na leitura do ficheiro XML.

3.1. UAT Execution

Passos:

- 1) Integração. Cada um dos developers tem as suas tarefas registadas para o respetivo sprint, estando definido que cada um fará testes às suas funções no final de cada sprint.
- 2) Execução de casos de teste. Todos os developers terão que testar as funcionalidades que se encontram a desenvolver, passando numa fase final os testes unitários a outros developers.
- 3) Uma vez terminados, todos os testes serão registados em relatório próprio, podendo no futuro serem acrescentado novos testes caso surjam bugs.

3.2. Reporting & data analysis

Dificuldades:

1. **Compreensão dos Requisitos:** Um dos desafios é garantir que todos os membros da equipa tenham uma compreensão completa dos requisitos para criar operadores, fornecedores e o processo de login.
2. **Isolamento de Testes Unitários:** Garantir que os testes sejam verdadeiramente unitários, ou seja, testando uma única função isoladamente, e não dependerem de outros módulos ou sistemas externos pode ser desafiador.
3. **Configuração Inicial:** Configurar o ambiente de teste pode ser uma dificuldade inicial. Isso inclui a configuração da base de dados, a criação de objetos de teste e a preparação de cenários de teste.

Feedback Geral:

1. **Organização:** Certificação de que os testes sejam fáceis de ler e entender. Utilizar nomes descritivos para os casos de teste e organizar os mesmo de forma lógica.
2. **Mensagens de Erro:** As mensagens de erro nos testes devem ser informativas e ajudar a identificar a causa de falhas.
3. **Documentação de Testes:** todos os testes devem ser bem documentados, explicando a lógica por trás de cada caso de teste e como se relacionam com os requisitos.

4. Features to be tested per Sprint

Para cada característica testada, é essencial definir claramente os critérios que determinarão se o teste passou ou falhou. Esses critérios ajudarão a equipa a avaliar a funcionalidade testada de forma objetiva.

Feature 1

4.1.1. Pass/fail criteria

Critério de Aprovação: O login é bem-sucedido quando as credenciais corretas são fornecidas, e o utilizador é redirecionado para a página principal.

Critério de Reprovação: Falha no login com credenciais erradas ou problemas técnicos que impeçam o login.

4.1.2. Test cases

- 1) Iniciar aplicação com o IDE IntelliJ.
- 2) Fazer Login com introdução de e-mail e password
- 3) Verificar se o login foi efetuado com sucesso.
- 4) Testar diversos tipos de e-mail e verificar se é validado ou não.