



Universidade do Minho
Escola de Engenharia

2016/2017

UNIVERSIDADE DO MINHO

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA

LABORATÓRIOS DE ENGENHARIA INFORMÁTICA

ENGENHARIA DE SISTEMAS DE SOFTWARE - J.M. FERNANDES

Documento de Requisitos - Baseado no Template *Volére*

Dashboard de Monitorização de Plataforma de IoT

Orientação: Eng. Manuel Alves (*Primecog*)

Adriano Dias Teixeira, A67663

João Miguel Fernandes, A67727

Jorge Pedro Simão, A64335

Luís Paulo Pedro, A70415

Luís Miguel Viana, A71548

Pedro Rites Lima, A61061

27 de Junho de 2017

Conteúdo

1	Motivações	5
1.1	Propósito do Produto	5
1.1.1	Problema do Utilizador ou Contexto do Projeto	5
1.1.2	Objetivos do Projeto	5
1.2	Cliente, Consumidor e Outras Partes Interessadas	6
1.2.1	O Cliente	6
1.2.2	O Consumidor	6
1.3	Utilizadores do Produto	6
1.3.1	Utilizadores Práticos do Produto	6
1.3.2	Prioridades Atribuídas aos Utilizadores	7
1.3.3	Participação do Utilizador	7
1.3.4	Utilizadores de Manutenção e Técnicos de Serviço	7
2	Limitações e Restrições do Projeto	8
2.1	Restrições Mandatórias	8
2.1.1	Restrições da Solução	8
2.1.2	Ambiente de Implementação do Sistema Proposto	8
2.1.3	Aplicações de Parceiros	8
2.1.4	Software 'Off-the-Shelf'	8
2.1.5	Restrições de Agenda	9
2.1.6	Restrições Orçamentais	9
2.2	Convenções e Definições de Nomenclatura	9
2.2.1	Definição de Termos e Acrónimos Usados no Projeto	9
3	Requisitos Funcionais	10
3.1	Alcance do Produto	10

3.1.1	Situação Atual	10
3.1.2	Contexto do Produto	10
3.2	Alcance do Produto	12
3.2.1	Limites do Produto	12
3.2.2	Lista de Casos de Uso do Produto	12
3.2.3	Especificação de Casos de Uso	13
3.3	Requisitos Funcionais e de Dados	20
3.3.1	Requisitos Funcionais	20
4	Requisitos Não-Funcionais	23
4.1	Requisitos de Aparência e Usabilidade	23
4.1.1	Requisitos de Estilo e Aspeto	23
4.2	Requisitos de Usabilidade e Humanidade	24
4.2.1	Facilidade de Uso	24
4.2.2	Requisitos de Aprendizagem	25
4.3	Requisitos de Performance	25
4.3.1	Confiabilidade e Disponibilidade	25
4.3.2	Requisitos de Capacidade	25
4.3.3	Escalabilidade e Extensabilidade	26
4.4	Requisitos Operacionais	26
4.4.1	Ambiente Físico Esperado	26
4.4.2	Requisitos para <i>Interfacing</i> com Sistemas Adjacentes	26
4.4.3	Requisitos de Lançamento	27
4.5	Requisitos de Segurança	27
4.5.1	Requisitos de Acesso	27
4.5.2	Requisitos de Integridade	28
4.5.3	Requisitos de Privacidade	30
5	Questões do Projeto	32
5.1	Questões Abertas	32
5.2	Soluções 'Off-the-Shelf'	32
5.2.1	Produtos Disponíveis:	32
5.2.2	Componentes Reutilizáveis:	33
5.2.3	Produtos Passíveis de Serem Copiados:	33
5.3	Problemas Emergentes	34

5.3.1	Efeitos no Sistema Actual:	34
5.3.2	Potenciais Problemas do Utilizador:	34
5.3.3	Limitações Que Possam Inibir o Novo Produto:	35
5.3.4	Problemas de Acompanhamento:	35
5.4	Tarefas	35
5.4.1	Planeamento do Projeto	35
5.4.2	Planeamento das Fases de Desenvolvimento	36
5.5	Migração Para o Novo Produto	37
5.5.1	Requisitos da Migração Para o Novo Produto	37
5.5.2	Dados Passíveis de Modificação/Tradução	37
5.6	Riscos	37

Lista de Figuras

3.1	Arquitetura aplicacional proposta para o contexto de desenvolvimento do produto de <i>software</i>	11
3.2	Diagrama de Use Cases do <i>software</i>	12
3.3	Caso de Uso - "Apagar utilizador"	13
3.4	Caso de Uso - "Definir permissões de widget"	14
3.5	Caso de Uso - "Editar perfis de utilizador"	14
3.6	Caso de Uso - "Registar nova conta de utilizador"	15
3.7	Caso de Uso - "Apagar widget"	15
3.8	Caso de Uso - "Carregar novo <i>layout</i> "	16
3.9	Caso de Uso - "Editar perfil"	16
3.10	Caso de Uso - "Gerar relatório estatístico"	17
3.11	Caso de Uso - "Instanciar novo widget"	17
3.12	Caso de Uso - "Efetuar login"	18
3.13	Caso de Uso - "Efetuar logout"	18
3.14	Caso de Uso - "Personalizar widget"	18
3.15	Caso de Uso - "Registar conta de utilizador"	19
3.16	Caso de Uso - "Remover conta de utilizador"	20
3.17	Caso de Uso - "Salvar <i>layout</i> da página"	20
5.1	Diagrama de Gantt ilustrativo do processo de desenvolvimento do produto de <i>software</i>	35

Capítulo 1

Motivações

1.1 Propósito do Produto

1.1.1 Problema do Utilizador ou Contexto do Projeto

Contextualizado na dinâmica da unidade curricular de Laboratórios de Engenharia Informática do 4º ano do Mestrado Integrado em Engenharia Informática na Universidade do Minho, sendo pretendido o desenvolvimento de uma aplicação web de monitorização/visualização em *real-time* de métricas de uma plataforma de Internet of Things (IoT). Esta aplicação deverá fornecer um *dashboard* cujas funcionalidades e configurações deverão ser personalizáveis de acordo com as preferências de cada utilizador. Este projeto visa, assim, a implementação de um ambiente de visualização que permita uma mais fácil e eficaz monitorização de diversos factores e métricas prioritárias no contexto do papel desempenhado, na plataforma de Internet of Things, por parte do utilizador.

1.1.2 Objetivos do Projeto

Pretende-se com este projeto de desenvolvimento de *software* a obtenção de um aplicativo que permita a inclusão de diversos *web widgets* numa *dashboard* inicialmente vazia, possibilitando assim a visualização de apenas a informação que o utilizador considerar relevante. Pretende-se que cada *widget* desenvolvido possa ser configurável, sendo possível utilizar um mesmo *widget* com configurações diferenciadas num mesmo painel de visualização. Desta forma, será garantido a um grupo de utilizadores (incluindo administradores e entidades subscritoras) uma plataforma de suporte aos processos de análise e monitorização de dispositivos IoT, sendo estas diferenciadas mediante as permissões associadas a cada utilizador.

1.2 Cliente, Consumidor e Outras Partes Interessadas

1.2.1 O Cliente

- Primecog

1.2.2 O Consumidor

- Primecog;
- Entidades subscritoras da plataforma IoT;

1.3 Utilizadores do Produto

1.3.1 Utilizadores Práticos do Produto

- Entidades Subscritoras

As entidades subscritoras da plataforma IoT de suporte ao projeto de desenvolvimento do presente *software* visam a utilização da aplicação desenvolvida como auxílio dos processos de análise e monitorização de dispositivos IoT com acessibilidade acrescida e garantia de coerência em tempo real. As funcionalidades dos *widgets* desenvolvidos deverão suportar diferentes intenções de utilização, diferenciando as informações apresentadas com base no tipo de utilizador registado. Assim, cada entidade deverá apenas visualizar no *dashboard* dados relevantes e contextualizados com o seu propósito de utilização do sistema, não permitindo o acesso a dados restritos ao painel de administradores da plataforma.

- Administradores de Plataforma

O painel de administração deverá, de forma análoga às entidades subscritoras do serviço, ser capaz de visualizar e monitorizar dispositivos conectados à plataforma IoT, ainda que deva ser dado tratamento diferenciado no acesso a dados que deverão ser restritos para os restantes utilizadores. Desta forma um administrador deverá ser capaz de, para além de visualizar dados em formato gráfico e obter relatórios estatísticos, definir alertas e notificações adaptáveis a diferentes parametrizações, bem como os processos administrativos comuns associados a uma plataforma desta tipologia (eliminação de registos, edição de dados em falta, entre outros).

1.3.2 Prioridades Atribuídas aos Utilizadores

- **Entidades Subscritoras/Administradores de Plataforma**

Uma vez que as entidades subscritoras e o painel de administração da plataforma serão os únicos utilizadores práticos da aplicação, é considerada de máxima prioridade a garantia de que os requisitos do sistema gerados após análise destes perfis de utilizador estejam de acordo com as intenções reais do cliente.

1.3.3 Participação do Utilizador

- **Entidades Subscritoras/Administradores de Plataforma**

Os utilizadores deverão participar no processo de desenvolvimento do produto, fornecendo *feedback* e aconselhamento, devendo ser assegurada a uma participação produtiva e prolífera no sentido de promoção da qualidade do produto obtido, reflectindo a solução inicial proposta. Em última instância, sendo os consumidores finais da aplicação, as críticas e opiniões dos utilizadores da plataforma ao longo do desenvolvimento do *software* conduzem para a garantia de usabilidade e objectividade do produto final, sem que seja corrido o risco de este não estar de acordo com o inicialmente requisitado.

1.3.4 Utilizadores de Manutenção e Técnicos de Serviço

- **Engenheiros de Dados**

O papel dos engenheiros de dados é o desenvolvimento, manutenção, teste e avaliação de soluções de *big data* no sentido de promover uma limpeza regular de dados e a implementação de soluções de escalabilidade dos dados armazenados de suporte à plataforma desenvolvida. Desta forma, algumas das funções associadas a este tipo de técnicos de serviço são o tratamento e análise de *logs* de erro e a monitorização da *performance* do sistema e seus dados.

- **Desenvolvedores**

A equipa de desenvolvimento, para além de estar continuamente envolvida no desenvolvimento de soluções de *software* num projecto específico, devem ainda preparar e implementar soluções determinando as especificações do sistema a ser desenvolvido, programando com recurso a tecnologias viáveis de estado-de-arte, técnicas e equipamento computacional. Devem colaborar ainda com outros técnicos de serviço, nomeadamente engenheiros de dados e analistas, no sentido de sumarizar o desenvolvimento e minorar o impacto de problemas de implementação que possam surgir no decorrer da análise dos requisitos especificados.

Capítulo 2

Limitações e Restrições do Projeto

2.1 Restrições Mandatórias

2.1.1 Restrições da Solução

a) O produto deve ser implementado com a framework Angular 2.

Motivo: O cliente está familiarizado com o ambiente de desenvolvimento especificado, sendo que pretende mantê-lo uniforme e, portanto, inalterado.

Critério de validação: A aplicação web é desenvolvida com recurso à framework Angular 2.

2.1.2 Ambiente de Implementação do Sistema Proposto

2.1.3 Aplicações de Parceiros

- Plataforma de IoT (Primecog)

Deverá ser utilizada como fonte de dados a camada de armazenamento da plataforma de IoT associada ao presente projeto de *software*. Desta forma, deverá ser garantida a implementação de uma solução que esteja conectada a esta fonte de dados e que permita a manipulação e visualização de métricas armazenadas nesta infraestrutura.

2.1.4 Software 'Off-the-Shelf'

- API de Gestão de Sessões de Utilizador

De forma a garantir a gestão de utilizadores e suas permissões, deverá ser utilizada uma API que permita uma agilização e minoração da exigência/complexidade de implementação necessárias.

- API de Ligação a Camada de Dados (Primecog)

A ligação à fonte de dados deverá ser efetuada através de uma API construída com esse propósito por parte do cliente. Desta forma, os dados utilizados pela camada de apresentação da aplicação serão facilmente adquiridos por parte da plataforma a desenvolver.

2.1.5 Restrições de Agenda

Ainda que tenha sido definido um planeamento de trabalho para todo o tempo de desenvolvimento da solução, todo o agendamento e apresentação de produtos intermédios (protótipos) ao cliente foi realizado semanalmente, pelo que os objetivos foram sendo estabelecidos realisticamente ao longo do tempo de vida do projecto.

2.1.6 Restrições Orçamentais

Derivado do contexto de desenvolvimento, não estão previstas quaisquer restrições orçamentais relativamente a licenças previstas do ambiente esperado de desenvolvimento. Quaisquer outros fundos necessários carecem de aprovação e financiamento por parte do cliente.

2.2 Convenções e Definições de Nomenclatura

2.2.1 Definição de Termos e Acrónimos Usados no Projeto

LEI: Laboratório em Engenharia Informática;

ESS: Engenharia de Sistemas de *Software*;

IoT: Internet of Things;

UI: User Interface;

API: *Application Programming Interface*;

Capítulo 3

Requisitos Funcionais

3.1 Alcance do Produto

3.1.1 Situação Atual

O desenvolvimento do presente produto de *software* visa colmatar a lacuna derivada da carência de uma plataforma acessível de gestão e visualização de informação relativa a plataformas de IoT. Desta forma, assume-se este produto de *software* como um novo produto, e não como qualquer actualização a qualquer outro *software* antecedente.

3.1.2 Contexto do Produto

A plataforma de gestão e visualização proposta a ser desenvolvida enquadra-se num contexto de posterior implementação num ambiente de dispositivos de *Internet of Things*, propondo-se a disponibilizar de forma gráfica diversos dados obtidos no decorrer do processo de captura de dados inerente a cada dispositivo utilizado no sistema do cliente. O diagrama seguinte ilustra a arquitetura proposta:

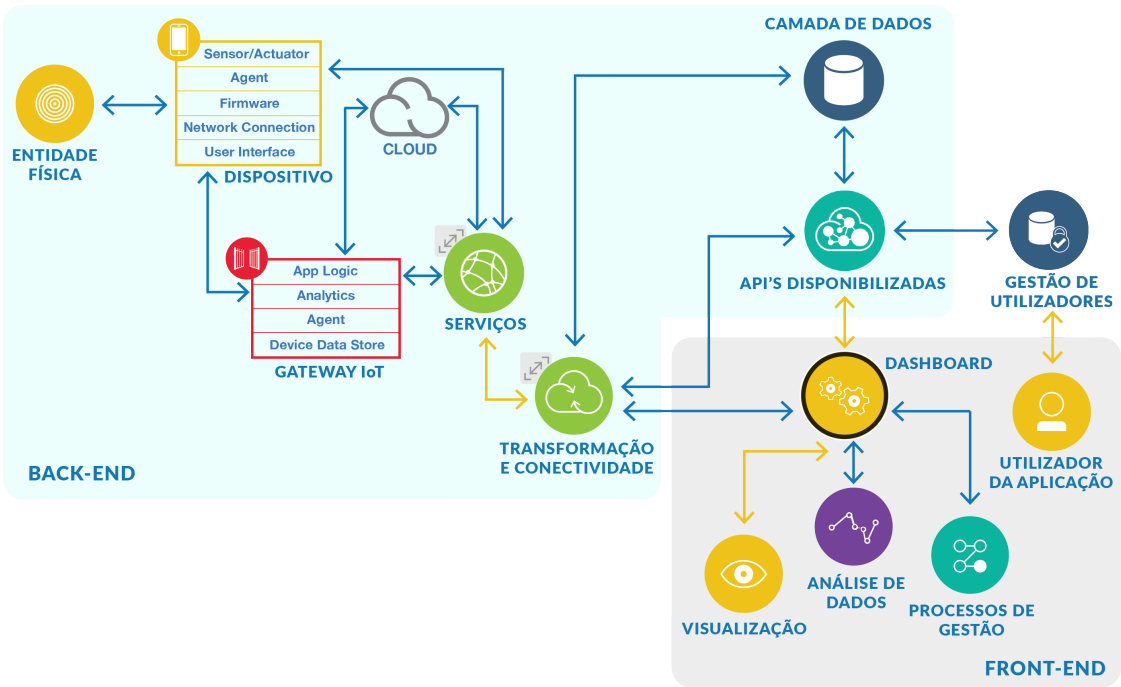


Figura 3.1: Arquitetura aplicacional proposta para o contexto de desenvolvimento do produto de *software*

3.2 Alcance do Produto

3.2.1 Limites do Produto

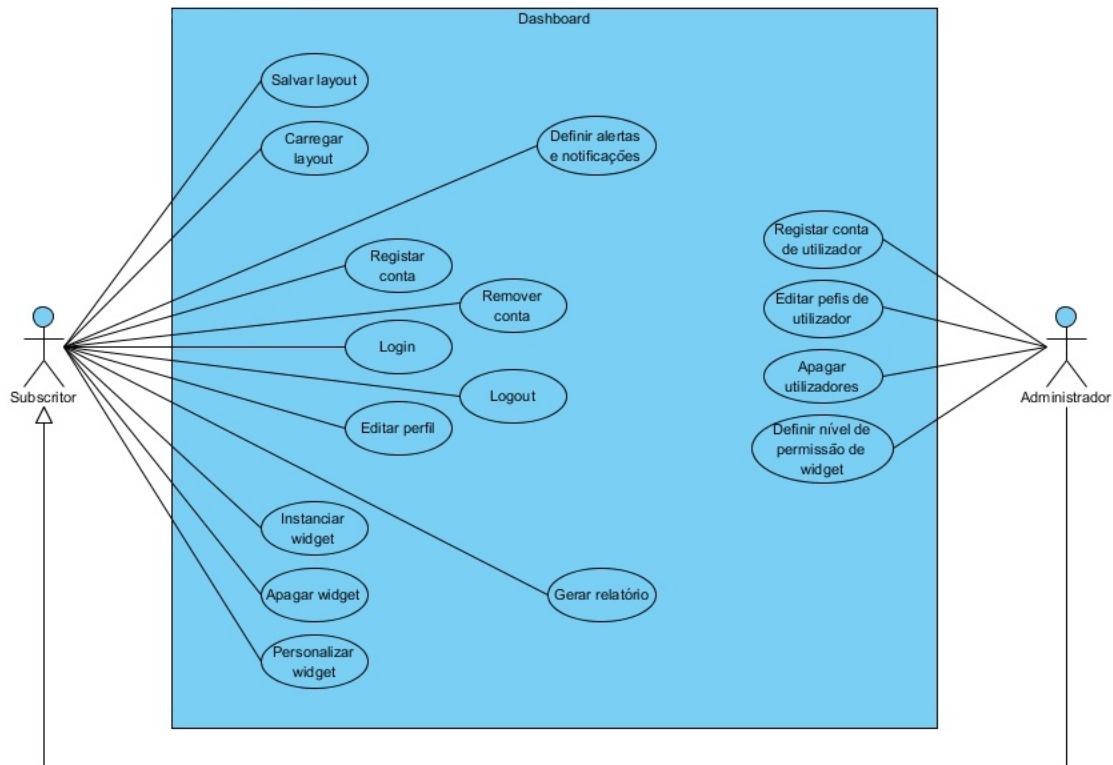


Figura 3.2: Diagrama de Use Cases do *software*

3.2.2 Lista de Casos de Uso do Produto

Entidades Subscritoras

- Registrar Conta
- Login
- Editar perfil
- Instanciar widget
- Personalizar widget
- Apagar widget
- Gerar relatório
- Carregar layout

- Salvar layout
- Logout
- Remover Conta

Administradores

- Registrar Conta
- Login
- Editar perfil
- Registrar Conta de Utilizador
- Editar perfis de utilizadores
- Apagar utilizadores
- Instanciar widget
- Personalizar widget
- Apagar widget
- Carregar layout
- Salvar layout
- Logout
- Remover Conta

3.2.3 Especificação de Casos de Uso

Use case: Apagar Utilizador		
Description: Apagar Utilizadores		
Precondition: O Admin tenha acesso a internet e esteja logado		
Postcondition: O admin apaga conta com sucesso		
	Actor	System
Normal Behavior:		1. Lista a lista de utilizadores
	2. Escolhe utilizador a apagar	
		3. Apaga conta e volta menu principal

Figura 3.3: Caso de Uso - "Apagar utilizador"

Use case: Definir nível de permissão de widget		
Description: Definir as permissões de widget		
Precondition: O Admin tenha acesso a internet e esteja logado		
Postcondition: O admin regista as permissões de widget		
	Actor	System
Normal Behavior:	1. Escolhe o widget que deseja alterar as permissões	
		2. Retorna as permissões que podem ser definidas
	3. Escolhe permissões que quer alterar	
		4. Regista alterações e volta ao menu principal

Figura 3.4: Caso de Uso - "Definir permissões de widget"

Use case: Editar Perfis de Utilizador		
Description: Alterar Perfil de um Utilizador		
Precondition: O Admin tenha acesso a internet e esteja logado		
Postcondition: O admin altera dados com sucesso		
	Actor	System
Normal Behavior:		1. Lista a lista de utilizadores
	2. Escolhe perfil a alterar	
		3. Retorna dados que podem ser alterados
	4. Altera dados	
		5. Regista dados e volta menu principal

Figura 3.5: Caso de Uso - "Editar perfis de utilizador"

Use case: Registrar conta		
Description: O Admin cria conta		
Precondition: Admin tenha acesso a internet		
Postcondition: Admin cria nova conta		
	Actor	System
Normal Behavior:	5. Admin Insere Utilizador	
	6. Admin Insere Password	
		7. Verifica se existe conta com mesmo utilizador
		8. Informa que a conta foi criada com sucesso
Alternative Behavior:		8.1. Informa que já existe uma conta com o mesmo nome do utilizador escolhido
	4.2 Retorna 1	

Figura 3.6: Caso de Uso - "Registrar nova conta de utilizador"

Use case: Apagar Widget		
Description: Apagar widget		
Precondition: Existir widget		
Pós-condição: Apagado widget da dashboard		
	Actor	System
Normal Behavior:	1. Escolhe widget que deseja apagar	
		2. O sistema apaga widget

Figura 3.7: Caso de Uso - "Apagar widget"

Use case: Carregar Layout		
Description: O utilizador escolhe layout que deseja carregar		
Precondition: O Utilizador tenha acesso a internet		
Postcondition: Utilizador carrega Layout		
	Actor	System
Normal Behavior:	1. Escolhe carregar layout	
		2. Lista Layouts
	3. Escolhe layout que quer carregar	
		4. Carrega Layout

Figura 3.8: Caso de Uso - "Carregar novo *layout*"

Use case: Editar Perfil		
Description: O utilizador altera campos da sua conta		
Precondition: O Utilizador tenha acesso a internet		
Postcondition: O utilizador edita o que deseja no seu perfil		
	Actor	System
Normal Behavior:	1. O utilizador altera o utilizador	
		2. Regista a alteração
		3. Retorna menu principal
Alternative Behavior:	1.1 O utilizador altera password	
		1.2 Retorna menu <u>principal</u>

Figura 3.9: Caso de Uso - "Editar perfil"

Use case: Gerar relatório		
Description: Gerar relatório do layout		
Precondition: O Utilizador tenha acesso a internet		
Postcondition: Gerado relatório		
	Actor	System
Normal Behavior:	1. Escolhe gerar relatório	
		2. Gerar relatório em pdf e devolve ao utilizador

Figura 3.10: Caso de Uso - "Gerar relatório estatístico"

Use case: Instanciar Widget		
Description: Iniciar novo widget		
Precondition: Estar num layout		
Pós-condição: Gerado widget		
	Actor	System
Normal Behavior:	1. Escolhe widget que deseja colocar	
		2. O sistema coloca widget na dashboard

Figura 3.11: Caso de Uso - "Instanciar novo widget"

Use case: Login		
Description: O utilizador efetua login		
Precondition: Utilizador tenha acesso a internet		
Postcondition: Utilizador efetuar login		
	Actor	System
Normal Behavior:	1. Insere o Utilizador	
	2. Insere o Utilizador	
		3. Verifica se existe conta
		4. Informa login foi feito com sucesso
Exception Behavior:		4.1. Informa que o login não foi efetuado
	4.2. Retorna 1	

Figura 3.12: Caso de Uso - "Efetuar login"

Use case: Logout		
Description: Utilizador efectua logout		
Precondition: O Utilizador tenha acesso a internet		
Postcondition: O utilizador efectua logout		
	Actor	System
Normal Behavior:	1. Utilizador escolhe opção logout	
		2. Volta ao menu inicial login

Figura 3.13: Caso de Uso - "Efetuar logout"

Use case: Personalizar Widget		
Description:		
Precondition: Existir widget		
Pós-condição: Personalizar widget da dashboard		
	Actor	System
Normal Behavior:	1. Insere dados no widget da dashboard	
		2. O sistema altera os dados

Figura 3.14: Caso de Uso - "Personalizar widget"

Use case: Registar conta		
Description: O utilizador cria conta		
Precondition: Utilizador tenha acesso a internet		
Postcondition: Utilizador cria nova conta		
	Actor	System
Normal Behavior:	1. Utilizador Insere Utilizador	
	2. Utilizador Insere Password	
		3. Verifica se existe conta com mesmo utilizador
		4. Informa que a conta foi criada com sucesso
Exception Behavior:		4.1. Informa que já existe uma conta com o mesmo nome do utilizador escolhido
	4.2 Retorna 1	

Figura 3.15: Caso de Uso - "Registar conta de utilizador"

Use case: Registrar conta		
Description: O Admin cria conta		
Precondition: Admin tenha acesso a internet		
Postcondition: Admin cria nova conta		
	Actor	System
Normal Behavior:	5. Admin Insere Utilizador	
	6. Admin Insere Password	
		7. Verifica se existe conta com mesmo utilizador
		8. Informa que a conta foi criada com sucesso
Alternative Behavior:		8.1. Informa que já existe uma conta com o mesmo nome do utilizador escolhido
	4.2 Retorna 1	

Figura 3.16: Caso de Uso - "Remover conta de utilizador"

Use case: Salvar layout		
Description: Guardar layout		
Precondition: O Utilizador tenha acesso a internet		
Postcondition: O utilizador guarda layout		
	Actor	System
Normal Behavior:	1. Utilizador escolhe guardar layout	
		2. Guarda Layout

Figura 3.17: Caso de Uso - "Salvar *layout* da página"

3.3 Requisitos Funcionais e de Dados

3.3.1 Requisitos Funcionais

a) O sistema deverá ser configurável de acordo com as preferências de cada utilizador.

ID: RF1

Motivo: Cada utilizador possui necessidades específicas e personalidade única, devendo o

dashboard ser adaptável às preferências e experiência de utilização pretendida por cada utilizador.

Critério de validação: As configurações dos dashboards de dois utilizadores distintos poderão diferir, sendo que a aplicação permite a edição de cada *widget* relativamente à fonte de dados ou a apresentação desta informação.

Prioridade: 2 (dois)

b) O dashboard terá que permitir adicionar web widgets de uma biblioteca

ID: RF2

Motivo: Os utilizadores configuram o dashboard, adicionando as funcionalidades que pretendem de acordo com as suas necessidades.

Critério de validação: Ao longo do tempo, a configuração do dashboard de um utilizador pode sofrer alterações, incluindo mais ou menos widgets.

Prioridade: 1 (um)

c) O utilizador deverá poder configurar individualmente cada widget, de forma independente dos restantes.

ID: RF3

Motivo: Cada widget possui características específicas, verificando-se uma heterogeneidade configuracional entre os widgets existentes.

Critério de validação: As opções de configuração de determinados widgets poderão não estar presentes em outros widgets.

Prioridade: 3 (três)

d) Cada *widget* deverá poder ser adicionado mais do que uma vez ao *dashboard*

ID: RF4

Motivo: Os utilizadores poderão pretender utilizar um mesmo *widget* com configurações díspares, incluindo no dashboard várias instâncias de um mesmo *widget* e configurando cada instância de acordo com as suas preferências.

Critério de validação: Cada instanciação de *widget* é independente da sua tipologia, devendo cada um possuir a sua própria configuração. Desta forma, o dashboard de um utilizador pode compreender a inclusão múltipla do mesmo *widget* com configurações distintas.

Prioridade: 1 (um)

e) A aplicação deve suportar a utilização de gráficos de diversos tipos, quer estáticos, quer dinâmicos.

ID: RF5

Motivo: A experiência de utilização da plataforma deverá ser promovida através do recurso a tecnologias que permitam uma mais acessível visualização das métricas fornecidas.

Critério de validação: Os *widgets* desenvolvidos e passíveis de serem apresentados na *dashboard* utilizam gráficos para apresentação de dados.

Prioridade: 4 (quatro)

Capítulo 4

Requisitos Não-Funcionais

4.1 Requisitos de Aparência e Usabilidade

4.1.1 Requisitos de Estilo e Aspeto

a) O dashboard deverá possuir um menu hierárquico, do lado esquerdo do ecrã, com links para cada um dos widgets existentes.

ID: RNF1

Motivo: A generalidade dos menus existentes em aplicações encontra-se do lado esquerdo do ecrã, sendo que as pessoas já se habituaram a esta característica, que é intuitiva.

Critério de validação: Do lado esquerdo do ecrã deverá estar sempre presente um menu com links para os widgets existentes, estruturado de forma hierárquica.

Prioridade: Máxima

b) O menu de widgets deverá colapsar e expandir, conforme necessário, aquando da navegação no mesmo.

ID: RNF2

Motivo: As opções existentes no menu de widgets estruturam-se de forma hierárquica, devendo a navegação no menu ser feita de modo a que seja possível situar-se facilmente e, ainda, poder ver o restante conteúdo do ecrã.

Critério de validação: Ao navegar no menu de widgets, este deverá expandir os sub-menus à medida que os mesmos são selecionados, bem como colapsar menus de um nível hierárquico superior à medida que os sub-menus são selecionados.

Prioridade: Máxima

c) O dashboard deverá apresentar, seguindo a ordem respetiva, links para todos os níveis hierárquicos de menus de widgets atualmente selecionados.

ID: RNF3

Motivo: O dashboard deverá permitir uma fácil navegação no menu de widgets, devendo haver links para todos os menus nos quais o atual se insere, permitindo um fácil retrocesso na hierarquia de menus de widgets.

Critério de validação: Aquando da navegação no menu de widgets, deverão ser apresentados links para todos os níveis hierárquicos de menus de widgets selecionados anteriormente.

Prioridade: Máxima

d) A *dashboard* deverá apresentar-se vazia na primeira utilização.

ID: RNF4

Motivo: É pretendida a instanciação de *widgets* personalizáveis de acordo com a informação requisitada pelo utilizador, não fazendo sentido a instanciação de *widgets* irrelevantes para o propósito do utilizador.

Critério de validação: A *dashboard* não instancia qualquer *widget* não pretendido pelo utilizador, devendo esta instanciação ser efetuada manualmente.

Prioridade: Máxima

4.2 Requisitos de Usabilidade e Humanidade

4.2.1 Facilidade de Uso

a) Deverão ser entregues versões prototipadas da plataforma a ser desenvolvida no decorrer deste processo.

ID: RNF5

Motivo: Deverão ser conduzidos testes de usabilidade e interação com o utilizador ao longo do processo de desenvolvimento com o propósito de assegurar a maior proximidade com os requisitos levantados com o cliente.

Critério de validação: Pelo menos uma versão de protótipo é produzida e testada durante a fase de desenho e implementação da solução.

Prioridade: Média

4.2.2 Requisitos de Aprendizagem

a) Um utilizador tipo não deverá ter dificuldade acrescida no manuseamento da plataforma, pelo que o processo de aprendizagem do *software* não deverá ser superior a 7 dias.

ID: RNF6

Motivo: A acessibilidade e manuseabilidade apresentada por uma aplicação reflecte a preocupação com os requisitos de usabilidade do *software* e com as questões relacionadas com o utilizador final da plataforma. Assim, quanto menor a curva de aprendizagem de um *software*, mais usável e acessível ele se torna, e portanto mais atractivo e eficaz funcionalmente relativamente à sua especificação.

Critério de validação: Um utilizador subscritor e um administrador não tomam mais que 7 dias para efetuar a curva de aprendizagem da plataforma desenvolvida.

Prioridade: Média

4.3 Requisitos de Performance

4.3.1 Confiabilidade e Disponibilidade

a) O produto deverá estar disponível para utilização 24 horas por dia, 365 dias por ano.

ID: RNF7

Motivo: Qualquer instância da plataforma deverá ser acessível ao utilizador a qualquer momento.

Critério de validação: Não existem restrições relativamente à disponibilidade do sistema e dos seus serviços.

Prioridade: Alta

4.3.2 Requisitos de Capacidade

a) A plataforma não deverá apresentar problemas visíveis relativos a acesso simultâneo e concorrente por parte de diversos utilizadores.

ID: RNF8

Motivo: A aplicação deve permitir o acesso concorrente aos mesmos componentes e funcionalidades.

Critério de validação: A concorrência é tratada e o sistema possui a capacidade de lidar com acesso simultâneo.

Prioridade: Alta

4.3.3 Escalabilidade e Extensibilidade

a) A arquitetura aplicacional da *dashboard* deve permitir o seu crescimento escalável e adição de funcionalidades de forma extensível, permitindo a acomodação de implementações adicionais posteriores ao lançamento do produto.

ID: RNF9

Motivo: O sistema deve permitir a adição posterior de funcionalidades de forma acessível e que não coloque em causa as decisões arquiteturais no processo de desenho da solução.

Critério de validação: Não existe um atraso visível ou um decréscimo observável na apresentação de páginas ou no manuseamento da plataforma por parte dos utilizadores.

Prioridade: Alta

b) O sistema deve permitir o crescimento do número de utilizadores sem que haja consequências na *performance* da plataforma.

ID: RNF10

Motivo: O sistema deve permitir o crescimento da base de utilizadores sem que a acessibilidade e fiabilidade da aplicação decresça.

Critério de validação: A plataforma não apresenta quaisquer consequências observáveis no decorrer do crescimento populacional dos seus utilizadores.

Prioridade: Alta

4.4 Requisitos Operacionais

4.4.1 Ambiente Físico Esperado

4.4.2 Requisitos para *Interfacing* com Sistemas Adjacentes

a) O produto deverá ser compatível com a versão 56.0.2924 do browser Google Chrome.

ID: RNF11

Motivo: O cliente não manifestou interesse relativo à compatibilidade do produto com outros browsers ou diferentes versões do Google Chrome.

Critério de validação: O produto deverá funcionar correta e completamente na versão 56.0.2924 do browser Google Chrome.

Prioridade: Máxima

4.4.3 Requisitos de Lançamento

a) O produto de *software* deverá estar disponível, com o seu conjunto de funcionalidade completo, durante o mês de Maio de 2017.

ID: RNF12

Motivo: O lançamento do produto deverá coincidir com os prazos impostos pelo cliente e equipa coordenadora do projecto de *software*.

Critério de validação: O produto entregue corresponde integralmente aos requisitos (funcionais/não funcionais) levantados com o cliente no decorrer do processo de especificação.

Prioridade: Máxima

4.5 Requisitos de Segurança

4.5.1 Requisitos de Acesso

a) Qualquer *widget* passível de ser utilizado no dashboard tem associados um nível de acesso e uma lista de permissões.

ID: RNF13

Motivo: Cada *widget* possui âmbitos e características próprias, devendo haver um controlo adequado dos acessos, não devendo ser fornecido a utilizadores o acesso a *widgets* cujas funcionalidades não correspondam ao seu propósito de utilização da plataforma.

Critério de validação: Cada utilizador poderá servir-se de determinado widget, somente se possuir credenciais de acesso válidas, sendo apenas apresentados como instanciáveis *widgets* cuja permissão de acesso seja comprovada.

Prioridade: Máxima

b) Utilizadores deverão registar-se usando credenciais privadas antes de poderem aceder à plataforma.

ID: RNF14

Motivo: Mecanismos de registo com credenciais privadas permitem a cada utilizador possuir a sua própria informação privada armazenada de forma individual.

Critério de validação: A aplicação requer o registo por parte dos utilizadores.

Prioridade: Máxima

c) Apenas administradores do sistema deverão poder consultar os detalhes privados dos utilizadores registados na plataforma.

ID: RNF15

Motivo: Nenhum dos utilizadores sem permissão de administrador deverão poder visualizar informação privada de outros utilizadores.

Critério de validação: Apenas administradores autorizados e previamente definidos podem consultar informação privada de utilizadores registados na plataforma.

Prioridade: Máxima

d) Os administradores devem definir quais as funcionalidades e privilégios dos utilizadores da plataforma.

ID: RNF16

Motivo: Nem todos os utilizadores deverão ter o mesmo nível de acesso e privilégios. Assim, o administrador do sistema deve definir quais os utilizadores que devem fazer tarefas específicas na plataforma de acordo com o seu papel no sistema.

Critério de validação: O sistema possui papéis de utilizador definidos equivocadamente, cada um com o seu nível de acesso e privilégios associados.

Prioridade: Máxima

4.5.2 Requisitos de Integridade

a) A plataforma deve prevenir a inserção de dados incorrectos.

ID: RNF17

Motivo: A plataforma recorre-se da fiabilidade dos dados armazenados. Se esta informação se tornar incoerente, então as consequências reflectem-se na usabilidade e fiabilidade do *software*.

Critério de validação: O sistema não permite a inserção de dados incorrectos.

Prioridade: Alta

b) As credenciais de acesso não podem ser duplicadas.**ID:** RNF18

Motivo: Nenhum utilizador deve poder aceder a dois perfis utilizando o mesmo par de credenciais.

Critério de validação: As credenciais armazenadas são únicas.

Prioridade: Alta

c) O sistema aloja estados coerentes em *backup*.**ID:** RNF19

Motivo: Se um estado de coerência é perdido, então o sistema deve poder recuperar dessa falha através do restabelecimento de um estado estável previamente armazenado.

Critério de validação: O sistema providencia mecanismos e rotinas de *backup*.

Prioridade: Medium

d) No caso de esquecimento das credenciais de acesso, um utilizador deve poder requisitar um novo par de credenciais.**ID:** RNF20

Motivo: Perder as credenciais de acesso não deve conduzir à inviabilidade do registo. Desta forma, o sistema deve poder permitir ao utilizador a recuperação da sua informação de acesso à plataforma.

Critério de validação: A plataforma permite ao utilizador a recuperação de *password*.

Prioridade: Alta

e) A informação privada relativa a registos de utilizadores e suas credenciais devem ser armazenadas convenientemente e com segurança.**ID:** RNF21

Motivo: Informação sensível, no caso de ser divulgada, pode levar a diversas irregularidades perante a lei. Assim, em nenhum caso esta informação deve ser disponibilizada a terceiros.

Critério de validação: O sistema aloja a informação privada de cada utilizador em estruturas de dados convenientes.

Prioridade: Alta

4.5.3 Requisitos de Privacidade

a) A informação relativa a cada sessão de utilizador deve ser armazenada apenas enquanto esta for necessária.

ID: RNF22

Motivo: Nenhuma informação desnecessária à utilização da plataforma deverá armazenada.

Critério de validação: O sistema apenas armazena informação de sessão durante o período de utilização da plataforma por parte do utilizador.

Prioridade: Alta

b) Os utilizadores devem ser capazes de retirar a sua permissão de processamento de informação privada a qualquer momento.

ID: RNF23

Motivo: Se os utilizadores não estiverem de acordo com a política de privacidade da plataforma, deverão poder cessar esta divulgação de dados a qualquer momento.

Critério de validação: O sistema possui uma opção que permite ao utilizador cessar a partilha de informação com o sistema.

Prioridade: Média

c) O sistema deve permitir ao utilizador a inspeção e correcção de diversas informações relativas ao seu registo de utilização do *dashboard*.

ID: RNF24

Motivo: O utilizador deve poder ser capaz de editar a sua informação privada a qualquer momento de utilização do *software*.

Critério de validação: O sistema fornece ao utilizador uma opção para editar a sua informação pessoal.

Prioridade: Alta

d) O sistema não deve apresentar as *passwords* em formato textual em qualquer formulário.

ID: RNF25

Motivo: As *passwords* correspondem a informação criticamente sensível, pelo que nunca deverão ser apresentadas no ecrã em formato textual.

Critério de validação: Os campos de texto relativos a *passwords* deverão ser preenchidos com o caractere '●' ou outro qualquer equivalente.

Prioridade: Alta

Capítulo 5

Questões do Projeto

5.1 Questões Abertas

Poderão surgir alguns problemas relativos à implementação do produto no ambiente contextual da empresa, pelo que:

- Deverá ser assegurado o desenvolvimento de um produto integrável com a tecnologia existente.
- O produto deverá cumprir as funcionalidades requeridas pelo cliente, ainda que possibilite futuras integrações de módulos e funcionalidades desenvolvidas pelo próprio.
- A manutenção do produto de *software*, apesar de não ser garantida, deverá ser facultada para questões técnicas de implementação e utilização do produto.

5.2 Soluções 'Off-the-Shelf'

5.2.1 Produtos Disponíveis:

Dada a especificidade da plataforma a ser desenvolvida, é possível afirmar que nenhum produto disponível possui as funcionalidades/conjunto de características que se assumem como essenciais para a coerência dos requisitos funcionais levantados com o cliente. Desta forma, acreditamos que o produto a ser desenvolvido é único, pelo que nenhum produto se enquadra numa alternativa considerada viável ao desenvolvimento do produto de *software* especificado.

5.2.2 Componentes Reutilizáveis:

- **Angular 2**

"Angular is a development platform for building mobile and desktop web applications using Typescript/JavaScript (JS) and other languages. Angular includes a wealth of essential features such as mobile gestures, animations, filtering, routing, data binding, security, internationalization, and beautiful UI components."

- **Bootstrap Framework**

"Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web."

- **Chart.js**

"Simple HTML5 Charts Javascript framework using the canvas element chartjs.org."

- **Angular.CLI**

"A command line interface for Angular, allowing to generate components, routes, services and pipes with a simple command."

- **ng2-websocket**

"The missing Angular WebSocket module for connecting client applications to servers by @AngularClass."

- **Angular 2 Grid**

"Angular 2 grid is a drag/drop/resize grid-based plugin directive for Angular 2."

5.2.3 Produtos Passíveis de Serem Copiados:

- **Azure by Microsoft**

"O Microsoft Azure é uma coleção em crescimento de serviços cloud integrados que os programadores e profissionais de TI utilizam para criar, implementar e gerir aplicações através da nossa rede global de datacenters. Com o Azure, obterá a liberdade de criar e implementar onde quiser, tudo com as ferramentas, aplicações e arquiteturas da sua preferência."

5.3 Problemas Emergentes

5.3.1 Efeitos no Sistema Actual:

O produto de *software* desenvolvido procura fornecer uma nova metodologia de gestão e visualização de dados sendo que, desta forma, deverá ser desenvolvida uma infraestrutura que possibilite implementar este conceito, sem que sejam alteradas funcionalidades ou propósitos empresariais estabelecidos pelo cliente. Assim, não deverá ser renovado ou instalado qualquer plataforma que interfira com o sistema actual onde o produto será alojado.

- A tecnologia de implementação do produto não deverá entrar em conflito com qualquer tecnologia ou sistema actualmente implementado por parte do cliente.
- Mudanças significativas na política da empresa ou relativas à própria utilização da ferramenta desenvolvida poderão inviabilizar o seu sucesso.
- O sistema desenvolvido poderá evoluir em termos escalares de conteúdo manuseado e dados geridos, pelo que a *performance* estará também dependente do servidor utilizado para alojar o produto.
- Não está estabelecida qualquer incompatibilidade conhecida ou inviabilidade na implementação do produto por parte do cliente.

5.3.2 Potenciais Problemas do Utilizador:

O produto desenvolvido não deverá nunca ser responsável pelo seu mau uso ou pelas reações despoletadas pelo seu manuseamento. Urge identificar quais os possíveis comportamentos apresentados pelos utilizadores, de forma a determinar a sua relevância e as medidas tomadas de forma a minimizar o impacto destas situações (negativas) no funcionamento geral do sistema. Destes possíveis comportamentos, destacam-se:

- Configuração de *widget* efectuada de forma errónea.
- Inserção de dados errados.
- Esquecimento de credenciais de acesso.
- Registo duplo de utilizador.
- Visualização pobre (causada por mau arranjo de janelas na grelha).
- Fonte de dados indisponível (visualização de dados estática).

5.3.3 Limitações Que Possam Inibir o Novo Produto:

- Servidor incapaz de lidar com acesso concorrente e gestão de dados em larga escala.
- Inexistência de formação no manuseamento do produto de *software*.
- Desinteresse ou não implementação da solução por parte do cliente.
- Instabilidade das fontes de dados e sistema de implementação.
- Emergência de nova tecnologia no mercado.
- Falência da empresa.
- Incapacidade funcional.

5.3.4 Problemas de Acompanhamento:

Após a entrega e apresentação do produto de *software* ao cliente, é expectável que o seu desenvolvimento verifique a consistência funcional e estrutural relativamente aos requisitos inicialmente explicitados. Torna-se importante o acompanhamento por parte da equipa de desenvolvimento após este período, no sentido de clarificar quais as situações nas quais o produto não corresponde ao esperado, de forma a resolver estas lacunas e não defraudar as expectativas realizadas pelo cliente em relação ao *software* adquirido.

5.4 Tarefas

5.4.1 Planeamento do Projeto

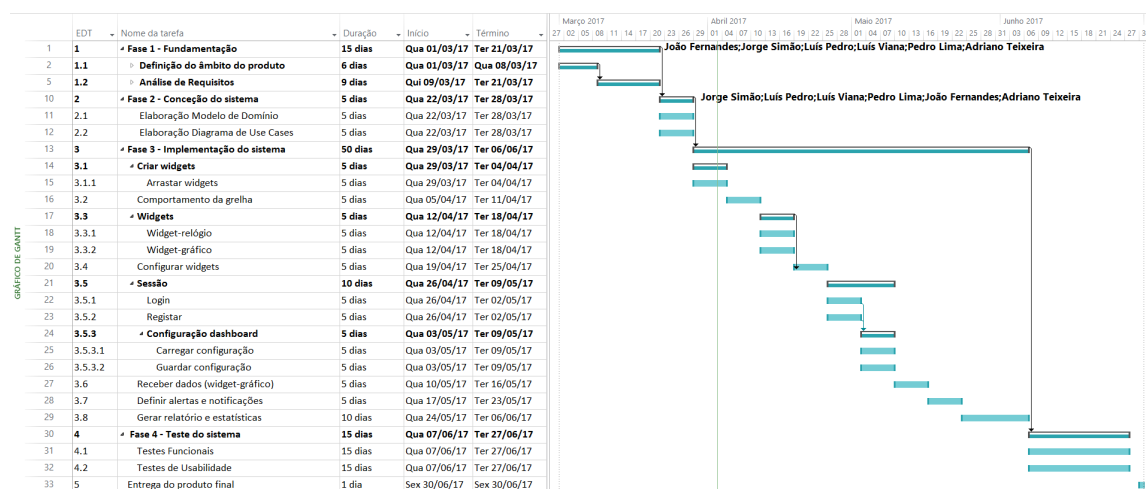


Figura 5.1: Diagrama de Gantt ilustrativo do processo de desenvolvimento do produto de *software*

5.4.2 Planeamento das Fases de Desenvolvimento

Fase 1

Nome/Descrição: Fundamentação, definição do âmbito do projecto e análise de requisitos.

Data de operabilidade: 21/03/2017

Componentes operacionais incluídos: n/d

Requisitos funcionais associados: n/d

Fase 2

Nome/Descrição: Concepção e modelação do sistema.

Data de operabilidade: 28/03/2017

Componentes operacionais incluídos: n/d

Requisitos funcionais associados: n/d

Fase 3

Nome/Descrição: Implementação do sistema

Data de operabilidade: 21/03/2017

Componentes operacionais incluídos: Widgets, Sessões, Integração de APIs.

Requisitos funcionais associados: RF1, RF2, RF3, RF4, RF5;

Fase 4

Nome/Descrição: Testes do sistema.

Data de operabilidade: 27/06/2017

Componentes operacionais incluídos: n/d

Requisitos funcionais associados: n/d

Fase 5

Nome/Descrição: Entrega final do produto de *software*.

Data de operabilidade: 30/06/2017

Componentes operacionais incluídos: n/d

Requisitos funcionais associados: n/d

5.5 Migração Para o Novo Produto

5.5.1 Requisitos da Migração Para o Novo Produto

Uma vez que não se espera a migração de um sistema previamente desenvolvido (visto que não é pretendido uma actualização mas sim a criação de um novo sistema de visualização/monitorização), não existirá qualquer faseamento nas fases de implementação física ou tecnológica, assim como não ocorrerão processos de conversão de dados previamente alojados. Desta forma, os dados não deverão ser migrados, mas sim interpretados de uma fonte externa e fornecidos ao *dashboard* de forma dinâmica com recurso a comunicação e estabelecimento de ligações HTTP.

5.5.2 Dados Passíveis de Modificação/Tradução

Para que o sistema funcione de forma eficaz e apresente informação coerente no contexto da sua utilização, torna-se necessário o tratamento da informação proveniente das camadas de dados da plataforma de IoT existente e o estabelecimento de ligações que correspondam aos processos de apresentação destes dados na aplicação desenvolvida. Estes dados são provenientes de diversos dispositivos e deverão ser tratados convenientemente para fornecimento digital aos utilizadores através do produto. Este processo de tratamento de dados tem como objectivo principal torná-los alcançáveis e usáveis por qualquer camada lógica do *dashboard* desenvolvido, apresentando-os ao utilizador em *real-time* ou não, sendo assim necessário um esforço de manutenção e certificação da integridade das estruturas de suporte a estes processos. Note-se que uma vez que o processo de tratamento de dados não corresponde a uma mera importação e inclusão desta informação num sistema criado com este efeito, corresponde a um processo que pode condicionar a fiabilidade e *performance* do sistema se não for executada de forma cautelosa. Assim, a responsabilidade de importar e tratar estes dados é acrescida, devendo o cliente (devido à sua experiência técnica no contexto do problema) acompanhar este processo de forma a que não sejam propagadas falhas para diferentes camadas estruturais do sistema, conduzindo inevitavelmente à inconsistência da aplicação e ao seu funcionamento errático.

5.6 Riscos

Riscos do Alcance e Qualidade

- Objectivos irrealistas relativamente a prazo de tempo.

Minoração de Risco: Prototipagem, estimativa de tempo e comunicação contínua com o

cliente.

Probabilidade: Alta

- Alcance do produto para além do estado da arte.

Minoração de Risco: Quantificação da especificação de atributos, prototipagem e pesquisa contínua.

Probabilidade: Baixa

- A utilização do produto envolve tarefas demasiado complexas para o utilizador.

Minoração de Risco: Decomposição de tarefas em tarefas simplificadas.

Probabilidade: Baixa

- Emergência de novas tecnologias e ferramentas.

Minoração de Risco: Pesquisa e uso de tecnologia atualizada e recente.

Probabilidade: Baixa

- Detecção tardia de defeitos no produto.

Minoração de Risco: Revisão do desenvolvimento do produto, *versioning* e prototipagem.

Probabilidade: Média

- Nova funcionalidade desejada ou pretendida pelo cliente.

Minoração de Risco: Revisão contínua do desenvolvimento do produto e manutenção de coerência estrutural.

Probabilidade: Alta

Riscos de Recursos

- Perda de produtos, fornecedores, pessoal ou infraestruturas.

Minoração de Risco:

textitBackup, pesquisa de fornecedores alternativos, promoção de trabalho em par e gestão de recursos humanos.

Probabilidade: Alta

- Orçamento cortado ou limitado.

Minoração de Risco: Reserva de parte do orçamento para situações de emergência e listagem de tarefas por prioridade e importância de implementação.

Probabilidade: Alta

- Falta de qualificação/competência do pessoal.

Minoração de Risco: Treino de recursos humanos, consultoria contratual.

Probabilidade: Alta

- Atraso no financiamento do projeto.

Minoração de Risco: Procura de financiamento externo.

Probabilidade: Média

- Riscos tecnológicos.

Minoração de Risco: Verificação de diversas combinações de tecnologia/componentes.

Probabilidade: Média

- Participantes subversivos.

Minoração de Risco: Atenção à dinâmica de grupo, isolamento e confrontação com participantes.

Probabilidade: Alta**Riscos de Agenda**

- Atraso em decisões críticas ou tarefas-chave.

Minoração de Risco: Divisão de tarefas, agendamento alternativo e métodos alternativos de colaboração.

Probabilidade: Alta

- Estagnação do desenvolvimento de certas tarefas.

Minoração de Risco: Promoção da execução duplicada de tarefas e entrega independente.

Probabilidade: Alta

- Falha no controlo de qualidade.

Minoração de Risco: Revisão e teste do produto, fornecimento de alternativas.

Probabilidade: Média

- Pressão temporária excessiva.

Minoração de Risco: *Versioning* e identificação das tarefas que poderão convergir em atraso.

Probabilidade: Alta

Bibliografia e Referências

- FERNANDES, J. M. and MACHADO, R.J., *"Requirements in Engineering Projects"*
Springer, Série Lecture Notes in Management and Industrial Engineering, ISBN 978-3-319-18596-5, 2016
- ROBERTSON, J. and ROBERTSON, S. *"Volere Requirements Specification Template"*
Edition 11, February 2006