



IPG Politécnico
|da|Guarda
Polytechnic
of Guarda

RELATÓRIO DE PROJETO

Licenciatura em Engenharia Informática

Micael José Gonçalves Capalão

dezembro | 2019





IPG

Politécnico
da Guarda

Polytechnic
of Guarda

Escola Superior de Tecnologia e Gestão

Instituto Politécnico da Guarda

RELATÓRIO DE PROJETO EM CONTEXTO DE ESTÁGIO

Análise de Requisitos em Projetos com *Salesforce*

Micael José Gonçalves Capelão

RELATÓRIO PARA A OBTENÇÃO DO GRAU DE LICENCIADO

EM ENGENHARIA INFORMÁTICA

Dezembro/2019

Agradecimentos

Em primeiro lugar agradecer aos meus pais pela oportunidade. Por todo o acompanhamento e apoio oferecido.

Agradecer a entidade de acolhimento pela possibilidade de concluir o estágio, assim como todos os membros da empresa.

Por último um agradecimento ao meu orientador Professor Celestino Gonçalves pela paciência e a serenidade que sempre mostrou e aconselhou.

Elementos identificativos

Aluno:

Nome: Micael José Gonçalves Capelão

Número: 1011158

Curso: Engenharia Informática

Orientador do projeto:

Nome: Professor Celestino Gonçalves

Supervisor do Projeto:

Nome: Eng. Carlos Pinto

Estabelecimento de ensino:

Escola superior de Tecnologia e Gestão – Instituto Politécnico da Guarda

Morada: Av. Dr. Francisco Sá Carneiro 50, 6300-559 Guarda

Duração do projeto em contexto de estágio:

Início: 11 de junho de 2018

Fim: 30 de julho de 2018

Resumo

O presente relatório tem como finalidade transmitir e apresentar todo o desenvolvimento do trabalho realizado na empresa Dom Digital, atualmente designada por Loba. Trabalho este realizado no âmbito de projeto em contexto de estágio para a unidade curricular de Projeto de Informática.

Este projeto consistiu em três etapas, sendo elas:

- O Levantamento e Análise de Requisitos de um caso de estudo em contexto real, para a empresa Bricomarché, onde se analisaria o desenvolvimento de dois serviços distintos, um de Reparações e outro de Encomendas.
- O Levantamento e Análise de Requisitos de uma aplicação nomeada de “Lista de Requisitos em *Salesforce*”.
- Implementação em *Salesforce* da aplicação analisada no segundo ponto “Lista de Requisitos em *Salesforce*”.

Este documento pretende detalhar todas as etapas no desenvolvimento do projeto, como análise e tratamento de requisitos e o desenvolvimento de diversos diagramas em concordância com os métodos utilizados diariamente pela Dom Digital.

Palavras-chave: *Salesforce*, Análise de Requisitos, Reparações, Encomendas

Abstract

The present document serves the purpose of presenting and transmitting all development and work done at Dom Digital Enterprise as a curricular internship project in the context of Project Subject of Computers Engineering Degree

This project was done in three phases, as follows:

- Analysis and Requirement Elicitation of a case study in a real life scenario, for the company Bricomarché where there will be analyzed the development of two distinct services, one for Repairs and another for Orders.
- The Analysis and requirement Elicitation of an app named as "Lista de Requisitos em *Salesforce*".
- The Implementation in *Salesforce* of the app analysed in the second point "Lista de Requisitos em *Salesforce*"

This document has the sole purpose of documenting every step of the project development, its conditions and analysis, as the projection of several diagrams in concordance with the daily methods applied by Dom Digital

Keywords: *Salesforce*, Requirements analysis, Repairs, Orders

Índice

Agradecimentos	iii
Elementos identificativos	iv
Resumo	v
Abstract.....	vi
Índice de Ilustrações	ix
Índice de Tabelas	x
Lista de Acrónimos.....	xi
1. Introdução.....	1
1.1. Contextualização	1
1.2. Caracterização sumária da instituição	2
1.3. Motivação	3
1.4. Objetivos previstos	3
1.5. Estrutura do documento.....	4
2. Estado da arte.....	5
2.1 Engenharia de Requisitos	5
2.1.1. Conceito de Requisito	7
2.1.2. Requisitos de Utilizador	8
2.1.3. Requisitos de Sistema	8
2.1.4. Requisitos funcionais e não funcionais	9
2.2. <i>Software on-premise</i> e <i>software on-cloud</i>	11
2.2.1. <i>Software on-premise</i>	11
2.2.2. <i>Software on-cloud</i> e <i>cloud computing</i>	11
2.3. <i>Salesforce</i>	16
3. Metodologia.....	20
3.1. Metodologia Agile – <i>Scrum</i>	20
3.2. <i>Trailheads</i>	22
3.3. Formação <i>Salesforce</i> presencial	24
4. Análise de Requisitos	25
4.1. Caso estudo em contexto real – Bricomarché	27
4.1.1. Lista de requisitos – Bricomarché	27
4.1.2. Diagramas de contexto	38
4.1.3. Diagrama Entidade Relacionamento e Diagrama de Classes	39
4.1.4. Dicionário de dados	41

4.1.5. Diagrama de Hierarquias	47
4.1.6. Diagramas de Fluxo	48
4.2. Caso de estudo Projeto Lista de requisitos em <i>Salesforce</i>	52
4.2.1. Lista de requisitos do caso de estudo	52
4.2.2. Diagrama entidade e relacionamentos e diagrama de classes	60
4.2.3. Dicionário de dados	61
4.2.4. Casos de uso	68
5. Tecnologias.....	73
5.1. Lightning components	73
5.2. Draw.io e UML	74
5.3. <i>Google Drive – Google Docs</i>	75
5.4. MavensMate	76
6. Implementação.....	77
6.1. Layouts	77
7. Conclusão	82
Referências Bibliográficas.....	84
Anexos.....	86
Exportação CSV	86
Exportação PDF.....	87

Índice de Ilustrações

Figura 1- <i>Banner</i> Dom Digital e Loba.....	2
Figura 2- Percentagem de Sucesso No Desenvolvimento de <i>Software</i> Erro! Marcador não definido.	
Figura 3- Principais Razões para o insucesso no desenvolvimento de <i>software</i>	6
Figura 4- Principais razões para o sucesso no desenvolvimento de <i>software</i>	7
Figura 5- Tipos de Requisito por nível.....	8
Figura 6- Requisitos não funcionais e constituintes	Erro! Marcador não definido.
Figura 7- <i>Cloud Computing</i> , características, tipos e modelos	12
Figura 8- Modelos de Distribuição de <i>Cloud Computing</i>	14
Figura 9- <i>Clouds</i> no <i>Salesforce</i>	18
Figura 10- Modelo <i>Scrum</i>	20
Figura 11- Perfil <i>Trailheads</i>	22
Figura 12- Badges <i>Trailheads</i>	23
Figura 13-Etapas constituintes da Engenharia de Requisitos.....	25
Figura 14- Estrutura tipo de um DERS Fonte: iscac.pt	Erro! Marcador não definido.
Figura 15- Diagrama de Contexto Bricomarché.....	Erro! Marcador não definido.
Figura 16- Diagrama Entidade Relacionamento Bricomarché Erro! Marcador não definido.	
Figura 17- Diagrama de classes Bricomarché	40
Figura 18- Diagrama de Hierarquias Bricomarché.....	Erro! Marcador não definido.
Figura 19- Diagrama de fluxo etapa Reparções.....	Erro! Marcador não definido.
Figura 20- Diagrama de Fluxo Etapa Encomenda.....	Erro! Marcador não definido.
Figura 21- Diagrama Entidade Relacionamento Lista de Requisitos <i>Salesforce</i>	Erro! Marcador não definido.
Figura 22- Diagrama de Classes Lista de Requisitos <i>Salesforce</i> Erro! Marcador não definido.	
Figura 23- Diagrama de casos de uso Lista de Requisitos <i>Salesforce</i>	68
Figura 24- Perfil de uma Oportunidade	Erro! Marcador não definido.
Figura 25- Layout Planificação e inserção de Valores, e <i>dashboard</i> Erro! Marcador não definido.	
Figura 26- Layout Listar Requisitos	79
Figura 27- Layout todos os Requisitos	Erro! Marcador não definido.
Figura 28- Formulário Novo Requisito	80
Figura 29- Exportação da Lista de Requisitos	Erro! Marcador não definido.
Figura 30- Lightning Components relacionamentos	Erro! Marcador não definido.
Figura 31- Bundle de um Lightning Component.....	Erro! Marcador não definido.
Figura 32- <i>Layout</i> do ambiente de trabalho <i>Draw.io</i>	75
Figura 33- Mavens Mate forma de relacionamento	76

Índice de Tabelas

Tabela 1- Vantagens e Desvantagens <i>Salesforce</i>	17
Tabela 2- Lista de Requisitos Bricomarché etapa reparação.....	28
Tabela 3- Lista de Requisitos Bricomarché etapa Encomenda	34
Tabela 4- Dicionário de dados - Tabela Account	42
Tabela 5- Dicionário de dados - Tabela Contact	42
Tabela 6- Dicionário de dados - Tabela Encomenda.....	43
Tabela 7- Dicionário de dados - Tabela Relatorios	44
Tabela 8- Dicionário de dados - Tabela Product	44
Tabela 9- Dicionário de dados - Tabela Fornecedor	45
Tabela 10- Dicionário de dados - Tabela Reparações	46
Tabela 11- Lista de Requisitos do caso de estudo Lista de Requisitos em <i>Salesforce</i>	53
Tabela 12- Dicionário de dados - Tabela Opportunity	63
Tabela 13- Dicionário de dados - Tabela Requisito	65
Tabela 14- Dicionário de dados - Tabela Account	66
Tabela 15- Dicionário de Dados - Tabela Contact	67
Tabela 16- Descrição caso de uso - Criar Conta	69
Tabela 17- Descrição Caso de uso - Criar Contacto.....	70
Tabela 18- Descrição caso de uso - Criar Oportunidade	71
Tabela 19- Descrição caso de uso - Criar Requisito.....	72

Lista de Acrónimos

- CRM- Customer Relationship Management
- CRUD - Create Read Update Delete
- CSS- Cascading Style Sheets
- CTO- Chief Technology Officer
- DERS- Documento de Especificação de Requisitos de *Software*
- ER- Engenharia de Requisitos
- HTML- HyperText Markup Language
- IaaS- Infrastructure as a Service
- IEEE-Institute of Electrical and Electronics Engineers
- Org - Organization
- PaaS- Plataforma as a Service
- SaaS- *Software* as a Service
- TI- Tecnologia da Informação
- UML- Unified Modeling Language

1. Introdução

1.1. Contextualização

O mercado das novas tecnologias está a desenvolver-se diariamente a grande velocidade, cada vez existem mais *gadgets* portáteis munidos das últimas aplicações feitas para auxiliar as mais diversas situações possíveis no quotidiano. Existem plataformas robustas em *cloud* que oferecem a capacidade de hospedar qualquer serviço, até de forma gratuita. Em suma, num mundo cada vez mais tecnológico existe por sua vez mais desenvolvimento de novo *software*, o que gera uma oferta crescente.

A questão que se coloca é, será que esta oferta em quantidade de *software* e serviços, prima também pela qualidade? Estão os *developers* preocupados em desenvolver o produto de forma otimizada, mais segura e com menos erros, ou há necessidade de oferecer o produto para as massas, no menor tempo de produção possível, descuidando a preocupação pelos métodos de desenvolvimento?

De forma a melhor descrever esses métodos de desenvolvimento de *software*, vão ser analisados dois casos de estudo. O primeiro caso consiste na análise de requisitos para dois serviços distintos pretendidos pela companhia Bricomarché, de onde resultaria um serviço de Reparções e outro de Encomendas. O segundo será uma análise e implementação de uma aplicação de nome “Lista de Requisitos”.

1.2. Caracterização sumária da instituição

A instituição de acolhimento do estágio foi a Dom Digital, situada na Guarda, é uma empresa bastante conhecida no panorama nacional pelos trabalhos desenvolvidos em *Salesforce*, sendo que foram os primeiros parceiros oficiais *Salesforce* em Portugal, em 2003. Tem serviços desde consultoria (*Salesforce Gold Partner*) até ao desenvolvimento de aplicações ou soluções na *cloud*, já com provas dadas no desenvolvimento com exemplos como o RTP *Play*, Dom Hotel e *speedy data*.

Durante o período no qual decorreu o estágio, a empresa estava a passar por um processo de crescimento por fusões e aquisições por parte da empresa LOBA, Figura 1.



FIGURA 1- BANNER DOM DIGITAL E LOBA

1.3. Motivação

Munido de vontade para ter experiência profissional em contexto real na área, procurei na Dom Digital uma proposta de projeto em contexto de estágio. Após uma conversa com a empresa foi-me questionado acerca das áreas que me despertavam maior interesse, tendo sido posteriormente sugerido o presente projeto com foco em Análise de Requisitos, parte inerente da Engenharia de *Software*.

Visto já ter interesse na área devido às diversas unidades curriculares ao longo do percurso académico como em Engenharia de *Software* I e II, mas também sentido a necessidade desta mesma área aquando do desenvolvimento de alguns trabalhos para disciplinas de cariz mais prático, como Programação ou Arquitetura de Computadores, esta necessidade tornou clara a importância de partir com uma boa base do “que se vai fazer, como e de que maneira” no momento da implementação.

1.4. Objetivos previstos

De entre outras metas, é importante notar os principais objetivos a concluir no estágio curricular:

- Análise de Requisitos em contexto real Projeto Bricomarché
- Análise de Requisitos ao Projeto Lista de Requisitos
- Implementação do último projeto

Estes objetivos pretendem ser alcançados por via de aprender e aplicar conhecimentos em *Salesforce* assim como desenvolver programação na Plataforma, aprender os paradigmas da Análise de Requisitos e a sua inclusão num contexto real para os projetos introduzidos.

1.5. Estrutura do documento

1. O presente relatório foi estruturado em sete capítulos, incluindo a Introdução, sendo que esta tem a finalidade de introduzir os paradigmas do projeto, as motivações, os objetivos, assim como este mesmo tópico que pretende explicitar a estrutura usada.
2. No segundo capítulo é abordado o estado da arte. Aqui foi desenvolvido o tema de Engenharia de Requisitos, assim como a noção de requisitos, devido à importância destes para com o presente relatório. São também abordados e comparados os tipos de distribuições de *software*, *on-premise* e *on-cloud*, assim como detalhado o processo de *cloud computing* de forma a melhor introduzir o conceito de *Salesforce*.
3. No terceiro capítulo é explicitado o método de desenvolvimento ágil, assim como a sua aplicação no contexto de estágio, *scrum* neste caso. É explicada a forma metódica como o estágio foi em direção aos objetivos previstos, detalhando a parte formativa em *Salesforce*, seja na plataforma de *e-learning* ou na formação presencial.
4. No quarto capítulo foi abordada a Análise de Requisitos em duas etapas:
 - A primeira etapa consistiu em fazer toda a análise de requisitos de um projeto em contexto real sobre o desenvolvimento de serviços de encomenda e reparação para um Bricomarché.
 - A segunda etapa, onde foi feita a Análise de Requisitos a um projeto de Folha de requisitos que posteriormente foi implementado. Esta segunda etapa contou com ferramentas de *Salesforce*, como *SchemaBuilder*, assim como é demonstrada as suas vantagens.
5. No quinto capítulo é detalhada a implementação do projeto no ambiente *Salesforce*, assim como as suas funcionalidades e automatismos.
6. No sexto capítulo, referente a Tecnologias, é feita uma breve descrição de ferramentas e tecnologias usadas que possam não ter sido descritas ou explicadas anteriormente no relatório.
7. Por fim, o sétimo capítulo pretende concluir o relatório, na forma de uma reflexão sumária de todo o projeto, medindo os objetivos alcançados mediante os propostos, assim como as lições aprendidas com o decorrer deste estágio.

2. Estado da arte

2.1 Engenharia de Requisitos

A Engenharia de Requisitos é um processo que visa originar um DERS (Documento de Especificação de Requisitos de *Software*) para, sem qualquer possibilidade de dúvida explicitar os requisitos a serem desenvolvidos posteriormente, assim como possibilitar a sua manutenção.

“Os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços que oferece e as restrições a seu funcionamento. Esses requisitos refletem as necessidades dos clientes para um sistema que serve a uma finalidade determinada, como controlar um dispositivo, colocar um pedido ou encontrar informações. O processo de descobrir, analisar, documentar e verificar esses serviços e restrições é chamado engenharia de requisitos (RE, do inglês requirements engineering).” [2, p. 102]

Sendo o DERS o objectivo final da Engenharia de requisitos, podemos concluir com a descrição de *Sommerville*, que este é um processo de múltiplas etapas, e que só após a conclusão de cada uma é possível originar um documento assim. Sabendo então a resultante da Engenharia de requisitos é necessário mostrar a importância da mesma para o bom e bem-sucedido desenvolvimento de *software*. Como podemos verificar pelos dados fornecidos pelo *Chaos report* [3] pelo “*Standish Group*”, existe uma percentagem muito grande de *software* que ou falha ou tem um sucesso limitado, ou seja, aquém do previamente projetado (Figura 2).



FIGURA 2- PERCENTAGEM DE SUCESSO NO DESENVOLVIMENTO DE SOFTWARE

ADAPTADO DE CHAOS REPORT[3]

Como tal é importante saber quais as causas e razões a que se deve a elevada taxa de projetos com um sucesso limitado ou insucesso (Figura 3), assim como as razões pelas quais os projetos acabaram por ser um sucesso (Figura 4). Como podemos verificar nos gráficos seguintes, as etapas de Engenharia de Requisitos ou as suas especificidades (como má explicitação de um requisito) são grandes causas do insucesso no desenvolvimento de *software*, assim como quando bem empregues são parte da solução para o sucesso.

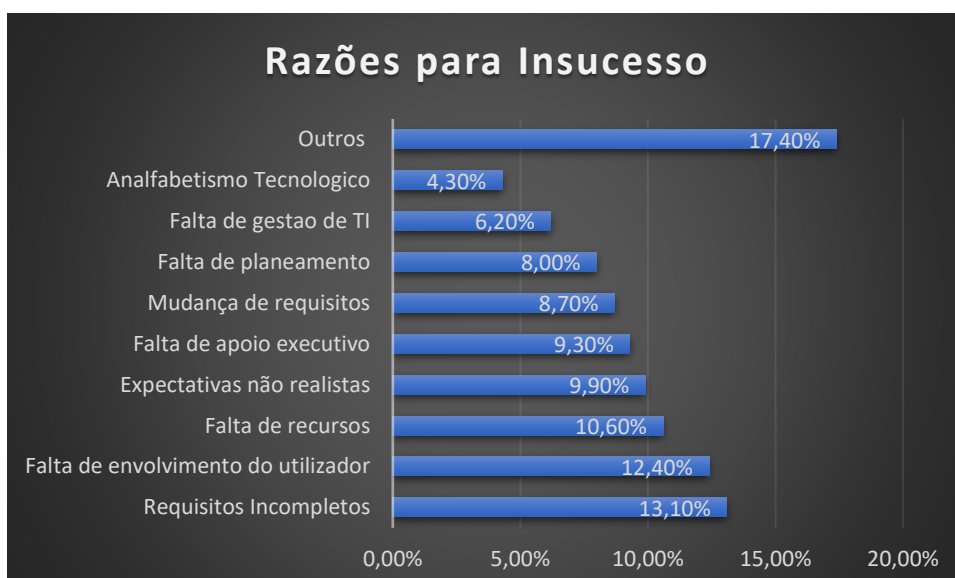


FIGURA 3- PRINCIPAIS RAZÕES PARA O INSUCESSO NO DESENVOLVIMENTO DE SOFTWARE

ADAPTADO DE CHAOS REPORT [3]

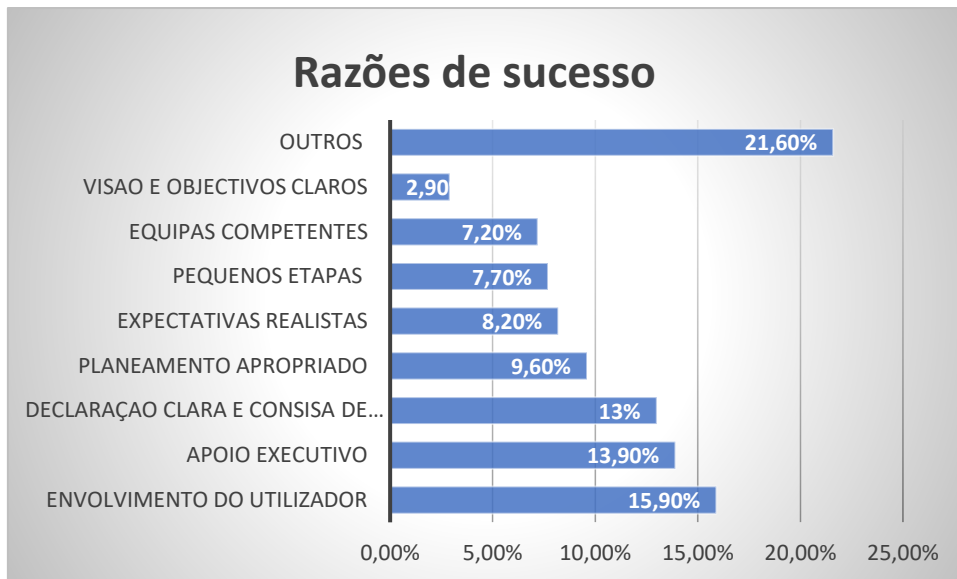


FIGURA 4- PRINCIPAIS RAZÕES PARA O SUCESSO NO DESENVOLVIMENTO DE SOFTWARE
ADAPTADO DE CHAOS REPORT [3]

Então, devido a inerente relação entre a Engenharia de Requisitos e o desenvolvimento de *software* de forma bem-sucedida, é importante definir as suas etapas, mas também definir o que são e para que servem os requisitos, sendo estes a base de toda a Engenharia de Requisitos.

2.1.1. Conceito de Requisito

Devido à sua importância, é necessário definir um requisito de forma a melhor o perceber. Segundo o guia “*Business Analysis Body of Knowledge*” (BABOK) [3], um requisito é identificado por estes seguintes pontos:

- Condição ou capacidade precisa pelo stakeholder para resolver o problema ou alcançar um objetivo.
- Condição ou capacidade que é necessária ter ou possuir pela solução ou por um componente da solução para satisfazer o contrato, predefinições, especificações ou qualquer outra especificação formalmente acordada e imposta.
- Uma representação documentada de uma condição ou capacidade como descrito em cima.

A forma como classificamos um requisito é importante pois podem prevenir problemas no desenvolvimento do processo de Engenharia de Requisitos, para isso deve ser clara a separação dos distintos níveis de descrição de um requisito. Como tal, é aceite dois tipos principais de requisitos, como demonstra a Figura 5.

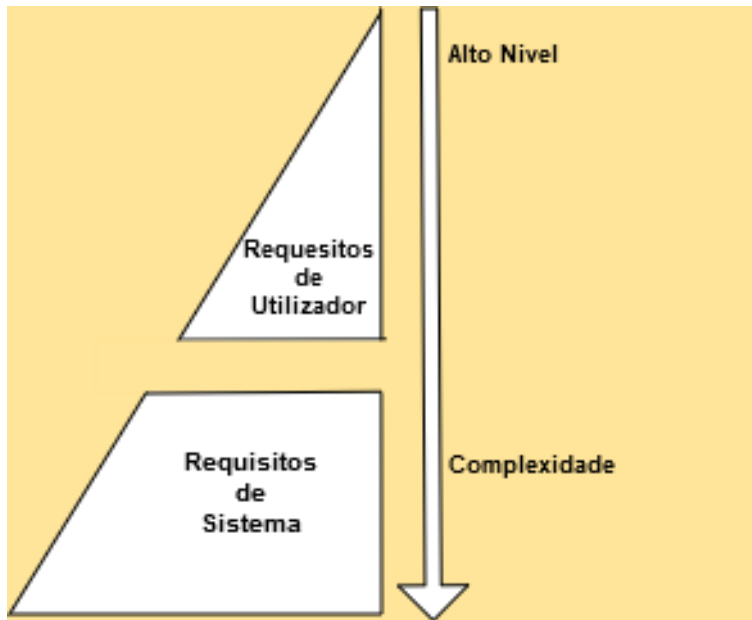


FIGURA 5- TIPOS DE REQUISITO POR NÍVEL

2.1.2. Requisitos de Utilizador

Os requisitos do utilizador são requisitos de alto nível de abstração onde é apresentado ponto de vista e objetivos permitidos pelo sistema ao utilizador. Devido ao uso de uma linguagem natural, com o auxílio de diagramas e gráficos para melhor compreensão dos serviços fornecidos pelo sistema (e as suas restrições), este tipo de requisitos é muitas vezes direcionado a ser lido pelos clientes ou *stakeholders*, devido a sua possível falta de conhecimento técnico.

2.1.3. Requisitos de Sistema

Os Requisitos de Sistema são de baixo nível de abstração, tendo assim mais em conta detalhes sobre as funções, objetivos e restrições, é utilizado uma linguagem técnica de forma a ter a descrição do requisito o mais explícito possível. Tendo em conta que é destinado a ser lido

por alguém com conhecimento técnico, como *developers* ou Programadores, este tipo de escrita permite de forma clara e sem enganos saber como, e de que maneira vai ser implementado o *software*.

Em jeito de conclusão, é necessário escrever os requisitos com diferentes níveis de complexidade, mediante os leitores em causa. Um cliente ou *stakeholder* não estará de todo interessado em saber como vai ser a implementação do sistema ou os seus detalhes, assim como a alguém que vai fazer parte do desenvolvimento do projeto lhe é supérfluo a quantidade de informação num requisito desenvolvido em alto nível.

2.1.4. Requisitos funcionais e não funcionais

Existe ainda outra forma comum de classificar requisitos:

Os requisitos funcionais, especificam funções do sistema ou alguma capacidade que o sistema deva realizar, assim como especificar o que fazer caso não aconteça algo. Podem ser diretamente tirados dos casos de uso.

Em contrapartida os requisitos não funcionais, em vez de especificar o que o sistema tem capacidade para fazer, declaram a maneira como o vai fazer. Pode também restringir o comportamento de certas funções, seja por desempenho, qualidade ou imposição de normas.

É de notar que ambos os tipos de requisitos são importantes e estão intrinsecamente ligados. A título de exemplo, seja um requisito não funcional sobre um processo de segurança, mas que com o seu desenvolvimento e detalhe origina vários outros requisitos funcionais. Esta propriedade de restrição, ou o facto de gerar outros requisitos mostra que na realidade os requisitos não são independentes. Por isso os requisitos de sistema não apenas especificam as funções ou capacidades de um sistema, mas acabam por especificar também as necessidades para garantir esses serviços e funções sejam concluídos corretamente. [5] [6]

Exemplo de Requisitos funcionais [6]:

“O sistema pode registar novas contas”

“O sistema oferece campos de pesquisa ao todo ou a um conjunto da base de dados”

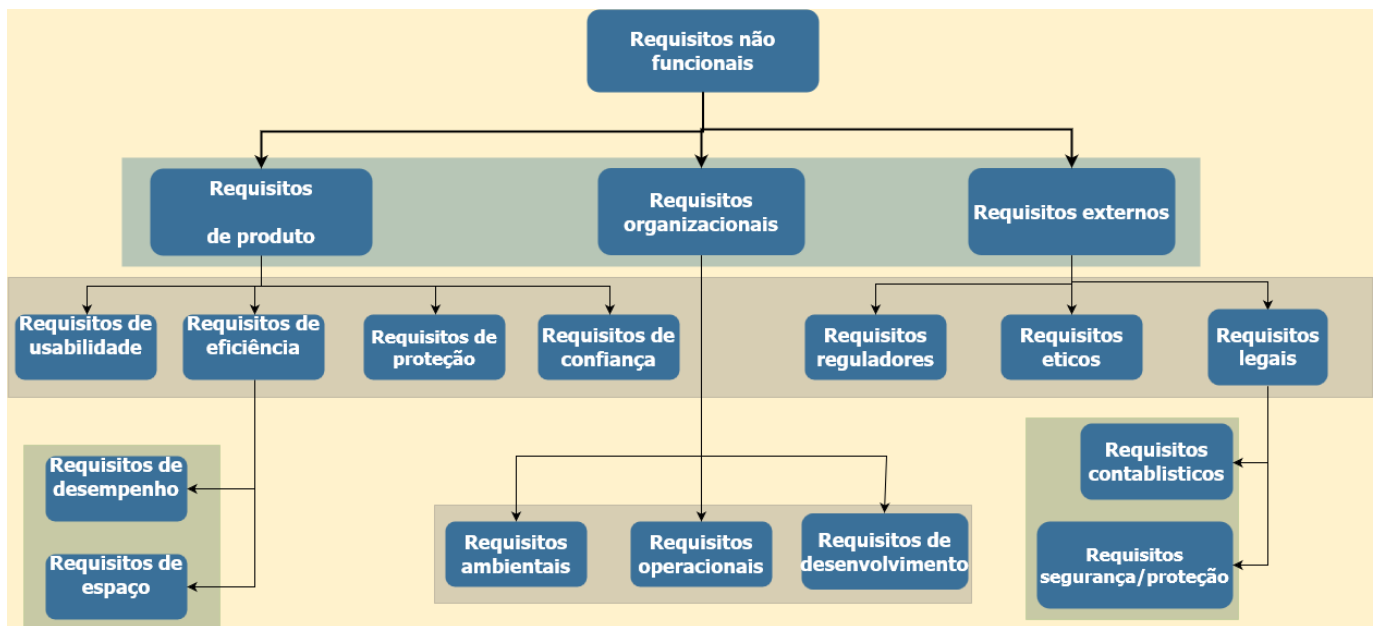


FIGURA 6- REQUISITOS NÃO FUNCIONAIS E CONSTITUINTES

Requisitos não funcionais

Tal como ilustrado na Figura 6, existem três grandes grupos no que toca à organização dos requisitos não funcionais. Se os requisitos caracterizarem ou identificarem o programa, eles são incluídos no grupo dos requisitos de produto. Por sua vez, se o requisito for baseado em normas políticas ou baseadas no modo de ação de uma empresa, chamamos de requisitos organizacionais. Requisitos externos, como identifica o nome são requisitos exteriores ao serviço, mas que têm de ser tidos em conta, seja o exemplo de normas éticas, procedimentos legais ou regulamentares [5].

Exemplos de requisitos não funcionais [6]:

“Cada vez que existe uma falha de sistema, deve ser imitado um relatório sobre as falhas”

“O sistema oferece a opção campos de pesquisa caso o acesso de segurança seja igual ou superior a developer”

“O suporte ao produto deverá ser feito exclusivamente num site *web*, sendo este uma base de conhecimento sobre o produto”.

2.2. *Software on-premise e software on-cloud*

No meio empresarial há dois tipos principais de distribuição de *software*, o *software on-premise*, o tipo de distribuição tradicional, e o *software on-cloud*.

2.2.1. *Software on-premise*

O *software on-premise* [7] é a forma tradicional de distribuição de *software*, consistindo este em ser instalado e usado nas infraestruturas do cliente, ou seja, implica fazer uso dos recursos computacionais disponíveis no lado do cliente. Por norma adquirir este tipo de *software* só implica a compra de uma licença ou cópia de *software*.

É considerado um sistema mais seguro, pois tudo acontece nas infraestruturas do cliente, filtrando assim muito do que entra e sai, pode também não necessitar de uma conexão a internet permitindo ser usado de forma offline.

A nível de desvantagens, como habitualmente implica uma compra dessa licença ou cópia de *software*, tem um custo inicial avultado. É necessário ter hardware com capacidade para correr os programas, que caso seja com tarefas muito exigentes será mais custoso. Apesar de oferecer um serviço de suporte ao cliente e a possível ajuda na integração inicial, implica uma equipa de pessoal especializado para fazer a manutenção e atualização do *software*.

2.2.2. *Software on-cloud e cloud computing*

O *software on-cloud* ou SaaS (*software as a service*) é um dos serviços disponibilizados pela “*Cloud computing*”, e como tal é melhor definir este último:

Cloud computing [8] é um conceito de distribuição de serviços (*software* e não só), que é constituído por cinco características fundamentais, integrado em quatro tipos de *Clouds* e é servido ao cliente em 3 tipos de modelos diferentes (Figura 7).

Devido a evolução tecnológica que quotidianizou o uso da internet, obtemos uma estabilidade e velocidade na rede que tornou cada vez mais o *cloud computing* uma solução aliciante.

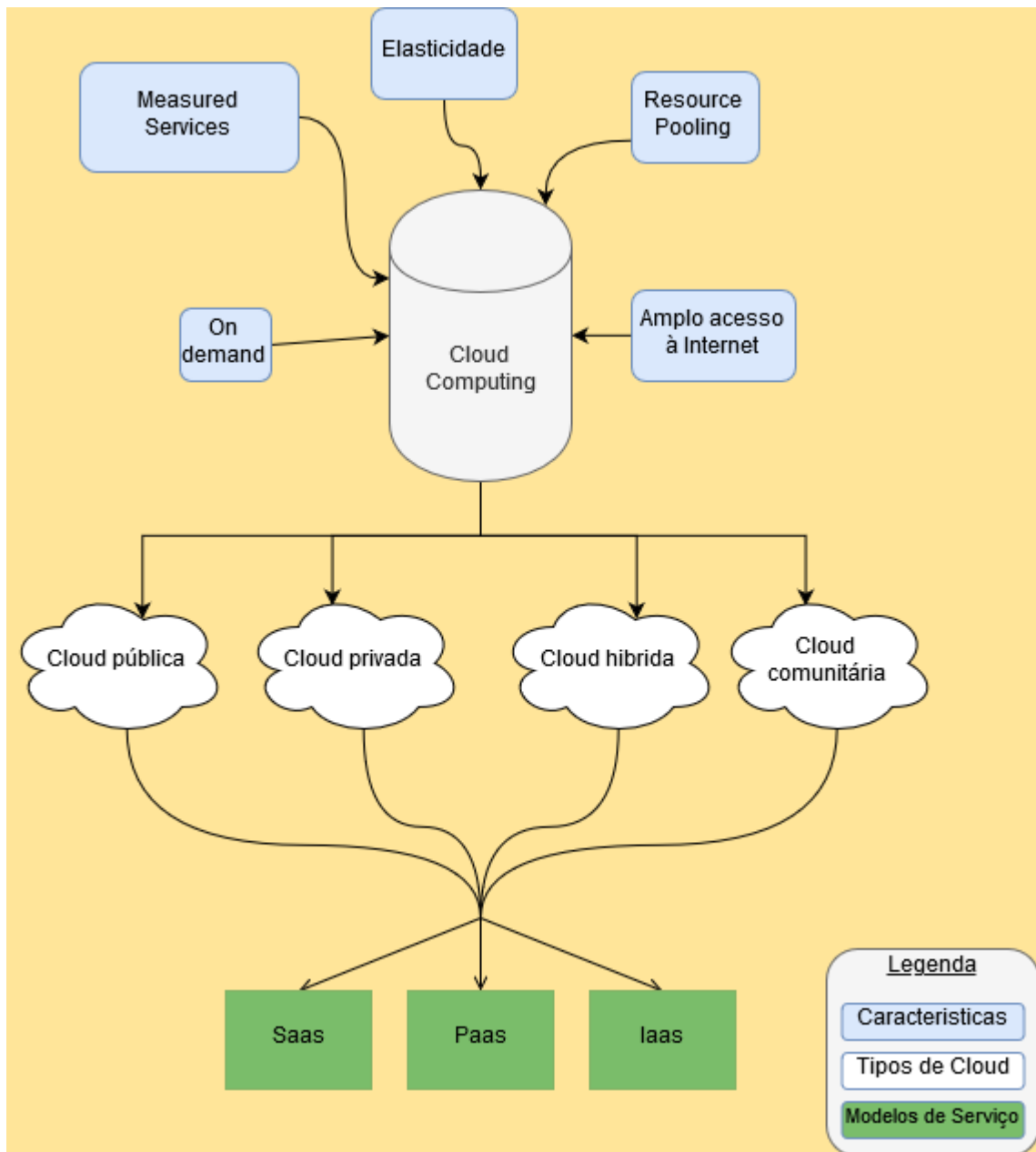


FIGURA 7- CLOUD COMPUTING, CARACTERÍSTICAS, TIPOS E MODELOS

A principal característica que define *cloud computing* [8], é a forma como pode ser obtido. O chamado “*on-demand*” ou “a pedido” evita o período de espera de dias até obter uma cópia/licença de *software* (como no caso do *software on-premise*), para alguns minutos ou horas, e como é praticado um sistema de “*pay per use*” é evitado o avultado custo das distribuições de *software on-premise*.

Devido a estar hospedado em uma nuvem, os sistemas aí embutidos controlam e otimizam os recursos usados no momento por meio de um recurso de medição num nível de abstração

relativamente ao tipo de serviço desempenhado, seja no controlo de contas ativas ou de uso excessivo de memória por serviços inativos (*measured services*).

No caso do *software* tradicional se fosse necessária mais memória ou poder de processamento iria ser necessário fazer um *upgrade* ao *hardware*, envolvendo custos e implementações. Na eventualidade desta necessidade ter sido ultrapassada e ter recursos a mais, esses mesmos recursos ficariam por usar, mas o custo monetário e técnico já tinha sido gasto. A elasticidade da nuvem pretende colmatar exatamente esse problema, caso estejam a ser precisos mais recursos, o sistema é facilmente escalável, até servir os recursos necessários, assim, como diminuir os recursos presentes no sistema quando estes deixam de ser precisos (sendo que mais recursos implicam um custo superior para o seu uso).

O chamado *Resource Pooling* consiste na partilha de recursos de sistema entre os vários clientes, fazendo com que aumente a taxa de uso dos mesmos, o que significa que menos recursos ficam no inativo fazendo então render mais os recursos disponíveis. É uma característica que permite ao *software* "particionar-se" de forma a servir vários clientes ao mesmo tempo.

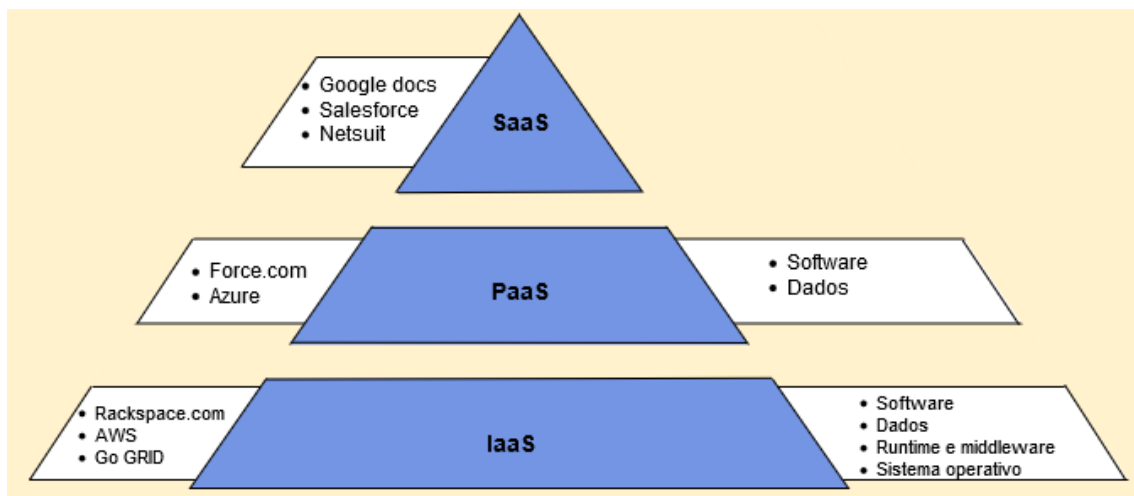
Por fim, o amplo acesso à internet (*Broad Network Access*) possibilita um tipo de acesso totalmente distinto aos demais tradicionais, onde tanto é possível trabalhar num *desktop*, *laptop*, *tablet* ou *smartphone* desde que tenha uma ligação à internet. Existindo uma, é possível aceder a todo o conteúdo e serviços num só sítio em qualquer lugar.

No que toca aos diferentes tipos de *clouds* [8] [9]:

- *Cloud* Pública em que toda a infraestrutura está localizada no lado da companhia da fornecedora da nuvem, sendo que esta localização está separada do cliente sem este ter modo físico de controlo sobre ela. Enquanto que bastante bom a nível de performance são mais vulneráveis a vários ataques.
- *Cloud* Privada é muito idêntica à *Cloud* Pública, mas existe um servidor dedicado para cada um dos clientes, existe também a possibilidade de ter uma *cloud* "on-premise", ou seja, no local, se a empresa assim entender o investimento. A migração é facilitada devido a centralização de dados, necessitando só a mudança física da *Cloud* para as instalações da empresa.

- *Cloud Híbrida* é uma junção das duas anteriores, em que existe uma *Cloud* Pública designada para operações com um baixo nível de risco, como entrar em contacto com clientes, e uma *Cloud* Privada quando a necessidade de segurança for maior, como manter dados sensíveis seguros.
- *Cloud Comunitária* parte do mesmo paradigma da pública, no entanto todas as organizações ou empresas que usam esta *cloud* são idênticas, ou de objetivos parecidos, fazendo com que seja possível associar comunidades a *clouds*.

No que toca aos modelos de distribuição (Figura 8) são destacados três principais tipos [11].



Estes são os mais abrangentes porque, por mais que haja outros tipos de modelo com pequenas adições ou diferenças, normalmente podem ser sempre incluídos nos principais modelos, sejam estes:

FIGURA 8- MODELOS DE DISTRIBUIÇÃO DE CLOUD COMPUTING

- *Software as a service* (SaaS) [11]: Pode ser considerada a evolução natural da distribuição de *software*, se considerarmos que a distribuição anterior por meios físicos (*software on-premise via cd-rom ou DvD*) foi durante muito tempo a única opção viável. Esta distribuição devido à falta de uma infraestrutura que conseguisse viabilizar um modelo como o SaaS, mas com a evolução, estabilidade e velocidade da internet foi permitido reunir condições para esta ser um modelo convencional. A única “responsabilidade” do utilizador é mesmo fazer uso do *software* e serviços

disponibilizados pelo fornecedor, sendo que tudo o resto é tratado pelo fornecedor, desde segurança, atualizações e suporte.

O modelo *SaaS* tem como público alvo os utilizadores finais, portanto níveis como infraestrutura ou arquitetura do sistema estão vedados ao utilizador. Estas restrições resultam em uma falta de liberdade ou soluções em certos casos, mas também deixando o sistema mais *user friendly* para ser operado.

- *Platform as a service (PaaS)* [10]: Este modelo é conceitualmente descrito sendo uma plataforma de computação que fornece uma solução integrada, conjunto de soluções ou serviços por meios de uma conexão de internet. Muitas das vezes estes conjuntos de soluções ou serviços podem ser componentes ou subsistemas usados no desenvolvimento de um produto ou aplicação. Seja por exemplo o caso de uma aplicação *web*, no seu desenvolvimento é necessário o suporte de um sistema operativo, um servidor, bases de dados e o *middleware* necessário ao desenvolvimento. É isto o oferecido pelo modelo PaaS, uma plataforma para o desenvolvimento e construção, portanto direcionado para *Developers*.
- *Infrastructure as a service (IaaS)* [10]: Pode ser também denominada por *Hardware as a service (Haas)*, este modelo tem como objetivo facilitar o acesso e fornecimento de recurso usados nos modelos previamente introduzidos. Seja hospedar servidores, armazenamento ou componentes de redes, são aqui situados os recursos usados pelos outros modelos, podendo em alguns casos servir também algum tipo de *software* se necessário. O público alvo deste tipo de modelo são arquitetos de *software*, onde a liberdade nos modelos de *SaaS* e *PaaS* não é suficiente para as necessidades ou fins objetivados, é também o modelo que põe mais recursos e serviços na responsabilidade do utilizador.

2.3. Salesforce

Salesforce é [12] uma CRM (*Customer Relationship Manager*) que oferece várias soluções em *cloud*, sendo por isso distribuído como um modelo de *Software as a service* (Saas). O conjunto de serviços oferecidos pela *Salesforce* permitem um maior grau de contacto e satisfação com o cliente, ajudando a fechar propostas ou a identificar novas possíveis, tudo isto com o suporte total de um serviço em *cloud*.

Alcançou mediatismo devido à inovação de ser das primeiras plataformas CRM em *cloud*. Esta liderança evoluiu para a atribuição de diversos prémios [13], nomeando 4 anos consecutivos de “*Forbes’ World’s Most Innovative Company* (Forbes: 2011, 2012, 2013, 2014)”, mais recentemente (2019) “*FORTUNE* 100 *Best Companies To Work For*”, segundo classificado em “*LinkedIn Top Companies*”. Para além dos prémios recebidos também podemos reconhecer o *Salesforce* estatisticamente, em uma página inicial do site oficial temos publicitado “crescimento em média de 27% nas vendas, 32% na conversão de “*leads*”, 34% de satisfação de cliente [12]”. Outros valores estatísticos [14] como 19.6% do mercado de crm’s global e 83% das empresas da *Fortune 500* (quinhentas maiores empresas dos estados unidos) serem clientes *Salesforce*. Foi também reportado em 2017 como tendo mais de 150.000 companhias a usar os seus serviços [12].

Na Tabela 1 são ilustradas algumas vantagens e desvantagens de usar *Salesforce*.

TABELA 1- VANTAGENS E DESVANTAGENS SALESFORCE

Prós	Contras
<i>Multitenancy</i>	Pequenas equipas podem ficar “ <i>overwhelmed</i> ” com a quantidade de recursos disponível
Seguro e Confiável - É um dos seus principais princípios	Grande ênfase no <i>deployment</i> em <i>cloud</i> , não existindo outros métodos de distribuição
Multiplataformas	Cada nova atualização altera bastante o <i>layout</i> de trabalho
Grande elasticidade	
Instalação (<i>deploy</i>) relativamente rápido	
Loja confiável onde é possível encontrar aplicações capazes de aumentar a produtividade	
<i>Low Skill floor - High Skill ceiling</i>	

Serviços *Salesforce*

A forma como o *Salesforce* pretende chegar ao cliente é pela forma de diferentes *clouds* (ainda que interligadas entre elas), cada uma oferecendo diferentes tipos de serviços e soluções (Figura 9).



FIGURA 9- CLOUDS NO SALESFORCE

FONTE: LOBA.PT

Sales Cloud

É também chamada de “*Salesforce crm*” pois é a parte mais virada para com as necessidades do cliente, como tal pode ser usada para ver o crescimento de negócios, encontrar novos clientes (*leads*) e fechar propostas. Permite também gerir oportunidades, gestão de “*leads*”, relatórios e *dashboards* (gráficos em tempo real).

Service Cloud

Enquanto que a *Sales cloud* tem como foco o negócio do cliente, a *service cloud* detém as ferramentas para melhorar o contacto com o cliente, como atendimento personalizado, seja por e mail, vídeo chamada ou telefone, e devido a ter todos os detalhes relevantes em relação a qualquer caso, é possível ir direto ao assunto o que melhora a produtividade da empresa.

Marketing cloud

É uma plataforma automatizada de *marketing* que pretende distribuir por diferentes canais, diferentes tipos de publicidade, de forma a esta ser o mais eficiente possível em públicos alvos diferentes. Seja por emails publicitários de cariz promocional, ou por mensagem ou notificações *push*, tudo num ambiente que permite mandar conteúdo relevante para a audiência certa de forma a gerar compromisso ou interesse.

Community Cloud

É o local onde se podem juntar todos os intervenientes e fazer uma comunidade, juntando pessoas relevantes ao negócio, pessoas que possam ter dúvidas sobre o produto e pessoas que possam tirar essas mesmas dúvidas. A possibilidade de juntar toda a gente facilita a colaboração e partilha de informações, melhorando a produtividade dos trabalhadores e satisfação do cliente.

Analytics Cloud

As análises ao produto são essenciais para o sucesso e para uma boa gestão de um negócio, estas análises permitem melhorar o desempenho ou a qualidade dos produtos. Como a *Salesforce* trabalha num ambiente centralizado, isto é, os dados encontram-se todos no mesmo sítio, podendo ser acedidos de diferentes modos e locais. Isto permite que em qualquer momento ou local seja possível aceder e analisar dados, seja a nível de campos repetidos, nulos ou até em alguns casos onde a verificação falhou.

Com esta análise é possível resolver muitas vezes problemas complicados e difíceis de identificar. É possível ser utilizada em qualquer dispositivo, e o resultado da análise pode ser de diferentes níveis de abstração.

App Cloud

É uma extensão do *Salesforce1* (aplicação mobile) que tem o intuito de juntar a plataforma *mobile*, *social* e *cloud* em uma só para os *developers*. É possível utilizar plataformas como o *Heroku* para desenvolver e fazer *deploy* a aplicações nas mais variadas línguas e tecnologias. É integrado também com a *Salesforce Lightning mobile framework*, que permite fazer um desenvolvimento conjunto de uma aplicação que funcione em diferentes dispositivos.

Tem também uma “*App Store*” onde é possível descobrir e comercializar aplicações previamente desenvolvidas que podem servir as necessidades, em vez de desenvolver uma cada vez que é necessário.

Commerce Cloud

É uma solução de comércio B2C (*Business-to-consumer*) ou B2B (*Business-to-Business*) que consiste numa plataforma de comércio com diferentes clientes, que capacita marcas a criar experiências de compra em diferentes canais, como o *mobile*, *web* e lojas.

3. Metodologia

3.1. Metodologia Agile – Scrum

Um processo de desenvolvimento de *software* é um conjunto de atividades parcialmente ordenada que se relacionam entre si e tem como finalidade obter um produto final, é um processo pertencente a Engenharia de *Software* sendo também um importante fator para obter *software* de qualidade. Estes diferentes métodos podem ser agrupados em duas grandes metodologias, clássica e Ágil.

A metodologia Ágil ao contrário da Clássica, não se identifica por apresentar a integridade de todo o produto, por uma análise completa ou pela definição de todas as categorias e requisitos. Invés disso, defende que se deve procurar por satisfazer o cliente através de entregas contínuas de *software* com valor, mantendo a comunicação constante com o cliente, e de mesmo modo mantendo a comunicação com a equipa. Algo comum na metodologia Ágil é a tentativa de minimizar o risco de desenvolvimento de *software* em curtos espaços de tempo, chamado de iteração. Uma iteração tem normalmente duração de um período entre uma a quatro semanas, onde cada iteração é tratada como uma parte do projeto total.

O *Scrum* foi o método Ágil utilizado no desenvolvimento dos projetos, sendo este constituído por várias etapas (Figura 10):

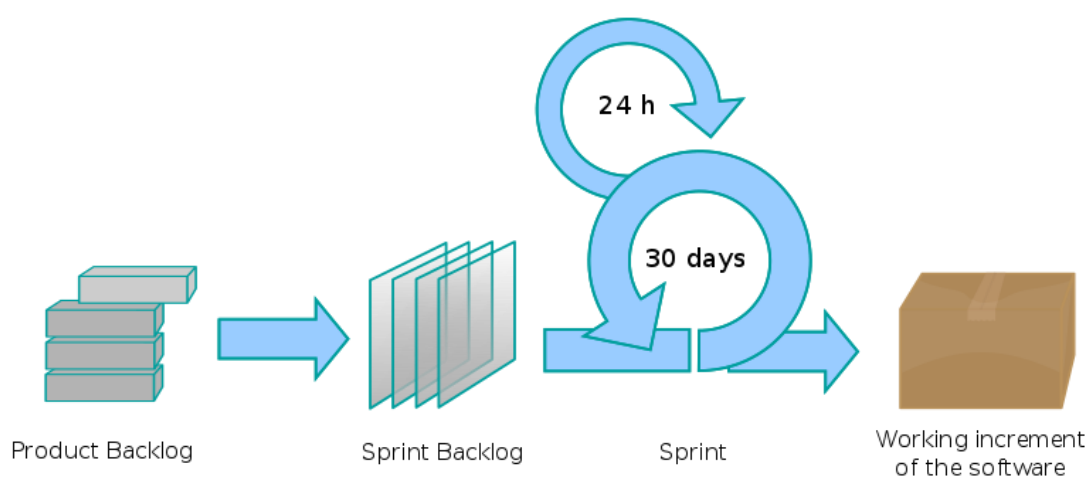


FIGURA 10- MODELO SCRUM

FONTE: WIKIPEDIA.ORG

Product Backlog, é uma lista ordenada de itens ou requisitos (funcionais e não funcionais) à responsabilidade do “*Product owner*”, sendo este que a mantém e organiza as suas prioridades (e a re-prioriza se necessário) assim como é definida a carga horária para cada requisito.

Sprint Backlog que pretende transformar os requisitos com maior nível de prioridade do *Product Backlog* em tarefas, definindo metas e formas de execução, para isto é feita uma reunião geral, onde também é debatido e agendado as tarefas a serem começadas no próximo *Sprint* (*Sprint Planning*).

Sprint é a fase de desenvolvimento, onde vão ser feitas as tarefas previamente definidas para o atual *sprint*. Existe muito a ideia de um quadro dividido em várias colunas sendo estas: “*Itens de Backlog*, Por fazer, A fazer e Feito!”, onde são colados papéis com as tarefas definidas e a posição destes vai sendo alterada consoante o seu estado de desenvolvimento. O *sprint* por norma dura entre uma a quatro semanas, e durante esta etapa é feita uma reunião diária (*daily scrum*) de forma a notificar o *Scrum Master* (normalmente o gestor do projeto) sobre algum previsto ou impedimento encontrado no desenvolvimento. No final de cada *Sprint* é feita mais uma reunião geral, onde é readaptado o *Product Backlog*. Se for necessário mediante as informações disponíveis, é também planeado o próximo *Sprint*.

Durante o estágio, foi definido um *Product Backlog*, o qual foi sofrendo algumas alterações, mas que permitiram o planeamento das tarefas seguintes. Foram realizadas também breves reuniões diárias, ou de dois em dois dias, mediante a disponibilidade do orientador, onde analisamos o que tinha sido desenvolvido, discutindo soluções e correções assim como as tarefas a priorizar nos seguintes dias.

Foi considerado durante o estágio três Sprints principais:

- O primeiro, foi a análise de requisitos em contexto real;
- O segundo, a análise de requisitos ao meu projeto a implementar;
- O terceiro, a implementação em *Salesforce* do meu projeto.

Sendo uma das vantagens do *Scrum* desenvolver pedaços “com valor” de um total, houve situações onde apesar de já me encontrar a trabalhar em tarefas de um novo *Sprint* ter de

voltar ao previamente feito para nova análise e retificação de alguns pontos. Este recuo acontecia principalmente após reuniões de “*daily scrum*” no dia, ou pela presença do CTO que também supervisionava os projetos e fazia por estar a par dos desenvolvimentos dos mesmos. Estas últimas reuniões podem ser consideradas uma “*week Scrum*”, apesar de não acontecer especificamente todas as semanas, podendo por vezes ocorrer mais do que uma vez por semana ou nem chegar a acontecer.

3.2. Trailheads

Um *Trailhead* refere-se ao ponto inicial de um trilho, e é por isso um bom nome escolhido para a página de tutoriais de *Salesforce*. Muitas vezes a palavra tutoriais é usada para referir coisas simples e introdutórias, apesar de também podermos encontrar tutoriais de principiante nos Trailheads estes têm diferentes graus de dificuldade. Podemos encontrar desde as primeiras introduções ao *Salesforce*, como usar os menus e afins, até aos super-emblemas, que podem demorar meses a fazer e implicam um conhecimento técnico avançado. As conquistas do utilizador são apresentadas num perfil por meio de uma pontuação e medalhas (*badges*) (Figura 11).

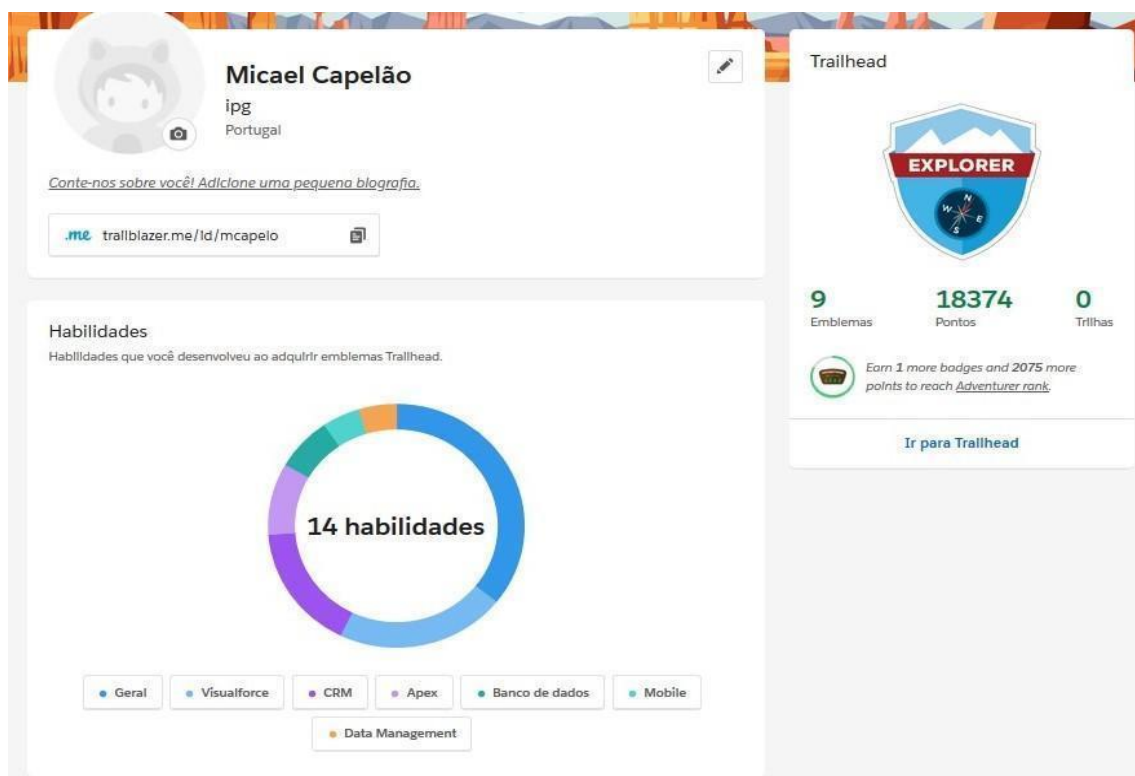


FIGURA 11- PERFIL *TRAILHEADS*

Os *Trailheads* criam uma Org (organização) fictícia (chamada de *playground*) onde o utilizador pode experimentar o que quiser sem correr riscos de alterar algo numa Org real de um cliente, podendo então aprender com os erros de uma forma segura. São introduzidas matérias, com um breve questionário no fim para confirmar a aprendizagem de conceitos. De seguida é colocado um problema em contexto prático sobre o conceito introduzido, na forma de um desafio a ser implementado na Org. Este por sua vez vai ser confirmado e avaliado pelo sistema, atribuindo então uma pontuação (mediante a dificuldade do *Trailhead*). Após vários *Trailheads* do mesmo tipo, mas de incremental dificuldade, é recebido uma *Badge* (emblema) como reconhecimento perante as aptidões aprendidas pelo utilizador relativamente ao tema em que estava a trabalhar (Figura 12).

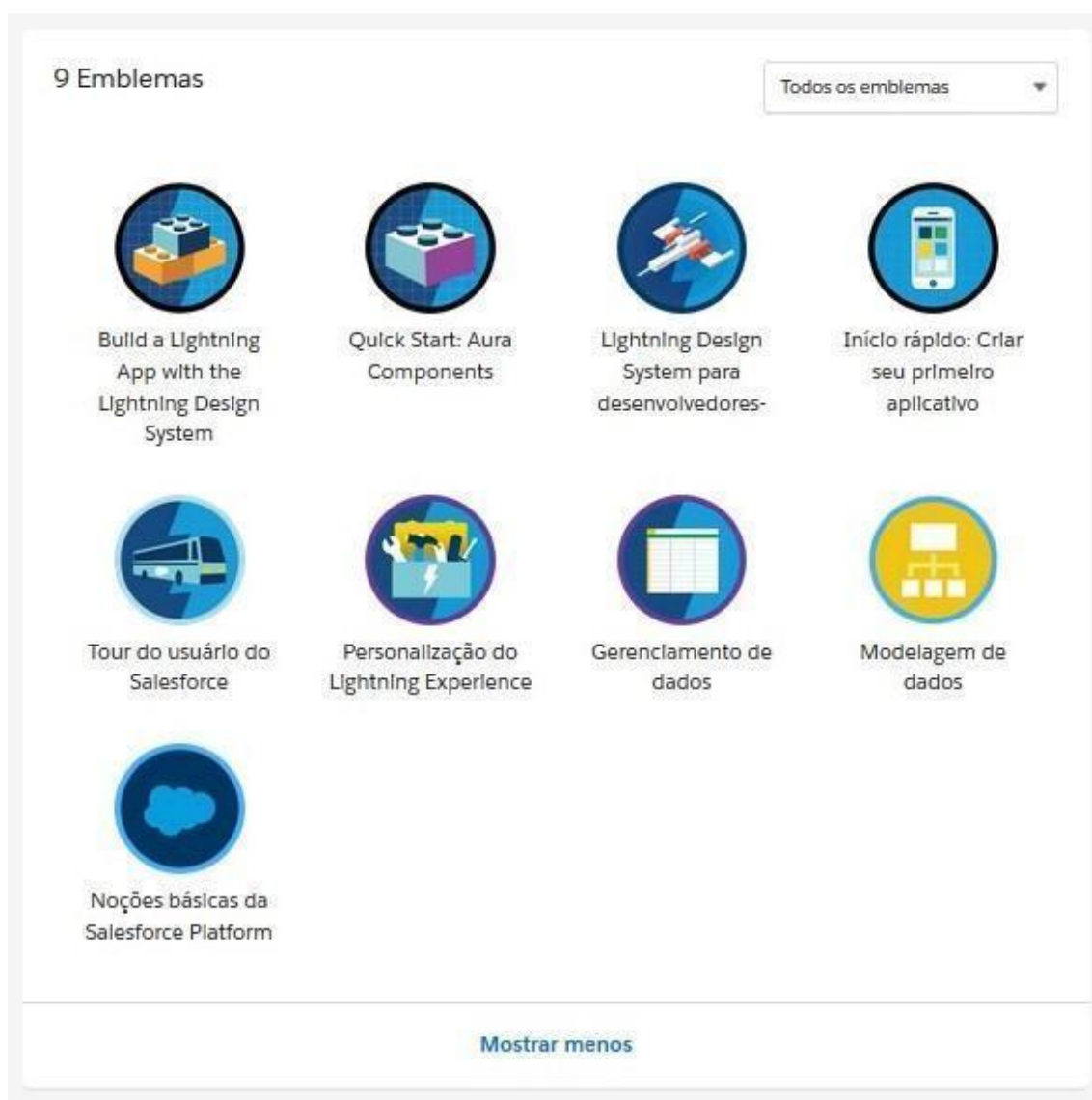


FIGURA 12- BADGES *TRAILHEADS*

Devido à imensidão de serviços e funções existentes no *Salesforce* era necessário este tipo de plataformas de aprendizagem pois para além de ser uma ótima maneira de centralizar informação e documentação técnica, a forma de aprender é tida como um desafio e a dificuldade gradual aumenta a vontade de aprender. Para além de melhorar o perfil (Figura 11) e apresentar outro tipo de badges, esta melhoria de perfil tem algumas vantagens profissionais, apesar de não tão “sério” como um certificado em alguma nova tecnologia, este pode ser partilhado e associado ao perfil do *LinkedIn* de forma a complementar o grau de capacidade e conhecimentos sobre *Salesforce*.

3.3. Formação *Salesforce* presencial

Como forma de alcançar alguns dos objetivos definidos no estágio, nomeadamente desenvolver e aplicar conceitos em *Salesforce*, houve a participação presencial na formação de *Salesforce* nas instalações da Dom Digital. Esta formação era de duas horas diárias (não fazendo parte das horas regulamentares do estágio), onde diferentes profissionais da empresa abordavam assuntos e tecnologias com as quais trabalhavam diariamente ou conceitos necessários para próximas sessões, portanto lecionada de forma contínua. Em fases iniciais foi introduzido os conceitos de CRM's, *Clouds* e *Salesforce*, assim como objetivado a mais valia de uma certificação em *Salesforce*.

Como forma de apoio foi também usado um *eBook* [15], “*Lightning Platform Fundamentals*” onde era apresentado o desenvolvimento de um projeto genérico abordando grande parte das funcionalidades da plataforma em uso. Seguir este livro ajudou em muito no aumento de agilidade e ambientação em relação ao *Salesforce*. Foram também abordados temas mais específicos como *Lightning componentes*, *Salesforce1*, *Marketing Cloud* e *Pardot*.

Ter Profissionais que para além do ponto de vista da formação que estão a dar, também trabalham diariamente com esse tipo de área, dá uma perceção diferente e muito elucidativa das diferenças entre o teórico e as aplicações práticas dos conceitos, foi sem dúvida uma experiência enriquecedora e que ajudou em muito no desenvolvimento do projeto.

4. Análise de Requisitos

Já foi explicitado previamente o que é a Engenharia de Requisitos dentro da Engenharia de *Software*, assim como foi denotada a sua importância. Como tal é importante perceber os estágios pelo qual a Engenharia de requisitos passa, de maneira a melhor perceber quando e como o trabalho é desenvolvido.

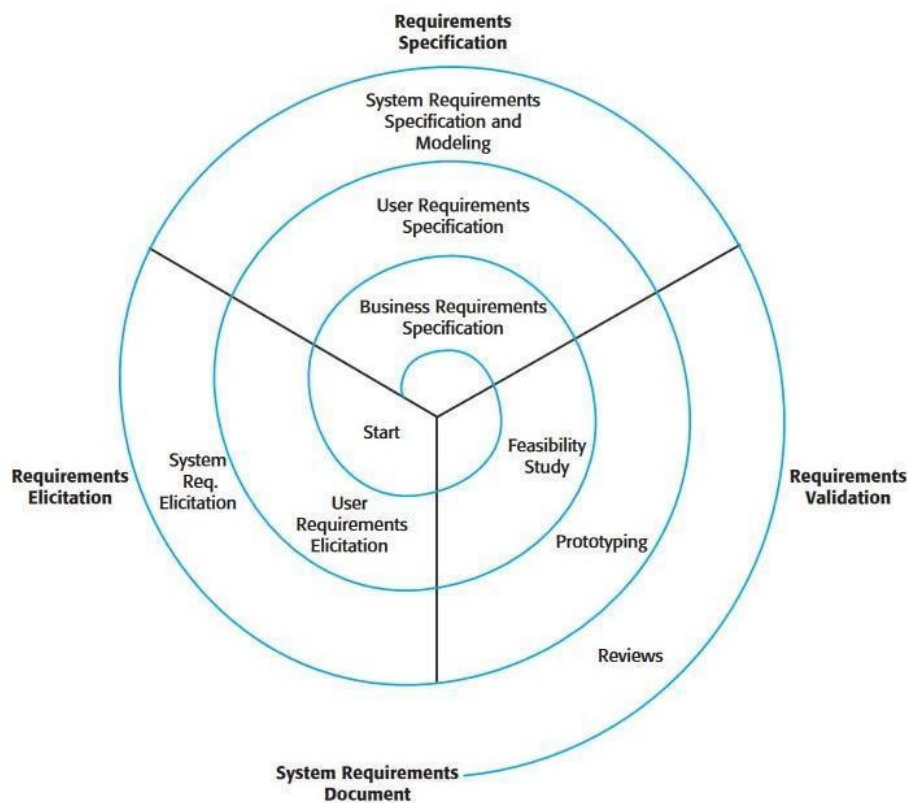


FIGURA 13-ETAPAS CONSTITUINTES DA ENGENHARIA DE REQUISITOS

A Engenharia de requisitos, apesar de se encontrar dividida em várias etapas diferentes, não se resume em acabar uma das etapas e seguir para a próxima. Esta é um processo iterativo [2] como é representado na figura 13, várias vezes a mesma etapa pode e deve ser repetida dependendo do nível de complexidade ou abstração pretendido. Podemos destacar quatro atividades de alto nível, sendo a primeira a que permite avaliar se o sistema deve sequer chegar a ser debatido, isto é, por meios de um estudo de mercado/viabilidade económica chega-se à conclusão se o projeto é ou não uma mais valia para a empresa, caso este seja positivo, podemos então dizer que na realidade o processo de engenharia de requisitos começa aqui.

As primeiras iterações vão ser de um nível de abstração mais alto, de forma a saber o que faz o sistema, quais as suas funções e serviços, antes de definir de que modo vão ser cumpridas essas tarefas.

É então feito o levantamento de requisitos (normalmente de alto nível, os “*user cases*” com alto nível de abstração, e menor complexidade técnica), apesar de existirem várias formas, as principais maneiras de fazer este levantamento de dados é reunindo com os stakeholders. Em reunião tenta-se ao máximo definir o sistema de forma a que satisfaça as necessidades do cliente. Caso não seja possível reunir sempre com os stakeholders, também é possível fazer este levantamento de uma forma mais informal por via de questionários ou utilizando documentação existente.

Com o resumo obtido do levantamento de informação, vamos praticar a Análise de Requisitos [16] propriamente dita, vão ser definidos e explicitados cada uma das necessidades referidas pelo cliente. Como referido é importante definir o que o sistema deve fazer nesta etapa, invés de procurar maneiras de como este o vai fazer.

De forma a assegurar um desenvolvimento processual ordenado, deve-se realizar nesta fase uma etapa de validação onde se pretende concluir se os requisitos definem o sistema que o cliente realmente quer. Caso haja a incerteza, no que toca a servir as necessidades do stakeholder, existe a possibilidade de fazer um protótipo.

As iterações continuam a aumentar o nível de complexidade, chegando por isso a ser debatido com mais detalhe os *system requirements*, assim como começar a classificar e organizar os requisitos. Após a sua listagem são debatidas as soluções para desenvolver o sistema na próxima etapa assim como a priorização e negociação dos requisitos.

São feitas as iterações necessárias até todos os requisitos estarem bem definidos e no final de todo este processo é gerado um documento de requisitos [16]. Este documento pretende

Especificação de Requisitos de Software
1. Introdução
2. Descrição Geral
3. Características Do Sistema
4. Requisitos de Interface Externo
5. Requisitos não funcionais
6. Outros Requisitos

apresentar toda a informação de um modo explícito e estruturado (Figura 14), de forma a ser um guia de suporte ao desenvolvimento e manutenção do sistema.

FIGURA 14- ESTRUTURA TIPO DE UM DERS
FONTE: ISCAC.PT

4.1. Caso estudo em contexto real – Bricomarché

No caso de estudo em contexto real, foi-me apresentada uma proposta do Bricomarché onde se pretendia implementar um serviço de reparações e outro de encomendas, em adição aos serviços que eles já prestavam. A primeira etapa da Engenharia de requisitos tinha já sido feita neste ponto, onde por meio de algumas reuniões que o meu supervisor de estágio tinha tido com os *stakeholder*, foi elaborado um resumo e a partir daí foi iniciado o processo de explicitar e desenvolver os requisitos.

4.1.1. Lista de requisitos – Bricomarché

É de seguida listado os requisitos que foram deduzidos por mim mediante os apontamentos e resumo da reunião com os *stakeholders*. Foi também fundamental reunir e discutir certos pontos para uma melhor e correta explicitação dos requisitos, pois algumas dúvidas ou incoerências foram surgindo devido à falta de contexto em alguns casos ou à natureza dúbia de tentar explicitar algo a partir de um resumo, ainda que objetivo, de uma terceira pessoa.


É descrito em listas diferentes os dois casos, uma para a etapa de reparação e outro para a de encomenda, pois era assim que os clientes pretendiam que fosse tratado.

Denoto que não deve ser considerado os valores inseridos nas horas, estes são só uma estimativa, como forma confirmar o funcionamento do cálculo da orçamentação no final de cada lista. Estimar um período temporal para o desenvolvimento de um requisito, é uma tarefa que exige bastante conhecimento na plataforma em uso assim como experiência de cariz profissional.

4.1.1.1. Lista de requisitos – Reparação

Foi o primeiro caso analisado, tratava em criar um serviço de reparação para itens previamente adquiridos em uma das lojas do grupo. Tinha como foco eliminar falhas humanas, quer na falta de informação ou a falha de resposta atempada, assim como possibilitar uma forma de organização informática, diminuindo o uso do formato físico de papel (Tabela 2).

TABELA 2- LISTA DE REQUISITOS BRICOMARCHÉ ETAPA REPARAÇÃO

							
Brico - Requirements Summary - Work Version 1.0							
Autoria: Micael Capelão							
Objetivos - Eliminar papel - e ter muitos processos automáticos Eliminar falhas							
Req ID	Requirement	Brought Up By	Comments and Questions	Solution Type	Requirement Status	Priority	Hours
1	Gestor de Reparações	Cliente			Open	Must Have	
1.1	Como ATOR (Gestor de Reparações) a ficha de reparações a ser preenchida deve ter várias áreas: - Cabeçalho - Identificação do PDV - Identificação Cliente - Identificação do Artigo - Datas - Anexos	Cliente					
2	Cabeçalho	Cliente					
2.2	No cabeçalho está o ID Reparação assim como a área designada para a etiqueta de reparação (posteriormente gerada)	Cliente			Open	Must Have	2,0
3	Identificação do PDV	Cliente			Open	Must Have	3,0

3.1	Área designada para identificar o ponto de venda tem os campos de: - Número de Ponto de venda - Localidade - Data de preenchimento - Preenchimento da guia por	Cliente	Número de PDV assim como Localidade são facultados pelo cliente. Data preenchimento é colocada automaticamente no dia em que a guia de reparação é concluída. O campo "Preenchimento da Guia por" será automaticamente o utilizador a preencher a guia de reparação em causa.		Open	Must Have	4,0
4	Identificação do Cliente	Cliente			Open	Must Have	3,0
4.1	Pretende identificar o cliente que solicita a reparação: - Nome - Morada - Contacto	Cliente			Open	Nice To Have	
4.2	Automaticamente preencher os campos acima referidos pelo número de sócio / cartão mosqueteiro	Dom Digital	Abrir a possibilidade de agilizar o processo de idêntica o cliente, assim como ver algum histórico de reparações e (se necessário) compras. Torna o cliente alguém de algum modo afiliado a empresa, algo mais do que um simples cliente da reparação		Open	Must Have	2,0
5	Identificação do Artigo	Cliente			Open	Must Have	2,0
5.1	Campos necessários para identificar o artigo: - ITM8 - EAN13 - Numero de serie do equipamento	Cliente	ITM8 número interno identificativo EAN13 Número identificativo de barras		Open	Must Have	2,0

	<ul style="list-style-type: none"> - Descrição - Número Factura/Talão - Data venda - Descrição da Avaria - Acessório - Apresentação do artigo 		<p>Numero de serie especifico do modelo , facultar sempre que possível</p> <p>Descrição - breve descrição gráfica do objeto, como marca, modelo, cor, tamanho relativo .</p> <p>Número da Factura/Talão - número da fatura talão emitido no ato de compra</p> <p>Descrição da Avaria - deve ser curta e sucinto na forma a explicar o problema.</p> <p>Acessórios - Checkbox</p> <ul style="list-style-type: none"> - Tem/nao tem acessórios - Se Sim,quais? <p>Apresentação do artigo - Checkbox</p> <ul style="list-style-type: none"> -Novo -Usado -Riscado/Amolgado -Muito Danificado <p>Checkbox - Caixa de origem ? (S/N)</p>				
6	Datas	Cliente			Open	Must Have	
6.1	<p>Área a apresentar todas as datas, algumas abertas outras automáticas:</p> <p>Automáticas-Fechadas:</p> <ul style="list-style-type: none"> ->Entrega do artigo para reparação ->Devolução do artigo depois de reparado <p>Editável - possibilidade manual</p> <ul style="list-style-type: none"> ->Entrega do bem em loja (reparado/trocado) e a 	Cliente			Open	Must Have	

	assinatura do PDV ->Data e Assinatura quando o cliente entrega o produto para reparação ->Data e Assinatura do cliente quando recebe o bem reparado/trocado							
7	Condições	Cliente			Open	Must Have		
7.1	Condições Serviço pós venda	Cliente	Condições sobre as quais assenta o acordo de reação do bem entre a reparadora e o cliente		Open	Nice To Have		
8	Anexos	Cliente			Open	Must Have		
8.1	No campo de anexos deverá ser possível adicionar documentos externos que auxiliem a reparação, é o caso: - Fotografia do Bem em causa - Faturas e/ou garantias - Seguros se existir	Cliente			Open	Nice To Have		
9	Observações	Cliente			Open	Must Have		
9.1	Pos-preenchimento dos dados é imprimido uma guia de reparação que consiga identificar pontos fulcrais e únicos da reparação em causa, guia esta que deverá ser colocada no cabeçalho	Cliente						
						Total Consulting	18	
						Req Analysis	8	
						Project Management	4	
						Training	16	
						Data Migration	4	


						QA / Tests	4
						ALM	0
						TOTAL PROJECT (h)	53
						Hourly Rate	5
						Overheads (€)	
						Discounts (%)	
						TOTAL PROJECT (€)	266

4.1.1.2. Lista de requisitos – Encomenda

No segundo caso analisado, era necessário criar um serviço que possibilitasse a encomenda de um ou mais artigos, em diversas quantidades, a um fornecedor. Visava possibilitar ao cliente escolher vários artigos de que precisasse e efetuar a respetiva compra, e consequentemente ser notificado aquando a sua chegada dos fornecedores. Foram então estendidos e explicitados os requisitos deduzidos pelo resumo da reunião com os *stakeholder* (Tabela 3).

Para além dos objetivos referidos no caso anterior, de reduzir falhas humanas e diminuir no uso de papel, este serviço tinha também como objetivo agilizar as vias de comunicação com as partes intervenientes, seja datas estimadas pelos fornecedores, como notificar o cliente da receção de artigos. Podemos identificar a forma como se davam estas relações e serviços no Diagrama de contexto da Figura 16.

TABELA 3- LISTA DE REQUISITOS BRICOMARCHÉ ETAPA ENCOMENDA

							
Brico - Requirements Summary - Work Version 1.0							
Autoria: Micael Capelão							
Objectivos - Eliminar papel - e ter muitos processos autom ^o aticos Eliminar falhas							
Req ID	Requirement	Brought Up By	Comments and Questions	Solution Type	Requirement Status	Priority	Hours
1	Gestor de encomendas	Cliente			Open	Must Have	
1.1	Como ATOR (Gestor de Encomendas) a ficha de encomendas a ser preenchida deve ter várias áreas: - Identificação Cliente - Identificação encomenda - Datas - Listagem de Artigos - Notificações - Montantes - Condições - Agreement	Cliente	"Layout" geral da página				8,0
2	Identificação Cliente						
2.1	A identificação do cliente passa por vários campos:	Cliente			Open	Must Have	2,0

	- Nome do Cliente - Morada - NIF - Email - Contacto						
3	Identificação Encomenda	Cliente			Open	Must Have	
3.1	A encomenda deve ter um ID Encomenda, número identificativo e um campo de Entrega(S/N)	Cliente			Open	Nice To Have	3,0
4	Datas	Cliente			Open	Nice To Have	
4.1	Área a apresentar todas as datas, algumas abertas outras automáticas: Automáticas-Fechadas: ->Data de Encomenda ao Fornecedor ->Data de Receção de Mercadoria ->Data de Entrega de Mercadoria Editável - possibilidade manual ->Data para Encomendar ->Data de Notificação ao Cliente	Cliente			Open	Must Have	5,0
5	Listagem de Artigos	Cliente			Open	Must Have	
5.1	Podem ser feitas várias introduções de artigos diferentes	Cliente			Open	Nice To Have	3,0
5.2	A introdução de um artigo é feita com os seguintes campos: - ITM 8 - Código EAN 13 - Designação / Fornecedor - Quantidade - Encomenda / Stock - Preço Unitário c/IVA	Cliente			Open	Must Have	8,0

5.3	O campo Total c/IVA é calculado automaticamente	Cliente	multiplicação entre quantidade encomendada e o preço por unidade		Open	Nice To Have	2,0
6	Notificações	Cliente			Open	Must Have	
6.1	Aquando a recepção da mercadoria, deve ser possível Notificar o Cliente	Cliente	Consequentemente, no campo " <i>Data de Recepção de Mercadoria</i> " é atualizado o valor para o correcto, assim como o valor do campo " <i>Data de Notificação ao Cliente</i> ".			Must Have	3,0
7	Montante	Cliente					
7.1	Na Área de Montante aparecem vários campos automaticamente preenchidos - Montante de Transporte - Montante Factual c/IVA - Montante do Sinal - Restante a Pagar	Cliente	O Montante de Transporte só aparece se na "Identificação da Encomenda" o campo "Entrega(S/N)" tenha sido respondido de forma afirmativa. O montante Factura c/IVA será o resultante calculado no Total c/IVA O Montante do Sinal existe somente se o cliente pagou algum artigo já em stock, ou se deu algum dinheiro de entrada (na encomenda), neste caso este valor é dedutível ao Total c/IVA e será apresentado no campo Restante a Pagar			Must Have	6,0
8	Condições	Cliente					
8.1	Área onde mostra as Condições de entrega	Cliente	Se o cliente for de acordo com condições impostas, terá de fazer uma assinatura como aceita as condições .			Must Have	2,0

9.1	Campo de confirmação	Cliente	Se a mercadoria tiver toda em ordem, conforme o acordado e encomendado o cliente assina assim como o vendedor			Important to Have	2,0
						Total Consulting	36
						Req Analysis	8
						Project Management	7
						Training	16
						Data Migration	4
						QA / Tests	7
						ALM	0
						TOTAL PROJECT (h)	78
						Hourly Rate	5
						Overheads (€)	300
						Discounts (%)	
						TOTAL PROJECT (€)	692

4.1.2. Diagramas de contexto

O diagrama de contexto (Figura 15) é por norma dos primeiros a ser estabelecido pois é um dos mais abrangentes, podendo incluir todo o sistema. Mostra as interfaces entre os sistemas e o contacto com as entidades externas. Não tendo em atenção a estrutura interna do sistema pois a sua finalidade é identificar limitações e restrições.

Podemos identificar uma fronteira (a azul) que identifica as áreas do serviço a ser desenvolvido dentro do *Salesforce*, assim como as comunicações para com as entidades internas e externas.

Denoto uma automatização na comunicação para com a entidade externa por parte do serviço *Salesforce*, sendo um dos pontos fortes de utilizar uma SaaS (designada pela linha a tracejado).

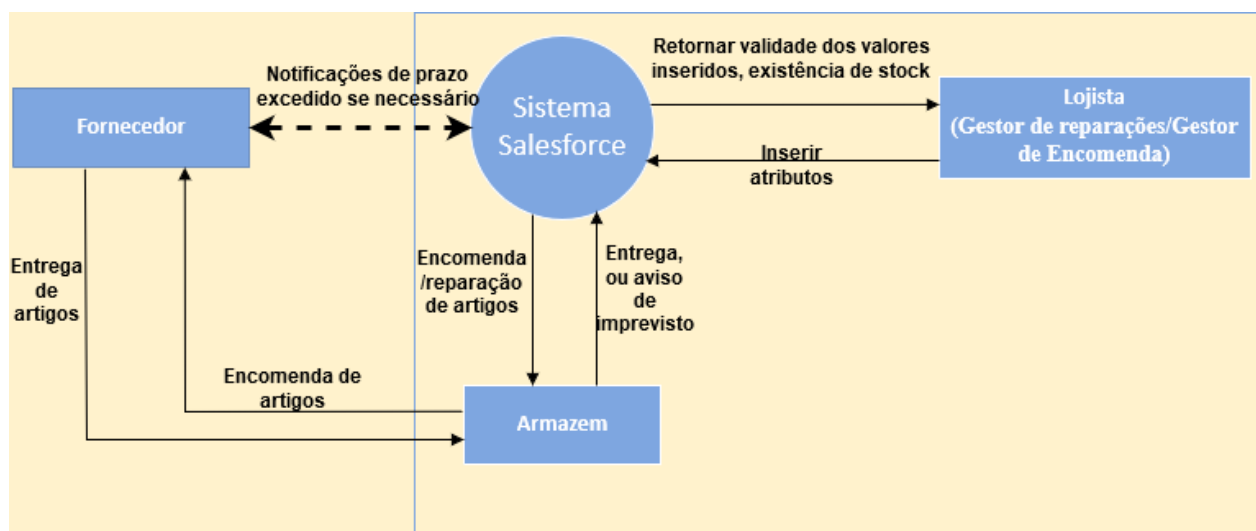


FIGURA 15- DIAGRAMA DE CONTEXTO BRICOMARCHÉ

4.1.3. Diagrama Entidade Relacionamento e Diagrama de Classes

No diagrama de entidade e relacionamento [17], são estendidas as entidades que atuam, quer de forma interna no sistema como também a inclusão da parte externa, de modo a estabelecer as relações entre essas entidades. É apresentado na Figura 16 um possível modelo para o Diagrama Entidade Relacionamento para o caso do estudo presente.

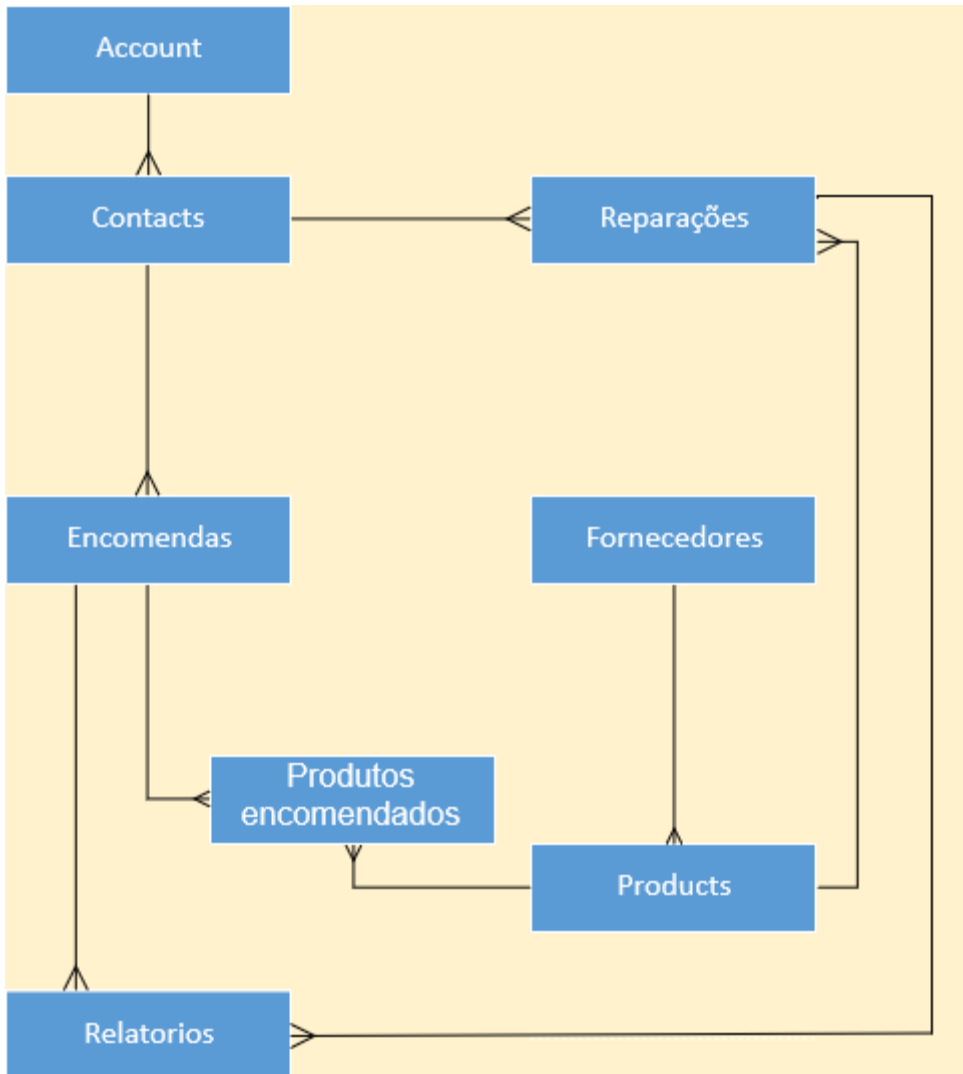


FIGURA 16- DIAGRAMA ENTIDADE RELACIONAMENTO BRICOMARCHÉ

Descrevendo as entidades na forma de classes é possível ver de forma mais explícita os constituintes da entidade, assim como o que as relaciona. Portanto o diagrama de classes é um diagrama de alta usabilidade durante o período de desenvolvimento devido as informações que permite ver de forma simples e rápida (Figura 17).

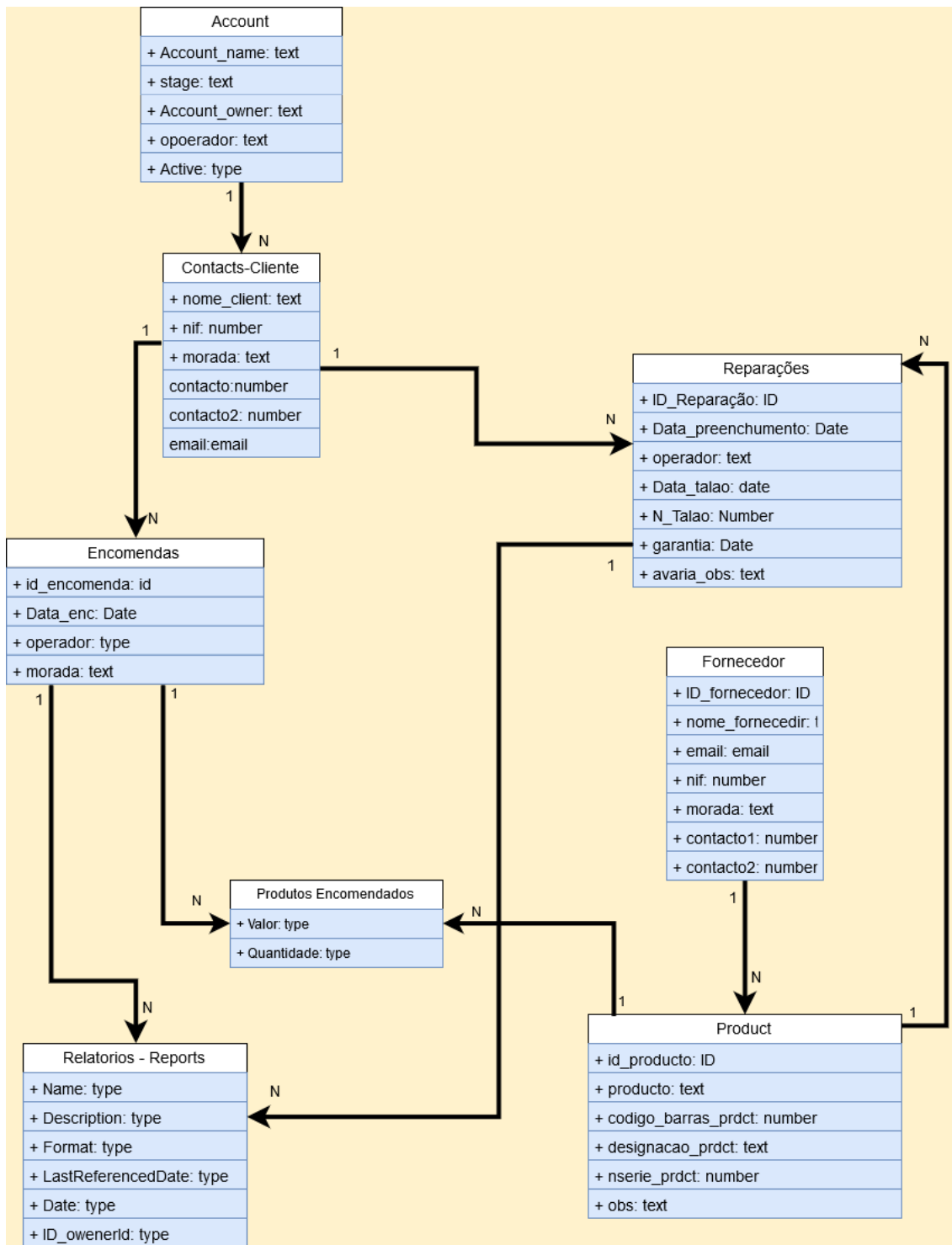


FIGURA 17- DIAGRAMA DE CLASSES BRICOMARCHÉ

4.1.4. Dicionário de dados

O dicionário de dados é uma lista de Metadados, ou seja, dados sobre outros dados, que pretende desenvolver e melhor explicar cada um dos dados, de forma a facilitar a forma como se identificam e organizam.

É uma forma de reduzir a utilização da definição semântica aquando da nomeação dos dados, de modo a que a utilização semântica passe só a ser como uma explicação clara do significado do dado. Por outro lado, é usada uma definição na forma de representação (tipo de dado) para identificar o tipo de armazenamento indicado.

Sendo que no projeto do Bricomarché só fiz a Análise de Requisitos, o processo de desenvolvimento foi atribuído aos profissionais da empresa, e ainda que se tenham baseado nesta análise, situações como os nomes atribuídos aos dados assim como o seu tamanho foram meramente uma forma de os descrever e melhor definir, não sendo iguais aos nomes finais do projeto posteriormente implementado.

Foi feito o aproveitamento de objetos genéricos do *Salesforce*, o objeto *Account* e *Contact* para introduzir no *Account* o Gestor de Reparação e no *Contact* o cliente, estas tabelas já apresentam relações entre elas daí a sua escolha. O objeto de Relatórios, *Report* também é um elemento genérico que foi alterado para possibilitar a receção de elementos geradores do pedido de relatório. São então listados os dicionários de dados para melhor descrever os objetos e causa (Tabela 4 a Tabela 10).

TABELA 4- DICIONÁRIO DE DADOS - TABELA ACCOUNT

Nome do campo	Valores por defeito	Tipo de dados	Descrição	Obrigatorio
Account_name	nulo	Lookup(Account)	Nome da conta que vai tratar da reparação ou encomenda, nome do operador	x
active	false	Picklist	flag para demonstrar se o serviço que esta a desempenhar esta em aberto ou nao	x
Account_owner	nulo	Text(120)	Dono da conta - gestor	
operator_id	AUTO_INCREMENT	id	numero identificativo unico do operador gestor de encomenda ou gestor de reparação	x
localidade	null	text(120)	localidade do ponto de venda onde é efectuado os serviços de encomenda e reparação	

TABELA 5- DICIONÁRIO DE DADOS - TABELA CONTACT

Nome do campo	Valores por defeito	Tipo de dados	Descrição	Obrigatorio
id_client	AUTO_INCREMENT	ID	Numero identificativo do cliente	x
nome_cliente	null	Text(120)	Nome do cliente	x

nif	0	Number(9,0)	numero de identificação fiscal do cliente	x
morada	null	Text(250)	endereço da residencia/entrega do cliente	
contact1	0	Phone(9,0)	Contacto do cliente	x
contact2	0	phone(9,0)	contacto secundario	
email	null	email	email do cliente	

TABELA 6- DICIONARIO DE DADOS - TABELA ENCOMENDA

Nome do campo	Valores por defeito	Tipo de dados	Descrição	Obrigatorio
Id_encomendas	AUTO_INCREMENT	Id	Numero identificativo da encomenda	x
Data_enc	TIMESTAMP	Date	Data em que foi feita a encomenda	
operador_name	null	Text (120)	operador que fez a encomenda, nome	x
product1	null	Number(10,0)	produto que vai ser encomendado	x
product_amout	0	Number (10,0)	quantidade do produto que vai ser encomendado	
nome_client	null	Text(120)	nome do client	x
nif	0	Number (9,0)	numero de identificação fiscal do cliente	x
morada	null	Text(255)	residencia ou local de entrega da encomenda	

TABELA 7- DICIONÁRIO DE DADOS - TABELA RELATORIOS

Nome do campo	Valores por defeito	Tipo de dados	Descrição	Obrigatorio
Name	null	Lookup (reparação/encomenda)	Titulo do relatorio referente a reparação ou encomenda	x
Description	null	Text (250)	Descrição e/ou observações	
Format	1	PickupList	formato em que vai ser emitido o relatorio	
LastreferencedDate	null	Lookup(date)	data da ultima modificação em relação a reparação / encomenda	
Date	TIMESTAMP	Date	Data do relatorio	x
Id_ownerid	AUTO_INCREMENT	ID	Numero identificativo de quem emitiu o relatorio	x

TABELA 8- DICIONÁRIO DE DADOS - TABELA PRODUCT

Nome do campo	Valores por defeito	Tipo de dados	Descrição	Obrigatorio
Id_producto	AUTO_INCTREMENT	ID	Numero identificativo do produto	x
product_name	null	Text(120)	nome do produto	x
codigo_barras_prdct	0	Numbre(23,0)	valor do codigo de barras que identifica o produto	

designacao_prdct	null	Text(250)	designação e descrição do produto	
nserie_prdct	0	Number(30)	numero de serie que identifica o produto	x
avaria_prdct	null	Text(250)	tipo de avaria , descrição	
obs	null	Text (250)	observações adicionais	

TABELA 9- DICIONÁRIO DE DADOS - TABELA FORNECEDOR

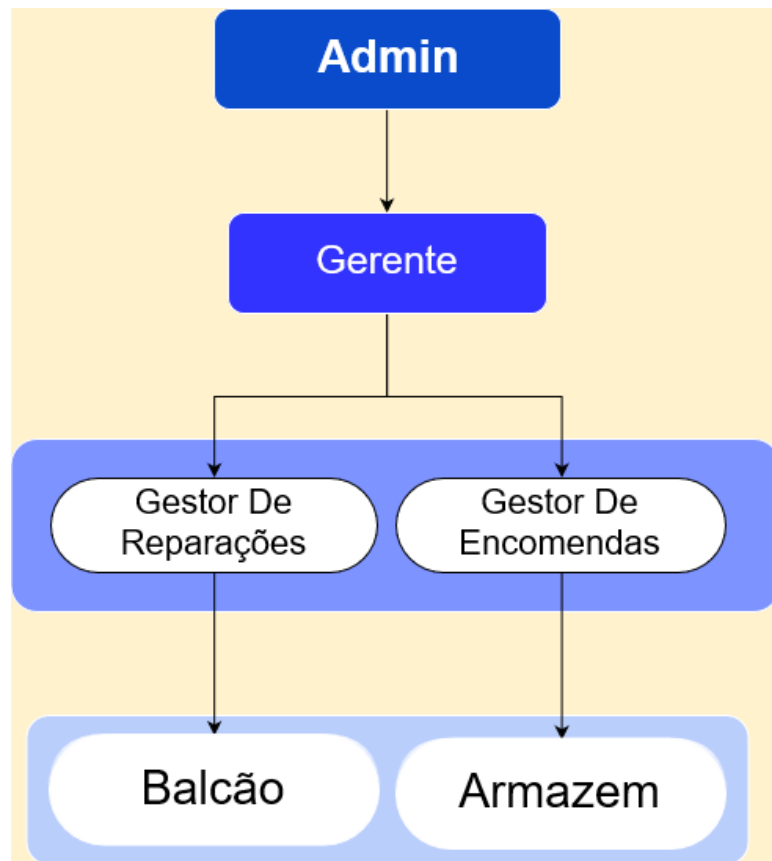
Nome do campo	Valores por defeito	Tipo de dados	Descrição	Obrigatorio
ID_fornecedor	AUTO_INCREMENT	ID	Número identificativo do fornecedor	x
Nome_fornecedor	null	Text(120)	nome do fornecedor	x
email	null	email	email do fornecedor	
nif	0	Number(9,0)	numero de identificação fiscl do fornecedor	x
morada	null	Text(250)	morada onde estao as instalacoes do fornecedor	
contacto1	0	Phone	contacto do fornecedor	x
contacto2	0	Phone	contacto secundario	

TABELA 10- DICIONÁRIO DE DADOS - TABELA REPARAÇÕES

Nome do campo	Valores por defeito	Tipo de dados	Descrição	Obrigatorio
ID_Reparcoes	AUTO_INCREMENT	ID	Numero identificativo do serviço de reparação	x
data_preenchimento	TIMESTAMP	Date	data de preenchimento do formulario de reparação	x
localidade	lookup(localidade)	Text(120)	local onde a reparação esta a ser pedida	
ID_operador	0	PickList	numero identificaditvo do operardor	x
operador_name	null	PickList	nome do operador que faz o pedido de reparação	
nome_cliente	null	Text(120)	nome do client	x
data_talao	null	Date	data do talao do artigo (garantia)	
n_talao	0	Number	numero do talao	x

4.1.5. Diagrama de Hierarquias

Como forma de garantir um claro sistema de organização, quer seja na atribuição de tarefas e casos de uso, na responsabilidade atribuída, mas também na fácil visualização das ordens hierárquicas temos o diagrama de hierarquias (Figura 18).



**FIGURA 18- DIAGRAMA DE HIERARQUIAS
BRICOMARCHÉ**

4.1.6. Diagramas de Fluxo

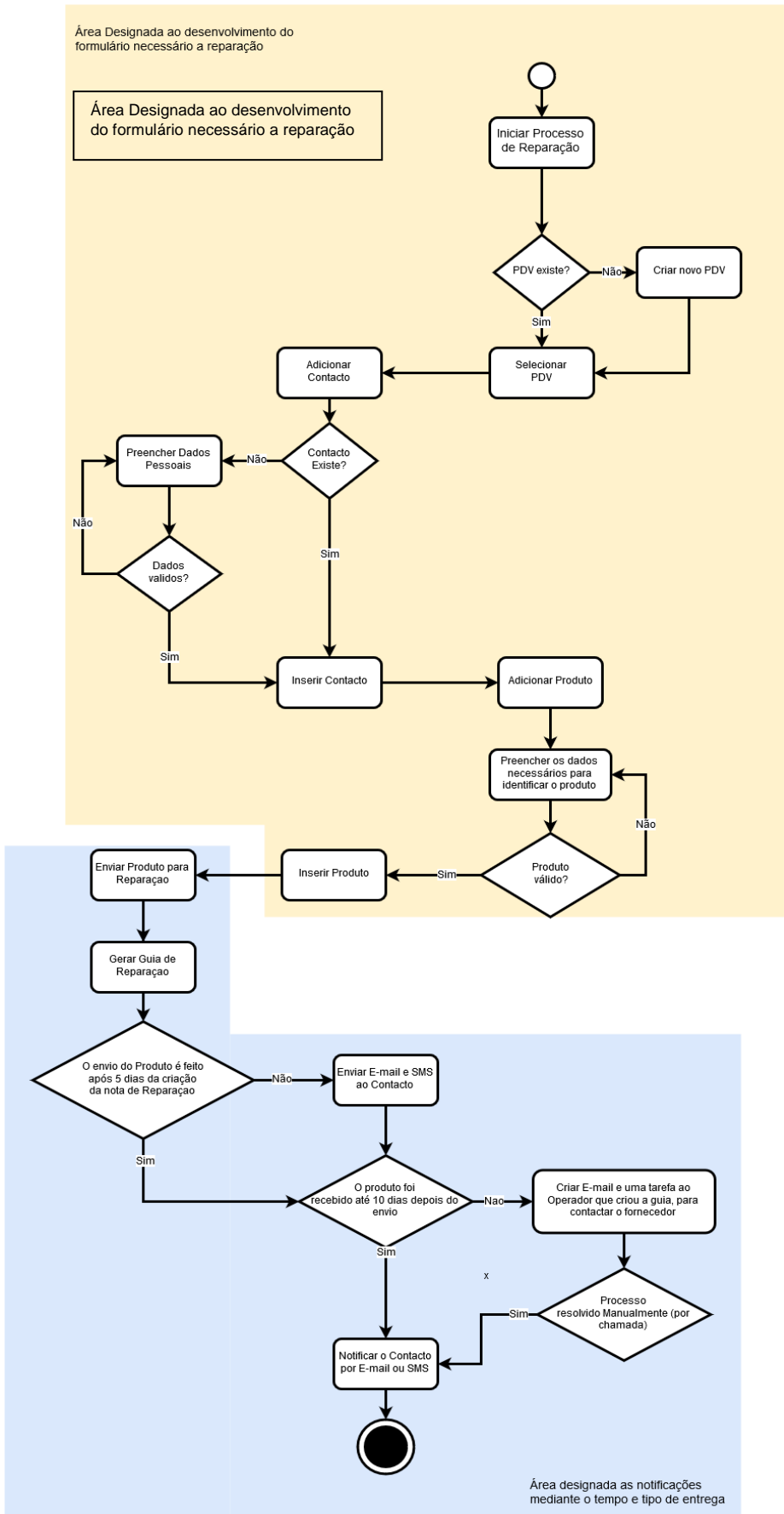
4.1.6.1. Diagrama de fluxos reparações

É apresentado uma hipótese de diagrama de fluxo (Figura 19) onde são apresentadas autenticações e verificações ao longo do fluxo do serviço, este diagrama tem a particularidade de estar dividida em duas áreas, pois apesar de o fluxo ser contínuo, são tidas em conta duas áreas distintas no seu progresso. A primeira área trata da criação e preenchimento de um formulário relativamente à reparação e dados pessoais do cliente, incluindo automatismos como preenchimento automático de dados pessoais e verificação de valores.

A segunda área é relativa à parte que trata e gere as informações comunicadas pelo fornecedor, assim como as notificações a serem efetuadas para com o cliente e automatismos de relativamente ao tempo e condições.

Área Designada ao desenvolvimento do formulário necessário a reparação

Área Designada ao desenvolvimento do formulário necessário a reparação



Área designada as notificações mediante o tempo e tipo de entrega

FIGURA 19- DIAGRAMA DE FLUXO ETAPA REPARAÇÕES

4.1.6.2- Diagrama de fluxo Encomenda

É apresentada uma hipótese de um diagrama de fluxo (Figura 20) para o serviço de encomendas pretendido. Esta conta com automatismos, como preenchimento automático do contacto mediante o Número de identificação fiscal, assim como validar este número e outros, como códigos de um artigo.

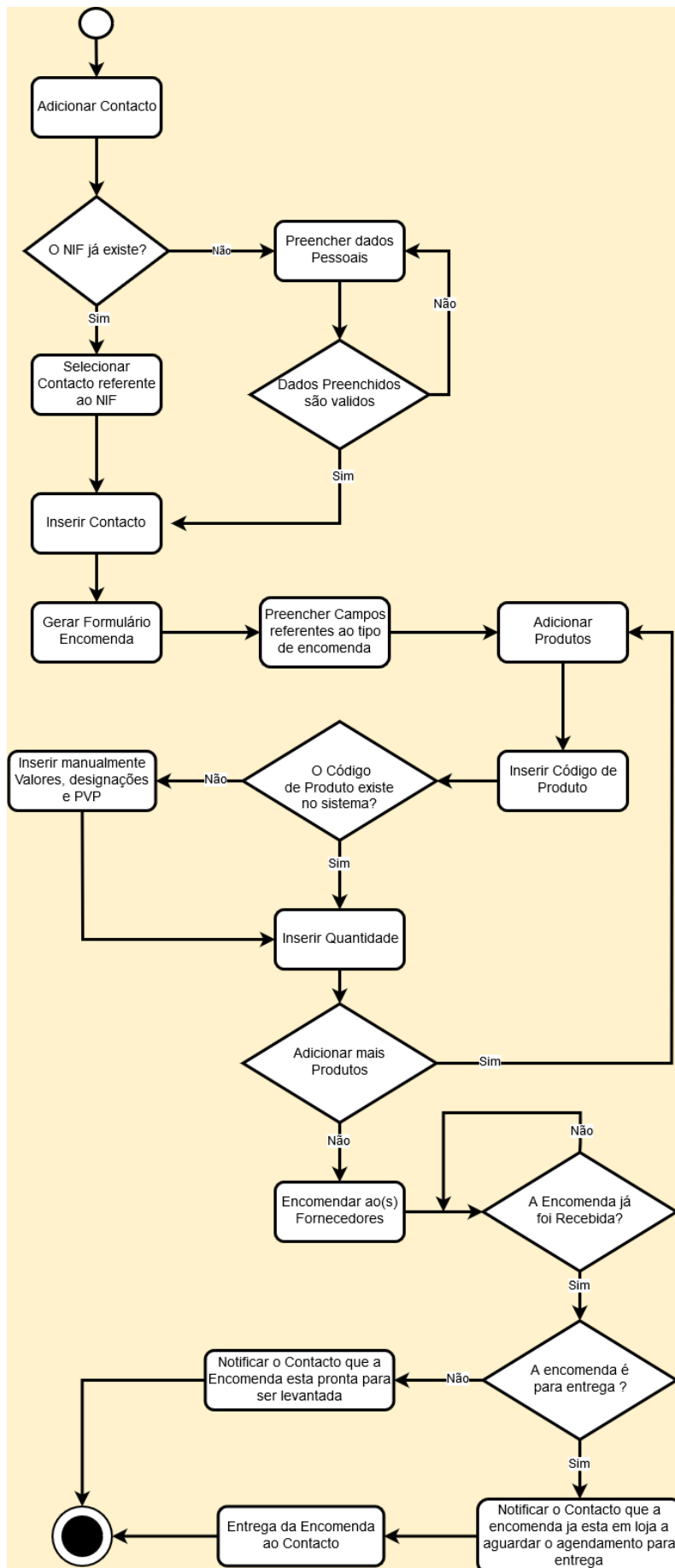


FIGURA 20- DIAGRAMA DE FLUXO ETAPA ENCOMENDA

4.2. Caso de estudo Projeto Lista de requisitos em *Salesforce*

Logo no capítulo 3, Estado da Arte, este relatório se diferenciou, pois não houve investigação de aplicações idênticas ou concorrentes. A razão é que o objetivo não era competir com as aplicações já existentes, mas sim fazer uso das diversas vantagens do *Salesforce*. Quase todas as tecnologias usadas pela empresa fazem parte dos serviços prestados pela *Salesforce*, mas no que toca aos levantamentos de dados, organização e desenvolvimento de requisitos, era efetuada num serviço de terceiros. Na necessidade de colmatar essa lacuna, surgiu a ideia de fazer uma aplicação “tipo folha de cálculo” idêntica ao Microsoft Excel ou Google Sheets, mas utilizando todas as vantagens disponíveis como pelos *Salesforce*, como:

- Centralização: O trabalho desenvolvido na empresa era quase exclusivamente feito dentro do *Salesforce*, juntar a isso a possibilidade de ter uma lista de requisitos na mesma plataforma em que estes estão a ser desenvolvidos é benéfico, seja a nível de facilidade de acesso, como a nível de produtividade devido à possibilidade de automatizar funções ou avisos.
- Caso os serviços previamente usados fossem P2U (*Pay to Use*) ou B2U (*Buy to Use*), é sempre benéfico conseguir substituí-lo por um outro *software* que não apresente nenhum encargo financeiro acrescido à empresa. Como no caso de a Dom Digital ser conhecida pelos serviços *Salesforce*, desenvolver nesta plataforma não seria nenhum encargo financeiro adicional.
- Mesmo sendo material de baixo risco de segurança (informações abstratas como resumos e apontamentos de reuniões com *stakeholder*), existe sempre a questão da segurança que é mais que assegurada na plataforma da *Salesforce*.

4.2.1. Lista de requisitos do caso de estudo

Foi considerado este como o segundo *sprint*, tinha como objetivo a Análise do Projeto Lista de Requisitos em *Salesforce*. Projeto este a ser implementado em uma fase seguinte.

Foi feita a análise de requisitos para o projeto onde foram estendidos e explicitados os requisitos numa lista (Tabela 11), requisitos estes deduzidos pelas diversas reuniões com o orientador de estágio e o CTO da empresa.

TABELA 11- LISTA DE REQUISITOS DO CASO DE ESTUDO LISTA DE REQUISITOS EM SALESFORCE



Implementação de lista de requisitos em *Salesforce* - Requirements Summary - Work Version 1.1

Autoria: Micael Capelão

O projeto tem como finalidade desenvolver uma ferramenta no auxílio do levantamento de dados, esta ferramenta visa agilizar o processo de interação entre os *Developers* e os Stakeholders assim como o entendimento mútuo. Para tal pretende-se desenvolver uma ferramenta usando *Salesforce*, o que nos dá a possibilidade de centralizar todo o processo, assim como ter toda a robustez disponibilizada pela plataforma.

Req ID	Requirement	Brought Up By	Comments and Questions	Solution Type	Requirement Status	Priority	Hours
1	Ligação com oportunidades				Open	Must Have	
1.1	Quero que uma oportunidade existente tenha um botão para que possa criar um "Resumo da Lista de Requisitos"	Dom Digital	Para cada oportunidade deve ser possível criar um "Resumo da Lista de Requisitos", uma folha funcional para o levantamento de requisitos. Este Resumo deve ter como título o nome da oportunidade em adição a versão atual.		Approved	Must Have	1
1.2	Na criação do "Resumo da Lista de Requisitos" ter uma Checkbox para cada um dos campos da lista de requisitos: - Campo de requisitos - Campo Brought Up By - Campo Comments and Questions	Dom Digital	Configurar que campos vão ser necessários no levantamento de requisitos ao selecionar os mesmos nas checkboxes, os valores default serão todos selecionados pois no fim do levantamento de requisitos, na configuração final será possível ocultar campos que se achem desnecessários.		Open	Must Have	1

	- Solution Type - Requirement Status - Priority level - Hours						
1.3	Na criação do "Resumo da Lista de Requisitos" ter uma Picklist (Multi-Select) para cada uma das opções de visuais/técnicas: - Logos - Visualização da orçamentação ou so totais de valores	Dom Digital	O requisitos Logos refere-se a adição de Logotipos da Empresa em si, ter disponíveis logotipos da DomDigital e LOBA. Permitir a escolha também de orçamentação e totais de valores.		Open	Must Have	1
	Tabela / objecto - Resumo da Lista de requisitos				Open	Must Have	
2	1 - Title				Open	Must Have	2
2.1	Deve ter um campo para inserir o título	Dom Digital	O título do documento deve ser correspondente a oportunidade na qual foi iniciada o levantamento de requisitos		Open	Must Have	
2.2	Por cada versão é criado um novo record	Dom Digital	mantém a integridade das versões prévias		Open	Must Have	
3	Project Overview				Open	Must Have	2
3.1	No hover/highlight deverá ser apresentado o valor deste campo	Dom Digital	Breve descrição sobre o tema do trabalho, descrição vista de um ponto de alto nível , abrangente mas sem entrar em detalhes		Open	Must Have	
	Tabela / objecto - Lista de requisitos				Open	Must Have	
4	Req_ID	Dom Digital			Open	Must Have	2
4.1	Coluna de incrementação automática por linha	Dom Digital	Deve ser em aberta de forma a ser possível criar subpontos (2->2.1 ; 2.2)		Open	Must Have	

4.2	Deve ser o número identificativo de toda a row	Dom Digital			Open	Must Have	
5	Requirement	Dom Digital			Open	Must Have	1
5.1	Um campo para inserir os dados	Dom Digital	O requisito deve ser o mais esclarecido possível assim com conotação técnica		Open	Must Have	
6	Campo Brought Up By	Dom Digital			Open	Must Have	1
6.1	Ter uma checklist com as diversas entidades participantes	Dom Digital			Open	Must Have	
7	Campo Comments and Questions	Dom Digital			Open	Must Have	1
7.1	Campo onde pode ser comentado e explicado o requisito pertencente.	Dom Digital	Se necessário breve explicação sobre algumas nuances que possam acontecer em relação ao requisitos, pode ser mais explicado e específico		Open	Must Have	
8	Solution Type	Dom Digital			Open	Must Have	1
8.1	picklist com o tipo de soluções disponíveis	Dom Digital	Melhor a forma de organização em longas listas de requisitos, por permitir designar a metodologia a usar na solução		Open	Must Have	
9	Requirement Status	Dom Digital			Open	Must Have	1
9.1	Checkbox com os estados de desenvolvimento do requisitos	Dom Digital	Permite esclarecer o estado em que se encontra o requisitos, se está a ser revisto, aprovado ou em aberto ainda		Open	Must Have	
10	Priority level	Dom Digital			Open	Must Have	1
10.1	Check box com diferentes níveis de prioridade	Dom Digital	Definir o nível de prioridade do requisito em causa		Open	Must Have	

11	Hours	Dom Digital			Open	Must Have	
1.1	Campo para inserir horas	Dom Digital	Tempo estimado para desenvolver o requisitos		Open	Must Have	
12	Montantes				Open	Must Have	6
12.1	<p>Deve existir vários campos numéricos no fim da lista de requisitos:</p> <p>Valores de horas previstas</p> <p>->Total Consulting</p> <p>-> Req Analysis</p> <p>-> Training</p> <p>-> Data Migration</p> <p>-> QA/ Test</p> <p>-> ALM</p> <p>-> Total Price</p> <p>Orçamentação</p> <p>-> Hourly Rate</p> <p>->Overheads(€)</p> <p>->Discounts(%)</p> <p>->Total Project(€)</p>	Dom Digital	<p>É possível selecionar (tanto no início como nas configurações finais) quais as opções a mostrar na área do montante.</p> <p>Valores de horas previstas</p> <p>->Total consulting - corresponderá ao tempo necessário para o levantamento de Requisitos</p> <p>->Req Analysis - o tempo necessário para analisar tudo o que foi entendido no levantamento de Requisitos</p> <p>->Project Management - A atribuição de tarefas</p> <p>->Training - Se os <i>Developers</i> precisam de formação em alguma área de forma a terem as habilitações necessárias a desenvolver o projeto</p> <p>->Data Migration - Estimativa do tempo necessário para migrar dados, se for necessário</p> <p>->QA/Tests - Tempo necessário para verificação de falhas assim como a sua prevenção aquando a entrega ao cliente.</p> <p>->Total Project em horas - Cálculo de todos os valores necessários ao desenvolvimento do <i>software</i>.</p> <p>Orçamentação</p> <p>->Hourly Rate - Preço pago a hora</p> <p>->Overheads(€) - Algumas despesas extra seja em material ou licenças</p> <p>->Discounts(%) - Abatimento num Valor Futuro Total de projeto em percentagem, se</p>		Open	Must Have	

			houver. ->Total Project(€) - Estimativa custo monetário total do desenvolvimento do projeto , calculado pelo valor em horas necessário para o desenvolvimento do mesmo e considerando o Hourly Rate assim como Overheads e aplicando se existir o Discount.				
13	New Row				Open	Must Have	2
13.1	Deve existir um botão criar mais uma row vazia dos valores acima descritos.	Dom Digital	Apesar de a lista por default começar com X número de campos disponíveis ao preenchimento, caso seja necessário um maior número é oferecida a possibilidade de os adicionar.		Open	Must Have	
	Tabela / objecto - Configurações finais	Dom Digital			Open	Must Have	
14	Logos	Dom Digital			Open	Must Have	2
14.1	Pick list (Multy Select)com os logotipos disponíveis	Dom Digital	É oferecida a possibilidade de incluir os logotipos disponibilizados pelas empresas encarregues.Estes são apresentados no início da folha em cima do Titulo		Open	Must Have	
15	Guardar	Dom Digital			Open	Must Have	1
15.1	É dada a possibilidade do usuário guardar o seu trabalho.	Dom Digital	O usuário pode não ter acabado ainda a lista de requisitos ou simplesmente pode querer guardar o desenvolvimento do trabalho até ao dado ponto.		Open	Must Have	
16	Ocultar Colunas	Dom Digital			Open	Must Have	3

16.1	Pick list (Multy Select) com as opções previamente escolhidas selecionadas	Dom Digital	Deve ser oferecida mais uma vez a possibilidade de ocultar colunas, tenham sido elas usadas ou não. O default value será o escolhido na criação da página, por consequência esta opção só irá atuar em alguma coisa se alguma opção for desselecionada.		Open	Must Have	
17	Exportação	Dom Digital			Open	Must Have	2
17.1	Gerar uma cópia em PDF	Dom Digital	É oferecida a possibilidade de exportar o trabalho no formato PDF		Open	Must Have	
17.2	Campo de pesquisa do harddisk	Dom Digital	Escolher o endereço onde a cópia PDF deverá ser guarda		Open	Must Have	
18	Observações	Dom Digital			Open	Must Have	2
18.1		Dom Digital			Open	Must Have	
						Total Consulting	33
						Req Analysis	4
						Project Management	7
						Training	0
						Data Migration	0
						QA / Tests	7
						ALM	2
						TOTAL PROJECT (h)	52

						Hourly Rate	4
						Overheads (€)	
						Discounts (%)	
						TOTAL PROJECT (€)	207

4.2.2. Diagrama entidade e relacionamentos e diagrama de classes

Como foi explicitado antes, o diagrama de entidade relacionamento permite estabelecer os relacionamentos com as diferentes entidades. Foram trabalhadas três tabelas genéricas já existentes do *Salesforce* (*Opportunity*, *Account*, *Contact*), nas quais foram adicionados os campos necessários para existir um relacionamento com a tabela *Requisitos* (Figura 21). A possibilidade de utilizar tabelas genericamente disponibilizadas pela *Salesforce*, permite um nível de organização e produtividade maior, pois grande parte das ligações e dependências entre elas já estão criadas.

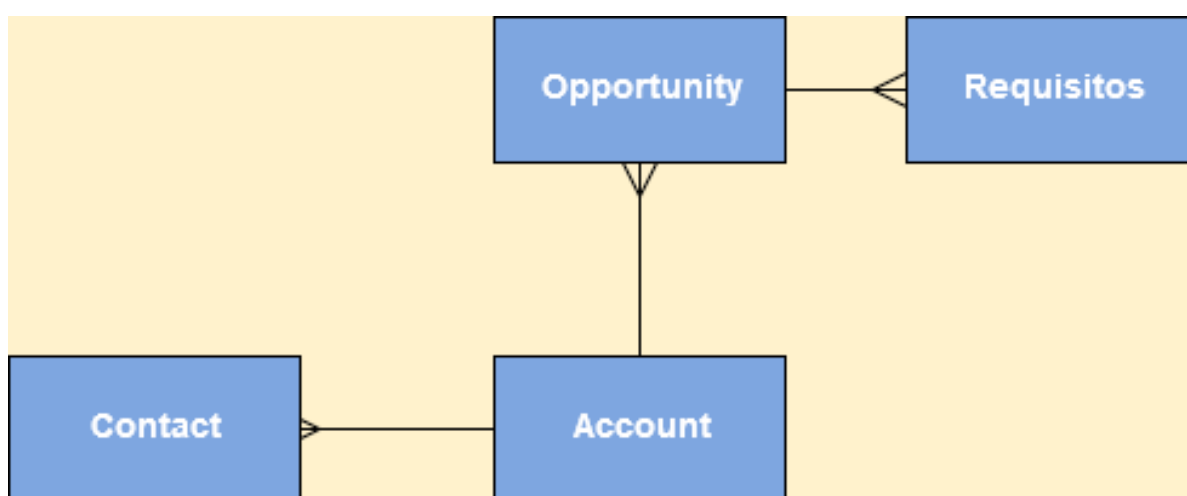


FIGURA 21- DIAGRAMA ENTIDADE RELACIONAMENTO LISTA DE REQUISITOS SALESFORCE

Utilizando a ferramenta *Schemabuilder* do *Salesforce* bastou selecionar os Objetos (tabelas, são tratadas como objetos em *Salesforce*) constituintes da aplicação implementada, é nos apresentado o diagrama de classes (Figura 22) e os seus relacionamentos. Para além de obter um diagrama ER (na forma de classes) também é possível juntar tabelas com outras que não estão de momento relacionadas. Possibilita assim um método de utilização extremamente “*user friendly*” de “*drag-and-drop*” para fazer tarefas que exigiram algum planeamento mais pesado e complicado num outro ambiente de análise de requisitos.

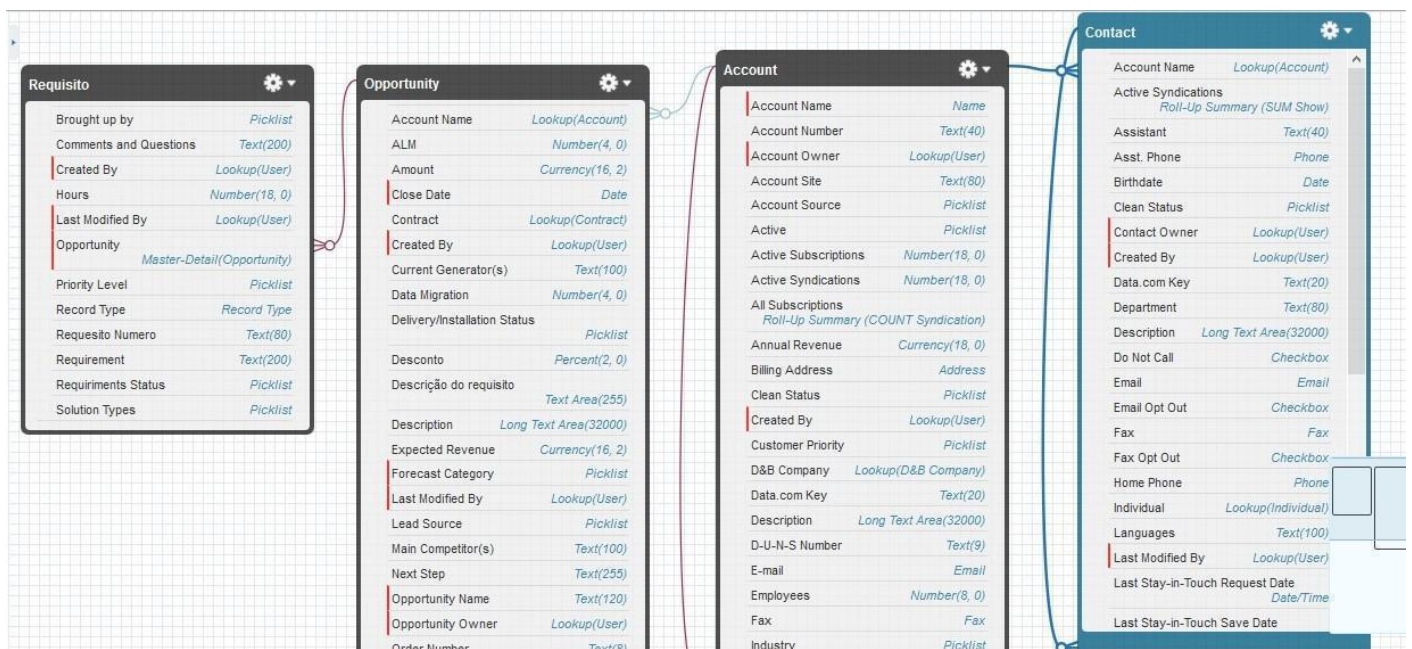


FIGURA 22- DIAGRAMA DE CLASSES LISTA DE REQUISITOS SALESFORCE

4.2.3. Dicionário de dados

O conceito de dicionário de dados já foi previamente introduzido, neste caso existe a vertente que o projeto foi implementado, sendo assim valores finais utilizados realmente no projeto final.

No caso do *Salesforce*, todos os dados têm também um respetivo nome identificativo na API (normalmente não é mais do que o sufixo “_c” ao nome do dado, salvo a exceção de dados já criados pelos *Salesforce* nos objetos genéricos) que permite fazer operações de maior complexidade como queries ou implementações.

Denoto também que no dicionário de dados faltam alguns campos que podem ser observados no diagrama de classes da Figura 21 , a razão disso é serem campos que não são utilizados

na aplicação implementada da lista de requisitos, sendo elementos genéricos pertencentes aos objetos genéricos também do *Salesforce* (não permite que sejam eliminados). Por isso de forma a apresentar um dicionário de dados explícito e referente ao Projeto da lista de requisitos foi optado por a saída dos mesmos.

Como já foi referido, o *Salesforce* vem com objetos previamente criados, objetos estes que já tem uma relação de hierarquias e dependências entre eles. Foi, portanto, escolhido três objetos genéricos do *Salesforce* (*Opportunity*, *Account*, *Contact*) devido a essas mesmas características. Uma oportunidade é para ser considerada como um projeto, é chamado de oportunidade porque o *Salesforce* oferece a possibilidade de aplicar vários estados para a forma em que se encontra o projeto (Se está em negociação, se já se encontra a ser desenvolvido) essa barra de estados foi usada também na implementação (Figura 23). A *Account* é um objeto que retrata a entidade ao cargo de completar o projeto (*Opportunity*) e como tal pode ter vários indivíduos integrantes, são estes retratados pelo objecto *Contact*. A tabela Requisitos foi um objeto criada com ligações a tabela *Opportunity*, pois cada requisito é relativo a somente uma oportunidade. É listado de seguida o dicionário de dados para cada uma das tabelas em causa (Tabela 12 a Tabela 15).

TABELA 12- DICIONÁRIO DE DADOS - TABELA OPPORTUNITY

Nome do campo	API Name	Tipo de dados	Descrição	Obrigatório
Account Name	AccountId	Lookup(Account)	Nome da conta	
ALM	ALM__c	Number(4, 0)	Gestao do ciclo de vida da aplicação	
Close Date	CloseDate	Date	Data de fecho da oportunidade se definido uma	x
Created By	CreatedById	Lookup(User)	Data de criação da oportunidade	x
Data Migration	Data_Migration__c	Number(4, 0)	Custos da migração de dados	
Desconto	Desconto__c	Percent(2, 0)	Desconto percentual	
Description	Description	Long Text Area(32000)	Breve descrição sobre oportunidade	
Last Modified By	LastModifiedById	Lookup(User)	Data de última modificação	x
Next Step	NextStep	Text(255)	Resultado da oportunidade quando esta é dada como fechada	
Opportunity Name	Name	Text(120)	Nome da oportunidade	x
Opportunity Owner	OwnerId	Lookup(User)	atual dono da oportunidade	x
Overheads (€)	Overheads__c	Currency(8, 0)	Despesas Gerais	
Preço por Hora	Pre_o__c	Currency(4, 1)	Preço definido por hora	
Probability (%)	Probability	Percent(3, 0)	Probabilidade da oportunidade, varia com o estado desta	x

Project Management	Project_Management__c	Number(4, 0)	Gestão de projeto em horas	
QA / Tests	QA_Tests__c	Number(4, 0)	Testes de qualidade e testes, em hora	
Req Analysis	Req_Analysis__c	Number(4, 0)	Analise de requisitos, em hora	
Soma de horas por Import	Soma_de_horas_por_Import__c	Roll-Up Summary (SUM Requisito)	Soma de horas de requisitos com a importância de Important	
Soma de horas por must have	Soma_de_horas_por_must_have__c	Roll-Up Summary (SUM Requisito)	Soma de horas de requisitos com importância de Must Have	
Soma de horas por niceth	Soma_de_horas_por_niceth__c	Roll-Up Summary (SUM Requisito)	Soma de horas de requisitos com importância de important to have	
Stage	StageName	Picklist	Estado atual da oportunidade	x
Total Consulting	Soma_de_horas__c	Roll-Up Summary (SUM Requisito)	Soma da hora defenida em cada requisito	
Total Projeto(h)	Total_Projecto_h__c	Formula (Number)	Soma das horas da consultoria e das outras necessidades	
Total Projeto(€)	Total_Projecto__c	Formula (Currency)	Orçamentação para a oportunidade	
Training	Training__c	Number(4, 0)	Treino em novas áreas do staff developer, em horas	
Type	Type	Picklist	Tipo de cliente	

TABELA 13- DICIONÁRIO DE DADOS - TABELA REQUISITO

Nome do campo	API Name	Tipo de dados	Descrição	Obrigatorio
Brought up by	Brought_up_by__c	Picklist	Quem teve a ideia do requisitos	
Comments and Questions	Comments_and_Questions__c	Text(200)	Comentários e Questões por colocar	
Created By	CreatedById	Lookup(User)	Autor do requisitos	x
Hours	Hours__c	Number(18, 0)	Horas planeadas para concluir a tarefa	
Last Modified By	LastModifiedById	Lookup(User)	Data da ultima modificação	x
Opportunity	Opportunity__c	Master-Detail(Opportunity)	Oportunidade associada	x
Priority Level	Priority_Level__c	Picklist	Nivel de prioridade (Must,importante,nice)	
Record Type	RecordTypeId	Record Type	Tipo de registo, para diferenciar niveis de prioridade	
Requesito Numero	Name	Text(80)	Numero manual de requisito, como um ID mas manual, premite sub pontuação como 1.1 ou 2.2.1	x
Requirement	Requirement__c	Text(200)	Corpo e descrição do requisitos	
Requiriments Status	Requiriments_Status__c	Picklist	Estado do requisitos, uma flag	
Solution Types	Solution_Tyoes__c	Picklist	Tipo de solução (confi,outofthebox,custom,partner)	

TABELA 14- DICIONÁRIO DE DADOS - TABELA ACCOUNT

Nome do campo	API Name	Tipo de dados	Descrição	Obrigatorio
Account Name	Name	Name	Nome da Conta	x
Account Owner	OwnerId	Lookup(User)	Dono da conta	x
Active	Active__c	Picklist	Estado de atividade da conta	
Active Subscriptions	Active_Subscriptions__c	Number(18, 0)	Subscrição de oportunidade ativa se alguma	
All Subscriptions	All_Subscriptions__c	Roll-Up Summary (COUNT Syndication)	Feed de todas as oportunidade subscritas pela conta	
Created By	CreatedById	Lookup(User)	Criado por	x
Last Modified By	LastModifiedById	Lookup(User)	Ultimo user a alterar a conta	x

TABELA 15- DICIONÁRIO DE DADOS - TABELA CONTACT

Nome do campo	API Name	Tipo de dados	Descrição	Obrigatorio
Account Name	AccountId	Lookup(Account)	Nome da conta a que esta associada o contacto	
Birthdate	Birthdate	Date	Data de nascimento	
Contact Owner	OwnerId	Lookup(User)	Dono do contacto	x
Created By	CreatedById	Lookup(User)	Criado por	x
Description	Description	Long Text Area(32000)	Descrição do contacto	
Email	Email	Email	Email do contacto	
Last Modified By	LastModifiedById	Lookup(User)	Ultima modificação em	x
Name	Name	Name	Nome do contacto	x

4.2.4. Casos de uso

O projeto tem um caminho bastante direto o que inviabiliza desenvolver diagramas de fluxo que acrescentem valor, pois iria haver poucos caminhos alternativos ou condicionais. Foi por isso escolhido definir os casos de uso do projeto, primeiro como diagrama (Figura 23) e depois descrevendo cada um deles (Tabela 16 a 19).

4.2.4.1. Diagrama de Casos de uso

São tidos em conta dois atores, um como Gestor de conta e outro como Gestor de Projeto. Podemos definir o Gestor de Conta como a pessoa que interage com o *stakeholder*, é por isso a pessoa que cria o projeto (Oportunidade), assim como a atribui ou cria a uma conta e respetivos contactos. Posteriormente, este Gestor de Conta passa a posse deste projeto (Oportunidade) a um Gestor de Projeto. Este Gestor de Projeto tem a função e poder de criar, editar, atualizar e apagar os requisitos. O seguinte diagrama, Figura 23, conta com os pré-requisitos de um registo na plataforma, assim como a sessão aberta por meio de um *login* válido.

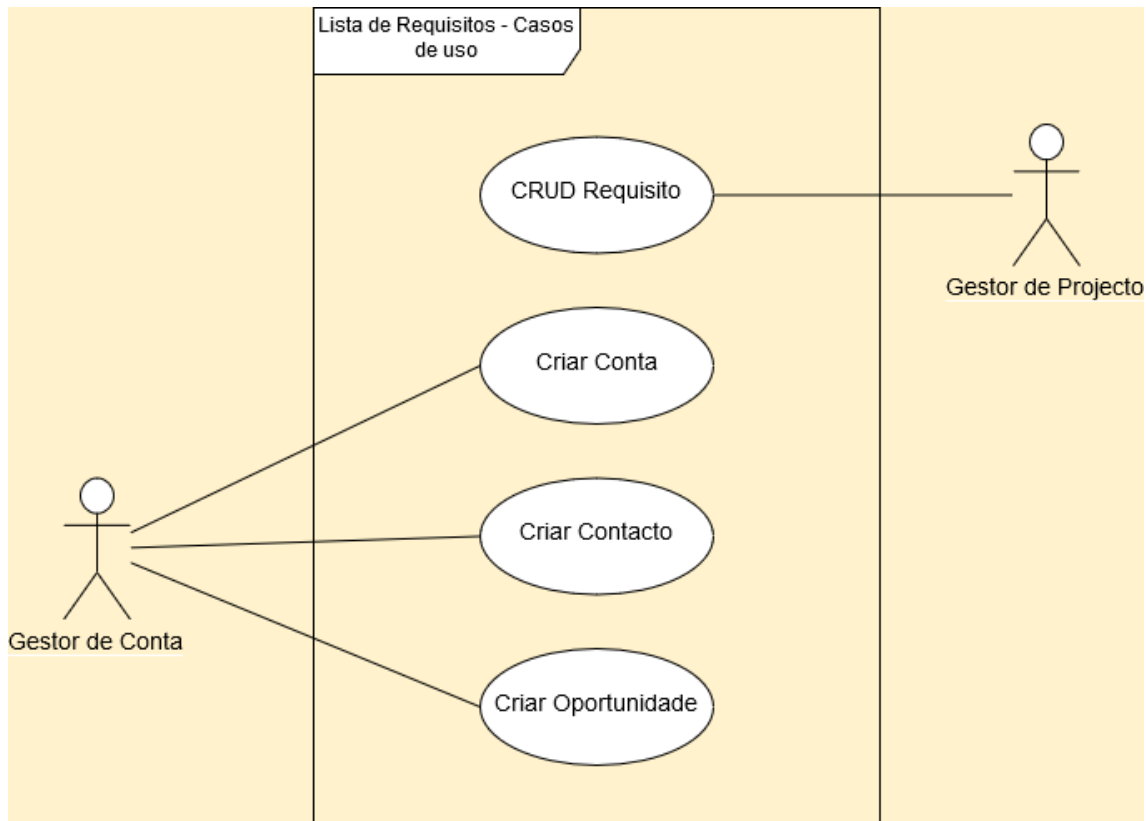


FIGURA 23- DIAGRAMA DE CASOS DE USO LISTA DE REQUISITOS SALESFORCE

4.2.4.2. Descrição de casos de uso

4.2.4.2.1. CASO DE USO – CRIAR CONTA

TABELA 16- DESCRIÇÃO CASO DE USO - CRIAR CONTA

Nome	Criar Conta
Descrição	Este caso de uso consiste em Criar uma conta para o utilizador
Ator envolvido	Gestor de Conta
Pré-Condição	Conta <i>Salesforce</i> Registada, Login válido
Fluxo Principal	<ol style="list-style-type: none">1. O Utilizador seleciona o AppLauncher (canto superior esquerdo da página)2. O Sistema apresenta todas as aplicações disponíveis na conta.3. O Utilizador seleciona a opção “Account”.4. O Sistema apresenta as contas existentes na Org e um menu de opções.5. O Utilizador seleciona a opção “New”6. O Sistema apresenta um formulário para a inserção dos dados da nova conta.7. O Utilizador preenche os campos do formulário8. O sistema insere a nova conta na base de dados.
Fluxo Alternativo	<p>3a. O utilizador pesquisa “Account” na barra de pesquisa e seleciona a opção “Account”</p> <p>8ª O sistema não aceita a criação de uma nova conta.</p>
Pós-Condição	O utilizador é redirecionado para o perfil da conta recém-criada
Suplementos ou Adornos	<ol style="list-style-type: none">1. Verificar se o utilizador preencheu todos os campos obrigatórios (*)

4.2.4.2.2. CASO DE USO – CRIAR CONTACTO

TABELA 17- DESCRIÇÃO CASO DE USO - CRIAR CONTACTO

Nome	Criar Contacto
Descrição	Este caso de uso consiste em Criar um Contacto para o utilizador
Ator envolvido	Gestor de Conta
Pré-Condição	Conta <i>Salesforce</i> Registada, Login válido
Fluxo Principal	<ol style="list-style-type: none"> 1. O Utilizador seleciona o AppLauncher (canto superior esquerdo da página) 2. O Sistema apresenta todas as aplicações disponíveis na conta. 3. O Utilizador seleciona a opção “Contacts”. 4. O Sistema apresenta os contactos existentes na Org e um menu de opções. 5. O Utilizador seleciona a opção “New” 6. O Sistema apresenta um formulário para a inserção dos dados do contacto. 7. O Utilizador preenche os campos do formulário 8. O sistema insere o novo contacto na base de dados.
Fluxo Alternativo	<p>3a. O utilizador pesquisa “Contacts” na barra de pesquisa e seleciona a opção “Contacts”</p> <p>8ª O sistema não aceita o registo</p>
Pós-Condição	O utilizador é redirecionado para o perfil do contacto recém-criado
Suplementos ou Adornos	<ol style="list-style-type: none"> 1. Verificar se o utilizador preencheu todos os campos obrigatórios (*)

4.2.4.2.3. CASO DE USO – CRIAR OPORTUNIDADE

TABELA 18- DESCRIÇÃO CASO DE USO - CRIAR OPORTUNIDADE

Nome	Criar Oportunidade
Descrição	Este caso de uso consiste em Oportunidade
Ator envolvido	Gestor de Conta
Pré-Condição	Conta <i>Salesforce</i> Registada, Login válido
Fluxo Principal	<ol style="list-style-type: none"> 1. O Utilizador seleciona o AppLauncher (canto superior esquerdo da página) 2. O Sistema apresenta todas as aplicações disponíveis na conta. 3. O Utilizador seleciona a opção “Listagem de Requisitos”. 4. O Sistema apresenta as oportunidades mais recentes no écran central e um menu de opções no lado direito. 5. O Utilizador seleciona a opção “New” 6. O Sistema apresenta um formulário para a inserção dos dados da nova oportunidade. 7. O Utilizador preenche os campos do formulário 8. O sistema insere a nova oportunidade na base de dados.
Fluxo Alternativo	<p>3a. O utilizador pesquisa “Listagem de Requisitos” na barra de pesquisa e seleciona a opção “Listagem de Requisitos”</p> <p>8ª O sistema não aceita a criação de uma nova conta.</p>
Pós-Condição	O utilizador é redirecionado para o perfil da oportunidade recém-criada
Suplementos ou Adornos	<ol style="list-style-type: none"> 1. Verificar se o utilizador preenche todos os campos obrigatórios (*)

4.2.4.2.4. CASO DE USO – CRIAR REQUISITO

TABELA 19- DESCRIÇÃO CASO DE USO - CRIAR REQUISITO

Nome	Criar Requisito
Descrição	Este caso de uso consiste em criar um requisito
Ator envolvido	Gestor de Projeto
Pré-Condição	Conta <i>Salesforce</i> Registada, Login válido
Fluxo Principal	<ol style="list-style-type: none"> 1. O Utilizador seleciona o AppLauncher (canto superior esquerdo da página) 2. O Sistema apresenta todas as aplicações disponíveis na conta. 3. O Utilizador seleciona a opção “Listagem de Requisitos”. 4. O Sistema apresenta as oportunidades mais recentes no écran central e um menu de opções no lado direito. 5. O Utilizador seleciona uma oportunidade. 6. O Sistema apresenta o resumo da oportunidade (o seu Stage, Detalhes, e uma vista abreviada de Requisitos caso estes já existam). 7. O Utilizador seleciona “View All” na vista abreviada de Requisitos. 8. O Sistema apresenta os requisitos referentes a oportunidade (caso exista algum) e apresenta um menu de opções a direita. 9. O Utilizador seleciona a opção “New”. 10. O Sistema apresenta um formulário para serem inseridos os dados constituintes do requisito O Utilizador preenche os dados. 11. O sistema insere o requisito na base de dados.
Fluxo Alternativo	<p>3a. O utilizador pesquisa “Listagem de Requisitos” na barra de pesquisa e seleciona a opção “Listagem de Requisitos”.</p> <p>5a. O Utilizador seleciona “new” para criar uma oportunidade nova.</p> <p>11ª O sistema não aceita a criação de um novo requisito</p>
Pós-Condição	O utilizador é redirecionado para a lista de requisitos
Suplementos ou Adornos	<ol style="list-style-type: none"> 1. Verificar se o utilizador preencheu todos os campos obrigatórios (*)

5. Tecnologias

Ao longo do estágio houve a necessidade de aprender a trabalhar e utilizar novos serviços e/ou tecnologias. Enquanto que algumas não acrescentariam valor em serem descritas no relatório, devido ao baixo uso ou pouco impacto no desenvolvimento, houve outras que foram essenciais ao longo de todo o trabalho, e como podem não ter sido definidas ou explicitadas em outro lugar, serve o presente capítulo para esse fim.

5.1. Lightning components

Os *Lightning components* [18] são uma plataforma de desenvolvimento de *software* com uma *user interface* em *Salesforce*. Existem duas partes constituintes de qualquer *lightning component*, a parte do cliente (*developers*) onde é usado *JavaScript* como linguagem, e o lado do servidor onde é usado Apex (Linguagem *StronglyTyped* baseada em *Java*, partilhando alguma da sua sintaxe, usada pelos *developers* para interagir e transacionar pedidos com o lado do servidor (*Force.com*), Figura 29.

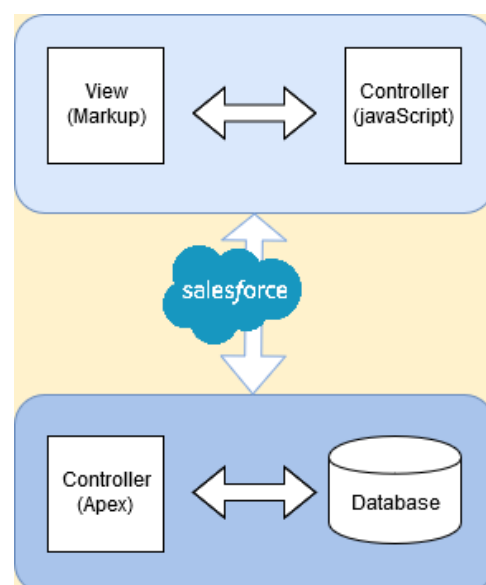


FIGURA 24- LIGHTNING COMPONENTS RELACIONAMENTOS

A comunicação entre componentes é feita na forma de troca de eventos a nível interno entre os vários constituintes, estes vários componentes que formam a aplicação é chamado de *bundle* (pacote), Figura 30.

- A *Component* contém toda a estrutura em HTML.
- *Controller* é o controlador principal do produto e é escrito em *JavaScript*.

Shortcut	Component Name	Action
Ctrl + Shift + 1	COMPONENT	
Ctrl + Shift + 2	CONTROLLER	Create
Ctrl + Shift + 3	HELPER	Create
Ctrl + Shift + 4	STYLE	Create
Ctrl + Shift + 5	DOCUMENTATION	Create
Ctrl + Shift + 6	RENDERER	Create
Ctrl + Shift + 7	DESIGN	Create
Ctrl + Shift + 8	SVG	Create

Bundle Version Settings

FIGURA 25- BUNDLE DE UM LIGHTNING COMPONENT

- *Helper* contém funções partilhadas por vários componentes, também escrito em *JavaScript*.
- *Style* contém todo o CSS relativamente ao escrito no *component*.
- *Documentation*, *Renderer*, *Design* e *SVG*, não foram usados durante o estágio, tem propósitos mais específicos ou estéticos, sendo o *Documentation* para documentar as funções e comportamentos do *Lightning component*, *Design* permite interagir com a ferramenta de design de aplicações da *Salesforce* (*Lightning AppBuilder*) e o *SVG* atribui um *icon* ao *Lightning component*.

Os *lightning components* foram úteis no desenvolvimento do projeto por serem formas rápidas de construir *software*. Utilizam línguas de marcação e programação conhecidas e oferecem maior liberdade no que toca a usar serviços internos do *Salesforce*, seja o caso por exemplo o uso de *queries* à base de dados que sem os *lightning components* é vedado. Permite na forma de *pop-up* desenvolver soluções que podem ser incluídas no ambiente de trabalho normal de uma Org do *Salesforce*.

5.2. Draw.io e UML

Durante o percurso académico, quando foi necessário fazer diagramas no âmbito de engenharia de *software*, foi sempre usado o UML – *Unified Modeling Language* como um standard, e devido a existir parcerias entre o Instituto Politécnico da Guarda com serviços da *Microsoft Office*, sempre foi usado o *software Microsoft Visio* para efetuar esses projetos. O objetivo de uma empresa será sempre ter lucro com as suas ações, é o propósito primário de qualquer negócio, então durante o meu estágio quando me foi colocada a necessidade de desenvolver esses mesmo diagramas e usei o *Microsoft Visio*, fui apresentado ao Draw.io uma tecnologia *open-source* com suporte para browsers (na realidade construída de propósito

para ser utilizada em browsers, tendo também uma integração completa com a plataforma Drive da Google).

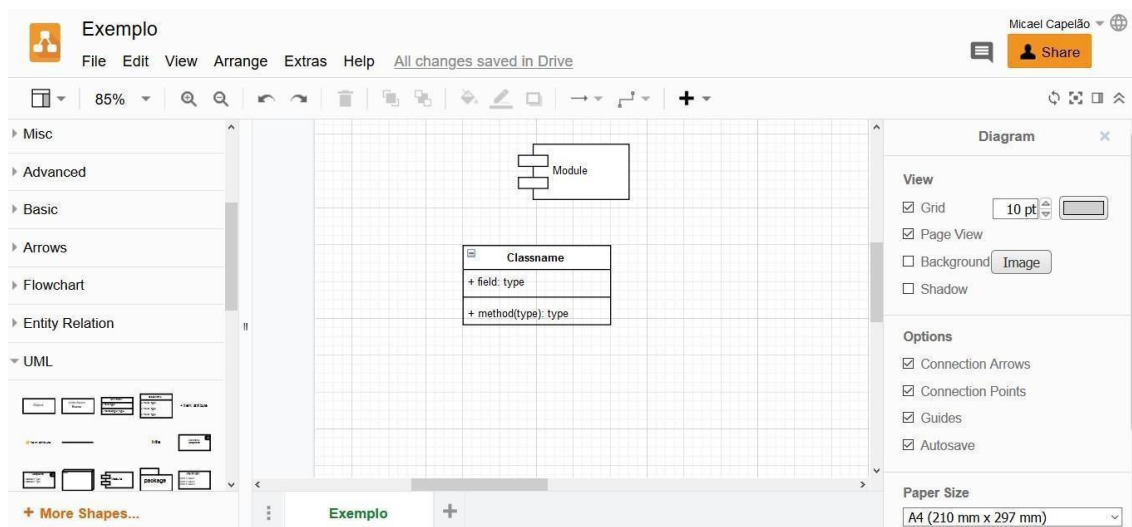


FIGURA 26- LAYOUT DO AMBIENTE DE TRABALHO DRAW.IO

É possível usar as formas estandardizadas do UML pois já se encontram incluídas, assim como é possível usar outras linguagens de modelação adicionando mais formas (*More Shapes*), ou então fazer uso de formas geométricas simples (Figura 31). Todos os diagramas foram desenvolvidos neste *software*.

5.3. Google Drive – Google Docs

Foi referido a integração do Draw.io com o Drive da Google, e é importante referir a importância que teve. É um serviço de armazenamento e sincronização em *cloud* da google que oferece um conjunto de soluções no modelo free-to-use. Como já foi referido, este tipo de solução é agradável do ponto de vista financeiro para uma empresa que não detenha já um conjunto de soluções parecido. Todos os desenvolvimentos durante e após o estágio foram armazenados nesta *cloud*. Foram utilizados os serviços do Google *Sheets* para fazer as listagens dos requisitos, o Google *Docs* para os resumos no levantamento de requisitos e o Google *Slides* na construção da apresentação parcial em estilo “PowerPoint”. Todos os documentos estavam sincronizados com a pasta no meu disco físico, o que me permitiu tanto a mim como ao meu orientador de estágio, estar sempre com o material atualizado e disponível, para consulta em qualquer lugar.

5.4. MavensMate

Aquando a etapa de desenvolvimento do projeto envolveu a utilização de *lightning components*, a facilidade e praticidade com que estes podem ser construídos acabou por ser um ponto negativo. O desenvolvimento destes é feito numa “Developer Console”, uma ferramenta embutida no serviço *Salesforce* que permite programar os *lightning components* utilizando o browser. Apesar de prático, esta solução não é eficiente quando o objetivo é desenvolver projetos mais longos, janelas pequenas atalhos inexistentes são algumas das dificuldades.

Com o MavensMate é possível desenvolver os *lightning components* no “conforto” de um editor de texto da preferência do programador (neste caso foi usado o *Sublime Text 3*) com todas as suas funcionalidades de facilidades. O MavensMate faz o *deploy* e controlo de versos nos servidores do *Salesforce* (Figura 32).

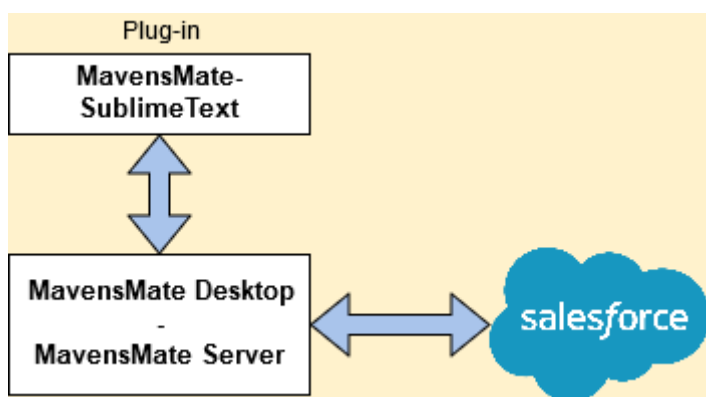


FIGURA 27- MAVENS MATE FORMA DE RELACIONAMENTO

Tudo o que é necessário é instalar um plug-in de controlo de pacotes no Sublime Text, instalar o MavensMate Desktop, que como já está integrado com o *Salesforce* vai pedir as credenciais relativamente a Org onde o trabalho está a ser desenvolvido. A partir daí é possível utilizar o MavensMate para criar, editar ou apagar projetos em *Salesforce* mas possibilitando essas funções a serem desenvolvidas num editor de texto preferido.

6. Implementação

6.1. Layouts

No primeiro layout (Figura 23) podemos ver a página inicial de uma oportunidade (Neste caso denominada de “Projeto Estágio – Lista de requisitos”). Esta página inicial tem várias informações e detalhes sobre a oportunidade, seja o Opportunity Owner o contacto associado a oportunidade. O estado atual em que se encontra a oportunidade que pode ser identificado pela barra de quatro etapas (levantamento de necessidades, investigação, negociação e fecho da oportunidade).

The screenshot shows the Salesforce interface for an Opportunity record. At the top, there's a navigation bar with 'Listagem de Requi...', 'Opportunities', and 'Requisitos'. The main header displays the Opportunity Name 'Projecto Estágio - Lista de Requisito Micael Capelão' and the Owner 'Micael Capelão'. Below this, a progress bar shows four stages: 'Levantamento de necessidades', 'Investigação', 'Negociação' (highlighted in blue), and 'Fecho'. The 'Details' section contains a table with the following data:

Field	Value
Opportunity Name	Projecto Estágio - Lista de Requisito Micael Capelão
Opportunity Owner	Micael Capelão
Account Name	Dom Digital
Close Date	01/08/2018
Probability (%)	75%
Stage	Negociação
Type	
Close Date	01/08/2018

Below the details is a section for 'Planificação e Inserção de valores'. To the right, the 'Related' section shows 'Requisitos (3+)' with a list of three requirements:

- Requirement: Fazer uma lista de requisitos para os layouts
Hours: 2
- Requirement: Criar uma imagem da Lista de requisitos em...
Hours: 5
- Requirement: Mostrar o painel de requisitos, onde é possi...
Hours: 3

FIGURA 28- PERFIL DE UMA OPORTUNIDADE

Por baixo dos detalhes da oportunidade podemos encontrar uma aba “Planificação e inserção de valores” que após expandida (Figura 24):

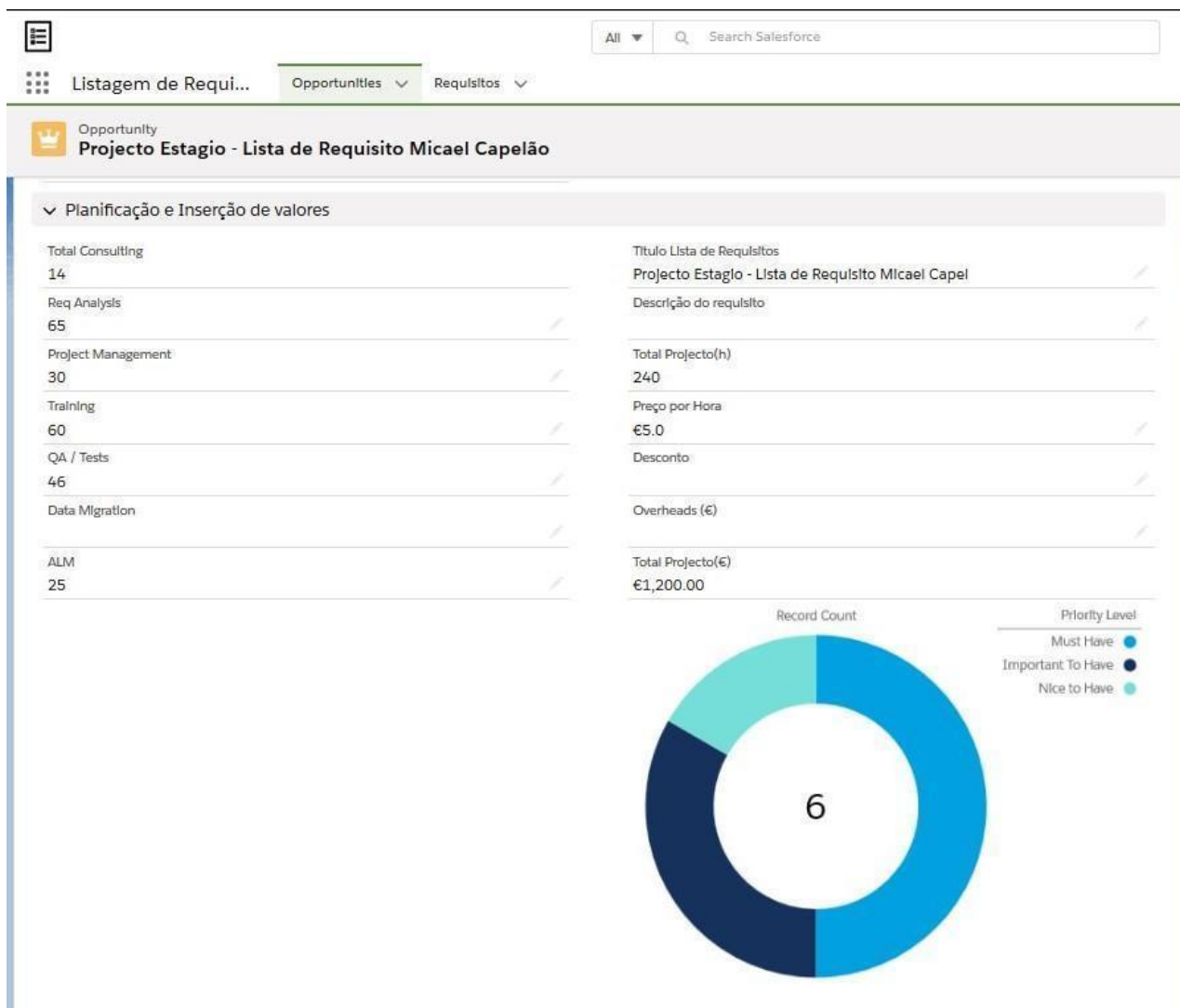


FIGURA 29- LAYOUT PLANIFICAÇÃO E INSERÇÃO DE VALORES, E DASHBOARD

Esta aba permite atribuir valores à orçamentação do projeto mediante as diferentes necessidades, sendo o “*Total consulting*” o campo resultante da soma das horas estipulada para cada requisito. “*Req Analysis*” as horas planeadas para a Análise de Requisitos.

“*Project Management*” horas planeadas para a gestão do projeto. “*Training*” tempo necessário para instruir a equipa de *developers* em uma nova tecnologia ou serviço. “*QA/Tests*” o tempo que vais ser despendido em Testes e Qualidade. “*Data Migration*” Caso seja necessário fazer a migração de dados, o tempo planeado para esta. “*ALM*” (*application lifecycle management*) planificar o tempo de gestão do ciclo de vida da aplicação.

É possível estipular um preço à hora, que mediante serem acrescentados gastos extras (Overheads) vai ser calculada uma estimativa do valor final do projeto.

É possível obter também informações via um *Dashboard* que organiza de forma gráfica os requisitos já existentes, neste caso o *Dashboard* lista os requisitos pelo nível de prioridade.

No menu da página inicial da oportunidade encontramos também a opção “Listar os Requisitos”, é um botão que vai abrir um pop-up em lightning componente que tem como finalidade listar de forma resumida os requisitos da oportunidade (Figura 25).

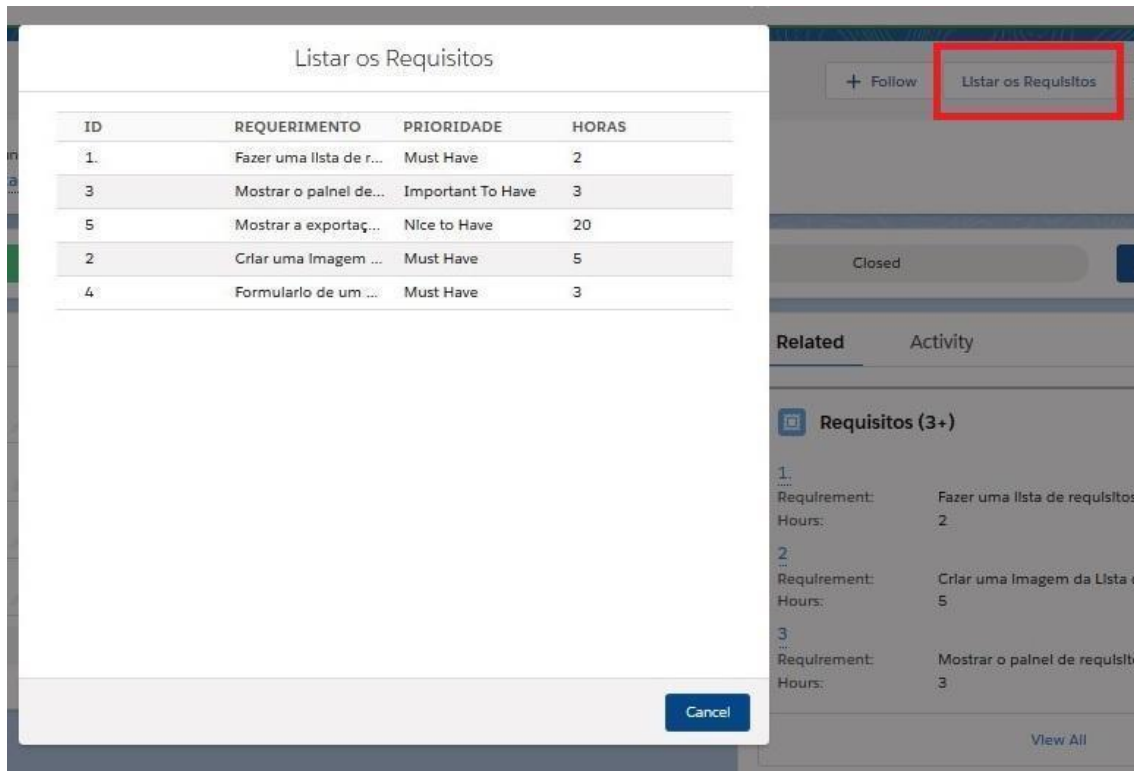


FIGURA 30- LAYOUT LISTAR REQUISITOS

Do lado direito da página inicial temos um breve feed dos requisitos pertencentes a oportunidade. Selecionando a opção “View All” é redirecionado para a página dos requisitos específicos a oportunidade (Figura 26), onde é possível consultar, ordenar ou editar os requisitos já existentes assim como criar um novo. É também possível filtrar usando a ferramenta de filtros do *Salesforce*.

Listagem de Requi... Opportunities Requisitos

Opportunities > Projecto Estagio - Lista de Requisito Micael Capelão

Requisitos

5 items · Sorted by Requisito Numero · Updated a few seconds ago

Requisi... ↑	Requirement	Hours	
1	1. Fazer uma lista de requisitos para os layouts	2	▼
2	2. Criar uma imagem da Lista de requisitos em forma resumida	5	▼
3	3. Mostrar o painel de requisitos, onde é possível editar e criar um Novo Requisito	3	▼
4	4. Formulário de um novo requisito generico	3	▼
5	5. Mostrar a exportação possível em CSV e PDF	20	▼

FIGURA 31- LAYOUT TODOS OS REQUISITOS

Selecionando “New” é apresentado um formulário (Figura 27) de criação de um novo requisito, com todas as opções já descritas na lista de requisitos referente ao projeto (em 4.2.1). Denoto que o “Requisito Número” foi a solução encontrada para manualmente atribuir um valor numérico ao requisito, de forma a possibilitar a inserção de numeração que indicasse uma relação com o antecedente (exemplo: 1 e 1.1).

New Requisito: Teste

Information

* Opportunity: Projecto Estagio - Lista de Requisito Micael... ✕

* Requisito Numero: 6

Priority Level: Must Have ▼

Requirement: Corpo do requisito!

Solution Types: Config ▼

Comments and Questions:

Brought up by: DomDigital ▼

Hours:

Requiriments Status: Open ▼

Cancel Save & New Save

FIGURA 32- FORMULÁRIO NOVO REQUISITO

Voltando ao ecrã principal existe outro botão “Resumo“ (Figura 28) onde podemos obter as opções de exportação dos requisitos. É possível exportar para um formato com uma finalidade mais técnica como o CSV (*Comma-separated values*), ou para um formato mais organizado como o PDF. Estas exportações incluem também os valores da orçamentação e estimativa de horas.

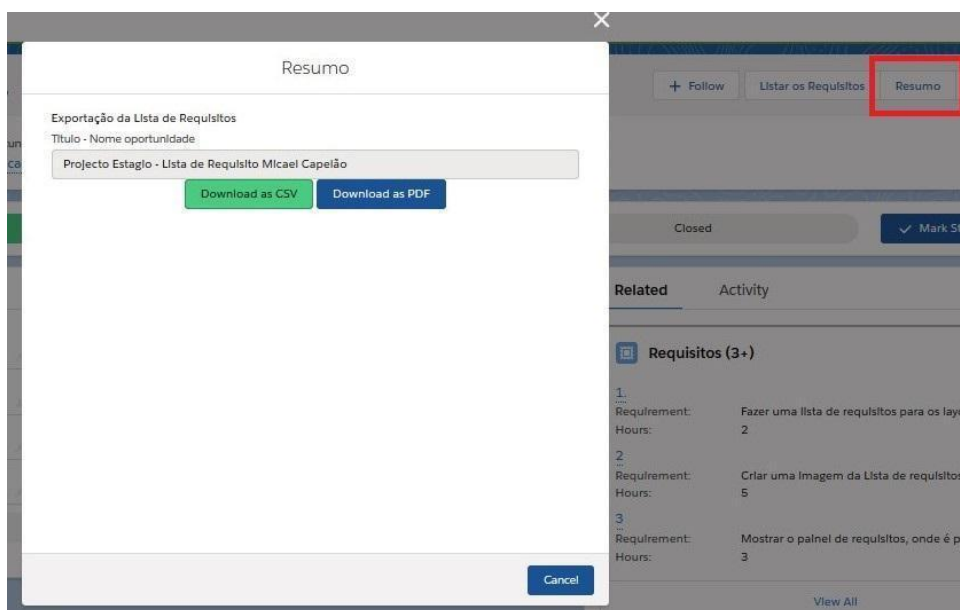


FIGURA 33- EXPORTAÇÃO DA LISTA DE REQUISITOS

7. Conclusão

A título de conclusão, podemos considerar que os objetivos foram alcançados neste estágio.

Relativamente ao ambiente *Salesforce*, toda a ambientação com as formações e Trailheads foram essenciais para a implementação do projeto e neste modo é evidente a agilidade com que permitiu desenvolver soluções, reconheço também as capacidades do *Salesforce* e a mais valia que foi a sua aprendizagem.

No âmbito da Engenharia de Requisitos, já era consciente da sua importância mais a nível teórico/académico, mas após estar presente nesta área num âmbito de contexto real é reconhecida uma importância ainda maior, seja a forma como facilita o trabalho a quem desenvolve, seja a forma como é possível reproduzir informações técnicas numa linguagem que clientes possam entender.

É muitas vezes realçado ao longo do relatório a importância da Análise de Requisitos, a importância de bem explicitar o que se quer fazer e como. Todavia é importante realçar que estas tarefas num contexto real são normalmente responsabilidade de profissionais com conhecimento profundo na plataforma usada, conhecendo assim as suas capacidades e limitações, e considerando este saber podem definir de melhor e mais prática maneira os requisitos. A razão da Análise de Requisitos acontecer desta maneira teve um impacto real neste relatório, ou seja, aquando o tratamento dos requisitos foi também o momento em que se deu a ambientação e contacto com a plataforma, então certos requisitos que foram definidos no início, não foram implementados na parte final. O sucesso parcial na implementação de alguns desses requisitos deveu-se ao já descrito no relatório, a falta de liberdade de um SaaS onde certas funções não são permitidas de fazer. Por isso devido ao não conhecimento profundo da plataforma no momento da listagem dos requisitos, estes podem (ainda que corretamente definidos) não ter sido explicitados da maneira mais prática e funcional para a plataforma em casa. Isto mais do que qualquer conhecimento teórico foi para mim a definitiva realização da importância da análise de requisitos, onde mesmo com tudo organizado, pensado e explicitado é possível haver falhas ou sucessos só parciais no desenvolvimento de *software*.

Como Trabalho futuro pretende-se usar as ferramentas e serviços que se encontram no *Salesforce* para efetuar testes de controlo e qualidade. Nenhum *software* se encontra

realmente concluído sem serem efetuados os respetivos testes. Pretende-se também fazer uma revisão e adição de possíveis funcionalidades como descritas na Listas de requisitos do projeto implementado.

Referências Bibliográficas

- [1] R. Ruggieri, “Requisitos quando não são Elucidados durante o Projeto.,” 11 2015 . [Online]. Available: <https://www.linkedin.com/pulse/requisitos-quando-n%C3%A3o-s%C3%A3o-elucidados-durante-o-projeto-ruggieri/>. [Acedido em 7 2018].
- [2] I. Sommerville, *Software engineering*, England: Pearson Education Limited, 2016.
- [3] I. I. o. B. Analysis, Business Analysis Body of Knowledge, Toronto, Ontario, Canada.: International Institute of Business Analysis, 2015.
- [4] O. Semusheva, “Requirements. Why is it important?,” [Online]. Available: <https://steelkiwi.com/blog/requirements-why-it-important/>. [Acedido em 11 2019].
- [5] A. M. D. S. FILHO, “devmedia,” 2008. [Online]. Available: <https://www.devmedia.com.br/artigo-engenharia-de-software-3-requisitos-nao-funcionais/9525>. [Acedido em 11 2019].
- [6] E. Figueiredo, “homepages.dcc.ufmg.br,” [Online]. Available: https://homepages.dcc.ufmg.br/~figueiredo/disciplinas/aulas/req-funcional-rnf_v01.pdf. [Acedido em 11 2019].
- [7] “Techopedia,” [Online]. Available: <https://www.techopedia.com/definition/26714/on-premises-software>. [Acedido em 11 2019].
- [8] M. Rouse, “Searchcloudcomputing.techtarget.com,” [Online]. Available: <https://searchcloudcomputing.techtarget.com/definition/cloud-computing>. [Acedido em 11 2019].
- [9] G. Čandrić, “globaldots.com,” 19 03 2013. [Online]. Available: <https://www.globaldots.com/blog/cloud-computing-types-of-cloud>. [Acedido em 11 2019].

- [10] M. Technosoft. [Online]. Available: <https://msatechnosoft.in/blog/tech-blogs/cloud-computing-types-advantages-limitations/attachment/types-of-cloud-computing-msa-technosoft>. [Acedido em 11 2019].
- [11] “Techopedia SaaS,” [Online]. Available: <https://www.techopedia.com/definition/155/software-as-a-service-saas>. [Acedido em 11 2019].
- [12] A. Sraders, “What is *Salesforce* and What Does It Do in 2018?,” 4 December 2018. [Online]. Available: <https://www.thestreet.com/technology/what-is-salesforce-14796378>. [Acedido em 11 2019].
- [13] “Recognitio-*Salesforce*,” *Salesforce*, [Online]. Available: <https://www.salesforce.com/company/recognition/>. [Acedido em 11 2019].
- [14] C. Smith, “DMR,” 6 September 2019. [Online]. Available: <https://expandedramblings.com/index.php/salesforce-statistics/>. [Acedido em 11 2019].
- [15] C. M. C. R. Phil Choi, *Lightning Platform Fundamentals*, 2016.
- [16] F. P. d. S. R. Belfo, “Especificação de Requisitos de *Software*,” 22 Outubro 2012. [Online]. Available: <http://www.iscac.pt/files/cursos/ucs/1540002255.pdf>. [Acedido em 11 2019].
- [17] lucidchart, “Lucidchart,” [Online]. Available: <https://www.lucidchart.com/pages/pt/o-que-e-diagrama-entidade-relacionamento>.
- [18] R. Coutinho, “slideshare,” 5 Agosto 2016. [Online]. Available: https://pt.slideshare.net/dom_digital/lightning-components-o-que-so-quais-os-seus-objectivos-e-exemplos-prticos. [Acedido em 07 2018].

Anexos

Exportação CSV

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Nome,Requerimentos,Nivel de Prioridade,Horas,Id																				
2	1,,"Fazer uma lista de requisitos para os layouts","Must Have","2","a090X00000jRoAyQAK"																				
3	3,,"Mostrar o painel de requisitos, onde é possível editar e criar um Novo Requisito","Important To Have","3","a090X00000jRoB8QAK"																				
4	5,,"Mostrar a exportação possível em CSV e PDF","Nice to Have","20","a090X00000jRoB9QAK"																				
5	2,,"Criar uma imagem da Lista de requisitos em forma resumida","Must Have","5","a090X00000jRoB3QAK"																				
6	4,,"Formulario de um novo requisito generico","Must Have","3","a090X00000jRoB4QAK"																				
7	Total Consulting,Req Analysis,Project Management,Training,QA / Tests,Data Migration,ALM,Total Projecto Horas,Preço,Desconto %,Overheads à,-,Total Projecto à,-																				
8	33,"65","30","60","46","","25","259","5","","","1295"																				
9																					
10																					
11																					
12																					
13																					
14																					
15																					
16																					
17																					
18																					
19																					
20																					
21																					
22																					
23																					
24																					
25																					
26																					
27																					

Exportação PDF

Exportação Requisitos para a Oportunidade:

- Projecto Estagio - Lista de Requisito Micael Capelão

Requisito Numero	Requirement	Priority Level	Hours
1.	Fazer uma lista de requisitos para os layouts	Must Have	2
3	Mostrar o painel de requisitos, onde é possível editar e criar um Novo Requisito	Important To Have	3
5	Mostrar a exportação possível em CSV e PDF	Nice to Have	20
2	Criar uma imagem da Lista de requisitos em forma resumida	Must Have	5
4	Formulario de um novo requisito generico	Must Have	3

Horas	
Total Consulting	33
Req Analysis	65
Project Management	30
Training	60
QA Teste	46
Data Migration	
ALM	25
Total Projecto(h)	259

Orçamentação	
Preço	5.0
Desconto(%)	
OverHeads(€)	
Total Projecto(€)	1295.00