

UAT Test Plan for Feira&Office

Índice

1.	Escopo.....	3
1.1.	Objetivos e requisitos comerciais.....	3
1.2.	Escopo.....	3
2.	Equipa de Testes.....	5
3.	Objetivos e resultados.....	6
3.1	Design e processo de testagem.....	6
3.2	UAT Execution.....	7
3.3	Relatórios e análise de dados.....	7
4	Requisitos de ambiente.....	8
4.1	Requisitos de hardware	8
4.2	Requisitos de software.....	9
5	Funcionalidades a ser testadas por Sprint	10
5.1	Testar adicionar operadores e fornecedores.....	10
5.1.1	Critérios de aprovação/reprovação.....	10
5.1.2	Casos de teste	10
5.2	Testar base de dados	10
5.2.1	Critérios de aprovação/reprovação.....	10
5.2.2	Casos de teste	10
5.3	Testar Login.....	11

1. Escopo

1.1. Objetivos e requisitos comerciais

- OS objetivo deste sprint são:
 - Validar o XSD
 - Validar importação de XML
 - Criar encomenda para aprovação
 - Realizar testes unitários do login
 - Realizar testes unitários de registo de fornecedores e operadores
 - Função editar operadores e fornecedores

1.2. Escopo

Testes Unitários para Ler Fornecedores:

Teste testAdicionarFornecedorBaseDeDados:

Objetivo: Verificar se é possível adicionar corretamente um fornecedor à base de dados.

Escopo:

Cria um fornecedor e tenta adicioná-lo à base de dados.

Verifica se a inserção foi bem-sucedida.

Teste testAdicionarFornecedorBaseDeDados_Falha:

Objetivo: Verificar se o tratamento de falhas ocorre corretamente ao adicionar um fornecedor.

Escopo:

Configura a simulação de uma falha na execução da query na base de dados.

Tenta adicionar um fornecedor e verifica se a inserção falhou.

Teste testAdicionarEExcluirFornecedorBaseDeDados:

Objetivo: Verificar se é possível adicionar e excluir corretamente um fornecedor da base de dados.

Escopo:

Adiciona um fornecedor à base de dados.

Tenta excluir o fornecedor e verifica se a exclusão foi bem-sucedida.

Testes Unitários para Login de Fornecedores:

Teste testVerificarLoginUtilizadorFornecedor:

Objetivo: Verificar se o método de login para um fornecedor funciona corretamente.

Escopo:

Adiciona lógica de simulação para um fornecedor.

Executa o método de verificação de login e verifica se retorna o tipo de utilizador esperado (UtilizadorFornecedor).

Testes Unitários para Login de Administrador:

Teste testVerificarLoginUtilizadorAdm:

Objetivo: Verificar se o método de login para um administrador funciona corretamente.

Escopo:

Adiciona lógica de simulação para um administrador.

Executa o método de verificação de login e verifica se retorna o tipo de utilizador esperado (UtilizadorAdm).

Testes Unitários para Login de Operadores:

Teste testVerificarLoginOperador:

Objetivo: Verificar se o método de login para um operador funciona corretamente.

Escopo:

Adiciona lógica de simulação para um fornecedor.

Executa o método de verificação de login e verifica se retorna o tipo de utilizador esperado (UtilizadorOperador).

Testes Unitários para Ler Operadores:

Teste testAdicionarOperBaseDeDados:

Objetivo: Verificar se é possível adicionar corretamente um operador à base de dados.

Escopo:

Cria um operador e tenta adicioná-lo à base de dados.

Verifica se a inserção foi bem-sucedida.

Teste testAdicionarOperBaseDeDados_Falha:

Objetivo: Verificar se o tratamento de falhas ocorre corretamente ao adicionar um operador.

Escopo:

Configura a simulação de uma falha na execução da query na base de dados.

Tenta adicionar um operador e verifica se a inserção falhou.

Teste testAdicionarEExcluirOperBaseDeDados:

Objetivo: Verificar se é possível adicionar e excluir corretamente um operador da base de dados.

Escopo:

Adiciona um operador à base de dados.

Tenta excluir o operador e verifica se a exclusão foi bem-sucedida.

Testes Unitários Validação de Email:

Teste testEnderecoDeEmailValido

Objetivo: Testa se os endereços de e-mail fornecidos são válidos.

Escopo: Adiciona lógica de simular emails, para testar se são válidos.

Teste testEnderecoDeEmailInvalido

Objetivo: Testa se os endereços de e-mail fornecidos são inválidos.

Escopo: Adiciona lógica de simular emails inválidos, para testar se são inválidos.

2. Equipa de Testes

Esta secção, lista os membros da sua equipa de GQ (Gestão de Qualidade) e quais serão as suas funções durante o UAT por Sprint

Exemplo:

Nome	Responsabilidades
Asaph Alves	Desenvolvedor e tester
David Fernandes	Desenvolvedor e tester
Pedro Valente	Coordenador de UAT
Tiago Marquez	Desenvolvedor e tester

3. Objetivos e resultados

3.1 Design e processo de testagem

LoginTest
- emailadmin : String - passwordadmin : String - emailoperador : String - passwordoperador : String - emailfornecedor : String - passwordfornecedor : String
+ void setUp() + void testVerificarLoginUtilizadorAdm() + void testVerificarLoginUtilizadorFornecedor() + void testVerificarLoginOperador()

LerOperadoresTest
+ email : String + password : String + void setUp() + void tearDown() + void testAdicionarOperBaseDeDados() + void testAdicionarOperBaseDeDados_Falha() + void testAdicionarEExcluirOperBaseDeDados()

LerFornecedoresTest
+ emailfornecedor : String + password : String + void setUp() + void tearDown() + void testAdicionarFornecedorBaseDeDados() + void testAdicionarFornecedorBaseDeDados_Falha() + void testAdicionarEExcluirFornecedorBaseDeDados()

ValidarEmailTest
+ void testEnderecoDeEmailValido() + void testEnderecoEmailInvalido()

3.2 UAT Execution

Tempo de execução do package TestesUnitários é de 1seg e 564ms.

Os testes executados foram:

LerOperadorTest:

- testAdicionarOperBaseDeDados
- testAdicionarOperBaseDeDados_Falha
- testAdicionarEExcluirOperBaseDeDados

LerFornecedorTest:

- testVerificarFornecedorBaseDeDados
- testVerificarFornecedorBaseDeDados_Falha
- testAdicionarEExcluirFornecedorBaseDeDados

LoginTest:

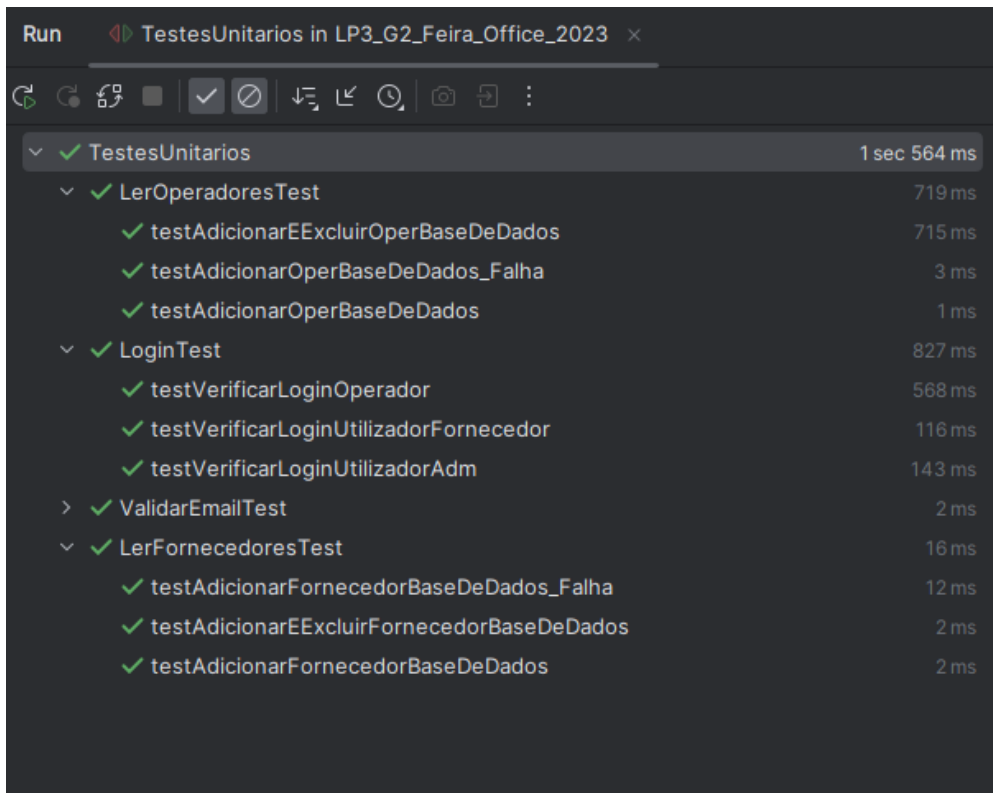
- testVerificarLoginOperador
- testVerificarLoginUtilizadorFornecedor
- testVerificarLoginUtilizadorAdm

ValidarEmailTest:

- testEnderecoDeEmailValido
- testEnderecoDeEmailInvalido

3.3 Relatórios e análise de dados

Em relação aos testes de Login, registo de fornecedores, registo de operadores e validação de emails:



✓ TestesUnitarios	1 sec 564 ms
✓ LerOperadoresTest	719 ms
✓ testAdicionarEExcluirOperBaseDeDados	715 ms
✓ testAdicionarOperBaseDeDados_Falha	3 ms
✓ testAdicionarOperBaseDeDados	1 ms
✓ LoginTest	827 ms
✓ testVerificarLoginOperador	568 ms
✓ testVerificarLoginUtilizadorFornecedor	116 ms
✓ testVerificarLoginUtilizadorAdm	143 ms
> ValidarEmailTest	2 ms
✓ LerFornecedoresTest	16 ms
✓ testAdicionarFornecedorBaseDeDados_Falha	12 ms
✓ testAdicionarEExcluirFornecedorBaseDeDados	2 ms
✓ testAdicionarFornecedorBaseDeDados	2 ms

4 Requisitos de ambiente

4.1 Requisitos de hardware












- Processador: Mínimo de 2,0 GHz ou mais rápido. Recomendado processador dual-core ou superior.
- Memória RAM: Mínimo de 2 GB; Recomendado 4 GB ou mais.
- Espaço em Disco: Mínimo de 1 GB de espaço livre, dependendo dos recursos instalados.
- Velocidade do Disco Rígido: Para melhorar o desempenho, instale em uma unidade de estado sólido (SSD).
- Placa de Vídeo: Suporte para uma resolução mínima de ecrã 720p.

4.2 Requisitos de software

Dependências de software:

```
<dependency>
  <groupId>org.junit.jupiter</groupId>
  <artifactId>junit-jupiter-api</artifactId>
  <version>${junit.version}</version>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>org.junit.jupiter</groupId>
  <artifactId>junit-jupiter-engine</artifactId>
  <version>${junit.version}</version>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <version>RELEASE</version>
  <scope>compile</scope>
</dependency>
```

Jars necessários:

	apiguardian-api-1.1.2.jar	04/12/2023 18:40	Executable Jar File	7 KB
	byte-buddy-1.14.9.jar	04/12/2023 18:40	Executable Jar File	4 113 KB
	byte-buddy-agent-1.14.9.jar	04/12/2023 18:40	Executable Jar File	251 KB
	jakarta.activation-2.0.1.jar	04/12/2023 18:40	Executable Jar File	61 KB
	jakarta.mail-2.0.1.jar	04/12/2023 18:40	Executable Jar File	674 KB
	junit-jupiter-api-5.10.0.jar	04/12/2023 18:40	Executable Jar File	207 KB
	junit-platform-commons-1.10.0.jar	04/12/2023 18:40	Executable Jar File	104 KB
	mockito-core-5.7.0.jar	04/12/2023 18:40	Executable Jar File	684 KB
	mockito-junit-jupiter-5.7.0.jar	04/12/2023 18:40	Executable Jar File	9 KB
	objenesis-3.3.jar	04/12/2023 18:40	Executable Jar File	49 KB
	opentest4j-1.3.0.jar	04/12/2023 18:40	Executable Jar File	14 KB

5 Funcionalidades a ser testadas por Sprint

5.1 Testar adicionar operadores e fornecedores

5.1.1 Critérios de aprovação/reprovação

Aprovação: O sistema deve permitir a adição bem-sucedida de operadores e fornecedores à base de dados.

Reprovação: A adição falha ou não atende aos requisitos especificados.

5.1.2 Casos de teste

Adicionar um operador à base de dados.

Adicionar um fornecedor à base de dados.

Verificar se os dados inseridos estão corretos e refletidos na base de dados.

5.2 Testar base de dados

5.2.1 Critérios de aprovação/reprovação

Aprovação: O sistema deve interagir corretamente com a base de dados, realizando operações de leitura e escrita conforme necessário.

Reprovação: Falhas ao conectar-se à base de dados ou problemas na execução de consultas.

5.2.2 Casos de teste

Conecta-se à base de dados com sucesso.

Executa uma consulta de leitura na base de dados.

Executa uma operação de escrita na base de dados.

Verifica se os dados na base de dados são consistentes com as ações realizadas.

5.3 Testar Login

5.3.1 **Aprovação:** O sistema deve autenticar corretamente os utilizadores com credenciais válidas e retornar o tipo de utilizador correto.

Reprovação: Falha na autenticação ou comportamento inesperado durante o processo de login.

5.3.2 Tentar fazer login com credenciais válidas.

Tentar fazer login com credenciais inválidas.

Verificar se o sistema lida corretamente com tentativas de login mal sucedidas.

Verificar se o sistema redireciona os utilizadores autenticados para a área correta

5.4 Testar Validar emails

5.4.1 **Aprovação:** O sistema deve apenas aceitar emails válidos e rejeitar emails inválidos.

5.4.2 **Reprovação:** O sistema não deve aceitar emails inválidos e rejeitar emails válidos.

5.5 Testes rejeitados

5.6 Testes falhados