



# Manual de Qualidade

**LabSync** · Engenharia de Software PL7 · Versão 2.1 · Data: 18/11/2018

---

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA - FCTUC - UNIVERSIDADE DE COIMBRA

### Tabela de versões:

Versão	Data da última alteração	Autores	Descrição
2.1	18/11/2018	Maximilian Rubin João Rodrigues Diogo Gonçalves Fábio Barata Diogo Pereira	Correção de erros. Acréscimo de processos.
2.0	30/10/2018	Maximilian Rubin João Rodrigues Fábio Barata Diogo Pereira Diogo Gonçalves	Atualização do layout do Manual de Qualidade. Acréscimo e correção dos processos em falta.
1.1	05/10/2018	João Rodrigues	Introdução dos processos de requisitos e testes
1.0	04/10/2018	Maximilian Rubin	Estrutura Inicial do manual.

# Índice

<b>Tabela de versões:</b>	<b>1</b>
<b>1. - Introdução</b>	<b>4</b>
1.1. - Contexto	4
1.2. - Apresentação do projeto	4
1.3. - Glossário	4
1.4. - Referências	4
1.5. - Plataformas usadas pela equipa	5
<b>2. - Regras gerais da equipa LabSync</b>	<b>6</b>
<b>3. - Apresentação dos processos da equipa</b>	<b>7</b>
3.1. - Gestão de Projeto [GP]	7
3.1.1. - MQ_LABSYNC_GP_PLAN_ALT_PLANO_PROJ	8
3.1.2. - MQ_LABSYNC_GP_PLAN_TRAB_SEM	9
3.1.3. - MQ_LABSYNC_GP_RECON_ATR_TRF	10
3.1.4. - MQ_LABSYNC_GP_INAT_MBR	11
3.1.5. - MQ_LABSYNC_GP_REC_ESFOR_SEM	13
3.1.6. - MQ_LABSYNC_GP_TEMPL_APRES_SEM	14
3.2. - Recursos Humanos [HR]	15
3.2.1 - MQ_LABSYNC_HR_PARTICIPAÇÃO	16
3.2.2 - MQ_LABSYNC_HR_RECON_ATR_TRF	17
3.2.3 - MQ_LABSYNC_HR_INAT_MBR	18
3.2.4 - MQ_LABSYNC_HR_MOTIVAÇÃO	19
3.2.5 - MQ_LABSYNC_HR_ORGANIZAÇÃO	20
3.3. - Riscos [RSK]	21
3.3.1. - MQ_LABSYNC_RSK_IDENTIFICAR	22
3.3.2. - MQ_LABSYNC_RSK_AVALIAR	23
3.3.3. - MQ_LABSYNC_RSK_RESOLVER	24
3.4. - Requisitos [REQ]	25
3.4.1.- MQ_LABSYNC_REQ_TABELA_REQ	26
Validação:	27
Métricas de avaliação deste processo:	27
3.4.2.- MQ_LABSYNC_REQ MOCKUPS	28
3.4.3.- MQ_LABSYNC_REQ_SRS	31
3.5. - Implementação [IMP]	33
3.5.1. - MQ_LABSYNC_IMP_SPRINT	34
3.5.2. - MQ_LABSYNC_IMP_ID	35
3.5.3. - MQ_LABSYNC_IMP_DEV	36
3.5.4. - MQ_LABSYNC_IMP_FIX	37

3.6. - Testes [TST]	38
3.6.1. - MQ_LABSYNC_TST_ESCRITA_TST	39
3.6.2. - MQ_LABSYNC_TST_AVR_REQ&COD	41
3.6.4. - MQ_LABSYNC_TST_DEPLOYMENT	43
3.7. - Qualidade [QUA]	44
3.7.1. - MQ_LABSYNC_QUA_VALIDAÇÃO_DOCS	45
3.7.2. - MQ_LABSYNC_QUA_ESTADO_PROJETO	46
3.7.3. - MQ_LABSYNC_QUA_ELABORAÇÃO_ATAS	48
3.7.4. - MQ_LABSYNC_QUA_REUNIÕES	49
3.8. - Ambiente [ENV]	50
3.8.1. - MQ_LABSYNC_ENV_ATENDIMENTO_TICKETS	51

# 1. - Introdução

## 1.1. - Contexto

Este manual de qualidade foi desenvolvido no âmbito da disciplina de Engenharia de Software do Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologias da Universidade de Coimbra pela equipa de LabSync [PL7] e sob orientação dos professores responsáveis Álvaro Manuel Reis da Rocha e Mário Alberto da Costa Zenha Relá.

## 1.2. - Apresentação do projeto

Este projeto, designado INFOR-INVESTIGADOR, é uma plataforma que pretende servir de suporte ao trabalho dos investigadores da Universidade de Coimbra, ainda que a sua aplicabilidade seja universal, sendo desenhado nesse pressuposto. Um investigador, neste contexto, tanto pode ser um estudante de Mestrado ou Doutoramento, como um investigador sénior ou um ao seu percurso de investigação, digamos assim, um ‘mentor’ digital, elas podem ser utilizadas e alargadas para suportar as tarefas de investigadores mais experimentados.

## 1.3. - Glossário

[GP]: Gestão de Projeto

[REQ]: Requisitos

[IMP]: Implementação

[TST]: Testes

[QUA]: Qualidade

[ENV]: Ambiente

[RSK]: Riscos

[HR]: Recursos Humanos

## 1.4. - Referências

Este Manual de Qualidade tem como principal referência o exemplo disponibilizado no InforEstudante pelo professor Mário Alberto da Costa Zenha Relá, que se encontra no Material de Apoio desta disciplina, cujo nome do ficheiro é “Exemplo1-MQ\_v3.0a”.

## 1.5. - Plataformas usadas pela equipa

### Discord

(<https://discordapp.com>)

Ferramenta de comunicação adotada pela unidade (LabSync), e o método primordial de comunicação da subunidade, seja esta comunicação interna (canal da subunidade) ou externa (canal geral da unidade, canais intermediários com Requisitos, Testes e Gestão).

### GitHub

(<https://desktop.github.com>)

Definido pelo cliente e adotado pela unidade como repositório para todos os artefatos produzidos pela subunidade de implementação.

### Trello

(<https://trello.com>)

Ferramenta adotada pela unidade para organização e distribuição de tarefas pelos membros da subunidade e que permite delinear uma data limite para conclusão dessas mesmas tarefas.

### Python/ Django

(<https://www.python.org> - <https://www.djangoproject.com>)

Linguagem de programação e Web framework adotadas pela subunidade para criação e desenvolvimento da WebApp, escolhidas no início do desenvolvimento por viabilizarem os problemas iniciais do desenvolvimento.

### Figma

(<https://www.figma.com>)

Plataforma para prototipagem e design destinada a mockups de alta fidelidade. Sendo uma plataforma online permite que estejam várias pessoas a trabalhar no mesmo projeto, podendo não só ver todo o trabalho desenvolvido como participar em tempo real em conjunto com os outros.

### Google Drive, Google Docs, Google Sheets

(<https://www.google.com/docs>)

Plataformas genéricas ao resto da equipa. Todos os membros da PL têm acesso aos documentos desta unidade.

### Outros

Ferramentas de criação e de edição, nomeadamente de artefatos referentes a IMP ficam ao critério de cada membro da subunidade, permitindo assim a estes trabalhar da forma em que se sentem mais adaptados e confortáveis, visto que esta metodologia não gera riscos de incompatibilidade e corrupção destes mesmos artefatos.

## 2. - Regras gerais da equipa LabSync

Esta secção do Manual tem como objetivo documentar as regras das decisões tomadas pela equipa assim como as competências de cada um nas mesmas.

1. Todas as decisões transversais ao projeto têm de ser tomadas, de forma democrática, pela equipa de gestores ficando a cargo do Coordenador de projeto o voto de qualidade;
2. Todas as decisões que tenham influência apenas dentro de uma determinada equipa são tomadas de forma democrática dentro dessa equipa, ficando o voto de qualidade a cargo do responsável da mesma;
3. Todas as decisões que tenham de ser tomadas ao nível de um determinado setor (Tecnologia, Processos ou Pessoas) serão tomadas de forma democrática pelos gestores desse setor ficando o voto de qualidade a cargo do Coordenador desse setor;
4. Sempre que alguma decisão de equipa ou setor possa causar a divisão dos mesmos, esta deve ser avaliada pelos gestores;
5. A não conformidade com estas regras poderá levar a perda do cargo;
6. Todas as reuniões realizadas têm de ter uma ata, que deverá ser enviada à equipa de qualidade para análise e arquivo da mesma;
7. Será facultado, todas as semanas, um formulário para a apreciação da semana e para facilitar o registo individual por parte da equipa de qualidade;
8. Qualquer elemento descontente com o seu cargo ou função poderá comunicá-lo a equipa de recursos humanos para que a sua situação seja alterada;
9. Este documento poderá ser alterado sempre que a equipa de gestão e de qualidade, em conformidade com o resto dos elementos do projeto, assim o entendam;

### 3. - Apresentação dos processos da equipa

#### 3.1. - Gestão de Projeto [GP]

<b>Unidade:</b> GP	<b>Data:</b> 27.10.2018	<b>Gestora:</b> Maria Moreira
<b>Versão:</b> 2.1	<b>Status:</b> Baselined	<b>Vice Gestor:</b> Fábio Ferreira

#### Lista de membros desta unidade:

Nome	Curso	E-mail
Maria Moreira	LDM	uc2016244493@student.dei.uc.pt
Fábio Ferreira	LEI	uc2016251749@student.uc.pt

#### O que é esta unidade e como funciona:

Esta unidade tem como função gerir toda a equipa, controlar e garantir o decorrer do projeto conforme o planeado.

Também é da função desta unidade a marcação de reuniões gerais ou de gestores, para discussão de assuntos nos quais não houve consenso dentro desse setor ou equipa.

#### Objetivos da unidade de GP:

- 1) Distribuição de tarefas pelas várias equipas, ficando ao encargo do Coordenador dessa subunidade a distribuição das mesmas.
- 2) Servir de ponto de ligação entre os três setores: (Pessoas, Produto e Qualidade).
- 3) Gerir a tomada de decisões e possíveis conflitos.
- 4) Fazer com que o projeto avance, tomando decisões nesse sentido.
- 5) Dar um rumo ao projeto, fazendo um Plano de Projeto.
- 6) Intervir, sempre que achar necessário, em alguma equipa.



### 3.1.1. - MQ\_LABSYNC\_GP\_PLAN\_ALT\_PLANO\_PROJ

#### Processo de Planificação e Alteração do Plano de Projeto

##### *Entry Criteria:*

Foi criado um documento com o Plano de Projeto, de modo a existir uma melhor organização do que vem a ser feito e do que é pretendido fazer. Este documento encontra-se disponibilizado na Drive do grupo nomeado por GP\_Plano de Projeto, e também no repositório GitHub na pasta Gestão de Projeto nomeado por PM\_V1\_Plano de Projeto.

##### *Task Description:*

Este processo está sempre em curso, começando no início do projeto, enumerando os diversos objetivos a ser cumpridos e vai sofrendo alterações à medida que avança, de acordo com o trabalho proposto e realizado.

Idealmente será alterado de duas em duas semanas, não só de acordo com o cumprimento ou à falta do mesmo relativamente às diferentes fases do projeto. Esta alteração pode ser feita pela Gestora de Projeto - Maria Moreira, ou pelo vice de Gestão de Projeto - Guilherme Nogueira.

Após este processo ser iniciado, os gestores de projeto como estando autorizados a mudar o plano de projeto, comunicam à gestora de projeto (Maria Moreira) as propostas de modificações que têm em mente de modo a que o Plano de Projeto seja adequado ao decorrer do projeto. Depois destas modificações serem aprovadas, será então modificado no documento para que este fique atualizado.

Se tais modificações não forem propostas, a gestora de projeto (Maria Moreira) e o vice (Guilherme Nogueira) reúnem-se num espaço de duas semanas de modo a analisarem o documento e discutirem se existe algo a modificar.

##### *Validation:*

- 1) Reunião com a Gestora de Projeto, Vice Coordenador de Projeto e Coordenador de Riscos de modo a analisar o documento e estudar alterações de acordo com o decorrer do projeto;
- 2) Redação de uma nova versão do Plano de Projeto dentro do período de 3 dias seguintes à reunião;
- 3) Revisão e aprovação do documento pela Gestão de Qualidade dentro de um período máximo de 12h após a entrega do documento a partir da Gestora de Projeto;
- 4) Publicação no channel general do Discord da nova versão do Plano de Projeto.

##### *Exit Criteria:*

Este processo é dado como actualizado quando é publicado e dado a conhecer a nova versão do documento no *channel* general do Discord e publicado na Drive do Projeto.

### *Métricas:*

- 1) Reunião com Gestão de Projeto e Gestão de Riscos num intervalo médio de duas semanas;
- 2) Realização de um documento resultante da reunião que categorize as mudanças a fazer;
- 3) Redação da nova versão do documento num tempo máximo de três dias seguintes à reunião;
- 4) Entrega da nova versão do Plano de Projeto para revisão à Gestão de Qualidade em média um dia antes do prazo acima estipulado (três dias seguintes à reunião);
- 5) Revisão da nova versão durante um período máximo de 12h, podendo voltar à Gestora de Projeto no máximo uma vez para alterações;
- 6) Publicação do documento atualizado no channel general do Discord em média 2h anteriores ao fim do prazo.

## **3.1.2. - MQ\_LABSYNC\_GP\_PLAN\_TRAB\_SEM**

### **Processo de Planificação do Trabalho Semanal**

#### *Entry Criteria:*

O processo tem início todas as semanas no final da aula PL.

#### *Task Description:*

No fim de cada aula PL, inicia-se uma nova semana de trabalho e os coordenadores de cada subunidade têm de organizar as tarefas a realizar durante essa semana de trabalho.

Esta parte do processo decorre até às 13h do dia seguinte (quarta-feira), sendo que até esta hora todos os coordenadores deverão ter uma lista de tarefas a realizar, com quantas pessoas são necessárias à realização das mesmas, uma deadline de entrega e uma descrição detalhada exposta no quadro da respectiva subunidade no Trello, onde a gestora de projeto (Maria Moreira) consegue acompanhar o desenvolvimento das tarefas das diversas subunidades ao longo de cada semana, e notificarem no channel privado de coordenadores do Discord que a lista já se encontra disponível.

#### *Validation:*

1. Entrega da lista de tarefas a realizar por cada disciplina RUP até às 13h de quarta-feira via Trello e notificação via Discord;
2. As tarefas propostas são validadas sempre que depois de colocada a lista de tarefas no canal apropriado ninguém se manifeste negativamente no prazo de 24h.

### *Exit Criteria:*

Este processo é dado como terminado quando todas as subunidades atualizam as tarefas da semana no respetivo quadro do Trello e a gestora de projeto (Maria Moreira) toma conhecimento das tarefas a realizar pela equipa na semana corrente.

### *Métricas:*

- 1) Pelo menos 80% das disciplinas RUP têm de entregar o documento de tarefas a realizar pela disciplina RUP na presente semana até às 13h de quarta-feira;
- 2) Idealmente todas as disciplinas RUP têm de entregar a lista de tarefas a realizar em média 1h30 antes do prazo estipulado (12h30 de quarta-feira);
- 3) As tarefas têm de estar revistas na totalidade em média até às 13h45 pela Gestora de Projeto (Maria Moreira) e Coordenador de Recursos Humanos (Guilherme Pontes).

## **3.1.3. - MQ\_LABSYNC\_GP\_RECON\_ATR\_TRF**

### **Processo de Reconhecimento de Atribuição De Tarefas**

### *Entry Criteria:*

Este processo entra em ação depois do processo MQ\_LABSYNC\_GP\_PLAN\_TRAB\_SEM ser dado como concluído, onde a Gestora de Projeto (Maria Moreira) e o Coordenador de RH (Guilherme Pontes) tomam conhecimento das tarefas a realizar por cada disciplina RUP na semana em questão.

### *Task Description:*

Após a revisão e aprovação das tarefas a realizar pela Gestora de Projeto e de RH, cada coordenador de cada disciplina RUP informa a partir do Discord que as tarefas estão expostas no Trello no máximo até às 15h, dando um prazo máximo de 6h30 (até às 21h30) de quarta-feira para cada membro tomar conhecimento das tarefas a realizar e encarregar-se de uma tarefa, tornando-se responsável de a realizar com sucesso, tendo sempre supervisão do coordenador da disciplina RUP em questão.

Após responsabilizar-se, o planeamento das tarefas semanais de cada disciplina RUP é atualizado de modo a que a descrição da tarefa tenha presente a identificação dos responsáveis no Trello.

Se, por alguma razão, nenhum membro se encarregue por uma tarefa até às 21h30, a situação é reportada ao Coordenador de RH, que se encarrega de definir um responsável para as tarefas em falta até às 23h de quarta-feira.

### *Validation:*

- 1) Aprovação das tarefas a realizar pela gestora de projeto (Maria Moreira).
- 2) Publicação das tarefas planeadas dentro de cada subunidade para cada membro se responsabilizar pelo menos por uma tarefa.
- 3) Tarefas atribuídas a todos os membros até às 23h de quarta-feira, onde serão guardadas através do *Trello*, dividido em boards por subunidades.

### *Exit Criteria:*

Este processo é dado como concluído quando todos os membros da equipa no geral possuem tarefas a realizar na semana de trabalho a iniciar.

### *Métricas:*

- 1) Publicação de uma notificação dando a informação de que as tarefas já estão planeadas(no *Trello*) no channel Discord da disciplina em questão a partir das 15h;
- 2) Atribuição de tarefas a todos os membros preferencialmente 1h30 antes do prazo estipulado (21h30 de quarta-feira);
- 3) Se a métrica anterior não se verificar, atribuição de tarefas a um mínimo de 75% dos membros até às 21h30 de quarta-feira;
- 4) Se faltar um responsável para alguma tarefa, a situação é reportada ao Coordenador de RH pelas 21h30, tendo 1h30 como prazo para tratar da situação;
- 5) Todos os elementos têm de ter uma tarefa atribuída até às 23h de quarta-feira.

## **3.1.4. - MQ\_LABSYNC\_GP\_INAT\_MBR**

### **Processo De Reconhecimento De Inatividade De Membros**

### *Entry Criteria:*

Este processo é ativo quando a Gestora(Maria Moreira) toma conhecimento de um membro que não comparece às aulas PL, não se encarrega voluntariamente com nenhuma tarefa proposta na disciplina RUP que se insere nem submete o formulário criado pela Gestora de Projeto durante um espaço de 2 semanas.

### *Task Description:*

No artefato produzido para supervisionar as presenças na aula PL (recolha de assinaturas na PL) e submissão do form criado pela Gestora de Projeto, que é atualizado todas as semanas

(<https://docs.google.com/spreadsheets/d/1Dq9Pqz1ZTk4Cflmrij3Vhw50yaeKwZayYwsHyLdiSX4/edit?usp=sharing>), é analisado pela Gestora se há algum caso de um membro que não tenha ido à aula nem submetido o form durante duas semanas seguidas ou não estando disponível no Discord, é declarado o estado de inativo àquele membro, sendo a situação depois reportada ao Coordenador de RH num espaço de 4h, que trata da situação.

Entre a comunicação da situação ao Coordenador de RH e a notificação da resolução da situação é estipulado um prazo de 24h.

### *Validation:*

- 1) Inatividade total de um membro da equipa num espaço de 2 semanas seguidas;
- 2) Reconhecimento da situação por parte da Gestora de Projeto (Maria Moreira);
- 3) Declaração do membro como inativo;
- 4) Comunicação da situação ao Coordenador de RH (Guilherme Pontes) para esta solucionar o problema;
- 5) Resolução do problema num espaço de 24h.

### *Exit Criteria:*

Este processo é dado como concluído quando a situação é reportada à Gestora de Projeto pelo Coordenador de RH como resolvida.

### *Métricas:*

- 1) Inatividade média de 85% por parte do membro da equipa em questão durante um período de tempo definido (duas semanas seguidas);
- 2) Declaração imediata do membro como inativo após reconhecimento da situação;
- 3) Comunicação da situação ao Coordenador de RH num espaço de 4h após reconhecimento da situação;
- 4) Problema resolvido em média 3h antes do prazo estipulado (24h após a comunicação da inatividade de um membro).

### 3.1.5. - MQ\_LABSYNC\_GP\_REC\_ESFOR\_SEM

#### Recolha De Esforço Semanal

##### *Entry Criteria:*

No final de cada semana (Segunda às 22h) há uma recolha das tarefas realizadas por cada membro.

##### *Task Description:*

Este processo passa por análise e comparação das tarefas propostas a desenvolver sobre as tarefas desenvolvidas.

Analisa a distribuição de tarefas feitas pelas várias equipas para verificar a sua conformidade com a deadline e com o rumo do projeto.

Para que esta recolha de informação seja feita de modo eficaz é usado um google form, semelhante ao que nos é disponibilizado pelo professor (<https://goo.gl/forms/zgzrq1LEcxZk60wE3>), dando um limite de submissão de resposta até às 22h de todas as segundas-feiras, de modo a que seja possível à gestora de projeto (Maria Moreira) organizar as tarefas realizadas de modo a conseguir perceber qual é a evolução de trabalho da sua equipa no geral e de cada disciplina RUP, para depois apurar algum problema existente ao Coordenador de RH.

Todos estes artefatos são produzidos numa google spreadsheet (<https://docs.google.com/spreadsheets/d/1Dq9Pqz1ZTk4Cflmrij3Vhw50yaeKwZayYwsHyLdiSX4/edit?usp=sharing>), de modo a poder guardar e avaliar as evoluções da equipa.

É também produzido um artefato que produz gráficos de evolução de esforço semanal por membro, disciplina RUP, proporção entre as horas estimadas e as horas trabalhadas, e a sua diferença com o número de horas estipulado semanal por equipa (<https://docs.google.com/spreadsheets/d/1Dq9Pqz1ZTk4Cflmrij3Vhw50yaeKwZayYwsHyLdiSX4/edit?usp=sharing>).

##### *Validation:*

- 1) Submissão do formulário por todos os membros da equipa até às 22h de segunda-feira.
- 2) Sempre que cumpridas as etapas descritas em cima e que depois de guardadas as informações no local indicado e ninguém se manifeste, ou que ninguém discorde da análise feita pela equipa de gestão de projeto;

##### *Exit Criteria:*

Este processo é dado como concluído quando a gestora de projeto (Maria Moreira) regista as tarefas realizadas na google spreadsheet e o esforço investido no mesmo para que haja coerência nas tarefas desenvolvidas e concretizadas.

### Métricas:

1. Registo do esforço semanal dos membros;
2. Se a métrica anterior não se cumprir, registo de pelo menos 80% da equipa até às 12h de cada terça-feira;

### 3.1.6. - MQ\_LABSYNC\_GP\_TEMPL\_APRES\_SEM

#### Processo de Templates de Apresentação Semanal

##### *Entry Criteria:*

Este processo é ativado a seguir à exportação da apresentação da semana anterior.

##### *Task Description:*

Este processo descreve a adaptação semanal do template de apresentações para as reuniões da equipa na PL.

A partir do momento que a Gestora de Projeto (Maria Moreira) criou o template de apresentações semanais, este foi disponibilizado a toda a equipa na Drive do grupo ([https://docs.google.com/document/d/1rlP860yoJxFMYfvT7m2BCTx5q\\_yJ1C9k6QOIJTdxJUc/edit#](https://docs.google.com/document/d/1rlP860yoJxFMYfvT7m2BCTx5q_yJ1C9k6QOIJTdxJUc/edit#)).

O vice-coordenador fica encarregue de apagar o conteúdo da semana anterior após a versão prévia ser exportada, para que a próxima possa ser preenchida sem confusões.

Seguidamente, cada equipa é responsável pela introdução dos dados relativos às atividades por si desenvolvidas, e desenvolver até terça-feira às 11 horas.

##### *Validation:*

Este processo é validado quando o template da apresentação semanal é exportado e é feito o seu upload no gitHub.

##### *Exit Criteria:*

O processo é aprovado, se for exportado com a informação disponibilizada pelas equipas até às 11 horas da manhã de terça-feira.

##### *Outputs/Artefactos Gerados:*

O *output* deste processo trata-se da apresentação semanal, para a reunião da equipa na PL, atualizada.

### Métricas:

1. Limpeza dos dados do template a editar;
2. Introdução dos novos dados relativos às tarefas da equipa;
3. Exportação e upload da apresentação semanal no gitHub.

### 3.2. - Recursos Humanos [HR]

<b>Unidade:</b> HR	<b>Data:</b> 27.10.2018	<b>Gestor:</b> Guilherme Pontes
<b>Versão:</b> 1.0	<b>Status:</b> Baselined	

#### Membro da unidade:

Guilherme Pontes	Gestor	LEI	pontes@student.dei.uc.pt
------------------	--------	-----	--------------------------

#### Função da unidade de HR:

Esta unidade tem como objetivo garantir o funcionamento e bem-estar de todos os membros, a comunicação eficaz entre as várias subunidades, evitando conflitos e facilitando a motivação individual de modo a atingir os objetivos estipulados para a equipa.

#### Objetivos:

- 1) Garantir a aderência dos membros às aulas PL, reuniões e tarefas a desenvolver.
- 2) Intervir, sempre que achar necessário, de modo a evitar conflitos.
- 3) Resolver conflitos, de modo a beneficiar o máximo de envolvidos possível e, acima de tudo, o rumo do projeto.
- 4) Recolher e interpretar dados relativos à organização e motivação da equipa.
- 5) Facilitar a comunicação entre subunidades.

#### Processos utilizados por esta unidade:

Processo	ID do Processo
1.Processo de avaliação da participação de cada membro	MQ_LABSYNC_RH_PARTICIPAÇÃO
2. Análise e classificação de um risco	MQ_LABSYNC_RH__RECON_ATR_TRF
3. Mitigar efeitos /Combater riscos possíveis	MQ_LABSYNC_RH_INAT_MBR
4. Recolha de motivação semanal	MQ_LABSYNC_RH_MOTIVAÇÃO
5. Processo de manutenção e organização da estrutura global da equipa	MQ_LABSYNC_RH_ORGANIZAÇÃO



### 3.2.1 - MQ\_LABSYNC\_HR\_PARTICIPAÇÃO

Processo que interpreta e qualifica a participação e empenho de cada membro na respectiva semana.

#### *Entry Criteria:*

Após a aula PL, todas as semanas.

#### *Validação:*

- Assinatura na folha de presenças da aula PL.
- Assiduidade nas reuniões externas à aula PL.
- Confirmação do preenchimento dos formulários semanais.
- Confirmação das tarefas concluídas por cada membro de cada subunidade na plataforma Trello.
- Participação nos canais de comunicação.

#### *Exit Criteria:*

Termina quando todos os dados estão validados ou acaba a semana (antes da aula PL).

#### *Métricas:*

- 1) Junção e interpretação destes dados em gráficos e tabelas de modo a avaliar a progressão da participação durante o decorrer do projeto.
- 2) Número de casos de risco comunicados à gestão de projeto.

### 3.2.2 - MQ\_LABSYNC\_HR\_RECON\_ATR\_TRF

Processo que analisa as tarefas semanais e os seus responsáveis e que se encarrega de distribuir essas tarefas da maneira mais uniforme possível por todos os membros da equipa.

#### *Entry Criteria:*

Quando, por alguma razão, existem tarefas sem responsável depois do prazo de atribuição.

#### *Validação:*

- Publicação das tarefas planeadas dentro de cada subunidade para cada membro se responsabilizar, pelo menos, por uma tarefa.
- Atribuição de responsáveis a tarefas pelo coordenador de RH (Guilherme Pontes) caso o prazo de atribuição pelo respetivos membros tenha expirado e se encontrarem membros sem trabalhos a realizar.

#### *Exit Criteria:*

Este processo é dado como concluído quando todos os membros da equipa possuírem tarefas a realizar na semana de trabalho a iniciar.

#### *Métricas:*

- 1) As tarefas a realizar são publicadas na plataforma Trello a partir das 15h de quarta-feira.
- 2) Os membros tornam-se responsáveis pelas respetivas tarefas da sua subunidade até às 21:30h.
- 3) Caso existam tarefas ainda sem responsáveis, o coordenador de RH atribui-as de forma mais uniforme possível aos membros, de modo a que todos tenham trabalhos atribuídos pelas 23h de quarta-feira.

### 3.2.3 - MQ\_LABSYNC\_HR\_INAT\_MBR

Análise do Processo 1 (MQ\_LABSYNC\_RH\_PARTICIPAÇÃO) de forma a identificar membros inativos.

#### *Entry Criteria:*

Assim que um membro fica duas semanas sem comparecer na aula PL, sem tarefas dadas como concluídas na plataforma Trello, sem participar na plataforma de comunicação Discord, ou sem responder ao formulário de Gestão de Riscos e RH e ao formulário de esforço semanal da Gestão de Projeto.

#### *Validação:*

- Inatividade total ou atividade insuficiente de um membro durante 2 semanas.
- Reconhecimento da situação por parte da Gestora de Projeto e/ou RH.
- Declaração do membro como inativo.
- Comunicação com o membro em questão, de modo a solucionar o problema e evitar aumento dos riscos.
- Resolução do problema num espaço de 24h.

#### *Exit Criteria:*

Processo dado como concluído quando o membro volta à atividade ou quando são alocadas as suas funções a outro membro, ficando o membro inativo fora da equipa.

#### *Métricas:*

- 1) Inatividade ou atividade abaixo dos 85% por parte do membro em questão durante 2 semanas consecutivas.
- 2) Declaração do membro como inativo.
- 3) Comunicação com esse membro de modo a perceber e solucionar o problema.
- 4) Problema resolvido e comunicado à gestão de projeto num prazo de 24h.

### 3.2.4 - MQ\_LABSYNC\_HR\_MOTIVAÇÃO

Processo que analisa o empenho e bem-estar de cada membro de modo a facilitar o trabalho de equipa em prol dos objetivos comuns.

#### *Entry Criteria:*

Após a aula PL.

#### *Validação:*

- É criado e partilhado um formulário (Google Forms) após aula PL juntamente com o Coordenador de Riscos.
- O formulário procura interpretar a satisfação individual de cada membro para com a equipa e sua respectiva subunidade e para com a carga de trabalho imposta.
- Serve também para os coordenadores (Gestão de Riscos e HR) tomarem conhecimento de situações conflituosas, ou de falhas de comunicação, empenho, entre outras, a serem reportadas pelos membros.
- Validado assim que todos tiverem respondido.

#### *Exit Criteria:*

Até às 22h do dia anterior à aula PL ou assim que todos os membros preencham o formulário.

#### *Métricas:*

- 1) Registo da satisfação geral dos membros.
- 2) Percentagem de problemas reportados pelos membros e problemas resolvidos.
- 3) Percentagem de satisfação dentro da respectiva subunidade.

### 3.2.5 - MQ\_LABSYNC\_HR\_ORGANIZAÇÃO

São usadas as métricas do processo anterior de modo a gerir os membros e cargos de toda a equipa, consoante os problemas e necessidades que vão surgindo.

#### *Entry Criteria:*

Situações de risco avaliadas através dos formulários, críticas ou denúncias de membros, situações que obrigam à mudança da disposição dos membros pelas várias equipas.

#### *Validação:*

- Assim que é analisada uma situação de risco (subunidade com carga de trabalho excessiva, tarefas que não estão a ser concluídas, membros que desejam mudar de cargo e denúncias ou reclamações reportadas aos gestores) há uma comunicação com todas as partes envolvidas de modo a perceber o ponto de vista de cada um e a situação de um modo objetivo.
- Resolução do problema tentando preservar a satisfação de todos os envolvidos e acima de tudo, beneficiando o rumo da equipa.

#### *Exit Criteria:*

Assim que os riscos estejam eliminados e/ou os membros satisfeitos.

#### *Métricas:*

- 1) Tempo de resolução e bem-estar dos membros envolvidos.

### 3.3. - Riscos [RSK]

<b>Unidade:</b> RSK	<b>Data:</b> 15.11.2018	<b>Gestor:</b> Ricardo Sintra Tavares
<b>Versão:</b> 2.0	<b>Status:</b> Baselined	

#### Lista de membros desta unidade

Ricardo Sintra Tavares	LEI	2014230130@student.uc.pt
------------------------	-----	--------------------------

#### Objetivos da Unidade de GR:

- 1) Antecipar e identificar o conjunto de riscos de negócio que poderão afetar o projeto.
- 2) Priorizar e monitorizar os riscos.
- 3) Utilizar as políticas, competências e responsabilidades para executar esta abordagem com sucesso.
- 4) Alertar para a presença destes riscos transmitindo os resultados preliminares aos membros da equipa.
- 5) Inverter as tendências de risco, mitigando ou resolvendo riscos existentes.

#### Processos utilizados por esta unidade:

Processo	ID do Processo
1. Identificação de um risco	MQ_LABSYNC_GR_IDENTIFICAR
2. Análise e classificação de um risco	MQ_LABSYNC_GR_AVALIAR
3. Mitigar efeitos /Combater riscos possíveis	MQ_LABSYNC_GR_RESOLVER

### **3.3.1. - MQ\_LABSYNC\_RSK\_IDENTIFICAR**

#### **Processo de Identificação de um risco**

Este processo descreve a base de trabalho da Unidade de Gestão de Riscos.

Os projetos de Engenharia de Software consistem num vasto número de pessoas a trabalhar em conjunto dentro da sua unidade em simultâneo com as outras dentro do projeto global. Esta realidade torna o projeto inevitavelmente arriscado pela variedade de problemas possíveis que podem surgir, de natureza humana ou não.

Como tal, esta unidade tem a responsabilidade de identificar todos os possíveis problemas que afetem o decorrer do projeto. A identificação de risco é um processo iterativo, à medida que o trabalho decorre mais informação é ganha e o processo de riscos tem de ser ajustado correspondentemente.

#### *Inputs:*

Metas do projeto, métricas de satisfação, “clima” global nas apresentações semanais, presenças nas aulas, feedback dado pelos membros da equipa, comunicações no Discord, resultados dos questionários.

#### *Entry Criteria:*

O processo é ativado quando ocorrem alterações num dos elementos de entrada, que possam indicar possíveis riscos. Neste caso o risco tem de ser validado pelo Coordenador de Equipa a que esse risco pertence ou pelo Coordenador de Projeto caso seja um risco mais geral.

#### *Task Description:*

Começa-se por perceber e apurar os detalhes do possível risco. Organiza-se o risco em relação à sua área de impacto e é transmitido ao(s) responsável(eis) daquela(s) área(s). Todos os detalhes do risco (descrição, consequências, etc) devem ser apresentados na apresentação semanal, assim como no Google Drive da equipa.

#### *Exit Criteria:*

O processo conclui quando o risco for identificado e descrito e a sua probabilidade de ocorrer diminuir após a sua análise ficar de acordo com todos os membros relacionados com o risco.

#### **Métricas de avaliação deste processo:**

- 1) Métricas de satisfação.
- 2) Número de riscos identificados.
- 3) Número de riscos que geraram problemas.

#### *Outputs gerados:*

Apresentação semanal sobre esta unidade, com os riscos identificados.

### 3.3.2. - MQ\_LABSYNC\_RSK\_AVALIAR

#### Processo de mitigar efeitos /combater riscos possíveis

Neste processo, com o risco já identificado, é necessário avaliá-lo, analisando a probabilidade do risco e o seu impacto.

#### Inputs:

Um novo risco, ou o aumento de probabilidade de ocorrência de um já identificado previamente.

#### Entry Criteria:

O processo é ativado aparecendo um risco claro no questionário (por implementar), através de identificação por parte de algum membro, ou por observação do Coordenador de Riscos.

#### Task Description:

O risco é avaliado, tendo em conta o impacto que pode ter para o projeto e categorizado com recurso a uma matriz de avaliação de risco:

Risk Assessment Matrix				
Impact of Risk (Consequence)	Major	Medium	High	Extreme
	Moderate	Medium	Medium	High
	Minor	Low	Medium	Medium
Seriousness of Risk = Probability x Impact		Unlikely (0-33%)	Moderately Likely (33%-66%)	Highly Likely (66%-100%)
		Probability of Risk (Likelihood)		

#### Bibliografia:

<http://www.theprojectmanagementblueprint.com/?p=277>

Após a sua classificação esta é discutida com os membros que podem vir a ser afetados pelo risco até se chegar a um acordo sobre o seu grau de risco. Concluída a avaliação, o nível desta é comunicado ao representante da unidade (ou unidades) em causa e à Gestora do Projeto.



#### *Exit Criteria:*

O processo conclui quando a avaliação do grau de severidade do risco estiver definida.

#### Métricas de avaliação deste processo:

1. Número de riscos identificados.
2. Número de riscos avaliados.
3. Média de avaliação.

#### *Outputs gerados:*

Um documento com a descrição do risco e a sua avaliação, guardado no Google Drive.

### **3.3.3. - MQ\_LABSYNC\_RSK\_RESOLVER**

Processo responsável pela extinção/redução do risco. Um risco encontra-se extinto quando a sua probabilidade de ocorrer diminui para valores insignificativos.

#### *Inputs:*

Riscos identificados e avaliados anteriormente.

#### *Entry Criteria:*

O processo é ativado sempre que surge um novo risco ou a avaliação de um existente aumente.

#### *Task Description:*

O Coordenador de Riscos notifica via Discord a ocorrência do risco. De seguida fala com a Gestora do Projeto e ao(s) coordenador(es) da(s) equipa(s) envolvida(s) de forma a chegar a um consenso de como resolver o risco. Feita a discussão, aplica-se a solução encontrada.

#### *Exit Criteria:*

O processo conclui após a discussão do risco com as entidades envolvidas e aplicação da solução encontrada sobre o mesmo.

#### Métricas de avaliação deste processo:

- 1) Métricas de satisfação.
- 2) Número de riscos identificados.
- 3) Número de riscos extintos ou mitigados.

#### *Outputs gerados:*

Solução encontrada para resolver o risco.

### 3.4. - Requisitos [REQ]

<b>Unidade:</b> REQ	<b>Data:</b> 18.11.2018	<b>Gestor:</b> Eduardo Guerra
<b>Versão:</b> 3.0	<b>Status:</b> Baselined	<b>Vice Gestora:</b> Carolina Batista

#### Lista de Membros da unidade:

Eduardo Guerra	LEI	Coordenador	uc2016231080@student.uc.pt
Carolina Batista	LDM	Vice-Coordenadora	uc2016240419@student.uc.pt
Bruno Simões	LDM	Colaborador	uc2016238684@student.uc.pt
Bruna Rosas	LDM	Colaboradora	uc2016239564@student.uc.pt
Joana Mateus	LDM	Colaboradora	uc2016225749@student.uc.pt

#### Trabalho desenvolvido pela Equipa de Requisitos:

A equipa de requisitos é constituída por um grupo de elementos cujo trabalho é produzir mockups, tabela de requisitos e o documento final de Software Requirement Specifications (SRS). Para isso são elaborados uma série de processos, métodos de trabalho que visam o desenvolvimento de um projeto contínuo e bem-sucedido, pretendendo repartir o esforço igualmente por cada membro da unidade.

#### Objetivos da Unidade de REQ:

1. A Equipa de Requisitos trabalha com o foco de corresponder às exigências do cliente, realizando uma tabela de requisitos que tem de estar em sintonia com os *mockups* que até serem validados permanecem em constante melhoramento. É um trabalho que tem de ser acompanhado pela unidade de implementação, pois serve-lhes de inputs e pode sofrer alterações consoante as dificuldades ou escolhas de implementação.
2. Faz a ponte entre o cliente e a Unidade de Implementação, sendo que o trabalho desenvolvido inicialmente foi em conjunto com o desta unidade. Os *outputs* desta unidade servem como critérios de entrada e de matérias primas para esta unidade e a de testes. Tem ainda o papel de investigar o melhor *design* possível a implementar e o melhor nível de usabilidade, para que o utilizador consiga perceber facilmente o funcionamento da plataforma e acima de tudo, usá-la.

## Processos utilizados por esta unidade:

Processo	ID do Processo
1. Criação da tabela de requisitos.	MQ_LABSYNC_REQ_TABELA_REQ
2. Desenvolvimento de mockups.	MQ_LABSYNC_REQ MOCKUPS
3. Construção do SRS.	MQ_LABSYNC_REQ_SRS

### 3.4.1.- MQ\_LABSYNC\_REQ\_TABELA\_REQ

#### Processo de criação da tabela de requisitos

##### Membros e as suas funções:

###### Responsável pela elaboração da tabela de requisitos:

Eduardo Guerra (LEI), coordenador de requisitos.

###### Responsáveis pela validação:

Fábio Barata (LDM), colaborador de qualidade,

Fernando Felício (LDM), coordenador de implementação.

##### *Inputs:*

Apontamentos da reunião com o cliente, datada a 2 de novembro.

##### *Entry Criteria:*

É através dos *inputs* dados pelo cliente, Mário Alberto da Costa Zenha Relá, que é possível definir quais os requisitos necessários à resolução do problema, sendo ainda possível dar o nosso ponto de vista de forma a trabalhar em conjunto e de encontrar a melhor solução.

Utilizamos depois estes *inputs* para construir o documento de requisitos com o objetivo de fornecer a informação necessária às unidades de Implementação e de Testes para que possam incluir e testar a informação presente no artefato gerado, servindo assim como critério de ativação deste processo.

## Descrição das tarefas:

O coordenador desta unidade deve reunir-se com o cliente apontando todos os aspetos importantes sobre o projeto.

O responsável deverá realizar a tabela no Microsoft Office Word logo que disponha de grande parte da informação. No início da tabela deverão estar descritas as causas que levaram à alteração da versão anterior da tabela, nomeadamente que requisitos foram alterados/removidos e erros de formatação. A cada requisito deverá estar associado o *sprint* e o número.

**Validação:** Entrega da tabela aos responsáveis para que a possam validar. Submissão no Discord e na pasta de Requisitos no GitHub-

## Monitorização deste processo pela equipa de qualidade:

A equipa de qualidade fica a par do documento de requisitos a partir do momento em que o recebe via *Discord*. Caso haja alguma dúvida, falha de formato ou falta de compreensão aquando do tratamento do documento, a unidade de requisitos será alertada.

## Validação:

Correção de erros ortográficos, legibilidade, coerência. O coordenador de Implementação, consoante as capacidades da unidade e os pedidos do cliente, aprova ou rejeita a validação deste documento. A linguagem tem de ser coerente e os requisitos deverão ser possíveis de implementar. Se isto tudo se verificar, torna-se uma tabela válida.

Caso haja alguma falha na tabela de requisitos, é preenchido um Google Form, guardado na Drive e cujo link está disponível no canal “*Links* Importantes” do Discord. As respostas deste formulário são usadas para corrigir as falhas do documento e servem de evidências de inspeção.

Sempre que uma validação falha, a tabela é reformulada, atualizando a sua versão.

## *Exit Criteria:*

O conteúdo deve ser legível para toda a equipa. Após validação feita pelo cliente, o artefato está pronto a ser transmitido à Unidade de Implementação para que possam implementar o projeto da melhor forma possível.

## Métricas de avaliação deste processo:

Métricas anotadas semanalmente e possíveis de consultar nos slides semanais;

- Número total de horas despendidas pelos membros que realizaram a tabela;
- Número de horas despendidas por cada membro;
- Número de membros atribuídos a cada tarefa;
- Se as tarefas atribuídas aos respectivos membros foram concluídas;
- Número de requisitos de versão para versão;
- Número de requisitos adicionados, alterados e/ou eliminados.

### *Outputs:*

Tabela de requisitos presente na pasta de Requisitos no GitHub.

### *LifeCycle:*

Versões diferentes todas as semanas após breves conversações com membros desta unidade e de implementação. No total houve dez versões principais. As duas primeiras antes da reunião com o cliente, sete após da reunião e a versão final após o projeto ter sido dado como concluído.

**Início a 30 de outubro de 2018;**

**Fim a 16 de novembro de 2018;**

### Monitorização deste processo pela equipa de qualidade:

A Equipa de Qualidade fica a par do documento a partir do momento em que é colocado no Discord e no GitHub. Todos os artefatos necessários à realização da tabela em questão encontram-se visíveis para a Unidade de Qualidade permitindo que o seu trabalho de validação seja concluído da melhor forma. Alguma dúvida ou falta de compreensão é esclarecida de imediato.

## **3.4.2.- MQ\_LABSYNC\_REQ MOCKUPS**

Este processo tem como objetivo a realização de mockups que vão formar o aspeto visual da plataforma.

### *Inputs:*

O artefato gerado pelo processo anterior, ou seja, a tabela de requisitos é utilizada como input para iniciar este processo.

### *Entry Criteria:*

Para iniciar a realização dos *mockups* é indispensável que o artefato anterior (tabela de requisitos) esteja devidamente validado. Esta validação é conseguida através do esforço de várias entidades como o cliente, Unidades de Requisitos, Implementação, Testes e Qualidade.

As razões pelas quais a validação da tabela de requisitos é tão importante para o desenvolvimento deste processo, são os seguintes:

- Como a equipa de requisitos tem um contacto direto com o cliente, deve analisar o documento para verificar se todos os requisitos estão em sintonia com os objetivos do cliente, prevenindo eventuais requisitos desenvolvidos ou adicionados.
- Como a tabela de requisitos vai ser lida por várias pessoas, a Equipa de Qualidade tem o papel de corrigir incoerências gramaticais e elegibilidade, para que as

mensagens sejam corretamente transmitidas. É importante que o autor da tabela de requisitos transmita a mensagem adequada aos autores dos mockups. Assim todos os documentos se sintonizam. Também é importante que as outras unidades percebam o pretendido, para que não haja requisitos devolvidos.

- A tabela de requisitos tem de estar em conformidade com os objetivos do cliente, para que o cliente não seja surpreendido com mockups que não satisfaçam esses objetivos.

- A Unidade de Implementação é relevante para perceber se é possível implementar determinado requisito. Assim, apenas é necessário adaptar a tabela de requisitos, e não os *mockups*.

- A Unidade de Testes também pode sugerir alterações na tabela de requisitos, prevenindo erros que possam surgir quando estiverem a testar e tornando desnecessário a realização de *mockups* ou a criação de páginas que não irão ser utilizadas.

Todos os pontos referidos anteriormente têm como consequências sucessivas alterações e atrasos no desenvolvimento de *mockups*, por isso a tabela de requisitos deve estar devidamente validada antes de se iniciar este processo.

### Descrição das tarefas:

- São atribuídas tarefas para a realização de *wireframes* (rascunhos dos *mockups* de alta fidelidade).

- Os *mockups* são feitos com base nos rascunhos e na tabela de requisitos.

- Verificação por parte da Unidade de Requisitos dos mockups, para perceber se esses elementos estão de acordo com a tabela de requisitos.

- Reunião com o cliente para perceber se os *mockups* e os seus objetivos condizem.

- Colocação dos *mockups* no Discord, no canal de requisitos-implementação para que a equipa de Implementação os possa validar.

- Colocação dos documentos finais no GitHub para que outras Unidades como Qualidade e Testes possam validar.

### Validação:

Os *mockups*, para serem validados, têm de ser legíveis e adequados à plataforma que os vai suportar, verificando também se não há impossibilidades de implementação, se a sua usabilidade é adequada e se cumpre os requisitos.

As validações dos mockups são feitos pelas Unidades de Requisitos, Implementação, de Qualidade e pelo cliente.

Requisitos verifica se os *mockups* e a tabela de requisitos estão em sintonia, observam se os *mockups* estão coerentes graficamente e se não falta nenhuma funcionalidade.

Implementação verifica se tem possibilidade de implementar e sugere soluções, com o objetivo de facilitar o seu trabalho.

Qualidade verifica se não existem erros ortográficos ou incoerências.

O cliente verifica se os seus objetivos foram cumpridos, sugere alterações com o intuito de melhorar o *design*, a usabilidade e para facilitar o trabalho a implementação.

Caso tenham sido encontradas falhas, a Equipa de Requisitos, deve validar de novo os documentos corrigidos.

### *Exit Criteria:*

Assim que cada um dos mockups e o diagrama de navegação são validados como descrito acima, o processo dá-se como finalizado. Os mockups são enviados à Unidade de Implementação para que possam elaborar o front-end e para que entendam o fluxo de navegação da plataforma. Deverão ainda seguir para a Equipa de Testes de maneira a que possam testar cada um dos requisitos.

### Métricas de avaliação deste processo:

Métricas anotadas semanalmente:

- Número de pessoas envolvidas em cada tarefa;
- Número de *mockups* (novos, finalizados, alterados, total)

### *Outputs:*

Dois documentos: conjunto de mockups.

### *Lifecycle:*

#### **Mockups:**

**Versão 1** - *Mockups* terminados a 25-09-2018, após primeira reunião com o cliente em conjunto com a unidade de requisitos.

**Versão 2** - *Mockups* terminados a 28-09-2018 com as alterações pedidas pelo cliente.

**Versão 3** - *Mockups* terminados a 15-10-2018 com os *mockups* da *sprint 2*.

**Versão 4** - *Mockups* terminados a 28-10-2018 após implementação pedir *mockups* mais específicos e *mockups* de páginas em falta.

**Versão 5** - *Mockups* terminados em 10-11-2018 com o redesign dos *mockups* , após reunião com cliente no dia 09-11-18.

**Versão 6** - *Mockups* terminados em 12-11-2018 com os objetivos de sprint 3 quase estabilizados.

**Versão 7** - *Mockups* terminados a 13-11-2018, após remoção de fotografias dos utilizadores nos posts (versão final dos *mockups* da *sprint 3*).

### Monitorização deste processo pela equipa de qualidade:

Assim que os *mockups* e o diagrama de navegação são concluídos, estes são colocados no GitHub para que a equipa de Qualidade os possa validar.

### 3.4.3.- MQ\_LABSYNC\_REQ\_SRS

Este é o documento que contém grande volume de trabalho da unidade de requisitos, ou seja contém os *mockups* com as respectivas legendas, o diagrama de navegação, fluxo de interação e os casos de uso.

#### *Inputs:*

Os artefatos gerados pelo processo anterior, ou seja, os documentos da tabela de requisitos e dos *mockups* foram usados como *input* para iniciar este processo.

#### *Entry Criteria:*

Para iniciar este processo é de extrema importância que a tabela de requisitos e as *mockups* estejam devidamente validados, já que vão servir de base na construção do documento ou farão parte do mesmo.

#### *Task Description:*

- São atribuídas tarefas para cada membro da Unidade de Requisitos.
- Verificação da validação dos inputs do SRS.
- Organização do documento (junção dos *mockups*, diagrama de navegação e criação de coerência visual do SRS)
- As tabelas de casos de usos são feitas com base na tabela de requisitos, *mockups* e diagrama.
- Colocação do SRS no Discord e no GitHub para posterior validação.

#### *Validação:*

A função de validação cabe às Unidades de Requisitos, Qualidade e Implementação. É necessário que o documento esteja coerente gramaticalmente e visualmente: paginado, intitulado, legendado, deve ter índice, tabela de versões, autores do documento e nome da equipa (Labsync). A unidade de qualidade analisa os fatores linguísticos e de estrutura, enquanto que requisitos e implementação centra a sua análise no conteúdo. Como neste documento consta toda a informação necessária para que a plataforma seja implementada, é de extrema importância que a Unidade de Implementação valide o SRS.



### *Exit Criteria:*

Assim que todos os dados do SRS estejam estabilizados e validados, o processo é concluído. Este documento vai servir essencialmente à Unidade de Implementação que terá as bases para realizar o seu trabalho, mas também poderá ser útil para outras unidades que terão num único documento as principais informações da unidade de requisitos.

### Métricas de avaliação deste processo:

Métricas anotadas semanalmente:

- Número de pessoas envolvidas em cada tarefa;

### Outputs:

O documento de requisitos.

### *Lifecycle:*

Diagrama de navegação:

**Versão 1** - Diagrama terminado a 04-10-2018, em sintonia com os objetivos da sprint 1.

**Versão 2** - Diagrama terminado a 11-10-2018 com os objetivos da sprint 2.

**Versão 3** - Diagrama terminado a 15-10-2018 com acréscimo de notas para que implementação percebesse melhor o fluxo de navegação.

**Versão 4** - Diagrama terminado a 6-11-2018 de acordo com a tabela de requisitos da sprint #3.

SRS:

**Versão 1** - SRS terminado a 04-10-2018.

**Versão 2** - SRS terminado a 24-10-2018, após adição dos *mockups* da *sprint 2* com as respetivas tabelas de caso de uso e atualização do diagrama de navegação.

**Versão 3** - SRS terminado a 28-10-2018, após implementação fazer pedido de novos *mockups* e aperfeiçoamento dos antigos. Portanto, as alterações do SRS centraram-se na substituição de *mockups* e adição de novos.

**Versão 4** - SRS terminado a 11-11-201, com a adição dos *mockups*, diagrama de navegação e tabelas de caso de uso da *sprint 3*.

### Monitorização deste processo pela equipa de qualidade:

Assim que o SRS foi finalizado, foi colocado no GitHub para que a equipa de Qualidade o pudesse validar.

### 3.5. - Implementação [IMP]

<b>Unidade:</b> IMP	<b>Data:</b> 28.10.2018	<b>Gestor:</b> Fernando Felício
<b>Versão:</b> 2.1	<b>Status:</b> Baseline	<b>Vice Gestor:</b> Francisco Barão

#### Lista de membros da equipa:

Fernando Felício	Coordenador	LDM	fmsfelicio@hotmail.com
Francisco Barão	Vice-Coordenador	LEI	s.franciscobarao@gmail.com
Pedro Alexandre	Colaborador	LEI	pedro.alex.ribeiro@hotmail.com
Miguel Gonzaga	Colaborador	LEI	mike.gonzagal 998@gmail.com
Pedro Mendonça	Colaborador	LEI	pedro.mendonça 1@hotmail.com

#### Processos utilizados por esta unidade:

<b>Processo</b>	<b>ID do Processo</b>
1 - Processo de iniciação de novo Sprint	MQ_LABSYNC_IMP_SPRINT
2 - Processo identificação de artefatos	MQ_LABSYNC_IMP_ID
3 - Processo de desenvolvimento	MQ_LABSYNC_IMP_DEV
4 - Processo de correção	MQ_LABSYNC_IMP_FIX

### 3.5.1. - MQ\_LABSYNC\_IMP\_SPRINT

#### Processo de iniciação de nova Sprint

A divisão do projeto da unidade está, por pedido do cliente, organizada por sprints, logo após cada novo sprint é necessário que haja uma organização do mesmo para que o desenvolvimento esteja adaptado aos obstáculos que o mesmo impõem.

#### *Entry Criteria:*

A cada novo sprint é esperado que os artefatos externos à subunidade, provenientes de Requisitos e colocados no GitHub e/ou Discord, sofram adições e/ou alterações. Cabe ao coordenador de IMP verificar se essas alterações existem para se dar início a este processo.

#### *Task Description:*

Após confirmadas as mudanças, notificadas via Discord e acessíveis através do repositório do GitHub e/ou Google Drive da unidade aos artefatos de Requisitos, esses mesmos são revistos internamente (IMP) para que todas as dúvidas dos developers sejam esclarecidas. Caso essas dúvidas existam será enviado via Discord para Requisitos o documento original assim como um anexo que explicita todas as dúvidas existentes sejam estas em termos de features a implementar ou alterações visuais (*mockups*).

Após a chegada do artefato de Requisitos, todos os membros da subunidade têm até um prazo máximo de 48 horas para visualizar e expor as dúvidas em relação ao artefato em questão, notificando a subunidade via Discord. Este prazo é encurtado para 24 horas quando se trata de um artefato que está a sofrer uma segunda alteração por parte de Requisitos.

#### *Validation:*

Após a chegada do artefato de Requisitos, este deve ser visualizado por todos os developers e após conclusão dessa tarefa, deve ser comunicado, via Discord (channel de IMP), que o documento foi examinado e eventuais dúvidas se estas existirem.

#### *Exit Criteria:*

Após a validação interna por parte de todos os membros da subunidade, o processo dá-se por terminado visto que se encontram reunidas as condições para iniciar o desenvolvimento.

#### Métricas:

- Percentagem de visualizações/feedback por membro de IMP  
Valor esperado: 100%.

### 3.5.2. - MQ\_LABSYNC\_IMP\_ID

#### Processo de identificação de artefatos

Para que a unidade trabalhe de forma eficiente e organizada, é necessário que haja uma organização de artefatos produzidos, via a sua identificação, seja no que toca a folders, diretorias ou o nome desses mesmos artefatos.

#### *Entry Criteria:*

É produzido um artefato (documento ou código) que precisa de ser catalogado, identificado e validado.

#### *Task Description:*

O(s) autor(es) do artefato identifica-o e informa o team-leader via Discord da localização do ficheiro.

O coordenador de IMP verifica o artefato, que é depois redirecionado para Testes e para Qualidade. O nome do ficheiro deve ser validado por estas entidades.

#### *Validation:*

O nome do artefato é intuitivo, e está agrupado com artefatos semelhantes em tema ou componente.

#### *Exit Criteria:*

O artefato está corretamente identificado, validado e encontra-se na diretoria correta.

#### Métricas:

- Percentagem de artefatos devolvidos por Testes e pela Qualidade;  
Valor máximo esperado: 20%.

### 3.5.3. - MQ\_LABSYNC\_IMP\_DEV

#### Processo de desenvolvimento

Este processo é o core da subunidade e para tal é necessário que haja uma distribuição homogênea, pensada e equilibrada das tarefas, que estejam atribuídas aos membros mais indicados da subunidade e por fim que sejam criadas deadlines que respeitem e mantenham o flow do trabalho no sistema de pipeline, não sendo demasiado ambiciosas.

#### *Entry Criteria:*

O processo de desenvolvimento pode ser iniciado após a conclusão do processo de iniciação da sprint (MQ\_LABSYNC\_IMP \_SPRINT), logo existe código a ser desenvolvido que é necessário guardar no GitHub.

#### *Task Description:*

A subunidade de IMP está por nós organizada, em duas componentes: a de Back-End e Front-End e após a análise feita ao artefato de Requisitos e ao que este mesmo contém é feita uma reunião, presencial ou remota via Discord para que sejam distribuídos membros da subunidade pelas duas componentes. Após este passo é necessário que o trabalho seja dividido em tarefas e que estas sejam distribuídas por membro(s) adequados da subunidade via Trello, isto inclui a criação de uma deadline, na qual a tarefa deve estar concluída. As tarefas depois de distribuídas e organizadas são distribuídas em branches tendo como principal o branch "development". Esta é a principal etapa de produção de código.

#### *Validation:*

Todos os elementos da subunidade devem ter, no mínimo, uma atividade atribuída por semana.

As tarefas têm de estar concluídas no prazo máximo de até 24 horas depois do prazo delimitado no Trello (24h após deadline abrange imprevistos).

Após a conclusão de cada tarefa, os artefatos referentes devem ser colocados no GitHub, assim como a subunidade de IMP deve ser notificada via Discord.

Todos os ficheiros criados devem obedecer ao processo de identificação de artefatos (MQ\_LABSYNC\_IMP \_ID) e conter informação sobre a sua autoria.

#### *Exit Criteria:*

O artefato produzido está funcional e pronto para ser corretamente identificado e colocado no GitHub para consequentemente ser testado pela subunidade externa de Testes.

### Métricas:

- 1) Percentagem de Testes com resultados negativos.  
Valor máximo esperado: 25%.
- 2) Carga de esforço semanal por membro da subunidade.  
Valor médio esperado: 04:00H.
- 3) Número de commits por semana.

## 3.5.4. - MQ\_LABSYNC\_IMP\_FIX

### Processo de correção de defeitos

Na produção de código não é esperado, mas é provável que existam defeitos que não são detectados internamente e só surgem quando o código é sujeito aos testes específicos da subunidade externa de testes. Se estes erros forem detectados, é necessário iniciar a sua correção/alteração do código.

#### *Entry Criteria:*

A subunidade externa de Testes notifica o coordenador de IMP, via Discord, que defeitos foram encontrados e em que artefatos.

#### *Task Description:*

Após receber a informação, via Discord, proveniente da subunidade de Testes, o coordenador de IMP identifica o(s) autore(s) do artefato que contém erros e volta a abrir uma tarefa no Trello, com uma deadline para que esse erro seja corrigido atempadamente e corretamente.

#### *Validation:*

As tarefas têm de estar concluídas no prazo máximo de até 24 horas depois do prazo delimitado no Trello (24h após deadline abrange imprevistos).

Após conclusão de cada tarefa, os artefatos referentes devem ser colocados no GitHub, assim como a subunidade de IMP deve ser notificada via Discord.

Todos os ficheiros criados devem obedecer ao processo de identificação de artefatos (MQ\_LABSYNC\_IMP\_ID).

#### *Exit Criteria:*

O artefato produzido está funcional e pronto para ser corretamente identificado e colocado no GitHub para ser novamente testado pela subunidade externa de Testes.

### Métricas:

- Percentagem de Testes(repetidos) com resultados negativos;  
Valor máximo esperado: 10% por repetição (valor inicial de 20% e >=0).

### 3.6. - Testes [TST]

<b>Unidade:</b> TST	<b>Data:</b> 16.11.2018	<b>Gestor:</b> Tomás Lopes
<b>Versão:</b> 2.0	<b>Status:</b> Baseline	<b>Vice Gestor:</b> Ricardo Gonçalves

#### Lista de membros desta unidade:

Tomás Lopes	LEI	Coordenador	tomasgabriel1998@gmail.com
Ricardo Gonçalves	LDM	Vice-Coordenador	rgoncalvestlm@gmail.com
João Campos	LDM	Colaborador	jgmmcampos@gmail.com
André Correia	LEI	Colaborador	andrefcorreia88@gmail.com
Guilherme Nogueira	LEI	Colaborador	gnogueira@student.dei.uc.pt

#### Processos utilizados por esta unidade:

Processo	ID do Processo
1. Processo para escrita de testes.	MQ_LABSYNC_TST_ESCRITA_TST
2. Processo para averiguar se requisitos estão a ser cumpridos.	MQ_LABSYNC_TST_AVR_REQ&COD
3. Processo para testar código.	MQ_LABSYNC_TST_CÓDIGO
4. Processo para deployment.	MQ_LABSYNC_TST_DEPLOYMENT

#### Objetivos da unidade de TST:

- 1) 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.
- 2) Garantir a qualidade e o bom funcionamento do código implementado.
- 3) Fazer deployment do programa.

### 3.6.1. - MQ\_LABSYNC\_TST\_ESCRITA\_TST

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 contêm 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 enviada 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 servem para averiguar se os requisitos não-funcionais estão a ser cumpridos, podendo estes 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.
- 4) 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

### *Entry Criteria:*

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

### *Task Description:*

São distribuídas tarefas pelos vários elementos da unidade. Cada elemento terá que usar, como base, os testes nos documentos “Testes funcionais” e “Testes não-funcionais”.

### *Exit Criteria:*

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.

### *Documentos:*

Todos os documentos estão no Git dentro da pasta de testes discriminado para cada Sprint. Estão lá presentes os documentos template para que servem de guia para os testes (em formato word) e posteriormente estão presentes os documentos de resultados (em formato PDF).

### 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.

### 3.6.2. - MQ\_LABSYNC\_TST\_AVR\_REQ&COD

#### Processo para averiguar se requisitos estão a ser cumpridos

É feito uma série de testes a partir da lista de requisitos. No caso dos requisitos funcionais, estes são testados para averiguar se foram 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.

Este processo verifica também o trabalho realizado pela equipa de implementação, ou seja, que satisfaça as normas de programação e interação com um utilizador (não permitindo a inserção de valores indesejados, por exemplo: uma expressão num campo numérico. Em suma, testa o código implementado.

#### *Inputs:*

- 1) O documento de requisitos enviado pela equipa de testes.
- 2) Versão do programa a testar.
- 3) O código realizado pela equipa de implementação que terá sido disponibilizado no GitHub, pelo coordenador da unidade de implementação.

#### *Output:*

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

#### *Entry Criteria:*

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 uma 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.

### *Task Description:*

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. Ao testar o programa são também usados documentos 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 (via Trello) 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.

### *Exit Criteria:*

O processo é dado como concluído quando cada elemento tenha realizado vários testes para cada requisito e este cumpra todos os requisitos, sendo posteriormente atualizado o projeto principal. Caso contrário é enviada uma mensagem à equipa de implementação para a sua correção.

### *Métricas de avaliação deste processo:*

- 1) Tabela com requisitos aprovados ou reprovados;
- 2) Número de testes realizados por requisitos;
- 3) Número e taxa de testes aprovados/reprovados por requisito;
- 4) Número de pessoas envolvidas em cada teste;
- 5) 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” e também os 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.

#### **3.6.4. - MQ\_LABSYNC\_TST\_DEPLOYMENT**

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.

##### *Entry Criteria:*

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.

### 3.7. - Qualidade [QUA]

<b>Unidade:</b> QUA	<b>Data:</b> 14.11.2018	<b>Gestor:</b> Maximilian Rubin
<b>Versão:</b> 2.0	<b>Status:</b> Baselined	<b>Vice Gestor:</b> Diogo Pereira

#### Lista de membros desta unidade:

Maximilian Rubin	LDM	Coordenador	uc2016231987@student.uc.pt
Diogo Pereira	LDM	Vice-coordenador	uc2016217862@student.uc.pt
Diogo Gonçalves	LDM	Colaborador	uc2016217851@student.uc.pt
Fábio Barata	LDM	Colaborador	uc2015259622@student.uc.pt
João Rodrigues	MIEEC	Colaborador	uc2015232235@student.uc.pt

#### Objetivos da unidade de QUA:

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

#### Processos utilizados por esta unidade:

Processo	ID do Processo
1. Processo de validação de documentos	MQ_LABSYNC_QUA_VALIDAÇÃO-DOCS
2. Processo de avaliação do estado do projeto	MQ_LABSYNC_QUA_ESTADO-PROJETO
3. Processo de elaboração de atas	MQ_LABSYNC_QUA_ELABORAÇÃO_ATA S
4. Processo de agendamento de reuniões	MQ_LABSYNC_QUA_REUNIÕES

### 3.7.1. - MQ\_LABSYNC\_QUA\_VALIDAÇÃO\_DOCS

#### Processo de validação de documentos

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

No dia em que a subunidade de Qualidade foi formada, o coordenador desta mesma unidade dividiu os colaboradores pelas diversas subunidades da equipa, de modo a agilizar o trabalho quando um documento é solicitado para validação, tornando o processo mais rápido e eficiente.

Caso um dos membros encarregue de fazer a validação não responder/não conseguir executar a tarefa pretendida até à data limite (deadline) definida no Trello pelo coordenador de QUA, a mesma passará para outro membro da equipa, sendo esta, delegada ao membro que apresente o menor número de tarefas associadas nesse dado momento.

#### Membros e as suas funções:

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

#### *Inputs:*

Os artefactos que servem de *input* a esta atividade são quaisquer tipo de documentos realizados por uma das subunidades. Como por exemplo, documentos de processos de métodos de trabalho.

#### *Entry Criteria:*

Para o arranque deste processo, é necessário que o responsável pelo documento solicite a sua validação ao Coordenador de Qualidade via Discord. Todos estes documentos são armazenados tanto no Google Drive como no GitHub.

#### Descrição das tarefas:

##### 1) **Avaliação da estrutura do documento:**

Nesta fase, verificamos se o documento de uma dada subunidade, contém os autores, o nome da equipa, PL, data, versão, introdução, cabeçalho e rodapé, seguindo os moldes do Manual de Qualidade geral da equipa. Todos os templates realizados pela equipa de gestão de projecto encontram-se no Google Drive da equipa.

##### 2) **Avaliação do conteúdo do documento:**

Nesta fase é feita a análise e detecção de erros, podendo estes últimos ser erros de origem gramatical, contextual e/ou falta de informação. Posteriormente é feita a correção dos erros de origem gramatical e/ou contextual. Caso seja detectada uma falta de informação, é comunicada à subunidade responsável de modo a clarificar/resolver a situação.

### 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.

## 3.7.2. - MQ\_LABSYNC\_QUA\_ESTADO\_PROJETO

### **Processo para monitorizar o curso do projeto e da equipa.**

Este processo foi criado com o objetivo de monitorizar o curso deste projeto e da equipa ao longo do tempo, certificando que as tarefas são cumpridas nas *deadlines* criadas no Trello para cada atividade, em cada sub-unidade.

### *Task Description:*

A Partir do momento que os coordenadores atribuem as tarefas aos diversos membros das subunidades e criam deadlines, cada elemento de qualidade certifica-se de que as tarefas cumprem as deadlines propostas por cada coordenador no Trello, para que não hajam falhas no progresso do trabalho.

Para além disso no final de cada semana com a recolha dos dados do Google *form* criados pela GP (Maria Moreira), é feita uma avaliação para percebermos se houver uma ou mais subunidades com alterações drásticas de horas de trabalho é analisada a sua situação, se não houve trabalho para realizar ou se não foi entregue a tempo. Caso haja anomalias, são posteriormente informadas à Gestora de Projeto.

### Membros e as suas funções:

A equipa de Qualidade, por questões de organização e distribuição de trabalho, dividiu-se e cada membro de Qualidade ficou diretamente responsável por monitorizar e avaliar os progressos de cada subunidade:

Maxmilian Rubin - Gestão de Projeto;  
Diogo Pereira - Ambiente;  
Diogo Gonçalves - Qualidade e Testes;  
João Rodrigues - Implementação e Recursos Humanos;  
Fábio Barata - Requisitos e Riscos.

### *Inputs:*

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

### *Entry Criteria:*

A monitorização dá-se por iniciada às 22h de segunda, que é a data limite do preenchimento do registo de esforço semanal do google form.

### *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:*

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

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

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



### 3.7.3. - MQ\_LABSYNC\_QUA\_ELABORAÇÃO\_ATAS

#### Processo da elaboração das atas

Semanalmente o coordenador de qualidade designa a um dos membros da sua subunidade para proceder à elaboração da respectiva ata, sendo que esse mesmo fica com a responsabilidade de minutar tudo o que seja relevante durante a reunião, bem como tirar fotos à folha de presença e ao caderno da gestora de projeto com o objetivo de controlar a assiduidade dos membros da equipa nas reuniões. No 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 das reuniões realizadas, tanto nas aulas PL como em qualquer outra reunião realizada pela equipa.

#### Membros e as suas funções:

A cada reunião o Coordenador 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 artefato que serve de input a esta atividade é o documento da ata.

#### *Entry Criteria:*

Este processo é iniciado após o 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.

#### *Task Description:*

Após a designação do membro responsável pela elaboração da ata, este deve 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 segue o modelo disponibilizado.

### **3.7.4. - MQ\_LABSYNC\_QUA\_REUNIÕES**

#### **Processo de agendamento de reuniões**

Este processo foi criado com a finalidade para facilitar a comunicação com as outras unidades, quando surgem dúvidas quanto à informação presente no Manual de Qualidade, quando deixam de existir documentos para validar e/ou para a realização de inspeções formais ao Manual de Qualidade.

#### **Membros e as suas funções:**

Todos os membros da equipa de qualidade deverão estar presentes nas reuniões marcadas, se possível.

#### ***Inputs:***

Documentos relacionados com o tema em questão.

#### ***Entry Criteria:***

Este processo tem início quando é necessário debater alguma questão pretendente entre os membros da equipa de qualidade e os restantes coordenadores.

#### ***Outputs gerados:***

- 1) O documento da Ata.
- 2) O documento da inspeção.

#### ***Exit Criteria:***

A reunião é dada como encerrada depois de todos os pontos em questão serem abordados.

### 3.8. - Ambiente [ENV]

<b>Unidade:</b> ENV	<b>Data:</b> 16.11.2018	<b>Gestor:</b> António Fraga
<b>Versão:</b> 2.0	<b>Status:</b> Baselined	<b>Vice Gestor:</b> João Miranda

#### Lista de membros desta unidade:

António Fraga	MIEEC	Coordenador	antoniofraga22@gmail.com
João Miranda	LEI	Vice-Coordenador	jomiranda710@gmail.com
Jiaqiang Chen	LDM	Colaborador	chen.jiaqiang@hotmail.com
Sérgio Morgado	LDM	Colaborador	sergio.morgado@hotmail.com

#### Processo utilizado por esta unidade:

Processo	ID do Processo
1. Processo de atendimento de tickets	MQ_LABSYNC_ENV_ATENDIMENTO_TICKETS

#### Objetivo da unidade:

- 1) Garantir o bom funcionamento e manutenção dos meios de comunicação e de trabalho comuns à toda a equipa;
- 2) Fornecer os meios que forem necessários às outras subunidades para o seu funcionamento correto dando as bases que forem necessárias para a implementação dos mesmos(tutoriais);
- 3) Atender a qualquer pedido das outras sub-unidades.

### 3.8.1. - MQ\_LABSYNC\_ENV\_ATENDIMENTO\_TICKETS

#### Processo de Criação e Manutenção dos meios de comunicação e de trabalho

Dada a necessidade de haver um meio de comunicação entre as outras subunidades com a de Ambiente, na qual seja rápido e fácil para as outras subunidades fazerem os seus pedidos de alterações de algum meio de trabalho ou de comunicação, ajuda ou qualquer outra necessidade que possam ter e queiram fazer um pedido a subunidade de ambiente. Para esse efeito foi aberto o *channel* do trello da nossa subunidade dando assim a possibilidade de qualquer membro da nossa equipa (LabSync) de criar um novo ticket com o seu pedido. Assim sendo, depois deste ter sido criado, fica aguardando que o coordenador, que é quem vai vendo ao longo da semana se há novos tickets, atribua um responsável principal e que os distribua pelos membros da unidade de Ambiente (podendo ser ele mesmo o responsável). Depois disso, passam para a coluna DOING. Quando estes estão concluídos pela nossa subunidade, passam para a coluna do DONE.

#### *Inputs:*

Ticket submetido pelo trello com as seguintes informações: pedido, deadline e ficando visível a pessoa que fez o pedido.

#### *Entry Criteria:*

- Ticket resolvido (terminado).
- Pessoa que fez o pedido estar satisfeita com o output.

#### Métricas de avaliação deste processo:

- 1) Tempo até o ticket ser distribuído e ter responsável (horas);
- 2) Tempo desde que o ticket foi distribuído até ser completado (horas);
- 3) Satisfação da pessoa que fez o pedido.

#### *Outputs gerados:*

- Vai depender do tipo de ticket que for submetido.