

# 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.....	6
3.3	Relatórios e análise de dados.....	6
4	Requisitos de ambiente.....	7
4.1	Requisitos de hardware .....	7
4.2	Requisitos de software.....	8
5	Funcionalidades a ser testadas por Sprint .....	9
5.1	Testar adicionar operadores e fornecedores.....	9
5.1.1	Critérios de aprovação/reprovação.....	9
5.1.2	Casos de teste .....	9
5.2	Testar base de dados .....	9
5.2.1	Critérios de aprovação/reprovação.....	9
5.2.2	Casos de teste .....	9
5.3	Testar Login.....	10
5.4	Testes que falharam .....	10
5.5	Testes rejeitados .....	10
5.6	Testar adicionarEncomenda .....	10

# 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 LerUtilizadores:

**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 Ler 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 Ler Operadores:

**Teste testVerificarLoginUtilizadorOperador:**

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).

**Teste testVerificarLoginOperador:**

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

Escopo:

Adiciona lógica de simulação para um operador.

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

**Teste testAdicionarOperadorBaseDeDados:**

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 testAdicionarFornecedorBaseDeDados\_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.

**AdicionarEncomendaTest:**

Objetivo: Verificar se o método de inserção de encomenda funciona de acordo com o esperado.

Escopo: Simula os objetos necessários para inserir encomenda, tais como encomenda, produto e linhaEncomenda

Adiciona uma encomenda e verifica se falhou.

## 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

### 3.2 UAT Execution

Tempo de execução do package TestesUnitários é de 1seg e 913ms.

Os testes executados foram:

LerOperadorTest:

- testAdicionarOperadorBaseDeDados
- testAdicionarOperadorBaseDeDados\_Falha

LerFornecedorTest:

- testVerificarFornecedorBaseDeDados
- testVerificarFornecedorBaseDeDados\_Falha
- testAdicionarEExcluirFornecedorBaseDeDados

LoginTest:

- testVerificarLoginOperador
- testVerificarLoginUtilizadorFornecedor
- testVerificarLoginUtilizadorAdm

AdicionarEncomendaTest:

- adicionarEncomenda

### 3.3 Relatórios e análise de dados

Em relação aos testes de Login, registo de fornecedores e registo de operadores.

O teste adicionarEncomenda, insere uma encomenda e caso o teste passe, elimina a encomenda da base de dados, caso contrário, o teste falhou.

## 4 Requisitos de ambiente

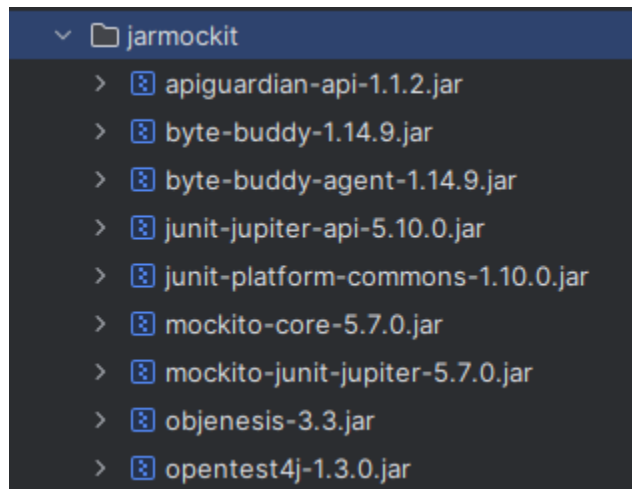
### 4.1 Requisitos de hardware

- 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>
```





## 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 malsucedidas.

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

## 5.4 Testes que falharam

5.4.1 LerFornecedoresTest -> testAdicionarFornecedorBaseDeDados

## 5.5 Testes rejeitados

O Teste de Login não é válido, pois não está a fazer comparação com a base de dados.

Os testes de LerOperadorTest e LerFornecedorTest, tem de ser melhorados.

Usar o SetUp e o TearDown nos testes.

## 5.6 Testar adicionarEncomenda

**5.6.1 Aprovação:** A encomenda deve ter os objetos encomenda, produtos e linhaEncomenda e a mesma deve ser validada e inserida na BD.

**5.6.2 Reprovação:** O adicionarEncomenda não é validado e a mesma não passa para a BD.