

Mestrado em Engenharia Informática

Estágio

Relatório Final

Plataforma de gestão de clientes e fornecedores

Erbi Figueiredo Silva

efs@student.dei.uc.pt

Orientador de DEI-FCTUC:

Prof. Dr. Raul Barbosa

Orientador da Borgstena:

Eng. Pedro Figueiredo

Data: 06 de Fevereiro de 2017



FCTUC DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Mestrado em Engenharia Informática

Estágio

Relatório Final

Plataforma de gestão de clientes e fornecedores

Erbi Figueiredo Silva

efs@student.dei.uc.pt

Elementos do Júri:

Prof. Dr. Mário Rela

Prof Dr. Paulo Simões



FCTUC DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Dedicatória

Dedico todo o trabalho e resultados ao meu pai que, na impossibilidade de ler, ver e/ou acompanhar o seu filho de forma física, estará sempre no coração de todos. Pela pessoa que era e pelos valores que transmitiu. Por ter sido sempre ele a apostar na minha vida e formação académica e, por ser aquele que jamais me deixou “atirar a toalha ao chão” quando as coisas não corriam bem. Pelos ensinamentos humildes, a saudade e o amor que deixou aos seus filhos e sua família. É a ele que agradeço tudo o que tenho, tudo o que sou e tudo aquilo que fiz e irei fazer.

“Não lamento a tua morte, celebro a tua vida” – Jimmy P.

A ele agradeço, a ele devo e a ele dedico.

Obrigado e até já, Pai!

Erbi Figueiredo Silva

Fevereiro de 2017

Agradecimentos

Em primeiro lugar à Borgstena pela oportunidade fantástica de realizar este estágio numa empresa de tão nome como esta. Deixar um especial agradecimento aos orientadores Pedro Figueiredo e ao João Valente pela sua disponibilidade, entrega, profissionalismo e amizade. Estiveram sempre disponíveis quando as dúvidas surgiam.

Não poderia deixar de agradecer aos restantes membros da equipa de IT da empresa, ao Joaquim Marques, Luís Martins e Jorge Silva que, de igual forma foram prestáveis, profissionais e sempre com uma disponibilidade e profissionalismo incríveis.

Tenho também a agradecer ao Jorge Ribeiro, um grande amigo que em muito tem sido importante na minha vida académica e, foi quem me abriu portas para entrar em contacto com os responsáveis para este estágio.

Também deixar um especial agradecimento aos professores e amigos que me acompanharam desde o início desta aventura académica e, que me ajudaram a alcançar os meus objetivos, em especial aos que nos momentos mais difíceis não deixaram de estar presentes, são eles o André Dinis, Celso Mendes, Ricardo Simões, Omar Assaad e Marco Pereira.

Quero agradecer também ao meu primo Licínio Ferreira por momentos de descontração e apoio nesta vida que ele tão bem conhecia e, por me ter orientado sempre que assim precisei.

Sem deixar de lado e, voltando novamente aos amigos, àqueles que me acompanham no dia a dia e que sem eles era impossível ter ultrapassado as barreiras que foram ultrapassadas, são eles o David Batista e ao Stefan Loureiro.

Aqui presto também um agradecimento muito especial a um dos maiores apoios da vida, à Cristiana Figueiredo pelo seu carinho e paciência nos momentos mais complicados.

Por último, mas não menos importante, a toda a minha família e, em especial à minha mãe e irmã, pelo apoio, motivação, incentivo e forças que me deram ao longo deste percurso depois de um grande percalço nas nossas vidas. Sem eles seria impossível estar a lutar pelos meus objetivos e, um grande obrigado pela aposta deles na minha formação. À família, que é o maior apoio de todos e, a eles um grande e sincero agradecimento.

Erbi Figueiredo Silva

Fevereiro de 2017

Resumo

Neste relatório é documentado o trabalho realizado pelo aluno Erbi Figueiredo Silva, no âmbito do estágio de Eng. de Software do Dep. Eng. Informática da Universidade de Coimbra.

O estágio decorreu nas instalações da Borgstena, uma multinacional Sueca que opera na área da indústria têxtil, e o objetivo foi criar um sistema que efectue a gestão de reclamações de clientes e a fornecedores, a gestão de planos de ação e ordens de compra. Este software, permite à empresa recolher informação acerca de duas entidades vitais para o funcionamento de uma empresa (clientes e fornecedores), permitindo assim melhorar as suas capacidades de trabalho para com as mesmas. A imagem de uma companhia é fundamental para a constante melhoria nos seus negócios, por isso é fundamental que os processos de comunicação para com as entidades referidas estejam também em constante melhoria.

Neste relatório encontram-se descritas cada fase do projeto, desde o planeamento à especificação da implementação, passando pelo estudo do estado da arte com a análise de ferramentas concorrentes, ferramentas de suporte e pela especificação de requisitos. Termina com as conclusões sobre o trabalho realizado e desenvolvimentos futuros.

O software desenvolvido foi testado e foi colocado em funcionamento na produção da empresa, demonstrando ser uma ferramenta sólida e eficiente, estando de encontro ao que era pretendido.

Palavras-Chave

Gestão de reclamações, Gestão de ordens de compra, Supplier Evaluation

Conteúdo

Lista de Figuras.....	xv
Lista de tabelas.....	xvii
Capítulo 1: Introdução	1
1.1 Entidade Acolhedora	1
1.2 Âmbito	2
1.3 Objetivos.....	2
1.4 Estrutura do documento	3
Capítulo 2: Gestão de projeto.....	5
2.1 Planeamento	5
2.1.1 Primeiro semestre de estágio.....	5
2.1.2 Plano para o Segundo semestre de estágio	6
2.1.2.1 Work Breakdown Structure	7
2.1.2.2 Estimativa de tarefas	7
2.1.2.3 Estimativa Bottom-up	7
2.1.2.4 Processo de estimação	7
2.1.3 Trabalho efetivo do Segundo Semestre	10
2.2 Acompanhamento	12
2.3 Metodologia de desenvolvimento	12
2.4 Gestão de riscos	13
Capítulo 3: Estado da arte.....	17
3.1 Análise de processos.....	17
3.1.1 Gestão de reclamações	18
3.1.2 Gestão de planos de ação	21
3.1.3 Gestão de ordens de compra	22
3.2 Análise de concorrência	25
3.2.1 CRM	25
3.2.1.1 Salesforce.....	25
3.2.1.2 NABD.....	26
3.2.1.3 Pipedrive	28
3.2.1.4 Conclusão.....	29
3.2.2 SRM	29
3.2.2.1 Gatewit – Gestão de fornecedores	29
3.2.2.2 Gatewit – Auctions.....	31
3.2.2.3 Gatekeeper	32
3.2.2.4 Conclusão.....	33
3.3 Análise interna	33
3.3.1 Industrial Financial System (IFS).....	34
3.3.2 Supplier Evaluation (SEval)	35
3.3.3 Borgstena Login Application (BLA)	35
3.3.4 Utilizadores finais	36
3.4 Análise de Tecnologias.....	38
3.4.1 PHP	38
3.4.2 Laravel	39
3.4.2.1 HTML e Javascript	40

3.4.2.2	Blade	40
3.4.2.3	Eloquent.....	40
3.4.3	PHP Unit	41
3.4.4	MySQL.....	42
3.4.5	Validação de documentação	43
3.4.5.1	Google API – Authenticator	43
3.4.5.2	Mensagens de texto.....	44
3.4.5.3	Conclusão.....	44
3.4.6	Dropzone.js	45
3.4.7	Chart.js	45
Capítulo 4: Especificação de Requisitos		47
4.1	Requisitos Funcionais	47
4.1.1	MC-IR: Inserir reclamação	48
4.1.2	PA-IPA: Inserir Plano de Ação	49
4.1.3	MF-OC: Ordens de compra.....	50
4.2	Requisitos não funcionais	51
4.2.1	Manutenibilidade	51
4.2.2	Segurança	51
4.2.3	Usabilidade	52
4.3	Restrições.....	52
Capítulo 5: Arquitetura		55
5.1	Camadas	55
5.1.1	Apresentação	56
5.1.2	Negócio.....	57
5.1.3	Dados.....	57
5.2	Componentes	58
5.3	Routing	61
5.4	Login	62
5.5	Validação de documentação	62
5.6	Estruturas de Dados.....	63
Capítulo 6: Trabalho desenvolvido		65
6.1	Protótipos	65
6.1.1	Criar reclamação.....	65
6.1.2	Inserir linhas de reclamação.....	66
6.2	Ajax Request	67
6.3	Mensagens de texto	68
6.4	Eventos do IFS	68
6.5	Dropzone.js	69
6.6	Chart.js.....	70
Capítulo 7: Testes		73
7.1	Verificação de requisitos funcionais	73
7.2	Validação de requisitos funcionais	76
7.3	Avaliação de Heurísticas	77

7.4	Verificação de requisitos não funcionais	79
7.4.1	Manutenibilidade	79
7.4.2	Segurança	79
7.4.2.1	SQL Injection	80
7.4.2.2	Cross-Site Request Forgery	81
7.4.2.3	Cross-Site Scripting	82
7.4.3	Usabilidade	82
Capítulo 8: Conclusão		85
8.1	Requisitos implementados	85
8.2	Dificuldades encontradas	86
8.3	Trabalho futuro	87
Referências		89
Anexo A – Levantamento de Requisitos		93
	MC-IR: Inserir reclamação	93
	MC-AR: Consultar Reclamação	95
	MC-CHR: Consultar histórico de reclamações	96
	MC-E: Estatísticas	97
	MC-RC: Reembolsar Cliente	99
	MC-AR: Ações para reclamações	100
	MC-IS: Integração do Seval	102
	PA-IPA: Inserir Plano de Ação	102
	PA-CPA: Consultar Plano de Ação	103
	PA-NPA: Notificações de Plano de Ação	104
	MF-RC: Registrar contas	105
	MF-MR: Mostrar reclamações	105
	MF-OC: Ordens de compra	106
	MF-AOC: Ações em Ordens de Compra	107
	MF-VE: Ver estatísticas	108
	MF-VA: Ver alterações	109
Anexo B – Diagramas de casos de uso		111
Anexo C – Test cases		113
Anexo D – Validação Requisitos Funcionais		133
Anexo E – Avaliação de heurísticas		135
Anexo F – Google Authenticator		139

Lista de Figuras

FIGURA 1.1 - CONTEXTUALIZAÇÃO DA EMPRESA.....	1
FIGURA 2.1: LISTA DE TAREFAS PLANEADAS DO PRIMEIRO SEMESTRE DE ESTÁGIO.....	6
FIGURA 2.2: LISTA DE TAREFAS PLANEADAS DO SEGUNDO SEMESTRE DE ESTÁGIO.....	6
FIGURA 2.3 - METODOLOGIA ITERATIVA.....	13
FIGURA 2.4: GESTÃO DE RISCOS.....	13
FIGURA 3.1 – CONCEITOS DE IDENTIFICAÇÃO DE PART-NUMBER.....	18
FIGURA 3.2 - FLUXO DE RECLAMAÇÃO SEM PORTAL.....	20
FIGURA 3.3 - FLUXO DE RECLAMAÇÃO COM PORTAL.....	21
FIGURA 3.4 - FISHBONE.....	22
FIGURA 3.5 - FLUXO DE ENCOMENDAS SEM PORTAL.....	23
FIGURA 3.6 - FLUXO DE ENCOMENDAS COM PORTAL.....	24
FIGURA 3.7 - PÁGINA SALESFORCE.....	26
FIGURA 3.8 - NABD, GESTÃO DE RECLAMAÇÕES.....	27
FIGURA 3.9 - PÁGINA PIPEDRIVE.....	28
FIGURA 3.10: HOMEPAGE DO GATEWIT.....	30
FIGURA 3.11 – GATEWIT AUCTIONS.....	31
FIGURA 3.12 – HOMEPAGE GATEKEEPER.....	32
FIGURA 3.13 – BORGSTENA BLA.....	36
FIGURA 3.14 – BORGSTENA BLA INTEGRADO COM O PORTAL.....	36
FIGURA 3.15 – PHP PURO JUNTO COM HTML.....	39
FIGURA 3.16 – EXEMPLO PHP COM LARAVEL.....	39
FIGURA 3.17 – EXEMPLO DE BLADE.....	40
FIGURA 3.18 – EXEMPLO DO ELOQUENT.....	41
FIGURA 3.19 – PHP UNIT LINHA DE COMANDOS.....	41
FIGURA 3.20 – PHP UNIT DATA PROVIDER.....	41
FIGURA 3.21 – PHP UNIT EXEMPLO DE NAVEGAÇÃO PELA PÁGINA.....	42
FIGURA 3.22 – GOOGLE AUTHENTICATOR.....	43
FIGURA 5.1: ARQUITETURA POR CAMADAS.....	56
FIGURA 5.2 - PADRÃO MVC.....	57
FIGURA 5.4 – BIRD EYE VIEW, SOLUÇÃO DO PROBLEMA DE 5.3.....	59
FIGURA 5.4 – BIRD EYE VIEW, ARQUITETURA GERAL.....	60
FIGURA 5.5: FUNCIONAMENTO DE ROUTES.....	61
FIGURA 5.6 – FUNCIONAMENTO DE LOGIN.....	62
FIGURA 5.7 – VALIDAÇÃO DE DOCUMENTAÇÃO.....	63
FIGURA 6.1 - CRIAÇÃO DO CABEÇALHO DA RECLAMAÇÃO.....	65
FIGURA 6.2 - ITENS INSERIDOS NA RECLAMAÇÃO.....	66
FIGURA 6.3 - MÉTODO DE INSERÇÃO DE ITENS NA RECLAMAÇÃO.....	66
FIGURA 6.4 – ATUALIZAÇÃO DE PROGRESSO DE RECLAMAÇÃO.....	67
FIGURA 6.5 – AJAX REQUEST, EXEMPLO.....	67
FIGURA 6.6 – ENVIO DE MENSAGEM ATRAVÉS DO PORTAL.....	68
FIGURA 6.7 – EXEMPLO DE ENCOMENDAS QUE O EVENTO DO IFS DESPOLETOU.....	69
FIGURA 6.8 - ECRÃ DE EXEMPLO DE “DRAG’N’DROP” DO DROPZONE.JS.....	69
FIGURA 6.9 - ECRÃ DE EXEMPLO DE “UPLOAD” DO DROPZONE.JS.....	70
FIGURA 6.10 – DECLARAÇÃO DO CANVAS PARA O CHART.JS.....	70
FIGURA 6.11 – CRIAÇÃO DO GRÁFICO EM CHART.JS.....	70
FIGURA 7.1 – RESULTADO SQL INJECTION BEM SUCEDIDO.....	80
FIGURA 7.2 – EXEMPLO DE POST NO POSTMAN.....	81
FIGURA 7.3 – ERRO DE PREVENÇÃO DE CROSS-SITE REQUEST FORGERY.....	81
FIGURA 7.4 – EXEMPLO DE PREVENÇÃO DE ERROS.....	82
FIGURA 7.5 – EXEMPLO DE AJUDA E DOCUMENTAÇÃO.....	83

Lista de tabelas

TABELA 2.1 – ESTIMATIVAS A 3 PONTOS.....	10
TABELA 2.2 – DESVIO NO PLANEAMENTO DO PROJETO DA TAREFA 3.2.2	11
TABELA 2.3 – DESVIO NO PLANEAMENTO DO PROJETO DA TAREFA 8	11
TABELA 2.4 – TAREFA 9 ANTECIPADA.....	11
TABELA 2.5 - TABELA TEMPLATE DE LEVANTAMENTO DE RISCOS	14
TABELA 2.6 - RISCOS LEVANTADOS	15
TABELA 2.7 - MATRIZ DE RISCOS	15
TABELA 3.1: COMPARAÇÃO DAS CRM ESTUDADAS	29
TABELA 3.2 – CONCLUSÃO DAS SRM	33
TABELA 3.3 - ANÁLISE GERAL DE UTILIZADORES.....	37
TABELA 3.4 – VANTAGENS E DESVANTAGENS DO GOOGLE AUTHENTICATOR VS SMS	44
TABELA 4.1 - DESCRIÇÃO DA TABELA DE REQUISITOS	47
TABELA 4.2 – REQUISITO FUNCIONAL INSERIR RECLAMAÇÃO	48
TABELA 4.3 – REQUISITO FUNCIONAL INSERIR PLANO DE AÇÃO	50
TABELA 4.4 – REQUISITO FUNCIONAL DE ORDENS DE COMPRA	50
FIGURA 5.3 – BIRD EYE VIEW, PROBLEMA DE ACESSO	58
TABELA 7.1 – TEMPLATE DO TESTE DE VERIFICAÇÃO DE REQUISITOS FUNCIONAIS	73
TABELA 7.2 – TABELA DE TESTE SOBRE O REQUISITO MC-IR.....	74
TABELA 7.3 – TABELA DE TESTE SOBRE O REQUISITO MR-IR COM ERRO.....	75
TABELA 7.4 – TEMPLATE DE AVALIAÇÃO DE HEURÍSTICAS	78
TABELA 8.1 – REQUISITOS FUNCIONAIS NÃO IMPLEMENTADOS	86

Acrónimos

<i>Acrónimo</i>	<i>Descrição</i>
<i>CRM</i>	Customer Relationship Management
<i>SRM</i>	Supplier Relationship Management
<i>ORM</i>	Object-Relational Mapping
<i>E/R</i>	Entidade/Relacionamento
<i>IFS</i>	Industrial and Financial System
<i>OC</i>	Ordem de compra
<i>ERP</i>	Enterprise Resource Planing
<i>SMS</i>	Short Message Service

Capítulo 1: Introdução

O presente documento tem como principal objetivo apresentar e descrever com detalhe o trabalho realizado pelo estagiário em questão, durante o ano letivo de 2016/2017 (Estágio Plurianual) na empresa Borgstena¹ com o âmbito de desenvolver uma plataforma de melhoria da comunicação entre esta e seus clientes e fornecedores. Orientado, por parte da faculdade, pelo Prof. Dr. Raul Barbosa e, por parte da empresa, pelo Eng. Pedro Figueiredo e coorientador Eng. João Valente, o estágio foi realizado nas instalações da empresa em questão. O estágio teve uma duração prevista de 2 semestres, onde no primeiro foram feitos o planeamento, o estudo do estado da arte, a especificação de requisitos e o desenho da arquitetura a desenvolver. No segundo semestre procedeu-se à implementação do sistema desenhado e consequentes testes. Posto isto, este documento serve também para reunir o trabalho elaborado ao longo do estágio.

1.1 Entidade Acolhedora

A Borgstena é uma multinacional Sueca que opera na área da indústria têxtil, sendo também conhecida como principal fornecedor de tecidos de automóveis na Escandinávia. Atua em vários pontos do mundo, como por exemplo Suécia, Brasil, China, República Checa, Portugal e Roménia. A empresa é dividida em 3 divisões de negócio, que são elas:

- **Borgstena Group:** esta entidade é responsável pela prestação de serviços comuns às divisões operacionais;
- **Borgstena Textile:** desenvolve, fabrica e comercializa têxteis para automóveis, escritórios ou casas;
- **Trimsol:** uma parceria entre a Borgstena e o grupo Coreano Dual² que permite desenvolver, fabricar e comercializar revestimentos de assentos para automóveis.

É na divisão Borgstena Group que o estagiário se encontra inserido, juntamente com os restantes membros da equipa de desenvolvimento de software. O objetivo da equipa é desenvolver ferramentas para todas as divisões operacionais.

A empresa, tal como mencionado anteriormente, é um fornecedor de têxteis sendo a sua principal área de atuação, o mercado automóvel. Importa clarificar que a mesma funciona também como cliente para a compra de matéria prima, como poderá verificar na imagem seguinte.

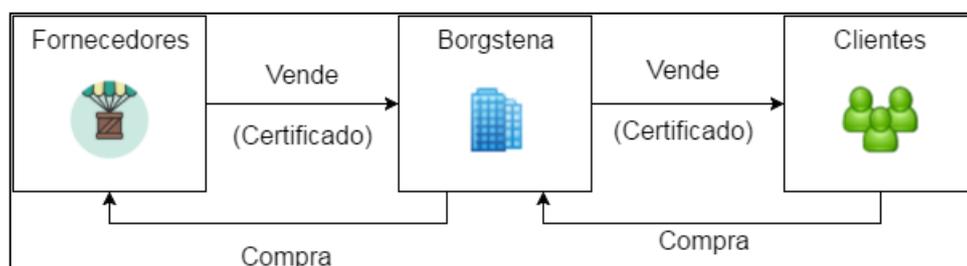


Figura 1.1 - Contextualização da empresa

¹ <http://www.borgstena.com/>

² <http://www.idual.co.kr/>

1.2 Âmbito

No ambiente empresarial é imperativo que as empresas tenham uma imagem bem clarificada acerca dos seus clientes e fornecedores, mas também é igualmente importante olhar para a imagem da própria empresa. Para isso, é importante que se realize uma gestão adequada da informação acerca de reclamações que venham dos clientes (importante para melhorar a própria imagem da empresa) no âmbito de melhorar o funcionamento da mesma. Armazenar e analisar informação de reclamações não só permite aumentar a performance de trabalho como também permite conhecer melhor o cliente, o que acaba por facilitar a comunicação com o mesmo.

De igual modo, uma boa gestão das reclamações que vão da empresa para os fornecedores facilita no processo de saber com quem se está a trabalhar (com um conjunto de reclamações é possível avaliar determinada entidade). Como exemplo, se existir um fornecedor ao qual já foram feitas várias reclamações, este poderá ser um mau indicador para o negócio e a empresa poderá optar por trabalhar com outro. Também o processo de encomendas é afetado pela avaliação dos fornecedores no foro interno da empresa.

Posto isto, é imperativo que se recorra a software que permita realizar a gestão dos dados que compõe cada um dos processos acima referidos. A este tipo de software que permite a melhoria da comunicação com clientes e fornecedores chama-se CRM (“Customer Relationship Management” ou em português “Gestão da relação com clientes”) e SRM (“Supplier Relationship Management” ou em português “Gestão da relação com fornecedores”) respetivamente.

1.3 Objetivos

De acordo com o âmbito apresentado, verificou-se que existe a necessidade da implementação de uma ferramenta que permita melhorar a comunicação com clientes e fornecedores. Apesar de já existirem diversas aplicações disponíveis, estas tendem a abranger um amplo mercado com vários clientes e vários mercados com localizações geográficas diferentes, não sendo possível adaptar essas aplicações às necessidades reais do contexto de uma empresa com o nível desta.

Posto isto, torna-se essencial esclarecer que uma ferramenta CRM (Customer Relationship Management) ajuda os utilizadores a gerir informação e melhora a comunicação com o cliente sendo que as ferramentas SRM (Supplier Relationship Management) facilitam os mesmos processos, mas para o lado dos fornecedores.

O objetivo do estágio foca-se no desenvolvimento de uma ferramenta que integre funcionalidades de CRM e de SRM. Esta, contém 3 módulos distintos, são eles:

- **Gestão de reclamações (de clientes e a fornecedores):** módulo que permite aos utilizadores inserir reclamações e analisar estatísticas sobre custos, quantidades reclamadas, estado dos problemas, entre outros;
- **Gestão de planos de ação:** módulo que permite gerir os planos de ação aplicados a determinada reclamação;

- **Gestão de ordens de compra:** módulo que permite ao fornecedor validar itens de uma determinada ordem de compra, disponibiliza informação acerca da avaliação efetuada pela Borgstena ao fornecedor autenticado, entre outros.

Importa ainda esclarecer o leitor que o último módulo será implementado num servidor à parte uma vez que o objetivo é facultar acesso de terceiros a informação interna e é do interesse da empresa salvaguardar possíveis ataques a informação que esteja noutros servidores.

De momento, existe um software desenvolvido na empresa que permite realizar a avaliação a fornecedores sendo que este deverá também ser integrado na plataforma externa para informar o utilizador da forma como a Borgstena o avalia.

Sendo este projeto iniciado em fevereiro do ano de 2016, ao longo do documento vai ser possível encontrar todo o trabalho e evolução pessoal e técnica do estagiário paralelamente à evolução do projeto em questão. De forma mais concreta, o documento irá conter todo o trabalho do estagiário, especificação de requisitos, configurações e instalações necessárias, planeamento de tarefas e modelos de dados, descrição do trabalho de desenvolvimento.

1.4 Estrutura do documento

O presente documento tem como objetivo documentar o trabalho realizado pelo estagiário ao longo do estágio na Borgstena e encontra-se estruturado da seguinte forma:

- **Capítulo 2 – Gestão de projeto:** neste capítulo é possível encontrar toda a forma como o projeto foi gerido durante o período de estágio. O planeamento que foi efetuado, o acompanhamento que o estagiário obteve e de que formas, a descrição da metodologia a utilizar e a gestão de riscos serão pontos que serão abordados neste capítulo.
- **Capítulo 3 – Estado da Arte:** neste capítulo é apresentado o estudo do estado da arte, análise de soluções encontradas, análise do modo de processos da empresa e formas de trabalhar e a análise de tecnologias estudadas e escolhidas para o desenvolvimento da plataforma.
- **Capítulo 4 – Especificação de requisitos:** este capítulo tem toda a análise efetuada no que toca a levantamento de requisitos funcionais e não funcionais, bem como diagramas de casos de uso e devida estrutura de dados prevista para a plataforma a desenvolver.
- **Capítulo 5 – Arquitetura:** neste capítulo é possível encontrar toda a análise e escolhas arquiteturais tomadas para o desenvolvimento da plataforma. Encontra-se também o funcionamento arquitetural das APIs e Framework que se irá utilizar.
- **Capítulo 6 – Trabalho desenvolvido:** neste capítulo é possível encontrar todo o trabalho desenvolvido pelo estagiário até à data. A implementação de protótipos, de pequenos exemplos de implementação das APIs e aplicações internas que foram utilizadas no projeto.
- **Capítulo 7 – Testes:** neste capítulo estão detalhados os testes que o estagiário aplicou ao trabalho desenvolvido por forma a encontrar e resolver possíveis problemas no software.

- **Capítulo 8 – Conclusão:** neste capítulo estão detalhadas as conclusões gerais acerca de todo o trabalho efetuado, quantidade de requisitos implementados, dificuldades encontradas e o trabalho futuro no portal.

Capítulo 2: Gestão de projeto

Neste capítulo encontram-se descritos os métodos abordados para gerir o projeto que foram utilizados ao longo do estágio. A forma de abordar um problema é uma das tarefas essenciais para um Engenheiro e, uma abordagem correta levará o projeto a ter uma maior probabilidade de sucesso. Com isto, importa clarificar que este capítulo se encontra dividido em 4 secções distintas, sendo elas:

- **Planeamento:** nesta secção encontra-se descrito o planeamento realizado pela empresa em conjunto com o aluno estagiário, com o intuito de definirem as datas e tarefas a realizar por este último. Também será possível encontrar os diagramas de Gantt [1, 2] relativos a cada semestre;
- **Acompanhamento:** nesta secção está descrita a forma como o estagiário é acompanhado pela faculdade e pela empresa em questão, bem como a forma como se organizaram as reuniões entre o aluno e a Borgstena;
- **Metodologia de desenvolvimento:** nesta secção é apresentada a metodologia de desenvolvimento escolhida, bem como todo o processo de pesquisa e análise referente à mesma;
- **Gestão de riscos:** nesta secção é apresentada a forma como foi abordada a gestão de riscos, bem como o levantamento dos mesmos. Contém também uma matriz de riscos que permitiu uma análise de riscos mais rápida para que se pudesse elaborar planos de mitigação para os mesmos.

2.1 Planeamento

Nesta secção é possível consultar o diagrama de Gantt com o respetivo planeamento de tarefas desde o início até à data de fim de estágio.

Os pontos que se seguem apresentam o planeamento dedicado ao primeiro semestre de estágio (2º semestre do ano letivo 2015/2016) e, ao segundo semestre de estágio (1º semestre do ano letivo 2016/2017). Os diagramas encontram-se divididos em tarefas. Como é possível observar, algumas contêm um conjunto de sub-tarefas, com o intuito de dividir por fases, algum do trabalho a realizar. Com isto será possível obter uma organização melhor.

O tempo alocado ao primeiro semestre foca-se mais na pesquisa de informação sobre a área de estágio tanto a nível do estado de arte, como ferramentas e competidores e, finalmente a escrita do relatório de estágio. No segundo semestre, o tempo será alocado com base na iniciação e conseqüente continuação do desenvolvimento do software pretendido.

2.1.1 Primeiro semestre de estágio

No decorrer do primeiro semestre de estágio que teve início a 29 de fevereiro de 2016, foram realizadas as tarefas apresentadas que se encontram na figura a seguir apresentada. Este semestre foi essencialmente dedicado à ambientação do estagiário no mundo do trabalho e também para que o aluno possa estudar, compreender, documentar e experimentar o problema proposto. Torna-se importante que estas fases sejam realizadas da forma adequada e com tempo pois são preponderantes para que o trabalho seja efetuado da forma correta.

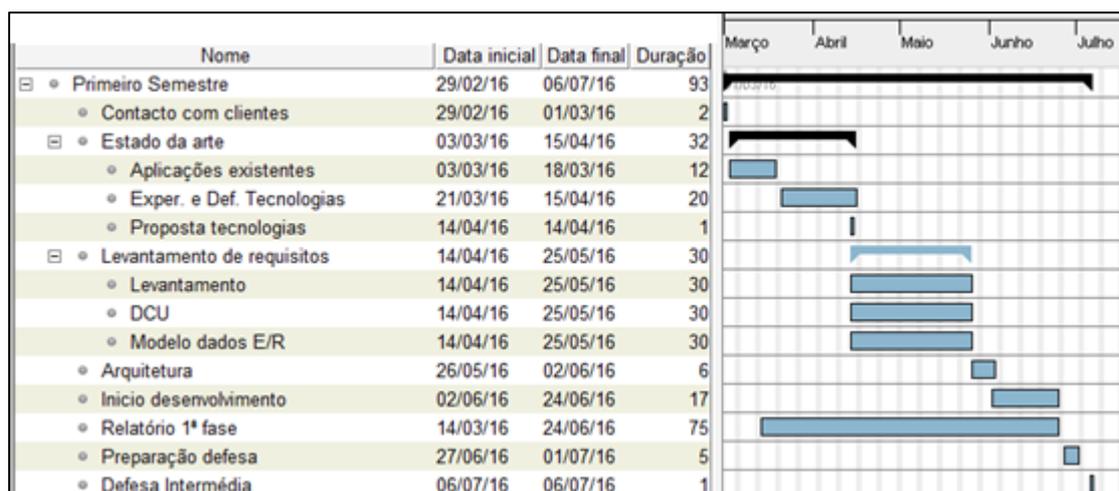


Figura 2.1: Lista de tarefas planeadas do primeiro semestre de estágio

De acordo com a imagem anterior houve alguns pontos que sofreram alterações de datas. Derivado a viagens ao estrangeiro por parte do gestor de fornecedores, a reunião identificada como “Contacto com clientes” foi realizada numa primeira fase apenas com o gestor de clientes e só mais tarde com o gestor de fornecedores, situação esta que em pouco alterou o planeamento, uma vez que assim foi possível separar os assuntos e estudar cada um de forma individualizada, bem como as aplicações existentes.

2.1.2 Plano para o Segundo semestre de estágio

O segundo semestre de estágio, com início a 11 de junho de 2016 (assim considerado pela empresa e pelo estagiário), teve como tarefas planeadas as a seguir apresentadas na imagem que se segue. Este semestre focou-se no desenvolvimento do projeto com base na investigação e análise realizadas no primeiro semestre.

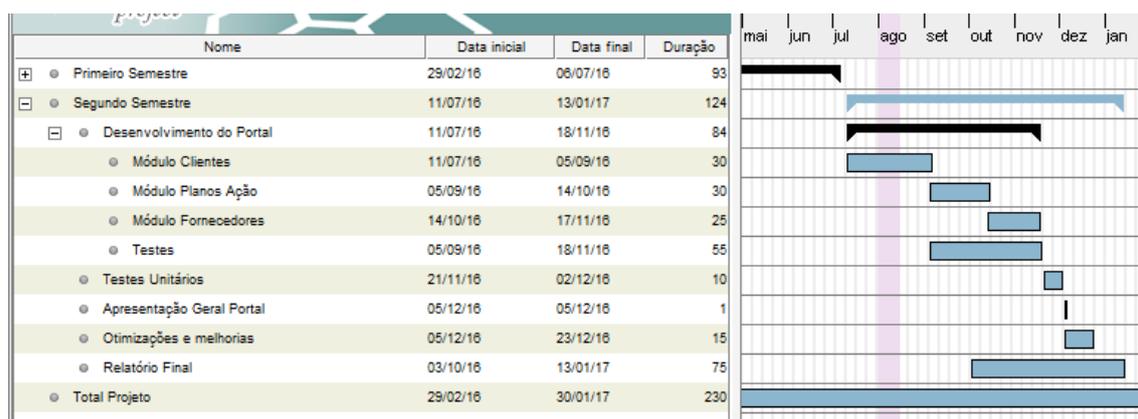


Figura 2.2: Lista de tarefas planeadas do segundo semestre de estágio

Com base na imagem anterior, é possível verificar que o desenvolvimento do portal se encontra dividido nos 3 módulos referidos anteriormente. Estes, por representarem tarefas complexas, foram divididas em sub tarefas mais simples com o intuito de facilitar a gestão e as estimativas de esforço necessário para a sua realização.

Importa ainda clarificar que o planeamento do segundo semestre de estágio sofreu alterações relativamente ao que estava descrito no relatório intermédio. Estas permitiram melhorar a organização do trabalho bem como a especificação das tarefas.

2.1.2.1 Work Breakdown Structure

O Work Breakdown Structure [3] (WBS) consiste no processo de divisão de tarefas mais complexas em outras mais simples. Com isto, é possível realizar estimativas de forma mais precisa e, facilita a gestão do projeto. Uma das regras usadas é a regra do 8/80, que implica que nenhuma tarefa demore menos de 8 horas nem mais que 80 horas. Importa esclarecer que um dia de trabalho é equivalente a 8 horas. Um WBS pode ser apresentado sob várias formas, desde estruturas a listas e tabelas.

2.1.2.2 Estimativa de tarefas

Para realizar a estimativa de esforço necessário para cada tarefa foi utilizado o método Estimativa de Três Pontos ou “Bottom-up Estimation” [4]. Neste, são atribuídos 3 valores de esforço a cada tarefa, que são eles:

- Melhor Caso
- Mais Provável
- Pior Caso

Posto isto, importa explicar a fórmula de cálculo para o valor esperado de esforço, que é a seguinte:

$$\text{(Melhor Caso + (4 x Mais Provável) + Pior Caso) / 6}$$

2.1.2.3 Estimativa Bottom-up

Para se realizar o planeamento foi seguida a abordagem de estimativa Bottom-Up. Como primeiro passo foi criado o WBS, de seguida foi estimada cada tarefa usando a Estimativa de Três Pontos e como último passo, soma-se todas as estimativas para se obter o esforço total.

Esta abordagem envolve seis passos que são fundamentais para que seja efectuada da forma correcta:

1. Seleccionar estimador ou equipa (estagiário)
2. Obter artefactos da aplicação
3. Criar o WBS
4. Estimar com a Estimativa de Três Pontos
5. Calcular esforço do projeto no seu todo
6. Refinação de estimativas se necessário

2.1.2.4 Processo de estimação

Para se realizar o processo de estimação foi necessário criar o WBS cujo o resultado se encontra listado a seguir:

1. Definição do modelo de dados
2. Definição de layouts a utilizar
3. Implementação do Módulo de Reclamações

- 3.1. Inserir reclamação
 - 3.1.1. Implementação do controlador numérico de reclamações
 - 3.1.2. Criação do formulário
- 3.2. Implementação de consulta de reclamação
 - 3.2.1. Mostrar reclamação
 - 3.2.2. Atualizar progresso
 - 3.2.3. Atualizar linha
 - 3.2.4. Inserção de nova linha
- 3.3. Implementação de históricos
 - 3.3.1. Mostrar histórico de reclamações
 - 3.3.2. Filtrar reclamações
- 3.4. Implementação de estatísticas
 - 3.4.1. Mostrar tops
 - 3.4.1.1. De clientes reclamantes
 - 3.4.1.2. De produtos reclamados
 - 3.4.1.3. De defeitos
 - 3.4.1.4. De problemas por mês
 - 3.4.1.5. Custos mensais
 - 3.4.1.6. Custos de reclamações por clientes
 - 3.4.1.7. Custos de defeitos reclamados
 - 3.4.1.8. Custos de produtos reclamados
- 3.5. Implementação de reembolso a cliente
 - 3.5.1. Pedido de autorização de nota de crédito
 - 3.5.2. Geração de código secreto
 - 3.5.3. Envio de sms com código secreto
 - 3.5.4. Validação de nota de crédito
 - 3.5.5. Passagem de nota de crédito pelo Departamento Financeiro
- 3.6. Implementação de ações em reclamações
 - 3.6.1. Atribuição de clientes a funcionários
 - 3.6.2. Atribuir responsáveis de validação de documentação
- 4. Implementação do Módulo de Planos de ação
 - 4.1. Criação de plano de ação
 - 4.1.1. Criar formulário
 - 4.1.2. Preenchimento de dados automático
 - 4.2. Consultar plano de ação
 - 4.2.1. Carregar plano de ação
 - 4.2.2. Editar plano de ação
 - 4.2.3. Exportar
 - 4.2.4. Fechar plano
 - 4.2.5. Anexar ficheiros
 - 4.3. Criar notificações
 - 4.3.1. Notificar vencimentos
 - 4.3.2. Obter calendário e datas de expiração
- 5. Implementação do módulo de fornecedores
 - 5.1. Registo de contas
 - 5.1.1. Registrar fornecedor
 - 5.1.2. Enviar mail com credenciais
 - 5.2. Mostrar reclamações
 - 5.2.1. Carregar reclamação
 - 5.2.2. Inserir plano de ação
 - 5.3. Ordens de compra

- 5.3.1.Criar eventos do IFS
- 5.3.2.Carregar e mostrar ordem de compra
- 5.3.3.Enviar mail a fornecedor
- 5.3.4.Ver lista de ordens de compra abertas
- 5.4. Ações em ordens de compra
 - 5.4.1.Confirmar item de ordem de compra
 - 5.4.2.Sugerir nova data
 - 5.4.3.Rejeitar pedido
 - 5.4.4.Inserir justificação
- 5.5. Estatísticas
 - 5.5.1.Mostrar histórico de encomendas
 - 5.5.2.Mostrar avaliações
- 6. Verificação de requisitos
- 7. Testes Unitários
- 8. Validação de requisitos
- 9. Migração de servidores
- 10. Criação de manual de aplicações

Após a criação do WBS foi aplicada a estimativa a três pontos para cada tarefa. O resultado poderá ser consultado na tabela seguinte.

Tarefa	Melhor Caso	Caso mais provável	Pior Caso	Caso expectado
Módulo Clientes				
1	0.5	1	2	1.08
2	0.5	1	2	1.08
3.1.1	0.5	1	2	1.08
3.1.2	1	2	3	2.00
3.2.1	1	1.5	2	1.50
3.2.2	1	2	4	2.17
3.2.3	1	2	3	2.00
3.2.4	1	1.5	2	1.50
3.3.1	0.5	0.5	1	0.58
3.3.2	0.5	0.5	1	0.58
3.4.1	2	2.5	3	2.50
3.5.1	1	1.5	2	1.50
3.5.2	0.5	1	2	1.08
3.5.3	1	1.5	2	1.50
3.5.4	1	1.5	2	1.50
3.5.5	1	2	2.5	1.92
3.6.1	0.5	0.5	1	0.58
3.6.2	0.5	0.5	1	0.58
TOTAL:	15	24	37.5	24.75
Módulo Planos de Ação				
4.1.1	2	2.5	3	2.5
4.1.2	0.5	1	1.5	1

4.2.1	2	2.5	3	2.5
4.2.2	1	1.5	2	1.5
4.2.3	1	1.5	2	1.5
4.2.4	0.5	1	1.5	1
4.2.5	1	1.5	2	1.5
4.3.1	0.5	1	1.5	1
4.3.2	0.5	1	1.5	1
TOTAL:	9	13.5	18	13.5
Módulo Fornecedores				
5.1.1	0.5	1	1.5	1.00
5.1.2	0.5	1	1.5	1.00
5.2.1	2	2.5	3	2.50
5.2.2	2	2.5	3	2.50
5.3.1	2	3	4	3.00
5.3.2	1	1.5	2	1.50
5.3.3	1	1.5	2	1.50
5.3.4	1	1.5	2	1.50
5.4.1	1	1.5	2	1.50
5.4.2	1	1.5	2	1.50
5.4.3	1	1.5	2	1.50
5.4.4	1	1.5	2	1.50
5.5.1	1	1.5	2	1.50
5.5.2	1.5	2.5	3.5	2.50
TOTAL:	16.5	24.5	32.5	24.5
Testes e validação				
6	5	7	10	7.14
7	5	7	10	7.14
8	5	7	10	7.14
9	2.5	5	7.5	2
10	2.5	5	7.5	5
TOTAL:	8.5	17	25.5	28

Tabela 2.1 – Estimativas a 3 pontos

Este processo revelou-se importante pois permitiu ajustar a complexidade do projeto bem como o esforço necessário para as tarefas em questão de acordo com o tempo de duração do estágio previsto.

2.1.3 Trabalho efetivo do Segundo Semestre

De acordo com as tarefas e estimativas descritas na secção anterior, o trabalho efetivo relativo ao segundo semestre sofreu, em alguns pontos, desvios no tempo de implementação dos mesmos. Os desvios mais significativos no que toca a atrasos foram os a seguir descritos:

Tarefa	Caso esperado	Esforço real
3.2.2	2.17	5
Justificação		
<p>É necessário guardar todas as atualizações de progresso de uma reclamação e, a mesma só pode entrar no estado “fechado” quando:</p> <ul style="list-style-type: none"> • Se existir, o plano de ação estiver no estado “fechado” <ul style="list-style-type: none"> ○ O plano de ação está no estado fechado quando todas as tarefas estiverem marcadas como “executadas”; • Se existirem, todos os pedidos de autorização de notas de crédito se encontrarem concluídos. <p>Inicialmente, o progresso e estado da reclamação seriam atualizados pelo gestor sem condições adicionais como as enunciadas nos pontos em cima descritos.</p> <p>Estas alterações levaram a uma remodelação na estrutura de dados bem como no conteúdo das páginas demonstradas ao utilizador e alterações de código a nível do servidor.</p>		

Tabela 2.2 – Desvio no planeamento do projeto da tarefa 3.2.2

Tarefa	Caso esperado	Esforço real
8	5	5 (atraso no início da tarefa)
Justificação		
<p>Devido a deslocações profissionais ao estrangeiro por parte dos utilizadores finais, esta tarefa não teve início na data marcada, tendo sido adiada.</p>		

Tabela 2.3 – Desvio no planeamento do projeto da tarefa 8

Apesar de se terem efetuado os atrasos registados nas tabelas acima descritas, é de se notar que outras tarefas foram implementadas antes do tempo previsto como o exemplo da que se segue:

Tarefa	Caso esperado	Esforço real
9	5	5 (adiantamento no início da tarefa)
Justificação		
<p>Devido à impossibilidade da tarefa 8 ser executada na data para o efeito, decidiu-se avançar com o desenvolvimento do manual, trocando assim as semanas desta tarefa com a anterior</p>		

Tabela 2.4 – Tarefa 9 antecipada

Importa ainda clarificar que as restantes tarefas sofreram pequenos desvios (atrasos ou adiantamentos) mas estes não foram significativos. O projeto esteve sempre dentro dos tempos previstos. Estas durações não foram registadas como as tarefas em cima mencionadas, uma vez que as diferenças de tempo entre o esforço real e o esforço expectável eram mínimas.

Posto isto, o projeto foi concluído na data inicialmente programada, mantendo-se sempre equilibrado nos tempos previstos.

2.2 Acompanhamento

No que se refere ao acompanhamento do projeto de estágio, da parte da faculdade, o aluno foi orientado pelo Prof. Dr. Raul Barbosa sendo que na empresa o aluno foi orientado pelo Eng. Pedro Figueiredo.

Posto isto, importa ainda referir que por exigência do trabalho, este último designou um coorientador para o acompanhamento ao estagiário, uma vez que foi necessário realizar viagens para fora do país e não teve a possibilidade de acompanhar a cem por cento o avanço de todo o trabalho que foi efetuado. A coorientação foi realizada por parte do Eng. João Valente, um dos elementos da equipa de desenvolvimento de software presentes.

Importa também esclarecer que foram feitas várias reuniões de acompanhamento ao projeto dividindo-se as mesmas em três tipos:

- **Semanais:** com o objetivo de acompanhar o estado e evolução semanal do trabalho efetuado pelo estagiário em questão;
- **Pontuais:** com o objetivo de tirar dúvidas ou apresentar trabalho, em qualquer altura da semana;
- **Apresentações:** estas serão efetuadas quando se pretende dar e obter feedback ao responsável pelo módulo a desenvolver, ou seja, apresentação de trabalho relativo ao módulo de clientes será apresentado ao gestor do mesmo.

Após o finalizar de cada uma das reuniões, foram regidas atas pelo responsável da reunião e enviadas para todos os restantes participantes da mesma com o intuito de guardar todos os assuntos tratados e decididos.

2.3 Metodologia de desenvolvimento

No ambiente de desenvolvimento, a peça chave para o sucesso, é uma boa escolha no que toca à metodologia a aplicar. Uma escolha adequada é fundamental para que todo o processo flua de acordo com o previsto. Dentro de um vasto leque de metodologias existentes, a escolha depende de vários fatores importantes e características inerentes ao projeto.

O projeto conta com apenas um elemento para o seu desenvolvimento, sendo este o próprio estagiário e autor deste documento. Os requisitos são levantados logo no início do projeto seguidos do desenho da arquitetura e design. Após estas fases segue-se o desenvolvimento do sistema, módulo a módulo. Ou seja, é implementado um módulo e são efetuados alguns testes. No total são 3 os módulos que foram desenvolvidos. Posto isto, a metodologia seguida foi a metodologia iterativa [5].

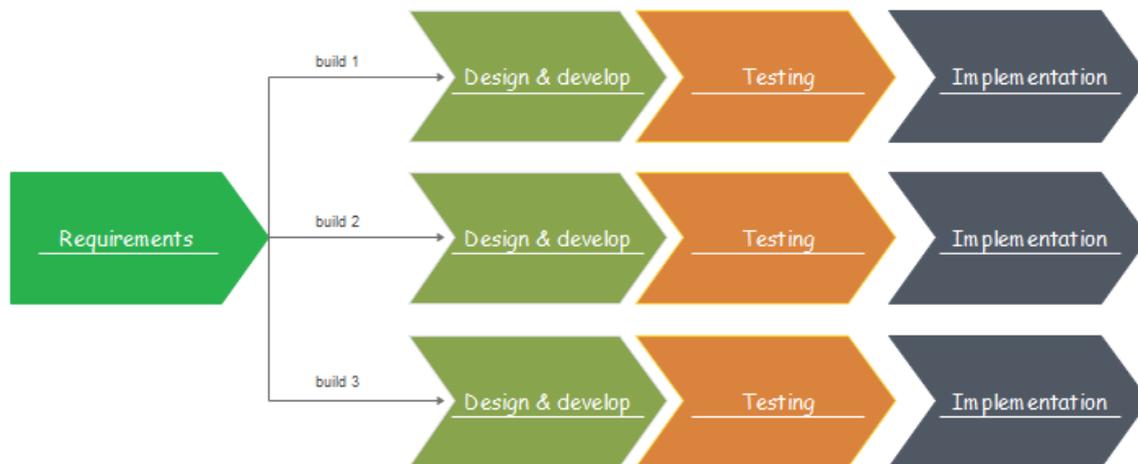


Figura 2.3 - Metodologia iterativa

A escolha desta tecnologia deve-se ao fato de o projeto ter sido dividido em 3 módulos distintos e os requisitos foram definidos e são conhecidos desde o início do mesmo, sendo que cada módulo foi implementado separadamente, depois testado e implementado. Numa primeira fase foi desenvolvido o módulo de reclamações e este foi testado pelo estagiário.

Esta metodologia, contudo, não foi seguida rigorosamente. Ou seja, quando o estagiário passava à implementação do módulo seguinte, o anterior estava implementado e a ser testado por a equipa responsável.

2.4 Gestão de riscos

A gestão de riscos [6] é essencial em qualquer gestão de projeto. É uma abordagem contínua, utilizada com o intuito de minimizar possíveis danos e reduzir a vulnerabilidade do nosso projeto a situações que poderão ocorrer.

Com isto, importa clarificar as seguintes características mensuráveis:

- **Impacto:** representada por 5 níveis (muito alto, alto, médio, baixo e muito baixo), transmite o potencial perigo do risco para com a aplicação;
- **Probabilidade:** representada por 5 níveis (frequente, provável, possível, improvável e raro) transmite a possibilidade de um risco ocorrer.

O ciclo de vida é constituído pelas 3 fases que se seguem na imagem:



Figura 2.4: Gestão de riscos

De acordo com as fases identificadas na imagem anterior, importa clarificar os seguintes conceitos:

- Identificar: fase de identificação do risco que pode ser prejudicial ao projeto;
- Avaliar: fase de avaliação do impacto e probabilidade do risco;
- Planear: fase de criação de um plano de mitigação.

A estrutura de levantamento do risco é a seguinte:

- ID: identificador único de cada risco
- Risco: descrição do risco levantado e sua consequência
- Impacto: atribuição do nível de impacto
- Probabilidade: atribuição do nível de probabilidade
- Solução: descrição do plano de mitigação a aplicar

Toda a documentação da gestão de riscos deve ser simples e rápida de entender, uma vez que caso um dos riscos ocorra, o plano de mitigação seja facilmente encontrado e o problema solucionado. É importante definir um plano de mitigação detalhado para que os problemas sejam facilmente controlados.

Como referido, o levantamento de riscos é uma fase fundamental de um projeto. Com isto, importa ainda esclarecer que o template da tabela que foi utilizada para o levantamento foi o que se encontra a seguir detalhada:

ID	Risco	Imp.	Prob.	Solução
Identificador do risco	Descrição do risco	Valor do impacto	Probabilidade do risco acontecer	Solução de mitigação do risco

Tabela 2.5 - Tabela template de levantamento de riscos

Posto isto e, tendo em consideração o projeto a ser desenvolvido, foram levantados os riscos, sendo que os mais relevantes se encontram a seguir listados:

ID	Risco	Imp.	Prob.	Solução
1	Falta de tempo por parte dos orientadores para acompanhar o projeto, nomeadamente nas fases de análise do problema e levantamento de requisitos; Pode levar a atraso no projeto e/ou análise errada por parte do estagiário	alto	provável	Pré definir reuniões de análise de acompanhamento de projeto
2	Inexperiência em projetos de gestão em ambiente empresarial; Pode levar ao aumento de erros no software, uma vez que o ambiente empresarial é diferente do hábito do estagiário	médio	possível	Reuniões de acompanhamento e preparação adequada no que toca a formação do

				autor sobre as estratégias da empresa
3	Falta experiência nas tecnologias usadas; Pode levar ao atraso no desenvolvimento do projeto	médio	improvável	Formação dada por um dos elementos da equipa no que toca a algumas ferramentas que se conheçam.
4	Dificuldade comunicação com os "clientes" (utilizadores finais); Pode levar a problemas e divergências na análise e levantamento de requisitos	muito alto	provável	O "cliente" informa outra pessoa sobre o projeto enquanto está de viagem
5	Pouco conhecimento sobre estratégias de negócio, nomeadamente da empresa; Pode levar a problemas na análise de requisitos e implementação dos mesmos	baixo	improvável	Pré definir reuniões de análise de acompanhamento de projeto com respetivos gestores
6	Alteração da ideia inicial do stakeholder; Levará à reformulação de todo o trabalho desenvolvido	alto	possível	Definir de início os requisitos mais importantes e, documentar todos os requisitos pedidos desde a primeira reunião (escrita de atas)

Tabela 2.6 - Riscos levantados

Posto isto e, de acordo com a tabela de valores de impacto e probabilidades, foi possível levantar a seguinte matriz de riscos:

	Raro	Improvável	Possível	Provável	Frequente
Muito alto				4	
Alto			6	1	
Medio		3	2		
Baixo		5			
Muito baixo					

Tabela 2.7 - Matriz de riscos

A matriz de risco é uma ferramenta muito útil na visualização rápida dos riscos. Com ela facilmente se identificam quais os prioritários. Os identificadores encontram-se distribuídos ao longo de três áreas diferentes que os classificam, que são as seguintes:

- **Área verde:** riscos que não apresentam consequências significativas para o projeto, podendo aceitar-se as mesmas sem graves danos ao projeto;
- **Área amarela:** são riscos que não apresentam uma ameaça séria, mas é necessário estar preparado caso estes ocorram pois são riscos que podem ter tendência a agravar-se;
- **Área vermelha:** esta é a área crítica da matriz de riscos, onde se apresentam os riscos graves, com elevada probabilidade e um impacto muito significativo e prejudicial ao projeto.

Posto isto, importa ainda clarificar que os riscos número 1, 2, 3 e 6 nunca se materializaram ao longo do projeto. No que toca ao risco número 4, este ocorreu algumas vezes, sendo necessário “ativar” o plano de contingência. O funcionário responsável em causa necessitou de se deslocar por tempo indeterminado e era destacada uma pessoa que ficaria responsável pelo projeto, sendo que este plano foi suficiente para colmatar o problema.

Para o risco número 5 também foi necessário ativar o plano de contingência, uma vez que o estagiário após estudar os processos e estratégias da empresa necessitou de esclarecer algumas situações. Caso estas não fossem esclarecidas, os processos no software poderiam ficar errados e teriam um impacto muito negativo no programa.

Posto isto, importa clarificar que após a ativação dos planos de contingência, os problemas foram solucionados e os riscos não trouxeram impacto ao projeto.

Capítulo 3: Estado da arte

Neste capítulo é descrito todo o estudo realizado sobre o estado da arte no âmbito do projeto de estágio. O capítulo encontra-se dividido em vários grupos, que são os seguintes:

- **Análise de processos:** neste ponto encontra-se detalhado o estudo e análise efetuado ao longo do estágio, aos diferentes processos utilizados pela empresa no que toca à gestão de clientes e fornecedores. Este estudo envolveu o chamado “trabalho de campo”, onde o estagiário se envolveu com os responsáveis em ambiente real para poder ter um conhecimento aprofundado de como a empresa lida e gere com todas as situações que podem ocorrer;
- **Análise de concorrência:** neste ponto encontra-se detalhado o estudo e análise efetuado às demais ferramentas concorrentes que atualmente se encontram disponíveis no mercado;
- **Análise interna:** neste ponto encontra-se detalhado o estudo e análise efetuados no contexto da empresa, numa tentativa de entender que ferramentas existem para ajudar nos processos de gestão e, que tipo de utilizadores serão os potenciais alvos de utilização do portal;
- **Análise de tecnologias:** neste ponto encontra-se detalhado o estudo e análise efetuados sobre tecnologias a utilizar face ao desenvolvimento do portal. Nele é possível verificar que tecnologias foram as escolhidas e, que outras foram estudadas bem como a justificação do porque da escolha ou não das mesmas.

Com isto, importa ainda referir que todas as análises efetuadas foram registadas, documentadas e aprovadas pela empresa.

3.1 Análise de processos

A análise de processos é a análise sobre a forma como a empresa, no momento atual, aborda os assuntos relacionados com a gestão de clientes e fornecedores, mais concretamente o estudo de como se gere toda a informação necessária acerca do produto a desenvolver.

Estes processos, embora pareçam de uma forma geral fáceis e de pouca complexidade, após um estudo aprofundado é possível concluir que são processos que envolvem um pequeno conjunto de tarefas árduas e complexas pois exigem o estudo e compreensão total de todas as normas, leis e estratégias que a empresa designou e envolve em volta dos mesmos. No caso da gestão de reclamações, por vezes, o cliente exige que o tratamento das mesmas siga um fluxo próprio deles, pelo que é necessário proceder a esse estudo e a Borgstena terá de se adaptar a esse cenário.

Posto isto, importa clarificar dois processos fundamentais de estudo que são:

- **Processo de gestão de reclamações:** onde se encontra detalhado o estudo, fluxo atual e pretendido. Este envolve diversos utilizadores e é necessário ter em consideração que o grande objetivo é a redução do tempo que atualmente é necessário para gerir todo o processo, inclusive a gestão de planos de ações.
- **Processo de gestão de planos de ação:** onde se encontra detalhado o estudo da forma como é gerido todo o trabalho relativo a planos de ação, desde a criação do mesmo até à conclusão de todas as tarefas.

- **Processo de gestão de ordens de compra:** onde se encontra detalhado todo o processo necessário para gerir as ordens de compra e que é utilizado atualmente pela empresa. Este envolve entender os conceitos que são utilizados por parte da empresa para encomendar e, como o fornecedor pode inserir as suas ordens de confirmação.

Importa ainda clarificar que todos os processos que se seguem, antes do software ser desenvolvido e implementado, eram realizados com recurso ao Excel [7]. Esta é a ferramenta que é utilizada e onde é efetuado todo o registo de reclamações e planos de ação. O Excel pode ser considerado a maior ERP de todo o mundo uma vez que é uma ferramenta muito utilizada e já há muitos anos.

3.1.1 Gestão de reclamações

Como nas demais empresas, cada uma tem as suas estratégias de comunicação com os clientes. A gestão de reclamações é parte fundamental dessa comunicação, uma vez que permite obter feedback altamente importante no que toca a clientes, isto porque permite corrigir erros, antecipar problemas, analisar os níveis de exigência do cliente, entre outros.

No contexto da Borgstena, quando uma reclamação é efetuada, esta é recebida via mail. Por norma, salvo exceções, o setor de Qualidade é responsável pela gestão de reclamações. Para isso, existe um fluxo interno na empresa que deve ser seguido rigorosamente desde o início do processo da reclamação até ao final da mesma. Posto isto, é importante clarificar os seguintes conceitos:

- **Customer Number:** identificador único que identifica um cliente;
- **Internal Part-Number:** identificador de um artigo da Borgstena;
- **Customer Part-Number:** código que relaciona um artigo da Borgstena a um cliente (associa um produto a uma venda a determinado cliente, por exemplo).

Estes conceitos podem ser visualizados na imagem seguinte onde os dados envolvidos são meramente fictícios e têm apenas o objetivo de explicar a relação entre os três pontos acima descritos.

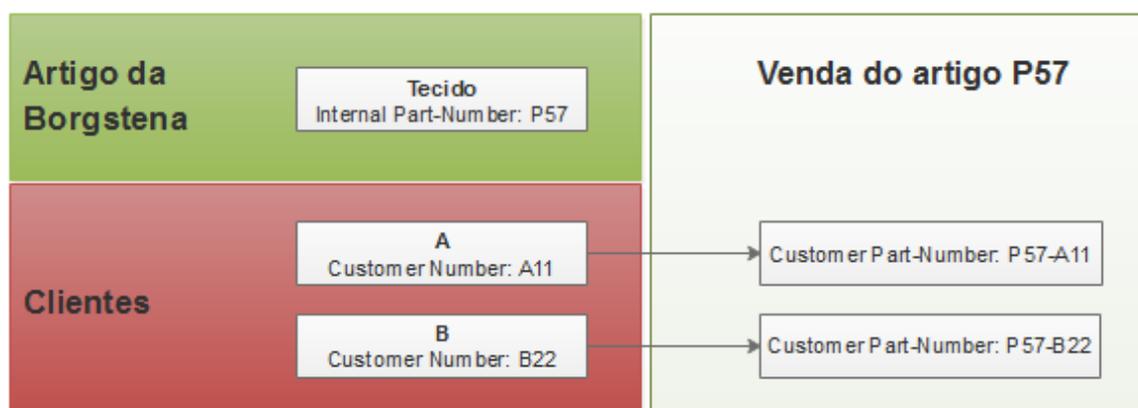


Figura 3.1 – Conceitos de identificação de Part-Number

Como se observa pela figura anterior, a Borgstena tem à venda um **Tecido** cujo **part-number** interno é o **P57**. A empresa tem também dois clientes que são o **A** e o **B** identificados pelo **customer number A11** e **B22** respetivamente. Ao vender o artigo à empresa A e B, o **customer part-number** será **P57-A11** e **P57-B22** respetivamente.

É de se notar que no IFS existe uma tabela que é capaz de cruzar estes dados, permitindo assim controlar e interligar os artigos da Borgstena com os seus clientes.

Tal como mencionado anteriormente, todo este processo é efetuado através de ficheiros Excel onde são criados “mapas” com os campos que são necessários e os responsáveis preenchem os campos. Um dos grandes problemas que se podem registar desta ferramenta é o facto de não se poder ter dois utilizadores com o mesmo ficheiro aberto. Ou seja, caso o utilizador A queira inserir uma reclamação, ele abre o ficheiro. Enquanto este está a preencher os dados, o utilizador B abre o ficheiro e insere os dados de outra reclamação. De seguida o utilizador A salva e fecha o ficheiro, seguindo-se do utilizador B. Os dados do primeiro serão todos perdidos. Isto é um grande problema que também se pretende colmatar com o software a desenvolver.

É importante referir que para efetuar o preenchimento de alguns campos é necessário proceder a algumas tarefas que podem demorar algum tempo pois são efetuadas de forma manual, que são:

- Procurar fornecedor no IFS e escrever os dados
- Procurar pelo Part number a reclamar e escrever os dados
 - Esta procura inclui procurar descrição, preço unitário, unidade, tecnologia, nível, entre outros
- Procurar pelo defeito numa tabela com centenas de dados
- Realizar o cálculo de custos de forma manual
- Criação de ficheiros de Pedidos de Autorização de Notas de crédito
- Deslocar-se pessoalmente a cada responsável para validação de documentos
- Após validação, deslocar-se ao financeiro com as aprovações para este passar nota de crédito

Estas tarefas que foram descritas anteriormente podem ser visualizadas no fluxo a seguir apresentado que, demonstra o processo de inserção de reclamação:

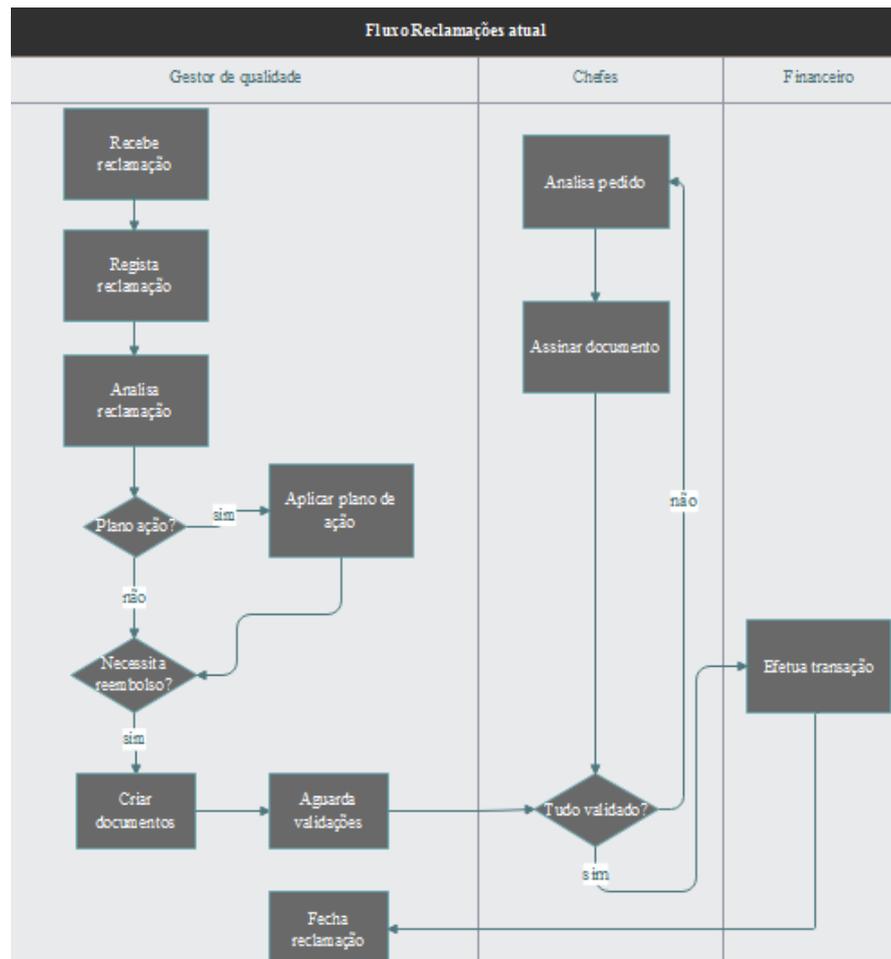


Figura 3.2 - Fluxo de reclamação sem portal

Como é possível observar, o gestor de qualidade para além de ter a dificuldade de inserir dados no Excel de forma manual ainda terá de imprimir e deslocar-se aos devidos responsáveis para a aprovação da documentação ser efetuada e só depois realizar o pedido ao departamento financeiro para o reembolso final ao cliente. Pretende-se que todo este processo seja agilizado de forma a que o gestor possa registar a reclamação e realizar todos os procedimentos necessários de forma automática e cómoda. Posto isto, o fluxo que atualmente é pretendido é o seguinte:

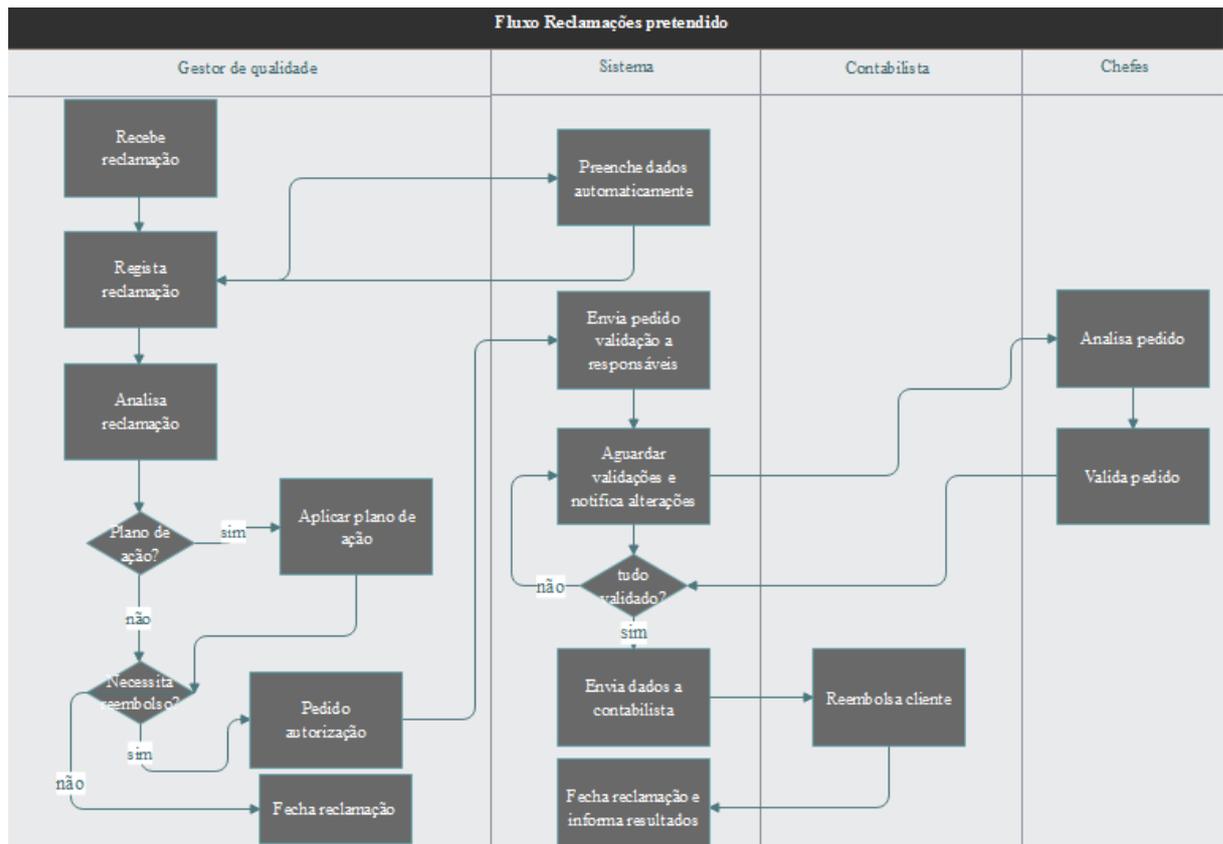


Figura 3.3 - Fluxo de reclamação com portal

Posto isto, é possível concluir que o sistema irá automatizar algumas tarefas do utilizador, permitindo ao mesmo agilizar e otimizar o processo.

3.1.2 Gestão de planos de ação

O *Eight Disciplines Problem Solving (8D problem solving)* [8, 9] ou plano de ações é um método desenvolvido pela *ford* com o objetivo de resolver problemas. É bastante aplicado no que toca ao controlo de qualidade de empresas. Tipicamente é aplicado por profissionais e o seu propósito é o de identificar, corrigir e eliminar problemas recorrentes. Numa fase inicial este processo era composto, tal como o nome indica, por 8 disciplinas. Mais tarde, a *ford* verificou a necessidade de implementação de uma nova disciplina (D0) para que o processo ficasse mais completo. Com isto, este processo é composto pelos seguintes pontos:

- D0 – Plano: Plano para resolver o problema e determinar os pré-requisitos;
- D1 – Equipa: Formar uma equipa com conhecimento do processo;
- D2 – Descrição do problema: Especificar e quantificar o problema, responder a questões do género “Quando?”, “o que?”, “porque?”;
- D3 – Plano de contenção: Definir e implementar acções de contenção;
- D4 – Determinar e verificar as causas: Identificar as causas que expliquem o porque do problema ter acontecido;
- D5 – Verificar correção: Verificar se a correção para o problema vai resultar;
- D6 – Definir e implementar ações corretivas: Definir e implementar as melhores acções corretivas;
- D7 – Evitar problemas: Modificar os sistemas para que num futuro o problema não volte a ocorrer;

- D8 – Felicitar a equipa: Reconhecer o esforço da equipa;

É de se notar que são 9 os pontos mencionados desta disciplina. Isto porque o ponto D0 é o mais recente e foi acrescentado mais tarde, com o intuito de completar este processo.

A Borgstena utiliza o 8D PS com as 8+1 disciplinas acima referidas mas existem certas circunstâncias em que tal pode não acontecer. Cada empresa tem o seu formulário e existem alguns clientes que preferem utilizar os seus do que os da Borgstena. Contudo, os gestores de qualidade que são responsáveis pela gestão de todo este processo, para além de preencher o formulário do cliente, devem preencher também no portal com o formato da Borgstena. Isto porque o pretendido é colocar a informação centralizada e seria de extrema dificuldade ter um sistema que estivesse preparado para analisar os dados de diferentes formulários.

Para proceder ao preenchimento destes documentos, é realizada uma reunião em que o gestor de qualidade se reúne com o responsável da secção do produto reclamado e concluem a fonte do problema e um plano de ações para combater o mesmo (inclui estratégias para corrigir o problema).

Todo este processo é seguido dos famosos “fishbones” [10] ou, “Ishikawa diagrams” fazendo referência ao seu criador. Este tem como objectivo mostrar as causas de um evento específico, colocando-as de forma hierárquica permitindo aos gestores de qualidade encontrarem soluções para eliminar a causa e não o sintoma do problema.

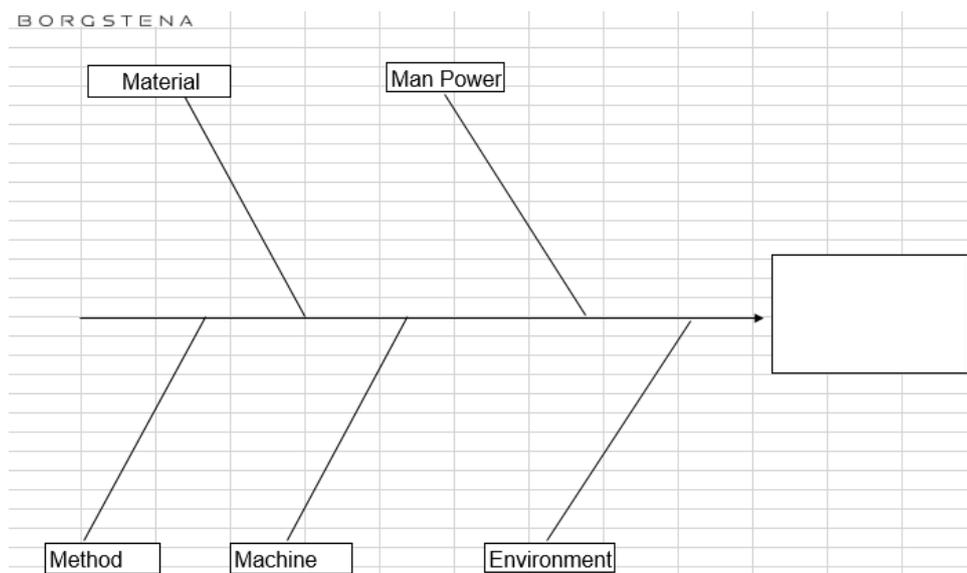


Figura 3.4 - Fishbone

Aplicar este processo aos planos de ações permite resolver muitos dos problemas encontrados. É de se notar que o tamanho do “fishbone” pode aumentar ou diminuir o seu tamanho de acordo com o problema que se pretende resolver. Este diagrama deve estar presente aquando a geração do plano de ação para o formato excel.

3.1.3 Gestão de ordens de compra

O estudo e análise do processo de gestão de compras é parte fundamental para obter dados acerca do software pretendido. Atualmente, este processo tem como base a troca de mails entre o responsável de compras e o fornecedor.

Posto isto, importa ainda clarificar os seguintes conceitos:

- **Ordem de compra:** uma encomenda é criada com recurso a uma ordem de compra. Ela, é um simples ficheiro pdf que contém um cabeçalho referente a dados de encomenda e, contém uma lista, item a item de tudo o que é necessário comprar ao fornecedor destinatário. É de se destacar que cada item pode ter data de entregas diferentes. Esta data pode ser a desejada pela empresa ou, atualizada para a data em que o fornecedor pretende entregar, sendo que prevalece sempre a pedida pela empresa;
- **Confirmação de ordem de compra:** o fornecedor tem a responsabilidade de aceitar o pedido de ordem de compra. Com isto, é necessário que o mesmo confirme item a item a entrega, isto é, indique se é possível entregar o item nos prazos e quantidades previstas. Caso não seja possível, o fornecedor deverá indicar o motivo e a empresa poderá atualizar ou enviar uma nova ordem de compra daquele item a outro fornecedor;
- **Industrial and Financial System (IFS) [11]:** é a ERP (Enterprise Resource Planing) da empresa. É uma base de dados industrial que permite às empresas inserir todo o tipo de dados dentro da mesma e acede-la através de *queries*.

Todos os conceitos acima mencionados fazem parte de um fluxo que é seguido atualmente para a gestão de encomendas (ou ordem de compra) que pode ser consultado na imagem seguinte:

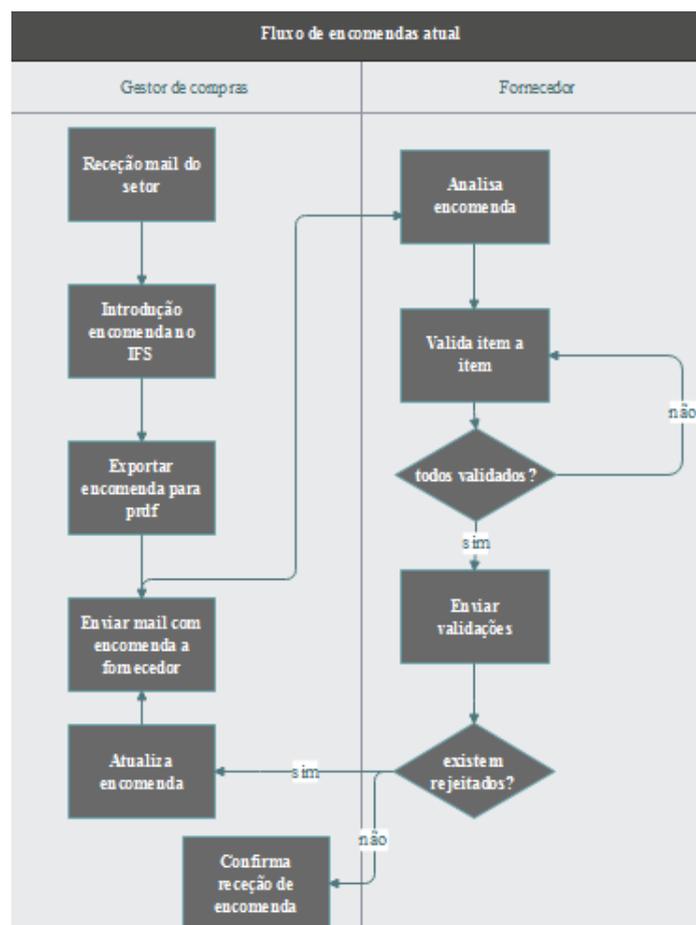


Figura 3.5 - Fluxo de encomendas sem portal

Como é possível observar, o responsável de compras cumpre um elevado número de tarefas sendo que estas poderiam ser facilitadas para o mesmo. Por exemplo, a gestão de

mails pode tornar-se confusa e desorganizada. O gestor é responsável pelo tratamento de várias encomendas a vários fornecedores, podendo estar a negociar diversas ordens de compra com o mesmo fornecedor. Importa ainda clarificar que a consulta de contactos dos fornecedores se torna uma tarefa complicada ao utilizar o próprio IFS. Quando a ordem de compra é gerada e exportada para excel, o gestor necessita de procurar o endereço de destinatário, anexar o pdf, escrever o conteúdo e só depois proceder ao envio do pedido. Estas tarefas têm como objetivo ser reduzidas por um sistema que permita a gestão de todo o processo por norma a facilitar o trabalho dos gestores.

Posto isto, o fluxo pretendido para a situação descrita, será o utilizador inserir ou atualizar apenas as encomendas no IFS sendo que as restantes tarefas são da responsabilidade do portal, tal como podemos observar na imagem a seguir:

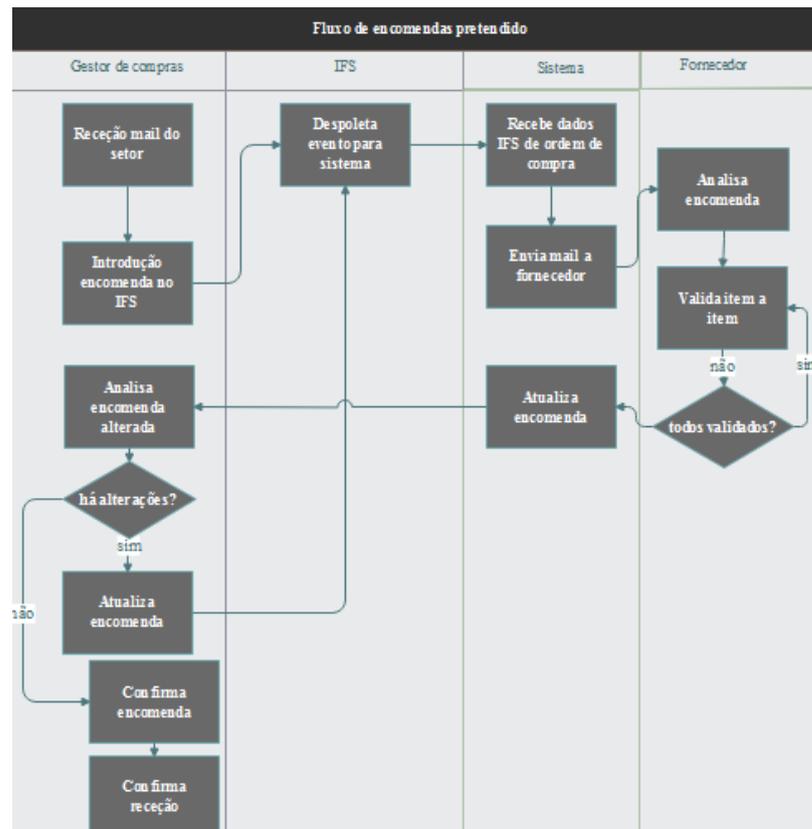


Figura 3.6 - Fluxo de encomendas com portal

Como podemos observar, o fluxo será agilizado de forma a que o gestor de compras tenha menos trabalho a realizar e desta maneira estes processos sejam organizados de forma mais correcta e registados num local adequado. Toda a informação de criação de mails e troca de informação é efetuada e criada pelo portal. É também pretendido que caso o utilizador necessite, possa inserir também uma mensagem que deverá ser enviada em conjunto com o conteúdo que o próprio portal gera.

Permite também a interação do fornecedor no portal evitando a tradicional troca de emails. Este valida os itens diretamente no portal.

Posto isto, importa ainda clarificar que com estas funcionalidades a informação encontra-se centralizada e organizada num sistema de base de dados e desta forma é também possível a obtenção de análises estatísticas sobre o fornecedor e outros dados relevantes.

3.2 Análise de concorrência

A análise de concorrência é uma análise que permite obter conhecimento sobre ferramentas concorrentes ao software a desenvolver. O conhecimento e utilização das mesmas é fundamental para entender aspetos chave que poderão ajudar no desenvolvimento do portal e trarão dados relevantes sobre a forma e processos que estão envolvidos por trás do sistema, sem que o utilizador principal se aperceba.

Posto isto, é fundamental que se entendam os seguintes conceitos [12 – 15]:

- **Customer Relationship Management (CRM):** ferramenta que realiza os processos de gestão de clientes.
- **Supplier Relationship Management (SRM):** ferramenta que realiza os processos de gestão de fornecedores.

Com isto, importa clarificar que o estudo deste tipo de ferramentas ajudou a levantar dados relevantes e, a entender melhor o conceito de gestão por detrás do software. Algumas ferramentas que se seguem foram indicadas pelos experientes profissionais da Borgstena por serem um bom exemplo dos requisitos que são pretendidos para o portal.

3.2.1 CRM

O principal objetivo das CRM é colocar o cliente como o principal foco dos processos de negócio para entender e antecipar as necessidades dos mesmos. Com uma ferramenta destas, as ações da empresa ficam voltadas para o cliente e aquilo que ele realmente precisa. Este tipo de software abunda no mercado, com licenças que se podem tornar um pouco excessivas em termos monetários, para empresas que comecem a crescer e onde os utilizadores do programa comecem a ser cada vez mais (sendo este último o caso da Borgstena).

Este tipo de software foca-se principalmente em seguir três tipos de abordagens:

- Filosófica: guiar o relacionamento da empresa com o *stakeholder*;
- Visão estratégica: ações e planeamento estratégico da empresa;
- Tecnologia da informação: prever condições estruturais, planeamento e iteração e valores.

As ferramentas existentes no mercado realizam um conjunto enorme de tarefas, desde a gestão de contratos, procedimentos para criação de questionários de satisfação dos clientes, análises estatísticas, entre outros.

De seguida, vai ser descrito o estudo efetuado pelo estagiário sobre ferramentas utilizadas no mundo empresarial.

3.2.1.1 Salesforce

O salesforce [16] é um portal que permite a gestão de clientes, ou seja, é uma ferramenta do género CRM. Como podemos observar na imagem a seguir apresentada, este apresenta um design visualmente atrativo e como os demais programas existentes no mercado, as tarefas ficam mais fáceis à medida que se trabalha com o mesmo.

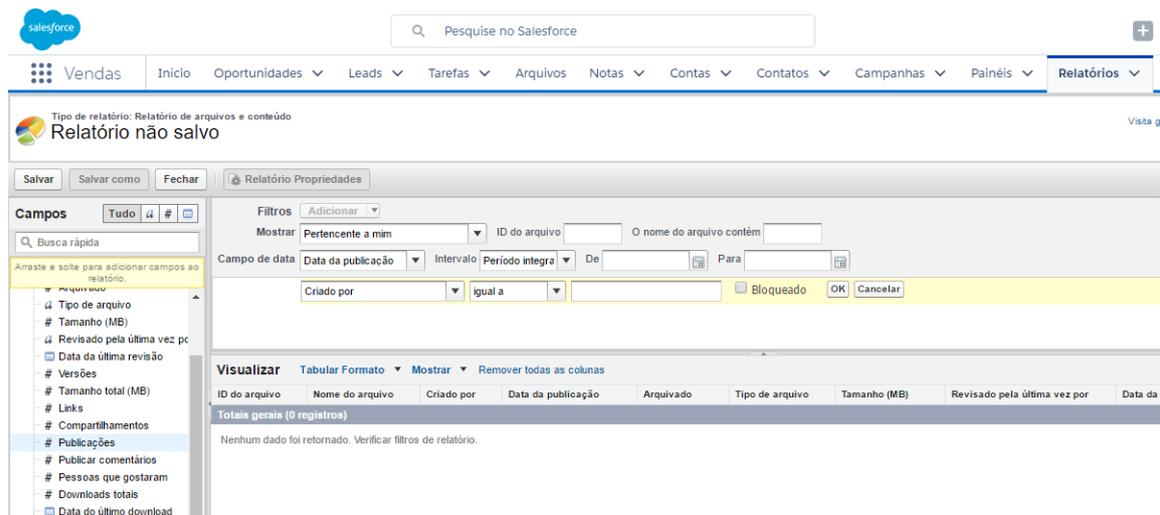


Figura 3.7 - Página salesforce

O salesforce é um software que tem como principal foco as vendas e evolução das mesmas. Com ele é possível verificar, agendar e fechar contratos, sendo também possível analisar graficamente a evolução de todo este processo ao longo do tempo para obter o status atual da empresa.

Esta plataforma fornece também uma vasta opção de templates de relatórios previamente criados bem como a opção de criação manual de um. Esta última permite estruturar o documento de acordo com o que é pretendido pela empresa para, por exemplo, criar uma reclamação e registar um formulário 8D no formato pretendido.

Para além das funcionalidades mencionadas, existe o salesforce developers [17] que é uma área para programadores, isto é, permite a determinada empresa criar aplicações que fiquem sincronizadas com os dados registados no salesforce. Esta integração pode ser realizada através de API's já disponibilizadas ou através do Heroku [18]. Este último é uma *cloud Platform-as-a-Service* (PaaS) [19] que suporta várias linguagens de programação, permitindo aos programadores uma liberdade na escolha da tecnologia a utilizar.

Uma das grandes desvantagens apontadas e que surgiram a público deste software foi o facto de este ter sofrido interrupções tornando impossível aceder ao serviço.

Como mencionado anteriormente, este software tem como principal foco as vendas e evolução das mesmas, funcionalidade que não é pretendida para a aplicação a desenvolver. Apesar de ser possível criar e gerar relatórios bem como desenvolver aplicações em várias linguagens de programação. Esta aplicação não traria vantagens para o projeto a desenvolver uma vez que a área de atuação do mesmo é diferente da que se pretende abranger, contudo demonstrou funcionalidades que podem ser utilizadas para num futuro serem integradas com o portal no âmbito de expandir o mesmo para outras áreas e objetivos.

3.2.1.2 NABD

A NABD [20] é a antiga Premitek, uma empresa de desenvolvimento de software fundada em 2002 que fornece a nível empresarial diversas ferramentas de gestão dos seus processos de negócio, incluindo software de gestão de reclamações. Este último designado de Complaint Management Software (CMS) [21].

O portal CMS oferece uma vasta gama de recursos ao utilizador e realiza a gestão de reclamações de forma eficaz e organizada. Apesar de a numeração das mesmas não ser

efetuada da forma que é pretendida pela empresa, a restante gestão de informação relativamente a datas, problemas, responsável, entre outros campos, é a pretendida. Por outro lado, este software não realiza a separação de clientes e responsáveis, ou seja, é possível registar reclamações de qualquer cliente, mesmo que o utilizador que efetue o registo não seja o responsável por essa entidade reclamante.

Como podemos observar pela figura abaixo, a gestão de reclamações apresenta-se da seguinte forma:

<input type="checkbox"/>	Case No	Date	Source	Status	Activity type	Target date	SLA	Against OU	Product/Service	Assigned to	Customer Name
This is a sample case subject # 1 for demo purpose											
<input type="checkbox"/>	1	05/18/2016		Open	Corrective				Product A		
This is a sample case subject # 2 for demo purpose											
<input type="checkbox"/>	2	05/18/2016		Open	Assesment						
reclamação da treta											
<input type="checkbox"/>	3	05/18/2016		Open	Assesment						

Figura 3.8 - NABD, gestão de reclamações

O NABD permite também atribuir um conjunto de tarefas que serão aplicadas para corrigir o problema descrito na reclamação (tarefas de um plano de ação). Esta, ficará em estado aberto até que o utilizador responsável indique outra informação.

Possui também uma API que permite integrar com aplicações que a empresa pretenda desenvolver, permitindo assim obter os dados registados no portal NADB com diversos programas criados pela Borgstena.

Um dos grandes problemas encontrados no software é o facto de que não é possível aplicar o plano de ações nos formatos pretendidos (formulários 8D). Uma das alternativas encontradas foi a anexação do formulário à reclamação, o que levaria novamente à utilização dos ficheiros de Excel, processo que se pretende remover.

Este software apresentou-se uma boa ferramenta de gestão de reclamações. Contudo, pretende-se um software que faça todo o processo de gestão e que seja possível integrar o mesmo com a estrutura dos planos de ação da Borgstena.

Apesar de não ser possível utilizar os formulários no formato pretendido, foi pensada uma solução com o objetivo de se utilizarem os recursos da plataforma NABD integrando a mesma com uma aplicação. Este processo iria envolver a API. A solução observada foi:

- Desenvolver uma aplicação de gestão de planos de ação nos formatos pretendidos. O objetivo seria utilizar a API para guardar reclamações no NABD e utilizar a API para integrar esses dados na aplicação desenvolvida;

Apesar de se demonstrar uma solução que pudesse facilitar o trabalho, a empresa decidiu rejeitar a proposta de integração com esta API uma vez que seria necessário proceder a um estudo aprofundado da API. Este teria de ser efetuado não só pelo estagiário, mas também pelos funcionários que iriam dar manutenção no futuro à aplicação.

Posto isto, a implementação de um portal “de raiz” que seguisse o fluxo do processo pretendido pela empresa, foi a solução escolhida.

3.2.1.3 Pipedrive

O Pipedrive [22] foi uma empresa criada pela junção de vários gestores de vendas que, após a sua experiência pessoal no que toca a utilização de CRMs, decidiram unir esforços em 2010 para desenvolver uma solução que atingisse o mercado mundial. Atualmente, contam com uma equipa de 168 elementos, sendo que construíram uma das ferramentas CRM mais utilizadas, dando como nome ao portal o mesmo da empresa.

O Pipedrive é um sistema open source que permite ao utilizador realizar uma gestão eficaz dos seus clientes. Com ele é possível dividir o cliente por vários grupos (cliente habitual, novo, antigo, entre outros), marcação de reuniões, obter análises sobre negócios que estejam a ser ou não lucrativos e, gerir os contratos com as entidades.

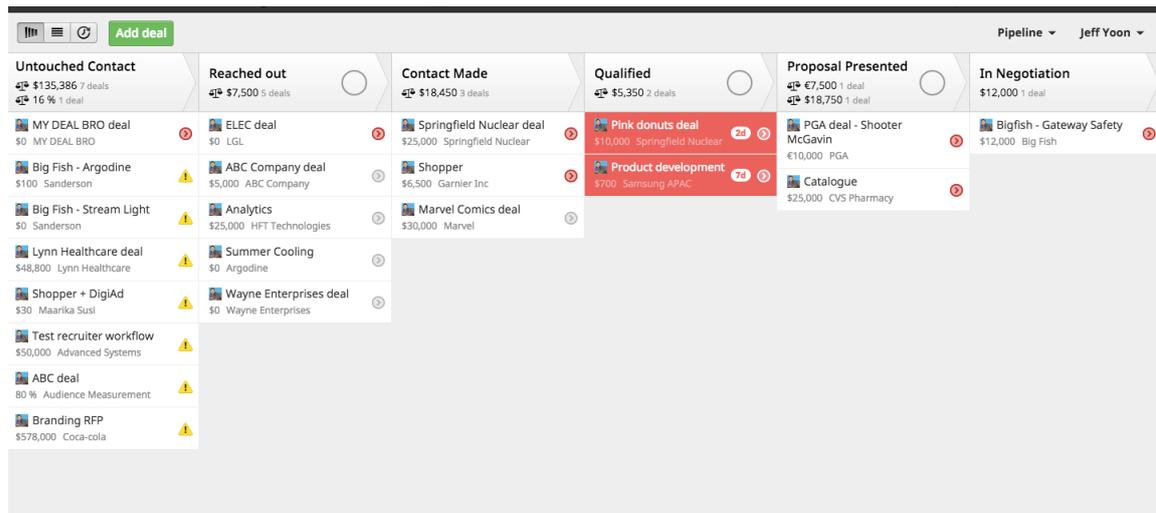


Figura 3.9 - Página Pipedrive

Usufrui de um design minimalista e atrativo, permitindo de igual forma a integração com várias aplicações (Google drive e Google agenda por exemplo). É possível desenvolver aplicações e integrar as mesmas com a API [23] do Pipedrive. Com ela é possível realizar pedidos SSL [24] para que se possa utilizar funções já programadas por esta empresa bem como utilizar o seu código e adaptá-lo.

O Pipedrive é uma ferramenta muito útil e possui funcionalidades que podem ser integradas no portal com o objetivo de facilitar o desenvolvimento do software a desenvolver, nomeadamente na gestão de ordens de compra (Note-se que apesar de ser um software voltado para os clientes, com poucas alterações é possível colocar a informação das ordens de compra dos fornecedores). Através da imagem acima podemos observar que existem uns widgets que possuem títulos (“Untouched Contact”, “Reached Out”, entre outros) que podem ser alterados para o pretendido da gestão de ordens de compra. Contudo, a aplicação não permitiria ao fornecedor conectar-se a uma plataforma para proceder à aceitação dos itens de determinada encomenda, tendo que esta funcionalidade continuar a ser efetuada de forma manual (via mail) ou ser efetuada através de uma integração utilizando a API e uma aplicação que obteria a informação das encomendas e a disponibilizava aos fornecedores. Contudo, a encomenda teria de ser inserida no IFS e este também deveria ser integrado com a aplicação a desenvolver e o Pipedrive. A empresa não viu vantagens neste processo sendo que optou por descartar a utilização das funções desta plataforma.

3.2.1.4 Conclusão

Após o estudo das ferramentas acima mencionadas, torna-se importante sintetizar as mesmas com as vantagens e desvantagens de cada uma. A tabela seguinte demonstra a síntese criada:

	Vantagens	Desvantagens
Salesforce	<ul style="list-style-type: none">• Licenças baratas;• Gestão de relatórios de qualquer tipo;	<ul style="list-style-type: none">• Recursos desnecessários;• Dificuldade de aplicação de estratégias de IT;• Disponibilidade;
NABD	<ul style="list-style-type: none">• Gestão de reclamações;• Integração com aplicações;	<ul style="list-style-type: none">• Não existe gestão de planos de ações;• Licenças caras;• Fluxo estratégico diferente do pretendido;
Pipedrive	<ul style="list-style-type: none">• Open source;• Integração com aplicações;	<ul style="list-style-type: none">• Funcionalidades open source não se adequam ao pretendido pela empresa;

Tabela 3.1: Comparação das CRM estudadas

Como podemos observar pela tabela 3.1, as soluções estudadas possuem vantagens e desvantagens de peso que foram necessárias ser consideradas e analisadas. A grande parte do motivo de descarte total das aplicações mencionadas deve-se ao facto de que é extremamente difícil adaptar as mesmas às necessidades da empresa. A exceção encontra-se no Pipedrive pois é uma ferramenta que possui funcionalidades que poderão ser integradas no futuro embora para a solução atualmente pretendida, estas não apresentem vantagens.

3.2.2 SRM

As ferramentas SRM colocam como foco principal os fornecedores. Têm como principal objetivo gerir toda a informação relacionada com esta entidade e reduzir custos através da otimização de processos de compra. Com uma ferramenta destas é possível a empresa estudar da melhor forma o fornecedor e saber qual o melhor para comprar o(s) produto(s) desejado(s). No mercado, existem ferramentas bastante úteis para as empresas. À semelhança das CRM, estas exigem que se pague uma licença para a utilização das mesmas.

A gestão de relacionamento entre o fornecedor e o vendedor representa uma oportunidade para melhorar as operações entre estes dois, na medida em que implementa melhorias nas práticas de trabalho de forma a alcançar um sucesso mútuo.

Nesta secção encontra-se descrito o estudo realizado a ferramentas de gestão de fornecedores.

3.2.2.1 Gatewit – Gestão de fornecedores

A gatewit [25] é uma empresa que se especializou no desenvolvimento de software para áreas comerciais, mais concretamente na área de gestão. As soluções são 100% portuguesas e têm como objetivo a redução dos custos e aumento do controlo sobre despesas

da empresa. No contexto deste estágio e nesta secção, está descrito o software de Gestão de Fornecedores no meio de outras soluções disponíveis.

Esta plataforma SRM é uma ferramenta que permite às entidades compradoras, públicas ou privadas, a automação e gestão da informação do desempenho dos fornecedores.

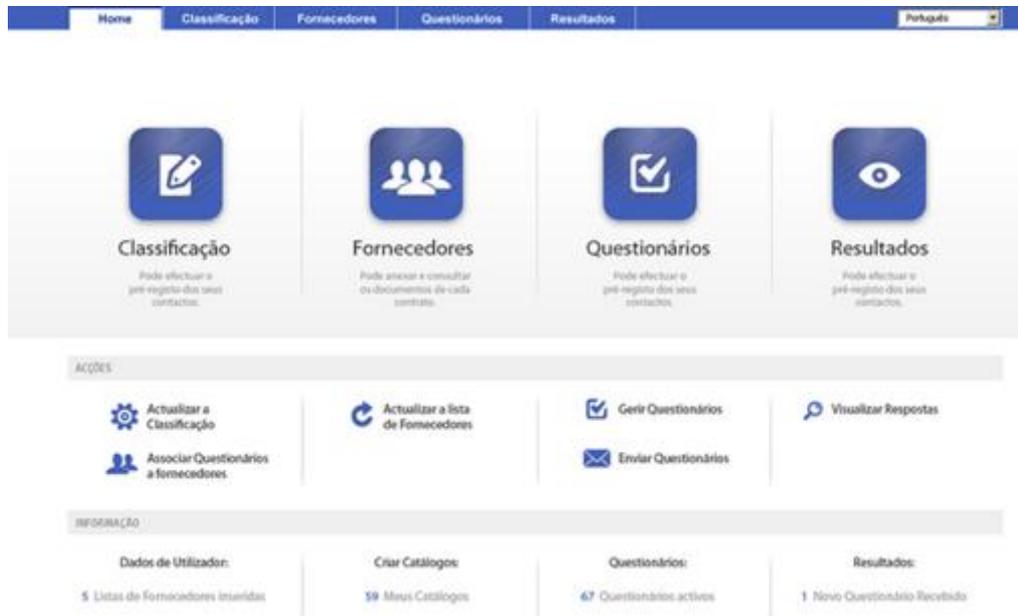


Figura 3.10: Homepage do Gatewit

Permite adicionar fornecedores para analisar e gerir através da importação de um ficheiro excell, criar e enviar questionários para as empresas fornecedoras responderem e permite consultar os resultados de uma pesquisa de comparação entre fornecedores.

Este software desenvolvido pela gateway, usufrui de um design minimalista e atractivo. Contudo, a grande limitação é o facto de que utiliza um número de recursos demasiado baixos uma vez que a empresa divide o seu software em vários, decisão essa que trás as seguintes desvantagens:

- Utilização de *software* diferente para acções diferentes, leva:
 - à desmotivação do utilizador
 - curva de aprendizagem cada vez maior
- Pagamento de várias licenças, leva a custos excessivos
- Esta empresa assume posições controversas em relação aos seus supervisores (muitos problemas com auditorias)

Devido a estas desvantagens, a empresa decidiu desenvolver um portal em que pudesse atender às suas necessidades numa plataforma apenas. Assim, os utilizadores não terão que utilizar diversos *softwares*.

Importa ainda clarificar que esta empresa foi suspensa em Dezembro de 2016 uma vez que não cumpria a lei, fazendo cobranças indevidas aos operadores económicos. Este resultado surge após a empresa ter falhado uma primeira auditoria em Março do mesmo ano e uma segunda em Dezembro. Dentro do intervalo de tempo da primeira à segunda auditoria,

a Gatewit deveria ter regularizado as “não conformidades” detetadas. Algo que não se verificou.³

3.2.2.2 Gatewit – Auctions

Como já tinha sido mencionado, a Gatewit separa o seu software por vários (um deles é o Auctions [26]). Existe uma vertente que permite aos fornecedores e clientes realizarem um leilão para que este último compre ao que licitar mais barato (valor mais baixo). Isto é, o cliente cria uma encomenda e seleciona os fornecedores que pretende envolver na compra. Estes, irão cotar (dar um preço) ao cliente. De seguida o software irá agendar um leilão que irá começar com os fornecedores selecionados e, com o valor mais baixo cotado (preço mais baixo).

A forma de licitação será sempre para colocar o preço menor, nunca para aumentar e, os envolvidos irão “lutar” para vender o produto. No final do leilão, o fornecedor que vence é aquele cujo o valor licitado é o mais reduzido. Assim, permite ao cliente comprar sempre de forma mais barata.

A grande vantagem da criação destes leilões é que o utilizador pode definir as suas próprias regras, não tendo que se restringir a valores por defeito.

Este software, à semelhança do anterior, possui um design atrativo e, as suas funcionalidades são fáceis de entender. Sem grande esforço, o utilizador consegue realizar as ações necessárias no que toca a atingir o resultado que pretende.

Consulte a classificação dos concorrentes participantes do leilão terminado

Na seguinte tabela poderá visualizar a classificação dos concorrentes e a melhor proposta de cada um deles.

Nome do Leilão:	Gás	Poupança Global:	0,0000 €
Descrição:	Fornecimento anual de Gás à BTP	Poupança Global(%):	0,00 %
Valor de Abertura:	27.540,00000 €	Nº de licitações aceite:	0

Resultados do Leilão				
Classificação	Concorrente	Nº de Licitações	Licitação	Adjudicação
-	-	0	(Sem Licitações)	
-	-	0	(Sem Licitações)	
-	-	0	(Sem Licitações)	
-	-	0	(Sem Licitações)	
-	-	0	(Sem Licitações)	
-	-	0	(Sem Licitações)	

VOLTAR EXPORTAR CANCELAR ADJUDICAÇÃO ADJUDICAR FINALIZAR

Figura 3.11 – Gatewit Auctions

As funcionalidades deste portal foram importantes de se observar uma vez que está previsto a integração de um módulo deste género que se integre ao módulo de gestão de ordens de compra. Perceber como desenvolver este último módulo de forma a que seja mais fácil adaptá-lo no futuro a este tipo de funções foi uma das tarefas incluídas neste estudo.

Importa ainda clarificar que esta empresa foi suspensa em Dezembro de 2016 uma vez que não cumpria a lei, fazendo cobranças indevidas aos operadores económicos. Este resultado surge após a empresa ter falhado uma primeira auditoria em Março do mesmo ano e uma segunda em Dezembro. Dentro do intervalo de tempo da primeira à segunda auditoria,

³ <http://www.jornaldenegocios.pt/empresas/tecnologias/detalhe/plataforma-de-compras-publicas-gatewit-foi-suspensa>

a Gatewit deveria ter regularizado as “desconformidades” detetadas. Algo que não se verificou.⁴

3.2.2.3 Gatekeeper

A Gatekeeper [27], fundada em 1992, é uma empresa de desenvolvimento de software e de consultoria que atua na área de gestão, tendo com objetivo reduzir os custos e aumentar a produtividade de uma empresa. O software desta empresa foi estudado apenas com recurso a uma licença trial de 30 dias. O grande objetivo deste software é providenciar ao utilizador uma forma rápida e simples de gerir os fornecedores e os seus contratos com os mesmos. Utiliza áreas de ecrã que fornecem feedback estatístico rápido e fácil de analisar pois baseiam as suas análises em gráficos de rápida análise. Esta é uma solução SaaS (Software as a Service).



Figura 3.12 – Homepage Gatekeeper

É um software que faz uso da simplicidade. Apesar de ser um tipo de programa mais direcionado para a gestão de contratos, permite também a criação de relatórios onde podem ser colocadas as reclamações que determinada empresa faz a um fornecedor e, os contratos realizados podem ser as ordens de compra pedidas ao mesmo. Com todos estes dados, é possível obter dados estatísticos fundamentais, como por exemplo, qual o melhor fornecedor que atua em determinado setor.

Este software dispõe também de uma API RESTful⁵ que permite realizar a integração com outras aplicações que a empresa queira desenvolver ou que já existam. Com ela é possível realizar a troca de dados entre o Gatekeeper e outra aplicação.

Apesar de possuir boas funcionalidades, estas não se adequam ao que a Borgstena necessita de momento para o portal que pretende desenvolver. O Gatekeeper mostrou-se uma boa ferramenta de gestão de fornecedores mas, no que toca ao objetivo pretendido, este não se

⁵ <https://pt.wikipedia.org/wiki/REST>

adequa. Contudo, poderá ser tida em conta no futuro, a utilização dos recursos deste software realizando a sua integração com outros módulos que poderão ser adicionados, ao portal.

3.2.2.4 Conclusão

Após o estudo das ferramentas acima mencionadas, torna-se importante sintetizar as mesmas com as vantagens e desvantagens de cada uma. A tabela seguinte demonstra a síntese criada:

	Vantagens	Desvantagens
Gatewit – Gestão de fornecedores	<ul style="list-style-type: none"> Boa gestão de fornecedores e compras 	<ul style="list-style-type: none"> Posições controversas face a superiores Vários módulos levariam a várias licenças Recentemente a empresa foi suspensa
Gatewit auctions	<ul style="list-style-type: none"> Fluxo de processo de leilões 	<ul style="list-style-type: none"> Não é o software que é pretendido desenvolver para o portal Recentemente a empresa foi suspensa
Gatekeeper	<ul style="list-style-type: none"> Boa gestão de processos de vendas 	<ul style="list-style-type: none"> Não se adequa às funcionalidades requeridas

Tabela 3.2 – Conclusão das SRM

Posto isto, é possível observar que o software da Gatewit, relativamente à gestão de fornecedores, é uma boa ferramenta que poderia ser utilizada como alternativa à implementação de uma aplicação. Infelizmente esta não oferece a segurança necessária à empresa (devido às controversias face a auditorias a que sofreu e recente suspensão do mercado), foi decidido que a solução mais viável seria a implementação do módulo de raiz. Desta forma os fluxos seriam implementados de acordo com o pretendido pela Borgstena e para tarefas de manutenção não seria necessário estudar as funcionalidades de alguma API ou depender de outro portal.

Importa ainda clarificar o leitor que estas decisões foram tomadas pelo estagiário e consequentemente apresentadas e aprovadas pela empresa.

3.3 Análise interna

A análise interna é o tipo de análise efetuada dentro da empresa. Ela tem como objetivo dar a conhecer que ferramentas estão disponíveis para desenvolver a aplicação e que tipo de utilizadores serão o “público alvo”.

Nesta secção o leitor irá encontrar a abordagem aos seguintes tópicos:

- **Industrial Financial System (IFS):** o ERP da empresa. Neste tópico é abordado o assunto sobre que informação é necessária desta plataforma, bem como serão realizados alguns dos acessos;
- **Supplier Evaluation (SEval):** aplicação que permite realizar e contém a avaliação dos fornecedores. Estudar esta aplicação tornou-se essencial, uma vez que um dos objetivos do portal será integrar os resultados do SEval com o software a desenvolver, isto no intuito de dar a conhecer ao fornecedor como está avaliado pela Borgstena;
- **Borgstena Login Application (BLA):** aplicação que permite realizar login apenas uma vez e disponibiliza as restantes aplicações a que determinado utilizador autenticado tem acesso;
- **Utilizadores finais:** análise de utilizadores finais e métodos para melhorar o desenvolvimento do portal;

A análise dos processos acima mencionados foi essencial para que o estagiário pudesse compreender as ferramentas existentes e como poderiam ser aplicadas ao projeto, bem como o estudo aos utilizadores finais no intuito de facilitar no design e disponibilização do conteúdo aos mesmos.

3.3.1 Industrial Financial System (IFS)

O IFS (Industrial Financial System) é o sistema ERP da empresa e permite inserir qualquer tipo de informação para dentro da base de dados do mesmo. Este sistema é dos mais antigos da empresa, sendo que é usado praticamente desde os primeiros dias da mesma. Este software interage diretamente com a base de dados e a sua funcionalidade é muito “primitiva”, uma vez que em termos de design gráfico não cumpre quaisquer normas das atuais, ou seja, não usufrui de um aspeto agradável à vista humana.

Posto isto, importa clarificar o leitor que apenas alguns dados são consultados pela aplicação à ERP, sendo eles:

- Lista de clientes;
- Lista de fornecedores;
- Faturas (clientes);
- Ordens de compra (fornecedores).

Importa ainda referir que o último ponto não é acedido diretamente pelo portal, ou seja, a aplicação não acede de forma direta ao IFS. Sempre que for inserida uma nova ordem de compra no IFS, este é responsável por gerar um evento que contém a informação necessária para atualizar a base de dados do portal [ver arquitetura, Capítulo 5, Secção 5.2].

Com esta restrição, é possível reduzir os acessos à base de dados do IFS, melhorando assim a sua performance uma vez que é acedido centenas de vezes por dia.

Como foi a primeira vez que a empresa se viu em um cenário onde não pretendia que o servidor comunicasse diretamente com o IFS, existiu a necessidade de, para além de estudar o sistema, se implementasse um simples mecanismo para testar se seria possível criar eventos do lado do IFS para enviar dados para o servidor. Com isto, verificou-se que é possível sempre que ocorre um determinado evento, correr um script em php e obter os resultados referentes às ordens de compra (por exemplo).

3.3.2 Supplier Evaluation (SEval)

O SEval (Supplier Evaluation) é um sistema desenvolvido pela equipa de desenvolvimento de software da Borgstena e, com o intuito de realizar as avaliações a fornecedores. Este software contém uma vasta gama de parâmetros necessários para a avaliação, mas, ao mesmo tempo é de fácil utilização. Uma vez que, de hábitos antigos, na empresa se utilizava muito ficheiros Excel para realizar as variadas operações necessárias, este programa utiliza um template que em muito vai ao encontro de folhas de Excel. Por exemplo, a apresentação e edição de dados, funciona com “*datatables*”⁶.

De entre uma vasta gama de parametrização para avaliação de fornecedores, é necessário dividir a mesma nos seguintes setores:

- **Compras** – Avaliada semestralmente pelos responsáveis de compras a fornecedores
- **Logística** – Cada fornecedor é avaliado por um elemento da logística, semestral e mensalmente;
- **Engenharia** – Cada fornecedor pode ser avaliado por um ou mais funcionários, semestralmente;
- **Avaliação de qualidade** – Compara e interpreta a quantidade de material recebido e reclamado gerando uma avaliação de qualidade de fornecedor mensal e semestralmente.

Este software demonstra ser uma ferramenta muito útil para a empresa e, encontra-se já em fase de desenvolvimento uma segunda versão do produto!

Importa referir que todo este software foi desenvolvido com recurso a php, html e javascript!

3.3.3 Borgstena Login Application (BLA)

O BLA (Borgstena Login Application) é uma aplicação que permite aos utilizadores realizar o processo de autenticação apenas uma vez. Isto é, uma vez que existem dezenas de aplicações criadas e existem utilizadores que têm acesso a mais que uma, verificou-se a necessidade de implementar um software que permitisse a autenticação do utilizador apenas uma vez e que assim fosse disponibilizada uma lista com as restantes aplicações ao qual este funcionário tem acesso.

Com isto, é possível gerir as aplicações bem como os utilizadores que acedem às mesmas, através de um simples sistema de login e um menu com a listagem do software a que tem acesso. Como exemplo, podemos observar na imagem seguinte o processo de autenticação, onde à esquerda está o menu com as aplicações a que o utilizador autenticado tem permissões de utilizar.

⁶ <https://datatables.net/>

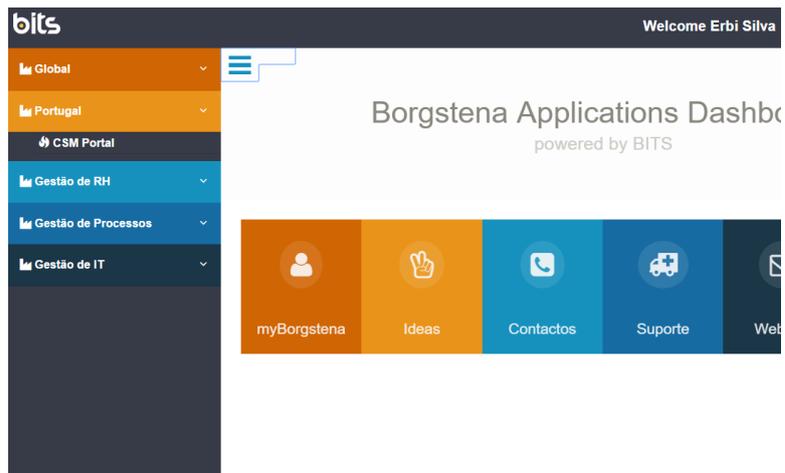


Figura 3.13 – Borgstena BLA

Ao escolher a aplicação e por ser uma vista dinâmica, o utilizador poderá “saltar” de aplicação em aplicação. Podemos ver um exemplo na imagem em baixo, onde o utilizador entrou na aplicação CSMPortal.



Figura 3.14 – Borgstena BLA integrado com o Portal

O conteúdo foi disponibilizado “dentro” da aplicação BLA, sendo que esta última está sempre acessível.

3.3.4 Utilizadores finais

Com a necessidade de criar um portal para diferentes tipos de utilizador e após um breve conhecimento fornecido por parte da restante equipa de desenvolvimento, verificou-se a necessidade de se realizar um estudo dos diferentes utilizadores que poderão utilizar o portal e seus diferentes módulos, pois existe o conhecimento de que os utilizadores finais de aplicações desenvolvidas por parte da equipa, nem sempre utilizam os programas que normalmente se destinam a facilitar as tarefas dos trabalhadores. Isto porque existe um forte hábito dos mesmos no que toca à utilização de sistemas antigos e, por vezes, muitos não sentem a obrigação ou a necessidade de transitar para algo novo e preferem utilizar aquele em que mais estão confortáveis.

Apesar de a ideia do desenvolvimento deste portal ter partido dos chefes pelos setores dos clientes e fornecedores, foi importante realizar o levantamento de breves características de quem vai utilizar o software pois não serão só os chefes, mas sim os diversos trabalhadores das secções a usufruir deste programa. É de se notar que este estudo foi realizado a pedido da empresa em questão, no âmbito de entender o ambiente onde se encaixa o software a desenvolver.

Posto isto, foi realizado um breve levantamento de características dos utilizadores finais da aplicação. Este estudo foi realizado com base em reuniões individuais e, dividiu-se em diversos pontos, que serviram para resumir um pouco o tipo de cliente com o qual se irá lidar. Esses pontos podem dividir-se em:

- Demografia e género;
- Utilização de tecnologia no dia-a-dia fora do trabalho (redes sociais, jogos, entre outros);
- Problema do fluxo atual de trabalho;
- Opinião sobre a implementação do portal;
- Histórico de aceitação a tecnologias;

Posto isto foi possível concluir e resumir o seguinte:

Demografia	Resumo
Faixa etária: >25 e <45 Género: Homens e mulheres	Demonstram grande motivação no que toca à ideia de existir um portal para facilitar as tarefas; São utilizadores que não demonstraram problemas e aceitaram bem a utilização de outros softwares criados e implementados pela equipa de desenvolvimento já existente; Utilizadores abertos a novas tecnologias e, que apresentam características de uma fácil adaptação às mesmas

Tabela 3.3 - Análise geral de utilizadores

Como podemos observar pela tabela anterior, apesar de, em termos de dados demográficos, os utilizadores se encontrarem num intervalo relativamente grande e, os géneros serem diferentes, estes são utilizadores que não demonstram problemas no que toca à utilização do portal. Como referido anteriormente, o atual fluxo de trabalho é demoroso e complexo e, uma das afirmações dos utilizadores reunidos foi:

“É preferível perder tempo a aprender a utilizar o portal, pois certamente não levará tanto tempo como o processo atual leva.”

Posto isto, podemos concluir que, é necessário desenvolver uma aplicação que apresente uma curva de aprendizagem o mais reduzido possível, mas principalmente que apresente formas de automatizar todas as tarefas que atualmente são realizadas manualmente. Estas tarefas, que em termos de programação à partida são fáceis de implementar (como exemplo, uma *dropdown* com o nome do cliente, mas que devolve o seu código) mas, que de forma manual exigem uma carga temporal excessivamente grande (como exemplo, procurar numa lista relativamente extensa, o código do cliente com determinado nome).

Apesar de ser uma ferramenta que poderá ser vista como um bom recurso pelos seus utilizadores finais, após este estudo é fundamental compará-lo com a experiência da atual equipa de desenvolvimento. Após a comparação dos resultados obtidos, concluiu-se que são utilizadores que facilmente tendem a mudar os requisitos do produto, ao contrário dos membros de outras secções de trabalho.

O estudo a estes utilizadores tornou-se de igual forma importante pois permitiu, já numa fase mais avançada do projeto, a apresentação de protótipos iterativos criados com

recurso a uma ferramenta chamada InvisionApp [28]. Com eles foi possível registrar alterações nas interfaces apresentadas.

Posto isto, importa ainda clarificar que foram seguidos os princípios de Jakob Nielsen [29]. Este apresenta 10 heurísticas gerais para o design das aplicações. São assim chamadas porque são regras gerais e não são orientações de usabilidade específicas. A lista de 10 heurísticas de Nielsen apresenta-se nos pontos seguintes:

1. **Visibilidade do estado do sistema:** O sistema deve informar sempre o utilizador do que está a acontecer e todas as ações precisam de feedback instantâneo
2. **Relacionar o sistema com o mundo real:** O sistema deve utilizar a linguagem de utilizador, ou seja, não deve mostrar mensagens que este não entenda;
3. **Liberdade e controlo do utilizador:** Facilitar o utilizador em situações de este estar “perdido” permitindo sempre desfazer ou refazer as suas ações;
4. **Consistência:** O sistema deve “falar” sempre a mesma linguagem. Ações iguais devem possuir, por exemplo, mensagens iguais;
5. **Prevenção de erros:** O sistema está preparado para prevenir algum engano do utilizador. Por exemplo, quando este quer eliminar algo, deve ser mostrada uma mensagem de confirmação;
6. **Reconhecer em vez de lembrar:** O utilizador não deve fazer uso da memória para realizar uma tarefa. O sistema deve oferecer ajuda contextual e informações que orientem o utilizador;
7. **Flexibilidade e eficiência de uso:** O sistema deve ser fácil para utilizadores inexperientes, mas também deve ser flexível para os mais experientes. O uso de teclas de atalho ou tabs é um exemplo;
8. **Estética e design minimalista:** Os diálogos do sistema devem ser simples, diretos e naturais. Presentes nos momentos certos.
9. **Ajudar os utilizadores a reconhecer, diagnosticar e tratar erros:** As mensagens de erro devem ser simples e que levem o utilizador a entender o que fez de errado para não voltar a fazer;
10. **Ajuda e documentação:** Um bom conjunto de documentação ou opções de ajuda são exemplos para facilitar o uso do software ao utilizador.

3.4 Análise de Tecnologias

Nesta secção estão descritas as tecnologias que foram utilizadas para o desenvolvimento do software. A escolha correta deste processo é um passo fundamental no que toca ao desenvolvimento de software. É também importante realizar uma análise sobre o futuro do sistema, uma vez que serão adicionados novos módulos e é necessária realizar a manutenção deste programa. Posto isto, segue-se a apresentação das tecnologias utilizadas no desenvolvimento da plataforma bem como uma breve demonstração de utilização das mesmas.

3.4.1 PHP

A linguagem de programação PHP [30] (um acrónimo recursivo para “PHP: Hypertext Preprocessor”, originalmente Personal Home Page) é uma linguagem de desenvolvimento de aplicações para o lado do servidor.

Esta será a linguagem a ser utilizada para o desenvolvimento do servidor que irá comunicar com a aplicação. PHP está no topo das escolhas no que toca ao desenvolvimento de conteúdo dinâmico World Wide Web (WWW). Tem a grande vantagem de que a restante equipa de desenvolvimento da empresa está familiarizada com esta tecnologia e utilizam-na regularmente, sendo que no futuro será assim mais fácil de realizar tarefas de manutenção e suporte à aplicação.

Contudo, PHP tem a grande desvantagem de juntar código do lado do servidor com a página web a ser demonstrada. Como exemplo:

```
<!DOCTYPE html>
<html>
<body>
<?php
    // Aqui ficará o código do servidor. Juntamente com a página HTML
    echo "Hello world!<br>";
?>
</body>
</html>
```

Figura 3.15 – PHP puro junto com HTML

O exemplo acima demonstrado é apenas um pequeno excerto de código, mas quando se trata de aplicações com centenas ou milhares de linhas, torna-se ilegível para quem está a programar. Com isto, surgiu a necessidade de obter informação sobre algumas “*frameworks*” disponíveis para facilitar o processo de desenvolvimento e organização do código.

3.4.2 Laravel

Laravel [31] é uma framework PHP livre e open-source que utiliza o modelo MVC (Modelo, Vista e Controlador). Esta foi a framework escolhida pois pode ser considerada uma ótima ferramenta, uma vez que a sua curva de aprendizagem também é bastante reduzida. Utiliza uma sintaxe simples e concisa e contém um gestor de dependências. A documentação de Laravel é bastante rica, o que facilita muito a que a aprendizagem desta ferramenta seja rápida.

Como mencionado, ao utilizar o padrão MVC esta separa o código do lado do cliente do lado do servidor, garantindo assim uma forma de organização e estrutura de código mais fácil e “limpa”. Como exemplo, podemos ver a função seguinte que tem uma função que terá todo o código em PHP e retorna para a vista uma variável para que seja mostrado ao utilizador os dados:

```
public static function authenticateUser() {
    (...)
    return view('AuthViews/login')->with('utilizador', $utilizador);
}
```

Figura 3.16 – Exemplo PHP com Laravel

Laravel utiliza também o Composer[32] que permite gerir todas as dependências do projeto e integra consigo o PHP Unit [33] que permite ao programador entrar em ambiente de testes, criar os mesmos e testar toda a aplicação de forma fácil e flexível.

Alguns sites conhecidos a nível internacional foram desenvolvidos com recurso a esta framework, tais como o fallout4.com, wattbike.com, dubicars.com e analyticsportfolio.com.

A framework, para além do PHP, utiliza também as tecnologias que estão mencionadas nas secções seguintes.

3.4.2.1 HTML e Javascript

HTML[34] é a abreviação para a expressão inglesa HyperText Markup Language, é uma linguagem de programação utilizada para a construção de páginas web.

Javascript [35, 36] é uma linguagem de programação interpretada. É uma linguagem bastante usada para correr scripts do lado do cliente e muito útil na construção e apresentação do conteúdo da página a desenvolver.

As vistas da aplicação serão construídas com recurso a HTML e a Javascript.

3.4.2.2 Blade

A framework Laravel usa o Blade [37] que é um motor de templates bastante simples mas muito poderoso. Este é o motor utilizado como front-end. Escrever dados enviados pelo servidor, no ecrã do cliente é bastante fácil e prático com esta tecnologia. Esta tem uma linguagem própria, contudo não restringe única e exclusivamente o programador a utilizá-la, podendo colocar código PHP no meio da vista (algumas frameworks não permitem esta opção. Restringem o código a ser com a tecnologia que vêm e não permitem utilizar outra).

O código Blade é compilado para código PHP simples e armazenado em cache até ser modificado. Os ficheiros com as vistas são gravados com o formato [nome_vista].blade.php e estes são armazenados numa pasta própria.

Um exemplo de como pode ser utilizado o código Blade no meio de uma vista é apresentado na imagem seguinte:

```
<td>{{ $utilizador->nome }}</td>
```

Como podemos observar, no meio de tags de código HTML podemos utilizar facilmente a variável que foi enviada para a vista na secção 3.4.2 e, obter o nome do utilizador.

Este motor também possui funcionalidades como ciclos e “if” que facilitam o processo de construção de página web e apresentação de conteúdo. Um exemplo do ciclo for pode ser consultado na imagem seguinte:

```
@foreach ($utilizador as $uti)
    <li>$uti->nome</li>
@endforeach
```

Figura 3.17 – Exemplo de Blade

3.4.2.3 Eloquent

A framework Laravel contém uma ORM [39] (abreviação para a expressão inglesa Object-Relational Mapping) que tem como objetivo converter os dados que vêm de uma base de dados em classes. Com isto, o programador pouco necessita de utilizar código SQL [40].

O Eloquent [38] permite fazer todo o tipo de consultas à base de dados, abrindo e fechando sempre as conexões à mesma, sem que o programador se preocupe com elas.

Posto isto, uma consulta a uma tabela chamada “Utilizador” que retorna a variável que foi utilizada nos exemplos anteriores, é efetuada da seguinte forma:

```
$Utilizador = DB::table('Utilizador')->get();
```

Figura 3.18 – Exemplo do Eloquent

A consulta acima demonstrada, guarda na variável \$Utilizador os dados da tabela “Utilizador” transformando os mesmos para uma classe.

3.4.3 PHP Unit

A framework Laravel contém uma API de testes que permite testar a aplicação. A tecnologia utilizada neste ambiente é o PHP Unit, uma framework de testes unitários orientada à programação em PHP, sendo uma instância da arquitetura xUnit [41].

Nos projetos Laravel, existe já uma pasta que é gerada automaticamente que contém um simples teste de exemplo. Nessa pasta é onde deve ser gerado e colocado todos os testes que se pretendem executar nas aplicações.

Após a criação dos testes é através da linha de comandos que estes são executados com o comando:

```
C:\wamp\www\CSMPortal>phpunit [CLASS]_
```

Figura 3.19 – PHP Unit linha de comandos

Onde [CLASS] é o nome da classe que contém o teste a executar. Um exemplo de um teste que pode ser realizado pode ser consultado na imagem seguinte.

```
public function addDataProvider() {
    return array(
        array(1, "name"),
        array('100110997', "name"),
        array('08232', "name"),
        array('XPTO', "name")
    );
}

/**
 * @dataProvider addDataProvider
 */
public function testAdd($a, $b)
{
    echo "Testing Client: " . $a . "\n";
    $this->assertNotNull(
        $this->client->searchCustomer($a, $b)
    );
}
```

Figura 3.20 – PHP Unit data provider

O código apresentado na imagem anterior demonstra um *data provider* que contém um array de dados que irão ser passados para a função “testAdd”. A função recebe estes dados através da anotação “@dataProvider addDataProvider” que está descrita em cima da função.

Com esta API, para além de asserções, é possível interagir diretamente com a aplicação, ou seja, é possível simular cliques, preenchimento de formulários, entre outras opções que a mesma disponibiliza. Um exemplo pode ser consultado na imagem seguinte.

```

public function testBasicExample()
{
    echo "Test B\n";
    echo "Trying to reach 'Create Complaint Header'\n";
    // Visit -> ir para a página com o link /createNew
    // 1 - irá para a página de criar nova reclamação
    $this->visit('/createNew')
        // 2 - Preencher formulário
        // ->type([INPUT], '[NOME DO CAMPO NO FORMULARIO]
        ->type('AS Wendre', 'customer_name')
        ->type('100110997', 'clientes')
        ->type('TESTE', 'complaint_num')
        // ->select: select box
        ->select('1', 'complaintType')
        ->select('1', 'companyType')
        ->type('21-12-2016', 'arrival_date')
        // 3 - Pressionar o botão submeter
        ->press('submeterBtn');
    echo "-----\n";
}

```

Figura 3.21 – PHP Unit exemplo de navegação pela página

O objetivo deste teste é iniciar a criação de uma reclamação. O utilizador com login efetuado encontra-se na página inicial e pretende visitar a página “Create New”. Os dados são inseridos nos locais apropriados e é pressionado o botão de submeter a reclamação. Se este processo ocorrer de forma inválida o PHP Unit informa que a função “testBasicExample” disparou um erro.

3.4.4 MySQL

A utilização de MySQL [42] é a escolha efetuada no que toca ao desenvolvimento e manutenção da base de dados. Esta possui características que a destacam das demais, como por exemplo:

- Portabilidade, suportando praticamente qualquer plataforma;
- Excelente desempenho e disponibilidade;
- Uso de poucos recursos de hardware;
- Software livre com base na GPL (entretanto, se o programa que aceder o MySQL não for GPL, então deverá ser adquirida uma licença);
- Replicação facilmente configurável;
- MySQL workbench que permite criar e ver o modelo de forma simples e fácil.

Para além das vantagens mencionadas nos pontos anteriores, esta também é a tecnologia utilizada pelos restantes elementos da equipa de desenvolvimento da Borgstena há bastante tempo, pelo que a familiaridade com MySQL também teve um peso importante na decisão de escolha da tecnologia. O desenho da base de dados é realizado no MySQL workbench e, graças às funcionalidades de “Forward Engineer” é possível converter o desenho em um script que de seguida é corrido no phpmyadmin⁷ (local onde a base de dados se encontra).

⁷ <https://www.phpmyadmin.net/>

3.4.5 Validação de documentação

Na gestão de reclamações existe a necessidade de realizar o lançamento de notas de crédito. As notas de crédito são documentos financeiros que têm como objetivo reembolsar o cliente caso o resultado final da reclamação assim o indique. Estes documentos são sempre validados por elementos da empresa e só depois são enviados para o departamento financeiro, onde serão processados e se dará a transação monetária. Com isto, existe a necessidade de reforçar a segurança da plataforma a desenvolver. Assim sendo, o mecanismo que se pretende é a introdução de um código antes da nota de crédito ser validada e, cada utilizador tem um código diferente para a mesma nota.

Posto isto, foram estudadas as tecnologias que se seguem.

3.4.5.1 Google API – Authenticator

O Google Authenticator [43, 44] foi uma das tecnologias estudadas, no âmbito de entender se esta seria uma boa ferramenta a utilizar no que toca à validação de documentação. Tal como o nome “authenticator”(em português “autenticador”) indica, esta API é utilizada em operações de autenticação. Esta tem como objetivo reforçar a segurança das contas dos utilizadores que, para além de realizarem o login com o seu username e password, têm também de consultar um código que é renovado de 30 em 30 segundos no seu smartphone e introduzir o mesmo. Um exemplo dos passos de segurança até à aprovação do documento pode ser visualizado na imagem seguinte.

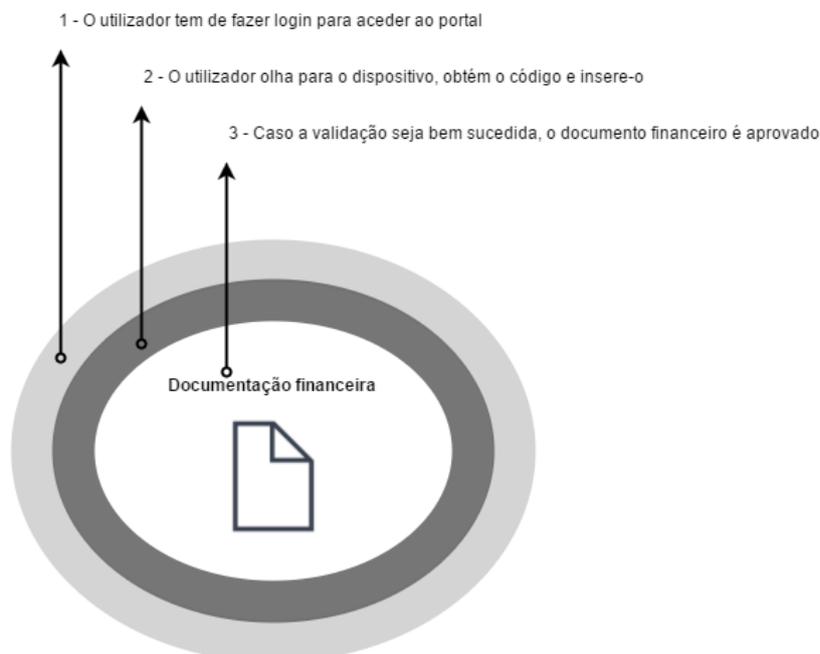


Figura 3.22 – Google Authenticator

Este autenticador depende sempre de uma chave secreta que a aplicação e o servidor necessitam de guardar. Esta “seed” é combinada com o tempo (data e hora) e é então gerado o código secreto.

No que toca ao projeto, esta ferramenta seria utilizada no ato de validação de documentos financeiros. Ou seja, quando são pedidas notas de crédito, estas têm de ser aceites pelos responsáveis de tal ação e, tal aprovação aconteceria quando estes se autenticassem na

aplicação, selecionassem o documento a aprovar e, introduzissem o código gerado pela API na aplicação. Esta última por sua vez, seria responsável por validar o código e caso este estivesse correto, o documento seria validado por aquele responsável.

Foi implementado um protótipo funcional que pode ser consultado no Anexo F, onde o utilizador insere o código da API e o sistema verifica se o código introduzido é válido ou não.

3.4.5.2 Mensagens de texto

As mensagens de texto, SMS (sigla que significa “Short Message Service”), foi uma das alternativas que se estudaram em relação ao Google Authenticator. A empresa possui um serviço gratuito de envio de mensagens de texto para os seus números. Este serviço inclui um script que está alojado nos servidores da operadora que tem como objetivo enviar um pedido HTTP com o destinatário e mensagem a enviar. Com isto, foi estudado e implementado um protótipo funcional. Quando uma nota de crédito é pedida, o servidor é responsável por registar o pedido, gerar um código secreto para cada utilizador, enviar uma mensagem de texto para o mesmo e, guardar o código encriptado na base de dados.

Após a receção, o utilizador acede ao portal, seleciona o documento e insere o código recebido por mensagem. O sistema por sua vez irá validar o código e consequente resultado de aprovação da nota de crédito.

3.4.5.3 Conclusão

Comparando as duas tecnologias descritas, ambas são soluções válidas e podiam ser utilizadas para solucionar o problema descrito de aprovação de documentação. Com elas é possível levantar as seguintes vantagens e desvantagens:

	Vantagens	Desvantagens
Google Authenticator	<ul style="list-style-type: none"> • Sem dependência da operadora de serviços; • Acesso rápido; • Funciona mesmo sem rede; 	<ul style="list-style-type: none"> • Maior esforço para entender como funciona para se implementar no código; • Exige mais código; • Exige instalação de uma app no dispositivo móvel; • Se chave partilhada descoberta, facilmente se descobre o código gerado;
Mensagens de texto	<ul style="list-style-type: none"> • Não necessita de instalação de nenhum recurso no telefone; • Fácil de implementar; • Notificação; 	<ul style="list-style-type: none"> • Dependência da operadora de serviços; • Fácil interceptação; • Se não houver rede, o serviço não funciona;

Tabela 3.4 – Vantagens e desvantagens do Google Authenticator vs SMS

Posto isto e, comparando as vantagens e desvantagens de ambos os serviços, podemos concluir que nenhum deles é uma forma eficaz de garantir a segurança da validação de

documentação. Mas a implementação de um passo a mais com uma destas tecnologias, apesar de não ser 100% eficaz, é mais segura do que a não implementação.

Tendo em consideração a tabela anterior, optou-se pela implementação da tecnologia de **Mensagens de texto**, porque esta não envolve ao utilizador final a instalação de um recurso no seu dispositivo e este não necessita de aprender a utilizar uma ferramenta nova (importa referir que por mais fácil que a aplicação do google authenticator seja de usar, os utilizadores finais nunca são dessa opinião).

3.4.6 Dropzone.js

O Dropzone.js [45] é uma API que permite implementar a funcionalidade de anexação de ficheiros. Uma vez que existe a necessidade de implementar a funcionalidade “drag’n’drop” na aplicação (muito útil no que toca à função de anexação de ficheiros) e, esta se torna uma tarefa extremamente árdua de programar, existiu a necessidade de procurar uma biblioteca que podesse implementar esta funcionalidade. Isto porque como já foi referido, é uma tarefa que iria ocupar demasiado tempo ao estagiário e, caso não existisse uma ferramenta que facilitasse a implementação desta funcionalidade, a mesma seria implementada mas apenas com a anexação a ser efetuada da forma tradicional (um botão que abre uma janela e permite navegar até escolher o ficheiro a anexar).

Com isto, foi possível encontrar e realizar alguns testes de exemplo com o Dropzone.js que é uma biblioteca “open source” que fornece um serviço de “drag’n’drop” de ficheiros e permite também a pré-visualização dos mesmos. Tem a vantagem de ser uma ferramenta altamente flexível pois permite o “upload” de qualquer tipo de ficheiros, fornecendo funções para o tratamento de qualquer necessidade que se queira implementar, facilitando assim o trabalho a quem está a utilizar a aplicação.

3.4.7 Chart.js

O Chart.js [46] é uma API em Javascript que permite ao programador criar gráficos de forma fácil e flexível. Esta, funciona do lado do cliente, ou seja, é executada quando é apresentado o conteúdo ao utilizador. Para isso é necessário receber os dados do servidor e processar os mesmos nas vistas que se pretendem apresentar. Integra funcionalidades de interação com os gráficos, permitindo assim facilitar a implementação de algumas funções, como por exemplo ver ou esconder determinadas linhas ou colunas de gráficos.

Capítulo 4: Especificação de Requisitos

Esta secção apresenta os requisitos do projeto que foram implementados no decorrer do estágio. Este capítulo encontra-se estruturado da seguinte forma:

- **Requisitos funcionais:** onde estão descritos os requisitos funcionais levantados e aprovados pela empresa.
- **Requisitos não funcionais:** irão ser descritos os requisitos não funcionais levantados, exigidos e aprovados pela empresa

Importa também referir que foram criados diagramas de casos de uso (DCU) para representar as iterações entre os diferentes atores e o sistema. Estes foram fundamentais no que toca a complementar os requisitos, uma vez que quando apresentados em reunião permitiu aos presentes opinar sobre as funcionalidades que estavam representadas. Isto revelou-se útil pois possibilitou a melhoria ou adição de novas funções. Para além dos diagramas, foram criados também protótipos iterativos para a aplicação que podem ser encontrados no Anexo A. Estes protótipos foram importantes uma vez que permitiram identificar, melhorar e adicionar mais requisitos ao sistema. Foram criados e implementados com recurso à ferramenta InvisionApp.

Posto isto, é relevante referir que todos os requisitos funcionais foram estimados e, o resultado dessas estimativas pode ser consultado na Secção 2.1.2.4 deste documento.

4.1 Requisitos Funcionais

Nesta secção encontram-se descritos os requisitos funcionais do portal a desenvolver. Informação mais detalhada é possível ser encontrada no documento SRS, desenvolvido com o intuito de registar todos os requisitos necessários e aprovados pela empresa.

Os requisitos encontram-se descritos em tabelas que se apresentam com o seguinte formato e conteúdo de dados:

ID	Nome	Módulo	Descrição	Dep.	Nível
Identificação do requisito	Nome do requisito	Módulo a que pertence	Descrição do requisito	Possíveis dependências	Nível de prioridade

Tabela 4.1 - Descrição da tabela de requisitos

Importa também clarificar que os níveis de prioridade classificam-se de acordo com o método “*MoSCoW*” [47], ou seja:

- “must have” – 3
- “should have” – 2
- “could have” – 1
- “won’t have” – 0

Importa ainda clarificar o leitor que os níveis 2 e 3 representam requisitos que são necessários implementar, sendo que o de nível 1 deverá ser apenas implementado no final dos outros dois caso exista possibilidade. Caso contrário, o requisito irá transitar para uma implementação de trabalho futuro.

Com isto, o levantamento de requisitos que se encontra a seguir apresenta apenas os requisitos mais relevantes, sendo que os restantes podem ser consultados no anexo A ao documento. Note-se que foi utilizada a tabela template demonstrada anteriormente, sendo ainda importante referir que ao longo do desenvolvimento do produto poderão surgir pequenas alterações e/ou melhorias aos requisitos.

4.1.1 MC-IR: Inserir reclamação

ID	Nome	Descrição	Dep.	Nível
IR-1	Preencher formulário de reclamação	Funcionalidade que permite preencher e validar o formulário da reclamação		3
IR-2	Selecionar cliente	Funcionalidade que permita selecionar determinado cliente e carregar part-numbers associados a ele do IFS	-	3
IR-3	Selecionar part-number	Funcionalidade que permite selecionar determinado part-number, filtrado por cliente e preencher dados do formulário automaticamente	IR-2	3
IR-4	Inserir nova linha	Funcionalidade que permite inserir nova linha na reclamação	-	3
IR-5	Cálculo automático de custos	Funcionalidade que obtém informação do IFS e apresenta cálculos de custos automaticamente de acordo com essa informação e com os dados da nova linha	IR-4	3
IR-6	Enviar mail para setor	Funcionalidade que envia um mail ao setor responsável pelo produto a ser reclamado	IR-1	1
IR-7	Controlo do identificador numérico de reclamações	Funcionalidade que permite a gestão automática da numeração de reclamações	-	3

Tabela 4.2 – Requisito funcional Inserir reclamação

IR-1: Preencher formulário de reclamação

Este requisito tem como objetivo fornecer ao utilizador um formulário apropriado para o preenchimento da reclamação. Também deverá ter as respetivas validações.

IR-2: Selecionar cliente

Este requisito tem como objetivo selecionar os clientes de determinado responsável, ou seja, um responsável autenticado terá uma lista de clientes só sua. Esta seleção irá automaticamente carregar os part-numbers associados a esse cliente (dados que vêm do IFS).

IR-3: Selecionar part-number

Este requisito tem como objetivo selecionar determinado part-number, ou seja, deverá filtrar todos os part-numbers de um cliente X e preencher os restantes dados do formulário de forma automática.

IR-4: Inserir nova linha

Este requisito tem como objetivo a inserção de uma nova linha (part-number reclamado) em determinada reclamação. Note que uma reclamação pode ter vários part-numbers reclamados.

IR-5: Cálculo automático de custos

Este requisito tem como objetivo o preenchimento automático de campos relativos a determinado part-number e cálculo de custos automático de acordo com essa informação e com os dados da nova linha

IR-6: Enviar mail para setor

Este requisito tem como objetivo o envio de um mail para o setor responsável da reclamação.

IR-7: Controlo do identificador numérico de reclamações

Funcionalidade que permite a gestão automática da numeração de reclamações.

Como exemplo, caso sejam reclamações de clientes:

- Se a reclamação for do tipo QUA (Qualidade) deverá ficar com o número 001
- Se uma segunda reclamação for do mesmo tipo, deverá ficar com o número 002
- Caso chegue uma terceira com tipo DEV (Desvio), deverá ficar com o número 001
- Se chegar nova reclamação do tipo QUA (Qualidade), deverá ficar com o número 003

Caso a reclamação seja a fornecedores o número é sempre incrementado não havendo restrições.

4.1.2 PA-IPA: Inserir Plano de Ação

ID	Nome	Descrição	Dep.	Nível
IPA-1	Criar formulário	Funcionalidade que permite criar um novo plano de ação		3
IPA-2	Preencher dados automaticamente	Funcionalidade que permite preencher automaticamente campos do formulário	-	3
IPA -3	Obter funcionários da empresa	Funcionalidade que permite obter todos os funcionários da empresa para atribuição de tarefas	-	3

Tabela 4.3 – Requisito funcional Inserir plano de ação

IR-1: Criar formulário

Este requisito tem como objetivo criar um plano de ação para determinada reclamação. O formulário a utilizar deverá ser sempre o da empresa. Só assim é possível gerir e garantir todos os dados necessários.

IR-2: Preencher dados automaticamente

Este requisito tem como objetivo o preenchimento automático de alguns campos do formulário.

IR-3: Obter funcionários da empresa

Este requisito tem como objetivo obter os funcionários da empresa. No plano de ação é necessário indicar ações de como solucionar o problema e, cada ação é atribuída a um funcionário.

4.1.3 MF-OC: Ordens de compra

ID	Nome	Descrição	Dep.	Nivel
OC-1	Obter dados do IFS	Funcionalidade que permite ao sistema receber dados do IFS	-	3
OC-3	Enviar mail a fornecedor	Funcionalidade que envia um mail ao fornecedor a avisar que existe uma nova ordem de compra a verificar	OC-2	3
OC-4	Atualizar ordem de compra	Funcionalidade que permite atualizar ordem de compra	OC-1	3
OC-5	Abrir ordem de compra	Funcionalidade que permite abrir uma ordem de compra	-	3
OC-6	Ver lista de ordens de compra	Funcionalidade que permite ver a lista de ordens de compra	-	3

Tabela 4.4 – Requisito funcional de ordens de compra

OC-1: Obter dados do IFS

Este requisito permite ao sistema esperar e receber eventos despoletados no IFS para obter dados relativos a uma nova ordem de compra que tenha sido lá inserida. Este requisito deverá ocorrer do lado do IFS sempre que uma ordem de compra é inserida, atualizada ou removida e, o sistema receberá o evento e tratá-lo-á de forma a fornecer informação atualizada ao fornecedor.

OC-2: Inserir nova ordem de compra

Este requisito permite ao sistema inserir a ordem de compra lida no requisito F-OC-1, na base de dados.

OC-3: Enviar mail a fornecedor

Este requisito permite ao sistema criar, estruturar e enviar um mail automático ao fornecedor envolvido na ordem de compra, alertando o mesmo para consultar o portal pois existe uma encomenda para o mesmo.

OC-4: Atualizar ordem de compra

Este requisito permite ao sistema atualizar a ordem de compra lida no requisito F-OC-1, na base de dados. Deverá executar o requisito F-OC-3 alertando o fornecedor que uma ordem de compra foi alterada.

OC-5: Abrir ordem de compra

Este requisito permite ao fornecedor ou ao utilizador interno da Borgstena abrir e ver o conteúdo de uma ordem de compra.

F-OC-6: Ver lista de ordens de compra

Este requisito permite ao fornecedor ou ao utilizador interno da Borgstena ver uma lista com todas as ordens de compra. No caso do fornecedor, este só verá ordens de compra que foram feitas ao próprio.

4.2 Requisitos não funcionais

4.2.1 Manutenibilidade

A manutenibilidade está relacionada com o custo da mudança e refere-se à facilidade com que um sistema de software pode acomodar mudanças. Tendo em consideração que os processos podem sofrer alterações, é importante que o código esteja preparado para ser facilmente alterado. Para isso é necessário ter em consideração que é necessária uma organização cuidada do código para que no futuro, a manutenção da aplicação seja simples e prática.

4.2.2 Segurança

Um sistema seguro tem a capacidade de resistir a tentativas de ataques externos ou internos.

Apesar de o Laravel garantir a segurança (caso o programador programe as queries de forma específica com funções que a framework disponibiliza) de alguns ataques como SQL Injection [48], Cross-Site Scripting (XSS) [49] e Cross-Site Request Forgery [50] foram realizados testes para garantir que estes ataques não iriam ocorrer na aplicação. Para além destes ataques, foi também tida em consideração que existe um módulo que irá ser acedido por terceiros (fornecedores). Este, deverá ser colocado num servidor que permita acessos externos mantendo as restantes aplicações em servidores cujo acesso é apenas e só apenas interno. Esta aplicação não irá realizar comunicação com nenhum outro software sendo que apenas o IFS despoleta eventos para a base de dados da mesma para que esta se mantenha atualizada no que toca às ordens de compra.

4.2.3 Usabilidade

A usabilidade é um requisito não funcional que avalia o quão fáceis as interfaces de utilizador são de usar. Para este estudo foram seguidos os 3 limites temporais de resposta enunciados por Jakob Nielsen[51]:

- **0.1 segundos:** limite para que o utilizador sinta que o sistema reage instantaneamente;
- **1.0 segundos:** limite quando o utilizador se começa a aperceber do atraso;
- **10 segundos:** limite para que o utilizador se mantenha focado. Os utilizadores precisam de uma reorientação.

Para além dos tempos foram tidas também em conta no desenho do design, a lista de heurísticas de Jakob Nielsen apresentada na secção 3.3.4.

4.3 Restrições

A Borgstena impôs duas restrições para o desenvolvimento ao projeto. A primeira foi o facto de que o módulo de gestão de fornecedores deveria ser colocado num servidor à parte que permitisse acessos externos. Este não deveria comunicar com nenhum dos internos de modo a evitar possíveis ataques.

A segunda restrição para o desenvolvimento do projeto foi o facto de que a aplicação deveria ser desenvolvida em PHP. Contudo, o estagiário efetuou o estudo de outras linguagens para verificar se existiam benefícios em relação ao PHP para o projeto a desenvolver. Posto isto, importa ainda referir que foram realizados pequenos exemplos práticos das tecnologias a seguir apresentadas.

- **Angular JS [53]**

Esta é uma tecnologia que permite estender o vocabulário HTML. Facilita o desenvolvimento de aplicações e seus testes. Esta não iria trazer benefícios ao projeto uma vez que Laravel incorpora o Blade.

- **React.js, Ember.js e Backbone.js [54]**

Estas são tecnologias concorrentes a Angular, pelo que são muito similares à mesma. Da mesma forma que a anterior, estas não trariam benefícios ao projeto uma vez que o Laravel incorpora o Blade.

- **CodeIgniter [55]**

É uma *framework* PHP semelhante a *Laravel*. Embora seja uma *framework* bastante simples de se utilizar, começou a entrar em declínio quando comparada com *Laravel*. A documentação da *Laravel* é muito boa quando relacionada com a existente de *CodeIgniter*, pelo que optar por esta última levaria a uma curva de aprendizagem maior por parte dos restantes membros de desenvolvimento da empresa. Também perdeu manutenção por parte da *ellislab*, sua criadora, começando a ficar demasiado desatualizada.

- **Django [56]**

É uma framework para desenvolvimento web, escrita em python, que utiliza o modelo MTV (Modelo-Template-Vista). Por ser uma linguagem totalmente nova para os demais elementos da equipa de desenvolvimento, esta é uma grande desvantagem pelo que não foram verificados benefícios para o projeto e seu futuro.

- **Ruby on Rails [57]**

É uma framework livre, escrita em Ruby. Utiliza o modelo MVC (Modelo-Vista-Controlador). Uma linguagem com “*runtime speed*” baixo quando comparada com php e tem pouca documentação disponível. Não foram verificados benefícios para o projeto e seu futuro.

- **.NET [58]**

É uma plataforma criada pela Microsoft. Todo e qualquer código gerado para esta tecnologia pode ser executado em qualquer dispositivo que possua um framework de tal plataforma. Este último é uma das grandes desvantagens de utilização desta plataforma e, somando a essa, Visual Studio utiliza versões demasiado caras e a migração de aplicações torna-se demasiado dispendiosa. Não foram verificados benefícios para o projeto e seu futuro.

- **C# [59]**

É uma plataforma Windows que permite desenvolver aplicações. Por ser uma tecnologia que envolve licenças demasiado caras. Não foram verificados benefícios para o projeto e seu futuro.

- **Struts e Java Persistence Api (JPA) [60]**

São duas tecnologias que permitem a criação de um servidor e apresentação do conteúdo de páginas com recurso ao Java. Sites como a marvell.com, boston.com e vodafone.co.uk são exemplos de sucesso utilizando estas tecnologias. Embora sejam boas alternativas para o PHP e sejam livres, a empresa decidiu manter a sua escolha inicial pois achou desnecessário obrigar a equipa de desenvolvimento a estudar esta tecnologia para uma futura manutenção.

- **Spring [61]**

À semelhança de struts, esta tecnologia permite a criação e apresentação de conteúdo de páginas com recurso ao Java. Pela mesma razão da anterior, foi descartada, uma vez que não foram encontradas vantagens relevantes para a substituição de PHP e, obrigar a equipa a estudar esta tecnologia para uma futura manutenção, apenas traria desvantagens.

Posto isto e, das tecnologias acima referidas, nenhuma das demais demonstrou ser uma mais valia para se utilizar na implementação do projeto pelo que PHP continuou como a linguagem escolhida.

Capítulo 5: Arquitetura

Neste capítulo está descrita a arquitetura planeada e escolhida para o desenvolvimento da aplicação. Esta, é um elemento fundamental no que toca à gestão de um projeto de software. Uma boa combinação de componentes arquiteturais levará a que o trabalho durante o desenvolvimento seja menor e, que o sucesso no final da implementação esteja mais perto. Posto isto, importa referir também que toda a arquitetura que é apresentada neste capítulo foi criada, apresentada e aprovada pela empresa Borgstena que, exigiu desde início que esta fosse simples de entender, mas eficaz no que toca ao software final pretendido.

No desenho da arquitetura é também fundamental e imprescindível que esta seja pensada de forma a alcançar ao máximo uma maior flexibilidade na integração de diferentes sistemas, ou seja, deve ser suficientemente flexível para suportar a adição ou substituição de componentes.

Posto isto, a constituição desta capítulo encontra-se dividida em 6 secções distintas, são elas:

- **Camadas:** nesta secção encontra-se apresentada a arquitetura no modelo de camadas com o intuito de indicar de formas simples, alguns componentes e tecnologias a utilizar camada a camada. Cada camada encontra-se devidamente explicada;
- **Componentes:** nesta secção encontra-se apresentada, de forma detalhada, toda a comunicação que irá ser necessária para que o portal funcione de forma correta, bem como todos os componentes e formas que se ligam entre si;
- **Routing:** nesta secção encontra-se apresentada a forma de como irá ser tratada, de forma automática com recurso ao Laravel, todo o processo de *routing*, isto é, como é que a aplicação irá redirecionar os utilizadores de acordo com os seus “roles” e permissões para as páginas adequadas;
- **Login:** nesta secção encontra-se apresentada a forma como é integrada a aplicação de Login existente na empresa com a aplicação;
- **Validação de documentação:** nesta secção encontra-se descrita a forma e os processos que a validação de documentação necessita para a aprovação de documentação financeira;
- **Estruturas de dados:** nesta secção encontra-se apresentado o modelo de dados que está a ser utilizado como suporte ao portal.

5.1 Camadas

Nesta secção está ilustrada e descrita a arquitetura do sistema. A arquitetura por camadas, por ser simples de visualizar e demonstrar o princípio base do funcionamento do sistema, pode ser visualizada na imagem a seguir representada. Cada camada ilustra as respetivas tecnologias que são utilizadas na mesma:

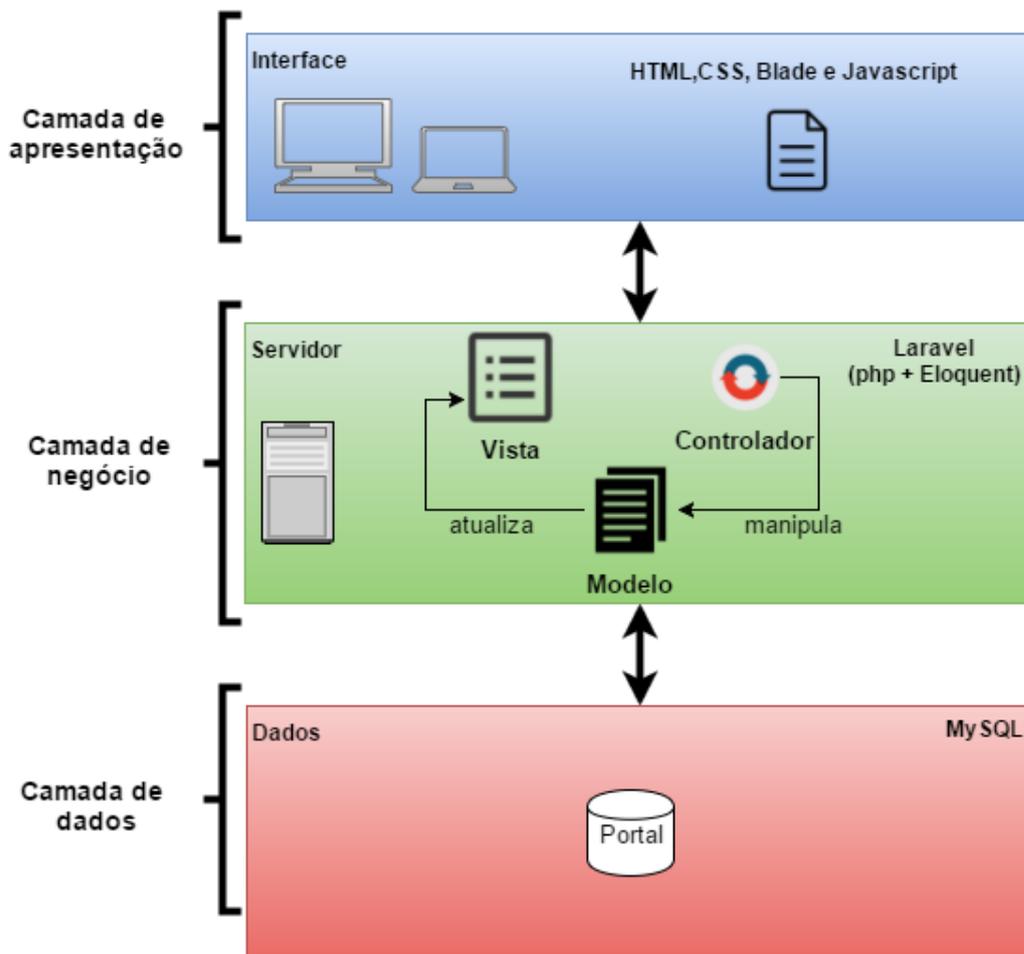


Figura 5.1: Arquitetura por camadas

Como é possível verificar, o sistema irá dividir-se em 3 camadas. Cada camada possui os seus componentes e tecnologias utilizadas nas mesmas. Facilmente se verifica que a camada de apresentação apenas comunica com a camada de negócio e, esta última apenas comunica com a camada de dados, ou seja, um pedido realizado pelo utilizador na camada de apresentação será de imediato processado na camada de negócio e, após esse processamento e, caso necessário, deverá recorrer à camada de dados realizando o caminho inverso até apresentar a vista de forma correta ao utilizador.

De seguida, irão ser apresentados, em detalhe, cada uma das camadas da plataforma bem como as tecnologias referentes a cada uma delas.

5.1.1 Apresentação

A camada de apresentação é a principal responsável por fornecer ao utilizador todas as interfaces web que ele necessita de visualizar. Com isto, torna-se necessário clarificar o seguinte conceito:

- **Interfaces web:** aplicações web que permitem aos utilizadores interagir com o sistema. Esta utilização pode ser de administração e gestão como de utilização comum.

Esta camada deve ser simples e estável. Nela são usadas as seguintes tecnologias:

- **HTML:** esta é utilizada na construção das páginas web;

- **CSS:** utilizada para definir a apresentação dos documentos escritos, neste caso, em html. Separa o formato do conteúdo facilitando assim a construção da página;
- **Blade:** tecnologia utilizada pela framework Laravel. Como ela é possível obter os dados enviados pelo servidor e, realizar todo o tipo de operações tais como ciclos;
- **Javascript:** utilizada para correr scripts do lado do cliente.

Estas tecnologias, são as principais responsáveis pela construção e apresentação correta das interfaces web.

5.1.2 Negócio

Nesta camada encontra-se o servidor com toda a lógica de negócio necessária para que os resultados da camada de apresentação sejam de acordo com as estratégias da empresa. A camada de negócio recebe e processa os pedidos vindos do cliente, retornando a sua resposta.

Nela é possível verificar que se encontra o padrão Modelo, Vista e Controlador (MVC). A camada de apresentação comunica com o controlador que por sua vez, manipula o modelo e, este último atualiza a vista que de seguida é retornada ao utilizador.

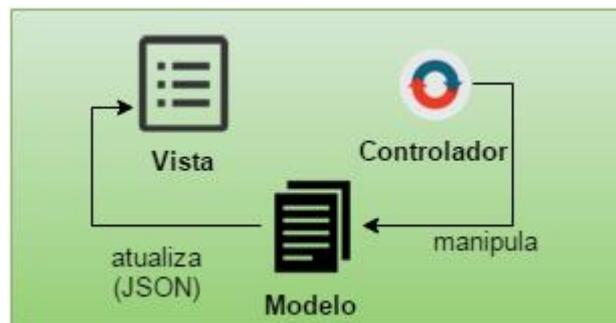


Figura 5.2 - Padrão MVC

Os dados que retornam do controlador para a vista serão objetos do tipo JSON. Este formato é muito utilizado pela framework Laravel e, permite uma troca de dados fácil e organizada.

Nesta camada podem ser encontradas as seguintes tecnologias:

- Framework Laravel
 - **Php**
 - **ORM Eloquent**

Estas, para além de permitirem a comunicação com a camada de apresentação, também são as responsáveis por procurar, criar, remover ou atualizar informação na camada de dados (Base de dados). Esta última camada encontra-se a seguir explicada.

5.1.3 Dados

A camada de dados permite persistir os dados necessários ao seu funcionamento. A camada de negócio comunica com esta através de consultas SQL. A camada de dados é composta pelos seguintes componentes:

- **Interno:** componente que contém a base de dados do portal. É lá que são realizadas todas as consultas SQL;
- **Externo:** componente que contém bases de dados existentes na empresa mas, que são externas ao portal. Com isto, importa clarificar os seguintes conceitos:
 - **IFS:** Base de dados onde são guardados quaisquer dados da empresa. Nesta é possível encontrar todo o tipo de informação necessária/desejada;
 - **SEval:** Base de dados de uma aplicação externa, com o intuito de avaliar os fornecedores.

Nesta camada é possível encontrar as seguintes tecnologias:

- a) MySQL

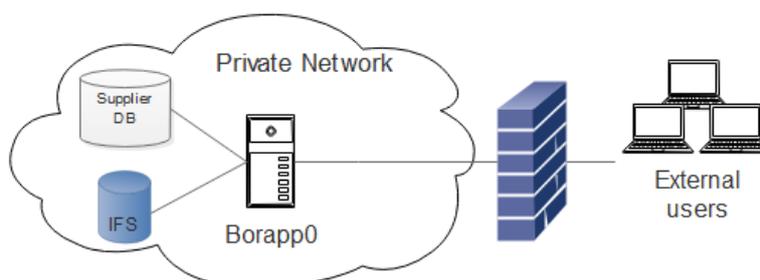
Com esta tecnologia é possível realizar consultas nas bases de dados necessárias.

5.2 Componentes

Todo o sistema irá trabalhar com métodos *GET* e *POST* feitos por parte do cliente (camada de apresentação) para o servidor (camada da lógica). Por sua vez o servidor ficará responsável por tratar esses métodos de acordo com as necessidades e irá enviar a resposta ao cliente.

Para garantir a segurança dos dados internos da empresa, mais concretamente os do IFS que detêm toda a informação vital da mesma, foi necessário estudar uma solução que evitasse o acesso “direto” (ver imagem seguinte) por parte de terceiros (neste caso fornecedores) a esta base de dados, ou seja, evitar manter o servidor a comunicar com o IFS, enquanto que terceiros estão conectados ao portal. Importa ainda clarificar a descrição dos servidores que são apresentados nas imagens seguintes:

- **Borlin0:** Servidor que permite acessos externos. É onde está o módulo da Gestão de ordens de compra;
- **Borapp0:** Servidor de produção onde estão os módulos de gestão de reclamações e planos de ação.



	Utilizadores		Base de dados		Comunicação bidirecional (Get e Post)
	Servidor		Firewall		

Figura 5.3 – Bird Eye View, problema de acesso

Posto isto, foram estudadas e propostas opções que propunham a solucionar e/ou atenuar o problema encontrado. Resta ainda clarificar que todas as decisões de escolha para a solução, onde o estagiário teve o objetivo de apresentar as soluções estudadas e indicar a que mais se adequava. As opções propostas são a seguir listadas:

- criar um servidor secundário, onde os fornecedores se conectam e, sempre que é necessário obter nova informação do IFS, este seria responsável por fazer o pedido a um servidor interno que por sua vez realizava as consultas no IFS e devolveria os dados pelo caminho inverso. Com esta solução, os fornecedores ficariam conectados a uma máquina que não interage diretamente com a base de dados que se pretende proteger, mas não é uma solução ideal.
- O servidor acede, de X em X tempo, ao IFS, fechando de imediato a comunicação. Esta solução seria utilizada em último recurso uma vez que continua a não impedir o acesso direto à base de dados;
- Sempre que existirem dados novos no IFS, este despoleta um evento onde envia a informação de novos dados para a base de dados de fornecedores;

Após a análise a opção c) foi a escolhida por parte da empresa. Importa ainda referir que foi realizado um pequeno teste para verificar se seria possível obter os dados enviados pelo IFS através de eventos, acerca de nova informação.

A figura que descreve a opção escolhida encontra-se a seguir representada.

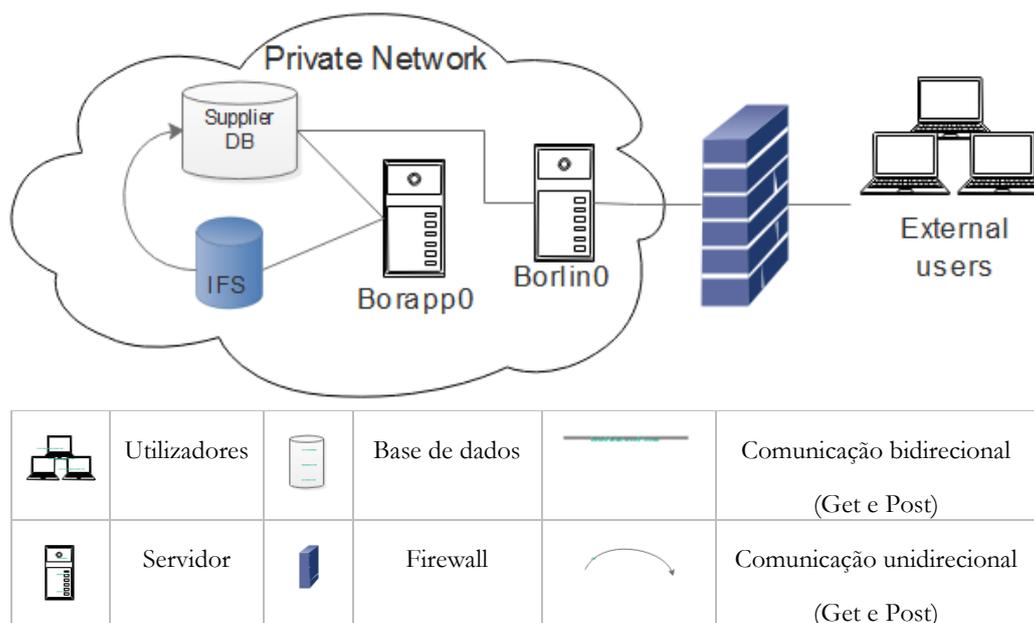


Figura 5.4 – Bird Eye View, solução do problema de 5.3

Como é possível constatar pela figura anterior, apenas o servidor e a base de dados do portal comunicam entre si, sendo que o IFS quando tiver novos dados, irá despoletar um evento que indicará ao servidor a nova informação a atualizar. Por sua vez, a base de dados do Portal será atualizada, sendo que sempre que um fornecedor se conectar e necessitar de obter informação, estes dados serão obtidos através da base de dados do Portal e nunca com acesso direto ao IFS. Note-se que os **eventos irão apenas enviar a informação necessária**, por exemplo, caso seja inserida uma nova ordem de compra, apenas os dados relativos a essa ação serão enviados para o servidor.

Posto isto, importa ainda referir os restantes componentes que fazem parte desta arquitetura e, que são imprescindíveis para o normal funcionamento do portal.

De acordo com a imagem a seguir representada, onde é possível verificar a comunicação entre componentes, importa ainda clarificar que o utilizador que se encontra na camada “Outside” é um fornecedor, sendo que o que se encontra na camada “Borgstena” é um gestor de clientes ou de fornecedores.

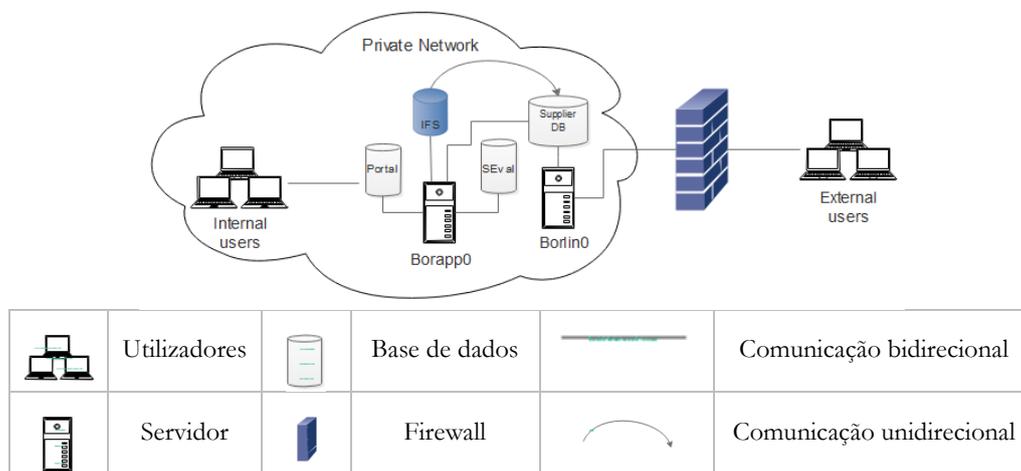


Figura 5.4 – Bird Eye View, arquitetura geral

A arquitetura do sistema, tal como demonstra a figura anterior, funciona à base de pedidos do cliente ao servidor. O servidor ou Server é o principal responsável por obter e distribuir a informação de que o cliente necessita. A descrição dos componentes acima representados é a seguinte:

- **Users** – Utilizador da Borgstena conectado à rede privada;
- **Server** – Servidor que recebe e responde a pedidos vindos por parte dos clientes;
- **IFS** – Base de dados industrial que atualmente é utilizada pela empresa para guardar e gerir toda e qualquer informação relevante para a empresa. Por questões de segurança, quando alguma informação é introduzida nesta base de dados, ela despoleta um evento com uma *query* que é enviada para o servidor do portal que, automaticamente irá tratar de manter a informação desse lado também atualizada. Note-se que na arquitetura está representada a direção em que se efetua a comunicação. Apenas o IFS transmite dados para o servidor e não o contrário (restrição descrita anteriormente).
- **Seval Database** – Base de dados que contém informação acerca de avaliações e reclamações realizadas a fornecedores. Esta guarda conteúdo detalhado de uma aplicação já existente na empresa e, que será útil para fornecer feedback aos fornecedores. Pertence a uma aplicação criada pela equipa de desenvolvimento existente.
- **Portal** – Base de dados MySQL onde serão guardados os dados.
- **Supplier** – Fornecedor que se irá conectar ao portal através de uma rede externa.

- **Private Network** – Identifica a rede privada que será acessível apenas a funcionários da empresa. Para aceder à mesma é necessário ser detentor de credenciais de acesso à mesma.

5.3 Routing

Devido ao projeto envolver funcionários com roles distintos, existe a necessidade de que haja uma separação do conteúdo que cada utilizador tem para as ações que pretende realizar. Com isto, importa lembrar que o portal será composto por 3 módulos distintos.

Para que seja possível a integração destes módulos no portal, é importante que se evitem cenários como o exemplo a seguir representado:

- “Sou gestor de fornecedores e consigo aceder a conteúdo de gestão de clientes”

Para que tal seja evitado, é estritamente necessário a implementação de funções de routing que, com a framework Laravel serão facilmente implementadas. Com uma estratégia correta de routing o cliente é encaminhado para o módulo com que deve interagir e, cada módulo possui as respetivas interfaces para que o utilizador possa realizar as operações necessárias.

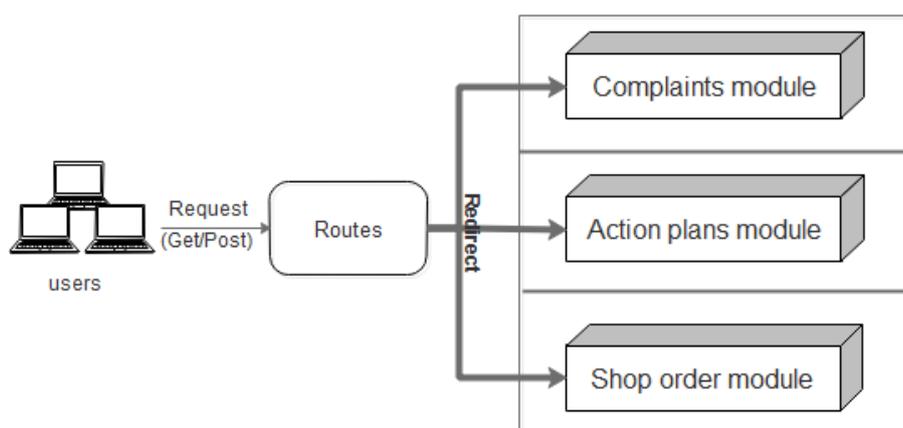


Figura 5.5: Funcionamento de routes

Como podemos observar pela figura anterior, o cliente quando tenta aceder ao CSM Portal, este é automaticamente redirecionado para o módulo correto. Caso o utilizador tenha o “role” que permite gerir reclamações, será encaminhado para o módulo de reclamações, por exemplo. De destacar que é **impossível** existirem **utilizadores com mais que um** “role”, daí a necessidade de separar as vistas e fornecer apenas o conteúdo que interessa a quem está a utilizar o portal.

Dentro dos módulos, como visto na especificação de requisitos, existem diferentes utilizadores e alguns só terão acesso a determinadas páginas, pelo que a mesma estratégia de routing também será aplicada.

5.4 Login

Como mencionado no capítulo 3 na secção 3.3.3 a Borgstena tem desenvolvida uma aplicação que é responsável por realizar o login de um determinado utilizador. Ela, irá validar o utilizador que se irá autenticar e após essa validação, irá mostrar um menu com todas as aplicações a que este tem acesso internamente na empresa.

Posto isto, existe a necessidade de integrar a aplicação com o Borgstena Login Application. Esta última, após o utilizador aceder à sua página principal, valida as credenciais do mesmo e cria uma variável de sessão com a informação de quem acabou de entrar no sistema.

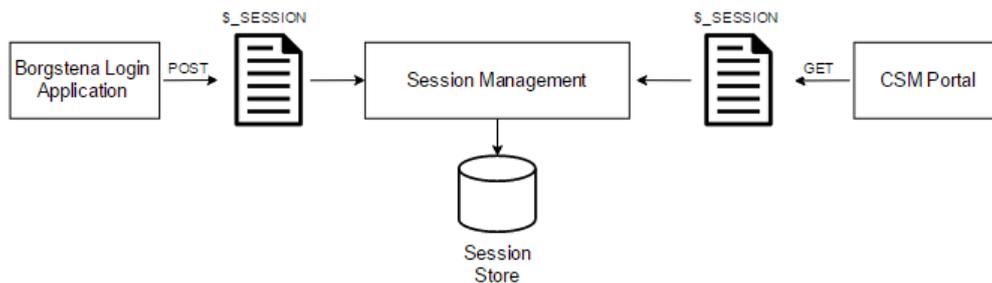


Figura 5.6 – Funcionamento de Login

A variável de sessão (`$_SESSION`) como podemos observar pela figura anterior, é armazenada pelo “Session Management”. Após o utilizador realizar o Login, terá acesso ao CSM Portal e, ao escolher o mesmo, este é responsável por obter a variável de sessão que contém dados relativos ao utilizador que está a tentar aceder ao portal.

Utilizando o sistema de Routing do Laravel (Secção anterior), é recebido um pedido de autenticação no CSM Portal que será automaticamente redirecionado para uma função que irá obter a variável de sessão guardada no browser, identificando que tipo de utilizador está autenticado e irá mostrar a página correta de acordo com o “role” que este desempenha (por exemplo, caso seja CEO é apresentada ver uma vista, caso tenha outro role é apresentada outra vista de acordo com o role do funcionário).

5.5 Validação de documentação

No que toca à validação de documentação é importante ter em conta que esta irá envolver um código secreto que deverá ser enviado para o responsável da aprovação. Importa ainda clarificar que é importante também inserir o mesmo de forma encriptada na base de dados para que garanta uma maior proteção desse dado. A encriptação utilizada é a encriptação por Base64.

Posto isto, segue-se a arquitetura de implementação desta funcionalidade:

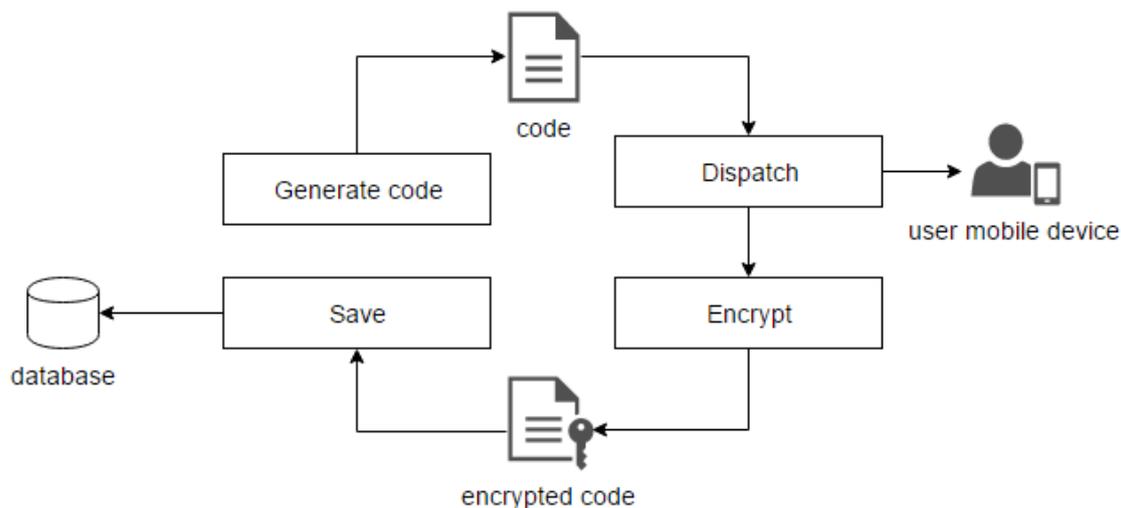


Figura 5.7 – Validação de documentação

Como é possível observar pela figura anterior, a aplicação é responsável por gerar um código que é enviado para o dispositivo móvel do utilizador e de seguida encriptar o mesmo para o guardar na base de dados. Importa ainda clarificar que quando o utilizador valida um documento, a comparação de códigos é feita encriptada. Ou seja, a mensagem indica o código “chave”. Quando inserido no portal, este é convertido para Base64 (“Y2hhdmU=”) e comparado com o dado salvo na base de dados que deverá ser igual.

5.6 Estruturas de Dados

A estrutura de dados do sistema foi desenhada e desenvolvida com o MySQL Workbench. Esta ferramenta revelou-se importante uma vez que permitiu criar o desenho da mesma e, como mencionado anteriormente, permite o “Forward Engineer” que permite a transformação do desenho que se segue na imagem seguinte, para código SQL.

Importa ainda clarificar que a base de dados que se segue faz ligação a outras fontes de dados que foram criadas pela empresa, contudo e por questões de segurança, esta não permitiu a divulgação das mesmas.

Como exemplo, a base de dados de funcionários armazena toda a informação dos colaboradores da empresa. Estes, são importantes nas ações no portal pelo que existe uma iteração com essa fonte.

Capítulo 6: Trabalho desenvolvido

Neste capítulo encontra-se descrito o trabalho desenvolvido com alguns exemplos que foram implementados pelo estagiário. Importa ainda clarificar que o capítulo se encontra dividido pelos seguintes pontos:

- **Protótipos:** onde se detalha os protótipos criados e testados com o utilizador final;
- **Ajax Request:** esta secção tem como objetivo demonstrar um exemplo de como a aplicação efetua os seus pedidos entre o cliente e o servidor, utilizando pedidos Ajax;
- **Mensagens de texto:** onde se detalha a implementação do serviço de sms para os responsáveis de aprovação de documentos;
- **Eventos do IFS:** exemplo dos eventos do IFS como explicado na Secção 5.2;
- **Dropzone.js:** nesta secção o leitor encontra detalhada a implementação da funcionalidade de “drag’n’drop” programada com recurso à API dropzone.js;
- **Chart.js:** esta secção contém detalhada a implementação da forma como são implementados os gráficos que são apresentados aos utilizadores.

Posto isto, importa ainda clarificar que as secções seguintes contêm alguns exemplos de funcionalidades reais que estão implementadas na aplicação.

6.1 Protótipos

Como referido no capítulo 3, secção 3.3.4, foram criados protótipos da aplicação com o objetivo de obter feedback por parte do utilizador sobre o design da aplicação. Estes protótipos permitiram ao estagiário corrigir e perceber como é que o utilizador final vê a aplicação e como é que pretende interagir com a mesma.

6.1.1 Criar reclamação

Os protótipos criados e estudados com os utilizadores que irão utilizar os mesmos, são a seguir apresentados.

Complaints [Create new](#)

Step-1

10% Complete

Customer:

Complaint number:

Type:

Arrival Date:

Aprov:

Progress:

Figura 6.1 - Criação do cabeçalho da reclamação

Como é possível observar pela figura acima apresentada e, comparando com o processo estudado no capítulo do estado da arte, estes são os campos necessários para a abertura da mesma. Serão sempre fixos pois serão os identificadores da mesma. Os dados que se podem observar no campo “customer code” ou “customer name” teriam de ser pesquisados manualmente pelos responsáveis, tarefa que demoraria imenso tempo. É essencial ter-se em consideração também que o cálculo do número da semana e, outros campos estão automatizados do lado do servidor, sendo que o utilizador não tem que se preocupar com a procura e inserção dos mesmos.

6.1.2 Inserir linhas de reclamação

Poderá ser possível visualizar na imagem a seguir apresentada, a descrição dos itens a reclamar e seus devidos problemas.

Complaints [Create new](#)

Customer: Britax / Status: Open / Date: 06/14/2016 / Number: 3 / Responsible:

50% Complete

Part-Number	Description	Technology	Defect	Unit	Action
+ Add new item					
Submit					

Figura 6.2 - Itens inseridos na reclamação

Na figura acima apresentada, é possível observar que existem um corpo onde irão ser listados os itens inseridos e reclamados pelo cliente com o respetivo “part-number”.

Na imagem a seguir apresentada é possível verificar a inserção de um novo item reclamado pelo cliente.

Add new item
×

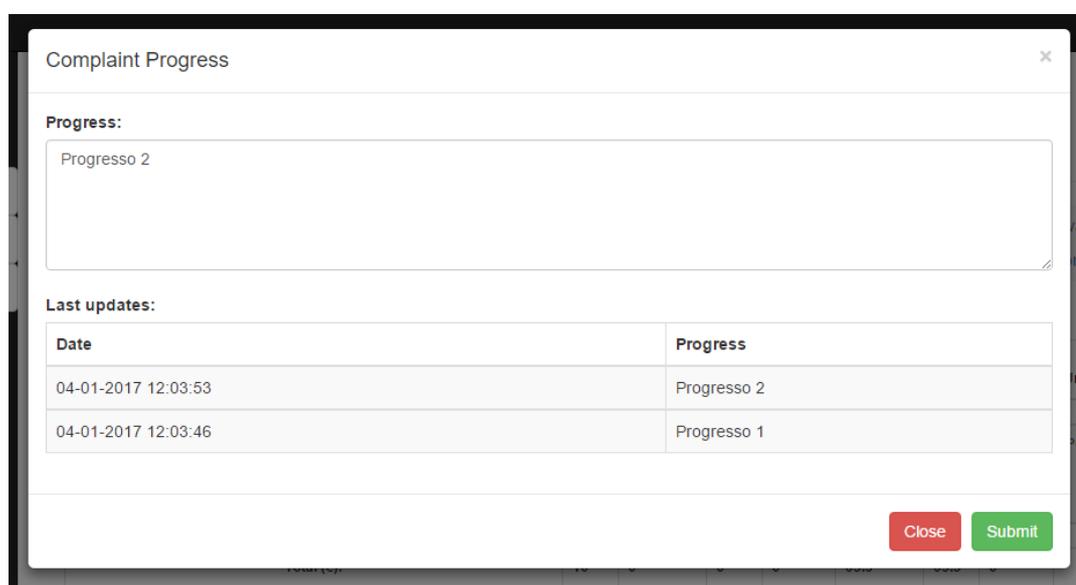
Part-Number:	<input type="text" value="-- Choose here --"/>	Description:	<input type="text"/>
Technology:	<input type="text" value="WK"/>	Defect:	<input type="text" value="Enter text"/>
Unit:	<input type="text" value="Covers"/>	Price unit:	€ <input type="text" value="0.00"/>
Meters Values:	<input type="text"/>	Batch:	<input type="text"/>
Issue cost:	€ <input type="text" value="0.00"/>	Scrapping cost:	€ <input type="text" value="0.00"/>
Extra cost:	€ <input type="text" value="0.00"/>	Logistic cost:	€ <input type="text" value="0.00"/>
Estimated cost:	€ <input type="text" value="0.00"/>	Real cost:	€ <input type="text" value="0.00"/>

Figura 6.3 - Método de inserção de itens na reclamação

Como é possível visualizar na figura 34, alguns campos serão preenchidos de forma automática. Por exemplo, quando o utilizador escolhe o “Part-number”, a descrição do mesmo será automática, algo que não acontece no sistema atual da empresa, onde o utilizador necessita de procurar tudo manualmente.

6.2 Ajax Request

Devido à integração do CSM Portal com a aplicação BLA (Borgstena Login Application referida no capítulo 3, secção 3.3.4), foi necessário implementar pedidos Ajax [62]. Com eles é possível realizar operações entre a vista e o servidor sem que o browser esteja constantemente a fazer o “*reload*” de toda a página mas sim apenas de partes da mesma. Como exemplo segue-se o processo de atualização de progresso de uma reclamação.



Date	Progress
04-01-2017 12:03:53	Progresso 2
04-01-2017 12:03:46	Progresso 1

Figura 6.4 – Atualização de progresso de reclamação

A imagem anterior mostra a janela modal que é utilizada para atualizar o progresso de determinada reclamação. Sempre que o utilizador escreve algo no campo “*Progress*” e clica no botão “*Submit*” é invocado um pedido Ajax. O código que é executado para enviar a atualização do progresso é apresentado na imagem seguinte.

```
$.ajax({
  type: "POST",
  url: href,
  data: $(this).serialize(),
  success: function( msg ) {
    $('#se-pre-con').hide();
    $('#myModalProgress').modal('hide');
    $('.modal-backdrop').hide();
  }
});
```

Figura 6.5 – Ajax request, exemplo

Como podemos verificar, trata-se de um pedido Ajax que tem como objetivo enviar para o servidor o formulário preenchido. Em caso de sucesso, é fechada a janela modal e a vista é

atualizada apenas no que toca ao seu progresso não sendo necessário realizar um “*reload*” da página no browser.

6.3 Mensagens de texto

As mensagens de texto no contexto da aplicação são parte vital para o bom funcionamento da mesma. Isto porque estas são as responsáveis por conter os códigos de aprovação de documentação financeira para os devidos responsáveis. Posto isto e, tal como referido na secção 3.4.5.2 a operadora que trabalha com a Borgstena fornece um script que permite realizar um pedido HTTP com o número e mensagem a enviar. Desta forma, o envio de mensagens é efetuado utilizando o seguinte código:

```
public static function sendPrivateMessage($number, $message){  
    try {  
        $client = new GuzzleHttpClient();  
        $apiRequest = $client->request('GET', 'http://SOME_IP/send_sms.php?user=USER&'  
            . 'password=SOME_PASSWORD&num='.$number.'&msg='.$message.'');  
        $content = json_decode($apiRequest->getBody()->getContents());  
    } catch (RequestException $re) {  
    }  
}
```

Figura 6.6 – Envio de mensagem através do portal

Como é possível verificar, o link do pedido envolve o IP destino (servidor da operadora), a chamada ao ficheiro php “send_sms.php” que é o script que a operadora fornece para este tipo de operações, o user e password que são da Borgstena. Posto isto, basta apenas passar ao link o número destinatário e a mensagem que se pretende enviar.

6.4 Eventos do IFS

Uma das componentes importantes do trabalho que irá ser desenvolvido para a parte da gestão de ordens de compra aos fornecedores é o acesso às encomendas que são introduzidas diretamente no IFS. Como já mencionado, o IFS é uma base de dados que guarda todo o tipo de informação da Borgstena e, por esses motivos é necessário proteger a mesma de acessos diretos. Para evitar esta questão, foi implementado um sistema que permita obter eventos do IFS e, com eles, preencher a base de dados com a informação necessária.

Posto isto e, tendo em consideração que foi a primeira vez que foi desenvolvido um sistema que recebesse eventos do IFS, foi necessário efetuar um estudo aprofundado com a equipa de programadores. Após se entender o modo de funcionamento dos eventos nesta base de dados, foi criada uma aplicação em PHP que simulasse o pretendido. Com isto, foi possível obter o seguinte resultado:

Procurar por Nr de PO ou Nr de Fornecedor...

Texto:

	Contract	Order_no	Vendor_no	Vendor_name	Objstate	Date_entered	Change_date
Details...	BTP	P139009	001768701	TOSCANA GOMMA SPA	Released	2016-06-29	2016-06-29
Details...	BTP	P139004	069454505	PROTECHNIC SA	Released	2016-06-29	2016-06-29
Details...	BTP	P139005	069454505	PROTECHNIC SA	Released	2016-06-29	2016-06-29
Details...	BTP	P139029	17482993	ANTEX S.A.	Released	2016-06-29	2016-06-29
Details...	BTP	P139032	193994 916	TWE Dierdorf GmbH & Co. KG	Released	2016-06-29	2016-06-29
Details...	BTP	P139031	261399182	TWD Fibres Service GmbH	Released	2016-06-29	2016-06-29
Details...	BTP	P139033	267297745	Trevira GmbH	Released	2016-06-29	2016-06-29
Details...	BTP	P139035	267297745	Trevira GmbH	Released	2016-06-29	2016-06-29
Details...	BTP	P139036	501638733	ATB - Acabamentos Têxteis de Barcelos, Lda	Released	2016-06-29	2016-06-29
Details...	BTP	P139034	501638733	ATB - Acabamentos Têxteis de Barcelos, Lda	Released	2016-06-29	2016-06-29
Details...	BTP	P139012	503443182	DyStar Anilinas Texteis Unip, Lda	Released	2016-06-29	2016-06-29
Details...	BTP	P139014	507218787	MCS Textile Solutions, Lda	Released	2016-06-29	2016-06-29
Details...	BTP	*118434	507218787	MCS Textile Solutions, Lda	Closed	2016-06-29	2016-06-29

Figura 6.7 – Exemplo de Encomendas que o evento do IFS despoletou

De acordo com a imagem anterior podemos observar as ordens de compra que têm sido introduzidas na base de dados do IFS. Sempre que um gestor responsável introduza uma encomenda, o evento ocorre e é recebido no servidor e tratado de acordo com a necessidade correta. Note-se que neste caso só está a ser guardado e mostrado ao cliente sendo que ainda não é possível realizar nenhum dos requisitos pretendidos.

6.5 Dropzone.js

Como já mencionado anteriormente, a aplicação irá necessitar de ter uma funcionalidade “drag’n’drop” para que se possam anexar ficheiros. Desta forma, será mais fácil ao utilizador anexar ficheiros, bastando apenas arrastar os mesmos, em vez de necessitar de abrir as tradicionais janelas e procurar a pasta e devido ficheiro. Com isto, foi implementada uma funcionalidade que permita ao cliente utilizar esta funcionalidade, como poderá visualizar na imagem a seguir apresentada, o exemplo que foi desenvolvido e que está pronto a utilizar no produto final.

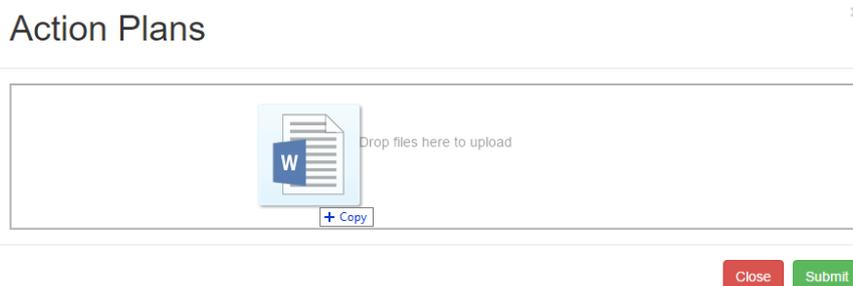


Figura 6.8 - Ecrã de exemplo de “drag’n’drop” do dropzone.js

Note-se que o ficheiro irá ser colocado dentro do retângulo, mas só será feito o seu upload para o servidor assim que o utilizador clicar em “submit”. Isto porque pode haver um engano e é desnecessário o servidor estar a ser sobrecarregado com ficheiros que podem ser inúteis e depois terão de ser apagados. O exemplo após a submissão é o seguinte:

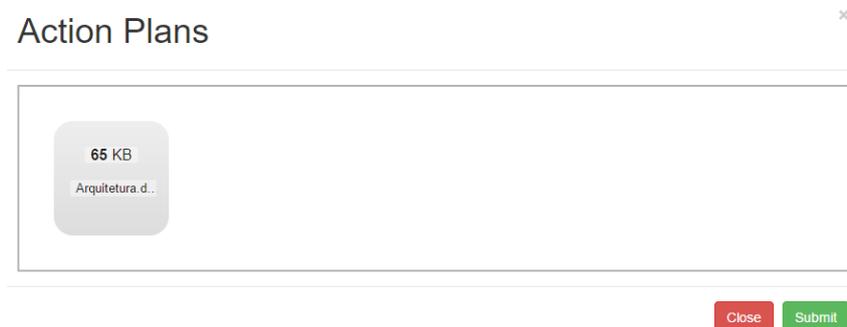


Figura 6.9 - Ecrã de exemplo de “upload” do dropzone.js

Importa ainda referir que a ação do botão “submit” é realizada graças às funções disponibilizadas pelo Dropzone.js que permitem controlar todos os eventos de “drag’n’drop” que forem realizados.

6.6 Chart.js

Tal como mencionado na secção 3.4.7, o Chart.js é uma API Javascript que permite gerar gráficos. Utiliza elementos Canvas que são integrados no HTML para a demonstração gráfica de informação. Como podemos observar pela imagem seguinte, o elemento canvas declara-se dentro do HTML:

```
<div class="row">
  <div class="col-lg-12" style="height: 400px;">
    <canvas id="ppm_p" class="col-lg-12" height="100%"></canvas>
  </div>
</div>
```

Figura 6.10 – Declaração do Canvas para o chart.js

Após indicarmos onde será criado o gráfico no código HTML, resta gerar o mesmo. Para isso é invocada uma função em Javascript:

```
function createChart(ctx, data, labels) {
  var myChart = new Chart(ctx, {
    type: 'bar',
    data: {
      labels: labels, responsive: true,
      datasets: [{
        type: 'bar', label: "Months",
        data: data,

```

Figura 6.11 – Criação do gráfico em Chart.js

Esta função, para além de ter opções para alterar o aspeto do gráfico (cores por exemplo), interpreta arrays de dados e de labels. Neste caso as labels podem ser meses e os dados podem ser custos. O tipo de gráfico é o “bar” ou de barras.

Capítulo 7: Testes

Neste capítulo encontram-se descritos os testes efetuados ao portal desenvolvido. Este encontra-se dividido em 4 secções diferentes, sendo elas explicadas nos pontos a seguir.

- **Verificação de requisitos funcionais:** nesta secção encontra-se descrito os testes efetuados ao código da aplicação. São efetuados um conjunto de passos, é introduzido um input e, verifica-se se o output é o esperado ou não. Esta verificação é efetuada com recurso ao PHP Unit;
- **Validação de requisitos funcionais:** nesta secção é explicada a forma como a aplicação foi testada com os utilizadores finais e, que problemas foram encontrados bem como a solução para os mesmos;
- **Avaliação de heurísticas:** nesta secção encontra-se descrito o feedback encontrado relativamente à utilização do portal por parte dos utilizadores finais;
- **Verificação de requisitos não funcionais:** nesta secção encontra-se descrito a verificação (por testes) dos requisitos não funcionais, anteriormente mencionados.

7.1 Verificação de requisitos funcionais

Para testar os requisitos funcionais da aplicação, foram criados casos de teste que permitiram verificar se determinado requisito estava bem implementado ou se continha erros. Se este apresentasse erros, era imediatamente corrigido e testado novamente para garantir que o problema estaria corrigido. Em casos em que uma funcionalidade implicava a alteração de outras já implementadas, estas últimas eram novamente testadas para garantir que não tinham sido comprometidas pelas alterações efetuadas. As validações que se seguem foram efetuadas todas com recurso ao PHP Unit, como explicado na secção 3.4.3.

A lista completa de testes pode ser consultada no Anexo C. Como se pode verificar, nem todos os requisitos que foram implementados passaram nos testes aplicados.

Importa ainda clarificar os campos das tabelas que foram usadas nos casos de teste:

ID	Requisito(s) relacionado(s)	Resultado
[Identificador do teste]	[Identificação do requisito que está a ser testado]	[Aprovado ou Reprovado]
Descrição	[Descrição do teste]	
Input	[Dados de input a serem introduzidos]	
Output Esperado	[Output esperado da aplicação]	
Output Real	[Output obtido com o teste]	

Tabela 7.1 – Template do teste de verificação de requisitos funcionais

Posto isto, serão apresentados dois exemplos de testes. O teste que se segue é o exemplo de um caso aprovado com sucesso:

ID	Requisito(s) relacionado(s)	Resultado
CT-1	MC-IR	Aprovado
Descrição	O objetivo deste teste é selecionar um cliente que não exista no sistema e como output a aplicação deverá informar o utilizador de erro e não deverá permitir avançar na introdução da reclamação.	
Passos	<ol style="list-style-type: none"> 1. Clicar em <i>“My Complaints”</i> 2. Escolher a opção <i>“Create New”</i> 3. Preencher dados 	
Input	<ul style="list-style-type: none"> • Customer name: <i>“TESTING”</i> • Customer code: <i>“9999AAAA”</i> • Complaint number: <i>“comp 001 testing”</i> • Type: <i>“Qua”</i> • Company: <i>“BTP”</i> • Arrival Date: <i>“12-12-2016”</i> 	
Output Esperado	Mensagem de erro: <i>“The customer is not correct!”</i>	
Output Real	Mensagem de erro: <i>“The customer is not correct!”</i>	

Tabela 7.2 – Tabela de teste sobre o requisito MC-IR

O teste que se segue apresenta o caso de uma reprovação, ou seja, o teste foi efetuado e o Output Real não era o Output Esperado:

ID	Requisito(s) relacionado(s)	Resultado
CT-5	MC-IR	Reprovado
Descrição	O objetivo deste teste é introduzir valores inválidos nos campos e a aplicação indicar os campos errados	
Passos	<ol style="list-style-type: none"> 1. Clicar em <i>“My Complaints”</i> <ul style="list-style-type: none"> • Escolher a opção <i>“Create New”</i> • Preencher dados com o Input do Caso de teste CT-1 • Clicar em <i>“Next Step”</i> • Clicar em <i>“Add new item”</i> 	
Input	<ul style="list-style-type: none"> • Part-Number: <i>“P510271101/1”</i> • Quantity: <i>“11,2”</i> • Defect: <i>“Riscas /stripes”</i> • Issue Cost: <i>“11,2”</i> • Scrapping Cost: <i>“11,2”</i> 	

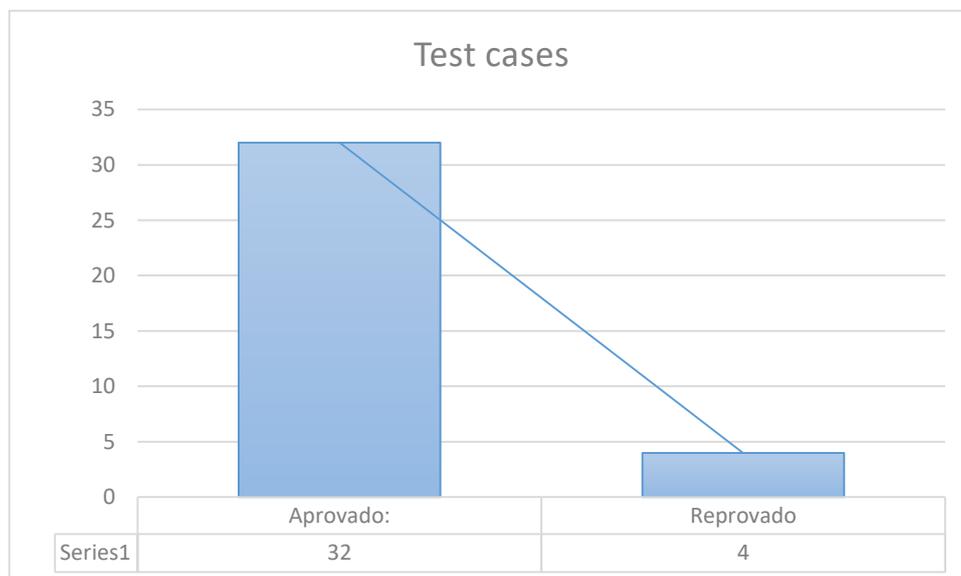
	<ul style="list-style-type: none"> • Logistic Cost: “11,2” • Extra Cost: “11,2” • Savings Cost: “11,2”
Output Esperado	O sistema deverá corrigir os números com vírgulas para pontos e aceitar a inserção no sistema
Output Real	O sistema não corrigiu as vírgulas para pontos.

Tabela 7.3 – Tabela de teste sobre o requisito MR-IR com erro

Como podemos verificar pelo Output Esperado, o sistema deveria ser capaz de corrigir os números inseridos com vírgulas (,) e passar as mesmas para pontos (.). Após o teste ser reprovado, o problema foi identificado e corrigido. As alterações efetuadas estenderam-se a outros requisitos, tais como:

- MC-AR
 - AR-3
 - AR-4
 - AR-5

Posto isto, o resultado de todos os testes efetuados pode ser consultado no gráfico que é a seguir apresentado. Nele é possível verificar a quantidade de testes executados bem como a quantidade dos que foram aprovados e reprovados.



Os testes que foram reprovados coincidem com um problema em comum. Os utilizadores finais quando inserem valores decimais podem usar vírgulas ou pontos (exemplo: 12.2 ou 12,2). O sistema não estava preparado para esta situação, sendo que no caso de o utilizador inserir números com vírgulas, no final só era guardada a parte inteira do mesmo. Após correção imediata dos problemas os testes voltaram a ser executados e aprovados.

7.2 Validação de requisitos funcionais

A validação dos requisitos funcionais é efetuada pelo utilizador final. No desenvolvimento do software é bastante comum os requisitos encontrarem-se totalmente implementados, mas no final o resultado não é o que o cliente pretende. Posto isto e apesar de terem sido implementados protótipos com o objetivo de os requisitos serem implementados o mais próximo possível do que é pretendido pelo cliente, achou-se necessário validar o software com o utilizador final antes deste entrar em produção.

Estes testes envolveram colocar a aplicação disponível em ambiente de testes para que os funcionários pudessem utilizar o portal, simulando os processos com dados que se aproximassem o mais próximo possível da realidade. Para isso foram utilizadas reclamações, planos de ação e ordens de compra reais. Não foi seguido um plano com regras e passos que o utilizador tivesse que seguir. Esta ação foi propositada uma vez que o objetivo, tal como já foi referido, é aproximar este teste o mais possível à realidade.

Apesar de não ter sido delineado um plano com regras e passos, foram também levantados os tempos que cada utilizador demorava para realizar determinadas tarefas. Este levantamento revelou-se importante pois assim foi possível avaliar o requisito não-funcional de usabilidade (Secção 7.5.3).

Posto isto, foram detetadas algumas falhas no software sendo as mais relevantes as que se seguem na seguinte lista (para a lista completa, consultar Anexo D):

- **Exportar reclamações a fornecedores para Excel**
 - **Problema:** a exportação de reclamações para formato Excel não estava disponível para os gestores de fornecedores;
 - **Solução:** apenas alguns campos mudam no que toca às reclamações de fornecedores pelo que a solução passou por adaptar as funções já criadas para exportação de reclamações de clientes;
- **Atualização de progresso nas reclamações**
 - **Problema:** é pretendido que todas as atualizações de progresso nas reclamações fossem registadas com data e hora para mais tarde serem consultados;
 - **Solução:** na estrutura de dados foi criada uma tabela que estaria relacionada com a reclamação. O utilizador ao abrir o progresso desta última tem acesso a todo o histórico de progresso relativo aquela reclamação;
- **Guardar ações em caso de falhas**
 - **Problema:** sempre que o utilizador cria uma reclamação e em caso de alguma falha (o utilizador errou navegou para outra página ou a energia falhou), os dados eram perdidos obrigando o utilizador a escrever novamente os dados;
 - **Solução:** A aplicação guarda os dados das reclamações de forma temporária numa tabela na estrutura de dados. Por exemplo, quando é criado um cabeçalho de reclamação, os dados ficam guardados até esta estar finalizada ou o utilizador alterar os dados;
- **Aceitar todos os itens de uma ordem de compra:**
 - **Problema:** o utilizador tinha de aceitar item a item. Caso o pedido envolvesse dezenas de linhas, este teria de aceitar uma por uma;
 - **Solução:** Implementação de um botão “Accept All” que permitia ao fornecedor aceitar todos os itens da ordem de compra;

Os problemas acima listados são os mais relevantes que foram registados durante esta ambiente de testes. Importa ainda referir que o ponto que se segue representa uma funcionalidade que deverá ser implementada no futuro de forma completa e que foi registada durante os testes de validação de requisitos. Existem reclamações de clientes onde estes reclamam problemas que não são relacionados com a Borgstena mas sim com o produto que o fornecedor vendeu. Estas são diretamente passadas do gestor de clientes para fornecedores. Embora tenha sido um requisito que não tenha sido inicialmente levantado, foi implementada uma funcionalidade que serve como solução.

- **Passar reclamação de clientes para fornecedores**

- **Problema:** transferir uma reclamação de clientes para o gestor de fornecedores. O objetivo é o sistema utilizar o máximo de dados possível e preencher automaticamente o formulário de reclamações a fornecedores;
- **Solução:** Implementação de um sistema de reclamações pendentes. O gestor de clientes transfere a reclamação para o gestor de fornecedores e este último é alertado via mail ficando com a reclamação pendente no portal. É gerada uma notificação quando o gestor se autentica e este pode efetuar o download. De seguida preenche o formulário e indica ao software que a reclamação está registada, deixando de estar pendente.

Relativamente a este último ponto importa referir que foi necessário ter em conta o planeamento de estágio bem como o tempo disponível para a implementação de uma solução.

7.3 Avaliação de Heurísticas

Um dos pontos salientados no Capítulo 3, Secção 3.3.4, foi a análise aos utilizadores e consequências que este estudo poderia trazer à aplicação criada. No seguimento da validação dos requisitos funcionais o estagiário teve o papel de avaliar a forma como os utilizadores interagem com a aplicação, com o objetivo de avaliar as heurísticas de Jakob Nielsen que foram implementadas. Apesar de esta situação ocorrer em um ambiente de testes foi importante ter algumas considerações uma vez que o utilizador poderia sentir-se pressionado com a presença de “avaliadores”. Posto isto, esta avaliação decorreu da seguinte forma:

- **Turnos:** os utilizadores foram definidos por turnos. Por exemplo, primeiro os responsáveis por reclamações de clientes. Num segundo turno os responsáveis por reclamações de fornecedores;
- **Local:** Sala de trabalho dos funcionários;
- **Avaliadores:** dois avaliadores sendo um deles o estagiário;
- **Utilizadores:** No máximo estariam 4 pessoas a ser avaliadas, 2 para cada avaliador;

Os problemas foram registados em tabelas. Estas tabelas podem ser consultadas em detalhe no Anexo E sendo que a explicação das mesmas se encontra a seguir apresentado.

No	Issue Description	Heuristic violated	Severity	Evaluator(s)	Proposed solution
[Número identificador do problema]	[Descrição do problema]	[Número da(s) heurística(s) violadas]	[Gravidade do problema]	[Quem apontou o erro]	[Soluções propostas pelo utilizador ou avaliador]

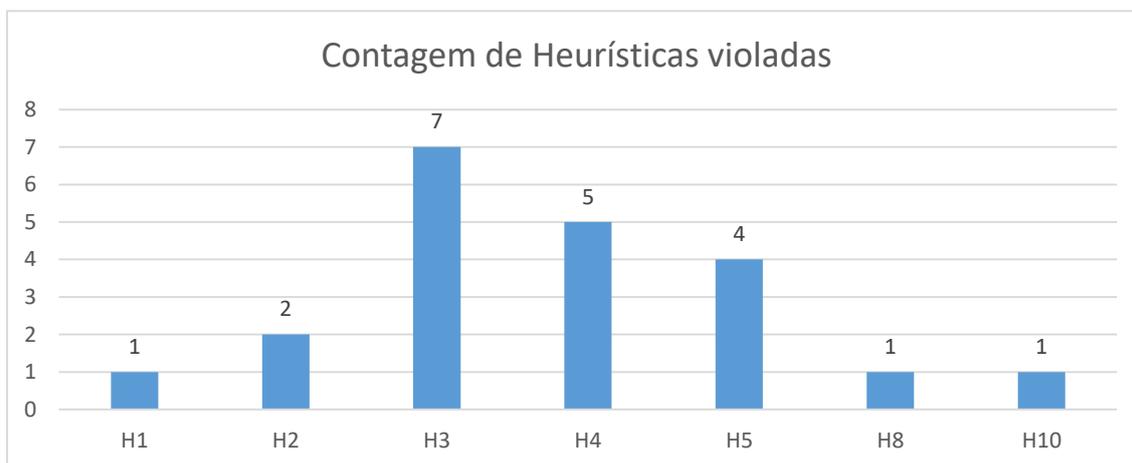
Tabela 7.4 – Template de avaliação de heurísticas

Importa ainda clarificar o leitor dos níveis de severidade que foram seguidos:

- **Nível A:** Problema grave que deve ser corrigido imediatamente. Este tem influência direta no sistema, ou seja, a informação poderá sofrer consequências ou alterações devido ao erro;
- **Nível B:** Problema médio que deve ser corrigido, mas que não influencia diretamente o sistema, ou seja, não afeta a informação, mas poderá causar um transtorno ao utilizador (por exemplo, este pode “ter-se perdido”);
- **Nível C:** Problema ligeiro que não representa qualquer perigo para a aplicação e seus dados, mas que o utilizador demonstrou estar desconfortável;
- **Nível D:** Sugestões dos utilizadores

É importante referir que não foram considerados problemas repetidos, isto é, no total de problemas não foram contabilizadas ocorrências repetidas.

Com isto, foi possível apontar alguns problemas na aplicação desenvolvida. O detalhe estatístico das heurísticas violadas segue-se no gráfico seguinte:



Como é possível observar as heurísticas que mais vezes falharam ao longo dos testes foram:

- H3: Liberdade e controlo do utilizador
- H4: Consistência
- H5: Prevenção de erros

Posto isto, importa ainda clarificar que o leitor da severidade dos problemas encontrados, sendo que o gráfico se encontra a seguir demonstrado:



Como é possível observar, foram encontrados problemas a nível de heurísticas. Foi importante solucionar estas falhas. De entre as soluções implementadas, é importante clarificar que todas foram solucionadas de acordo com as escolhas efetuadas pela equipa de desenvolvimento incluindo o estagiário e respetivos utilizadores finais.

7.4 Verificação de requisitos não funcionais

Nesta secção serão apresentados os requisitos não funcionais da aplicação. Estes são características que o sistema deve possuir em adição às funcionalidades e que foram avaliados no final da implementação do sistema para verificar se estes estão assegurados. Os requisitos não funcionais descritos nos pontos seguintes são: manutenibilidade, segurança, usabilidade

7.4.1 Manutenibilidade

Como forma de garantia do requisito mencionado, o código foi dividido por classes. Por exemplo, funções relativas a ações em reclamações estão na classe “Complaints”. Foi também tido em consideração que o código deve estar bem comentado e indentado para uma fácil interpretação. Para além dos cuidados referidos, as funções encontram-se também divididas dentro das classes. A classe tem uma área onde se encontram todas as funções responsáveis por métodos do tipo POST que envolvem “alterar algo” na base de dados e outra área para GET que envolvem “obter algo” da base de dados.

Posto isto, importa ainda referir que as funções foram mantidas o mais curtas e simples possíveis, separando sempre as diferentes tarefas que podem ocorrer na chamada a uma função.

7.4.2 Segurança

Como foi indicado em capítulos anteriores e, tendo em consideração que existe um módulo que ficará acessível do exterior, foi necessário ter em consideração medidas de segurança extra.

Este módulo que será o responsável pela gestão de ordens de compra e será acessido pelos fornecedores irá encontrar-se num servidor próprio que aceita pedidos externos. Posto isto, o fornecedor poderá conectar-se ao portal só e só apenas através deste servidor, não sendo possível ligar-se a outros. Importa ainda esclarecer que esta aplicação necessita de dados que estão no IFS. Este último encontra-se alojado em outro servidor diferente. A comunicação entre estes dois é feita apenas do IFS para a aplicação e não o contrário. O IFS, sempre que lhe forem inseridos dados novos, irá despoletar um evento que enviará os novos dados

(apenas os necessários) para a aplicação. Esta última jamais comunica diretamente com o servidor.

Posto isto, importa ainda referir outros tipos de segurança que estão implementados em todos os módulos desenvolvidos, que são:

7.4.2.1 SQL Injection

O SQL Injection é um tipo de “injeção” de código SQL na aplicação. Isto porque basta a um utilizador colocar numa caixa de texto uma *query*, submeter a mesma e a aplicação irá interpretar a *query* com a base de dados e retornar informação vital. Este é um pequeno exemplo de como este ataque pode destruir aplicações e bases de dados.

A framework Laravel juntamente com a sua ORM Eloquent utiliza o PDO (*PHP Data Object*) que não permite a injeção de código SQL nas queries. Contudo, esta proteção não é totalmente segura. Isto é, imaginemos o seguinte código que foi um dos exemplos corridos

```
SELECT * FROM company WHERE company = 'TESTE or 1=1';
```

como teste (assumindo que “*TESTE*” or *1=1* são os dados que vêm de um formulário):

O resultado da query em cima seria a listagem de todas as companhias da Borgstena, embora nenhuma se chame “teste”.

No Laravel é possível efetuar consultas de duas formas distintas. A primeira é a que se segue:

```
$dadosDoFormulario = "TESTE or 1=1";  
  
DB::select("SELECT * FROM company WHERE company = " . $dadosDoFormulario);
```

A forma que se apresenta permite a introdução direta de código SQL. Esta não é uma forma eficaz para prevenir a injeção sendo que o ataque seria um sucesso para o hacker, devolvendo as companhias na base de dados:

```
array:2 [▼  
  0 => {#282 ▼  
    +"id_company": 1  
    +"company": "BTP"  
  }  
  1 => {#284 ▼  
    +"id_company": 2  
    +"company": "BTCZ"  
  }  
]
```

Figura 7.1 – Resultado SQL Injection bem sucedido

Contudo, existe uma segunda forma de consultas que é apresentada a seguir. Esta previne a injeção de SQL utilizando as potencialidades da ORM Eloquent.

```
$dadosDoFormulario = "TESTE or 1=1";  
  
DB::table('company')->where('company', '=', $dadosDoFormulario)->get();
```

Esta, quando executada, irá devolver um array vazio porque a consulta quando gerada para MySQL, terá o seguinte aspeto (note-se que os dados do formulário são tratados como se fossem uma string apenas):

```
SELECT * FROM company WHERE company = "TESTE or 1=1";
```

Como consequência desta “compilação” dos dados do formulário, o problema SQL Injection encontra-se resolvido.

7.4.2.2 Cross-Site Request Forgery

O *Cross-Site Request Forgery* (CSRF) é um ataque que força um utilizador final a executar ações indesejadas numa aplicação Web no qual estes estão atualmente autenticados. Um ataque bem-sucedido, através de um simples formulário HTML, pode simplesmente transferir fundos de uma conta para a outra (por exemplo).

A framework Laravel guarda automaticamente um *token* CSRF que tem a informação do utilizador que está autenticado na aplicação. Como exemplo de utilização, imagine-se um formulário a ser submetido do cliente para o servidor. Este último irá comparar com uma variável de sessão o *token* enviado no formulário. Se estes coincidirem significa que o utilizador que se autenticou na aplicação é o mesmo que está a fazer as submissões de novos dados.

Com recurso ao Postman⁸, uma API que permite realizar pedidos POST e GET a aplicações foram efetuados testes para verificar a veracidade da segurança do Laravel em relação a ataques CSRF. Para isso, foi utilizado um simples método POST que na aplicação tem como objetivo remover dados da base de dados. O link é o que se segue:



Figura 7.2 – Exemplo de Post no Postman

Interpretando o link acima representado, este indica a remoção de um material com o id 5. Realizando o envio, a aplicação identifica logo que o pedido é inválido pois o token não coincide com a sessão que está registada no servidor.



1. in VerifyCsrfToken.php line 67
2. at VerifvCsrfToken->handle(object(Request), object(Closure))

Figura 7.3 – Erro de prevenção de Cross-site Request Forgery

Posto isto, importa ainda referir que este foi um de outros testes idênticos realizados cujo resultado obtido foi igual.

⁸ <https://www.getpostman.com/>

7.4.2.3 Cross-Site Scripting

Este tipo de ataque consiste em inserir scripts nas páginas web do lado do cliente. Imagine-se que é inserido o seguinte script na aplicação.

```
My list <script>alert("spam spam spam")</script>
```

Este simples ataque iria lançar uma janela de alerta com a mensagem “spam spam spam”. Mais uma vez, a framework Laravel previne este tipo de ataques. Ela renderiza as variáveis dentro de tags “{{ variável }}” que iriam interpretar a string acima referida da seguinte forma:

```
My list &lt;script&gt;alert("spam spam spam")&lt;/script&gt;
```

Esta interpretação irá evitar que o ataque tenha sucesso uma vez que não será interpretado como um script. O exemplo representa o teste que foi efetuado na aplicação e cujo resultado em nada prejudicou a aplicação pois a string era sempre interpretada como texto e não como um script.

7.4.3 Usabilidade

Como foi mencionado em 4.2.3, foram seguidas algumas linhas de guia para garantir a usabilidade da interface gráfica desenvolvida. Relativamente às 10 heurísticas de Jakob Nielsen é possível encontrar por toda a aplicação o cuidado e implementação das mesmas. Posto isto, seguem-se alguns exemplos delas:

- **5: Prevenção de erros**

A imagem que se segue representa uma situação em que o sistema indica ao utilizador que está prestes a confirmar que já registou uma reclamação que estava pendente.

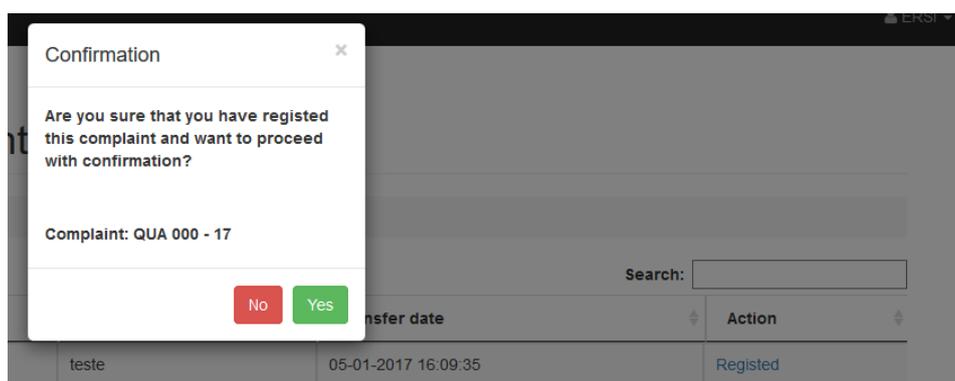


Figura 7.4 – Exemplo de prevenção de erros

10: Ajuda e documentação

Na imagem que segue é possível verificar um exemplo de um “tooltip” que ajuda o utilizador a entender a ação que irá desempenhar. Naquele caso, o rato sobre o visto a verde indica que é para aceitar a linha.



Figura 7.5 – Exemplo de Ajuda e Documentação

No que toca ao mencionado na Secção 4.2.3, foi possível verificar que a funcionalidade de exportação para Excel das reclamações e planos de ação necessitava de uma melhoria. Isto porque quando o utilizador clicava na opção de exportação, não existia um feedback indicando que o portal estava a gerar o relatório, ou seja, violava assim a primeira heurística referente à Visibilidade do estado sistema. O utilizador não obtinha informação sobre o que estava a acontecer e sentia-se perdido, voltando a clicar ou pedindo ajuda a quem estava a avaliar.

Posto isto, foi introduzida uma funcionalidade que transmite um feedback ao utilizador de que o relatório está a ser gerado.

Capítulo 8: Conclusão

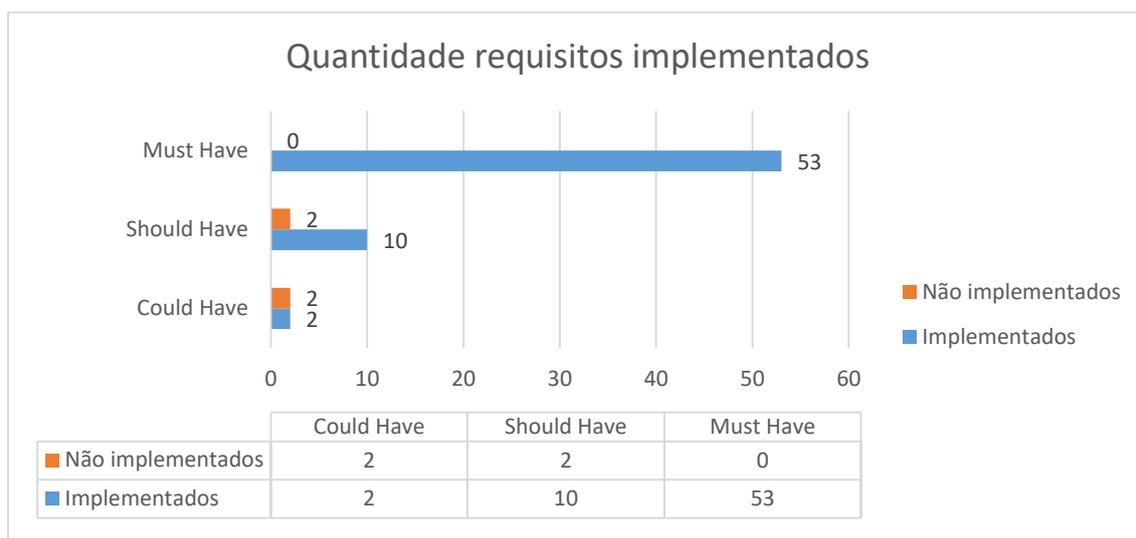
Este capítulo serve para apresentar as reflexões finais ao estágio realizado pelo autor deste documento. Nele são apresentados os requisitos implementados, dificuldades encontradas, bem como o trabalho futuro na aplicação.

Este estágio possibilitou um vasto conjunto de oportunidades de aprendizagem e desafios, tanto a nível pessoal, como profissional. Os objetivos traçados foram devidamente alcançados tendo também um feedback muito positivo por parte dos utilizadores finais.

Importa ainda clarificar que o software sofreu uma auditoria. Esta baseia-se na visita de profissionais à empresa que verificam e certificam os processos da empresa. Caso algum processo não esteja, na ótica deles, da forma mais correta. Estes deixam sempre o parecer deles anotando sempre os problemas que são registados. Em relação ao software desenvolvido, o feedback obtido por parte dos auditores foi muito satisfatório não tendo recebido qualquer anotação negativa para correção do mesmo.

8.1 Requisitos implementados

Nesta secção encontram-se detalhados os requisitos funcionais que foram implementados na aplicação, bem como os que não foram. É de se notar que os requisitos que não foram implementados, seguem-se com a devida justificação. Posto isto, segue-se um gráfico que representa a quantidade de requisitos implementados, por tipo. Como explicado na Secção 4.1, os requisitos podem ser de quatro tipos diferentes (“must have”, “should have”, “could have” e “won’t have”).



Como podemos verificar pelo gráfico anterior, existem requisitos que não foram implementados, sendo estes do tipo “should have” e “could have”. Os requisitos que não foram implementados são os seguintes:

Requisito	Tipo	Justificação
RC-2	Should-have	Por ser um processo novo a empresa optou por enviar a informação das credenciais de forma manual sem que a aplicação tivesse qualquer interação de forma direta com o fornecedor.
MR-2	Could-have	Os fornecedores, numa primeira fase, poderão anexar os documentos referentes aos planos de ação, no portal. Alguns fornecedores utilizam o seu próprio formato de planos de ação. Apesar de a Borgstena disponibilizar os seus e pedir nesse formato, alguns fornecedores não cumprem o solicitado.
IR-6	Could-have	Este requisito não foi visto como uma mais valia para o software uma vez que é sempre necessário os funcionários deslocarem-se pessoalmente para debaterem sobre o problema.

Tabela 8.1 – Requisitos funcionais não implementados

Posto isto e, relativamente ao gráfico apresentado, é possível verificar que os requisitos do tipo “must have” foram todos implementados, sendo que esses são referentes às funcionalidades mais importantes que o portal contém.

8.2 Dificuldades encontradas

Nesta secção encontram-se descritas as dificuldades encontradas ao longo do estágio pelo aluno e autor deste documento.

A nível pessoal, o facto de se estar a trabalhar num ambiente completamente novo para o estagiário é, numa primeira fase, uma das grandes dificuldades. Para além da adaptação e integração a um ambiente empresarial é necessário também ter conhecimento de regras e normas, estratégias que a empresa utiliza, as diversas áreas em que atua, entre outros.

A nível do desenvolvimento do projeto, uma das grandes dificuldades encontradas foi a comunicação. Esta, por vezes, era extremamente difícil uma vez que os responsáveis se encontravam em viagens ou em reuniões e, era ainda mais dificultada uma vez que nunca eram colocados todos os cenários possíveis. Ou seja, quando o estagiário entendia o problema, surgia sempre um “senão” tornando as tarefas mais árduas.

As dificuldades mencionadas, quer a nível pessoal e de desenvolvimento de projeto levaram a outra que foi o facto de o objetivo tardar a ser compreendido na sua totalidade. Estando o aluno em um ambiente completamente novo, não conhecendo os processos e estratégias de forma rigorosa como cada responsável, a difícil comunicação e a dimensão do projeto levaram a que o estudo do estado da arte e levantamento de requisitos se tornasse uma tarefa extremamente complexa.

Contudo, as dificuldades permitiram adquirir conhecimentos e critérios de trabalho que conduziram a uma solução sólida e funcional.

8.3 Trabalho futuro

O estágio, como é descrito no capítulo 1, teve como objetivo a implementação de um portal com 3 módulos e que sirva de base de um software CRM e SRM. Isto é, no futuro irão ser adicionados e/ou melhorados os já existentes, com o objetivo de se completar a solução cada vez mais por forma a conduzir a empresa a processos que possam ser melhorados e centralizando a informação cada vez com mais rigor. Centralizar e cruzar os dados relativos a clientes e/ou fornecedores é fundamental uma vez que permitem à própria empresa cada vez melhorar mais a sua forma de comunicação com estas entidades apresentadas.

Relativamente aos módulos implementados estão já pensadas melhorias e otimizações das funcionalidades programadas. Como exemplo, está prevista a implementação de funções que permitam “linkar” uma reclamação a outra. Isto é, caso o cliente faça uma reclamação que vem no seguimento da outra, a segunda fica automaticamente “linkada” á primeira.

Existe também já programada a implementação de um módulo que siga um tipo de leilões para vendas. O objetivo do mesmo é inserir um produto a comprar e colocar um preço. De seguida, os fornecedores poderão licitar. As 10 licitações mais baixas irão ser conduzidas para leilão onde a entidade fornecedora terá um determinado tempo para colocar a sua proposta. No final, o fornecedor “vencedor” será aquele que licitar mais baixo. Este módulo corre no exemplo do descrito em 3.2.2.2.

Posto isto é possível concluir que o software desenvolvido é “uma gota em um oceano” que irá ser melhorado e otimizado consoante as necessidades dos responsáveis da Borgstena. Neste momento, ele demonstra ser uma solução sólida e fiável, mas os processos podem sofrer alterações ao longo do tempo, bem como as necessidades podem vir a ser outras, contudo importa oferecer a continuidade do software à aplicação.

Referências

- [1] M. Vieira, Slides das aulas de Gestão de Projetos. Universidade de Coimbra, Coimbra, Portugal, setembro-dezembro 2015
- [2] “Gantt Project” <https://www.ganttproject.biz/>. Acesso: 03-03-2016
- [3] P. D. M. Vieira, “Module 3: Planning & Tracking – Planning & Monitoring,” 2015.
- [4] P. D. M. Vieira, “Module 3: Planning & Tracking - Bottom-up Estimation: A Real Process,” 2015.
- [5] “Wikipédia: Iterative development”
https://www.tutorialspoint.com/sdlc/sdlc_iterative_model.htm. Acesso: 10-03-2016
- [6] P. D. M. Vieira, “Module 3: Planning & Tracking – Risk Management”, 2015.
- [7] “Microsoft Office: Excel” <https://products.office.com/pt-pt/excel>. Acesso: 03-04-2016
- [8] “Eight Disciplines Problem Solving”
https://en.wikipedia.org/wiki/Eight_Disciplines_Problem_Solving. Acesso: 14-04-2016
- [9] “8D Problem Solving Process” <https://www.mindtools.com/pages/article/8d-problem-solving.htm>. Acesso: 14-04-2016
- [10] “Fishbone” <https://en.wikipedia.org/wiki/Fishbone>. Acesso: 14-04-2016
- [11] “IFS” <http://www.ifsworld.com/us/>. Acesso: 25-04-2016
- [12] “Relationship Management: CRM and SRM” <http://www.lexisnexis.com/sites/en-us/business-solutions/relationship-management.page> Acesso: 01-03-2016
- [13] “Customer Relationship Management” <http://www.itinfo.am/eng/customer-relationship-management/>. Acesso: 01-03-2016
- [14] “Como gerir clientes”
<http://www.strategyexpert.com/articles/managingclientexpectations>. Acesso: 03-03-2016
- [15] “Estratégias para gerir fornecedores” <http://www.industryweek.com/companies-amp-executives/10-strategies-managing-suppliers>. Acesso: 03-03-2016
- [16] “Salesforce” <https://www.salesforce.com/>. Acesso: 03-03-2016
- [17] “Salesforce developer” <https://developer.salesforce.com/>. Acesso: 03-03-2016
- [18] “Heroku” <https://www.heroku.com/>. Acesso: 03-03-2016
- [19] “Wikipédia: Plataforma as a Service”
https://en.wikipedia.org/wiki/Platform_as_a_service. Acesso: 03-03-2016
- [20] “NABD website” <https://www.nabdsys.com/>. Acesso: 07-03-2016
- [21] “NABD: Complaint Management Software”
<https://www.nabdsys.com/complaint-management-software.html>. Acesso: 07-03-2016
- [22] “Pipedrive” <https://www.pipedrive.com/pt>. Acesso: 10-03-2016
- [23] “Pipedrive API” <https://developers.pipedrive.com/v1>. Acesso: 10-03-2016
- [24] “Digicert: What is a SSL” <https://www.digicert.com/ssl.htm>. Acesso: 10-03-2016
- [25] “Gatewit Supplier Information Management” <https://sim.gatewit.com/> Acesso: 20-03-2016
- [26] “Gatewit: Auctions” <https://auctions.gatewit.com/>. Acesso 20-03-2016
- [27] “Gatekeeper” <https://www.gatekeeperhq.com/vendor-management-software>
Acesso: 20-03-2016
- [28] “InvisionApp” <https://www.invisionapp.com>. Acesso: 20-03-2016

- [29]“Ten Usability Heuristics” <https://www.nngroup.com/articles/ten-usability-heuristics/>. Acesso: 25-03-2016.
- [30]“PHP” <http://php.net/>. Acesso: 14-04-2016
- [31]“Laravel” <https://laravel.com/>. Acesso: 14-04-2016
- [32]“Composer” <https://getcomposer.org/>. Acesso: 14-04-2016
- [33]“PHP Unit” <https://phpunit.de/>. Acesso: 14-04-2016
- [34]“Wikipédia: HTML” <https://pt.wikipedia.org/wiki/HTML>. Acesso: 14-04-2016
- [35]“JavaScript” <https://developer.mozilla.org/pt-PT/docs/Web/JavaScript>. Acesso: 14-04-2016
- [36]“Learn JavaScript” <https://www.javascript.com/>. Acesso: 14-04-2016
- [37]“Blade Templates” <https://laravel.com/docs/5.3/blade>. Acesso: 15-04-2016
- [38]“Eloquent” <https://laravel.com/docs/5.3/eloquent>. Acesso: 15-04-2016
- [39]“ORM: Object Relational Mapping” https://en.wikipedia.org/wiki/Object-relational_mapping. Acesso: 15-04-2016
- [40]“Wikipédia: SQL” <https://pt.wikipedia.org/wiki/SQL>. Acesso: 15-04-2016
- [41]“xUnit” <https://en.wikipedia.org/wiki/XUnit>. Acesso: 15-04-2016
- [42]“MySQL” <https://www.mysql.com/>. Acesso: 16-04-2016
- [43]“Google Authenticator with PHP” <http://hayageek.com/two-factor-authentication-with-google-authenticator-php/>. Acesso: 30-03-2016
- [44]“Google Authenticator” https://en.wikipedia.org/wiki/Google_Authenticator. Acesso: 30-03-2016
- [45]“Dropzone.js” <http://www.dropzonejs.com/> Acesso: 30-03-2016
- [46]“Chart.js” <http://www.chartjs.org/>. Acesso: 30-03-2016
- [47]“Moscow method” https://en.wikipedia.org/wiki/MoSCoW_method. Acesso: 15-03-2016
- [48]“SQL Injection” http://www.w3schools.com/sql/sql_injection.asp. Acesso: 10-10-2016
- [49]“Cross-Site Scripting (XSS)” [https://www.owasp.org/index.php/Cross-site_Scripting_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)). Acesso: 10-10-2016
- [50]“Cross-Site Request Forgery” [https://www.owasp.org/index.php/Cross-Site_Request_Forgery_\(CSRF\)](https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)). Acesso: 10-10-2016
- [51]“Response time limits: Article by jakob nielsen.” <https://www.nngroup.com/articles/response-times-3-important-limits/>. Acesso: 25-03-2016.
- [52]“Wikipédia: Use Case Diagram” https://en.wikipedia.org/wiki/Use_case_diagram. Acesso: 30-03-2016
- [53]“AngularJS HTML DOM” http://www.tutorialspoint.com/angularjs/angularjs_html_dom.htm. Acesso: 21-03-2016
- [54]“Angular vs React vs Ember vs Backbone” <https://gist.github.com/makmanalp/9b7f50aa16d21c2f9d37>. Acesso: 21-03-2016
- [55] “Codeigniter” <http://www.codeigniter.com/userguide3/tutorial/>. 21-03-2016
- [56]“Django” <https://www.djangoproject.com/> Acesso: 21-03-2016
- [57]“Ruby on Rails” <http://rubyonrails.org/>. Acesso: 21-03-2016
- [58]“Microsoft .NET” https://pt.wikipedia.org/wiki/Microsoft_.NET Acesso: 21-03-2016
- [59]“C#” <https://msdn.microsoft.com/en-us/library/67ef8sbd.aspx> Acesso: 21-03-2016
- [60] “Struts 2.0 and JPA integration” <http://javabeat.net/struts-2-0-and-jpa-integration/> Acesso: 28-03-2016
- [61]“Spring” <https://spring.io/> Acesso: 28-03-2016

- [62]“Ajax Request” http://www.w3schools.com/xml/ajax_intro.asp. Acesso: 24-04-2016
- [63]“Laracasts” <https://laracasts.com/>. Acesso: 30-10-2016
- [64]“Lavacharts” <http://lavacharts.com/>. Acesso: 25-06-2016
- [65]“Cascading style sheets.” <https://www.w3.org/Style/CSS/>. Acedido: 24-08-2016
- [66]“What is SQL Injection?” <http://www.acunetix.com/websecurity/sql-injection/>. Acesso: 28-11-2016
- [67]“What is CSRF and XSRF?”
<http://searchsoftwarequality.techtarget.com/definition/cross-site-request-forgery>. Acesso: 28-11-2016
- [68]“Software maintainability” <http://www.castsoftware.com/glossary/software-maintainability>. Acesso: 4-10-2016
- [69]“How to handle demanding clients”
<http://softwareengineering.stackexchange.com/questions/5415/how-to-handle-demanding-clients>. Acesso: 16-09-2016
- [70]“Why complaints are good for business?”
<http://www.superoffice.com/blog/customer-complaints-good-for-business/>. Acesso: 03-05-2016
- [71]“Benefits of customer complaints” <http://thethrivingsmallbusiness.com/what-are-the-5-benefits-of-customer-complaints/>. Acesso: 04-05-2016
- [72]“How to write a report” <https://www.cse.iitk.ac.in/users/braman/students/good-report.html>. Acesso: 06-05-2016
- [73]“What is the importance of statistics in business?”
<https://www.reference.com/math/importance-statistics-business-7bb737fb7b8cf1f1>. Acesso: 08-05-2016

Anexo A – Levantamento de Requisitos

MC-IR: Inserir reclamação

ID	Nome	Descrição	Dep.	Nivel
IR-1	Preencher formulário de reclamação	Funcionalidade que permite preencher e validar o formulário da reclamação		3
IR-2	Selecionar cliente	Funcionalidade que permita selecionar determinado cliente e carregar part-numbers associados a ele do IFS	-	3
IR-3	Selecionar part-number	Funcionalidade que permite selecionar determinado part-number, filtrado por cliente e preencher dados do formulário automaticamente	IR-2	3
IR-4	Inserir nova linha	Funcionalidade que permite inserir nova linha na reclamação	-	3
IR-5	Cálculo automático de custos	Funcionalidade que obtém informação do IFS e apresenta cálculos de custos automaticamente de acordo com essa informação e com os dados da nova linha	IR-4	3
IR-6	Enviar mail para setor	Funcionalidade que envia um mail ao setor responsável pelo produto a ser reclamado	IR-1	1
IR-7	Controlo do identificador numérico de reclamações	Funcionalidade que permite a gestão automática da numeração de reclamações	-	3

Tabela 3.2 - Tabela de requisitos inserir reclamação

IR-1: Preencher formulário de reclamação

Este requisito tem como objetivo fornecer ao utilizador um formulário apropriado para o preenchimento da reclamação. Também deverá ter as respetivas validações.

IR-2: Selecionar cliente

Este requisito tem como objetivo selecionar os clientes de determinado responsável, ou seja, um responsável autenticado terá uma lista de clientes só sua. Esta seleção irá automaticamente carregar os part-numbers associados a esse cliente (dados que vêm do IFS).

IR-3: Selecionar part-number

Este requisito tem como objetivo selecionar determinado part-number, ou seja, deverá filtrar todos os part-numbers de um cliente X e preencher os restantes dados do formulário de forma automática.

IR-4: Inserir nova linha

Este requisito tem como objetivo a inserção de uma nova linha (part-number reclamado) em determinada reclamação. Note que uma reclamação pode ter vários part- numbers reclamados.

IR-5: Cálculo automático de custos

Este requisito tem como objetivo o preenchimento automático de campos relativos a determinado part-number e cálculo de custos automático de acordo com essa informação e com os dados da nova linha

IR-6: Enviar mail para setor

Este requisito tem como objetivo o envio de um mail para o setor responsável da reclamação.

IR-7: Controlo do identificador numérico de reclamações

Funcionalidade que permite a gestão automática da numeração de reclamações.

Como exemplo:

- Se a reclamação for do tipo QUA (Qualidade) deverá ficar com o número 001
- Se uma segunda reclamação for do mesmo tipo, deverá ficar com o número 002
- Caso chegue uma terceira com tipo LOG (Logística), deverá ficar com o número 001
- Se chegar nova reclamação do tipo QUA (Qualidade), deverá ficar com o número 003

MC-AR: Consultar Reclamação

ID	Nome	Descrição	Dep.	Nível
AR-1	Carregar reclamação	Funcionalidade que permite abrir e analisar uma reclamação	-	3
AR-2	Atualizar progresso	Funcionalidade que permite atualizar o progresso de determinada reclamação. Deverá ser guardado um histórico de cada alteração	-	3
AR-3	Atualização de linha	Funcionalidade que permite atualizar uma linha da reclamação.	-	3
AR-4	Cálculo automático de alterações	Funcionalidade que permite calcular, de acordo com as alterações, os custos da reclamação	AR-3	3
AR-5	Inserir nova linha	Funcionalidade que permite adicionar nova linha à reclamação preenchendo o formulário.	IR-1, IR-2, IR-3, IR-4, IR-5	3
AR-6	Exportar reclamação	Funcionalidade que permite exportar a reclamação para excel	-	3

Tabela E.2 - Tabela de requisitos análise de reclamações

AR-1: Analisar Reclamação

Este requisito tem como objetivo permitir ao utilizador abrir e mostrar o conteúdo de uma reclamação de forma fácil e simples de analisar. A abertura do conteúdo deverá ser apresentada em modo editável.

AR-2: Atualizar progresso

Este requisito tem como objetivo atualizar o progresso de uma determinada reclamação e guarda um histórico. No ato de atualização deverá apresentar esse histórico.

AR-3: Atualização da linha

Este requisito tem como objetivo atualizar uma determinada linha que tenha sido inserida anteriormente. O utilizador poderá alterar apenas alguns campos.

AR-4: Cálculo automático de alterações

Este requisito tem como objetivo calcular automaticamente os campos necessários de acordo com as alterações que forem efetuadas em AR-3.

AR-5: Inserir nova linha

Este requisito tem como objetivo inserir uma nova linha na reclamação. Envolve os requisitos IR-1, IR-2, IR-3, IR-4 e IR-5

AR-6: Exportar Reclamação

Este requisito tem como objetivo exportar uma reclamação para excel. O programa deverá processar os dados, utilizar um template e preencher o mesmo com os dados!

MC-CHR: Consultar histórico de reclamações

ID	Nome	Descrição	Dep.	Nivel
HR-1	Consultar histórico	Funcionalidade que permite ao utilizador ver o histórico de reclamações	-	3
HR-2	Filtrar reclamação	Esta função irá permitir filtrar o histórico de reclamações por diversos tipos	HR-1	3

Tabela E.3 – Tabela de requisitos histórico de reclamações

HR-1: Consultar histórico

Este requisito tem como objetivo fornecer ao utilizador um histórico de reclamações registadas na base de dados.

HR-2: Filtrar reclamação

Este requisito tem como objetivo fornecer um modo de filtragem de reclamações. Os filtros devem permitir filtrar por estado de reclamação, datas, entre outros.

MC-E: Estatísticas

ID	Nome	Descrição	Dep.	Nível
E-1	Mostrar top clientes reclamantes	Permite mostrar o top de clientes reclamantes da empresa	-	2
E-2	Mostrar top produtos reclamados	Permite mostrar ao utilizador o top de produtos reclamados	-	2
E-3	Mostrar top defeitos	Permite mostrar ao utilizador o top de defeitos em reclamações	-	2
E-4	Contagem problemas p/ mes	Permite mostrar ao utilizador a contagem de reclamações por mes	-	2
E-5	Custos reclamações mensais	Permite mostrar ao utilizador o custo de todas as reclamações, por mes	-	3
E-6	Mostrar top de custos clientes	Permite mostrar o top dos custos que cada reclamação teve por cliente	-	3
E-7	Mostrar top de custos defeitos	Permite mostrar o top dos custos que cada defeito teve	-	3
E-8	Mostrar top de custos produtos	Permite mostrar o top dos custos que cada produto teve	-	3

Tabela E.5 - Tabela de requisitos mostrar estatísticas

E-1: Mostrar top 10 clientes

Este requisito permite ao utilizador obter o top de clientes reclamantes na aplicação, ou seja, os clientes que mais reclamam.

E-2: Mostrar top 10 produtos

Este requisito permite ao utilizador obter o top de produtos reclamados na aplicação, ou seja, os produtos mais reclamados.

E-3: Mostrar top 10 defeitos

Este requisito permite ao utilizador obter o top de defeitos registados nas reclamações na aplicação, ou seja, os defeitos envolvidos em mais reclamações.

E-4: Contagem problemas p/ mes

Este requisito permite ao utilizador obter graficamente um conjunto de meses e suas contagens de defeitos por cada um. A contagem deverá ser o total dos defeitos.

E-5: Custos reclamações mensais

Este requisito permite ao utilizador obter graficamente um conjunto de meses e a soma dos custos de cada reclamação ao longo de cada mês.

E-6: Mostrar top custos de clientes

Este requisito permite ao utilizador obter graficamente os custos que cada cliente deu à empresa.

E-7: Mostrar top custos de defeitos

Este requisito permite ao utilizador obter graficamente os custos que cada defeito deu à empresa.

E-8: Mostrar top custos de produtos

Este requisito permite ao utilizador obter graficamente os custos que cada produto deu à empresa.

MC-RC: Reembolsar Cliente

ID	Nome	Descrição	Dep.	Nível
RC-1	Pedir autorização de nota de crédito	Permite enviar a ficha de pedido de reembolso para o chefe(s) responsável(eis) autorizar(em)	-	3
RC-2	Anexar documento	Funcionalidade que permite anexar um documento	-	2
RC-3	Enviar mail aviso	Permite enviar um aviso ao destinatário de que existe uma nota de crédito para aprovar e, que deve aceder ao portal	RC-1	2
RC-4	Validar nota de crédito	Funcionalidade que irá permitir aos responsáveis validar a nota de crédito.	RC-1	3
RC-5	Validação de Nota de crédito pelo Dep. Financeiro	Funcionalidade que irá permitir, após aprovada a nota de crédito, reembolsar o cliente.	RC-1, RC-4	3
RC-6	Geração de código secreto	Funcionalidade que gera código secreto para cada utilizador responsável para aprovar um documento	RC-1	3
RC-7	Envio de código secreto via sms	Funcionalidade que permite enviar o código secreto via sms ao responsável	RC-6	3

Tabela 3.3 - Tabela de requisitos reembolsar cliente

RC-1: Pedir autorização de nota de crédito

Este requisito permite, de acordo com os valores da reclamação, gerar e enviar um pedido de autorização de uma nota de crédito aos responsáveis. Note-se que como é o Gestor de Qualidade que faz estes pedidos, o sistema assume desde logo a aprovação desse responsável. Quando os valores são:

- <2.500€, o pedido deverá ser aprovado pelo Gestor de Qualidade e pelo chefe de departamento de qualidade;
- >5.000€, o pedido deverá ser aprovado pelos Gestor de Qualidade, chefe de departamento de qualidade, *Chief Operational Officer Europe* (COO Europe) e *Chief Executive Officer* (CEO).

Nota que uma reclamação pode ter várias notas de crédito assim como uma nota de crédito pode ser referente a várias reclamação do mesmo cliente.

RC-2: Anexar fatura desvio

Este requisito permite ao utilizador a funcionalidade de anexar uma fatura desvio à reclamação presente.

RC-3: Enviar mail aviso

Este requisito deverá permitir a geração de um mail que será enviado para o responsável em questão. Este, deverá alertar de que existe uma nova nota de crédito ou fatura desvio no sistema para ser aprovada.

RC-4: Validar nota de crédito

Este requisito tem como objetivo validar uma nota de crédito. O responsável irá receber no seu mail um link para o porta, caso o responsável queira dar autorização para a passagem da nota de crédito (NC), terá de colocar o código do google authenticator e, realizar a autenticação com recurso a essa API. Caso seja aceite, a autorização será dada e registada.

RC-5: Validação de Nota de crédito pelo Dep. Financeiro

Este requisito tem como objetivo, após a validação da nota de crédito em RC-4, reembolsar o cliente. Após a aprovação da nota de crédito, esta irá ser enviada para o Dep. Financeiro, onde irá ser realizado todo o processo de reembolso.

RC-6: Geração de código secreto

Este requisito tem como objetivo a geração de um código secreto que servirá para o utilizador responsável valide o documento que foi gerado em RC-1.

RC-7: Envio de código secreto via sms

Este requisito tem como objetivo o envio de uma sms para o responsável de validação de documentos, contendo o código secreto gerado em RC-6.

MC-AR: Ações para reclamações

ID	Nome	Descrição	Dep.	Nível
AR-1	Atribuir clientes a funcionário	Permite ao chefe de qualidade atribuir clientes a funcionários. Cada cliente terá um funcionário responsável.	-	3
AR-2	Obter informação sobre reclamação	Permite obter informação acerca de quantidade de reclamações abertas, fechadas, entre outros	-	2
AR-3	Atribuir responsáveis para validação de documentação	Permite identificar quais serão os responsáveis para aprovar a NC	RC-1	3
AR-4	Atribuir financeiro responsável para passagem de nota de crédito	Permite identificar quais os funcionários que serão responsáveis por finalizar o processo de reembolso.	RC-1	3

AR-5	Atribuir fornecedores a funcionário	Permite ao responsável de fornecedores atribuir esta entidade a funcionários. Cada fornecedor terá um funcionário responsável.	-	3
-------------	-------------------------------------	--	---	---

Tabela 3.3 - Tabela de requisitos reembolsar cliente

AR-1: Atribuir clientes a funcionário

Funcionalidade que irá obter todos os funcionários da empresa e permitir ao chefe de qualidade atribuir um cliente a um funcionário.

AR-2: Obter informação sobre reclamação

Este requisito permite ao utilizador obter, por exemplo, a quantidade de reclamações em aberto, fechadas, entre outros. É um requisito que apresenta dados de rápida análise.

AR-3: Atribuir responsáveis para validação de documentação

Funcionalidade que irá obter todos os funcionários da empresa e permitir ao administrador identificar os responsáveis pela validação de documentação.

AR-4: Atribuir financeiro responsável para passagem de nota de crédito

Funcionalidade que irá obter todos os funcionários da empresa e permitir ao administrador identificar os responsáveis pela finalização do ciclo de vida de determinada nota de crédito.

AR-5: Validação de Nota de crédito pelo Dep. Financeiro

Este requisito tem como objetivo, após a validação da nota de crédito em RC-4, reembolsar o cliente. Após a aprovação da nota de crédito, esta irá ser enviada para o Dep. Financeiro, onde irá ser realizado todo o processo de reembolso.

AR-6: Atribuir fornecedores a funcionário

Funcionalidade que irá obter todos os funcionários da empresa e permitir ao responsável por estes atribuir um fornecedor aos mesmos.

MC-IS: Integração do Seval

ID	Nome	Descrição	Dep.	Nivel
SE-1	Consultar reclamação	Funcionalidade que permite consultar uma reclamação utilizando uma vista do Seval	-	2
SE-2	Obter avaliações	Funcionalidade que permite obter dados relativos ao fornecedor que informem acerca da sua avaliação interna	-	1

Tabela 3.3 - Tabela de requisitos reembolsar cliente

AR-1: Atribuir clientes a funcionário

Funcionalidade que irá obter todos os funcionários da empresa e permitir ao chefe de qualidade atribuir um cliente a um funcionário.

PA-IPA: Inserir Plano de Ação

ID	Nome	Descrição	Dep.	Nivel
IPA-1	Criar formulário	Funcionalidade que permite criar um novo plano de ação		3
IPA-2	Preencher dados automaticamente	Funcionalidade que permite preencher automaticamente campos do formulário	-	3
IPA -3	Obter funcionários da empresa	Funcionalidade que permite obter todos os funcionários da empresa para atribuição de tarefas	-	3
IPA-4	Inserir funcionários externos	Funcionalidade que permite adicionar funcionários externos ao plano de ação	-	3

Tabela 3.2 - Tabela de requisitos inserir reclamação

IPA-1: Criar formulário

Este requisito tem como objetivo criar um plano de ação para determinada reclamação. O formulário a utilizar deverá ser sempre o da empresa. Só assim é possível gerir e garantir todos os dados necessários.

IPA -2: Preencher dados automaticamente

Este requisito tem como objetivo o preenchimento automático de alguns campos do formulário.

IPA -3: Obter funcionários da empresa

Este requisito tem como objetivo obter os funcionários da empresa. No plano de ação é necessário indicar ações de como solucionar o problema e, cada ação é atribuída a um funcionário.

IPA -4: Inserir funcionários externos

Este requisito tem como objetivo inserir os funcionários externos, ou seja, de outras empresa. No plano de ação é necessário indicar ações de como solucionar o problema e, cada ação é atribuída a um funcionário.

PA-CPA: Consultar Plano de Ação

ID	Nome	Descrição	Dep.	Nível
CPA-1	Abrir plano de ação	Funcionalidade que permite carregar e mostrar os dados de determinado plano de ação	-	3
CPA-2	Editar plano de ação	Funcionalidade que permite editar um plano de ação	-	3
CPA-3	Exportar plano de ação	Funcionalidade que permite exportar o plano de ação para excel	-	3
CPA-4	Fechar plano de ação	Funcionalidade que permite fechar um plano de ação	-	3
CPA-5	Anexar ficheiros	Funcionalidade que permite a anexação de ficheiros no plano de ação	-	3
CPA-6	Concluir tarefa	Funcionalidade que permite indicar que determinada tarefa se encontra concluída	-	3

Tabela 3.2 - Tabela de requisitos inserir reclamação

CPA-1: Abrir plano de ação

Este requisito tem como objetivo carregar e mostrar os dados guardados de um plano de ação referente a determinada reclamação

CPA-2: Editar plano de ação

Este requisito tem como objetivo editar um plano de ação e guardar as informações alteradas ou acrescentadas.

CPA-3: Exportar plano de ação

Este requisito tem como objetivo exportar para um ficheiro excel um plano de ação. O ficheiro excel deverá ser no formato próprio da Borgstena

CPA-4: Fechar plano de ação

Este requisito tem como objetivo fechar um plano de ação. O fechar de um plano de ação indica que todas as tarefas foram executadas e que “não existem” mais procedimentos para corrigir o problema.

CPA-5: Anexar ficheiros

Este requisito tem como objetivo anexar ficheiros a um plano de ação. Existe um grande número de clientes que envia à Borgstena um plano de ação num formato diferente daquele

utilizado. Assim, é importante anexar o ficheiro por eles enviados para não serem perdidos dados que possam vir a ser importantes.

CPA-6: Concluir tarefa

Este requisito tem como objetivo indicar que a tarefa indicada para o utilizador está concluída.

PA-NPA: Notificações de Plano de Ação

ID	Nome	Descrição	Dep.	Nível
NPA-1	Notificar vencimentos	Funcionalidade que permite notificar ao utilizar que determinado plano de ação se encontra perto da data de expirar.	-	2
NPA-2	Obter calendário de datas de expiração	Funcionalidade que permite mostrar ao utilizador um calendário com datas de expiração de um plano de ação	-	3

Tabela 3.2 - Tabela de requisitos inserir reclamação

NPA-1: Notificar vencimentos

Este requisito tem como objetivo notificar os utilizadores que um plano de ação se encontra a 7 ou menos dias de expirar

NPA-2: Editar plano de ação

Este requisito tem como objetivo obter um calendário que indica quantos e quando é que os planos de ação expiram

MF-RC: Registrar contas

ID	Nome	Módulo	Descrição	Dep.	Nível
RC-1	Registrar fornecedor	-	O portal deverá permitir ao administrador registrar os fornecedores que poderão ter acesso ao portal	-	2
RC-2	Enviar mail com credenciais	-	Deverá ser enviado um mail com as credenciais dos fornecedores	RC-1	2

Tabela E.1 - Tabela de requisitos registrar contas

S-OC-1: Registrar fornecedor

Este requisito tem como objetivo permitir ao administrador do sistema acrescentar fornecedores manualmente

S-OC-2: Enviar mail com credenciais

Este requisito tem como objetivo permitir ao sistema enviar um mail ao utilizador registado em S-OC-1 com as credenciais de acesso.

MF-MR: Mostrar reclamações

ID	Nome	Módulo	Descrição	Dep.	Nível
MR-1	Consultar reclamações	-	Funcionalidade que permite mostrar as reclamações que a empresa fez a ele	-	3
MR-2	Inserir plano de ação	-	Funcionalidade que permite ao fornecedor preencher o formulário 8D no formato da empresa	MR-2	1

Tabela E.1 - Tabela de requisitos registrar contas

MR-1: Consultar reclamações

Este requisito tem como objetivo consultar reclamações que a Borgstena tenha feito ao fornecedor.

MR-2: Inserir plano de ação

Este requisito tem como objetivo permitir ao fornecedor preencher o plano de ação no formato da Borgstena.

MF-OC: Ordens de compra

ID	Nome	Descrição	Dep.	Nível
OC-1	Obter dados do IFS	Funcionalidade que permite ao sistema receber dados do IFS	-	3
OC-3	Enviar mail a fornecedor	Funcionalidade que envia um mail ao fornecedor a avisar que existe uma nova ordem de compra a verificar	OC-2	1
OC-4	Atualizar ordem de compra	Funcionalidade que permite atualizar ordem de compra	OC-1	3
OC-5	Abrir ordem de compra	Funcionalidade que permite abrir uma ordem de compra	-	3
OC-6	Ver lista de ordens de compra	Funcionalidade que permite ver a lista de ordens de compra	-	3

Tabela 3.5 - Tabela de requisitos ordens de compra

OC-1: Obter dados do IFS

Este requisito permite ao sistema esperar e receber eventos despoletados no IFS para obter dados relativos a uma nova ordem de compra que tenha sido lá inserida. Este requisito deverá ocorrer do lado do IFS sempre que uma ordem de compra é inserida, atualizada ou removida e, o sistema receberá o evento e tratá-lo-á de forma a fornecer informação atualizada ao fornecedor.

OC-2: Inserir nova ordem de compra

Este requisito permite ao sistema inserir a ordem de compra lida no requisito F-OC-1, na base de dados.

OC-3: Enviar mail a fornecedor

Este requisito permite ao sistema criar, estruturar e enviar um mail automático ao fornecedor envolvido na ordem de compra, alertando o mesmo para consultar o portal pois existe uma encomenda para o mesmo.

OC-4: Atualizar ordem de compra

Este requisito permite ao sistema atualizar a ordem de compra lida no requisito F-OC-1, na base de dados. Deverá executar o requisito F-OC-3 alertando o fornecedor que uma ordem de compra foi alterada.

OC-5: Abrir ordem de compra

Este requisito permite ao fornecedor ou ao utilizador interno da Borgstena abrir e ver o conteúdo de uma ordem de compra.

F-OC-6: Ver lista de ordens de compra

Este requisito permite ao fornecedor ou ao utilizador interno da Borgstena ver uma lista com todas as ordens de compra. No caso do fornecedor, este só verá ordens de compra que foram feitas ao próprio.

MF-AOC: Ações em Ordens de Compra

ID	Nome	Descrição	Dep.	Nível
OC-1	Confirmar item de ordem de compra	Funcionalidade que permite ao fornecedor confirmar um item da ordem de compra	-	3
OC-2	Sugerir nova data de item	Funcionalidade que permite ao fornecedor indicar a data possível de entrega de encomenda	-	3
OC-3	Rejeitar pedido	Funcionalidade que permite ao fornecedor rejeitar um item da ordem de compra	-	3
OC-4	Inserir justificação	Funcionalidade que obriga o fornecedor a justificar o porque da rejeição do item da ordem de compra	OC-3	3

Tabela E.7 - Tabela de requisitos ações em ordem de compra

OC-1: Confirmar item de ordem de compra

Este requisito permite ao fornecedor confirmar a entrega de um dos itens presentes na ordem de compra.

OC-2: Sugerir nova data de item

Este requisito permite ao fornecedor sugerir uma nova data de entrega, diferente da pedida pelo utilizador interno da Borgstena. Esta data pode ser ou não aceite por parte da empresa.

OC-3: Rejeitar pedido

Este requisito permite ao fornecedor rejeitar o pedido de um item presente na ordem de compra.

OC-4: Inserir justificação

Este requisito obriga o fornecedor a justificar o porque do cancelamento de um dos itens pedidos na ordem de compra, requisito F-AOC-3.

MF-VE: Ver estatísticas

ID	Nome	Descrição	Dep.	Nível
VE-1	Ver histórico de encomendas	Funcionalidade que permite ao fornecedor ver um histórico de encomendas	-	3
VE-2	Ver avaliação mensal	Funcionalidade que permite ao fornecedor verificar como está avaliado na borgstena de forma mensal	-	3
VE-3	Ver avaliação semestral	Funcionalidade que permite ao fornecedor verificar como está avaliado na borgstena de forma semestral	-	3
VE-4	Ver avaliação anual	Funcionalidade que permite ao fornecedor verificar como está avaliado na borgstena de forma anual	-	3

Tabela E.6 - Tabela de requisitos ver estatísticas

VE-1: Ver histórico de encomendas

Este requisito permite ao fornecedor ver um histórico das encomendas que a Borgstena tenha feito aos mesmos.

VE-2: Ver avaliação mensal

Este requisito permite ao fornecedor verificar como está avaliado de forma mensal na Borgstena.

VE-3: Ver avaliação semestral

Este requisito permite ao fornecedor verificar como está avaliado de forma semestral na Borgstena.

VE-3: Ver avaliação anual

Este requisito permite ao fornecedor verificar como está avaliado de forma anual na Borgstena.

MF-VA: Ver alterações

ID	Nome	Descrição	Dep.	Nível
VE-1	Listar encomendas	Funcionalidade que permite ao fornecedor ver um histórico de encomendas	-	3
VE-2	Abrir encomenda	Funcionalidade que permite ao fornecedor verificar como está avaliado na borgstena de forma mensal	-	3

Tabela E.6 - Tabela de requisitos ver estatísticas

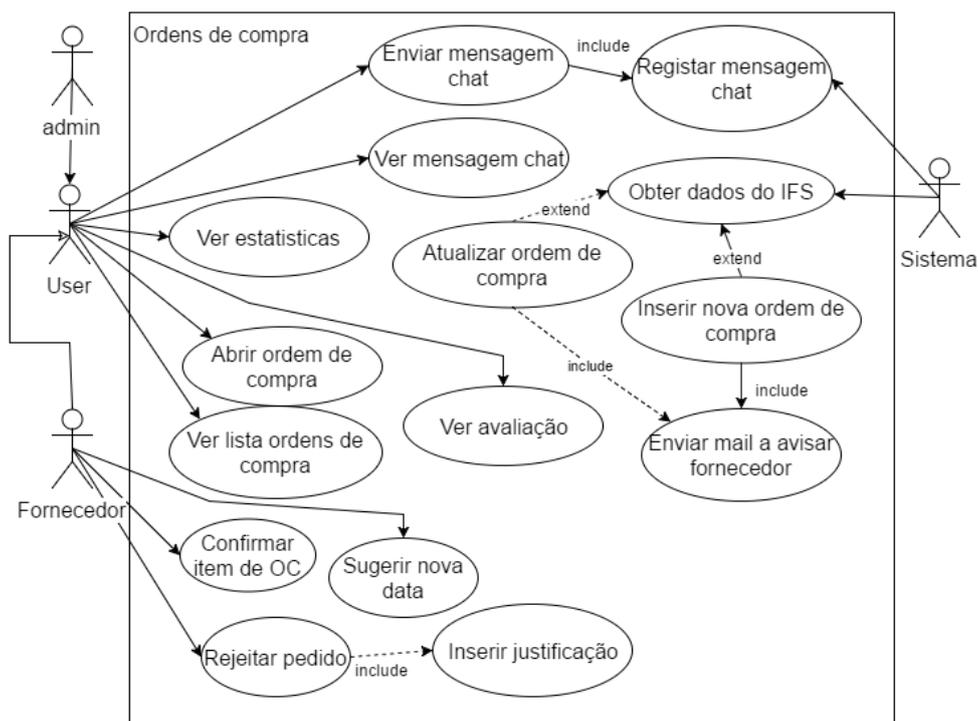
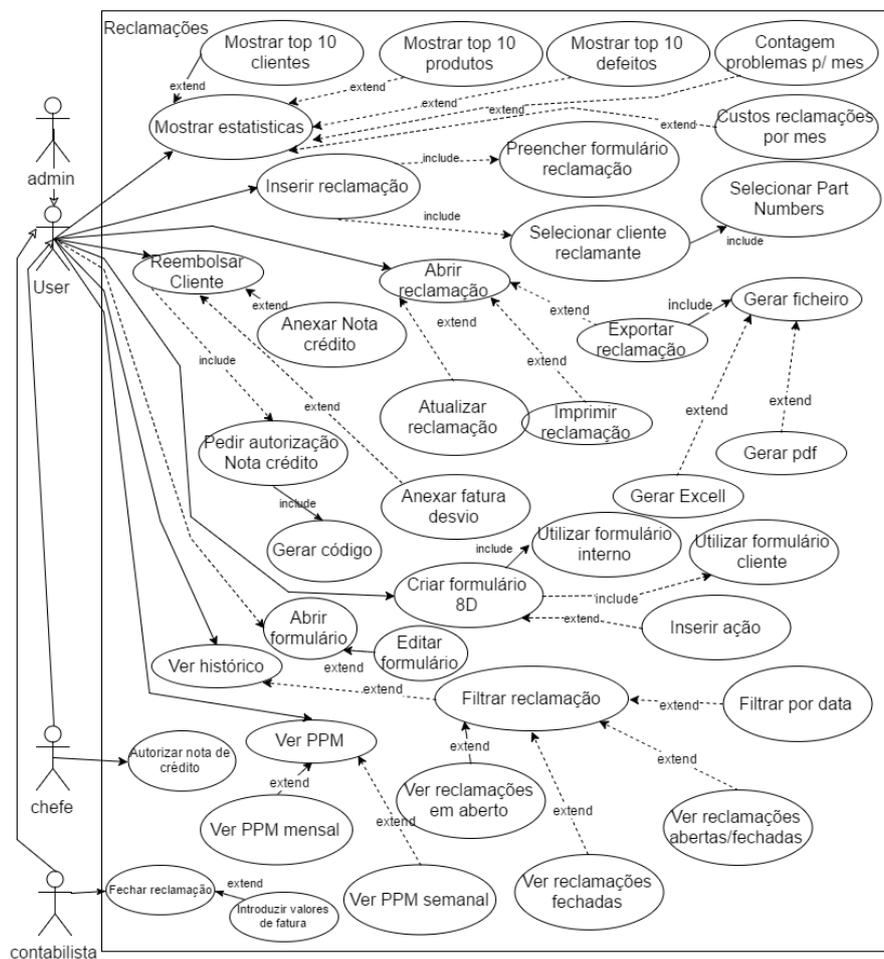
VE-1: Listar encomendas

Este requisito permite ao utilizador ter uma lista de encomendas que foram alteradas

VE-2: Abrir encomenda

Este requisito permite ao utilizador abrir e ver as alterações na encomenda.

Anexo B – Diagramas de casos de uso



Anexo C – Test cases

ID	Requisito(s) relacionado(s)	Resultado
CT-0	MC-IR: IR-1, IR-2, IR-3, IR-4, IR-5, IR-7	Aprovado
Descrição:	O objetivo deste teste é introduzir uma reclamação no sistema com 1 part-number a ser reclamado e seu devido defeito. Nota que o defeito é o último input, sendo representado pelo seu código.	
Input	“AS Wendre”, “100110997”, “comp 001 testing”, “Qua”, “BTP”, “12-12-2016”, “P510271101/1”, “2”	
Output Esperado	<ul style="list-style-type: none"> • P510271101/1 • P510281101/1 • SEVERAL 	
Output Real	<ul style="list-style-type: none"> • P510271101/1 • P510281101/1 • SEVERAL 	

ID	Requisito(s) relacionado(s)	Resultado
CT-1	MC-IR: IR-1, IR-2, IR-3, IR-4, IR-5, IR-7	Aprovado
Descrição:	O objetivo deste teste é introduzir uma reclamação no sistema com 1 part-number a ser reclamado e seu devido defeito. Nota que o defeito é o último input, sendo representado pelo seu código.	
Passos	<ol style="list-style-type: none"> 1. Clicar em “<i>My Complaints</i>” 2. Escolher a opção “<i>Create New</i>” 3. Preencher dados 	
Input	<ul style="list-style-type: none"> • Customer name: “AS Wendre” • Customer code: “100110997” • Complaint number: “comp 001 testing” • Type: “Qua” • Company: “BTP” • Arrival Date: “12-12-2016” 	
Output Esperado	<ul style="list-style-type: none"> • P510271101/1 • P510281101/1 • SEVERAL 	
Output Real	<ul style="list-style-type: none"> • P510271101/1 • P510281101/1 • SEVERAL 	

ID	Requisito(s) relacionado(s)	Resultado
CT-2	MC-IR: IR-1, IR-2	Aprovado
Descrição	O objetivo deste teste é selecionar um cliente que não exista no sistema e como output a aplicação deverá informar o utilizador de erro e não deverá permitir avançar na introdução da reclamação.	
Passos	<ol style="list-style-type: none"> 4. Clicar em <i>"My Complaints"</i> 5. Escolher a opção <i>"Create New"</i> 6. Preencher dados 	
Input	<ul style="list-style-type: none"> • Customer name: "TESTING" • Customer code: "9999AAAA" • Complaint number: "comp 001 testing" • Type: "Qua" • Company: "BTP" • Arrival Date: "12-12-2016" 	
Output Esperado	Mensagem de erro: "The customer is not correct!"	
Output Real	Mensagem de erro: "The customer is not correct!"	

ID	Requisito(s) relacionado(s)	Resultado
CT-3	MC-IR	Aprovado
Descrição	O objetivo deste teste é introduzir part-numbers a reclamar e submeter a reclamação para o sistema.	
Passos	<ol style="list-style-type: none"> 1. Clicar em <i>"My Complaints"</i> 2. Escolher a opção <i>"Create New"</i> 3. Preencher dados com o Input do Caso de teste CT-1 4. Clicar em "Next Step" 5. Clicar em "Add new item" 	
Input	<ul style="list-style-type: none"> • Part-Number: "P510271101/1" • Quantity: "10" • Defect: "Riscas /stripes" 	
Output Esperado	A reclamação irá aparecer com sucesso no sistema	
Output Real	A reclamação foi inserido com sucesso	

ID	Requisito(s) relacionado(s)	Resultado
CT-4	MC-IR	Aprovado
Descrição	O objetivo deste teste é introduzir valores inválidos nos campos e a aplicação indicar os campos errados	
Passos	2. Clicar em <i>“My Complaints”</i> <ul style="list-style-type: none"> • Escolher a opção <i>“Create New”</i> • Preencher dados com o Input do Caso de teste CT-1 • Clicar em <i>“Next Step”</i> • Clicar em <i>“Add new item”</i> 	
Input	<ul style="list-style-type: none"> • Part-Number: <i>“P510271101/1”</i> • Quantity: <i>“AAA”</i> • Defect: <i>“Riscas /stripes”</i> • Issue Cost: <i>“AAA”</i> • Scrapping Cost: <i>“AAA”</i> • Logistic Cost: <i>“AAA”</i> • Extra Cost: <i>“AAA”</i> • Savings Cost: <i>“AAA”</i> 	
Output Esperado	O sistema deverá dar erro nos campos mal preenchidos	
Output Real	O sistema reportou os erros corretamente	

ID	Requisito(s) relacionado(s)	Resultado
CT-5	MC-IR	Reprovado
Descrição	O objetivo deste teste é introduzir valores inválidos nos campos e a aplicação indicar os campos errados	
Passos	<ol style="list-style-type: none"> 1. Clicar em <i>"My Complaints"</i> 2. Escolher a opção <i>"Create New"</i> 3. Preencher dados com o Input do Caso de teste CT-1 4. Clicar em "Next Step" 5. Clicar em "Add new item" 	
Input	<ul style="list-style-type: none"> • Part-Number: "P510271101/1" • Quantity: "11,2" • Defect: "Riscas /stripes" • Issue Cost: "11,2" • Scrapping Cost: "11,2" • Logistic Cost: "11,2" • Extra Cost: "11,2" • Savings Cost: "11,2" 	
Output Esperado	O sistema deverá corrigir os números com vírgulas para pontos e aceitar a inserção no sistema	
Output Real	O sistema não corrigiu as vírgulas para pontos.	

ID	Requisito(s) relacionado(s)	Resultado
CT-6	MC-AR: AR-3, AR-4	Reprovado
Descrição	O objetivo deste teste é editar uma reclamação	
Passos	<ol style="list-style-type: none"> 1. Clicar em <i>"My Complaints"</i> 2. Clicar em "Show All" 3. Escolher a reclamação "000-16" 4. Clicar numa das linhas 	
Input	<ul style="list-style-type: none"> • Part-Number: "P510271101/1" • Quantity: "22,2" • Issue Cost: "22,2" • Scrapping Cost: "22,2" • Logistic Cost: "22,2" • Extra Cost: "22,2" • Savings Cost: "22,2" 	
Output Esperado	A reclamação deverá ser guardada com sucesso	
Output Real	O sistema não corrigiu as vírgulas para pontos.	

ID	Requisito(s) relacionado(s)	Resultado
CT-7	MC-AR: AR-6	Aprovado
Descrição	O objetivo deste teste é exportar a reclamação para excel	
Passos	<ol style="list-style-type: none"> 1. Clicar em “<i>My Complaints</i>” 2. Clicar em “Show All” 3. Escolher a reclamação “000-16” 4. Clicar em “Export File” 	
Input	-	
Output Esperado	Ficheiro excel com a reclamação	
Output Real	Ficheiro excel com a reclamação	

ID	Requisito(s) relacionado(s)	Resultado
CT-8	MC-AR: AR-2	Aprovado
Descrição	O objetivo deste teste é atualizar o progresso da reclamação	
Passos	<ol style="list-style-type: none"> 1. Clicar em “<i>My Complaints</i>” 2. Clicar em “Show All” 3. Escolher a reclamação “000-16” 4. Clicar em “Progress” 	
Input	<ul style="list-style-type: none"> • Progress: “Isto é um progresso” 	
Output Esperado	<ul style="list-style-type: none"> • Date: [data atual do teste] • Progress: “Isto é um progresso” 	
Output Real	<ul style="list-style-type: none"> • Date: [data atual do teste] • Progress: “Isto é um progresso” 	

ID	Requisito(s) relacionado(s)	Resultado
CT-9	MC-AR	Aprovado
Descrição	O objetivo deste teste é anexar ficheiros	
Passos	<ol style="list-style-type: none"> 1. Clicar em “<i>My Complaints</i>” 2. Clicar em “Show All” 3. Escolher a reclamação “000-16” 4. Clicar em “Attachment” 5. Arrastar ficheiros 	
Input	<ul style="list-style-type: none"> • Arrastar um ficheiro para a aplicação 	
Output Esperado	<ul style="list-style-type: none"> • Ficheiro carregado com sucesso 	
Output Real	<ul style="list-style-type: none"> • Ficheiro carregado com sucesso 	

ID	Requisito(s) relacionado(s)	Resultado
CT-10	MC-AR: AR-5	Reprovado
Descrição	O objetivo deste teste é inserir uma nova linha na reclamação	
Passos	<ol style="list-style-type: none"> 1. Clicar em “<i>My Complaints</i>” 2. Clicar em “Show All” 3. Escolher a reclamação “000-16” 4. Clicar em “Insert new item” 	
Input	<ul style="list-style-type: none"> • Part-Number: “P510271101/1” • Quantity: “22,2” • Issue Cost: “22,2” • Scrapping Cost: “22,2” • Logistic Cost: “22,2” • Extra Cost: “22,2” • Savings Cost: “22,2” 	
Output Esperado	Reclamação deverá ser guardada com sucesso	
Output Real	O sistema não corrigiu as vírgulas para pontos.	

ID	Requisito(s) relacionado(s)	Resultado
CT-11	PA-IPA	Aprovado
Descrição	O objetivo deste teste é criar um plano de ação	
Passos	<ol style="list-style-type: none"> 1. Clicar em “<i>My Complaints</i>” 1. Clicar em “Show All” 2. Escolher a reclamação “000-16” 3. Clicar em “Add Action Plan” 	
Input	<ul style="list-style-type: none"> • Product: “Teste” • Due date: “07-12-2016” • Date Issue Occured: “01-12-2016” • What is the problem: “TESTE” • Why is it a problem: “TESTE” 	
Output Esperado	<ul style="list-style-type: none"> • Plano de ação inserido com sucesso 	
Output Real	<ul style="list-style-type: none"> • Plano de ação inserido com sucesso 	

ID	Requisito(s) relacionado(s)	Resultado
CT-12	PA-CPA	Aprovado
Descrição	O objetivo deste teste é editar um plano de ação	
Passos	<ol style="list-style-type: none"> 1. Clicar em “<i>My Complaints</i>” 2. Clicar em “Show All” 3. Escolher a reclamação “000-16” 4. Clicar em “Open Action Plan” 	
Input	<ul style="list-style-type: none"> • Product: “AAA” • Due date: “07-12-2016” • Date Issue Occured: “01-12-2016” • What is the problem: “AAA” • Why is it a problem: “AAA” 	
Output Esperado	<ul style="list-style-type: none"> • Plano de ação atualizado com sucesso 	
Output Real	<ul style="list-style-type: none"> • Plano de ação atualizado com sucesso 	

ID	Requisito(s) relacionado(s)	Resultado
CT-13	MC-RC: RC-1, RC-3, RC-6, RC-7, RC-2	Aprovado
Descrição	O objetivo deste teste é pedir uma autorização de nota de crédito	
Passos	<ol style="list-style-type: none"> 1. Clicar em “My Complaints” 2. Clicar em “Show All” 3. Escolher a reclamação “000-16” 4. Clicar em “Request Payment Authorization” 	
Input	<ul style="list-style-type: none"> • Customer contact: “AAA” • BT Contact: “AAA” • Invoice n°: “FT 003-16” • Anexar ficheiro 	
Output Esperado	<ul style="list-style-type: none"> • Pedido de autorização realizado com sucesso • Código enviado para telemóveis de responsáveis 	
Output Real	<ul style="list-style-type: none"> • Pedido de autorização realizado com sucesso • Código enviado para telemóveis de responsáveis 	

ID	Requisito(s) relacionado(s)	Resultado
CT-14	MC-RC: RC-4, RC-3	Aprovado
Descrição	O objetivo deste teste é autorizar uma nota de crédito	
Passos	<ol style="list-style-type: none"> 1. Clicar na linha onde o “Doc. Number” é 1 2. Clicar em “Accept” 3. Introduzir o código enviado para o telemóvel 	
Input	<ul style="list-style-type: none"> • Code: [Código do telemóvel] 	
Output Esperado	<ul style="list-style-type: none"> • Nota de crédito validada com sucesso 	
Output Real	<ul style="list-style-type: none"> • Nota de crédito validada com sucesso 	

ID	Requisito(s) relacionado(s)	Resultado
CT-15	MC-RC: RC-3, RC-5	Aprovado
Descrição	O objetivo deste teste é validar a nota de crédito no dep. financeiro	
Passos	<ol style="list-style-type: none"> 1. Clicar na linha onde o “Doc. Number” é 1 2. Clicar em “Accept” 3. Clicar em “Submit” 	
Input	-	
Output Esperado	<ul style="list-style-type: none"> • Nota de crédito registada e finalizada 	
Output Real	<ul style="list-style-type: none"> • Nota de crédito registada e finalizada 	

ID	Requisito(s) relacionado(s)	Resultado
CT-16	MC-AR	Aprovado
Descrição	O objetivo deste teste é atribuir clientes aos funcionários	
Passos	<ol style="list-style-type: none"> 1. Clicar em “Assign Customers” 2. Clicar no cliente 	
Input	<ul style="list-style-type: none"> • Search: “Autoneum CZ s.r.o.” • Employee: “Erbi Figueiredo Silva” • Code: “990105” 	
Output Esperado	<ul style="list-style-type: none"> • O cliente deverá ser atribuído com sucesso 	
Output Real	<ul style="list-style-type: none"> • O cliente foi atribuído com sucesso 	

ID	Requisito(s) relacionado(s)	Resultado
CT-17	MC-AR	Reprovado
Descrição	O objetivo deste teste é atribuir clientes a um funcionário inexistente	
Passos	<ol style="list-style-type: none"> 1. Clicar em “Assign Customers” 2. Clicar no cliente 	
Input	<ul style="list-style-type: none"> • Search: “Autoneum CZ s.r.o.” • Employee: “TESTE” • Code: “123123123” 	
Output Esperado	O sistema deverá indicar o erro	
Output Real	Foi atribuído o cliente ao funcionário	

ID	Requisito(s) relacionado(s)	Resultado
CT-18	MC-CHR	Aprovado
Descrição	O objetivo deste teste é consultar o histórico das reclamações e filtrar as mesmas	
Passos	<ol style="list-style-type: none"> 1. Clicar em “My Complaints” 2. Clicar em “Show All” 	
Input	<ul style="list-style-type: none"> • Search: “AS Wendre” • Search: “Open” • Search: “ERSI” 	
Output Esperado	O sistema deverá filtrar as reclamações	
Output Real	O sistema filtrou as reclamações	

ID	Requisito(s) relacionado(s)	Resultado
CT-19	MC-AR: AR-3, AR-4, AR-5	Aprovado
Descrição	O objetivo deste teste é atribuir responsáveis pela validação de documentação	
Passos	1. Realizar login com conta de administrador	
Input	<ul style="list-style-type: none"> • CEO: “Erbi Figueiredo Silva” Code: “990105” • Qual. Chief: “Erbi Figueiredo Silva” Code: “990105” • Suppliers: “Erbi Figueiredo Silva” Code: “990105” • Clients: “Erbi Figueiredo Silva” Code: “990105” 	
Output Esperado	O sistema deverá registar as alterações e ao realizar os testes CT-13, os pedidos devem ser enviados para o utilizador registado no input.	
Output Real	O sistema registou e enviou corretamente os pedidos de validação de documentação.	

ID	Requisito(s) relacionado(s)	Resultado
CT-20	MC-IS	Aprovado
Descrição	O objetivo deste teste é consultar uma reclamação a um fornecedor, utilizando uma vista do SEval.	
Passos	<ol style="list-style-type: none"> 1. Clicar em “My Complaints” 2. Clicar em “Show All” 	
Input	-	
Output Esperado	Mostrar todas as reclamações e itens de reclamações	
Output Real	O sistema mostrou o output esperado	

ID	Requisito(s) relacionado(s)	Resultado
CT-21	PA-CPA: CPA-1, CPA-2 / PA-IPA: IPA-3, IPA-4	Aprovado
Descrição	O objetivo deste teste é editar um plano de ação e atribuir tarefas a funcionários	
Passos	<ol style="list-style-type: none"> 1. Clicar em “My Complaints” 2. Clicar em “Show All” 3. Clicar na reclamação “000-017” 4. Clicar em “Open Action Plan” 	
Input	<ul style="list-style-type: none"> • Why is it a problem: “Alterei isto” • Add Member: “João Valente” • Add External: <ul style="list-style-type: none"> ○ Name: “Externo” ○ Contact: “Externo” ○ Category: “Teste” • Permanent Corrective Action: <ul style="list-style-type: none"> ○ Corrective action: “Teste” ○ Date: “2016-12-04” ○ Respons: “João Valente” • Permanent Corrective Action: <ul style="list-style-type: none"> ○ Corrective action: “Teste2” ○ Date: “2016-12-08” • Respons: “Erbi Silva” 	
Output Esperado	As alterações devem ser registadas e os funcionários adicionados corretamente	
Output Real	O sistema mostrou o output esperado	

ID	Requisito(s) relacionado(s)	Resultado
CT-22	PA-CPA: CPA-3	Aprovado
Descrição	O objetivo deste teste é exportar o plano de ação	
Passos	<ol style="list-style-type: none"> 1. Clicar em “My Complaints” 2. Clicar em “Show All” 3. Clicar na reclamação “000-017” 4. Clicar em “Open Action Plan” 	
Input	-	
Output Esperado	Ficheiro Excel com formato do plano de ação da borgstena	
Output Real	O sistema mostrou o output esperado	

ID	Requisito(s) relacionado(s)	Resultado
CT-23	PA-CPA: CPA-5	Aprovado
Descrição	O objetivo deste teste é anexar um ficheiro ao plano de ação	
Passos	<ol style="list-style-type: none"> 1. Clicar em “My Complaints” 2. Clicar em “Show All” 3. Clicar na reclamação “000-017” 4. Clicar em “Open Action Plan” 5. Clicar em “attachment” 	
Input	<ul style="list-style-type: none"> • Arrastar um ficheiro para a aplicação 	
Output Esperado	O ficheiro deverá ser guardado como anexo	
Output Real	O sistema mostrou o output esperado	

ID	Requisito(s) relacionado(s)	Resultado
CT-24	PA-NPA	Aprovado
Descrição	O objetivo deste teste é verificar as notificações do plano de ação	
Passos	-	
Input	-	
Output Esperado	Notificação de aproximação de vencimento do plano de ação	
Output Real	As notificações e o calendário foram demonstrados	

ID	Requisito(s) relacionado(s)	Resultado
CT-25	PA-CPA: CPA-4	Aprovado
Descrição	O objetivo deste teste é fechar o plano de ação	
Passos	<ol style="list-style-type: none"> 1. Clicar em “My Complaints” 2. Clicar em “Show All” 3. Clicar na reclamação “000-017” 4. Clicar em “Open Action Plan” 5. Clicar em “Close action plan” 	
Input	-	
Output Esperado	“Ação inválida pois existem tarefas que ainda não foram concluídas”	
Output Real	“Ação inválida pois existem tarefas que ainda não foram concluídas”	

ID	Requisito(s) relacionado(s)	Resultado
CT-26	PA-CPA: CPA-6	Aprovado
Descrição	O objetivo deste teste é concluir as tarefas	
Passos	<ol style="list-style-type: none"> 1. Clicar em “My Complaints” 2. Clicar em “Show All” 3. Clicar na reclamação “000-017” 4. Clicar em “Open Action Plan” 5. Clicar em “Done” nas duas tarefas criadas 	
Input	-	
Output Esperado	“Done”	
Output Real	“Done”	

ID	Requisito(s) relacionado(s)	Resultado
CT-27	PA-CPA: CPA-4	Aprovado
Descrição	O objetivo deste teste é fechar o plano de ação	
Passos	<ol style="list-style-type: none"> 1. Clicar em “My Complaints” 2. Clicar em “Show All” 3. Clicar na reclamação “000-017” 4. Clicar em “Open Action Plan” 5. Clicar em “Close action plan” 	
Input	-	
Output Esperado	O plano de ação deve ficar fechado	
Output Real	Plano de ação fechado	

ID	Requisito(s) relacionado(s)	Resultado
CT-28	MF-RC	Aprovado
Descrição	O objetivo deste teste é registar fornecedores no portal	
Passos	<ol style="list-style-type: none"> 1. Realizar login com conta de administrador 	
Input	<ul style="list-style-type: none"> • Fornecedor: “Teste” • Email: “erbi.silva@borgstena.com” 	
Output Esperado	Receber um mail de notificação a informar do registo e password	
Output Real	Mail recebido com o registo e password	

ID	Requisito(s) relacionado(s)	Resultado
CT-29	MF-MR	Aprovado
Descrição	O objetivo deste teste é mostrar as reclamações realizadas a determinado fornecedor logado	
Passos	1. Realizar login com conta de um fornecedor	
Input	<ul style="list-style-type: none"> • Username: “Erbi” • Password: “Erbi” 	
Output Esperado	Reclamação relativa ao fornecedor	
Output Real	Reclamação relativa ao fornecedor	

ID	Requisito(s) relacionado(s)	Resultado
CT-30	MF-OC: OC-1	Aprovado
Descrição	O objetivo deste teste é obter as reclamações introduzidas no IFS	
Passos	[Para realizar este teste foi necessário introduzir as mesmas no IFS]	
Input	-	
Output Esperado	Reclamações registadas no IFS	
Output Real	Reclamações registadas no IFS	

ID	Requisito(s) relacionado(s)	Resultado
CT-31	MF-OC: OC-4, OC-5, OC-6	Aprovado
Descrição	O objetivo deste teste é abrir, editar uma ordem de compra e editar a mesma no IFS. Esta deverá ser listada na aplicação	
Passos	[Para realizar este teste foi necessário introduzir as mesmas no IFS]	
Input	-	
Output Esperado	Reclamações atualizadas	
Output Real	Reclamações atualizadas	

ID	Requisito(s) relacionado(s)	Resultado
CT-32	MF-AOC: OC-1	Aprovado
Descrição	O objetivo deste teste é confirmar um item da ordem de compra	
Passos	<ol style="list-style-type: none"> 1. Clicar na Order No P1313958 2. Clicar no visto (a verde) 	
Input	-	
Output Esperado	Item confirmado	
Output Real	Item confirmado	

ID	Requisito(s) relacionado(s)	Resultado
CT-33	MF-AOC:OC-2	Aprovado
Descrição	O objetivo deste teste é confirmar um item da ordem de compra	
Passos	<ol style="list-style-type: none"> 1. Clicar na Order No P1313989 2. Clicar no lápis (a cinzento) 	
Input	-	
Output Esperado	Data alterada	
Output Real	Data alterada	

ID	Requisito(s) relacionado(s)	Resultado
CT-34	MF-AOC:OC-3, OC-4	Aprovado
Descrição	O objetivo deste teste é confirmar um item da ordem de compra	
Passos	<ol style="list-style-type: none"> 1. Clicar na Order No P1313989 2. Clicar em X (a vermelho) 	
Input	<ul style="list-style-type: none"> • Reason: “Teste” 	
Output Esperado	Item rejeitado	
Output Real	Item rejeitado	

ID	Requisito(s) relacionado(s)	Resultado
CT-35	MF-VE: VE-2, VE-3, VE-4	Aprovado
Descrição	O objetivo deste teste é verificar se é possível obter as avaliações relativas ao fornecedor	
Passos	1. Clicar em “Evaluation”	
Input	-	
Output Esperado	Gráficos de avaliação	
Output Real	Gráficos de avaliação	

ID	Requisito(s) relacionado(s)	Resultado
CT-36	MF-VE: VE-1	Aprovado
Descrição	O objetivo deste teste é verificar se é possível obter o histórico de encomendas	
Passos	1. Clicar em “Historic”	
Input	-	
Output Esperado	Lista de histórico de encomendas	
Output Real	Lista de histórico de encomendas	

ID	Requisito(s) relacionado(s)	Resultado
CT-37	MF-VA	Aprovado
Descrição	O objetivo deste teste é listar as encomendas e verificas as alterações	
Passos	1. Realizar login com conta de gestor de fornecedores e encomendas	
Input	-	
Output Esperado	Lista de encomendas alteradas	
Output Real	Lista de encomendas alteradas	

Anexo D – Validação Requisitos Funcionais

- **Exportar reclamações a fornecedores para Excel**
 - **Problema:** a exportação de reclamações para formato Excel não estava disponível para os gestores de fornecedores;
 - **Solução:** apenas alguns campos mudam no que toca às reclamações de fornecedores pelo que a solução passou por adaptar as funções já criadas para exportação de reclamações de clientes;
- **Atualização de progresso nas reclamações**
 - **Problema:** é pretendido que todas as atualizações de progresso nas reclamações fossem registadas com data e hora para mais tarde serem consultados;
 - **Solução:** na estrutura de dados foi criada uma tabela que estaria relacionada com a reclamação. O utilizador ao abrir o progresso desta última tem acesso a todo o histórico de progresso relativo aquela reclamação;
- **Guardar ações em caso de falhas**
 - **Problema:** sempre que o utilizador cria uma reclamação e em caso de alguma falha (o utilizador errou navegou para outra página ou a energia falhou), os dados eram perdidos obrigando o utilizador a escrever novamente os dados;
 - **Solução:** A aplicação guarda os dados das reclamações de forma temporária numa tabela na estrutura de dados. Por exemplo, quando é criado um cabeçalho de reclamação, os dados ficam guardados até esta estar finalizada ou o utilizador alterar os dados;
- **Aceitar todos os itens de uma ordem de compra:**
 - **Problema:** o utilizador tinha de aceitar item a item. Caso o pedido envolvesse dezenas de linhas, este teria de aceitar uma por uma;
 - **Solução:** Implementação de um botão “Accept All” que permitia ao fornecedor aceitar todos os itens da ordem de compra;
- **Passar reclamação de clientes para fornecedores**
 - **Problema:** transferir uma reclamação de clientes para o gestor de fornecedores. O objetivo é o sistema utilizar o máximo de dados possível e preencher automaticamente o formulário de reclamações a fornecedores;
 - **Solução:** Implementação de um sistema de reclamações pendentes. O gestor de clientes transfere a reclamação para o gestor de fornecedores e este último é alertado via mail ficando com a reclamação pendente no portal. É gerada uma notificação quando o gestor se autenticar e este pode efetuar o download. De seguida preenche o formulário e indica ao software que a reclamação está registada, deixando de estar pendente.
- **Cabeçalho da reclamação não está editável**
 - **Problema:** Impossível alterar alguns dados do cabeçalho de uma reclamação. Estas alterações são importantes no que toca a atualização de dados;
 - **Solução:** Disponibilizar uma opção para edição deste cabeçalho.
- **Reclamações com estado “Cancelado” não podem ser contabilizadas**
 - **Problema:** Quando se vê os detalhes de uma reclamação, o total é contabilizado com o somatório das colunas do “Real Cost” + o “Complaint Cost”

- **Solução:** Quando se faz o somatório, retirar todas as que estão no estado “Cancelled”
- **Chefe de financeiros precisa de mais detalhe**
 - **Problema:** A vista o chefe de financeiros precisa de ver algumas estatísticas e as reclamações sem poder alterar as mesmas;
 - **Solução:** Disponibilizar as vistas a este utilizador

Anexo E – Avaliação de heurísticas

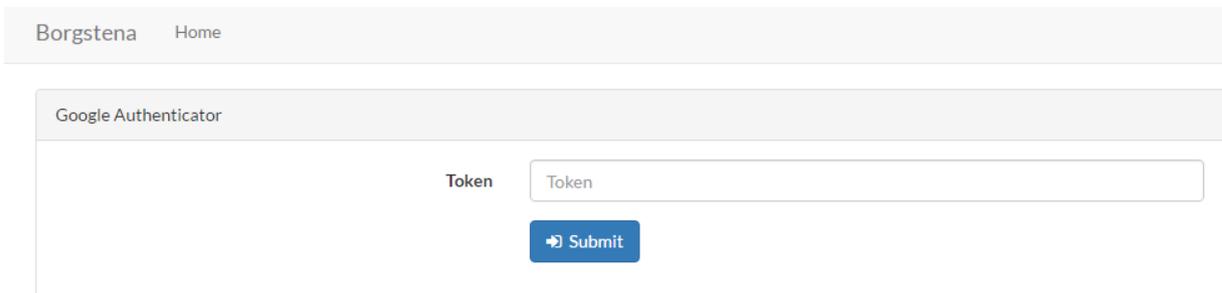
No	Issue Description	Heuristic Violated	Severity	Evaluator(s)	Proposed Solution
1	Nos clientes, Create new, botão Reset sem mensagem de confirmação	5	B	JVAL	Colocar uma mensagem de confirmação para saber se é para prosseguir
2	Plano de ação nos clientes a ser inserido corretamente mas sem mensagem de feedback de sucesso	1	C	JVAL	Colocar mensagem de confirmação
3	Caixa de texto dos costs do create new permite letras quando só devia permitir números	5	A	JVAL	Permitir apenas números
4	No plano de ação não tem nenhum botão para voltar à reclamação a que aquele plano se refere	3	B	JVAL	Colocar um botão para voltar para trás para a reclamação
5	Tabela do Show All nos clientes tem lá uma coluna em português e o resto em inglês	4	C	JVAL	Colocar em inglês a label
6	Nos fornecedores, edição da reclamação, o defeito aparece em PT	4	B	JVAL	Colocar em inglês os dados que vêm da base de dados.
7	Na criação de reclamação de fornecedores, a data deve aparecer automaticamente a de "hoje"	-	D	JVAL	-
8	Custos com caixas de texto vazias devem ser assumidos como zero e não NULL	5	A	JVAL	Colocar como defeito o valor 0
9	Após introduzir reclamação de fornecedores, apareceu uma mensagem de um ficheiro JSON	3, 4	B	JVAL	Retirar linha de código pois parece ser um teste para mostrar o lado do servidor
10	Aplicação dos fornecedores: Aceitação de ordem de compra, a data permite letras	5	A	ERSI	Permitir apenas datas, alterar a validação do Javascript
11	Cor dos botões de submit e update de um item... O formulário é o mesmo mas a cor é diferente dos botões	4	C	ERSI	Alterar a cor para o azul
12	Caixa de texto do insert new complaint de fornecedores,	-	A	ERSI	Aumentar a div onde esta se encontra

	relativamente ao source, está escondida				
13	Cálculo de PPM a apresentar valores demasiado baixos	3	A	ERSI	Verificar a fórmula no código\Verificar a criação do gráfico
14	Request payment authorization. Financeiro. Quando valida, a modal fecha mas o ecrã por vezes fica bloqueado	3	A	ERSI	Colocar o .backdrop
15	No Show All das reclamações falta informação	-	D	ERSI	Colocar o preço estimado, complaint status, action plan status e financial status
16	Dashboard dos planos de ação. No calendário. O utilizador disse que não entendeu se aquilo representava a data de abertura de PA ou outra coisa...	10	C	ERSI	Colocar uma "ajuda" a explicar o que é aquilo
17	Realizar o logout da aplicação	3	C	ERSI	Dar formação da aplicação BLA. Nada relacionado com o portal
18	Erro a ver histórico avaliação de PPM's. A mensagem de erro apresenta uma mensagem própria do Laravel	2	B	ERSI	Corrigir a mensagem de erro. Caso não seja erro, verificar a nível de código o problema
19	Planos de ação, o layout está extenso	8	C	ERSI	Colocar as <div> a abrir e fechar através de uma "seta" para encurtar a mostragem de informação
20	Exportar reclamações para excell	-	D	ERSI	-
21	Colocar gráficos de quantidade de reclamações abertas, fechadas como estatísticas (tipo tabela de status)	-	D	ERSI	-
22	Show All, falta lá o ano para verificar anos anteriores	-	B	ERSI	Colocar uma funcionalidade para navegar por ano
23	Status date aparece 0000-00-00 em vez de vazio	2,4	C	ERSI	Colocar uma verificação pois na BD é que guarda 0000-00-00 quando é null

24	Exportação de reclamação para excel. Quando é SEVERAL a descrição é SEVERAI.Está incompleta	-	C	ERSI	Completar a descrição na base de dados quando inserida.
25	Reclamação de compensação não aparecem na extend view	3	C	ERSI	Filtrar as reclamações para mostrar tudo menos compensação
26	eliminar reclamação de fornecedor. Esta é eliminada mas no refresh aparece lá na mesma	3	B	ERSI	Eliminar dados das tabelas adjacentes. Dá erro de integridade

Anexo F – Google Authenticator

Na imagem a seguir apresentada é possível verificar o ecrã onde se insere o código que está a ser demonstrado no dispositivo móvel.



The screenshot shows a web interface with a header bar containing 'Borgstena' and 'Home'. Below the header is a form titled 'Google Authenticator'. The form contains a label 'Token' next to a text input field with the placeholder text 'Token'. Below the input field is a blue button with a right-pointing arrow and the text 'Submit'.

Na imagem a seguir apresentada é possível verificar uma simples mensagem que indica que o código inserido no ecrã do ponto anterior corresponde ao código correto.



Na imagem a seguir apresentada é possível verificar uma simples mensagem que indica que o código inserido no ecrã do ponto 6.2.1 corresponde a um valor incorreto.

