

## QUALIDADE

Disciplina: Qualidade	Data: 17/09/2018	Coordenadores: Diogo Gonçalves
Status: baselined	Versão: v1.0	Vice: Diogo Pereira

### Lista de membros desta unidade:

Diogo Gonçalves	LDM	Coordenador	uc2016217851@student.uc.pt
Diogo Pereira	LDM	Colaborador	uc2016217862@student.uc.pt
Maxmilian Rubin	LDM	Colaborador	uc2016220558@student.uc.pt
Fábio Barata	LDM	Colaborador	uc2015259622@student.uc.pt
João Rodrigues	MIEEC	Colaborador	uc2015232235@student.uc.pt

### Objetivos da Unidade de Qualidade:

- Assegurar a qualidade da estrutura e conteúdo dos documentos elaborados pelas diversas subunidades
- Avaliar os processos das outras unidades.
- Avaliar o estado do projeto.

### Processos utilizados por esta unidade:

1.Processo de validação de documentos	MQ_VALIDAÇÃO-DOCS
2. Processo de avaliação do estado do projeto	MQ_ESTADO-PROJETO
3.Processo de elaboração de Atas	MQ_ELABORAÇÃO-ATAS

## MQ\_TST\_VALIDAÇÃO-DOCS

Disciplina: Qualidade	Data: 17/09/2018	Coordenadores: Diogo Gonçalves
Status: baselined	Versão: v1.0	Vice: Diogo Pereira

### Descrição do processo:

Este processo descreve a essência da validação de documentos por parte da unidade de Qualidade e Testes.

No dia em que a subunidade de Qualidade foi criada o seu gestor dividiu os colaboradores pelas diversas subunidades. Deste modo quando um documento é solicitado para validação irá facilitar o trabalho pois as tarefas já estarão predefinidas. Caso um dos membros encarregue de fazer a validação não responder/não conseguir executar a tarefa pretendida até a data limite a mesma passará para outro dos membros escolhido pelo gestor de qualidade.

### Pessoas e os seus roles:

Cada membro está encarregue por uma a duas subunidades, tratando da validação de documentos da mesma(s).

### Inputs:

Os artefactos que servem de input a esta atividade são quaisquer tipo de documentos realizados por umas da subunidades. Como por exemplo documentos de processos de métodos de trabalho.

### Critério de entrada/ativação:

Para o arranque deste processo, é necessário que o responsável pelo documento solicite a sua validação ao Gestor de Qualidade.

### Descrição das tarefas:

#### 1. Avaliação da estrutura do documento:

-Nesta fase verificamos se o documento tem capa com autores, nome da equipa e PL, data, versão, introdução, cabeçalho e rodapé. Todos os templates realizados pela equipa de gestão de projecto encontram-se na Google Drive da equipa.

#### 2. Avaliação do conteúdo do documento:

- Análise do conteúdo ao que foi solicitado e correção de erros de conteúdo
- Correção de erros ortográficos

**Métricas de avaliação deste processo:**

- Número de documentos validados (por semana);
- Esforço (por pessoa, número de horas gastas);
- Número de alterações feitas em cada documento;

**Outputs gerados:**

- Lista de alterações feitas no documento;
- Sheet com informação dos responsáveis por validação de documento, assim como o número de horas gastas na sua validação.

## MQ\_ESTADO-PROJETO

Disciplina: Qualidade	Data: 17/09/2018	Coordenadores: Diogo Gonçalves
Status: baselined	Versão: v1.0	Vice: Diogo Pereira

### Descrição do processo:

Este processo foi criado com o objetivo de monitorizar o curso deste projeto e da equipa ao longo do tempo.

### Pessoas e os seus roles:

Cada um dos membros da equipa de qualidade está responsável por supervisionar uma a duas subunidades através dos formulários semanais, onde está detalhado o tempo estimado de trabalho, o tempo efetivo de trabalho e com a descrição de cada tarefa e quem a fez.

### Inputs:

Esta atividade terá como input um google form, onde cada colega de cada unidade tem de reportar o que fez durante a semana.

### Critério de entrada/ativação:

A monitorização dá-se por iniciada às 23h:59 de Domingo, que é a data limite do preenchimento do questionário do google.

### Descrição das tarefas:

Recolha dos dados do Google form, nomeadamente a soma das horas e as tarefas de cada disciplina. Ao notar que uma ou mais subunidades teve uma alteração drástica de horas de trabalho é analisada a sua situação, se não houve trabalho para realizar ou se não foi entregue a tempo.

### Outputs gerados:

Gráfico elaborado pela parte de gestão de projeto onde refere horas de trabalho estimadas, as horas de trabalho efetivas, tarefas realizadas e quem as fez.

### Métricas de avaliação deste processo:

- Número de pessoas que respondem ao questionário;
- Esforço (por disciplina RUP, número de horas gastas);

### Monitorização deste processo pela equipa de Qualidade:

Os outputs (documentos word) são monitorizados pelo Gestor de Qualidade a fim de detectar anomalias e de analisar o estado do projeto para perceber em que fase este se encontra.

## MQ\_ELABORÇÃO-ATAS

Disciplina: Qualidade	Data: 17/09/2018	Coordenadores: Diogo Gonçalves
Status: baselined	Versão: v1.0	Vice: Diogo Pereira

### Descrição do processo:

Semanalmente o gestor de qualidade designa um dos membros da sua subunidade para proceder à elaboração da respetiva ata, sendo que esse mesmo fica com responsabilidade de tomar notas de tudo o que seja relevante, bem como tirar fotos à folha de presença e ao caderno da gestora de projeto a fim de poder controlar a assiduidade dos colegas. Na google drive está disponibilizado um template da ata, produzido pela subunidade de gestão de projeto que por uma questão de uniformização de procedimentos deverá ser utilizado para registo de cada uma das reuniões.

### Pessoas e os seus roles:

-A cada reunião o gestor de qualidade escolhe um colaborador que fica encarregue da realização da ata, esse mesmo toma notas de tudo o que seja relevante para a elaboração do projeto.

### Inputs:

O artefacto que serve de input a esta atividade é o documento da ata.

### Critério de entrada/ativação:

-Este processo inicia após encerramento de cada reunião, sendo que o membro designado para a elaboração da ata em apreço terá um prazo de 24h para a concretização da mesma.

### Descrição das tarefas:

Logo após a designação do membro responsável pela elaboração da ata deve o mesmo proceder em conformidade com o template disponibilizado no google drive com vista à elaboração do documento tendente ao registo de todos os acontecimentos relevantes durante a reunião.

### Métricas de avaliação deste processo:

- Se a ata da reunião semanal é entregue em menos de 24h.
- Se a ata da segue o modelo disponibilizado

## Testes e Deployment (TD)

Disciplina: TD	Data: 2018.10.02	Coordenadores: Eduardo Guerra
Status: baselined	Versão:	Vice: Tomás Lopes

### Lista de membros desta unidade:

Eduardo Guerra	LEI	Coordenador	uc2016217851@student.uc.pt
João Campos	LDM	Colaborador	jgmmcampos@gmail.com
Tomás Lopes	LEI	Vice-coordenador	tomasgabriel1998@gmail.com
André Correia	LEI	Colaborador	andrefcorreia88@gmail.com
Ricardo Gonçalves	LDM	Colaborador	rgoncalvestlm@gmail.com

### Processos utilizados por esta unidade:

1. Processo para escrita de teses	MQ-ESCRITA_TST
2. Processo para averiguar se requisitos estão a ser cumpridos	MQ-AVR_REQ
3. Processo para testar código	MQ-TST_CÓDIGO
4. Processo para deployment	MQ-DEPLOYMENT

### Objetivos da unidade de TST:

- Garantir que os requisitos do cliente são implementados e que o código desenvolvido é bem aplicado, tendo em conta os diagramas elaborados pela unidade de requisitos.
- Garantir a qualidade e o bom funcionamento do código implementado.
- Fazer deployment do programa.

## MQ-ESCRITA\_TST

Disciplina: TD	Data: 2018.10.02	Coordenadores: Eduardo Guerra
Status: baselined	Versão:	Vice: Tomás Lopes

### Descrição do processo:

Consiste na escrita do procedimento a ser realizado por um tester (pessoa que testa código), em particular as pré-condições, os inputs (texto a ser preenchido e botões a clicar) e o resultado esperado. As pré-condições contém o estado do sistema antes do teste ser realizado (dados da aplicação e página em que o utilizador se encontra), os inputs referem-se aos dados inseridos pelo tester e que pretendem alterar o sistema (exemplo, criar uma nova entidade) e o output diz respeito ao estado final do sistema e resultados obtidos.

Para os requisitos funcionais, existe o documento “Testes funcionais”.

São baseados na lista de requisitos. Cada teste tem:

1. Objetivo do teste
  2. Página de pré-condição
  3. Valores de pré-condição
  4. Ação a realizar
  5. Resultado esperado
  6. Resultado obtido
  7. Data, realizador e horas despendidas no teste
- Um requisito é aprovado se o resultado esperado coincidir com o resultado pretendido. Caso contrário, é enviado uma mensagem à equipa de implementação de forma a estes corrigirem o programa.

Para os requisitos não-funcionais, existe o documento “Testes não-funcionais”. Estes testes serve para averiguar se os requisitos não-funcionais estão a ser cumpridos, podendo os testes ser de natureza diversa. Existem testes de carga e testes de usabilidade.

Cada teste de carga tem:

1. Constituição do teste
2. Reação do sistema (a que ponto começa a mostrar alterações)
3. Autor e horas despendidas

Se as métricas o indicarem, pode ter de haver correção do código.

Cada teste de usabilidade tem:

1. Constituição do teste
2. Número de cliques ideal
3. Número de cliques efetuado
4. Tempo ideal
5. Tempo efetivo
6. Tarefa concluída? (o utilizador pode não conseguir completar a tarefa, sendo um forte argumento a favor da dificuldade da utilização do programa)
7. Autor e horas despendidas

Caso as métricas desejáveis e as métricas obtidas sejam muito diferentes, pode ter de haver um redesign do layout.

**Inputs:**

Lista de requisitos  
Lista de mockups

**Critério de entrada/ativação:**

Uma mensagem gerada no Discord, enviada da team leader de implementação, para o coordenador desta unidade a informar qual a tarefa a ser implementada.

**Descrição das tarefas:**

São distribuídas tarefas pelos vários elementos da unidade. Cada elemento terá que usar, como base, os testes no documentos "Testes funcionais" e "Testes não-funcionais".

**Critério de saída:**

Assim que cada requisito contenha um conjunto de use cases suficiente para que seja testado(em média seis testes por requisito), então o processo dá-se como concluído, sendo que será mais tarde utilizado por esta unidade para testar código implementado. Apesar de concluído e em uso num outro processo, este documento pode ser novamente reaberto, para adição/edição de casos de uso.

**Métricas de avaliação deste processo:**

Número de testes criados por requisito;  
Qualidade das pré-condições e do output;  
Horas de trabalho semanais por elemento;

**Outputs gerados:**

Gera-se uma tabela de testes com as etapas a serem realizadas aquando de um teste. Esta tabela será preenchida na verificação de cumprimento de requisitos e de testes de código.

**Monitorização deste processo pela equipa de Qualidade:**

A unidade de qualidade faz uma comparação entre a tabela de requisitos e a tabela de testes para saber se os testes criados testam efetivamente o código implementado, se vão de acordo com os requisitos do cliente.

**Monitorização deste processo pela equipa de Qualidade:**

A unidade de qualidade faz uma comparação entre a tabela de requisitos e a tabela de testes para saber se os testes criados testam efetivamente o código implementado, se vão de acordo com os requisitos do cliente.



## MQ-AVR\_REQ

<b>Disciplina:</b> TD	<b>Data:</b> 2018.10.02	<b>Coordenadores:</b> Eduardo Guerra
<b>Status:</b> baselined	<b>Versão:</b>	<b>Vice:</b> Tomás Lopes

Processo para averiguar se requisitos estão a ser cumpridos

### Descrição do processo:

É feito uma série de testes a partir da lista de requisitos. No caso dos requisitos funcionais, estes são testados para averiguar se forma implementados ou não. No caso dos requisitos não-funcionais, estes são testados de acordo com os testes e métricas definidas no documento “Testes não-funcionais”. Os requisitos não funcionais devem ser testados de preferência com pessoas não familiares ao projeto.

### Inputs:

O documento de requisitos enviado pela equipa de testes.  
Versão do programa a testar.

### Output:

Lista de funcionalidades implementadas e por implementar.  
Lista de testes elaborado por esta unidade de modo a testar requisitos não-funcionais juntamente com as métricas recolhidas pelos diferentes elementos da equipa.  
Horas despendidas por cada elemento em cada tarefa.

### Critério de ativação:

O coordenador da unidade de implementação envia uma mensagem pelo Discord ao coordenador desta unidade de que há uma nova versão a ser testada. Este encarrega-se de elaborar lista de testes para requisitos não-funcionais e de distribuir tarefas. O resto da unidade acrescenta outros testes para requisitos não-funcionais caso ache pertinente.

### Descrição das tarefas:

São distribuídas tarefas pelos vários elementos da unidade. Cada elemento terá que averiguar se o sistema funciona corretamente e se está de acordo com os requisitos do cliente. Também deve recolher métricas e testar o programa inclusivamente com indivíduos fora do contexto de testes e implementação, de forma a obter dados realistas e demonstrativos do utilizador-alvo.

### Critério de saída:

Assim que cada elemento tenha realizado vários testes para cada requisito, este pode ser dado como concluído ou, caso o requisito não esteja a ser cumprido, é enviada uma mensagem à equipa de implementação de forma a que corrija a situação.

**Métricas de avaliação deste processo:**

Tabela com requisitos aprovados ou reprovados.

Número de testes realizados por requisitos.

Horas de trabalho semanais por elemento;

**Outputs gerados:**

Gera-se uma tabela de requisitos, que conforme tenham sido cumpridos ou não, recebem uma denominação de “aprovado” ou “reprovado”.

## MQ-TST\_CÓDIGO

<b>Disciplina:</b> TD	<b>Data:</b> 2018.10.02	<b>Coordenadores:</b> Eduardo Guerra
<b>Status:</b> baselined	<b>Versão:</b>	<b>Vice:</b> Tomás Lopes

Processo para testar código

### Descrição do processo:

Este processo tem como objetivo averiguar se o trabalho desenvolvido pela equipade implementação foi bem realizado, estando de acordo com os requisitos e que satisfaça as normas de programação e interação com um utilizador (não permitindo a inserçãode valores indesejados, por exemplo: uma expressão num campo numérico). Em suma, é um processo que testa o código implementado.

### Inputs:

O código realizado pela equipa de implementação que terá sido disponibilizado no GitHub, pelo coordenador da unidade de implementação, assim como o documento “Testes funcionais” e “Testes não-funcionais”.

### Critério de entrada/ativação:

Uma mensagem gerada no Discord, enviada do coordenador de implementação, para o coordenador desta unidade. Esta mensagem consiste em informar a unidade de testes de que uma tarefa já foi desenvolvida e que está pronta para ser testada.

### Descrição das tarefas:

São divididas tarefas e realizados testes de códigos, com base nos documentos “Testes funcionais” e “Testes não-funcionais”. Para os testes funcionais, o tester deve inserir em cada parâmetro o valor descrito para o mesmo. Por último deve alterar o estado do teste para passed ou failed, consoante o resultado do teste esteja de acordo com o output descrito no vetor de testes. Caso o teste falhe, o requisito não foi satisfeito na totalidade, sendo gerada uma mensagem à unidade de implementação para que esta possa corrigir o erro/satisfazer o requisito presente na tabela de requisitos. Quando um conjunto de testes para o mesmo requisito se der como concluído, é gerada uma mensagem para a equipa de implementação, a informar de que o código passou os testes, ou em caso de falha, é informada de tal, para que a tarefa de implementação de código correspondente ao requisito em causa, seja reaberta e o código corrigido. Para os testes não-funcionais, são realizados por elementos de preferência que estejam (de preferência) fora do contexto de implementação e testes e supervisionados por um elemento da unidade de testes. Este deve indicar ao utilizador as ações a realizar e recolher métricas e comentários com base na experiência do utilizador. Caso os testes indiquem uma má experiência de utilizador, é enviada uma mensagem ao coordenador de implementação a mostrar o resultado dos testes de forma a este poder corrigir e melhorar a experiência de utilização.

**Critério de saída:**

Assim que o código implementado satisfizer todos os testes presentes na tabela de testes e a aplicação desempenhe os requisitos presentes na tabela de requisitos então o processo é dado como terminado, é atualizado o projeto principal.

**Métricas de avaliação deste processo:**

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

**Outputs gerados:**

O artefacto gerado são documentos “Testes funcionais” e “Testes não-funcionais”, agora com o resultado dos testes (aprovado/reprovado), com a data em que foi testado e, com observações ao mesmo.

**Monitorização deste processo pela equipa de Qualidade:**

A unidade de qualidade faz uma leitura sobre o resultado de cada teste e averigua, em caso de falha, se a unidade de implementação foi notificada e se sabe o que deve editar.

**MQ-DEPLOYMENT**

<b>Disciplina:</b> TD	<b>Data:</b> 2018.10.02	<b>Coordenadores:</b> Eduardo Guerra
<b>Status:</b> baselined	<b>Versão:</b>	<b>Vice:</b> Tomás Lopes

**Descrição do processo:**  
O deployment é feito na própria máquina do tester, estando a aplicação limitada a esse utilizador.

**Inputs:**  
O código desenvolvido pela unidade de implementação, disponível no GitHub. Versão da aplicação.

**Output:**  
Aplicação deployed.

**Critério de entrada/ativação:**  
Existência de uma versão do produto devidamente testada, para ser feito o deployment.

**Validação:**  
Operação realizada sem erros fatais.

**Métricas de avaliação deste processo:**  
Versão da aplicação;

**Outputs gerados:**  
Número de horas despendidas na tarefa.  
Versão da aplicação.