**3.5 Testes (TST)**

|  |  |  |
| --- | --- | --- |
| Unidade: TST | Data: 2018.11.06 | Coordenador: Carlos Sobral |
| Status: baselined | Versão: 1.0 | Vice-Coordenador: João Tomás |

**Lista de membros desta unidade:**

|  |  |  |  |
| --- | --- | --- | --- |
| Carlos | MEEC | Coordenador | carlosfssobral@gmail.com |
| João Tomás | LEI | Vice-Coordenador | joao.miguel.tomas@hotmail.com |
| Ana Teresa | MEEC | Colaboradora | anateresa.salgueiro31@gmail.com |
| Miguel | MEEC | Colaborador | Migueldiogo7@gmail.com |
| Isabel | LEI | Colaboradora | isabelmrcc@gmail.com |

**Objetivos da unidade de TST:**

* Garantir a qualidade de conteúdo e de estrutura do trabalho realizado pela unidade de implementação, recorrendo ao processo de validação
* Garantir que os requisitos do cliente são implementados e que o código desenvolvido é aplicado corretamente
* Realizar testes de caixa-preta relativos aos diferentes aspetos da aplicação, nomeadamente o *login*, *feed* e perfil.
* Realizar testes automatizados
* *Deployment* das diferentes versões do produto na plataforma *web*

**Processos utilizados por esta unidade:**

|  |  |
| --- | --- |
| 1. Processo para testar as funcionalidades implementadas | MQ\_TST\_TESTE\_FUNCIONALIDADES |
| 1. Processo de escrita de testes | MQ\_TST\_TEST\_VECTOR |
| 1. Processo de reuniões | MQ\_ENV\_REUNIÕES |
| 1. Processo de *Deployment* | MQ\_TST\_DEPLOYMENT |

**3.5.1 MQ\_TST\_TESTE\_FUNCIONALIDADES**

|  |  |  |
| --- | --- | --- |
| Unidade: TST | Data: 2018.11.06 | Coordenador: Carlos Sobral |
| Status: baselined | Versão: 1.0 | Vice-Coordenador: João Tomás |

**Descrição do processo:**

Este processo tem como objetivo verificar se o código implementado por parte da equipa de implementação fornece ao utilizador as funcionalidades desejadas por parte dos requisitos do cliente.

**Inputs:**

Diferentes casos de testes, conforme o MQ\_TST\_TEST\_VECTOR (documento produzido no processo 3.5.2), de forma a testar as diferentes funcionalidades implementadas na aplicação ORCWall e também de acordo com o documento dos requisitos para o sprint em questão.

Apresentações semanais:

https://github.com/carladrim/carladrim.github.io/tree/master/Testes/Sprint%201[%20-%20Testes/apresenta%C3%A7%C3%B5es%20semanais (1ª sprint)](https://github.com/carladrim/carladrim.github.io/tree/master/Testes/Sprint%201%20-%20Testes/apresenta%C3%A7%C3%B5es%20semanais)

[https://github.com/carladrim/carladrim.github.io/blob/master/Testes/Sprint%203%20-%20Testes/Sprint3\_week2.pptx (3ª sprint)](https://github.com/carladrim/carladrim.github.io/blob/master/Testes/Sprint%203%20-%20Testes/Sprint3_week2.pptx)

<https://github.com/carladrim/carladrim.github.io/blob/master/Apresenta%C3%A7%C3%B5es%20e%20Documenos%20da%20Semana%20Atual/TST/Sprint4_week3.pptx> (4ª sprint)

**Critério de entrada/ativação:**

Uma mensagem automática do GitHub que apareça no Slack, enviada após uma submissão no GitHub por parte de algum dos membros da equipa de implementação. Esta mensagem informa que uma nova funcionalidade foi desenvolvida e está pronta para ser testada.

**Descrição das tarefas:**

As tarefas são divididas entre os diversos membros da unidade de testes, ficando cada um encarregue de testar uma certa funcionalidade implementada. Os testes realizados estão de acordo com o MQ\_TST\_TEST\_VECTOR (documento produzido no processo 3.5.2).

O tester deve inserir em cada parâmetro o valor descrito para o mesmo. Por fim, deve alterar o estado do teste para *passed* or *failed*, caso o resultado esteja, ou não, de acordo com o output descrito no vetor de testes. Caso o teste falhado, o requisito não é satisfeito e é gerada uma mensagem para a unidade de implementação, para que esta possa corrigir o erro, de forma a satisfazer o requisito presente na tabela de requisitos. Quando um conjunto de testes é concluído, é gerada uma mensagem para a equipa de implementação, a informar que o código passou nos testes, ou, em caso de falha, a informar que é necessário reabrir a tarefa em causa para que o código seja corrigido.

**Critério de saída:**

Assim que o código passar com sucesso em todos os testes presentes na tabela de testes e a aplicação cumpra os requisitos presentes na tabela de requisitos, o processo é dado como terminado e o **processo de *Deployment* pode ser iniciado.**

**Métricas de avaliação deste processo:**

* Número de pessoas envolvidas em cada teste;
* Horas de trabalho semanais por cada elemento da unidade;
* Número de testes realizados por requisito;
* Taxa de testes aprovados/reprovados por requisito;

**Outputs gerados (artefactos):**

O artefacto gerado é a tabela MQ\_TST\_TESTES, onde é apresentado o resultado dos testes efetuados (aprovado/reprovado), com a data em que foi testado e com as respetivas observações sobre o mesmo.

**3.5.2 MQ\_TST\_TEST\_VECTOR**

|  |  |  |
| --- | --- | --- |
| Disciplina: TST | Data: 2018.11.06 | Coordenador: Carlos Sobral |
| Status: baselined | Versão: 1.0 | Vice-Coordenador: João Tomás |

**Descrição do processo:**

Consiste na escrita dos testes a ser realizador por um tester (pessoa encarregue de testar o código), em particular as pré-condições, os inputs (texto a ser preenchido e botões a clicar) e o resultado esperado.

As pré-condições contêm o estado do sistema antes do teste ser realizado (dados da aplicações e página onde o utilizador se encontra), os inputs referem-se aos dados inseridos pelo testes e que pretendem alterar o sistema (p.ex. criar um novo utilizador) e o output diz respeito ao estado final do sistema e resultados obtidos.

Este documento contém uma série de casos de testes para cada requisito criado.

**Inputs:**

O **MQ\_REQ\_SIS** foi o documento utilizado como input para este processo visto que contém a tabela de requisitos e listas de *mockups*, ambos estritamente necessários para a compreensão do código a implementar e dos consequentes testes a realizar.

**Critério de entrada/validação:**

Uma mensagem gerada no Slack, por parte de um dos elementos da unidade de implementação, para a unidade de testes, a informar qual a tarefa a ser implementada. Deste modo, o coordenador da unidade de testes pode distribuir as tarefas pelos elementos da sua unidade, para que seja possível aos elementos da mesma preparar testes para o requisito a ser implementado.

**Descrição das tarefas:**

As tarefas são distribuídas pelos elementos da unidade. Cada elemento terá que usar, como base, os artefactos de input e desenvolver um conjunto de testes que permitam averiguar se o sistema funciona como esperado e se corresponde aos requisitos do cliente.

**Critério de saída:**

A partir do momento em que cada requisito contenha um conjunto de casos de teste suficiente para que seja testado (cerca de cinco testes por requisito), então o processo dá-se por concluído. Este será mais tarde utilizado pela unidade de testes para testar as funcionalidades implementadas. Apesar de estar concluído e a ser usado por um outro processo, este documento pode ser reaberto, para adição ou edição de casos de teste.

**Métricas de avaliação deste processo:**

* Número de testes criados por requisito;
* Carga de trabalho semanal por elemento (em horas);
* Qualidade das pré-condições e do *output*;

**Outputs gerados (artefactos):**

É gerada uma tabela de testes que irá conter as etapas necessárias para realizar cada caso de teste e também campos para preencher num outro processo.

**3.5.4 MQ\_TST\_DEPLOYMENT**

|  |  |  |
| --- | --- | --- |
| Disciplina: TST | Data: 2018.11.06 | Coordenador: Carlos Sobral |
| Status: baselined | Versão:1.0 | Vice-Coordenador: João Tomás |

**Inputs:**

Verificar o código desenvolvido pela unidade de implementação, que está disponível no GitHub. Ver a Versão da aplicação.

**Critério de entrada/ativação:**

Aprovação por parte dos elementos da unidade, após a finalização, com sucesso, dos testes realizados.

**Descrição das tarefas:**

As tarefas são distribuídas pelos elementos da unidade. Cada elemento terá que usar as aplicações disponíveis para a realização dos testes.

**Validação:**

É feita a validação do produto, depois de todos os testes e operações realizadas estarem a funcionar sem erros fatais.

**Critério de saída:**

Os Testes realizados ao produto final estarem concluídos com sucesso. E o produto está pronto para ser entregue ao cliente.

**Métricas de avaliação deste processo:**

* Número de pessoas envolvidas em cada teste;
* Horas de trabalho semanal de cada elemento;
* Qualidade do output.

**Outputs gerados (artefactos):**

Número de horas despendidas na tarefa

Versão da aplicação