



Departamento de Engenharia Informática  
Faculdade de Ciências e Tecnologia  
Universidade de Coimbra

# **FactSegur**

**Implementação de um Software de Gestão de  
processos para Reguladores de Sinistros**

**Carlos Freixo**

**Relatório de estágio  
Mestrado em Engenharia Informática**

**Estágio Plurianual 2014/2015 - 2015/2016  
4 de Fevereiro de 2016**



# **FactSegur**

## **Implementação de um Software de Gestão de processos para Reguladores de Sinistros**

### **Relatório de estágio**

#### **Mestrado em Engenharia Informática**

**Autor:**

**Carlos Freixo**

**Orientador:**

**Professor Doutor Paulo Rupino**

**Júri Arguente:**

**Professor Doutor Tiago Baptista**

**Júri Vogal:**

**Professor Doutor Álvaro Rocha**

**Estágio Plurianual 2014/2015 - 2015/2016**

**4 de Fevereiro de 2016**



## Resumo

FactSegur é um projeto idealizado no âmbito de um estágio curricular do Mestrado em Engenharia Informática, desenvolvido por dois estagiários, de onde irá resultar um sistema de informação que permita a gestão dos processos de uma empresa da área de Peritagens/Regulação de Sinistros, bem como a resolução das suas necessidades ao nível da faturação, tendo assim de cumprir os requisitos legais portugueses.

Este relatório diz respeito aos aspetos desenvolvidos pelo aluno A, encarregado dos aspetos gerais do sistema e da realização do componente de gestão de processos e geração de relatórios.

Neste documento são apresentadas as diversas etapas atravessadas pelo estagiário neste projeto, sendo descritos os processos de estudo do estado da arte, escolha da metodologia de desenvolvimento, realização da análise de requisitos e especificação da arquitetura.

Após a finalização destas fases é iniciada a implementação do software, sendo em seguida testado e realizada a documentação destes processos.

Finalmente é apresentada uma conclusão, referindo os planos futuros idealizados para este projeto.

**Palavras-chave:** Sistema de informação, gestão documental, regulação de sinistros, peritos de seguros, gestão de processos, *cloud*, *content management system*



# Índice

1	Introdução.....	1
1.1	Entidades auxiliares.....	1
1.2	Âmbito.....	2
1.3	Definição de objetivos.....	2
1.4	Planeamento.....	3
1.4.1	Planeamento total.....	3
1.4.2	Desvios ao planeamento.....	3
1.5	Estrutura do documento.....	5
2	Estado da arte.....	7
2.1	Contexto.....	7
2.2	Metodologia de pesquisa.....	9
2.3	Software de regulação de sinistros.....	10
	AIS.....	11
	Alyce.....	11
	Claimloop.....	11
	ClaimRuler.....	12
	ClickClaims.....	12
	CloudWorks.....	12
	FileTrac.....	13
	mrinformatica.....	13
	Periagro.....	14
	PeriAva.....	14
	SinistroWeb.....	14
	Virtual Claims Adjuster.....	14
2.4	Software de gestão documental.....	15
	Alfresco One.....	15
	Asite Document Management.....	15
	Contentverse.....	16
	Eclipse Document Management.....	16
	eFileCabinet.....	16
	filedepot.....	16
	FileHold.....	17
	LogicalDoc.....	17
	SmartFile.....	17
2.5	Tabela comparativa.....	17

On site ou Cloud.....	18
Software específico da área.....	18
Componente de gestão documental.....	18
Gestor de workflow.....	18
Portal para utilizadores externos.....	18
Aplicação mobile.....	19
Idioma português.....	19
Módulo de faturação.....	19
Preço.....	19
2.5.1 Discussão dos resultados.....	21
2.6 Conclusão.....	23
3 Metodologia de desenvolvimento.....	25
4 Análise de requisitos.....	27
4.1 Introdução.....	27
4.1.1 Propósito e âmbito do sistema.....	27
4.1.2 Objetivos e critérios de sucesso para o projeto.....	28
4.2. Sistema em uso.....	28
4.3. Sistema proposto.....	29
4.3.1 Visão geral.....	29
4.3.1.1 Núcleo principal.....	29
Mensagens.....	29
Definições.....	30
Perfil.....	30
Pesquisa.....	31
4.3.1.2 Módulo de contactos.....	31
Contactos.....	31
4.3.1.3 Módulo de Processos.....	32
Processos.....	32
4.3.2 Requisitos funcionais.....	32
4.3.2.1 Sistema geral.....	35
Autenticação de Utilizador.....	35
Recuperação de Palavra-Chave de Utilizador.....	35
Modificação dos dados pessoais.....	35
Pesquisa geral.....	35
Aceder às definições do sistema.....	36
Editar utilizador.....	36
Adicionar utilizador.....	36
Apagar utilizador.....	36
Listar nível de acesso de utilizadores.....	36
Editar nível de acesso de utilizadores.....	36



Adicionar nível de acesso de utilizadores.....	36
Apagar nível de acesso de utilizadores.....	37
Listar utilizadores externos.....	37
Editar utilizador externo.....	37
Adicionar utilizador externo.....	37
Apagar utilizador do portal.....	37
Listar nível de acesso de utilizadores externos.....	37
Editar nível de acesso de utilizadores externos.....	37
Adicionar nível de acesso de utilizadores externos.....	37
Apagar nível de acesso de utilizadores externos.....	38
Visualizar logs.....	38
Terminar sessão.....	38
4.3.2.2 Mensagens.....	38
Mostrar dashboard de mensagens.....	38
Mostrar alertas.....	38
Apagar alerta.....	38
Listar mensagens (pessoais).....	38
Ver mensagem.....	39
Responder à mensagem.....	39
Reencaminhar a mensagem.....	39
Apagar mensagem.....	39
Criar nova mensagem.....	39
Listar as minhas tarefas.....	39
Ver minha tarefa.....	39
Listar tarefas atribuídas.....	39
Editar tarefa atribuída.....	40
Criar nova tarefa atribuída.....	40
Apagar tarefa atribuída.....	40
4.3.2.3 Contactos.....	40
Listar contactos Individuais.....	40
Ver contacto Individual.....	40
Adicionar/Editar contacto Individual.....	40
Listar contactos de Empresa.....	40
Ver contacto de Empresa.....	40
Adicionar/Editar contacto de Empresa.....	41
Apagar contacto (Individual ou de Empresa).....	41
4.3.2.4 Processos.....	41
Listar Processos.....	41
Adicionar processo.....	41
Editar processo.....	41
Anular processo.....	41

4.3.2.5 App Mobile.....	41
Login.....	41
Logout.....	42
Listar processos.....	42
Editar processo.....	42
4.3.3 Diagrama de casos de uso.....	43
4.3.4 Requisitos não-funcionais.....	44
Segurança.....	45
Modificabilidade.....	45
Disponibilidade.....	45
Auditoria.....	46
Desempenho.....	46
Usabilidade.....	46
4.3.5 Conclusão.....	46
<b>5 Arquitectura e especificação.....</b>	<b>47</b>
5.1 Estrutura Geral do Sistema.....	48
5.1.1 CloudFlare.....	49
5.1.2 Nagios.....	50
5.1.3 PostgreSQL.....	50
5.1.4 Apache HTTP Server.....	51
5.1.5 Drupal.....	51
5.1.6 Cliente Móvel.....	52
5.2 Arquitetura do software.....	52
5.2.1 Padrão arquitetural.....	53
5.2.2 Perspectiva em camadas do Drupal.....	54
5.2.3 Outros aspetos relativos ao Drupal.....	60
5.2.4 Acerca dos módulos da aplicação FactSegur.....	60
5.2.4.1 Utilizadores internos.....	61
5.2.4.2 Utilizadores externos.....	62
5.2.4.3 Contactos.....	63
5.2.4.4 Mensagens.....	63
5.2.4.5 Tarefas.....	64
5.2.4.6 Basic Webmail.....	64
5.2.4.7 Processos.....	65
5.2.4.8 Definições.....	65
5.2.4.9 Pesquisa.....	65
5.2.4.10 Logs.....	65
5.2.4.11 Drupal Web Service.....	66
5.2.5 Modificações ao tema do Drupal.....	66
5.3 Modelo de dados.....	67
5.4 Conclusão.....	71

6 Implementação.....	72
6.1 Ambiente de desenvolvimento.....	72
6.2 Instalação do Drupal.....	74
6.3 Base de dados.....	75
6.4 Módulos.....	76
6.4.1 Conceitos base.....	76
6.4.1.1 Criação e instalação de um módulo.....	76
6.4.1.2 Validação e submissão de <i>forms</i> .....	78
6.4.1.3 O que é o <i>Menu routing</i> .....	79
6.4.1.4 O que são <i>Auto-loader wildcards</i> .....	80
6.4.1.5 <i>Views</i> .....	81
6.4.1.6 Permissões.....	82
6.4.1.7 Sistema de ficheiros do Drupal.....	82
6.4.2 Listagem de módulos produzidos.....	83
6.4.2.1 Pesquisa avançada.....	83
6.4.2.2 <i>Log</i> .....	85
6.4.2.3 Utilizadores internos.....	86
6.4.2.4 Processos.....	89
6.4.2.5 Mensagens.....	93
6.4.3 Módulos externos.....	94
6.4.3.1 <i>Captcha</i> .....	94
6.4.3.2 <i>Field Group</i> .....	94
6.4.3.3 <i>Date</i> .....	95
6.4.3.4 <i>Date Popup Timepicker</i> .....	95
6.4.3.5 <i>Views</i> .....	95
6.4.3.6 <i>Views Dynamic Fields</i> .....	97
6.4.3.7 <i>Views PHP</i> .....	97
6.4.4 Módulos independentes.....	97
6.5 Tema.....	98
6.5.1 Conceitos introdutórios.....	98
6.5.1.1 Ficheiro <i>.info</i> .....	98
6.5.1.2 Ficheiros <i>.tpl.php</i> .....	99
6.5.1.3 <i>Template.php</i> .....	101
6.5.2 Implementação do tema.....	101
6.6 Conclusão.....	102
7 Testes.....	105
7.1 Testes funcionais.....	105
7.1.1 Testes unitários.....	105
7.1.2 Testes de integração.....	106
7.1.2 Testes de sistema.....	106
7.2 Testes não funcionais.....	107

7.2.1 Testes de performance.....	108
7.2.2 Testes de segurança.....	110
7.2.3 Conclusão.....	110
<b>8 Conclusão.....</b>	<b>111</b>
8.1 Trabalho futuro.....	111

## Lista de imagens

Imagem 1: Planeamento 1º semestre.....	3
Imagem 2: Planeamento 2º semestre.....	4
Imagem 3: Calendário final.....	4
Imagem 4: Comunicação entre Seguradoras e Peritos de Seguros.....	7
Imagem 5: Processo resumido de um perito de seguros.....	8
Imagem 6: Modelo em cascata.....	25
Imagem 7: Diagrama de casos de uso.....	43
Imagem 8: Vista de topo do sistema.....	48
Imagem 9: Fonte: <a href="https://www.cloudflare.com/ddos">https://www.cloudflare.com/ddos</a> .....	49
Imagem 10: Fonte: <a href="https://www.cloudflare.com/ddos">https://www.cloudflare.com/ddos</a> .....	50
Imagem 11 (Fonte: <a href="http://deninet.com/files/How%20Does%20Drupal%207%20Work.pdf">http://deninet.com/files/How%20Does%20Drupal%207%20Work.pdf</a> ).....	53
Imagem 12: Sistematização da arquitetura da Framework Drupal.....	53
Imagem 13 (Fonte: <a href="https://www.drupal.org/getting-started/before/overview">https://www.drupal.org/getting-started/before/overview</a> ).....	55
Imagem 14: Uma outra perspetiva em camadas do Drupal.....	57
Imagem 15: Algumas dos componentes que fazem parte da API do Drupal.....	58
Imagem 16: Fluxo de apresentação de dados do Drupal.....	59
Imagem 17: Vista de módulos da aplicação.....	61
Imagem 18: Tabelas para o módulo de utilizadores internos.....	67
Imagem 19: Tabelas relativas ao módulo de utilizadores externos.....	68
Imagem 20: Tabelas relativas ao sub-módulo de tarefas.....	68
Imagem 21: Tabelas relativas ao módulo de contactos.....	69
Imagem 22: Tabelas relativas ao módulo de processos.....	70
Imagem 23: Ambiente de desenvolvimento.....	73
Imagem 24: Exemplo de ecrã de instalação do Drupal.....	74
Imagem 25: Exemplo do conteúdo de um ficheiro <code>&lt;module_name&gt;.install</code> .....	75
Imagem 26: Exemplo de um ficheiro <code>.info</code> .....	76
Imagem 27: Exemplo estrutural de um ficheiro <code>.module</code> .....	77
Imagem 28: Página de módulos do Drupal.....	78
Imagem 29: Diagrama de fluxo de validação e submissão de um form.....	79
Imagem 30: Menu routing no Drupal.....	79
Imagem 31: Método de funcionamento dos auto-loading wildcards.....	81
Imagem 32: Gestão de ficheiros (upload).....	82

Imagem 33: Gestão de ficheiros (download).....	83
Imagem 34: Construção do form da pesquisa avançada.....	84
Imagem 35: Armazenamento e obtenção de logs.....	85
Imagem 36: Exemplo de apresentação de um form.....	86
Imagem 37: Diagrama de fluxo para a criação de um utilizador.....	87
Imagem 38: Diagrama de fluxo da remoção de um utilizador.....	88
Imagem 39: Diagrama de fluxo de criação de novo processo.....	90
Imagem 40: Geração de um relatório.....	91
Imagem 41: Diagrama de fluxo de envio de mensagem.....	93
Imagem 42: Exemplo de tabs no Drupal.....	94
Imagem 43: Exemplo de um Date Popup.....	95
Imagem 44: Ecrã de configuração de uma vista do módulo Factsegur Internal Users.....	96
Imagem 45: Drupal Chat.....	97
Imagem 46: Exemplo do conteúdo do ficheiro .info de um tema.....	98
Imagem 47: Regiões definidas para o tema deste projecto.....	99
Imagem 48: Exemplo de aplicação dos ficheiros .tpl.php.....	100
Imagem 49: Excerto do ficheiro page.tpl.php.....	100
Imagem 50: Override de funções no Drupal.....	101
Imagem 51: Teste de carga à adição de um processo.....	108
Imagem 52: Teste de carga para a visualização de listagens de processos.....	109
Imagem 53: Teste de carga para a visualização de um determinado processo.....	109

## Lista de tabelas

Tabela 1: Tabela comparativa de softwares.....	20
Tabela 2: Tabela de requisitos funcionais.....	34
Tabela 3: Atributos de Qualidade e sua priorização.....	45
Tabela 4: Informação necessária num ficheiro .info.....	76
Tabela 5: Informação utilizada pelo menu routing [72].....	80
Tabela 6: Tabela de routing do Factsegur advanced search.....	84
Tabela 7: Tabela de routing do módulo Factsegur Log.....	86
Tabela 8: Tabela de routing do módulo Factsegur Internal Users.....	89
Tabela 9: Tabela de routing do módulo de processos.....	93
Tabela 10: Tabela para o registo de testes de sistema.....	107

## Glossário

**Content Management System** - É um software que fornece mecanismos de colaboração e administração desenhado de modo a permitir que utilizadores com poucos conhecimentos de linguagens de programação ou markup possam criar e gerir conteúdos web com relativa facilidade.

**Drupal** - *Content Management System* gratuito e *open source*, escrito em PHP e distribuído sob a GNU General Public Licence.

**Model View Controller** - Padrão arquitetural de software para implementação de interfaces de utilizadores. Divide um determinado software em 3 partes interligadas, para separar a representação interna da informação da forma como ela será apresentada ao utilizador.

**Regulação de Sinistro** - Processo administrativo pelo qual a seguradora verifica as causas e as consequências de um evento, face a um pedido de indemnização de um cliente. Este processo é composto por documentos e dados obtidos junto ao segurado e/ou terceiro eventualmente envolvido. É através da regulação de um sinistro que a seguradora apura o valor do prejuízo, quando existente, para servir de base à possível indemnização, de acordo com as condições contratuais estabelecidas para o seguro em questão. A regulação de sinistro, normalmente, está prevista num contrato de seguro.

**Simple Object Access Protocol** - É um protocolo usado em computação, sendo usado pelos Web Services na sua comunicação. Utiliza XML como formato da mensagem.

**Web Service** - Método de comunicação entre dois dispositivos eletrónicos através de uma rede. É uma função de software fornecida num endereço de rede na Web, com o serviço *always on* no conceito de *utility computing*.



## **Acrónimos**

**API** – *Application Programming Interface*

**CMS** – *Content Management System*

**MVC** – *Model-View-Controller*

**SOAP** – *Simple Object Access Protocol*

**VPN** – *Virtual Private Network*

**DBMS** – *Database Management System*

**IDE** – *Integrated Development Environment*



# Capítulo 1

## Introdução

Após alguns anos a gerir a rede interna de uma empresa de Regulação de Sinistros em Coimbra, foi proposta a realização de um software que suprisse as suas necessidades. O software utilizado, embora cumprisse os objetivos para o qual tinha sido delineado, tornava-se obsoleto e, após cada atualização do sistema operativo utilizado, eram perdidas funcionalidades<sup>1</sup> ou então eram demonstrados comportamentos imprevistos.

O presente documento visa relatar o desenvolvimento desse novo software, desde a fase de inicial, de contextualização, passando pela análise de requisitos, definição da arquitetura, implementação e terminando na fase de verificação e validação do projeto.

Neste capítulo inicial, o objetivo será a introdução do leitor às entidades que deram origem a este projeto, ao seu âmbito e objetivos gerais. Além disso será apresentado o planeamento inicial do projeto e os desvios ao planeado, bem como a estrutura do documento.

### 1.1 Entidades auxiliares

A ReSi - Regulação de Sinistros Lda., com sede e escritório em Coimbra, exerce atividades e tem como objeto a regulação de sinistros dos vários ramos da indústria seguradora, tendo sido constituída em 29 de Dezembro de 1998 e iniciado atividade em Janeiro de 1999 [1].

Através da *know-how* adquirido ao longo dos anos por esta empresa, foi permitida a absorção do seu conhecimento e estudo dos seus processos de modo a fomentar a criação de um sistema que permitisse resolver as suas necessidades e, ao mesmo tempo, as necessidades de outras empresas na mesma área profissional.

### 1.2 Âmbito

Neste estágio será desenvolvido um software para profissionais e empresas do ramo da Regulação de Sinistros.

<sup>1</sup> Nunca foi investigada a fundo a causa de perda de funcionalidades, mas desconfia-se que foi devido às novas versões dos DLL's existentes em versões superiores do Windows não serem compatíveis com versões anteriores.

Sendo este o âmbito principal deste estágio, no entanto haverá sempre a preocupação de permitir que este software possa ser adaptado a outras áreas profissionais, através de um planeamento cuidadoso de todos os aspetos desta aplicação.

### **1.3 Definição de objetivos**

Neste projeto de estágio pretende-se desenvolver uma aplicação específica para a área de peritos de seguros. Esta área possui um conjunto de tarefas a serem desenvolvidas por cada profissional. Têm de abrir processos, redigir documentos relativos a peritagens, tirar fotos e catalogá-las dependendo do processo a que dizem respeito, estar em contacto com as empresas que lhes requisitam serviços, partilhando a informação que foi recolhida, e faturar o serviço prestado.

Neste processo, um profissional necessita de um conjunto de ferramentas que, de alguma forma, lhe facilitem o trabalho, utilizando um programa que lhe permita gerir os clientes e os processos, um editor de texto, um programa de gestão documental ou um servidor privado onde possa arrumar os seus ficheiros por pastas, um portal ou, geralmente, email para estar em contacto com os seus clientes e ainda um programa de faturação (certificado) de modo a poder passar as faturas relativamente ao serviço que presta. Tudo isto pode-se passar a nível individual ou ser tarefa de várias pessoas dentro de uma empresa.

No entanto, repare-se na fragmentação de todas estas tarefas. É necessário utilizar uma ferramenta específica para cada uma destas atividades e muitas vezes não há qualquer interligação entre elas, não é possível passar a informação de um lado para o outro e isso pode implicar uma perda de tempo desnecessária, uma curva de aprendizagem para aprender a utilizar cada uma das ferramentas, ou até gastos superiores ao que seria desejado para obter os mesmos resultados se tudo estivesse integrado na mesma aplicação.

Embora existam aplicações que já têm várias tarefas integradas, há sempre algum aspeto que deixa a desejar, como veremos em capítulos posteriores, um componente que estava em falta ou então um preço demasiadamente elevado para pequenas e médias empresas que, neste contexto económico, lutam contra a adversidade em que o país se encontra.

Sendo assim, o objetivo principal deste projeto é realizar um software que contenha todas estas características necessárias para um programa na área de peritos de seguros, algo que forneça um pacote completo, um “escritório” pronto a usar, com as características que um profissional desta área necessite. Foram reunidas, com o conhecimento obtido através da análise do software que se encontra no mercado e de entrevistas com profissionais do ramo de peritos de seguros um conjunto de funções gerais necessárias para este projeto, que serão postas em

prática no capítulo 3, com a reunião dos requisitos fundamentais deste projeto.

## 1.4 Planeamento

### 1.4.1 Planeamento total

Aquando da apresentação intermédia do estágio, foi apresentado um calendário com as tarefas já efetuadas até ao momento, e o que se planeava fazer a partir dali, tal como se pode ver na imagem 1.

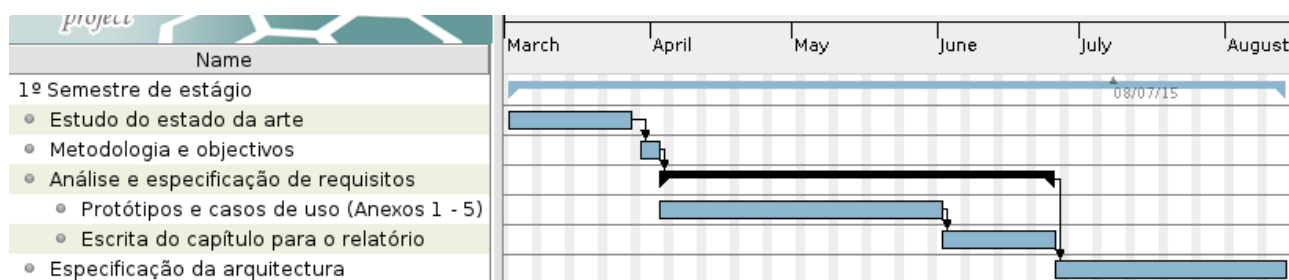


Imagem 1: Planeamento 1º semestre

Em 2015, de 2 de Março a 27 de Março tinha sido realizado o estudo do estado da arte e até 2 de Abril tinha sido finalizado o capítulo de Metodologia e objetivos.

A análise e especificação de requisitos fora terminada a 25 de Junho e era suposto terminar o capítulo de arquitetura até 13 de Agosto, iniciando a partir daí a implementação do projeto, tendo sido apresentado uma proposta de calendarização.

### 1.4.2 Desvios ao planeamento

Dado que na avaliação intermédia não tinha sido bem aceite a utilização do Drupal como uma ferramenta para o desenvolvimento do projeto, devido à falta de justificação, os estagiários sentiram a necessidade de rever este aspeto. Para tal foi construído um documento que, de uma forma justificada, tentou encontrar uma ferramenta que se adequasse ao projeto. Esse estudo pode ser encontrado no anexo 6. Embora a conclusão final tenha sido de novo o Drupal<sup>2</sup>, foi gasto bastante tempo nessa tarefa. Além disso, um outro fator externo levou a que o término do capítulo de Arquitetura só se efetuasse a 19 de Outubro, sendo que só a partir daí é que foi iniciada a fase de implementação.

O planeamento idealizado inicialmente para essa fase é mostrado na imagem 2.

<sup>2</sup> Na comparação final de frameworks, pesou o facto de o estagiário já ter alguma experiência no Drupal.

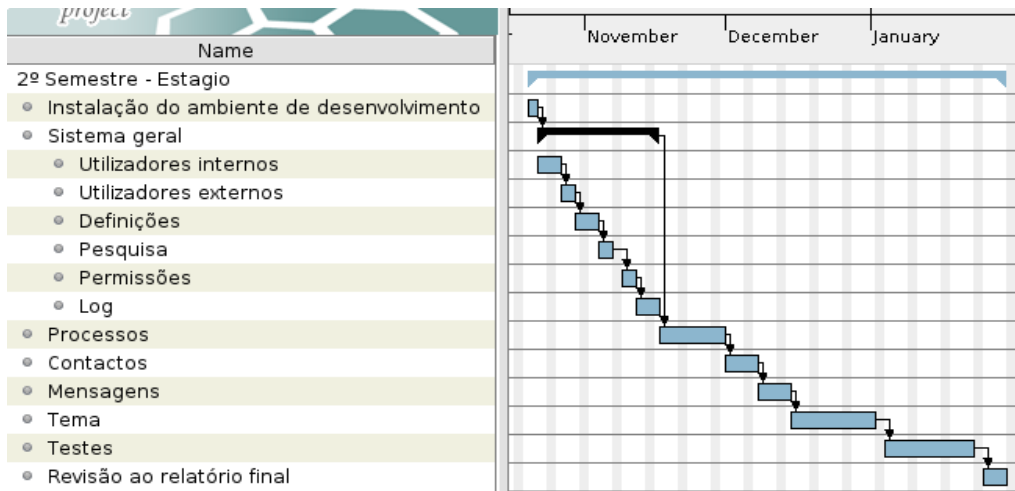


Imagem 2: Planeamento 2º semestre

Neste planeamento previa-se o término da implementação do sistema geral até 16 de Novembro, do módulo de processos até 30 de Novembro, do módulo de contactos até 7 de Dezembro e do de mensagens até 14 de Dezembro. A partir daí efetuar-se-ia o tema e os testes. No entanto, o aspeto final do cronograma ficou como o apresentado na imagem 3.

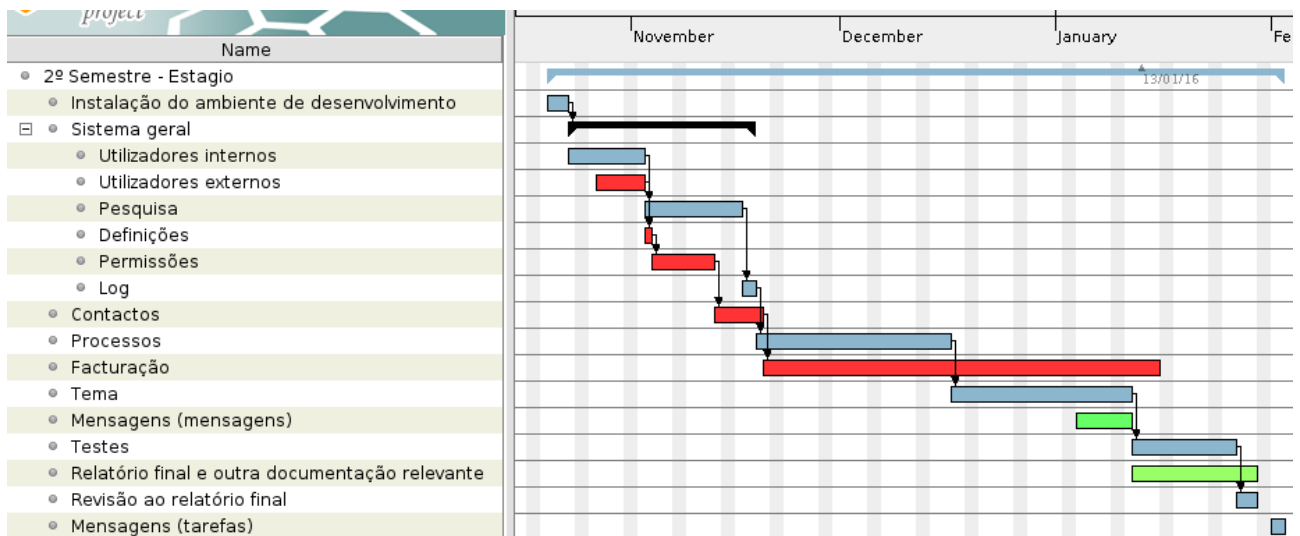


Imagem 3: Calendário final

Embora o projeto até tenha tido uma fase onde se encontrou adiantado, no final ficou com um atraso estimado de 7 dias relativamente ao planeado, e com bastantes modificações de calendarização. Algumas tarefas tiveram de ser paralelizadas (a vermelho, as tarefas realizadas pelo outro estagiário) e a verde estão marcadas as tarefas que tiveram de ser efetuadas em paralelo por este estagiário.

No entanto, todo este processo foi seguido com muita atenção, de modo a arranjar soluções prontamente no caso de haver um atraso muito elevado. Para isso, foram

enviados com regularidade semanal, ao orientador, relatórios 15/5 onde se descrevia o estado do projeto. Estes relatórios podem ser consultados no Anexo 8.

## **1.5 Estrutura do documento**

Este relatório intermédio encontra-se dividido em 4 capítulos essenciais.

No Estado da Arte irá ser mostrado o contexto onde se insere este sistema, e que softwares já existem que contêm um conjunto de características essenciais que poderiam, de alguma forma, ser úteis na área de regulação de sinistros. Para complementar esta informação é ainda adicionada a metodologia que foi utilizada para a pesquisa deste software. No final do capítulo são apresentadas as comparações entre eles e retiradas as respetivas conclusões.

No capítulo de Metodologia de Desenvolvimento é apresentada, de uma forma breve, qual a metodologia que irá ser utilizada no desenvolvimento desta aplicação.

Em seguida temos a Análise de Requisitos, onde será mostrado o âmbito desta aplicação, qual o sistema que se encontra em utilização e qual a solução que é por nós apresentada. Para tal é mostrada uma visão geral do sistema, com um conjunto de requisitos (funcionais e não funcionais) fundamentais para a aplicação.

No quinto capítulo, temos a secção de Arquitetura onde é feita a escolha da arquitetura, de acordo com os requisitos definidos anteriormente.

No capítulo seis é abordado o processo de implementação do software e são descritos todos os aspetos necessários para a sua compreensão.

No capítulo seguinte são descritos o conjunto de testes efetuados sobre o projeto de modo a averiguar o seu bom funcionamento e correção de acordo com os requisitos estipulados.

Este documento é finalizado por um capítulo de conclusão, onde se faz o balanço geral do que foi realizado neste projeto e são apresentados os planos futuros para o seu futuro.





## Capítulo 2

### Estado da arte

#### 2.1 Contexto

A regulação de sinistros (ou peritagem de seguros) é um processo administrativo pelo qual uma seguradora vai realizar ou contratar uma vistoria com o intuito de verificar quais as causas e as consequências de uma determinada ocorrência que levou a que um segurado apresentasse um pedido de indemnização. Através deste processo, são apurados os possíveis valores de compensação, se esta ocorrência está coberta pelo seguro ou mesmo se se trata de um pedido fraudulento [2]. Uma “explicação” resumida deste processo encontra-se exposto na imagem 4.

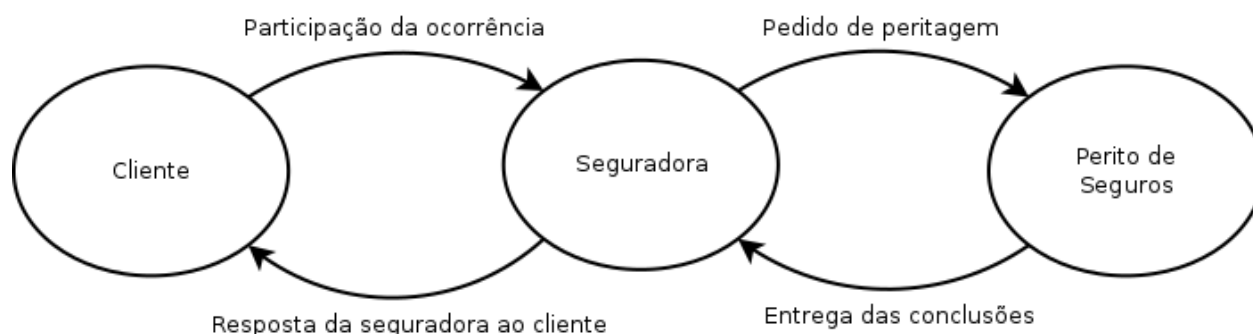


Imagem 4: Comunicação entre Seguradoras e Peritos de Seguros

Sendo assim, um regulador de sinistros (ou perito de seguros) é um profissional qualificado para desempenhar este conjunto de funções. Ele terá de fazer as peritagens relevantes ao processo em causa, criando um conjunto de documentos pertinentes. Irá apurar as causas e os efeitos, tirar fotografias para documentar a situação e, para análise posterior, criará a partir daí um conjunto de documentos com a sua análise à situação para reportar à companhia de seguros que o contratou, tal como indica a imagem 2.

No âmbito deste projeto, procura-se um software que consiga efetuar, de forma eficiente, um conjunto de tarefas específicas para o ramo de Peritos de Seguros.

Foi levantado um conjunto de características através do estudo das tarefas realizadas diariamente por estes profissionais e, sendo assim, esta aplicação terá as seguintes características gerais:

- Módulo de Faturação, para tratar da faturação da empresa
- Módulo de Gestão de Processos de Peritagens da empresa, integrando com o módulo de faturação
- Módulo de Gestão Documental, de modo a que todos os documentos da empresa relativos aos seus processos e faturas sejam fáceis de encontrar e gerir
- Portal para comunicação de documentos entre o perito de seguros e a seguradora

Claro que, embora a presença destas características seja um fator fundamental para a consideração do software em termos de concorrência, ter-se-á de ter em conta, também, a sua presença geográfica (idioma, em que países é comercializado) e os preços que praticados (pois podem ser produtos direcionados para grandes empresas, mas com preços inoportáveis para pequenas e médias empresas).

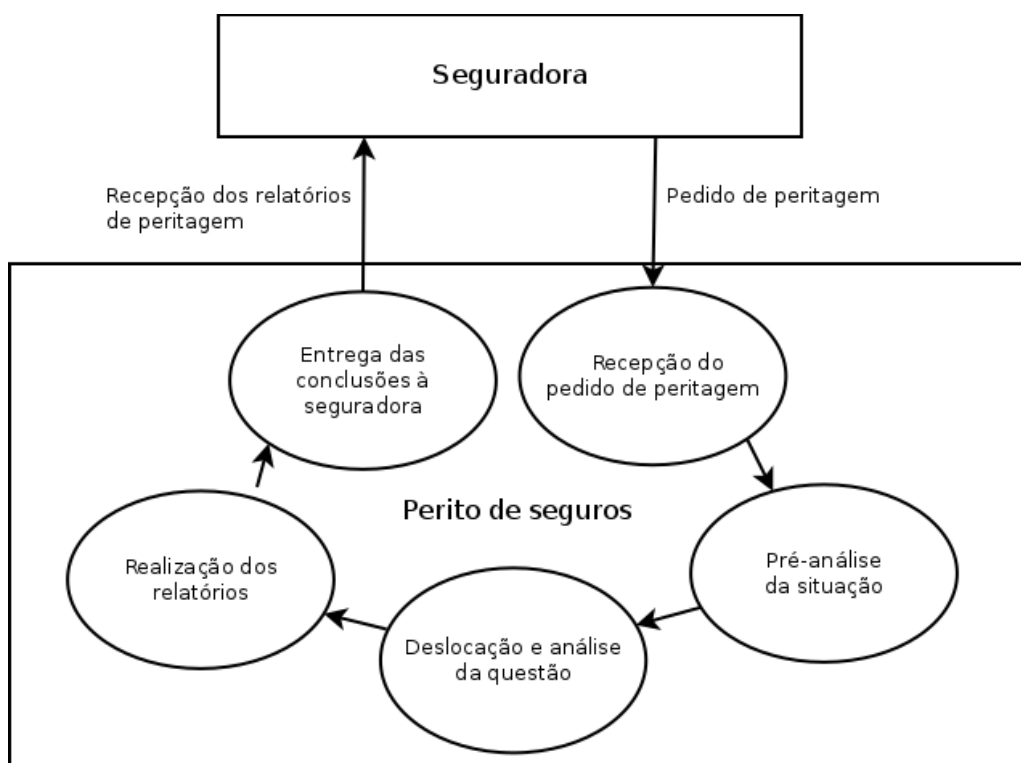


Imagem 5: Processo resumido de um perito de seguros

Um outro aspeto relevante é a presença de um módulo de faturação ou, pelo menos, possibilidade de integração com um software de faturação, de modo a que assim consiga resolver os aspetos relativos à área profissional do perito de seguros (criação de relatórios, gestão documental, etc), bem como aspetos relativos à faturação de serviços prestados por estes profissionais. Mais informações relativas à certificação de software podem ser encontradas no Portal das Finanças [3].

## 2.2 Metodologia de pesquisa

Dado que existe um conjunto de fatores relevantes para este software, foi necessário construir uma metodologia para efetuar a pesquisa de uma forma eficiente, de modo a encontrar resultados realmente relevantes para o assunto em questão.

Embora este projeto de estágio nada tenha a ver com a área de faturação, em primeiro lugar efetuou-se uma pesquisa no Portal das Finanças, tendo em vista a realização de uma pesquisa para software que de alguma forma fosse vocacionado para esta área [4]. Esta pesquisa foi efetuado em primeiro lugar pois esse seria o sítio mais provável para se encontrar um software virado para a área de regulação de sinistros com faturação certificada incluída<sup>3</sup>. Isto porque este estágio está em colaboração conjunta com o estágio de outro colega com vista à integração do software a ser aqui construído, com uma parte de faturação, o que será uma mais valia no âmbito geral da aplicação.

Em seguida foi-se em busca de software nacional ou internacional vocacionado para a área de peritagens de seguros, sem qualquer preocupação com integração ou não com software de faturação. Isto permitiu descobrir que funções estes softwares cumprem neste momento, quais as suas principais preocupações e quais os principais concorrentes que este projeto enfrenta ou pode vir a enfrentar.

Para a pesquisa dentro do mercado português (ou que, pelo menos, publicite em língua portuguesa), foram utilizados termos de pesquisa tais como software de “regulação de sinistros”, “peritagens de seguros”, “avaliação de bens e danos” ou similares.

No caso do mercado internacional, foi necessário procurar qual seria a tradução fiel dos termos de pesquisa em português. Após alguma pesquisa, por tentativa e erro, descobriram-se quais os melhores termos. Foram então efetuadas pesquisas com as expressões “*claim analysis*”, “*insurance adjuster*”, “*insurance claim software*”, “*claims handling firm*” e derivações. Além disso, muita informação foi também retirada do site Capterra [5]. Não foi possível fazer um levantamento

---

<sup>3</sup> O Portal das Finanças apresenta uma listagem de software certificado, sendo por isso uma fonte relevante de pesquisa.

exaustivo, dado que a nível internacional parece existir

Para as empresas internacionais, por falta de informação e sendo que muitas vezes existe uma falta de informação nos sites, para proteção de propriedade intelectual da empresa, é muito difícil discernir se o software em causa realmente diz respeito à área onde este projeto se insere ou se, por outro lado, diz respeito a uma área similar mas divergente e se, de alguma forma, irá cumprir os requisitos estipulados.

Sendo assim, para alguns dos softwares indicados neste sub-capítulo, foi enviado um email ou efetuado contacto telefónico, onde se mostrava a intenção de comprar e utilizar o software numa empresa de peritagens de seguros e se questionava a sua relevância para a área e para o mercado português. Ao todo, foram contactadas 10 empresas, das quais 8 responderam ao pedido de informações. Através das respostas obtidas foi possível complementar a informação contida neste documento.

No total foram escolhidos 12 softwares na área e são descritos na secção 2.3.

Convém, entretanto, fazer a distinção entre um software que é específico para uma área, e um software que pode ser aplicado numa determinada área. Na secção anterior procurou-se um software construído e pensado no âmbito dos profissionais de regulação de sinistros. No entanto, pode-se ainda aplicar a esta área um tipo de software que permite a gestão dos documentos gerados por estes profissionais no seu emprego. Falamos, assim, de software de gestão documental.

Na secção 2.4 é efetuada a pesquisa de mercado de softwares de gestão documental. Esta pesquisa tem um motivo: através de alguns softwares de gestão documental é possível gerir os projetos, documentos e *workflows* desses mesmos documentos dentro de uma empresa. Sendo assim, com algum software complementar de edição de documentos e faturação, poderá ser viável e mais barato para uma empresa optar por este meio para resolver as suas necessidades informáticas. Ainda mais, dado que alguns softwares de gestão documental são de código aberto ou fornecem algum género de API para integração de outro software e novos módulos, podem-se tornar extremamente apetecíveis para solucionar os problemas de uma empresa comum de peritos de seguros, pois poderão aparecer empresas interessadas em integrar essas soluções. A pesquisa efetuou-se através do Google, tendo-se usado em seguida os dados apresentados pelo site Capterra [6].

Finalmente, na secção 2.5, será apresentada uma tabela comparativa que irá sumariar as características relevantes do software encontrado.

## **2.3 Software de regulação de sinistros**

Nesta secção serão descritos os 12 soluções de software especificamente construídos para a área de peritos de seguros. Serão apresentadas as suas características gerais e, caso se aplique, os resultados dos contactos efectuados às

empresas que os produziram.

### **AIS**

O AIS [7] (Advanced Insurance System), é um software produzido pela Garvin-Allen Solutions Limited que, segundo os próprios, é um líder mundial em sistemas de software para companhias de seguros, com a reputação de desenvolverem soluções avançadas para os seus clientes.

Segundo informações disponibilizadas na *webpage* do software, este fornece administração de apólices, faturação, regulação de sinistros, para empresas de propriedade e sinistros.

No entanto, após contacto com a empresa, foi-nos indicado que o módulo de regulação de sinistros implicava que fosse usado o software de Administração de Apólices, pelo que este software não se aplicava a empresas no ramo das peritagens de seguros, mas sim a empresas que lidassem diretamente com a venda de seguros.

Sendo assim não é feito um estudo comparativo deste software no final desta secção.

### **Alyce**

Alyce [8] é um software produzido pela empresa Brightwork, Inc. fundada por Ted Luekens em 1998, servindo micro, pequenas e médias empresas, municípios e auto-segurados<sup>4</sup>.

ALYCE está principalmente vocacionado para este tipo de entidades, lidando com aspetos tais como “Workers' Compensation, Auto Liability, Auto Property, General Liability, and Property claims”.

Em resposta ao email enviado para esta empresa, foi-nos dito que não é âmbito deste programa ser um produto internacional e que, assim sendo, não disponibilizava a língua portuguesa no seu software.

### **Claimloop**

Segundo informação encontrada no site, Claimloop [9] é uma ferramenta online gratuita para simplificar a gestão de ficheiros, centralizar a interação com os vendedores e fornecer o mais alto nível de satisfação ao segurado.

O ClaimLoop disponibiliza um serviço de *cloud*, com aspetos tais como gestão documental, geração de *workflows* e portal.

Foi enviado um email a pedir acesso à demonstração para que se pudesse ter uma melhor ideia de quais seriam as suas características e de que forma poderiam resolver o problema em mãos, mas não foi obtida qualquer resposta e, sendo assim, apresenta-se aqui apenas a informação disponibilizada pelo site.

---

4 Para informações sobre auto-segurados: <http://en.wikipedia.org/wiki/Self-insurance>

## **ClaimRuler**

É um software produzido pela IT Strategies Group que é uma empresa especializada na produção de software para empresas dos ramos de seguros, contabilidade e mercados legais.

O ClaimRuler [10] permite fazer a gestão de processos e ficheiros, ao mesmo tempo que se atribuem tarefas na *cloud*. Além disso, permite fazer a gestão documental da empresa, gerir os *workflows*, e disponibiliza meios para efetuar a comunicação entre a empresa e os seus clientes, de modo a facilitar o trabalho comum de ambos.

Pelo vídeo que é apresentado<sup>5</sup>, parece ser bastante ajustado à área, falhando apenas na parte de disponibilização de um componente de faturação de acordo com a legislação portuguesa.

## **ClickClaims**

ClickClaims [11] é um software da E-Claim.com, LLC. Esta empresa foi fundada em 1999 por Thomas J. Brown, um perito de seguros durante 10 anos para uma firma independente de regulação de sinistros, tendo, por conseguinte, bastante know-how na área.

Este software permite controlar as permissões dos utilizadores relativamente a acesso aos dados, gerir *workflows* e protocolos de modo a otimizar a eficiência da empresa. Além disso permite dar acesso a utilizadores externos aos utilizadores que foram para tal definidos, de modo a poderem aceder a dados específicos. Faz ainda controlo de dados redundantes e tem a possibilidade de integração com outras plataformas.

Infelizmente, depois de terem sido contactados por email, foi reportado que só clientes dos Estados Unidos e Canadá é que podem usar este software pois não têm idioma português e suporte para fusos horários internacionais.

## **CloudWorks**

Na página do CloudWorks [12] do LinkedIn [13], encontra-se o seguinte texto:

“Utilizando a experiência obtida em 30 anos a criar software à medida, entre os quais 20 foram focados no desenvolvimento de soluções para o sector segurador/peritagens, pretendemos criar uma empresa de desenvolvimento de soluções baseadas na Cloud que seja uma referência no sector.”

Sendo assim, foi efetuada uma pesquisa para encontrar algum software feito por esta empresa, relativo a esta área. No entanto não foi encontrada qualquer referência a um software feito, mas sim um *know-how* na área por parte desta empresa que permitia formar parcerias de modo a desenvolver um software à medida da empresa desta esta área.

---

5 <https://www.youtube.com/watch?v=fCACczWUHeU>

Apesar de o número de telefone disponibilizado não estar ativo, após o envio de um email, foi recebida rapidamente a resposta com os detalhes do sistema.

Pela informação que foi fornecida, trata-se de um sistema bastante completo, com aplicação de consulta e envio de fotos e documentos para telemóvel, permite desenhar *workflows* dos processos de modo a adaptar-se à realidade da empresa, gestão documental de modo a gerir o arquivo de documentos da empresa, portal para as seguradoras que são clientes da empresa possam consultar o progresso dos relatórios.

Dado que não se encontra mais nenhuma informação na Internet e que não existe ainda qualquer publicitação deste produto ou serviço, crê-se que esta empresa esteja em vias de lançamento deste serviço.

### **FileTrac**

O FileTrac [14] “permite ao perito independente – mesmo que seja uma empresa de grande dimensão ou um utilizador particular – simplificar o workflow, permitindo o aumento da produtividade e confiança”, segundo a informação do site.

Dado que não foi obtida qualquer resposta ao email que foi enviado, para se poder ter uma opinião mais fundamentada procurou-se por vídeos de demonstração deste software.

Pelo que foi dado a transparecer<sup>6</sup>, trata-se de um software que permite gerir e atribuir tarefas aos utilizadores, ao mesmo tempo que se dá acesso a documentos e outros ficheiros relevantes relativos a um ou mais processos.

### **mrinformatica**

Na sua *webpage* é indicado que “a MR Informática [15] é uma jovem empresa Moçambicana, fundada por dois especialistas informáticos portugueses, que se dedicam às ciências da informação desde 1998”.

Embora numa página *web*<sup>7</sup> e na sua página do *Facebook*<sup>8</sup> seja referido que têm “soluções informáticas *Cloud* para empresas de peritagem” e que são autores de um “sistema de gestão de processos de regularização de sinistros” não foi encontrada qualquer informação relativa a este software na sua página *web*.

Após contacto por email, foi fornecida a informação que o software faz gestão de utilizadores / clientes, gestão de processos, gestão documental básica (associação de documentos a processos), acesso a clientes (dá para definir se o utilizador é uma Seguradora/Cliente, com acessos específicos). Este software pode ser fornecido para Portugal. Além disso, são parceiros certificados Primavera, e disponibilizaram-se a integrar a solução deles com o Primavera.

6 <https://www.youtube.com/watch?v=pa1kiecw4aM>

7 <http://idnacional.com/mrinformatica-1141749.html>

8 <https://www.facebook.com/MRInformatica.pt>

## **Periagro**

Este software foi encontrado através de uma pesquisa no Portal das Finanças, na parte de software de faturação certificado

Segundo informação no site “a PERIAGRO [16] é uma empresa de peritagem direcionada para o mercado segurador, que sempre se posicionou como “*market leader*” neste segmento de mercado. A sua área de atividade abrange um sector muito específico, como é o caso das avaliações ligadas ao mundo rural.”

Mesmo desconfiando que este software seria mais virado para a vertente agrícola, entrou-se em contacto com a empresa. De fato, qualquer que fosse a vertente deste software, foi certificado com o propósito de ser um software próprio e, assim sendo, não é comercializado.

## **PeriAva**

Tal como o PeriAgro, o software PeriAva [17] foi encontrado a partir de uma pesquisa no Portal das Finanças, na secção de programas certificados e partilha o seu nome com a empresa que o criou.

Após bastante pesquisa na Internet, não foi encontrada qualquer referência a características deste software e a página da empresa contém apenas a informação “Página temporariamente indisponível”.

Decidiu-se então entrar em contacto telefónico com a empresa e após uma breve conversa ao telefone com a empresa PeriAva, foi-nos dito que o software é interno e que não é para uso comercial por parte de outras empresas.

## **SinistroWeb**

SinistroWeb é um software online que permite fazer a gestão de processos de uma empresa de peritagens. Além disso ainda permite a gestão de clientes, documentos e até da faturação da empresa [18].

Dado que é um software brasileiro, o processo de faturação não está de acordo com a legislação portuguesa pelo que não pode ser utilizado em território nacional para esse fim, no entanto pode-se revelar uma solução para empresas que já usem um determinado tipo de software para efetuar faturação mas que necessitem de um outro para gerir os seus processos.

## **Virtual Claims Adjuster**

Virtual Claims Adjuster (propriedade de Code Evolution Inc.) é uma corporação privada em software e soluções web para o ramo de litígios relacionados com seguros<sup>9</sup>, bem como para outras indústrias ao longo do mundo [19].

Embora seja disponibilizada muito pouca informação na página, havia a hipótese de ver uma demo online, por isso preencheu-se o formulário e enviou-se o

9 “Claim Industry” eram os termos usados na página



requerimento para a supracitada demonstração.

Foi recebido um email a solicitar informações sobre a empresa a que dizia respeito o pedido de demonstração. Enviou-se um email de resposta a indicar que se estaria à procura de um software para uma empresa no ramo de regulação de sinistros e o conjunto de características que se procurava nesse software.

A resposta fornecida indicava que o software cumpria todos os requisitos que eram pedidos, com a exceção do software ter língua portuguesa e faturação de acordo com a legislação portuguesa. Além disso foi fornecida uma listagens de preços e opções de pagamento.

## 2.4 Software de gestão documental

### **Alfresco One**

John Newton (co-fundador da Documentum) e John Powerll (um antigo COO da Business Objects) fundaram a Alfresco, Inc, em 2005, tendo esta empresa como produto principal o Alfresco One [20].

É, de fato, uma solução para gestão documental bastante completa, baseada em Java, e *open source (LGPL)* [21]. Pode ser fornecida uma conta na cloud, ou então é também oferecida a hipótese de se fazer download do software para instalação num servidor próprio [22].

Além de conter, obviamente, a parte de gestão documental, é ainda possível gerir o *workflow* de documentos [23] e editá-los, através da integração com o Google Docs [24], por exemplo. Tem ainda a hipótese de instalação do idioma português [25].

Apesar de todos estes prós, não foi encontrado qualquer módulo de faturação de acordo com a legislação portuguesa. No entanto, dado que este software é *open source* e o Alfresco conta com um grande conjunto de contribuidores, pode um dia vir a existir alguma integração com algum programa ou site externo.

### **Asite Document Management**

Segundo informação disponibilizada pela própria em empresa, a Asite ajuda pessoas a partilhar informação e construir conhecimento num ambiente seguro na *cloud* [26].

Pela informação disponibilizada pelo site, o que é dado a entender é que este software é uma parte de um projeto maior chamado Adodle, um software com múltiplos componentes disponibilizado na *cloud* [27].

Além da gestão documental, este software oferece também a hipótese de edição de documentos em diversos formatos (estando entre eles os ficheiros mais comuns utilizados nos editores de texto). Existe a possibilidade de gerar *workflow* de

documentos, mas nada é falado relativamente a outros idiomas presentes no software ou qualquer módulo de faturação de acordo com a legislação portuguesa.

### **Contentverse**

Contentverse é um software da empresa Computhink, que foi fundada em 1994 em Chicago.

Este software é descrito pelos seus produtores como um CMS capaz de realizar gestão documental e criar e gerir *workflows* de modo a facilitar o trabalho de uma empresa. Além disso suporta uma enorme quantidade de formatos de ficheiros que pode abrir e editar, mesmo que a aplicação “nativa” não esteja instalada [28].

### **Eclipse Document Management**

O Eclipse Document Management é um produto da docStar, fundada em 1996, sendo uma divisão da Astria Solutions Group, LLC.

É um software de gestão documental que oferece a hipótese de utilização do serviço na *cloud* ou nas instalações da empresa onde se pretende utilizar. Oferece a hipótese de geração de *workflow* de documentos sendo que não foi encontrada qualquer forma de edição de documentos (possivelmente apenas com editores externos). Não foi encontrada qualquer informação que indicasse o suporte para a língua portuguesa [29].

Além disso não foi encontrada qualquer informação que indicasse a existência de um módulo de faturação, muito menos que estivesse de acordo com a legislação portuguesa.

### **eFileCabinet**

Propriedade da eFileCabinet Inc, fundada em 2001, começou por ser uma ferramenta para guardar registo em firmas de contabilidade. Com o tempo, foi evoluindo e hoje em dia é mais um software de gestão documental, que permite manter o controlo dos ficheiros da empresa, ao mesmo tempo que dá acesso a utilizadores externos, através de um portal, a documentos que lhes digam respeito e que tenham sido anteriormente disponibilizados pela empresa [30].

Além disso faz a gestão e *logging* do *workflow* dos documentos para que esteja de acordo com as diversos standards existentes e pelo qual uma empresa tenha de se gerir.

Pelo que é dado a perceber nos vídeos de demonstração, é uma aplicação bastante completa, com características muito interessantes.

### **filedepot**

FileDepot é um módulo de Drupal que permite integrar nesse CMS a capacidade de gestão de ficheiros [31].

Drupal é um CMS (Content Management System) open-source escrito em PHP e distribuído sob a GNU General Public License, sendo usado como framework de back-end para pelo menos 2.1% de todos os sites na Web [32].

É ainda possível enviar notificações entre os utilizadores do sistema, mas em termos de capacidades relevantes, este módulo é bastante limitado. No entanto, com a integração de outros módulos no Drupal pode-se tornar uma opção interessante.

### **FileHold**

Produto da FileHold Systems Inc., designer e produtora de software de gestão documental para médias e grandes empresas.

É uma aplicação que pode ser instalada no computador e a partir daí pode-se fazer upload dos documentos e outros ficheiros para um servidor remoto, fazendo a gestão documental desses ficheiros, gerar e gerir *workflows* [33].

### **LogicalDoc**

Fundada em 2006, Logical Objects é uma empresa mundial de software que resolve as necessidades de Gestão Documental Empresarial dentro de um grande conjunto de sectores e mercados.

Este software permite fazer a gestão documental dos ficheiros internos, definição de permissões, gestão e criação de *workflows*, gestão de utilizadores, edição de documentos, disponibilizam uma *app mobile* e ainda existe a possibilidade de edição de documentos dentro do próprio ambiente [34].

Sendo assim é uma solução tão completa como o Alfresco One, falhando apenas na parte de faturação de acordo com a legislação portuguesa mas, mesmo assim, uma ferramenta capaz de resolver a maioria das questões de uma empresa que necessite de ter uma boa ferramenta de gestão documental.

### **SmartFile**

Fundada em 2009, SmartFile é uma companhia de Indianápolis, Indiana, cujo software partilha o nome da empresa.

SmartFile é um software que permite a gestão de ficheiros corporativos, tanto dentro das empresa, como por um serviço disponibilizado na *cloud*.

Este software permite a atribuição de permissões a um conjunto de pastas ou ficheiros nos repositórios de uma determinada empresa. Não foram identificadas mais características relevantes para o âmbito deste projeto [35].

## **2.5 Tabela comparativa**

Apresenta-se em seguida a tabela comparativa com as características principais do

software encontrado.

Antes, no entanto, é necessário mostrar quais os critérios que foram adotados na sua construção e que aspetos importantes são referidos, sendo o nome da coluna a que dizem respeito referidos entre parênteses a **negrito**. Estes critérios foram obtidos pelo estudo das necessidades<sup>10</sup> da empresa de regulação de sinistros que serve de apoio a este projeto.

## On site ou Cloud

Embora a tendência dos últimos anos seja que cada vez mais software tenha a sua existência na *cloud*, e que cada vez mais serviços sejam prestados por este meio, ainda assim pode ser uma mais valia para um software desta área ter a opção de ser instalado localmente, pois muitas empresas podem não querer que os seus dados e os dos seus clientes estejam alojados fora do recinto físico da empresa.

Sendo assim, torna-se importante referir se o software pode ser instalado localmente (**On site**) e/ou se é algo que é usado na *cloud* (**Cloud**).

## Software específico da área

Um software que seja construído especificamente a pensar numa área profissional vai permitir que se produzam, em princípio, melhores resultados e de forma mais eficiente. Consequentemente, um software pensado para a área de regulação de sinistros fornecerá soluções que não serão apresentados por outro tipo de softwares mais genéricos tais como os softwares de gestão documental, que também são apresentados e comparados. Este aspecto é contemplado na tabela comparativa (**Soft. Área**).

## Componente de gestão documental

Na empresa que fomentou a realização deste projeto, é utilizada um servidor de ficheiros<sup>11</sup> para que os peritos possam guardar informações relativas aos processos, tal como imagens, documentos ou outro tipo de ficheiros. Sendo assim, faz sentido que o software permita efetuar a gestão destes documentos, de modo a que estes elementos possam ser facilmente catalogados.

## Gestor de workflow

Outro aspeto importante é a existência ou não da hipótese de definição de *workflows* (**Workflow**). Como nem todas as empresas partilham do mesmo método de trabalho, torna-se importante para um software deste tipo deixar definir qual o *workflow* pretendido pelos seus utilizadores.

## Portal para utilizadores externos

A existência de um portal (**Portal**) neste tipo de software também é de bastante

<sup>10</sup> Estudo das tarefas realizadas diariamente, ferramentas utilizadas e entrevistas aos colaboradores da empresa.

<sup>11</sup> É utilizado um servidor Debian Linux com Samba instalado [36].

relevância, pois é efetuada uma enorme quantidade de contactos entre o cliente e a empresa de peritos no que diz respeito à troca de informações ou de ficheiros relacionados com os processos com os quais lidam todos os dias. A existência de um canal próprio e adequado para a troca destas informações é relevante.

### **Aplicação mobile**

Além disso, a associação de uma *app mobile* (**App Mobile**) para uma aplicação deste género é um item de grande interesse. Dado que, por exemplo, são adicionadas muitas fotografias aos processos, e dado que os telemóveis hoje em dia são quase elementos indispensáveis em algumas áreas (e nesta pode ser uma mais valia devido à existência da máquina fotográfica, que são de cada vez maior qualidade), uma aplicação que disponibilizasse este meio de comunicação com a aplicação principal seria uma grande mais-valia.

### **Idioma português**

Um outro aspeto relevante é a existência do idioma português (**Idioma PT**) para os softwares encontrados. Embora vivamos numa sociedade cada vez mais poliglota (em particular, relativamente ao domínio do inglês), o fato de estar disponível ou não a língua portuguesa na aplicação pode ser um motivo de escolha ou de descarte.

### **Módulo de faturação**

Embora não contemplado neste estágio, a existência de um módulo de faturação certificado poderá ser um grande contributo para o sucesso de um software nesta área, pois integrará algo que é necessário que a empresa de regulação de sinistros ao mesmo tempo que, ao permitir o uso de dados já existentes no programa, irá poupar tempo ao utilizador.

### **Preço**

Por fim, um grande aspeto que poderá ser até um fator decisivo, é o preço a pagar pelo software. Dado que muitas empresas que prestam este género de serviços são pequenas ou médias empresas, o seu orçamento para aplicar na área da informática poderá não ser muito grande, sendo que um software com um preço muito elevado estará descartado logo à partida.

	On site	Cloud	Soft. Área	Gest. Doc.	Workflow	Portal	App Mobile	Idioma PT	Fact.	Preço
<b>Softwares de Regulação de sinistros</b>										
Claimloop	✗	✓	✓	✓	✗	✓	✓	✗	✗	0€
Claimruler	✗	✓	✓	✓	✓	✓	✗	✗	✗	[15 – 40]\$/ ut. (1)
CloudWorks	✗	✓	✓	✓	✓	✓	✓	✓	✓ <sup>(2)</sup>	75€ / mês / ut. (3)
FileTrac	✗	✓	✓	✓	✓	✓	✓	✗	✗	Taxa / fich. (4)
mrinformatica	✓	✗	✓	✓	✓	✓	✗	✓	✓ <sup>(2)</sup>	--
SinistroWeb	✗	✓	✓	✓	✓	✓	✗	✓	✗	– (4)
VCA	✗	✓	✓	✓	✓	✓	✓	✗	✗	[4–9.25]\$/reg. or [64–234]\$/ut. (mensal)
<b>Softwares de Gestão Documental</b>										
Alfresco One	✓	✓	✗	✓	✓	✓	✓	✓	✗	--
Asite	✗	✓	✗	✓	✓	✓	✗	✗	✗	[15–60]€/mês
ContentVerse	✗	✓	✗	✓	✓	✓	✓	✗	✗	[20–40]\$/ ut. / mês
Eclipse	✗	✓	✗	✓	✓	✗	✓	✗	✗	--
eFileCabinet	✓	✓	✗	✓	✓	✓	✗	✗	✗	[30–55]\$/ ut. / mês (5)
FileDepot	✓	✓	✗	✓	✗	✗	✗	✗	✗	0€
FileHold	✓	✓	✗	✓	✓	✗	✓	✗	✗	\$3.750 + 200\$/ut. ou taxa por ut. (6)(7)
LogicalDoc	✓	✓	✗	✓	✓	✓	✓	✓ <sup>(8)</sup>	✗	[600 – 1500]€ + [15 – 40]€/ut. + 22% anual
SmartFile	✓	✓	✗	✓	✗	✗	✗	✗	✗	[30 – 100]\$/ ut./ mês (9) (10)

Tabela 1: Tabela comparativa de softwares

## Legenda

- (1) <https://www.claimruler.com/claim-ruler-pricing> (desatualizado, não foi encontrada outra informação)
- (2) Usando software externo
- (3) Preço e informações da aplicação fornecidas por email
- (4) Não havia informação disponível de preço e não foi obtida resposta para o contacto
- (5) <http://www.efilecabinet.com/efilecabinet-desktop/#plans-pricing>
- (6) [http://www.filehold.com/sites/all/files/file-uploads/FileHold\\_Express\\_Datasheet\\_Apr2011.pdf](http://www.filehold.com/sites/all/files/file-uploads/FileHold_Express_Datasheet_Apr2011.pdf)
- (7) [http://www.filehold.com/sites/all/files/file-uploads/FileHold\\_Enterprise\\_Datasheet\\_Apr2011.pdf](http://www.filehold.com/sites/all/files/file-uploads/FileHold_Enterprise_Datasheet_Apr2011.pdf)
- (8) Português do Brasil
- (9) <https://www.smartfile.com/pricing/>
- (10) Para utilizadores ilimitados, o preço pode ser negociado diretamente com a empresa

### 2.5.1 Discussão dos resultados

Em primeiro lugar tem-se os softwares AIS, Alyce, ClickClaims, PeriAgro e PeriAva que, por motivos explicados anteriormente, não foram avaliados e, assim sendo, não são apresentados na tabela.

Sendo que as características gerais estão bem visíveis na tabela e que cada um dos itens foi explicado anteriormente, apresenta-se um breve comentário sobre os preços de modo a não deixar dúvidas acerca da interpretação da tabela.

- O software ClaimLoops apresenta uma solução gratuita para profissionais desta área, conforme o indicado na sua página.
- O Claimruler, tal como outros softwares na área, apresenta um preço por utilizador dependente da quantidade de utilizadores que vão utilizar o sistema<sup>12</sup>, sendo que o preço varia entre os 15\$ e os 40\$.
- O preço fornecido pela empresa produtora do CloudWorks foi de 75€ por cada utilizador, pago mensalmente. O preço apresentado foi dado mediante a apresentação do cenário corrente da empresa, com 8 utilizadores. Não se sabe se farão um preço diferente para empresas com maior número de utilizadores.
- Para o FileTrac não foi possível obter informações acerca do preço, mesmo após contacto direto com a empresa produtora. A única informação disponível foi a existência de uma “taxa” por cada ficheiro gerado.
- Relativamente ao software da mrinformatica, não foram obtidos pormenores do preço do software.
- Para o SinistroWeb não foram obtidas informações acerca do preço.

<sup>12</sup> Quanto mais utilizadores, menor será o preço por utilizador.

- O VCA (Virtual Claims Adjuster) apresenta dois modelos de pagamento, por utilizador e por quantidade de processos da empresa<sup>13</sup>. Por utilizador o preço varia entre 64\$ e 234\$ mensais, e por processo o preço varia entre os 4\$ e os 9.25\$.
- À espera da resposta do Alfresco
- O Asite apresenta um preço entre 15€ e 60€, sem quaisquer constrangimentos relativamente a ficheiros ou a utilizadores.
- O ContentVerse apresenta um preço mensal entre 20\$ e 40\$ por cada utilizador<sup>14</sup>.
- Não se encontrou qualquer informação de preço relativamente ao Eclipse, e não houve resposta ao contacto por email.
- O eFileCabinet apresenta um preço mensal entre os \$30 e os 55\$ por utilizador<sup>15</sup>.
- O FileDepot, embora apresente um preço de 0€ mensais, é apenas um módulo de Drupal, tendo de ser adaptado pela própria empresa para poder ser utilizado.
- O FileHold Express tem um pagamento único de \$3,750.00 pelo software do servidor, scanning software, 5 licenças de utilizadores e 6 meses de suporte técnico. Podem ser adicionadas mais 15 licenças de utilizador por 200\$ cada uma. A partir dos 6 meses, o preço do suporte técnico terá um preço anual de 20% do preço inicial de licenciamento. Para empresas com mais de 20 utilizadores existe a opção de pagamento de uma taxa por utilizador, mas não foi disponibilizada qualquer informação sobre esse preço, deverá depender da quantidade.
- O LogicalDoc tem diversas ofertas de preço (dependendo das potencialidades do sistema que se pretenda utilizar). Assim sendo, a entrada inicial varia entre os 600\$ e os 1500\$, sendo adicionado a isso uma taxa por utilizador entre 15\$ e 40\$, mais uma taxa de renovação anual de 22% do preço pego no registo inicial.
- Para o SmartFile, o preço por mês varia entre os 30\$ e os 100\$, consoante a quantidade de utilizadores. No entanto, caso a empresa pretenda um conjunto ilimitado de utilizadores, pode negociar o preço diretamente com os seus produtores.

Como se pode ver, a maioria dos softwares apresenta um preço proibitivamente elevado para as pequenas e médias empresas desta área, em Portugal, e se o problema não for o preço, existe a barreira do idioma (a maioria dos softwares não

13 Quando mais processos ou quanto mais utilizadores forem apresentados, menor será o preço por cada um.

14 Mais uma vez com a modalidade de preço menor por utilizador, consoante a quantidade de utilizadores.

15 Modalidade similar ao do ContentVerse.



apresenta a opção do idioma português).

Além disso, chega-se à conclusão que não existe nenhum software verdadeiramente com as características que se procuram. Mesmo aqueles que se aproximam mais (por exemplo, o CloudWorks), utilizam software externo para esse efeito, sendo que estão dependentes de uma empresa externa para terem uma componente de faturação certificada no seu software.

## **2.6 Conclusão**

Neste capítulo pode-se encontrar quais os principais concorrentes ao sistema que se pretende construir neste estágio e que características fundamentais apresentam. Além disso, ganhou-se uma perspetiva dos preços praticados e criou-se uma noção se o mercado alvo está apto a comportar esses preços.

Depois desta fase, o aluno está preparado para fazer o levantamento de requisitos e encontrar os aspetos essenciais que o sistema deverá suportar.



## Capítulo 3

# Metodologia de desenvolvimento

Dado que este projeto contém um conjunto de objetivos de desenvolvimento bem definidos e que não é provável que venham a sofrer qualquer alteração durante o seu desenvolvimento, optou-se por um modelo de desenvolvimento em cascata.

O modelo em cascata é um processo de design sequencial, usado em processos de desenvolvimento de software, onde o progresso é visto como uma corrente descendente uniforme (como uma cascata), através de um conjunto de fases [37].

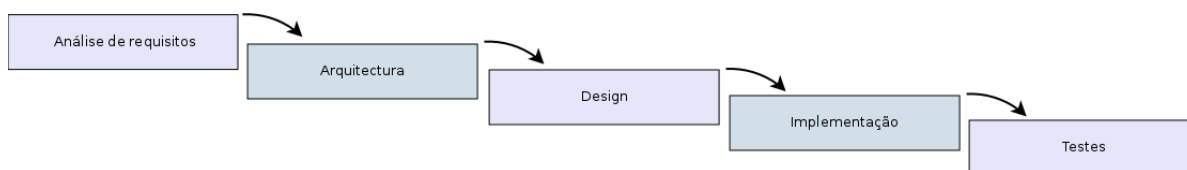


Imagem 6: Modelo em cascata

Na análise de requisitos vão ser realizadas tarefas com vista à determinação das necessidades e condições impostas para atender as exigências de um (conjunto de) cliente(s), tendo em vista a resolução do(s) seu(s) problema(s) [38].

Para ajudar à consolidação dos requisitos, foi criado um conjunto de protótipos da interface que são apresentados nos Anexos 1, 2, 3, 4, 5, respetivamente com o seu caso de uso.

Na fase de arquitetura de software serão criadas as estruturas relevantes para o projeto e a respetiva documentação. Este processo é necessário para que o projeto possa ser compreendido e contém os elementos de software, bem como as suas propriedades e a relação entre eles [39].

Sendo assim, no final destas 2 fases será possível proporcionar-se uma discussão deste conjunto de questões, de modo a garantir que as fases seguintes serão bem sucedidas e que se encontram num bom caminho.

Na fase de implementação, serão codificados os aspeto que foram sendo

levantados nos processos anteriores, garantindo que o projeto está de acordo com os aspetos relevantes para os utilizadores alvo.

Finalmente, todo este processo irá ser concluído com uma fase de testes, de modo a garantir a sua correção e todos os requisitos de qualidade considerados mais relevantes.

# Capítulo 4

## Análise de requisitos

A análise de requisitos é uma fase crítica para o sucesso de um projeto de software. Os requisitos devem ser documentados, mensuráveis, verificáveis, rastreáveis, relacionados com as necessidades de negócio identificados ou oportunidades, e definido com nível de detalhe suficiente para a construção do sistema [38].

Os requisitos devem ainda de acordo com as necessidades do cliente e o quanto maior o grau de pormenor e completude que apresentar este documento, maior será a probabilidade de termos, no final, um produto próximo do pretendido pelos seus futuros utilizadores.

Dever-se-á obter uma descrição completa do sistema em termos de requisitos funcionais e não-funcionais, tendo sempre o cuidado de se escrever o texto de uma forma não técnica, pois a linguagem deverá estar dentro do domínio do cliente/utilizador alvo deste projeto pois é uma parte do público alvo desta secção [40].

### 4.1 Introdução

#### 4.1.1 Propósito e âmbito do sistema

O propósito deste projeto será desenvolver um sistema que ajude na resolução das tarefas diárias enfrentadas por uma empresa do ramo de regulação de sinistros. Assim sendo, e como já foi referido anteriormente, terá de ter um conjunto de características específicas orientadas para esse propósito.

Terão de ser contempladas funcionalidades tais como gestão de utilizadores, gestão de processos, gestão documental, um portal para troca de informação com os clientes da empresa, uma *app mobile* para obtenção de dados e envio de ficheiros relevantes para a aplicação, um gestor de *workflows* e um módulo de faturação (fora do âmbito deste estágio).

Relativamente ao âmbito, este sistema será aplicado à área de peritos de seguros.

Todos os aspetos aqui pensados estarão concentrados nessa área profissional, todos as tarefas que um profissional dessa área tenha de lidar no seu dia-a-dia estarão contemplados, de forma a tornar a sua tarefa mais fácil e eficiente.

No entanto, este sistema deverá ser o mais modular possível para que, através da criação de novos módulos ou complementando os existentes, este software possa ser utilizado noutros âmbitos.

### **4.1.2 Objetivos e critérios de sucesso para o projeto**

Sendo este um projeto de estágio curricular, o principal objetivo de sucesso será a conclusão e total cumprimento de todos os objetivos aqui propostos.

## **4.2. Sistema em uso**

Na empresa que neste momento serve de base e incentivo para a criação desta nova aplicação, encontra-se em utilização um software já antigo que permite a gestão dos processos da empresa e a realização da sua respetiva faturação. No entanto este programa já não é compatível com os sistemas operativos mais recentes, e a sua utilização começa-se a revelar um risco pois, como já aconteceu no passado, a qualquer momento devido a uma atualização do sistema operativo, pode deixar de funcionar, dificultando assim o dia a dia de quem lá trabalha.

Essa aplicação encontra-se dividida em duas partes: a aplicação principal e a aplicação portátil. A aplicação principal dá acesso a todos os componentes, tal como gestão de utilizadores, gestão de processos, contabilidade e faturação. A aplicação portátil permite fazer o preenchimento de processos e, após a chegada ao escritório, efetuar a sincronização com a base de dados que lá se encontra (a base de dados encontra-se no formato Microsoft Access).

Também nesse escritório encontra-se um servidor Linux com Samba instalado, de modo a permitir a acomodação de ficheiros de forma partilhada na empresa. Para os acessos externos é usada a aplicação ExpanDrive, que simula uma pasta de rede no ambiente Windows, embora a ligação seja efetuada por SFTP.

No entanto, com as novas regras para certificação de programas de faturação, a parte de faturação da aplicação atrás referida já não pode ser usada e, sendo assim, é usado uma outra aplicação externa para este fim.

Como se pode ver, o cenário está longe de ser o ideal. Além de se estar a usar uma aplicação que a qualquer momento pode deixar de funcionar, o resto dos componentes estão, de certa forma, “desagregados”, não existe qualquer comunicação entre os ficheiros que estão no servidor, os processos que estão na aplicação e as faturas que são emitidas.

Este cenário não se passará apenas nesta empresa, pois, como se viu no capítulo

anterior, não existem muitas soluções integradas para esta área e, portanto, é de esperar que o cenário encontrado neste empresa tenha paralelo noutras empresas da mesma área.

## 4.3. Sistema proposto

Como se viu na secção anterior, o cenário não é o mais favorável para a otimização dos processos da empresa. Além de as aplicações não terem qualquer integração uma com as outras, a outra aplicação que está a ser utilizada poderá ter comportamentos imprevisíveis ao longo do tempo, devido à atualização do sistema operativo.

Nesta secção ir-se-á mostrar uma nova proposta de software, para tentar corrigir e resolver os problemas atrás indicados. Será mostrada uma visão geral da aplicação, sem apresentar todos os detalhes ao pormenor, pois as tabelas de caso de uso, detalhando todos os aspetos a implementar da aplicação, serão poderão ser consultadas nos anexos 1, 2, 3 e 4.

Resta ainda dizer que os requisitos aqui contidos foram obtidos através da condução de entrevistas com o cliente e também através da análise dos documentos por eles criados para responder às suas necessidades profissionais.

### 4.3.1 Visão geral

#### 4.3.1.1 Núcleo principal

##### ***Mensagens***

Nesta secção poder-se-ão ver as mensagens enviadas por diversas entidades, sejam eles o sistema em si (gerando alertas automáticos, ou programados) ou outros utilizadores.

Sendo assim, podemos ver, da esquerda para a direita e de cima para baixo uma caixa com os *Alertas* do sistema, as *Mensagens*, as *Tarefas* e os *Processos* atribuídos.

Os *Alertas* são mensagens programadas em diferentes zonas da aplicação que avisam o utilizador de determinados eventos ocorridos no sistema (por exemplo, a entrada de um novo processo que está associado ao utilizador em questão). Carregando no título de cada alerta poderemos ver o texto que lhe está associado e existe ainda a hipótese de apagar alertas que já não sejam necessários.

As *Mensagens* são missivas enviados por outros utilizadores do sistema para o utilizador que se encontra autenticado nessa sessão. Tal como nos *Alertas*, ao carregar no título podemos ver o conteúdo de uma determinada mensagem. Além disso também se poderá apagar, responder, reencaminhar e criar/enviar novas mensagens para outros utilizadores do sistema.

As *Tarefas* são, tal como o nome indica, uma incumbência determinada por um outro utilizador para o utilizador que está autenticado nesse momento. Pode-se aceder ao conteúdo da tarefa carregando no respetivo título, sabendo-se assim qual o conteúdo da tarefa e quem a criou.

A última janela lista os *Processos* que estão atribuídos a esse utilizador, podendo-se aceder diretamente aos dados do processos e realizar todas as operações necessárias sobre eles (apagar, preencher, mudar estado, etc).

## **Definições**

As definições do sistema podem ser acedidas através do ícone que se encontra no lado esquerdo do ecrã, no canto inferior.

Através deste interface podemos aceder às opções de configuração deste sistema. Temos então a nossa lista dividida em quatro secções principais, os *Utilizadores*, o *Portal*, os *Processos* e o *Sistema*.

Na primeira opção da primeira secção podemos listar os *Utilizadores* (primeiro botão; nessa listagem também existe a opção de adicionar, modificar os dados ou apagar um utilizador), adicionar utilizador (segundo botão) ou adicionar/modificar/apagar privilégios de acesso ao sistema.

Na secção *Portal* temos um conjunto de opções muito similares à secção de *Utilizadores* mas, para este caso, virado para utilizadores externos à empresa, ou seja, clientes que estejam interessados em aceder a dados específicos, mas que não seja trabalhadores da empresa que utiliza a aplicação. Sendo assim pode-se listar os utilizadores externos (primeiro botão; nessa listagem também existe a opção de adicionar, modificar os dados ou apagar um utilizador externo), adicionar um utilizador externo (segundo botão) e adicionar/modificar/apagar privilégios de acesso ao sistema.

Na secção seguinte, *Processos*, temos um ícone que nos permite definir o *workflow* dos processos. Nesta definição, pode-se definir que tipo de comportamentos o sistema irá ter (alertas, envio de emails, etc) sempre que se muda o *estado*<sup>16</sup> de um processo.

Por fim, em *Sistema*, vamos ter o acesso aos *logs* da aplicação. Nos *logs* serão escritas todas as informações e ações relevantes realizadas sobre o programa.

## **Perfil**

O acesso ao *Perfil* pode ser feito através do terceiro botão a contar da direita, no topo do interface.

---

16 O estado de um processo indica qual o ponto de avanço em que ele se encontra, desde o seu início até ao seu encerramento.



Nesta secção, o utilizador poderá modificar os seus dados pessoais, que se irá refletir naquilo que os outros utilizadores vêm acerca do utilizador presente.

Além dos dados, também aqui o utilizador poderá modificar a sua chave de acesso ao sistema.

## **Pesquisa**

A *Pesquisa*<sup>17</sup> do sistema pode ser acedida através através no segundo botão a contar do lado direito, no topo do interface.

Aqui o utilizador pode definir em que parte da aplicação (relatórios, contactos, etc) pretende efetuar a sua pesquisa. Além disso poderá escolher os campos nos quais pretende efetuar essa pesquisa.

Os resultados irão aparecer no quadro, no lado direito dos dados a serem pesquisados.

### **4.3.1.2 Módulo de contactos**

#### **Contactos**

A secção de *contactos* deste projeto de estágio pode ser acedida carregado no botão com a forma de um caderno de contactos que se encontra no retângulo cinzento do lado esquerdo, o segundo a contar de cima.

Aqui pode-se gerir os contactos individuais, bem como os contactos coletivos, que estão associados à empresa que utiliza a aplicação, qualquer se seja o seu papel relativamente a eles. Como se pode ver através da imagem, por baixo dos botões de perfil e pesquisa encontram-se outros dois, sendo o que está do lado esquerdo o botão para acesso ao interface de contactos individuais e o que está do lado direito o botão para acesso aos contactos coletivos.

Nos contactos individuais pode-se introduzir os dados mais relevantes relativamente a pessoas que estejam ligadas à empresa. Além disso, é possível fazer a gestão desses contactos adicionando, editando e removendo contactos.

A secção de contactos coletivos encontra-se mais vocacionada para grupos de contactos, ou seja, empresas ou outras entidades que representam um aglomerado de contactos.

Sendo assim, nesta secção, também será possível preencher os dados relevantes para este tipo de contacto. Além disso, obviamente, a gestão será feita, sendo possível criar, apagar, e editar contactos coletivos.

Estando apresentados este tipo de contactos coletivos, é adicionada aqui uma nota. Nos utilizadores individuais existe um campo onde é possível indicar a que

<sup>17</sup> Embora seja aqui chamado apenas de “Pesquisa”, esta secção é, na verdade, uma pesquisa geral sobre o sistema. Outros módulos podem “adicionar” aqui os seus parâmetros de pesquisa, permitindo assim que o utilizador faça pesquisas sobre o sistema.

contacto coletivo um determinado contacto individual pertence, permitindo assim a “integração” entre estes dois tipos de contactos.

### 4.3.1.3 Módulo de Processos

#### **Processos**

A secção de *Processos* pode ser acedida carregando no ícone que se encontra por baixo dos *contactos*.

Esta é a secção mais importante para este estágio, pois é fundamental para a área de peritos de seguros a sua existência. Aqui pode-se construir relatórios de peritagens, pedidos por seguradoras, aquando da ocorrência de sinistros.

Nele estão contidos todos os elementos necessários para a realização de relatórios preliminares e finais acerca de uma determinada ocorrência. Além disso é possível a identificação da pessoa responsável pela investigação, as pessoas responsáveis pelo início do processo, testemunhas e outros possíveis envolvidos.

Caso seja necessário, poder-se-á agregar ao processo documentos, fotografias e outro tipo de ficheiros que sejam relevantes para o processo em questão.

Tal como foi referido anteriormente, também é possível modificar o estado em que se encontra um processo, de modo a que sejam notificados utilizadores ou sejam realizadas outras ações de acordo com os *workflows* definidos.

Além disso, um outro aspeto importante contemplado é a criação de documentos para envio e impressão através de *templates* pré-definidos. Assim sendo, os utilizador podem apenas preencher os campos necessários na aplicação e o documento intermédio e final será gerado a partir desses dados.

Finalmente, como é óbvio, tarefas como criação, remoção e edição de processos também se encontram contempladas.

### 4.3.2 Requisitos funcionais

Na tabela 2 são listados os requisitos funcionais para este sistema e, após essa listagem é feita uma breve apresentação de cada um deles.

ID	Requisitos funcionais	Prioridade
<b>Sistema geral (Anexo 1)</b>		
UC001 - 1	Autenticação de Utilizador	<b>Must</b>
UC002 - 1	Recuperação de Palavra-Chave de Utilizador	
UC003 - 1	Modificação dos dados pessoais	

ID	Requisitos funcionais	Prioridade	
UC004 - 1	Pesquisa geral		
UC005 - 1	Aceder às definições do sistema		
UC006 - 1	Listar utilizadores do sistema		
UC007 - 1	Editar utilizador		
UC008 - 1	Adicionar utilizador		
UC009 - 1	Apagar utilizador		
UC010 - 1	Listar nível de acesso de utilizadores		
UC011 - 1	Editar nível de acesso de utilizadores		
UC012 - 1	Adicionar nível de acesso de utilizadores		
UC013 - 1	Apagar nível de acesso de utilizadores		
UC014 - 1	Listar utilizadores externos		
UC015 - 1	Editar utilizador externo		
UC016 - 1	Adicionar utilizador externo		
UC017 - 1	Apagar utilizador externo		
UC018 - 1	Listar nível de acesso de utilizadores externos		
UC019 - 1	Editar nível de acesso de utilizadores externos		
UC020 - 1	Adicionar nível de acesso de utiliz. externo		
UC021 - 1	Apagar nível de acesso de utilizadores externo		
UC022 - 1	Visualizar <i>logs</i>		
UC023 - 1	Terminar sessão		
<b>Mensagens (Anexo 2)</b>			
UC001 - 2	Mostrar <i>dashboard</i> de mensagens		<b>Could</b>
UC002 - 2	Mostrar alertas		
UC003 - 2	Apagar alerta		
UC004 - 2	Listar mensagens		
UC005 - 2	Ver mensagem		
UC006 - 2	Responder à mensagem		
UC007 - 2	Reencaminhar a mensagem		
UC008 - 2	Apagar mensagem		
UC009 - 2	Criar nova mensagem		

ID	Requisitos funcionais	Prioridade
UC010 - 2	Listar as minhas tarefas	
UC011 - 2	Ver minha tarefa	
UC012 - 2	Listar tarefas atribuídas	
UC013 - 2	Editar tarefa atribuída	
UC014 - 2	Criar nova tarefa atribuída	
UC015 - 2	Apagar tarefa atribuída	
<b>contactos (Anexo 3)</b>		
UC001 - 3	Listar contactos Individuais	<b>Must</b>
UC002 - 3	Ver contacto Individual	
UC003 - 3	Adicionar contacto Individual	
UC004 - 3	Editar contacto Individual	
UC005 - 3	Apagar contacto Individual	
UC006 - 3	Listar contactos de Empresa	
UC007 - 3	Ver contacto de Empresa	
UC008 - 3	Adicionar contacto de Empresa	
UC009 - 3	Editar contacto de Empresa	
UC010 - 3	Apagar contacto Empresa	
<b>Processos (Anexo 4)</b>		
UC001 - 4	Listar Processos	<b>Must</b>
UC002 - 4	Adicionar processo	
UC003 - 4	Editar processo	
UC004 - 4	Apagar processo	
<b>App Mobile (Anexo 5)</b>		
UC001 - 5	<i>Login</i>	<b>Could</b>
UC002 - 5	<i>Logout</i>	
UC003 - 5	Listar processos	
UC004 - 5	Editar processo	

Tabela 2: Tabela de requisitos funcionais

O número indicado a seguir a cada ID de caso de uso indica o anexo onde ele se

encontra.

Como se pode ver pela tabela apresentada, foi utilizada a técnica MoSCoW [41] para avaliar a importância relativa de cada um dos itens apresentados, com a prioridade a ser atribuída, do mais elevado para o menos elevado, por “*Must*”, “*Should*”, “*Could*” e “*Won't*” (*have*).

Para mais informações acerca de cada um destes requisitos, no topo de cada secção é indicado o anexo que deverá ser consultado.

### **4.3.2.1 Sistema geral**

#### ***Autenticação de Utilizador***

Dado que este sistema precisa de proteção ao nível dos acessos que são efetuados, é necessário que exista um interface para que o utilizador introduza as suas credenciais (nome de utilizador a palavra-chave) para poder ter acesso ao sistema.

#### ***Recuperação de Palavra-Chave de Utilizador***

É necessária a implementação de um interface que permita a um utilizador que se tenha esquecido da sua password e possibilidade de reobter as suas credenciais de acesso, sendo-lhe enviado uma mensagem para o seu endereço de email. Para proteção extra, será também necessário o preenchimento de um campo com um *captcha*.

#### ***Modificação dos dados pessoais***

Dado que estes são colaboradores<sup>18</sup> de uma empresa, o fornecimento de um suporte para que os dados pessoais tais como telefone, morada, etc, sejam guardados, é relevante.

Sendo assim, este interface permitirá ao utilizador que ele modifique os seus dados sempre que necessário, bem como a sua palavra-chave, caso veja necessidade disso.

#### ***Pesquisa geral***

Dado que em determinado momento a informação contida será difícil de gerir sem um sistema de pesquisa, este requisito é de extrema importância.

Sendo assim, o sistema a desenvolver deve possibilitar a pesquisa sobre todos os conteúdos da aplicação. Dados contidos nos diversos componentes da aplicação (utilizadores, contactos, processos, etc) poderão ser alvo de pesquisa, de modo a que, o ato de encontrar informação relevante, seja uma tarefa simples e rápida.

---

<sup>18</sup> Entende-se por colaborador um profissional assalariado que trabalhe para e empresa que está a usar este sistema.

### ***Aceder às definições do sistema***

Dado que em determinado momento, consoante o conjunto de módulos que o sistema possa vir a ter<sup>19</sup>, é importante ter um interface que permita escolher quais as definições que às quais se quer aceder, de uma forma visualmente simples. Sendo assim, terá de ser implementado neste projeto uma função com essas características.

### ***Editar utilizador***

Deve existir no sistema um meio que permita edição dos dados de um determinado utilizador e que permita também a modificação das credenciais de acesso dele.

### ***Adicionar utilizador***

Sendo que o sistema é povoado por utilizadores que realizam os suas actividades profissionais, este deve ter a capacidade de adicionar novos utilizadores, conforme necessário.

### ***Apagar utilizador***

Dado que um determinado utilizador pode deixar de trabalhar para uma empresa, ou pura e simplesmente deixar de utilizar esta plataforma, é um requisito necessário que exista um meio para remover utilizadores.

### ***Listar nível de acesso de utilizadores***

Os acessos ao sistema por parte de diversos utilizadores pode não ser o mesmo. Se um utilizador pode ter acesso a determinadas partes do sistema, isso não quer dizer que um outro utilizador também possa ter os mesmos.

Sendo assim, terá de ser implementado um meio para que se possam criar níveis de acesso ao sistema, cada um com as suas especificidades, de modo a que estes possam ser atribuídos a utilizadores específicos.

Assim sendo, deverá ser possível também listar esses níveis de acesso, de modo a que a sua edição e remoção sejam tarefas simples.

### ***Editar nível de acesso de utilizadores***

Estando cientes da existência de níveis de acesso, o sistema deverá permitir que eles possam ser editados com diferentes tipos de privilégios de acesso.

### ***Adicionar nível de acesso de utilizadores***

Conforme o que foi relatado nos dois requisitos anteriores, torna-se claro que é necessário que o sistema permita a adição de novos níveis de acesso.

---

<sup>19</sup> Mesmo sendo este estágio relativo a um software para a área de peritos de seguros, é um requisito por parte do estagiário que o sistema possa um dia vir a ser povoado por um outro conjunto de módulos para diferentes áreas profissionais.

### **Apagar nível de acesso de utilizadores**

Se um determinado nível de acesso deixa de ser utilizado, torna-se necessária a sua remoção. Sendo assim, é necessária a implementação de um meio que torne isto possível.

### **Listar utilizadores externos**

Já antes foi referida a existência destes utilizadores externos. São elementos que, embora não exerçam a sua profissão na empresa que utiliza este sistema, necessitam de acesso a dados nele contidos (clientes que necessitem de ter acesso aos seus processos). É então necessário a criação de um interface que permita que a gestão destes utilizadores seja simples e eficiente.

### **Editar utilizador externo**

Sempre que necessário, deve ser possível editar os dados de um determinado utilizador externo, seja para modificação de dados pessoais, ou até mesmo das credenciais de acesso.

### **Adicionar utilizador externo**

Para que os utilizadores externos possam ter acesso ao sistema, é necessário, primeiramente a sua inserção no sistema, sendo portanto necessário a implementação deste aspeto.

### **Apagar utilizador do portal**

Se um determinado utilizador externo tiver de deixar de ter acesso ao sistema, é necessário que seja possível a sua remoção. Este requisito diz respeito a esse aspeto.

### **Listar nível de acesso de utilizadores externos**

Tal como para os utilizador “normais”, também os utilizadores externos terão de estar sujeitos a um determinado nível de acesso ao sistema. Para facilitar a diferenciação entre estes dois tipos de utilizadores, é também diferenciado o sistema de níveis de acesso.

Sendo assim, torna-se premente a existência de um interface que permita a gestão deste aspeto do sistema.

### **Editar nível de acesso de utilizadores externos**

Tal como para o outro tipo de níveis de acesso, será necessário implementar um meio para que estes níveis de acesso possam ser editados.

### **Adicionar nível de acesso de utilizadores externos**

Para que possam existir níveis de acesso no sistema, é necessário inicialmente que estes sejam adicionados.

### **Apagar nível de acesso de utilizadores externos**

Caso um determinado nível de acesso deixe de ser necessário, é necessário que existam meios para este possa ser apagado.

### **Visualizar logs**

Sendo que num sistema deste tipo é necessário algum tipo de *logging*<sup>20</sup>, é necessária a implementação deste aspeto. Além disso, é necessário um interface que permita visualização fácil deste aspeto da aplicação.

### **Terminar sessão**

Quando um utilizador termina de usar o sistema, não faz sentido que deixa a sua sessão iniciado, pois outro utilizador pode usar a sua sessão com intenções menos benignas e modificar dados do sistema sem o seu consentimento, podendo provocar danos.

Sendo assim torna-se necessária a implementação de um método que permita ao utilizador finalizar a sua sessão.

## **4.3.2.2 Mensagens**

### **Mostrar dashboard de mensagens**

Dado que existem vários tipos de mensagens no sistema, torna-se interessante ter um meio que mostre as diversas mensagens num único interface. Sendo assim, é necessário implementar na aplicação um *dashboard* que permita ver, de uma só vez, todas as mensagens novas que um utilizador recebeu.

### **Mostrar alertas**

Além do *dashboard*, é conveniente também a realizar a implementação de um interface que permita ver apenas os alertas e filtrar a lista para ver mais facilmente apenas aqueles que necessitam de ser visualizados.

### **Apagar alerta**

Quando um determinado alerta deixa de ser necessário, porque a sua utilidade deixou de existir, deve haver um meio para efetuar a sua remoção permanente do sistema, por parte do utilizador.

### **Listar mensagens (pessoais)**

Tal como os alertas, deve existir um meio para listar as mensagens pessoais que foram enviadas para um determinado utilizador, dando ainda acesso a filtros que permitam visualizar apenas a lista de mensagens pretendidas. Sendo assim, torna-se necessária a implementação desta funcionalidade.

---

20 Registo de ações dos utilizadores e do sistema



### **Ver mensagem**

A partir do momento que se tem uma lista de mensagens, não fará sentido que o seu conteúdo não seja visualizável. Deve então ser implementado um interface que mostre o conteúdo da mensagem, bem como por quem foi enviada e quais os seus anexos.

### **Responder à mensagem**

Muitas vezes, as mensagens recebidas necessitarão de uma resposta para o seu remetente. Sendo assim, a implementação de um mecanismo e de um interface para esta tarefa torna-se necessário.

### **Reencaminhar a mensagem**

Quando se recebe uma mensagem, também pode ser necessário, em vez de responder, enviá-la para outra pessoa que não o remetente. Sendo assim, deverá ser implementado um interface para este aspeto.

### **Apagar mensagem**

As mensagens, além de ocuparem espaço, podem deixar de ser necessárias, pois o seu âmbito deixa de existir e o conteúdo deixa de ser necessários. A implementação de um método de remoção de mensagens é um requisito necessário.

### **Criar nova mensagem**

Para que existam mensagens, é necessário criar e definir um destinatário, além de definir que ficheiros de anexo se querem adicionar. Será então um requisito criar um meio para que esta ação possa ser efetuada.

### **Listar as minhas tarefas**

Mais uma vez, embora o *dashboard* possa ser um ótimo meio para mostrar as tarefas mais recentes, é necessário que haja um interface que permita listar e filtrar as tarefas que foram atribuídas a um determinado utilizador.

### **Ver minha tarefa**

Sendo assim, o sistema deve possibilitar que o utilizador possa ver os conteúdos de um tarefa que lhe foi atribuída.

### **Listar tarefas atribuídas**

Tal como um utilizador pode receber tarefas por parte de outros utilizadores, também as poderá atribuir. O sistema deve permitir que um utilizador crie novas tarefas, editando o seu conteúdo e podendo adicionar ficheiros de anexo, se necessário.

### ***Editar tarefa atribuída***

Caso seja detetado um erro, ou se queira adicionar mais informação a uma determinada tarefa atribuída, é necessário que o sistema forneça um meio para que isto seja possível.

### ***Criar nova tarefa atribuída***

E é claro, para que haja novas tarefas, o sistema deve permitir a sua criação, fornecendo o meio para que um utilizador crie uma nova tarefa, com um determinado conteúdo e ficheiros anexos.

### ***Apagar tarefa atribuída***

Quando uma tarefa é concluída, ou deixa de ser necessária, o seu criador pode então apagar aquela tarefa para que esta deixe de aparecer na listagem.

## **4.3.2.3 Contactos**

### ***Listar contactos Individuais***

Para que possam ser guardadas informações importantes sobre pessoas relacionadas com a empresa, é necessário implementar um sistema que armazene os contactos, com todos os dados relevantes. Assim sendo, também será necessária a implementação de um interface que permita visualizar esses contactos e filtrar esses dados conforme for necessário.

### ***Ver contacto Individual***

Como todos os dados contidos num contacto não poderão ser visualizados apenas na lista (apenas um conjunto de campos irá aparecer), é necessária a criação de um interface para visualização de um contacto individual.

### ***Adicionar/Editar contacto Individual***

Sem a possibilidade de adição de um contacto, não haveria novos contactos no sistema. Assim sendo, o sistema deve possibilitar a adição (e, obviamente, edição) de contactos.

### ***Listar contactos de Empresa***

Tal como se passa com os contactos individuais, o mesmo deve suceder com os contactos de empresa (ou coletivos). O sistema deve permitir que o utilizador possa aceder a uma listagem bem organizado (e filtrar resultados) da lista deste tipo de contactos.

### ***Ver contacto de Empresa***

Mais uma vez, o sistema deve possibilitar a visualização de um contacto de

empresa, de modo a que todos os dados que lhe dizem respeito possam ser mostrados ao utilizador.

### ***Adicionar/Editar contacto de Empresa***

Sem a possibilidade de adição de um contacto de empresa, não haveria novos contactos de empresa no sistema. Assim sendo, tal como para os contactos individuais, o sistema deve possibilitar a adição (e, obviamente, edição) de contactos.

### ***Apagar contacto (Individual ou de Empresa)***

Caso um contacto deixe de ser necessário, a aplicação deve permitir que este seja removido do sistema. Sendo assim, deverá ser implementado um meio para que o utilizador possa realizar esta ação.

## **4.3.2.4 Processos**

### ***Listar Processos***

Para que os processos possam ser geridos de uma forma eficiente, é necessária que o sistema possua um meio de os listar de uma maneira organizada e fácil de visualizar, possibilitando o uso de filtros. Sendo assim, o sistema deverá possuir esta possibilidade.

### ***Adicionar processo***

Para que existam processos no sistema, é necessário em primeiro lugar adicioná-los. Sendo assim, o sistema deve ser capaz de fornecer ao utilizador a capacidade de adição de novos processos.

### ***Editar processo***

Como os dados de um determinado processo guardado podem ter de ser modificados, deve ser implementado no sistema um meio para editar os campos de um item deste género.

### ***Anular processo***

Como um determinado processo pode ter sido criado por engano, ou até deixado de ser necessário, o sistema deve possuir a capacidade de permitir que um determinado utilizador o possa anular.

## **4.3.2.5 App Mobile**

### ***Login***

Para que o sistema esteja protegido contra acessos indevidos, de cada vez que um utilizador se liga ao sistema através desta aplicação móvel, terá de preencher as suas

credenciais.

### ***Logout***

Tal como deve existir um mecanismo para realizar a autenticação, também deverá existir um mecanismo complementar para finalizar a sessão.

### ***Listar processos***

A listagem de processos deve-se efetuar de uma forma similar à aplicação web, permitindo a filtragem de resultados e a visualização de uma lista de processos relevantes para o utilizador apresentados de uma forma visualmente organizada.

### ***Editar processo***

Esta aplicação terá uma versão reduzida do conjunto de funcionalidades existentes na aplicação web. Apenas estarão disponíveis as opções para adicionar imagens ao processo e outros ficheiros anexos.



Na imagem 7 podemos ver o diagrama de casos de uso para os requisitos funcionais que foram referidos. A ligação entre cada um dos elementos reflete o “caminho” a ser percorrido para se poder efetuar uma determinada ação no sistema. Para uma melhor visualização do diagrama, foi utilizado um esquema de cores onde o azul representa os aspetos o “Sistema geral”, o amarelo representa a secção de “Mensagens”, o verde representa os casos de uso dos “contactos” e o vermelho o dos “Processos”.

#### 4.3.4 Requisitos não-funcionais

Nesta secção são indicadas as principais preocupações a nível de requisitos não funcionais. O âmbito será apenas de descrever os atributos de qualidade pelo quais a aplicação se pautará e não as estratégias para os consolidar. Estas serão apresentadas no próximo capítulo, sobre a arquitetura do sistema.

Na tabela que se apresenta de seguida, na primeira coluna estará identificado o atributo de qualidade, na segunda coluna a preocupação levantada por esse atributo e na terceira coluna o grau de importância que esse atributo terá: *High*, *Medium* ou *Low* (Alto, Médio ou Baixo).

Os dados contidos nesta tabela foram efetuados em conjunto pelo estagiário autor deste documento e o estagiário B (Marco Pedrosa).

Atributo de Qualidade	Preocupação	Importância (High/Medium/Low)
<b>Segurança</b>	Autenticação obrigatória para a maioria das funcionalidades	<b>High</b>
<b>Segurança</b>	Proteção de dados armazenados (confidencialidade)	<b>High</b>
<b>Segurança</b>	Proteção das comunicações entre o cliente e o servidor (confidencialidade)	<b>High</b>
<b>Segurança</b>	Proteção do sistema a ataques informáticos	<b>High</b>
<b>Modificabilidade</b>	Separação em módulos e redução das inter-dependências	<b>High</b>
<b>Modificabilidade</b>	Permitir inserção de novas funcionalidades em módulos e novos módulos sem existir complexidade excessiva	<b>High</b>
<b>Disponibilidade</b>	Deteção e Recuperação de falhas (automática quando possível)	<b>High</b>
<b>Auditoria</b>	Todas as operações que efetuem mudanças nos dados devem ser registados num sistema de <i>Logging</i>	<b>Medium</b>
<b>Desempenho</b>	Tempo de Resposta	<b>Medium</b>
<b>Usabilidade</b>	Utilização deve apresentar uma baixa curva de	<b>Low</b>

	aprendizagem	
--	--------------	--

Tabela 3: Atributos de Qualidade e sua priorização

## **Segurança**

Dado que se trata de uma aplicação que vai lidar com dados privados e sensíveis relativos aos clientes e também dados críticos da empresa que utiliza a aplicação, a segurança é um atributo de qualidade de extrema importância.

Assim sendo, terão de ser contemplados aspetos e implementados mecanismos para resolver questões tais como a autenticação obrigatória para efetuar qualquer ação sobre o sistema, proteção de comunicações entre cliente e servidor e dados da base de dados usando encriptação e proteção de sistemas contra ataques informáticos através de, por exemplo, *firewalls*.

## **Modificabilidade**

Neste momento, este estágio encontra-se vocacionado para a criação de um software para resolver as questões relativas a empresas do ramo de regulação de sinistros. No entanto, estamos apenas a lidar com as necessidades e métodos de trabalho de um empresa o que, embora valide um conjunto de requisitos que serão necessários inter-empresas, correrão o risco de não focar aspetos e necessidades específicas de outra empresa. Portanto, além da preocupação fundamental de realizar uma aplicação que seja o mais geral possível, existe a necessidade também que esta seja o mais modificável possível.

Um outro aspeto que terá de se ter em conta é o plano futuro para esta plataforma, não é seu objetivo cingir-se apenas a esta área mas sim, um dia, através da escolha de módulos específicos, poder ser usada em diversas áreas profissionais.

Terão então de ser contemplados aspetos tais como a separação em módulos da aplicação e a redução ao máximo da interdependência entre eles, além da preocupação em criar módulos cuja complexidade não seja excessiva de modo a que as modificações que possam vir a ser necessário efetuar sejam realizadas o mais rápida e eficazmente possível.

## **Disponibilidade**

Num mercado cada vez mais exigente de resultados e competitivo, a questão da disponibilidade é de grande importância por dois motivos.

Em primeiro lugar, se não houver grande disponibilidade do sistema, se este estiver em “baixo” um grande número de vezes, isso irá afetar a empresa que necessita de efetuar o seu trabalho e que dependerá desta aplicação para o poder realizar.

Em segundo lugar, se o sistema não estiver disponível sempre que necessário, uma determinada empresa não terá outra hipótese senão optar por uma solução que lhe aumente a produtividade.

Tendo isto em vista, aspetos como deteção e recuperação de falhas terão de ser pensados e implementados, possivelmente através de utilização de servidores alternativos e *watchdogs*.

### **Auditoria**

Num sistema deste género, onde existe um grande fluxo de dados e de ações, é conveniente saber quem realizou uma determinada ação, de modo a que, ações indevidas possam ser identificadas e resolvidas.

Sendo assim, todas as ações produzidas (ou, pelo menos, as mais importantes), devem ser registadas na base de dados através de um sistema de *logging*.

### **Desempenho**

O desempenho também é um aspeto importante a ter em conta, visto que, se a aplicação se mostrar lenta e irresponsiva, isso irá ter impacto na produtividade dos colaboradores da empresa que utiliza o sistema no seu dia a dia (além de gerar frustração nos seus utilizadores).

### **Usabilidade**

Quando um determinado utilizador começa a lidar com um novo software, existe uma curva de aprendizagem envolvida. Essa curva de aprendizagem irá ter impacto na produtividade dos utilizadores e, se a aplicação apresentar um interface complicado e/ou incoerente, por muito que possa ter todas as funcionalidades necessárias para a resolução dos problemas de uma determinada empresa, não irá ser uma mais valia e será uma ferramenta que se descartará num período de tempo reduzido.

Sendo assim, a usabilidade será um aspeto a ser levado em conta na implementação desta aplicação.

## **4.3.5 Conclusão**

Neste capítulo foram identificados os requisitos funcionais e não funcionais pelos quais este projeto se guiará na sua construção.

Além disso, tendo em conta também os atributos de qualidade, teremos identificados todos os aspetos necessários para que se possa analisar e construir a sua arquitetura.



# Capítulo 5

## Arquitectura e especificação

Depois de efetuada a análise de requisitos contendo as funcionalidades que o sistema deve conter e os atributos de qualidade pelos quais se deve reger, neste capítulo é apresentada a sua arquitetura, onde deverá ser exposto de que modo esses objetivos serão cumpridos.

Tal como o que é descrito na proposta de estágio deste projeto, o objetivo é o desenvolvimento de uma aplicação *web-based*.

Nas ciências computacionais, uma *web application* ou *web app* é uma aplicação de software cliente-servidor, onde o cliente (ou interface de utilizador) corre num *browser*. Este tipo de aplicações são populares devido à ubiquidade dos *web browsers* e a conveniência da sua utilização como cliente, bem como a facilidade para fazer *updates* e manter aplicações web sem distribuir ou instalar software em, potencialmente, centenas ou milhares de computadores [42].

Querendo aproveitar todo o tipo de vantagens que este modelo aplicacional oferece, será utilizado no desenvolvimento deste projeto.

No anexo 6 é efetuada a justificação da escolha do Drupal para a implementação deste projeto.

Neste capítulo será adotada uma explicação da arquitetura em *zoom*, ou seja, passa-se de uma vista geral do sistema para os seus aspetos mais particulares.

Na secção 5.1 será mostrada a estrutura geral do sistema, com os componentes que se encontram em torno do software em si, representado pelo Drupal.

Na secção 5.2 será mostrada a arquitetura do software em si e tudo aquilo que foi herdado do Drupal, ao mesmo tempo que se mostra onde se enquadram, no meio da sua arquitetura, os módulos que se pretendem desenvolver.

Finalmente, na secção 5.3 será mostrado o modelo de dados, ou seja, o conjunto de tabelas necessárias à construção dos módulos propostos e conseqüente integração com o modelo de dados existentes do Drupal.

## 5.1 Estrutura Geral do Sistema

Começa-se por apresentar aqui a arquitetura geral do sistema. Não existe ainda a preocupação de uma descrição detalhada de cada um dos componentes, pretendendo-se apenas uma explicação genérica de cada um e de que modo atingir os requisitos não funcionais expostos no capítulo anterior. Mais à frente será apresentada uma explicação mais detalhada de cada um dos componentes e o motivo pelo qual foram escolhidos

Pode-se visualizar, através da vista de topo do sistema, na imagem 8, os seus principais componentes.

As ligações, tanto do cliente *web* como do cliente *mobile*, serão primeiramente encaminhadas pelo CloudFlare<sup>21</sup>. Estas ligações estarão encriptadas por SSL com uso de certificados digitais<sup>22</sup>.

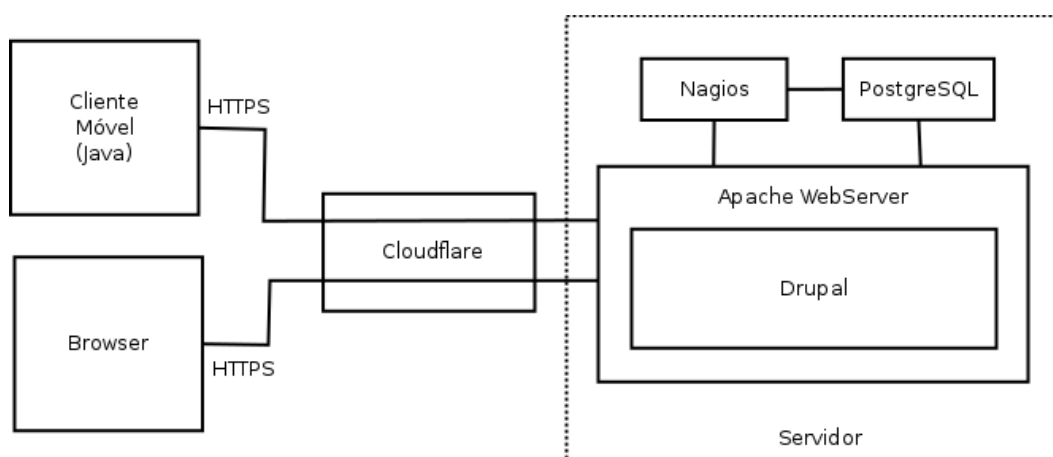


Imagem 8: Vista de topo do sistema

Para um cliente que se ligue através de um *browser*, a comunicação ir-se-á efetuar por HTTPS, tendo como destino o servidor HTTP Apache<sup>23</sup>, que irá ter acesso às páginas pretendidas, sendo utilizando para isso o Drupal. O estudo conduzindo à escolha desta framework para utilização neste projeto encontra-se no Anexo 6.

Por sua vez, os dados relativos à aplicação estarão guardados numa base de dados gerida pelo PostgreSQL<sup>24</sup>, gerindo o fornecimento e armazenamento dos mesmos.

Para fazer a monitorização dos componentes que se encontram no servidor, atrás referidos, optou-se pela utilização do Nagios<sup>25</sup>.

21 <https://www.cloudflare.com/>

22 <https://openvpn.net/>

23 <http://httpd.apache.org/>

24 <http://www.postgresql.org/>

25 <https://www.nagios.org/>

### 5.1.1 CloudFlare

A CloudFlare é uma companhia Norte-Americana que fornece uma rede de distribuição de conteúdos e serviços DNS distribuídos, encontrando-se entre o visitante e o utilizador do serviço, agindo como uma *reverse proxy* para *websites*. A sua rede protege, acelera e melhora a disponibilidade de um *website* ou aplicação móvel com modificações no DNS [43].

A sua utilização permite a proteção contra ataques DDNS, o que, caso sucedam, podem afetar de uma forma letal a disponibilidade e segurança do sistema proposto por este estágio.

Num cenário destes, o atacante reúne recursos, tal como *botnets* ou DNS *recursors* inseguros, e mimica o endereço IP do alvo. Os recursos reunidos enviam então uma grande quantidade de *replies* para o alvo, levando-o a ficar *offline* [44] (imagem 9).

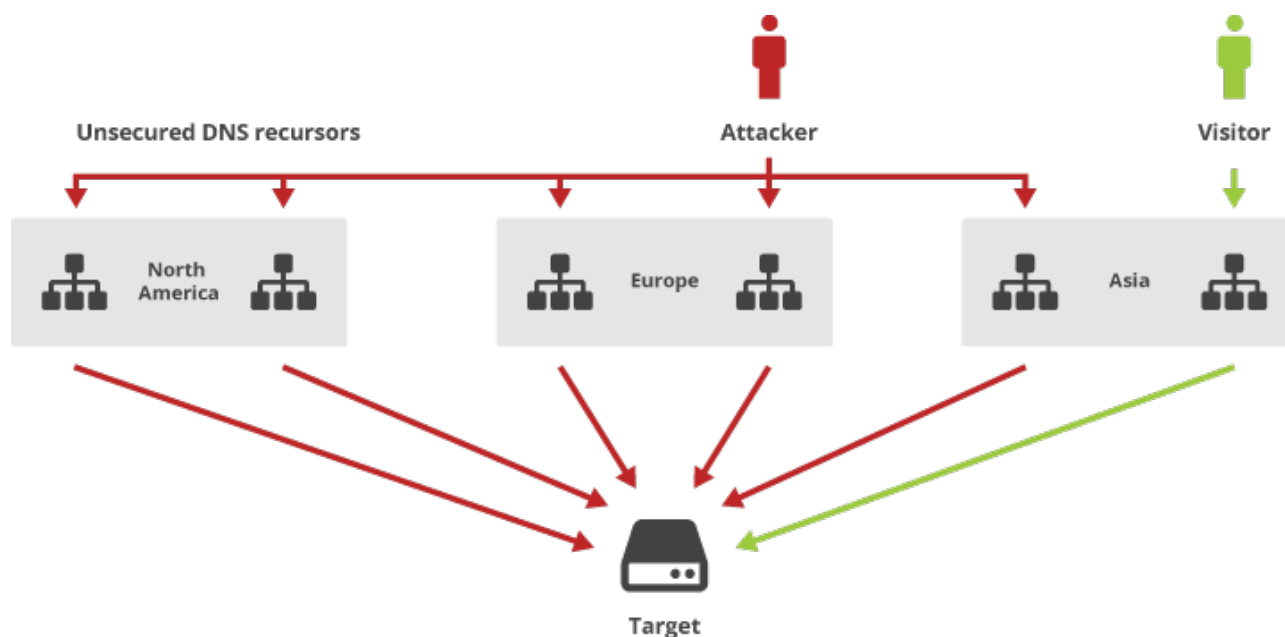


Imagem 9: Fonte: <https://www.cloudflare.com/ddos>

Num cenário onde o CloudFlare está presente, um atacante toma as mesmas medidas que anteriormente, mas os *replies* são bloqueados regionalmente pelos seus *data centers*. Tráfego legítimo, no entanto, continua a poder aceder com normalidade [44] (imagem 10).

Além disso, fornece proteção contra ataques SMURF<sup>26</sup>, ACK<sup>27</sup> e LAYER 7<sup>28</sup>.

Pelo fato deste serviço disponibilizado ser gratuito e pelas excelentes críticas que

26 [https://en.wikipedia.org/wiki/Smurf\\_attack](https://en.wikipedia.org/wiki/Smurf_attack)

27 [https://en.wikipedia.org/wiki/SYN\\_flood](https://en.wikipedia.org/wiki/SYN_flood)

28 [https://en.wikipedia.org/wiki/Application\\_layer\\_DDoS\\_attack](https://en.wikipedia.org/wiki/Application_layer_DDoS_attack)

recebeu [45], tornou-se aliciante a sua inserção neste projeto.

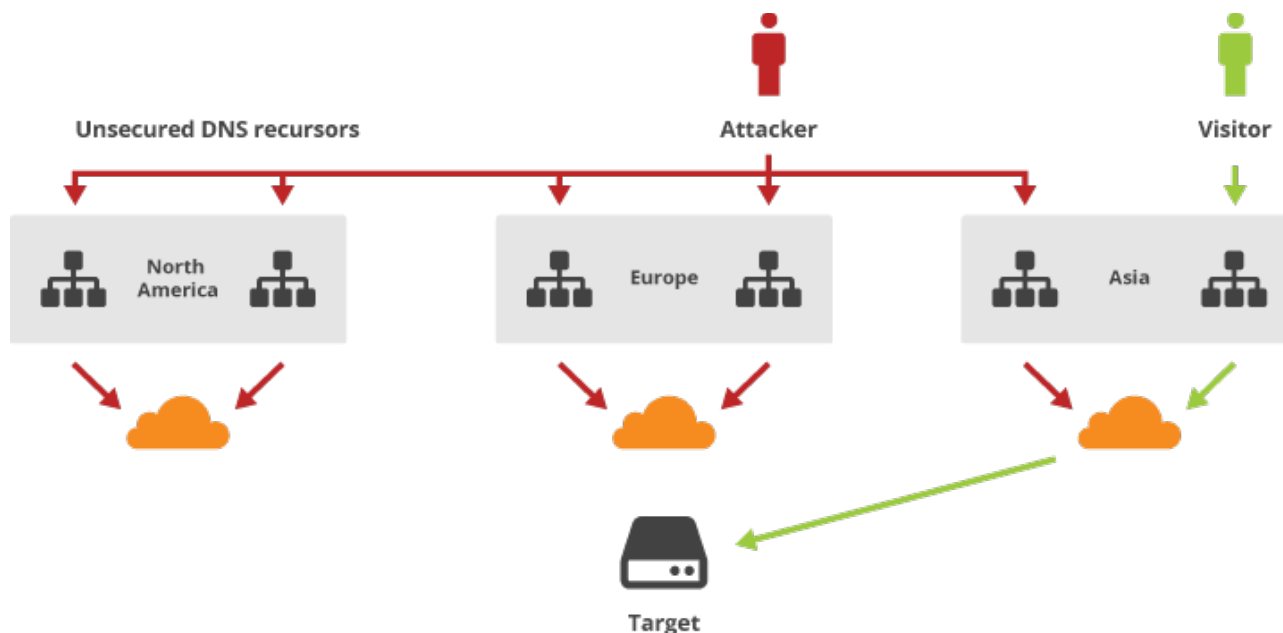


Imagem 10: Fonte: <https://www.cloudflare.com/ddos>

### 5.1.2 Nagios

É um software *open-source* para monitorização de sistemas, redes e infraestrutura. Oferece serviços de monitorização e alerta para servidores, aplicações e serviços, alertando quando alguma algum erro ocorre e quando o problema foi resolvido [46].

A sua principal função será a monitorização de estado (se está a correr, se se encontra bloqueado ou se pura e simplesmente terminou a sua execução) de componentes críticos do sistema [47] (conforme pode ser visualizado na vista de topo da arquitetura). Isto permitirá ao sistema apresentar uma maior disponibilidade do que se este componente não estivesse presente.

Dado ser um software com provas dadas e pelo fato de ser disponibilizado gratuitamente, foi escolhido para ser integrado no sistema.

### 5.1.3 PostgreSQL

PostgreSQL é um sistema de gestão de base de dados objeto-relacional com ênfase na extensibilidade e compatibilidade com os standards. Enquanto servidor de base de dados, tem como função primária o alojamento seguro de dados, com suporte de boas práticas, permitindo a sua obtenção a pedido de softwares externos. Consegue lidar com cargas desde softwares instalados localmente em máquinas singulares até aplicações de Internet com múltiplos utilizadores em ambientes concorrenciais. É desenvolvido pelo PostgreSQL Global Development Group, um grupo constituído por

muitas companhia e contribuidores individuais. É gratuito e *open-source*, sob a PostgreSQL Licence, uma licença de software-livre permissiva [48].

Tendo em vista todas estas características, este software permitirá um bom desempenho para gravação e obtenção de dados, além de garantir que os dados são guardados de uma forma segura, se se seguir boas práticas. Em termos de performance e escalabilidade, apresenta até melhores resultados que os seus competidores diretos [49][50].

Tendo todos estes aspetos em conta, este software mostrou ser a escolha acertada para a gestão da base de dados.

### 5.1.4 Apache HTTP Server

O Apache é o servidor web mais usado do mundo, desde Abril de 1996, e é desenvolvido por uma comunidade de programadores, sob os auspícios da Apache Software Foundation, sendo *open-source* (Apache Licence) [51].

Dado todos os anos de uso e boa experiência por parte dos utilizadores, e suportando um imenso conjunto de configurações e extensões, torna-se a opção mais indicada e confiável para integrar neste projeto.

Sendo que o Drupal é uma *framework* desenvolvida em PHP, o módulo relativo a esta linguagem também terá de estar instalado e activado no servidor web.

### 5.1.5 Drupal

Conforme foi discutido em pormenor no Anexo 5, a *framework* a ser utilizada para o desenvolvimento da aplicação será o Drupal.

Drupal é um Content Management System/Framework Open Source escrito em PHP. Esta *framework* disponibiliza os blocos necessários para a criação de um site sem grande necessidade de conhecimentos prévios de programação e fornece todas as ferramentas necessárias para o desenvolvimento de novos módulos e modificação dos já existentes.

Este CMS foi criado com o objetivo de ser multi funcional dentro do contexto da Web. Um dos seus maiores atrativos reside no fato de um programador poder utilizar a ferramenta com total liberdade desde os projetos mais simples, como uma *webpage* ou um *blog*, até aos mais complicados, como uma rede social, etc [52].

Esta *framework* possui uma licença GPL e é mantida por uma comunidade global liderada por Dries Buytaert, o seu criador que, juntamente com Jay Batson, fundou a empresa Acquia que se dedica a dar suporte e a fornecer produtos e serviços baseados em Drupal [53]. Atendendo à grande popularidade desta *framework*, existe todo um universo de empresas dedicadas ao desenvolvimento de módulos à medida e de conteúdos. Existe igualmente uma grande comunidade *open-source* que desenvolve módulos para esta plataforma, sendo estes acessíveis a partir do site

Drupal.org. Esta comunidade é formada por pessoas singulares ou equipas que se dedicam a complementar as funções existentes, ampliando o espectro de utilização desta ferramenta já em si bastante poderosa [54].

Sendo o Drupal utilizado para diversos fins, tal como já foi referido, a gama de entidades que o utilizam vão desde a Casa Branca, a Zynga e o Ikea, entre outras [54].

Apesar de apresentar sérias barreiras para novos utilizadores, o *Drupal* é considerado por muitos o melhor CMS do mercado. Detém uma boa performance, é tecnicamente mais avançado, modificável e gratuito. Para colmatar as dificuldades de usabilidade, apresenta um enorme conjunto de tutoriais e suporte por parte dos seus utilizadores. Nas mãos de um utilizador experiente evidencia grandes vantagens sobre a concorrência não apenas em termos de performance mas também no capítulo da extensibilidade [55].

Em suma, o Drupal apresenta-se como uma excelente escolha para utilização neste projeto de estágio. No entanto, dado que este CMS apresenta uma arquitetura própria, o software irá, por conseguinte, herdá-la. Convém portanto efetuar um estudo mais aprofundado sobre arquitetura do Drupal.

### **5.1.6 Cliente Móvel**

O cliente móvel será uma aplicação para dispositivos móveis que não será desenvolvida no âmbito deste estágio, por restrições temporais. No entanto é aqui referida dado que também foi especificada em pormenor na análise de requisitos.

## **5.2 Arquitetura do software**

Tendo sido feita uma descrição mais geral do sistema, e os componentes que rodeiam o software a ser realizado, convém dirigir maior pormenor a este último aspeto.

Tendo em conta uma perspetiva externa relativamente ao sistema, o Drupal é uma *framework* PHP e, como tal, necessita de um servidor Web, com o respetivo módulo de PHP ativado, para que o seu código possa ser executado. Os dados que com que o software irá lidar estarão guardados numa base de dados e, todos estes componentes, funcionarão sobre o sistema operativo que fará a comunicação com o hardware. Esta estratificação está representada através da imagem 11.

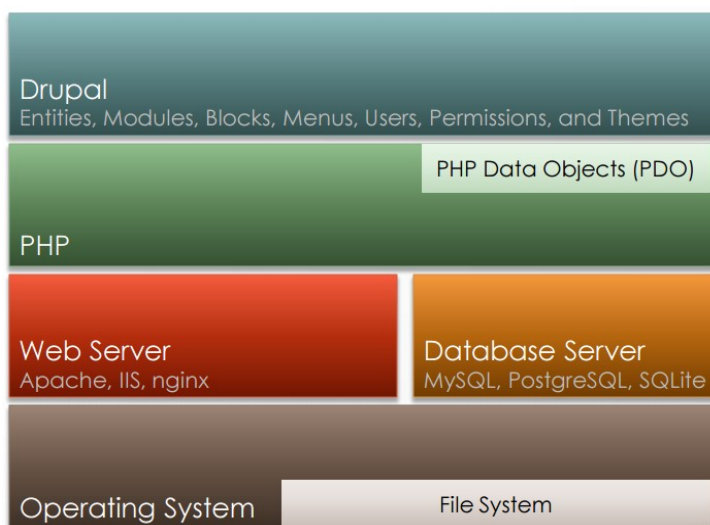


Imagem 11 (Fonte: <http://deninet.com/files/How%20Does%20Drupal%20%27%20Work.pdf>)

## 5.2.1 Padrão arquitetural

Uma página do *Drupal* é constituída por um conjunto de blocos. Cada bloco é uma entidade separada que retorna o seu output para a página que incorpora esses dados. Estes blocos representam as várias caixas que aparecem nas várias regiões do *Website*. Os mesmos podem ser utilizados para representar qualquer tipo de conteúdo, pelo que são utilizados como blocos de construção da página a apresentar. Para controlar o modo como a informação contida nos *nodes* é devolvida e apresentada nos blocos são utilizadas *Views* [52].

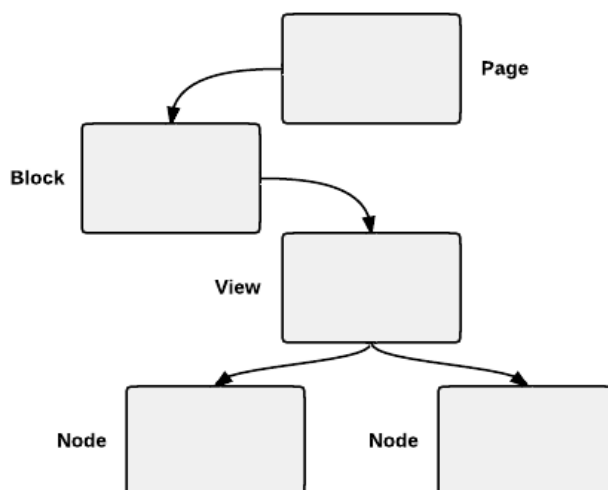


Imagem 12: Sistematização da arquitetura da Framework Drupal

(fonte: <http://robknight.org.uk/blog/2011/02/explaining-architectural-tiers-drupal/>)

Todos os conteúdos num *Website Drupal* estão armazenados e são tratados como *nodes*. Cada um destes representa um conteúdo individual. Tratando todos os conteúdos como *nodes*, permite uma flexibilização da criação de novos tipos de conteúdos e facilita a aplicação de novas características e alterações a todos os *nodes* de um tipo [52].

A arquitetura interna de cada um destes elementos é bastante semelhante. Eles estão divididos em três componentes:

- *Componente de abstração*, que obtém e processa os dados;
- *Componente de apresentação*, que gere o aspeto visual;
- *Componente de controlo*, que coordena aspetos tais como o fluxo de controlo e comunicação entre os outros dois componentes.

Este tipo de organização obedece ao padrão arquitetural Presentation-Abstraction-Control (PAC) [56].

O PAC é um padrão arquitetural de software orientado à interação, similar ao padrão Model-View-Controller (MVC). Em ambos os padrões, existe um sistema interativo constituído por três componentes que são responsáveis por diferentes aspetos específicos das funcionalidades da aplicação no entanto, o MVC não apresenta qualquer hierarquia de agentes e os seus constituintes são capazes de comunicar entre si [57].

As principais características do PAC são a utilização de uma estrutura hierárquica de agentes. Cada agente consiste numa tríade (apresentação, abstração e controlo), sendo que a componente de apresentação está isolada da de abstração, e ambas do exterior. Toda a comunicação interna é mediada pela componente de Controlo, bem como a comunicação entre agentes. Desta forma é melhorada a experiência de utilização da aplicação, dado que é possível mostrar o interface antes da componente de abstração ter sido totalmente inicializada [56].

## 5.2.2 Perspectiva em camadas do Drupal

Para melhor compreender o seu funcionamento, vai-se analisar uma perspetiva em camadas do CMS e o modo como a informação flui entre elas (Imagem 13).

Na base do sistema encontra-se uma coleção de *nodes* – a *pool* de dados. Antes de qualquer coisa poder ser mostrada no site, tem de ser introduzida como um dado [58].

É na camada seguinte que os módulos residem. Os **módulos** são *plugins* funcionais que ou fazem parte do núcleo do Drupal (são incluídos na sua raiz), ou são contribuídos pelos membros que constituem da comunidade. Os módulos modificam a funcionamento do núcleo do Drupal, permitindo a modificação de items de dados



(campos) nos diversos tipos de *nodes*, fazer sistemas de *e-commerce*, programaticamente ordenar e mostrar o conteúdo e muito mais. Existem milhares de diferentes opções nos repositórios de contribuições de Drupal [58].

Na próxima camada encontram-se os blocos e os menus. Os **blocos** geralmente fornecem o output de um determinado módulo ou podem ser criados para mostrar o que quer que o programador pretenda, podendo ser colocados em diversos locais do template (tema<sup>29</sup>). Podem ser configurados produzir o *output* de diversas maneiras, bem como ser mostrados apenas em determinadas páginas ou apenas para determinados utilizadores. Os **menus** são os responsáveis pela navegação no Drupal, definindo o conteúdo que será mostrado por cada *menu path* (URL relativo). Os menus são elementos nucleares do Drupal que dão acesso às páginas criadas [58].

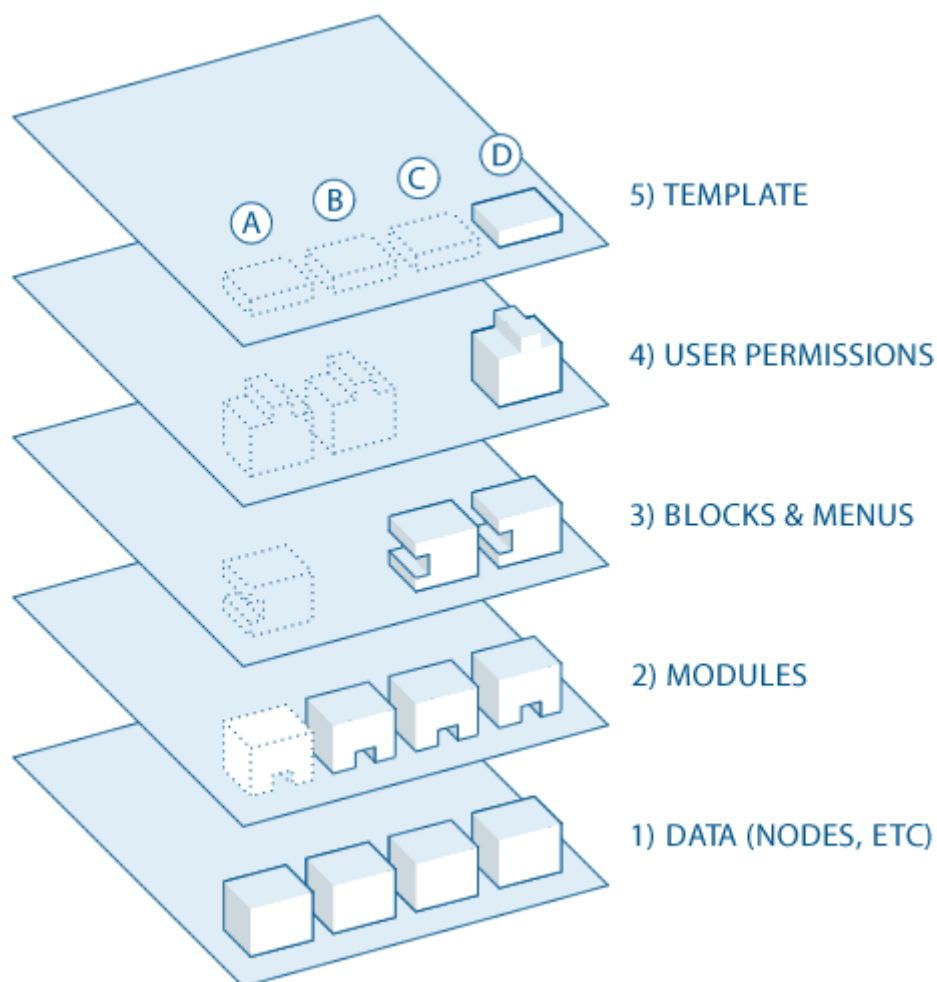


Imagem 13 (Fonte: <https://www.drupal.org/getting-started/before/overview>)

29 Do original em inglês *theme*, que representa a parte do Drupal responsável pela criação do aspeto visual do site.

Em seguida, temos as **permissões** que, através da sua configuração, permitem determinar o que diferentes tipos de utilizadores podem ou não ter acesso. Estas podem ser definidas para diversos *roles*, e por sua vez, aos utilizadores é dado um determinado *role*<sup>30</sup> de modo a definir as suas permissões [58].

No topo encontra-se o tema do site (a “pele”<sup>31</sup>). É constituído essencialmente por XHTML e CSS, contendo também algumas variáveis em PHP, para que o conteúdo gerado possa ir para os locais apropriados. Também incluído em cada tema está um conjunto de funções que podem ser usadas para se sobreporem às funções standard nos módulos de forma a fornecer controlo completo sobre o conteúdo gerado [58].

A arquitetura do Drupal pode ser algo complicada de explicar, devido à interconexão complexa de todas as peças que o compõem. É um sistema grande e modular, que pode modificar o seu comportamento dependendo do conjunto de módulos que estão instalados e em funcionamento. Existem também determinadas convenções, devido à forma como o Drupal e a sua arquitetura evoluíram, que o podem tornar um pouco “estranho” para quem está a iniciar a sua utilização [58].

O Drupal é um sistema centrado no conteúdo e no utilizador<sup>32</sup>, girando em torno do conceito da existência de um conjunto de utilizadores, com um conjunto de permissões, que criam conteúdo. Estes conceitos estão muito embutidos na sua arquitetura e nos módulos que o constituem, sendo este um fator que terá de ser levado em conta quando se escolhe esta *framework* para um determinado projeto [58].

Um outro aspeto relevante, é o de o Drupal ser extremamente modular. Isto não significa apenas que o Drupal contém um conjunto de módulos e que podem ser adicionadas características ao sistema através da adição de outros módulos, significa que, se se retirar todos os módulos que o constituem, o Drupal na realidade não faz nada, é apenas um conjunto de API's [58].

Além disso, é um sistema *event-driven* através da implementação de um sistema de *hooks*, do qual se falará mais adiante. Devido à forma como este sistema está implementado, basicamente tudo o que ocorre no Drupal são eventos a serem ativados tanto pelo seu *core*, como pelos outros módulos que são adicionados, e os restantes módulos em modo de escuta a esses eventos de modo a darem algum tipo de resposta à sua ocorrência [58].

O Drupal é também *skinnable* ou *themable*, querendo isto dizer que todo o conteúdo gerado pelo será visualizado pelo utilizador final, terá de passar por uma camada de apresentação [58].

Convém entretanto apontar um olhar mais atento, de uma perspetiva de alto-nível, aos componentes essenciais que compõem esta *framework* (Imagem 14).

---

30 Papel, função, cargo, responsabilidade.

31 *Skin*

32 Do inglês *content and user centric*.

No nível mais baixo encontra-se o conjunto de APIs do Drupal. Embora a maioria do que constitui o Drupal sejam os módulos, na sua essência encontram-se um conjunto de APIs<sup>33</sup> que permite que esse conjunto de módulos efetuem as suas funcionalidades. Representado ao lado deste conjunto de APIs encontram-se os módulos *Node* e *User*. Isto acontece porque estes são dois dos módulos mais essenciais do Drupal, pois este sistema deixa até de fazer sentido se estes módulos não se encontrarem em funcionamento, tão assente está a sua arquitetura no conteúdo e nos utilizadores, como se referiu anteriormente. Acima disto temos um conjunto de módulos que constituem o core, além de outros módulos contribuídos ou programados pelo utilizador [58].

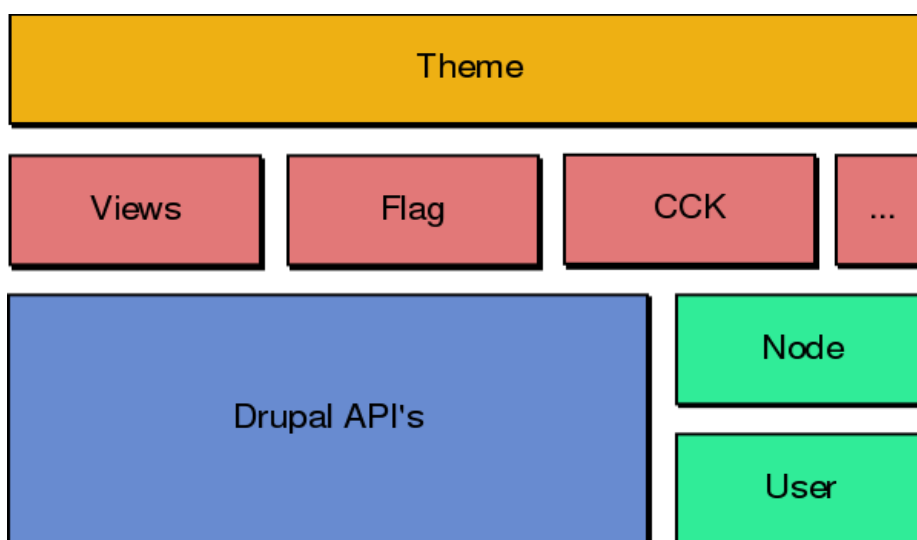


Imagem 14: Uma outra perspetiva em camadas do Drupal

Finalmente temos a camada de apresentação<sup>34</sup>, que está encarregada de dar à informação a apresentar, o aspeto e a organização pretendida [58].

Focando a atenção na camada da API, e naquilo que reside no seu interior, e que permite o funcionamento dos seus módulos. Estas são algumas das APIs presentes no Drupal (Imagem 15).

**Menu** gere diversas tarefas chave dentro do Drupal, o que algumas *frameworks* externas chamarão de *routing*. Quando uma determinada *path* é requerida por um pedido, cabe a esta API tomar a decisão de a que módulo pedir para construir esse conteúdo. Além disso, permite a construção dos menus de navegação e das *breadcrumbs* que permitem saber em que local do site um utilizador se encontra e voltar a uma localização anterior.

**Database Abstraction**, como o seu nome indica, permite fazer pesquisas em bases de dados, independentemente de qual seja o software de base de dados

<sup>33</sup> Se se for ver de uma perspetiva de arrumação da sua instalação, serão os ficheiros que se encontram na pasta *includes*.

<sup>34</sup> *Theme layer*.

com se esteja a lidar, além de criar, apagar e atualizar o conteúdo das tabelas existentes.

**Session Handling** permite guardar os dados da sessão de um utilizador ao

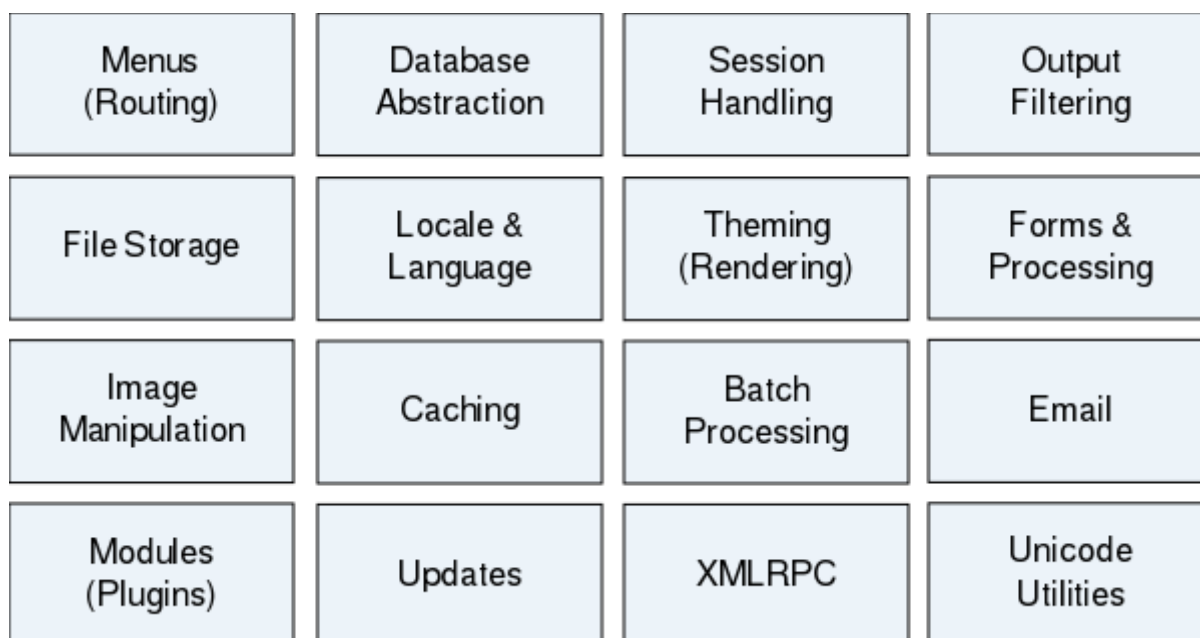


Imagem 15: Algumas dos componentes que fazem parte da API do Drupal mesmo tempo que faz o rastreamento das suas ações.

**Output Filtering** permite filtrar strings de modo a lidar com ataques tais como Cross Site Scripting<sup>35</sup> (por exemplo, injeção de Javascript), isto é, permite efetuar uma filtragem do input realizado pelo utilizador de modo que não haja consequências nefastas para o site ou para outros utilizadores.

Tarefas relativas a gestão de ficheiros em disco e base de dados estão a cargo de **File Storage**.

**Locale & Language** lida com questões de localização, fuso horário, idioma, etc.

**Theming** permite fazer passagem dos objetos que contêm a informação no Drupal para HTML ou outro formato qualquer de apresentação que se pretenda.

**Forms & Processing**, funções que controem uma representação abstracta de um form HTML e que permitem o processamento e a exibição desses forms.

**Modules**, que permite fazer a gestão dos módulos presentes no sistema e interacção com o sistema de *hooks*.

**Unicode Utilities**, que permite lidar com strings em Unicode.

Além disso fornece APIs para manipulação de imagem (**Image Manipulation**),

35 É um tipo de vulnerabilidade de segurança tipicamente encontrada em aplicações web, consistindo na injeção de *scripting* do lado do cliente em páginas web que possam ser visualizados por outros utilizadores [59].

**Caching, Batch Processing<sup>36</sup>, Email, Updates, XMLRPC**, entre outros.

Tendo em mente que o Drupal fornece um conjunto de ferramentas à partida para o funcionamento do site, convém perguntar como é que o sistema lida quando recebe um determinado pedido para a apresentação de uma página, de uma forma simplificada e de alto-nível.

Inicialmente alguém faz um pedido por uma determinada página e esse pedido é enviado para o servidor, onde se encontra o Drupal (1). Esse pedido é recebido e é tratado pelo ficheiro `index.php` (3), que se encarrega de carregar as APIs que compõem a *framework* (representado pelas setas vermelhas). Assim que este conjunto de ficheiros é carregado, o **Menu API** encarrega-se de traduzir o endereço que foi enviado e tenta descobrir a cargo de que módulo se encontra a construção daquela parte do site. Assim que existe um módulo que é encarregado desta tarefa<sup>37</sup>, é feito o pedido (5) e é efetuada a construção do conteúdo da página pedida, ao mesmo tempo que envia *requests* a outros módulos (representado pelas setas a verde), de modo a saber se existem outros módulos que pretendem efetuar alguma modificação ao conteúdo.

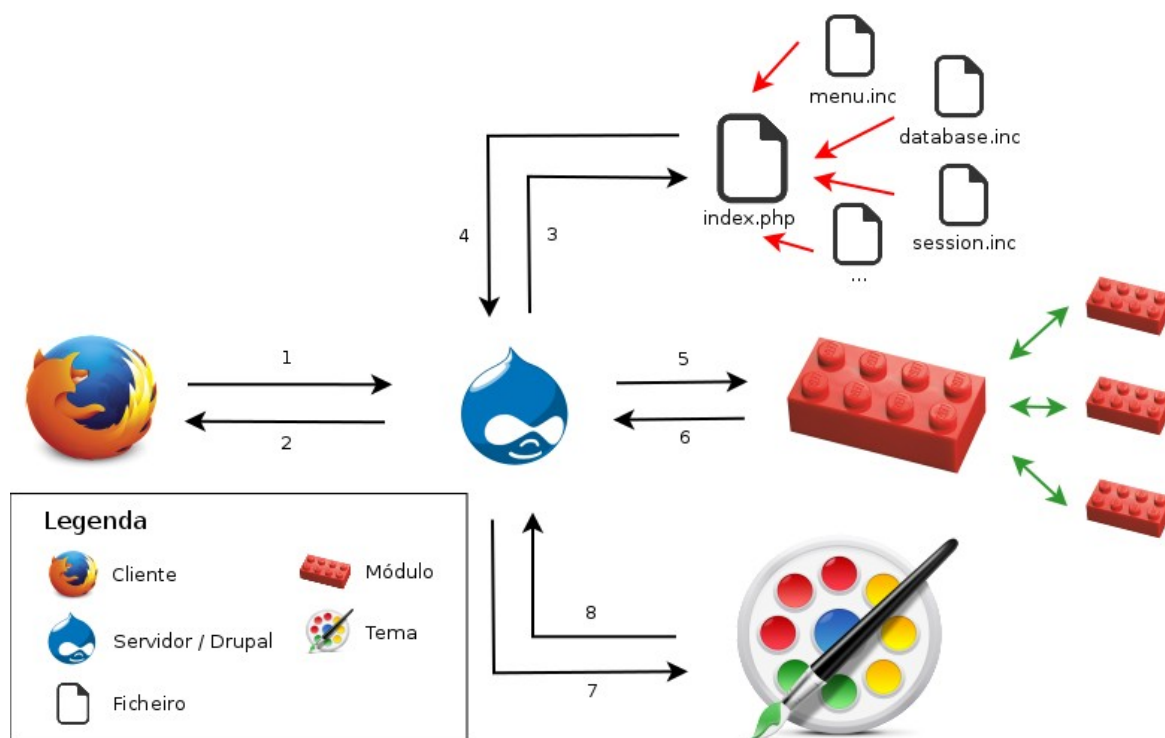


Imagem 16: Fluxo de apresentação de dados do Drupal

Após a formação do conteúdo de um determinada página, esse HTML irá ser tratado pelo Drupal de forma a que seja apresentado consoante o tema que esteja definido no sistema naquele momento (7). Sendo assim, esse conteúdo é tratado e

<sup>36</sup> Execução de um conjunto de processos, num computador, sem intervenção manual [60].

<sup>37</sup> Pode não existir nenhuma referência par ao pedido, sendo lançado um erro 404.

posteriormente reenviado para o utilizador, que recebe assim o resultado pretendido pelo seu pedido (2).

### 5.2.3 Outros aspetos relativos ao Drupal

Como se viu, o Drupal é constituído por diversos módulos, logo no seu Core, sem contar com a quantidade de módulos que fazem parte do seu universo de contribuições por parte de *developers* independentes que nada têm a ver com o planeamento e desenvolvimento do Drupal em si.

Os **hooks** são os *event listeners* e, de uma forma bastante resumida, quando é executada a visualização de uma página, o Drupal procura, nos módulos que estejam ativados, “contribuições” para os dados presentes na página. Isto torna possível para um módulo definir novos urls e páginas do site (*hook\_menu*), adicionar conteúdo às páginas (*hook\_block*, *hook\_footer*, etc), adicionar tabelas adicionais à base de dados (*hook\_schema*), e muito mais.

Esta funcionalidade tem como base a pesquisa de funções com nomes específicos na listagem de funções de um módulo e aproveita existência da função *function\_exists* do PHP para a sua implementação.

No Drupal, a informação é transportada através do uso de arrays encadeados<sup>38</sup>, chamados de **InfoArrays**<sup>39</sup>, que, dependendo do seu conteúdo, pode ser interpretado de diversas maneiras pelas APIs.

Também utilizando arrays, os **Renderables** são utilizados para conter elementos que mais tarde serão renderizados para HTML pelo Drupal.

O Drupal também implementa um sistema de **Callbacks**, utilizando estes arrays de informação e inserindo o nome de funções no seu interior de modo a que estas sejam chamadas no momento oportuno. Um exemplo da utilização deste sistema é através de um *Renderable* que, para a validação de um form que ele representa, é adicionada uma outra função além das já utilizadas.

Aproveitando também a forma como se encontram implementados estes sistemas de arrays, os **Alter Hooks** permitem a modificação do conteúdo já formado por um determinado módulo inquirindo outros módulos se pretendem fazer alguma modificação à informação que se encontra ali presente.

### 5.2.4 Acerca dos módulos da aplicação FactSegur

Pelo que vimos na secção anterior, o Drupal subdivide-se em vários módulos, que afetam o sistema, aumentam-no e comunicam entre si de modo a gerar novos conteúdos facilmente.

---

<sup>38</sup> Possível tradução encontrada pelo estagiário para *nested arrays*.

<sup>39</sup> Na realidade, muitos dos dados e outros aspetos do Drupal são tratados através de arrays: o sistema de menus, os *schemas* gerados para a base de dados, a definição de todos os passos pretendidos para a *Batch Api*, são apenas alguns exemplos dos casos onde *arrays* são utilizados.

Existirá um conjunto de módulos que irá modificar o funcionamento de componentes já existentes do Drupal que são apresentados na imagem 17 dentro do Sistema Geral.

Em seguida temos um conjunto de módulos que terão de ser construídos de raiz, dado que dentro do sistema não existe nada implementado que contenha o tipo de comportamentos esperados deles.

Também referido na imagem 17 encontra-se um módulo que permite a comunicação entre a aplicação móvel e o sistema. Todos estes aspetos serão discutidos nas secções seguintes.

Finalmente, dado que não é feita qualquer referência a aspetos relativos a aspetos de sessão<sup>40</sup>, é de notar que o Drupal implementa de raiz e gere todos estes aspetos. Os ecrans, tal como foram apresentados nos documentos anexos de casos de uso, serão construídos através apenas da modificação do tema do Drupal.

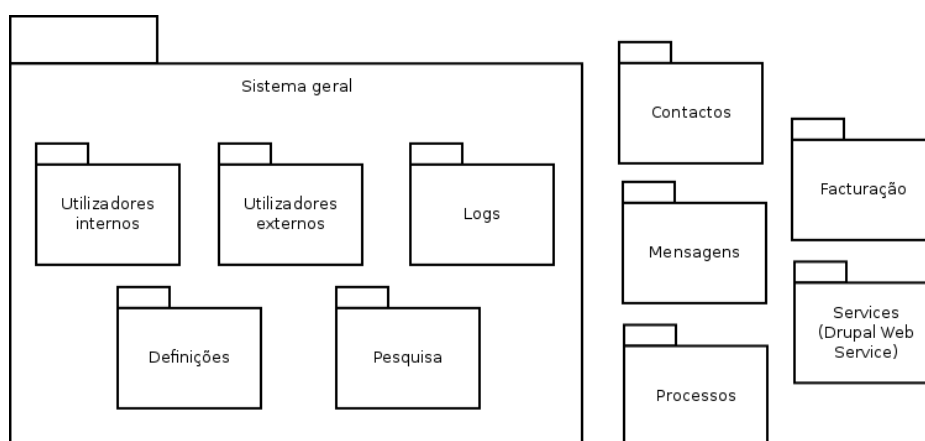


Imagem 17: Vista de módulos da aplicação

### 5.2.4.1 Utilizadores internos

<b>Módulo de utilizadores internos</b>	
Este módulo estará encarregado do conjunto de ações necessárias sobre os utilizadores do sistema. Como o Drupal já tem em si implementado um sistema de gestão de utilizadores, serão utilizados os seus mecanismos para obter os <i>forms</i> e dados dos utilizadores do sistema.	
<b>Listar utilizadores</b>	Serão utilizadas <i>Views</i> para construir a listagem de utilizadores e os respetivos filtros.
<b>Editar utilizador</b>	Os dados e o <i>form</i> respetivo ao utilizador serão obtidos através de dados já contidos na base de dados, utilizando os mecanismos do Drupal.
<b>Adicionar utilizador</b>	O <i>form</i> a ser usado será obtido a partir do que já se encontra construído de raiz no Drupal.

40 Login, logout, manutenção de dados relativos à sessão do utilizador.

<b>Apagar utilizador</b>	Será implementada uma função para este propósito, utilizando os mecanismos para remoção de utilizadores do Drupal.
<b>Listar nível de acesso</b>	Será usada uma <i>View</i> com a listagem de <i>roles</i> existentes no sistema. De notar que o Drupal implementa de raiz o sistema de permissões.
<b>Editar nível de acesso</b>	Será obtida a lista de permissões para um determinado nível de acesso do sistema, através de mecanismos do Drupal.
<b>Adicionar nível de acesso</b>	O <i>form</i> a ser usado será obtido a partir do que já se encontra construído de raiz no Drupal.
<b>Apagar nível de acesso</b>	Será implementada uma função para este propósito, utilizando os mecanismos para remoção de níveis de acesso do Drupal.

### 5.2.4.2 Utilizadores externos

<b>Módulo de utilizadores externos</b>	
Este módulo estará encarregado do conjunto de ações necessárias sobre os utilizadores externos do sistema, ou seja, clientes interessados em ter acesso a dados relativos aos seus processos. Como o Drupal já tem em si implementado um sistema de gestão de utilizadores, serão utilizados mecanismos do Drupal para obter os forms e dados dos utilizadores do sistema. Este módulo é, na sua essência, idêntico ao módulo de utilizadores internos.	
<b>Listar utilizadores</b>	Serão utilizadas <i>Views</i> para construir a listagem de utilizadores e os respetivos filtros.
<b>Editar utilizador</b>	Os dados e o form respetivo ao utilizador serão obtidos através de dados já contidos na base de dados, utilizando os mecanismos do Drupal.
<b>Adicionar utilizador</b>	O form a ser usado será obtido a partir do que já se encontra construído de raiz no Drupal.
<b>Apagar utilizador</b>	Será implementada uma função para este propósito, utilizando os mecanismos para remoção de utilizadores do Drupal.
<b>Listar nível de acesso</b>	Será usada uma <i>View</i> com a listagem de <i>roles</i> existentes no sistema. De notar que o Drupal implementa de raiz o sistema de permissões.
<b>Editar nível de acesso</b>	Será obtida a lista de permissões para um determinado nível de acesso do sistema, através de mecanismos do Drupal.
<b>Adicionar nível de acesso</b>	O form a ser usado será obtido a partir do que já se encontra construído de raiz no Drupal.
<b>Apagar nível de acesso</b>	Será implementada uma função para este propósito, utilizando os mecanismos para remoção de níveis de acesso do Drupal.



### 5.2.4.3 Contactos

<b>Módulo de contactos</b>	
Este módulo estará encarregado do conjunto de ações necessárias para gerir os contactos associados ao sistema.	
<b>Listar contactos individuais</b>	Serão utilizadas <i>Views</i> para construir a listagem de contactos e os respetivos filtros.
<b>Ver contacto individual</b>	Será construída uma função que permita a obtenção e mostragem dos dados, utilizando mecanismos do Drupal.
<b>Adicionar contacto individual</b>	Será construída uma função que permita a adição de um novo contacto, utilizando mecanismos do Drupal.
<b>Editar contacto individual</b>	Será construída uma função que permita a edição de um contacto, utilizando mecanismos do Drupal.
<b>Apagar contacto individual</b>	Será construída uma função que permita a remoção de um contacto, utilizando mecanismos do Drupal.
<b>Listar contactos de empresa</b>	Serão utilizadas <i>Views</i> para construir a listagem de contactos e os respetivos filtros.
<b>Ver contacto de empresa</b>	Será construída uma função que permita a obtenção e mostragem dos dados, utilizando mecanismos do Drupal.
<b>Adicionar contacto de empresa</b>	Será construída uma função que permita a adição de um novo contacto, utilizando mecanismos do Drupal.
<b>Editar contacto de empresa</b>	Será construída uma função que permita a edição de um contacto, utilizando mecanismos do Drupal.
<b>Apagar contacto de empresa</b>	Será construída uma função que permita a remoção de um contacto, utilizando mecanismos do Drupal.

### 5.2.4.4 Mensagens

<b>Módulo de mensagens</b>	
Este módulo estará encarregado de mostrar as mensagens de alerta de fim de prazo de processo, mini-lista de emails do utilizador que se encontra ligado, listagem de tarefas e listagem de processos atribuídos	
<b>Listar alertas</b>	Serão utilizadas <i>Views</i> para construir a listagem de alertas, mostrado os processos cuja data de conclusão se encontra próxima.
<b>Listar mensagens</b>	Serão utilizadas <i>Views</i> para fazer a listagem de mensagens (emails) que um determinado utilizador tem na sua Inbox.
<b>Listar tarefas</b>	Serão utilizadas <i>Views</i> para fazer a listagem das tarefas que foram atribuídas ao utilizador que se encontra ligado.

<b>Listar processos</b>	Serão utilizadas <i>Views</i> para fazer a listagem de processos que estão associados ao utilizador que se encontra ligado.
-------------------------	---

### 5.2.4.5 Tarefas

<b>Módulo de tarefas</b>	
Este módulo estará encarregado de mostrar e permitir a criação, edição e remoção de tarefas. É um sub-módulo do componente Mensagens.	
<b>Listar minhas tarefas</b>	Serão utilizadas <i>Views</i> para construir a listagem de tarefas atribuídas ao utilizador que se encontra ligado.
<b>Ver minha tarefa</b>	Será construída uma função que obtém os dados da base de dados e mostra-os ao utilizador.
<b>Listar tarefas atribuídas</b>	Serão utilizadas <i>Views</i> para fazer a listagem das tarefas que foram atribuídas ao utilizador que se encontra ligado.
<b>Editar tarefa atribuída</b>	Serão utilizadas <i>Views</i> para construir a listagem de tarefas que foram atribuídas pelo utilizador que se encontra ligado.
<b>Criar nova tarefa atribuída</b>	Será construídas funções que mostra o <i>form</i> para a criação de novas tarefas e guardam na base de dados novas tarefas.
<b>Apagar tarefa atribuída</b>	Será construída uma função que permite a remoção de uma determinada tarefa.

### 5.2.4.6 Basic Webmail

O Basic Webmail é um módulo contribuído pela comunidade do Drupal<sup>41</sup> e que permite o acesso ao email de um determinado utilizador.

Todas as funções necessárias para a utilização de um sistema de email estão asseguradas por este módulo. No entanto será necessária a adaptação deste módulo ao tema idealizado para o projeto, sendo que este aspeto está contemplado na calendarização.

### 5.2.4.7 Processos

<b>Módulo de processos</b>	
Este módulo permitirá a gestão de processos do sistema, bem como a sua criação, remoção e edição. Será um dos módulos fundamentais a ser desenvolvidos, sendo que este	

41 [https://www.drupal.org/project/basic\\_webmail](https://www.drupal.org/project/basic_webmail)

estágio não fará sentido sem a sua realização por completo.	
<b>Listar processo</b>	Serão utilizadas <i>Views</i> para construir a listagem de processos atribuídas ao utilizador que se encontra ligado.
<b>Adicionar processo</b>	Será usado o sistema de forms do Drupal para a construção dos campos necessários para a inserção de um novo projeto no Drupal, que terá de conter todos os campos necessários tal como definido no documento de casos de uso relativo a este módulo. Além disso será usada a API de base de dados do Drupal para a inserção de novos processos.
<b>Editar processo</b>	Será usado o sistema de forms do Drupal para a construção dos campos necessários para a obtenção e edição de um novo projeto no Drupal, que terá de conter todos os campos necessários tal como definido no documento de casos de uso relativo a este módulo. Além disso será usada a API de base de dados do Drupal para a obtenção e edição de novos processos.
<b>Apagar processo</b>	Será construída uma função que lidará com este aspeto, utilizando a API de base de dados do Drupal.

#### 5.2.4.8 Definições

Esta parte consistirá apenas na obtenção, dentro dos restantes módulos, daqueles que pretendem apresentar dados nesta secção através de *Hooks*. Para que a parte de definições se encontre o mais independente possível dos módulos que se encontram presentes, torna-se necessária a utilização deste sistema, pelo que este módulo não será mais do que um interface de visualização das configurações permitidas por cada um dos módulos.

#### 5.2.4.9 Pesquisa

Este sistema será construído através da utilização de *Views* e do sistema de *Hooks* do Drupal. Os módulos que pretendam apresentar um interface de pesquisa terão de notificar este módulo e, através do sistema de *Views*, os dados pretendidos poderão ser pesquisados e acedidos.

#### 5.2.4.10 Logs

Cada um dos módulos do Drupal estará encarregado de gerar a informação necessária para que depois o módulo de Logs a possa apresentar. Este aspeto já se encontra contemplado e implementado, sendo que apenas terá de ser necessário programar a obtenção desses dados e construir o tema que dará o aspeto desejado a este módulo, conforme o que foi delineado na fase de especificação de requisitos.

#### 5.2.4.11 Drupal Web Service

Para se poder obter os dados, que se encontram guardados no sistema, através

uma aplicação externa<sup>42</sup>, será necessária a construção de uma API que permita a obtenção dos mesmos.

Após efectuar um conjunto de pesquisas na Internet em busca de módulos já construídos pela comunidade que permitam este tipo de comunicação, foi encontrada a informação que existem 3 APIs REST que são mais utilizadas [61], a **Services**<sup>43</sup>, **RestWS**<sup>44</sup> e **RestFUL**<sup>45</sup>. Além disso, foi encontrada uma quarta, a **REST/JSON**<sup>46</sup>, que não apresenta muitos aspetos implementados e a sua taxa de utilização é muito reduzida, pelo que a possibilidade de integração neste projeto foi descartada à partida.

Embora as restantes apresentem todos os aspetos necessários para a consideração do seu uso neste software, a que mais confiança gerou foi a *Services*, que apresenta uma quantidade de sites que a utilizam mais elevada, e a quantidade de *commits* é também mais frequente. Além disso, um outro aspeto que leva a suportar ainda mais a tese de que esta será a escolha acertada, é que ela é desenvolvida pela empresa do autor do Drupal, a Acquia, que é a principal fornecedora de suporte à *framework* em questão.

Sendo assim, este será o módulo que permitirá a comunicação entre a aplicação móvel e o sistema e serão utilizados os seus mecanismos já implementados para efetuar este processo.

## 5.2.5 Modificações ao tema do Drupal

Conforme já foi visto anteriormente, o Drupal possui mecanismos para que o aspeto do site possa ser modificado, tendo várias skins construídas pela própria comunidade e permitindo, através desta API, a construção de novos temas.

Sendo assim para que o site apresente o aspeto planeado, em vez de serem os módulos a gerir a questão da forma como serão apresentados visualmente ao utilizador final, será utilizado o sistema de temas integrados.

Os módulos irão passar para esta camada de apresentação os dados que querem apresentar. Em seguida, os aspetos gerais de apresentação, tais como os *textfields*, *quadros*, etc, serão delineados por este componente e, exceto em casos específicos, serão gerais para todos os módulos, permitindo assim que o *look-and-feel* do software na sua generalidade seja consistente.

## 5.3 Modelo de dados

O Drupal necessita de uma base de dados para o seu funcionamento, tendo para

42 No caso deste projeto será necessário para efetuar a comunicação entre a aplicação móvel e o Drupal.

43 <https://www.drupal.org/project/services>

44 <https://www.drupal.org/project/restws>

45 <https://www.drupal.org/project/restful>

46 [https://www.drupal.org/project/rest\\_json](https://www.drupal.org/project/rest_json)

isso na sua base de dados um conjunto de tabelas. No entanto, dado que se pretende ampliar o conjunto de funcionalidades do Drupal com um conjunto de novos módulos, é necessário na fase de arquitetura estabelecer um modelo de dados que permita, aquando da implementação, saber de forma inequívoca como é que esses dados serão guardados ou obtidos.

Em primeiro lugar, dado que o módulo de utilizadores internos amplia as funcionalidades dos utilizadores gerais do Drupal, é necessário adicionar um conjunto de campos que permitam este objetivo.

Na imagem 18 é apresentado um conjunto de tabelas, sendo que a **users** é uma tabela já existente previamente e para permitir o armazenamento dos novos dados foi criada a tabela **users\_internal**. Esta tabela irá conter os dados que foram definidos no anexo 1, no caso de uso UC007-1.

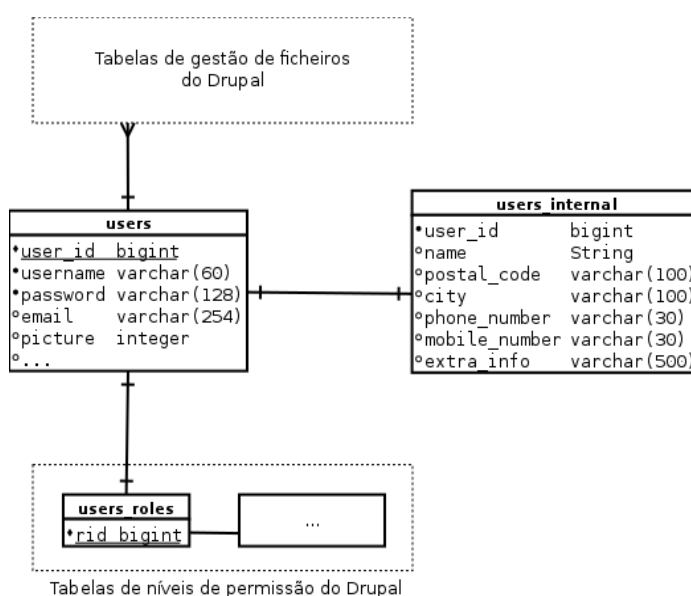


Imagem 18: Tabelas para o módulo de utilizadores internos

Relativamente ao módulo de utilizadores externos, também será necessário acrescentar uma tabela auxiliar que contenha os dados da empresa ao qual esse utilizador está associado. Como se pode ver na imagem 19 para guardar os dados de uma empresa é usada a tabela **contacts\_enterprise**<sup>47</sup>, e a relação entre ela e os utilizadores externos será efetuada através da tabela **users\_external**.

47 Embora esta tabela esteja mais directamente relacionada com o módulo de contactos, como se verá mais à frente, é mostrada nesta imagem de forma a melhor se compreender a relação entre os diversos componentes.

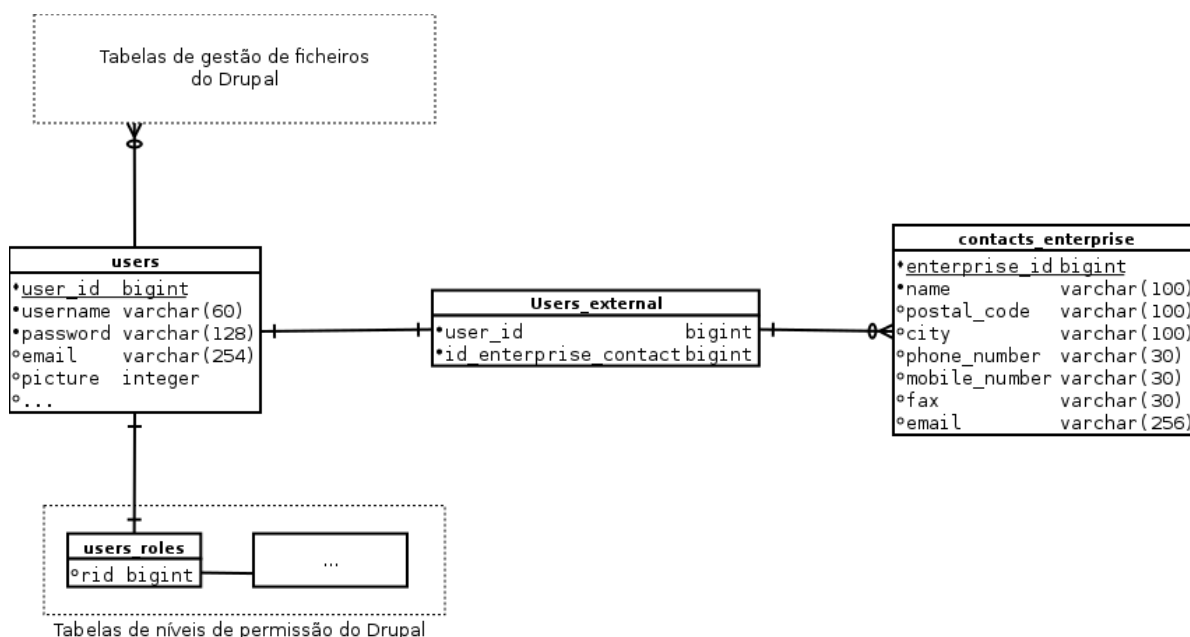


Imagem 19: Tabelas relativas ao módulo de utilizadores externos

Nos dois casos, para a gestão das permissões dos utilizadores, é utilizado um conjunto de tabelas que lidam com esses aspetos, tal como se encontra apresentado nas imagens, através das relações entre a tabela **users** e as tabelas de gestão de ficheiros e das tabelas de níveis de permissão do Drupal.

No módulo de mensagens, dado que a maioria dos aspetos que o contemplam são pesquisas noutras áreas do sistema através do sistema de Views, apenas a componente de tarefas irá necessitar de armazenar os seus dados na base de dados. Quando é criada uma nova tarefa é necessário identificar o criador (através da *foreign-key* `id_user_author`) e o utilizador a quem a tarefa foi atribuída (`id_user_designated`), bem como dar um título, um conteúdo e uma data limite à tarefa (imagem 20).

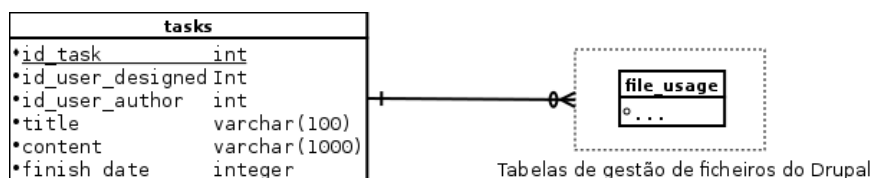


Imagem 20: Tabelas relativas ao sub-módulo de tarefas

Tal como foi especificado na fase de requisitos, existem duas modalidades de contactos, os contactos individuais e os coletivos (ou empresariais). Dado que podem existir contactos individuais associados aos contactos coletivos, esse aspeto é representado na relação entre as duas tabelas através da *foreign-key* *enterprise\_id* na primeira tabela da imagem 21. Os restantes dados necessários, conforme tinha sido documentado anteriormente, também se encontram representados.

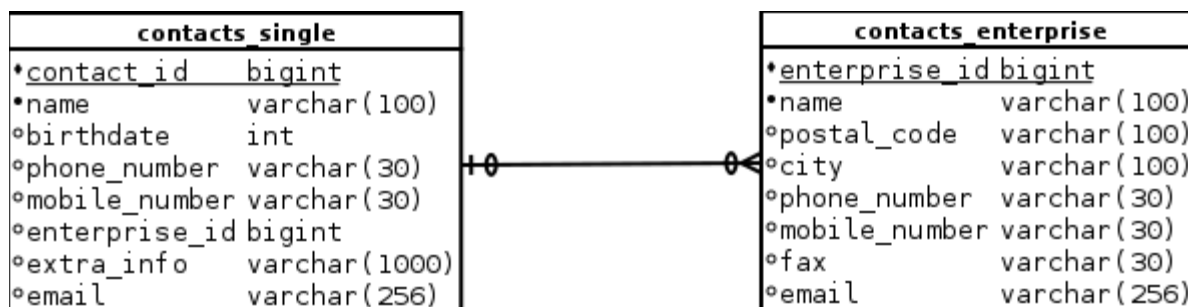


Imagem 21: Tabelas relativas ao módulo de contactos

Dado a complexidade do módulo de processos, é o que apresenta a maior quantidade de tabelas (imagem 22). Tal como foi especificado na fase de requisitos, na tab Dados Gerais, os dados do quadro Dados Internos é representado pelos quatro primeiros elementos da tabela, através de *foreign-keys*, estabelecendo assim a relação com as tabelas **users** e **states**. No quadro Dados Seguradora, os campos são representados pelo *id\_enterprise*<sup>48</sup>, *enterprise\_process\_number* e pela tabela **policies**, dado que poderão existir uma ou mais apólices com um ramo associado. No quadro Dados Sinistro os seus campos estarão representados pelos itens *casualty\_address*, *casualty\_postal\_code*, *casualty\_city*, *casualty\_date* e pela tabela *casualty\_type*.

48 O requerente terá de ser um dos contactos colectivos.

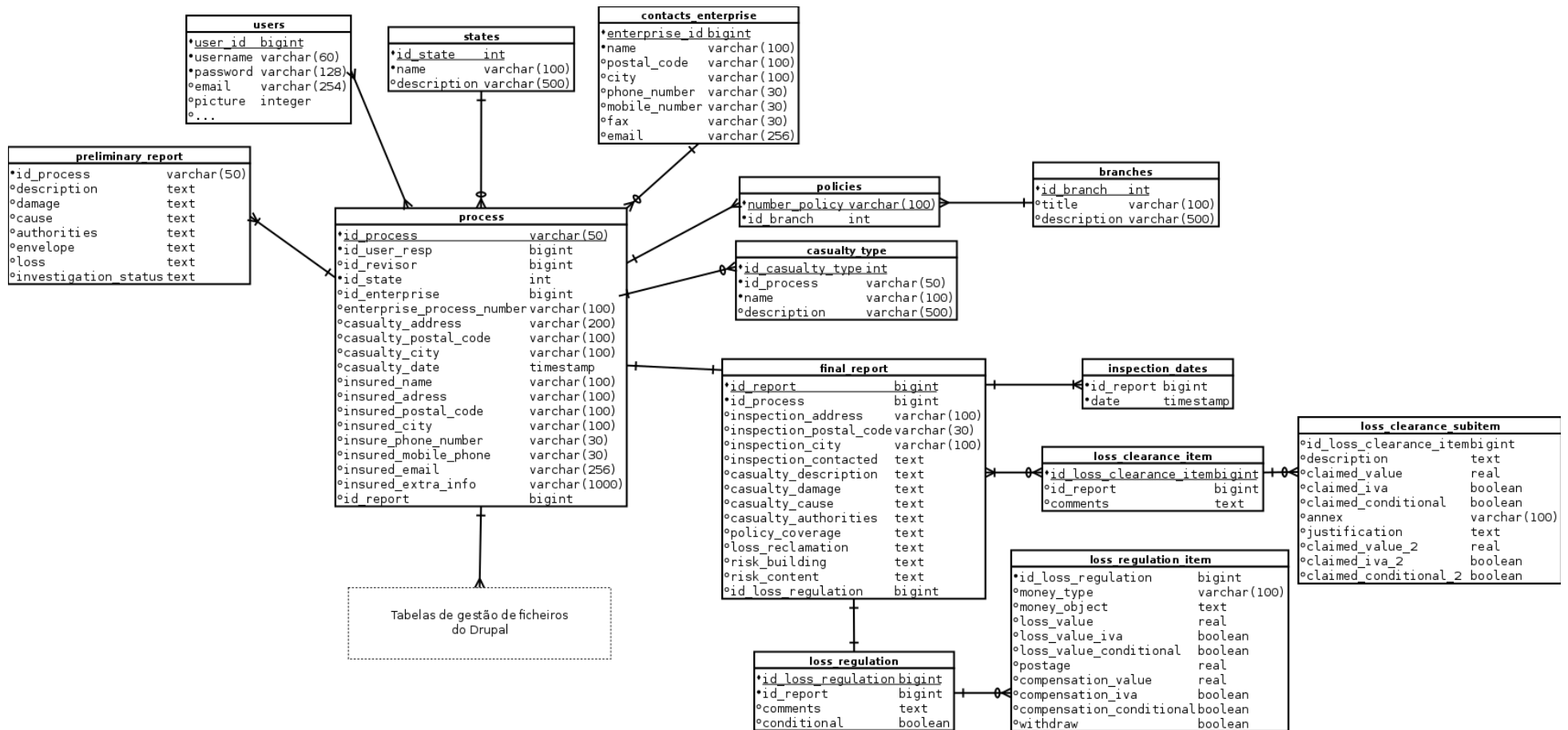


Imagem 22: Tabelas relativas ao módulo de processos



Passando para a tab Segurado, os dados estão representados na tabela **process** desde o campo *insured\_name* até ao campo *insured\_extra\_info*.

Na tab seguinte, dado que podem existir vários relatório preliminares associados ao mesmo processo, existe uma tabela complementar onde os seus dados podem ser guardados, a **preliminary\_report**.

Os dados da tab Relatório Final são guardados na tabela **final\_report** e nas respectivas tabelas auxiliares **inspection\_dates**, **loss\_clearance\_item**, **loss\_clearance\_subitem**, **loss\_regulation** e **loss\_regulation\_subitem**.

As informações relativas aos ficheiros associados aos relatórios serão guardados nas tabelas já existentes no Drupal para esse efeito.

## 5.4 Conclusão

Nesta secção foram definidos os aspetos essenciais relativos à arquitetura da aplicação. Estando concluída esta fase procede-se à implementação dos aspetos referidos nos requisitos, respeitando a arquitetura delineada e as tecnologias escolhidas para o seu desenvolvimento.

# Capítulo 6

## Implementação

Neste capítulo pretende-se fazer a descrição do processo de implementação deste software, de acordo com o conjunto de requisitos reunidos no capítulo 4 e a arquitetura definida no capítulo 5.

Inicialmente será discutido o ambiente de desenvolvimento - conjunto de ferramentas e processos pelos quais se optou para a realização deste software.

Em seguida serão discutidos os aspetos de implementação relativamente aos módulos produzidos, sendo esta discussão iniciada por uma breve exposição de conceitos necessários para a sua compreensão.

Finalmente serão explanados os aspetos relativos à construção do tema, sendo antes referido um conjunto de tópicos introdutórios.

### 6.1 Ambiente de desenvolvimento

Aquando da realização de um projeto de software, é necessário um ambiente desenvolvimento onde o programador (ou conjunto de programadores) o possa efetuar. Neste ambiente terão de estar contempladas todas as ferramentas necessárias para a sua execução, bem como um ambiente de teste que simulará o real<sup>49</sup> onde ocorrerá a utilização por parte dos utilizadores finais [62][63].

A escolha apropriada destas ferramentas é fundamental, pois um conjunto adequado poderá facilitar a vida a quem o desenvolve.

Na imagem 23 é apresentada uma representação esquemática desse ambiente de desenvolvimento.

---

49 Ambiente de produção.

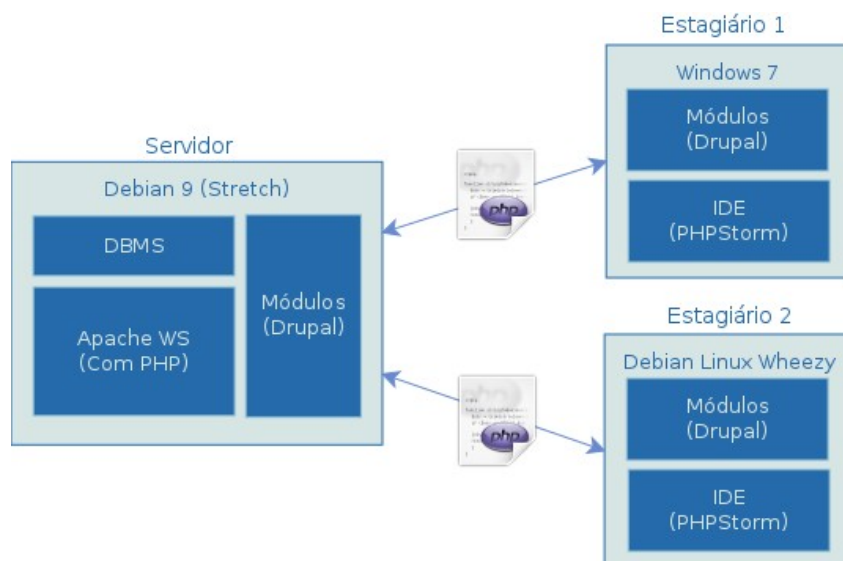


Imagem 23: Ambiente de desenvolvimento

Estando este projeto a ser realizado por duas pessoas<sup>50</sup>, foram estudadas opções que permitissem o seu desenvolvimento conjunto, num ambiente partilhado. Para tal, utilizou-se um servidor<sup>51</sup> onde os dois colaboradores poderiam partilhar a mesma versão do mesmo *web server*, *dbms* e sistema operativo, condição necessária para se afirmar a existência de compatibilidade entre o código produzido por ambos. Além disso, quaisquer operações de instalação, remoção ou atualização de software manifestar-se-iam no ambiente de qualquer um dos programadores.

A escolha do IDE<sup>52</sup> prendia-se com 3 fatores fundamentais: existência de uma versão para Windows 7 e para Debian Linux<sup>53</sup>, ser gratuito, orientado para programação em PHP<sup>54</sup> e com características que facilitassem a utilização da API do *Drupal*<sup>55</sup>.

Efetuada a pesquisa, optou-se pela utilização do *PHPStorm* 9.0.2<sup>56</sup>, que apresentava todas as características pretendidas<sup>57</sup>, contando ainda com excelentes referências por parte dos seus utilizadores [64][65][66]. Os ficheiros dos módulos são produzidos localmente e, através de mecanismos do *PHPStorm*, transferidos para o servidor.

De referir ainda que após o final da implementação de cada módulo, era feita a sua integração no ambiente de desenvolvimento do outro estagiário, de modo a garantir

50 Embora este relatório diga respeito a um estágio individual, o projeto como um todo está a ser desenvolvido por dois estagiários, em dois estágios distintos.

51 Na realidade utilizou-se uma máquina virtual instalada num servidor, mas este aspeto não fará diferença.

52 Embora não fosse fundamental a utilização do mesmo IDE, chegou-se à conclusão que seria interessante a utilização do mesmo, permitindo a partilha de conhecimento e experiência.

53 Sistemas operativos utilizados nos computadores pessoais de cada um dos estagiários.

54 Ou, pelo menos, que tivesse algum plugin que permitisse o code highlighting e oferecesse outras características para *web development*.

55 *Code-completion*, *code-formatting*, etc.

56 <https://www.jetbrains.com/phpstorm/specials/phpstorm/phpstorm.html>

57 Embora este IDE seja pago, apresenta uma solução gratuita para estudantes (<https://www.jetbrains.com/student/>).

que a sua coexistência não gerava qualquer conflito e, nos módulos que usavam as funcionalidades uns dos outros, que estas se encontravam corretamente implementadas.

## 6.2 Instalação do Drupal

A instalação do Drupal é um processo automatizado, sendo pedido ao utilizador uase de dadom conjunto de informações acerca do site que se está a construir<sup>58</sup> e da sua bs. No final deste processo é criado um utilizador com privilégios de administrador, que poderá aceder a todos os conteúdos criados dentro do site Drupal.

Ainda assim, inicialmente é necessária a criação de uma pasta à qual o *web server* tenha acesso, para conter os ficheiros do *Drupal*. Adicionalmente, é necessário criar a base de dados, guardando as informações de acesso para, em seguida, as fornecer à instalação do *Drupal*.


A título de exemplo, na imagem 24 [67] encontra-se um exemplo de um ecrã de instalação do Drupal.

Configure site | Drupal - Mozilla Firefox

Configure site | Drupal

192.168.56.2/drupal/install.php?profile=standard&locale=en

Configure site



- ✓ Choose profile
- ✓ Choose language
- ✓ Verify requirements
- ✓ Set up database
- ✓ Install profile
- Configure site
  - Finished

**SITE INFORMATION**

**Site name \***  
192.168.56.2

**Site e-mail address \***  
admin@localhost  
Automated e-mails, such as registration information, will be sent from this address. Use an address ending in your site's domain to help prevent these e-mails from being flagged as spam.

**SITE MAINTENANCE ACCOUNT**

**Username \***  
admin  
Spaces are allowed; punctuation is not allowed except for periods, hyphens, and underscores.

**E-mail address \***  
admin@localhost

**Password \***  
Password strength: **Fair**

**Confirm password \***  
Passwords match: yes

To make your password stronger:

- Add uppercase letters
- Add numbers
- Add punctuation

Imagem 24: Exemplo de ecrã de instalação do Drupal

58 Poderá haver necessidade de modificar a permissão a um conjunto de ficheiros, mas o processo de instalação lança avisos sempre que necessário.

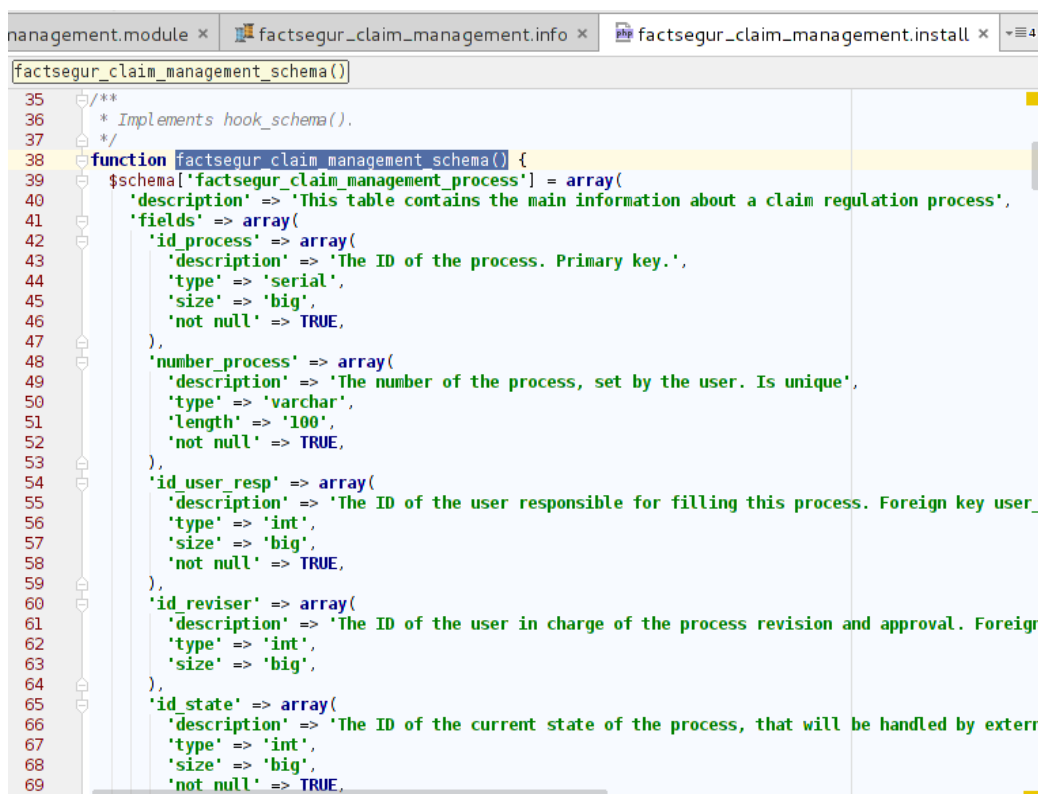
Neste caso, são pedidos dados relativos ao site<sup>59</sup> e ao seu administrador<sup>60</sup>.

## 6.3 Base de dados

Para que a aplicação esteja de acordo com os requisitos definidos, serão efetuadas modificações à base de dados de modo a que novas informações relativas a cada um dos módulos possam ser guardadas e obtidas. As funções responsáveis por estas modificações estão definidas dentro de cada um dos módulos efetuados, num ficheiro “.install”<sup>61</sup>.

De modo a não haver conflitos entre as tabelas do Drupal e as de outros módulos que porventura possam vir ser geradas, foi necessário definir uma regra de nomenclatura para as tabelas utilizadas pela nossa aplicação. A regra seguida foi a adição do sufixo <nome do módulo> ao nome da tabela que se pretende criar. Dado que o nome do módulo é único dentro do “ecossistema” do *Drupal*, a probabilidade de conflito é mínima.

Na imagem 25 pode-se ver o conteúdo de um ficheiro .install.



```

35  /**
36   * Implements hook_schema().
37   */
38  function factsegur_claim_management_schema() {
39    $schema['factsegur_claim_management_process'] = array(
40      'description' => 'This table contains the main information about a claim regulation process',
41      'fields' => array(
42        'id_process' => array(
43          'description' => 'The ID of the process. Primary key.',
44          'type' => 'serial',
45          'size' => 'big',
46          'not null' => TRUE,
47        ),
48        'number_process' => array(
49          'description' => 'The number of the process, set by the user. Is unique',
50          'type' => 'varchar',
51          'length' => '100',
52          'not null' => TRUE,
53        ),
54        'id_user_resp' => array(
55          'description' => 'The ID of the user responsible for filling this process. Foreign key user_',
56          'type' => 'int',
57          'size' => 'big',
58          'not null' => TRUE,
59        ),
60        'id_reviser' => array(
61          'description' => 'The ID of the user in charge of the process revision and approval. Foreign',
62          'type' => 'int',
63          'size' => 'big',
64        ),
65        'id_state' => array(
66          'description' => 'The ID of the current state of the process, that will be handled by extern',
67          'type' => 'int',
68          'size' => 'big',
69          'not null' => TRUE,

```

Imagem 25: Exemplo do conteúdo de um ficheiro <module\_name>.install

59 IP e endereço de email do site.

60 Conta correspondente ao *superuser* do site, terá acesso a todos e conteúdos e todos os privilégios.

61 Função <nome\_do\_módulo>\_schema [68].

Neste caso é apresentada uma parte da função `factsegur_claim_management_schema()`, do módulo de gestão de processos, onde estão a ser definidos um conjunto de campos da tabela `factsegur_claim_management_process`.

## 6.4 Módulos

Os módulos são um aspeto essencial, pois a implementação deste projeto resume-se fundamentalmente a um conjunto de módulos.

Nesta secção serão discutidas as suas implementações e, existindo um conjunto de conceitos transversais a todos eles, estes serão discutidos previamente, de modo a evitar repetições.

### 6.4.1 Conceitos base

#### 6.4.1.1 Criação e instalação de um módulo

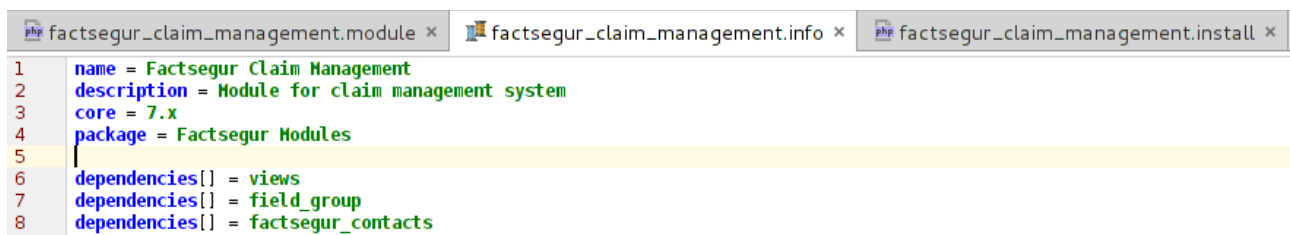
O *Drupal* dispõe de um sistema de reconhecimento e instalação de módulos. Para tal, um determinado módulo tem de obedecer a um conjunto de critérios para poder ser utilizado.

Um módulo<sup>62</sup> terá de estar contido dentro de pastas específicas do site *Drupal*<sup>63</sup> de modo a poder ser reconhecido pelo sistema; será constituído por uma pasta e, no seu interior, irá conter um ficheiro “.info” com a informação apresentada na tabela 4<sup>64</sup> [69]:

<b>name</b>	Define o nome do módulo. Este irá ser o nome que será mostrado na página de módulos do <i>Drupal</i> .
<b>description<sup>65</sup></b>	Uma pequena descrição que dirá ao administrador do site o que é que este módulo faz.
<b>core</b>	A versão do <i>Drupal</i> para a qual este módulo foi produzido.

Tabela 4: Informação necessária num ficheiro .info

Na imagem 26 pode-se ver o exemplo do conteúdo de um ficheiro .info.



```

1 name = Factsegur Claim Management
2 description = Module for claim management system
3 core = 7.x
4 package = Factsegur Modules
5
6 dependencies[] = views
7 dependencies[] = field_group
8 dependencies[] = factsegur_contacts

```

Imagem 26: Exemplo de um ficheiro .info

Além dos campos indicados na tabela 4, também se pode ver a declaração de outros módulos dos quais este depende.

Quando o módulo está ativo, é o ficheiro `.module` que o Drupal irá localizar em primeiro lugar. Este contém os métodos essenciais para o seu funcionamento e/ou referências para outros ficheiros relevantes, sendo mostrado o conteúdo de um destes ficheiros, na imagem 27, a título de exemplo.

```

factsegur_claim_management_menu():array
238 +function factsegur_claim_management_form_factsegur_permissions_myaddRole_form_alter(&$form, &$form_state, $form_id) {...}
243
244 +function factsegur_claim_management_get_module_name() {...}
247
248 function factsegur_claim_management_form_alter($form, &$form_state){
249 }
250
251 /** Implements hook_form_FORM_ID_alter(). ...*/
254 function factsegur_claim_management_form_factsegur_claim_management_claim_form_alter(&$form, &$form_state, $form_id) {
255 }
256
257 function factsegur_claim_management_field_widget_form_alter(&$element, &$form_state, $context) {
258 }
259
260 +function branch_load($branch_id) {...}
263
264 +function factsegur_claim_management_branch_load($branch_id) {...}
278
279 +function claim_type_load($claim_type_id) {...}
282
283 +function factsegur_claim_management_claim_type_load($claim_type_id) {...}
297
298 +function state_load($state_id) {...}
301
302 +function factsegur_claim_management_state_load($state_id) {...}
316
317 +function claim_process_load($claim_process_id) {...}
320
321 +function factsegur_claim_management_claim_process_load($claim_process_id) {...}
545
546 /** Implements hook_exit(). ...*/
550 +function factsegur_claim_management_exit($destination = NULL) {...}
554
555 +function factsegur_claim_management_set_return_path($url){...}
578
579 +function factsegur_claim_management_get_return_path(){...}

```

Imagem 27: Exemplo estrutural de um ficheiro `.module`

Pode-se ver aqui o conjunto de funções que se encontram no interior.

O módulo criado poderá ser habilitado na página de gestão de módulos do *Drupal*, visível na imagem 28.

ENABLED	NAME	VERSION	DESCRIPTION
<input checked="" type="checkbox"/>	Advanced Search Module		Requires: Views (enabled), Chaos tools (enabled), Views Dynamic Fields (enabled), Views UI (enabled), Views PHP (enabled)
<input checked="" type="checkbox"/>	Factsegur Claim Management		Module for claim management system Requires: Views (enabled), Chaos tools (enabled), Field Group (enabled), Field (enabled), Field SQL storage (enabled), Factsegur Contacts Management (enabled), Date (enabled), Date API (enabled), Date popup timepicker (enabled), Libraries (enabled), System (enabled), Date Popup (enabled)
<input checked="" type="checkbox"/>	Factsegur Contacts Management		Module for contacts management (single and enterprise). Requires: Views (enabled), Chaos tools (enabled), Date (enabled), Date API (enabled), Date popup timepicker (enabled), Libraries (enabled), System (enabled), Date Popup (enabled) Required by: Factsegur Claim Management (enabled), Factsegur External Users (enabled)
<input checked="" type="checkbox"/>	Factsegur External Users		Module for external users management. Requires: Views (enabled), Chaos tools (enabled), Factsegur Contacts Management (enabled), Date (enabled), Date API (enabled), Date popup timepicker (enabled), Libraries (enabled), System (enabled), Date Popup (enabled)
<input checked="" type="checkbox"/>	Factsegur Internal Users		Module for internal users management. Requires: Views (enabled), Chaos tools (enabled)
<input checked="" type="checkbox"/>	Factsegur Log		Module for log viewing. Requires: Views (enabled), Chaos tools (enabled), Date (enabled), Date API (enabled), Date Popup (enabled), Date popup timepicker (enabled), Libraries (enabled), System (enabled), Date Views (enabled)
<input checked="" type="checkbox"/>	Factsegur Messages		Show the various messages generated by the other users of the system. Requires: Views (enabled), Chaos tools (enabled)

Imagem 28: Página de módulos do Drupal

À esquerda, na imagem, pode-se ver o conjunto de módulos relativos a este projeto devidamente ativados no sistema.

A principal preocupação na programação destes componentes prende-se na integração eficaz com os módulos já existentes. Quando uma determinada funcionalidade pretendida já estava contemplado no *Drupal* ou existia um módulo com a capacidade de a executar, era integrada nos módulos em construção.

### 6.4.1.2 Validação e submissão de *forms*

Quando é efetuada uma submissão de um *form*, ocorre a validação do seu conteúdo e, em seguida, a sua submissão.

Para implementar estes dois passos no *Drupal*, é necessário implementar as função `<nome_do_form>` para criação do *form*, `<nome_do_form>_validate` para efetuar a sua validação e `<nome_do_form>_submit` para efetuar a sua submissão.

Quando um utilizador carrega no botão de submissão, tal como se pode ver na imagem 30, é inicialmente efetuada a validação. No caso de ser encontrado algum erro, é mostrada uma mensagem ao utilizador com a respetiva informação<sup>66</sup>.

<sup>66</sup> Quando é encontrado um erro, é necessário chamar a função `form_set_error`, que irá fazer com que o método de submissão não executado e irá mostrar uma mensagem de erro ao utilizador [70].



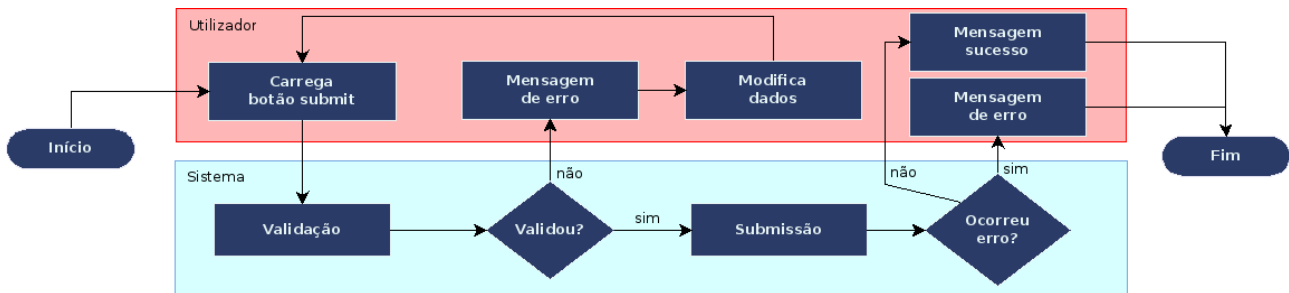


Imagem 29: Diagrama de fluxo de validação e submissão de um form

Quando é executado o método de submissão do *form*, caso seja encontrado algum erro, o utilizador também será notificado<sup>67</sup>. Em caso de sucesso, os dados são guardados na base de dados.

### 6.4.1.3 O que é o *Menu routing*

Este é um componente essencial do *Drupal*, recebendo um URL da página pretendida e retornando o conteúdo página web correspondente.

Se tivermos um *url* do género *http://[url\_site\_drupal]/pagina/item*, a parte “pagina/item”, à qual se dá o nome de *route*, vai ser tratada de modo a retornar o resultado correspondente [71].

Como se pode visualizar na imagem 30, quando um utilizador introduz o *url* este irá ser traduzido pelo módulo de *Menu Routing* do *Drupal Core*, que se anteriormente se encarregara da formação de uma tabela com as correspondências entre uma determinada *route* e o respetivo módulo e função ao qual diz respeito<sup>68</sup>.

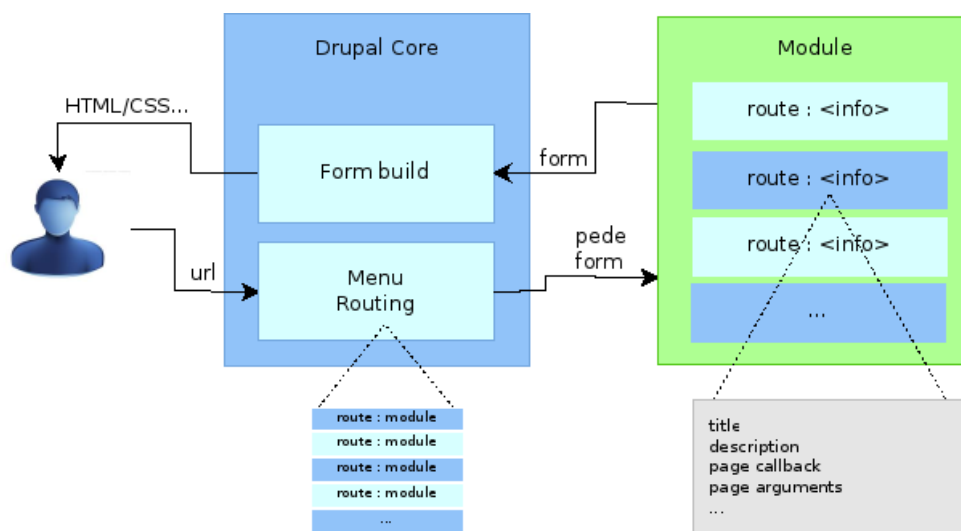


Imagem 30: Menu routing no Drupal

<sup>67</sup> No caso da submissão, as mensagens são enviadas para o utilizador através da função *drupal\_set\_message()* e, em caso de erro ou sucesso, é guardada esta informação na tabela de log através do método *watchdog()* do Drupal.

<sup>68</sup> Inicialmente, o módulo também passa a informação de *routing* ao *Menu Routing* do *Drupal*, através da função *hook\_menu*.

Quando a função correspondente é executada, é devolvido um *form*<sup>69</sup> que será tratado pelo *form building* do *Drupal* e devolvido sob a forma de HTML ao utilizador.

A informação de *routing* aceita um conjunto de argumentos, sendo os utilizados durante a implementação deste projeto apresentados na tabela 5.

<b>title</b>	Obrigatório. O título do <i>menu item</i> .
<b>description</b> <sup>70</sup>	A descrição do menu item.
<b>page callback</b>	A função chamada para mostrar a página web, quando o utilizador escolhe um determinado <i>path</i> .
<b>access callback</b>	Função que determina se o utilizador tem acesso ou não ao conteúdo (boolean).
<b>access arguments</b>	Um <i>array</i> de argumentos para passar para a função de <i>access callback</i> .
<b>file</b>	Ficheiro onde se encontra a função pretendida.
<b>type</b>	<i>Bitmask flag</i> que indica propriedades do menu item.

Tabela 5: Informação utilizada pelo menu routing [72]

Estes dados são então interpretados pelo *Drupal* quando é feito um acesso a uma determinada parte do sistema.

#### 6.4.1.4 O que são *Auto-loader wildcards*

Quando se utiliza uma *route* e na sua composição está um item com o formato *%[item\_name]*<sup>71</sup>, está-se a dar uso ao sistema de *wildcards* do *Drupal*.

Uma *wildcard* é substituível por qualquer valor e será tratado pela função à qual está associada uma determinada *route*. A função, obviamente, terá de saber o que fazer com o valor [73].

Em determinados casos, pode-se ainda utilizar este sistema aliado ao carregamento automático de um objeto ao qual essa *wildcard* diga respeito<sup>72</sup>.

69 Também podem ser efetuadas outras ações, dependendo daquilo que se pretende.

70 Embora os outros dois campos seja obrigatórios, este é apenas recomendado.

71 Por exemplo <http://sitenome.com/node/%node>, com route a ser *node/%node*.

72 Utilizando, neste caso, o sistema de *auto-loading wildcards*.

Conforme se pode ver na imagem 31, quando é utilizada uma *wildcard*, o *routing* do *Drupal* vai encaminhar para o módulo e função correta, mas antes de ser executada a função, é executado o código do *loader* da *wildcard*, obtendo os dados que lhe dizem respeito.

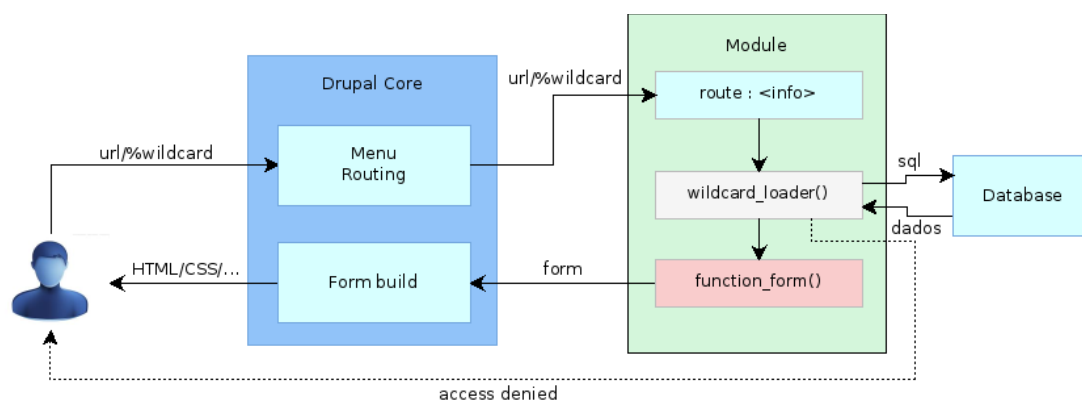


Imagem 31: Método de funcionamento dos auto-loading wildcards

Além disso, a função da *wildcard* pode ainda ser um mecanismo de verificação de permissões de acesso, sendo negado o acesso a um determinado conteúdo em questão se a função retornar *FALSE*.

### 6.4.1.5 Views

O módulo de *Views* é essencial para este projeto tanto mais que, na maioria das vezes, quando é apresentada uma tabela com dados, é utilizado este módulo <sup>73</sup>.

Para efetuar a filtragem dos dados contidos na tabela, são utilizados os *exposed filters*, campos que permitem ao utilizador introduzir dados relativamente a um ou mais *items* apresentados<sup>74</sup>.

Para um tratamento prévio dos dados, são utilizados *handlers*. Um *handler* é um objeto que é parte da *view* e do fluxo de construção da *query*. São objetos; geralmente, os *handlers* base<sup>75</sup> irão bastar, mas com alguma frequência existe a necessidade de complementar<sup>76</sup> o *handler* base de modo a atingir um outro objetivo [74].

Um *handler* funciona como um identificador dos campos das tabelas contidas na base de dados às quais uma *view* pode ter acesso e com as quais pode construir resultados, funcionando assim como um intermediário entre o módulo e a informação que este disponibiliza às *Views*. Além disso, indica de que forma essa informação irá ser tratada e como irá ser apresentada.

<sup>73</sup> Para mais informações sobre *Views*, consultar a secção **6.4.3.5 Views**.

<sup>74</sup> O *refresh* da página está configurado para ser feito por AJAX.

<sup>75</sup> *Handlers* fornecidos pelo módulo de *Views*, de raiz.

<sup>76</sup> *Need to override*.

### 6.4.1.6 Permissões

Cada um dos módulos deste projeto tem implementada uma função que define uma lista de privilégios que o utilizador terá de possuir para poder aceder aos seus diversos componentes<sup>77</sup>. Essas permissões são depois definidas em diversos pontos do código<sup>78</sup>, e podem ser configurados o tipo de acesso de cada utilizador através do módulo de permissões efetuado pelo outro estagiário.

### 6.4.1.7 Sistema de ficheiros do Drupal

Embora cada um dos módulos tenha a sua própria implementação, o acesso ao sistema de ficheiros é feito através de um mecanismo comum.

O *Drupal* possui duas formas de lidar com os ficheiros: tratá-los como sendo públicos, acessíveis ao utilizador e ao exterior, ou como sendo privados, fornecendo mecanismos de acesso intra-modulares e por utilizador. Por motivos de segurança, é utilizada a segunda forma.

Na imagem 32 é apresentado de forma esquemática a forma como é gerido o sistema de upload de ficheiros pelo Drupal.

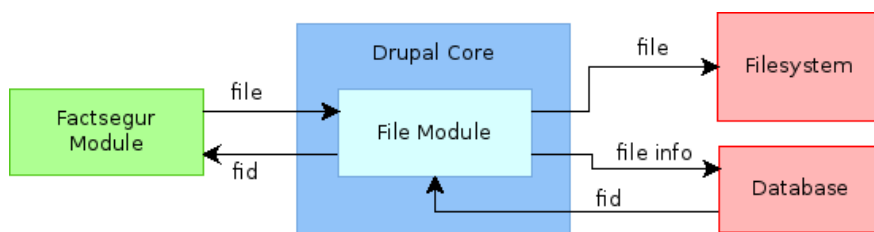


Imagem 32: Gestão de ficheiros (upload)

Quando um módulo faz um pedido de armazenamento do ficheiro, é o módulo de gestão de ficheiros do *Drupal* que se encarrega dessa tarefa, sendo efetuado o upload do ficheiro para o servidor e, quando é feita a confirmação para que o ficheiro seja guardado permanentemente, essa informação é colocada na tabela apropriada da base de dados e o ficheiro colocado na pasta correta, configurada anteriormente no *Drupal*.

Por outro lado, temos o processo de download de um ficheiro, representado na imagem 33.

<sup>77</sup> Esta lista de permissões é retornada pela função `hook_permissions()`.

<sup>78</sup> Por exemplo, podem ser definidas diretamente nas *routes*.

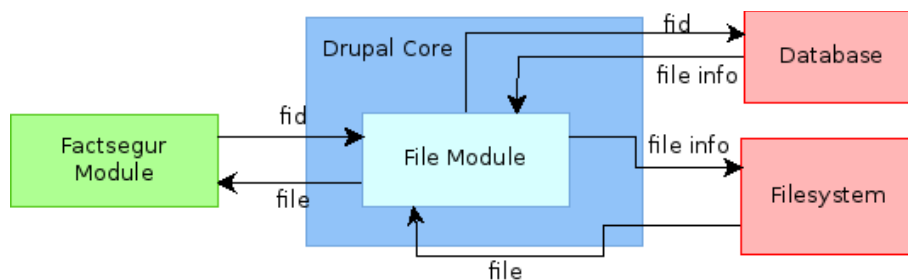


Imagem 33: Gestão de ficheiros (download)

Para a obtenção do ficheiro, o módulo comunica com o sistema de gestão de ficheiros do *Drupal*, fornecendo o *fid*<sup>79</sup> do ficheiro pretendido. É feito um pedido à base de dados sobre informações do ficheiro, sendo em seguida obtido do sistema de ficheiros.

## 6.4.2 Listagem de módulos produzidos

Nesta secção serão apresentados os módulos produzidos durante a fase de implementação. Discutir-se-ão todos os aspetos relevantes, tais como os desafios ultrapassados para a sua implementação, bem como dependências que têm relativamente a outros módulos.

### 6.4.2.1 Pesquisa avançada

Embora este não tenha sido o primeiro módulo a ser realizado, faz sentido ser o primeiro a ser apresentado pois o seu funcionamento depende quase totalmente da informação fornecida pelos outros módulos.

O principal aspeto a ter em atenção aquando da criação deste módulo era torná-lo totalmente independente dos restantes, de modo a que pudesse apresentar os resultados pretendidos de uma pesquisa independentemente dos módulos instalados. Para tal, utilizou-se o sistema de *hooks* do *Drupal*, de modo que é função dos restantes módulos avisá-lo caso possuam *views* para pesquisas avançadas<sup>80</sup>.

Na imagem 34 é esquematizado o modo como se processa a construção do *form* de pesquisa avançada.

79 File ID

80 Através do `hook_form_alter` ou variações [75].

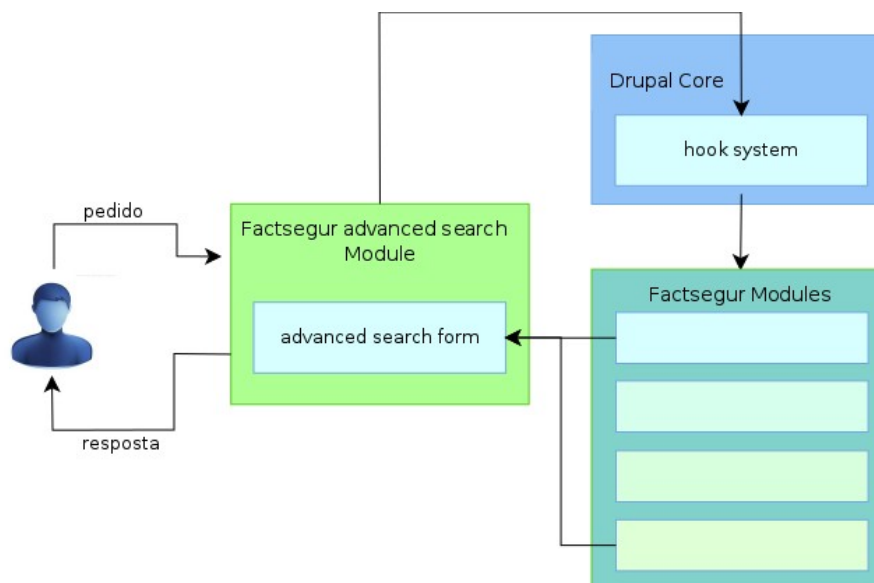


Imagem 34: Construção do form da pesquisa avançada

Quando é feito um pedido por parte de um utilizador, o *core* do *Drupal* entra em ação e procura, entre os módulos existentes, aqueles que querem efetuar alguma alteração ao *form* da pesquisa avançada, indicando assim que possuem meios para que os seus dados possam ser pesquisados neste interface.

Conforme se pode ver na tabela 6, este módulo apresenta apenas duas *routes*, sendo que a diferença essencial entre elas é que a primeira *route* disponibiliza apenas a lista para escolha da pesquisa e a segunda apresenta a lista de escolhas e a *view* para pesquisa que foi escolhida anteriormente.

Routing Table	
<code>home/advanced_search</code>	Mostra uma listagem de módulos nos quais se pode efetuar uma pesquisa avançada, podendo-se escolher um deles.
<code>home/advanced_search/%</code>	Além de fazer o mesmo que a <i>route</i> anterior, mostra uma <i>view</i> onde se podem escolher os diversos <i>items</i> com os quais se pode efetuar uma pesquisa avançada dentro de um módulo.

Tabela 6: Tabela de routing do Factsegur advanced search

Internamente, este módulo está totalmente integrado com o módulo *Views Dynamic Fields*,<sup>81</sup> estando este escondido do interface, permitindo assim que os campos seja mostrados e atualizados a pedido do utilizador.

81 Para mais informações sobre este módulo, consultar a secção 6.4.3.6.

### 6.4.2.2 Log

O *Drupal* possui um sistema de *logging* interno, tendo o módulo de *logging* afeto a este projeto que lidar unicamente com a disponibilização e visualização de mensagens geradas pelos outros módulos. De cada vez que realizam uma ação que necessita de ser registada, estes executam uma função específica do *Drupal*<sup>82</sup>, conforme se pode visualizar na imagem 35.

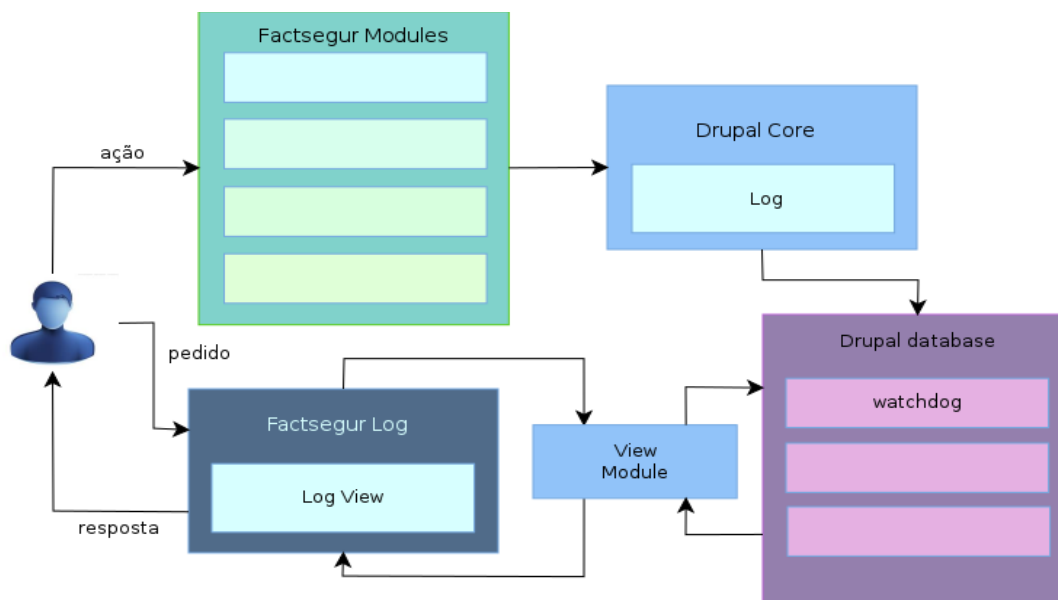


Imagem 35: Armazenamento e obtenção de logs

Optou-se pelo armazenamento de uma mensagem de *log* quando é realizada uma determinada ação no sistema que o justifique<sup>83</sup>. Essas mensagens são então guardadas na base de dados<sup>84</sup>, com um identificador específico<sup>85</sup> de modo a que possam ser depois identificadas na *view* como sendo do conjunto de módulos realizados neste estágio.

Para que os dados constantes na tabela de *logs* possam ser tratados pelo módulo de *views*, foi necessária a codificação de *handlers*<sup>86</sup> e só em seguida foi possível construir a vista para exibição dos dados.

Quando um utilizador acede ao interface de *logging*, é iniciada a construção de uma *view* que vai inquirir diretamente a base de dados para a obtenção dos dados, sendo construída e devolvida para ser apresentada pelo módulo ao utilizador.

Na tabela 7, é apresentada a lista de routes deste módulo.

82 Função *watchdog()* [76].

83 Neste momento, no caso deste projeto, conforme definido nos requisitos, é efectuado um *log* quando ocorre alguma ação de criação, atualização ou remoção de dados da base de dados.

84 Tabela *watchdog*.

85 Na tabela *watchdog*, o campo *type* tem como função o armazenamento do tipo de mensagem. Quando é gerada uma mensagem de log por um dos módulos deste projeto, é adicionado o prefixo *factsegur\_*.

86 Ver secção 6.4.1.5.

Routing Table	
home/definitions/log	Mostra a uma <i>view</i> contendo as mensagens de logs geradas pelos módulos factsegur.

Tabela 7: Tabela de routing do módulo Factsegur Log

Como se pode ver, existe apenas uma *route* definida, estando ela encarregada da apresentação da *view* com a listagem de *logs*.

### 6.4.2.3 Utilizadores internos

Antes de mais, convém lembrar que um utilizador interno é o tipo de utilizador que representa os colaboradores de uma empresa de peritos de seguros <sup>87</sup>.

Dado que o *Drupal* já contava com a capacidade de gestão de utilizadores<sup>88</sup> e de sessões<sup>89</sup>, o objetivo deste módulo seria a integração com o módulo já existente.

Conforme se pode ver na imagem 36, quando um utilizador pretende modificar alguma informação referente ao módulo de utilizadores internos, é feito um pedido ao módulo *User* do *Drupal* pelo seu respetivo *form* e, em seguida, são adicionados os campos extra pretendidos<sup>90</sup>.

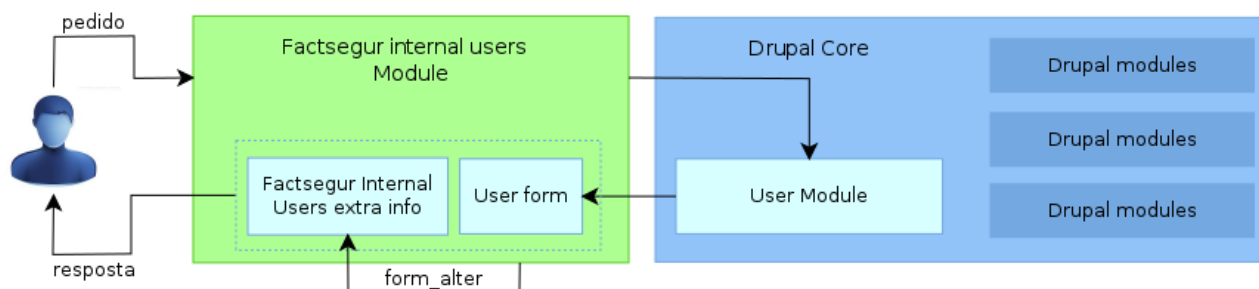


Imagem 36: Exemplo de apresentação de um form

Da mesma maneira, quando o utilizador cria ou edita a sua informação ou a de outro utilizador, o módulo de utilizadores internos primeiramente faz um pedido ao módulo de utilizadores do *Drupal* pelo seu *form*. Quando este *form* é recebido, é efetuada a alteração e adicionados os campos necessários, já preenchidos com os respetivos dados.

Dada a coexistência necessária entre os dois módulos, um dos principais desafios a ultrapassar era a necessidade de integridade entre os dados guardados pelos dois

<sup>87</sup> Por outro lado, temos o utilizador externo que representa um cliente com acesso a determinadas partes do sistema.

<sup>88</sup> O módulo *User* do *Drupal* conta já com campos para guardar nome do utilizador, email, palavra-passe, status, permissões e fotografia.

<sup>89</sup> Autenticação, registos, etc.

<sup>90</sup> Os campos extra adicionados servem para guardar informação sobre o endereço, código-postal, cidade, número de telefone (fijo e móvel), endereço de email e comentários.



módulos na base de dados, dado dizerem respeito ao mesmo utilizador. Sendo assim, quando um utilizador era criado ou apagado, tinha de haver uma comunicação entre ambos os módulos para que, em caso de erro, fossem canceladas as modificações efetuadas pelo outro.

De acordo com a imagem 37, quando é iniciado o processo de criação de um novo utilizador, o módulo *Users* tenta guardar a informação. Caso seja bem sucedido, vai ser a vez do módulo de utilizadores do Factsegur tentar guardar os dados. Em caso de sucesso, os dados são guardados e é apresentada uma mensagem a indicá-lo.

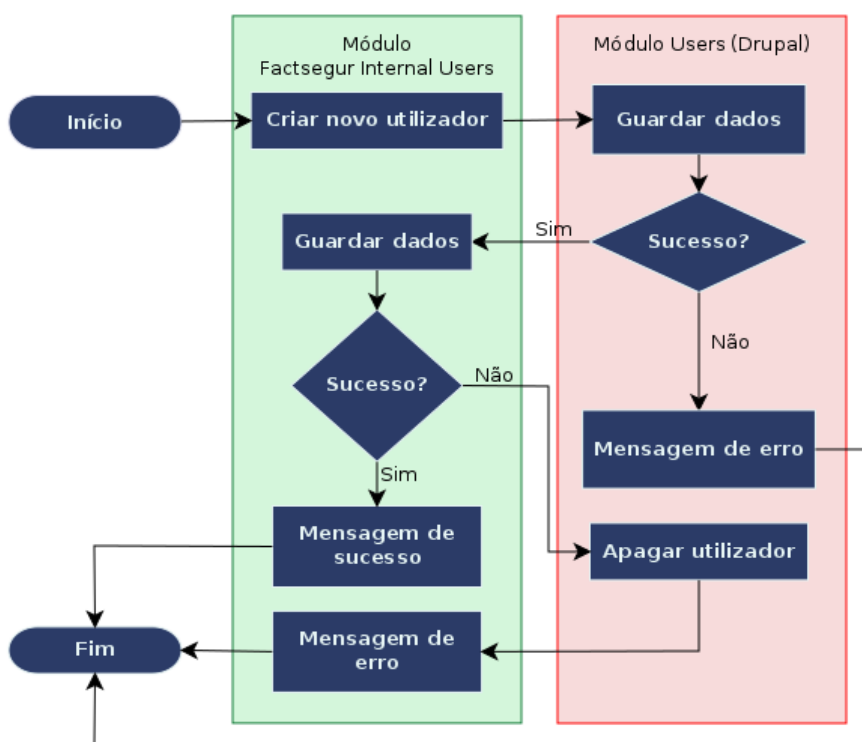


Imagem 37: Diagrama de fluxo para a criação de um utilizador

No caso do módulo *Users* do *Drupal* encontrar um erro, é gerada uma mensagem de erro e nada mais é feito. No caso de um erro ocorrer no módulo Factsegur, é ordenado que o módulo correspondente do *Drupal* apague o utilizador que criou anteriormente e seja gerada uma mensagem de erro para avisar o utilizador que ocorreu um erro.

Um outro desafio ultrapassado é o caso da remoção de um utilizador, levantando mais uma vez problemas de integridade de dados, no caso de ocorrer algum erro num dos módulos.

Como é mostrado na imagem 38, em primeiro lugar tenta-se apagar os dados do utilizador criados pelo módulo *User*, e só em seguida, em caso de sucesso, são

apagadas as informações gerados pelo módulo Factsegur.

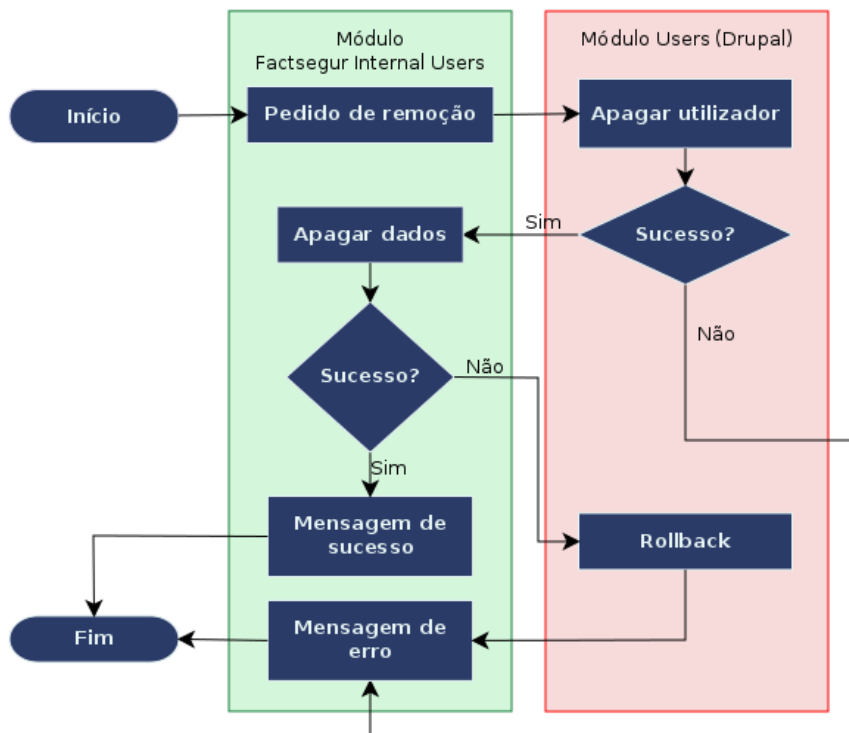


Imagem 38: Diagrama de fluxo da remoção de um utilizador

No caso deste passo falhar, é realizado um *rollback* que retorna a base de dados ao seu estado original, antes do pedido de remoção.

Finalmente, é apresentada a lista de *routes* deste módulo, na tabela 8.

Routing Table	
<code>home/definitions/users/profile</code>	Permite a edição do perfil do utilizador que se encontra autenticado.
<code>home/definitions/users/profile/edit/%internal_user</code>	Permite a edição do perfil de um outro utilizador.
<code>home/definitions/users/profile/view/%internal_user</code>	Mostra o interface de visualização de um utilizador interno, sem hipótese de edição.
<code>home/definitions/users/profile/delete/%internal_user</code>	Mostra o interface para apagar um utilizador. Caso o utilizador escolha a remoção do utilizador do sistema, este é removido.
<code>home/definitions/users/add</code>	Mostra o interface para a adição de um novo utilizador ao sistema.
<code>home/definitions/users/list/internal</code>	Mostra uma listagem de utilizadores internos.
<code>home/advanced_search/users/internal</code>	Permite a realização de uma pesquisa com mais items de procura entre os utilizadores internos. Parte integrada com o módulo de pesquisa avançada.

Tabela 8: Tabela de routing do módulo Factsegur Internal Users

Existe um total de 7 *routes* definidas, que lidam com aspetos tais como a visualização, adição, edição e remoção de utilizadores.

#### 6.4.2.4 Processos

O desafio principal deste módulo era a construção de raiz de um mecanismo com capacidade de armazenamento, obtenção e geração de relatórios, bem como o armazenamento de ficheiros anexos a um relatório de regulação de sinistros.

Como se pode ver, pelo diagrama de fluxo apresentado na imagem 39, quando um utilizador pretende criar um novo processo terá de, obrigatoriamente, preencher os campos do bloco “Dados Internos”, na *tab* “Dados gerais”, podendo guardar o processo nesse momento, ou então preencher os restantes campos pretendidos. Além disso, poderá ainda adicionar anexos e outros ficheiros ao processo<sup>91 92</sup>.

91 Os anexos são imagens que irão integradas no documento gerado pela módulo de processos. Os ficheiros são quaisquer tipos de ficheiros que o utilizador queira ver integrados na ficha do processo, mas não farão parte do documento, Poderão ser, por exemplo, diversas versões dos documentos gerados para um determinado processo em particular.

92 Dado que o processo de armazenamento e obtenção de ficheiros por parte do Drupal já foi falado anteriormente, apenas será aqui referida a existência desta funcionalidade.

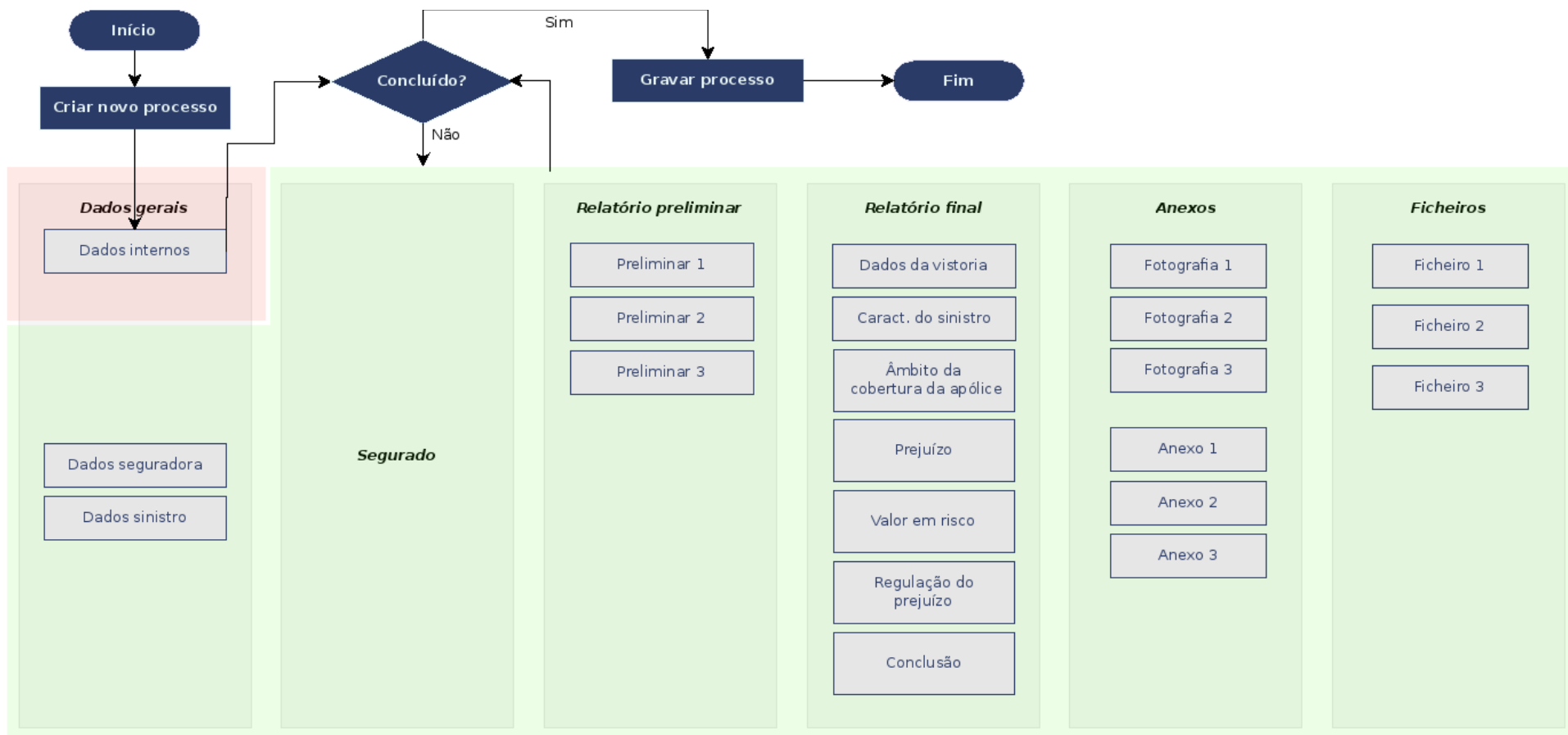


Imagem 39: Diagrama de fluxo de criação de novo processo

Após o armazenamento do processo na base de dados, irá ser disponibilizada a opção para geração de documentos tendo como base os dados contidos no processo ao qual diz respeito.

Existem dois tipos de ficheiros que podem ser gerados a partir de um processo, o relatório preliminar e o relatório final<sup>93</sup>. Cada um deles possui um *template* associado para a sua geração, mas o processo de construção é similar, resumido na imagem 40.

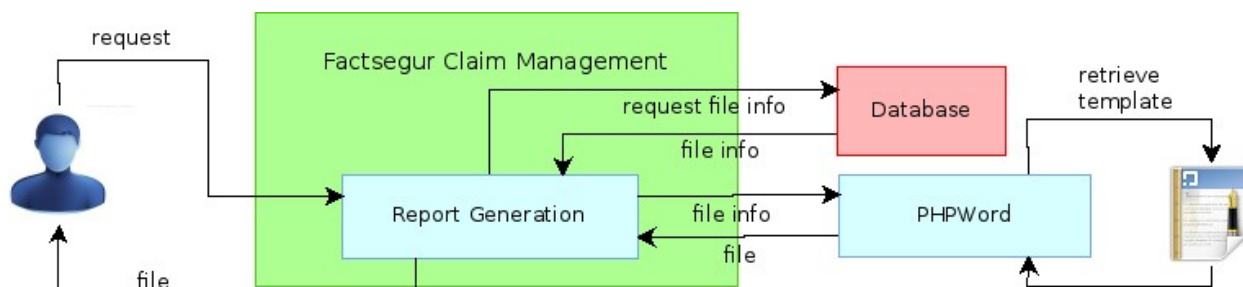


Imagem 40: Geração de um relatório

Quando um utilizador requer a geração de um relatório de um processo, a base de dados é inquirida acerca das informações contidas nele. Em seguida, essa informação é passada para o *PHPWord*<sup>94</sup>, que obtém o ficheiro *template* correto e preenche os campos no seu interior com a informação correspondente. No final, o ficheiro gerado é disponibilizado ao utilizador, no formato *docx*.

Além das funcionalidades descritas anteriormente, também foram implementados mecanismos para gestão<sup>95</sup> de tipos de sinistro, estados do processo e ramos, campos disponíveis para escolha no formulário do processo.

Em seguida, na tabela 9, temos a lista de *routes* definidas para este módulo.

93 Estes dois tipos de relatório diferem no seu conteúdo pelos seguintes aspetos: um relatório preliminar contém um conjunto de impressões prévias relativamente a um determinado sinistro, não sendo vinculativo. Um relatório final contém já todo o conjunto de valores associados aos prejuízos e devidas ilações acerca da validade das reivindicações por parte do segurado.

94 *PHPWord* é uma biblioteca escrita em PHP, que fornece um conjunto de classes para ler e escrever diferentes formatos de ficheiros. No momento da escrita deste relatório, a versão utilizada suportava Microsoft Office Open XML (OOXML ou OpenXML), OASIS Open Document Format for Office Applications (OpenDocument or ODF), Rich Text Format (RTF), HTML e PDF [77].

95 Listagem, edição, adição e remoção.

Routing Table	
<code>home/claim_management/claim/add</code>	Mostra o interface para a criação de um novo processo.
<code>home/claim_management/claim/null/%claim_process</code>	Mostra o interface para a anulação do processo com id=%claim_process.
<code>home/claim_management/claim/edit/%claim_process</code>	Mostra o interface de edição para o processo com id=%claim_process.
<code>home/claim_management/claim/list</code>	Mostra o interface de listagem de processos aos quais o utilizador atual tem acesso.
<code>home/claim_management/claim/branches</code>	Mostra o interface com a lista dos ramos de seguros que existem atualmente no sistema.
<code>home/claim_management/claim/branches/edit/%branch</code>	Mostra o interface de edição de ramos para o ramo com id=%branch.
<code>home/claim_management/claim/branches/delete/%branch</code>	Mostra o interface de remoção do ramo com id=%branch.
<code>home/claim_management/claim/claim_types</code>	Mostra o interface com a listagem dos tipos de sinistros existentes atualmente no sistema.
<code>home/claim_management/claim/claim_types/edit/%claim_type</code>	Mostra o interface de edição para o tipo de sinistro com id=%claim_type.
<code>home/claim_management/claim/claim_types/delete/%claim_type</code>	Mostra o interface de remoção para o tipo de sinistro com id=%claim_type.
<code>home/claim_management/claim/states</code>	Mostra o interface com a listagem de estados existentes atualmente no sistema.
<code>home/claim_management/claim/states/edit/%state</code>	Mostra o interface para edição de um determinado estado com id=%state.
<code>home/claim_management/claim/states/delete/%state</code>	Mostra o interface de remoção de um determinado estado com id=%state.
<code>home/claim_management/claim/generate_preliminar/%claim_process</code>	Gera um relatório preliminar com os dados contidos num processo, a partir de um <i>template</i> previamente fornecido.
<code>home/claim_management/</code>	Gera um relatório preliminar com os dados contidos num

claim/generate\_final/  
%claim\_process

processo, a partir de um *template* previamente fornecido.

Tabela 9: Tabela de routing do módulo de processos

Estando definidas 15 *routes* ao todo, estas lidam com os aspetos relacionados com a gestão de processos de regulação de sinistros.

### 6.4.2.5 Mensagens

O módulo de mensagens permite o envio de mensagens entre utilizadores e, além disso, fornece um sistema de atribuição de tarefas.

A imagem 41 mostra o funcionamento da parte de envio de mensagens deste módulo.

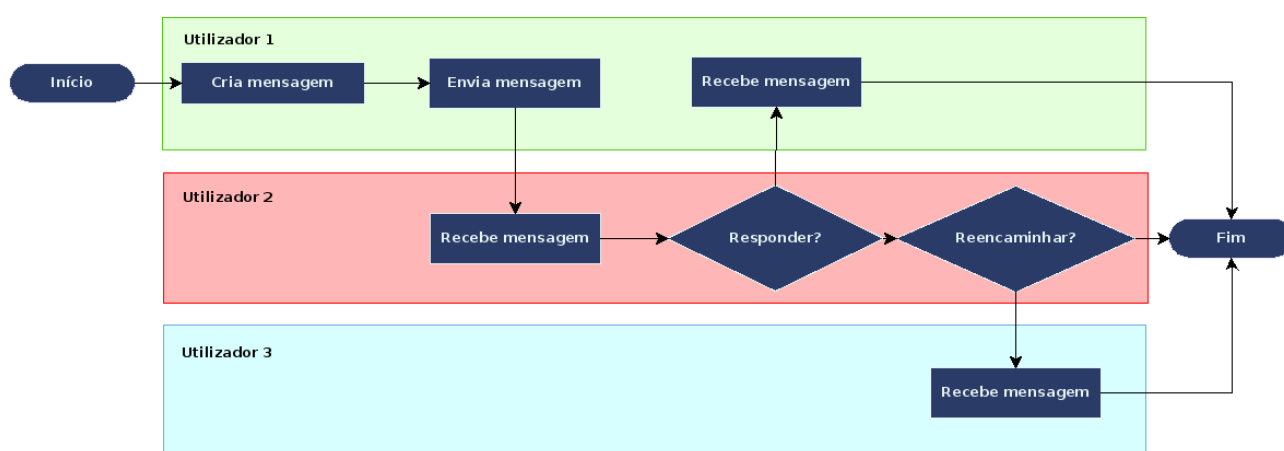


Imagem 41: Diagrama de fluxo de envio de mensagem

Quando um utilizador cria uma mensagem, esta é enviada para o utilizador alvo, que nesse momento pode escolher entre duas opções: responder ou reencaminhar. No caso de responder, o utilizador original recebe a sua mensagem de resposta. A opção de reencaminhamento permite que a mensagem seja enviada para um outro utilizador. De referir ainda que este sistema de mensagens possibilita a agregação de ficheiros ao seu conteúdo, permitindo assim a sua partilha. A apresentação da lista de mensagens fica a cargo de um conjunto de *views* desenhadas para esse efeito.

O sistema de atribuição de tarefas tem uma implementação interna muito similar ao de envio de mensagens, mas o seu funcionamento é ligeiramente diferente, de modo a adequar-se à sua função.

Quando uma tarefa é criada, esta pode a partir desse momento ser visualizada pelo autor e pela pessoa escolhida para a desempenhar. O recetor não pode apagar a tarefa, apenas modificar o estado para concluída, caso seja esse o caso. Apenas o criador da tarefa pode apagá-la<sup>96</sup>, podendo visualizar qual o estado em que esta se

<sup>96</sup> Na realidade a tarefa não é apagada fisicamente, é apenas modificada uma flag que faz com que não seja mostrada nas *views*.

encontra pela view respetiva. Para tal, encontram-se implementadas 3 *views* onde o utilizador pode ver as tarefas que atribuiu, as que lhe foram atribuídas e aquelas que já finalizou.

## 6.4.3 Módulos externos

Além do core do *Drupal*, foram ainda utilizados um outro conjunto de módulos externos, fornecidos pela comunidade. Nesta secção faremos a sua apresentação e será indicado o local onde são utilizados.

### 6.4.3.1 Captcha

O *CAPTCHA*<sup>97</sup> é um teste pergunta-resposta colocado frequentemente em formulários web para determinar se o utilizador é humano. O seu objetivo é bloquear submissões por parte de *spambots*, *scripts* automatizados que produzem *spam*. Este módulo fornece esta funcionalidade para virtualmente qualquer utilizador que se depare com um formulário de *Drupal* [79].

No caso desta aplicação, esta funcionalidade é utilizada no apenas *form* de requisição de nova password.

### 6.4.3.2 Field Group

Dado que o *Drupal* não suporta por defeito a construção de *tabs* horizontais na sua API, das quais podemos ver um exemplo na imagem 42.

Imagem 42: Exemplo de tabs no Drupal

Foi então necessário arranjar uma solução para este problema, de modo a que o interface estivesse de acordo com o que foi estipulado nos requisitos.

97 Acrónimo de “Completely Automated Public Turing Test To Tell Computers and Humans Apart” [78].



Embora não esteja a ser utilizado com esse intuito, este módulo *Field Group* permite o agrupamento de campos. Além disso, disponibiliza diversos formatos para a sua apresentação, entre eles o *horizontal tabs* [80]. Sendo assim, em termos de código, é aproveitada esta funcionalidade aquando da construção de *tabs* horizontais, no módulo de gestão de processos.

### 6.4.3.3 Date

Este módulo contém um *field date/time* flexível, um tipo *Date field* e um Date API que outros módulos podem utilizar, em conjugação com o que já é fornecido pelo *Drupal* [81].

O principal motivo para a utilização deste módulo, é o facto de disponibilizar um *popup calendar* para a escolha da data num *field date/time*, tal como se pode ver na imagem 43, algo que não é implementado de raiz pelo *Drupal*.

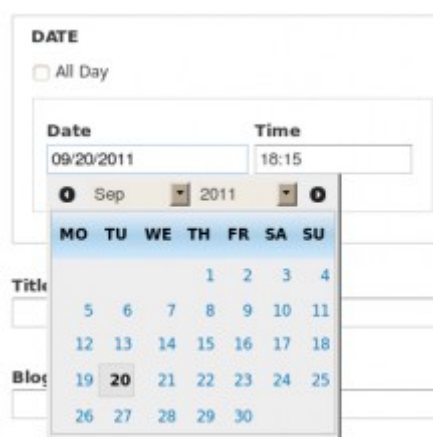


Imagem 43: Exemplo de um Date Popup

Encontra-se em utilização nos módulos onde são utilizados campos de datas como, por exemplo, o módulo de processos.

### 6.4.3.4 Date Popup Timepicker

O *Date Popup Timepicker* adiciona mais opções para elementos de *date\_popup* e *date\_popup\_widgets* para campos date fornecidos pelo módulo Date [82], sendo este módulo utilizado no âmbito deste projeto para adicionar a opção de se poder colocar horas e minutos num campo *date*.

### 6.4.3.5 Views

De uma forma geral, este módulo permite mostrar os dados dos diversos módulos de um site *Drupal*, bem como tratar do aspeto da apresentação dos mesmos. É um módulo com alguma complexidade e que oferece um conjunto variado de

configurações e opções, tal como se pode ver na imagem 44 [83].

The screenshot shows the 'Displays' configuration page for a view named 'Block'. The interface is organized into several columns and sections:

- Block details:** Shows the display name 'Block' and a 'clone Block' button.
- TITLE:** Title is 'Factsegur Internal Users - List Internal Users'.
- FORMAT:** Format is 'Table'.
- FIELDS:** Lists fields such as '(Users) User: Uid (Uid)', '(Users) User: Picture (Photo)', '(Users) User: Name (User Name)', '(Users) Factsegur External Users: Enterprise ID (Enterprise ID)', '(Users) User: E-mail (Email)', 'Global: PHP', and 'Global: Custom text'.
- FILTER CRITERIA:** Lists criteria like '(Users) User: Name (exposed)' and '(Users) User: E-mail (exposed)'.
- SORT CRITERIA:** Lists criteria like '(Users) User: Name (asc)'.
- BLOCK SETTINGS:** Block name is 'Factsegur Internal Users...', Access is 'Permission | View user profiles'.
- HEADER, FOOTER, PAGER:** Options for adding header/footer and pager settings (e.g., 'Use pager: Full | Paged, 10 items').
- Advanced:** A section for advanced settings including 'CONTEXTUAL FILTERS', 'RELATIONSHIPS' (Factsegur External Users: Uid), 'NO RESULTS BEHAVIOR', 'EXPOSED FORM' (Exposed form style: Basic), and 'OTHER' (Machine Name: block, Comment: No comment, Use AJAX: Yes, etc.).

Imagem 44: Ecrã de configuração de uma vista do módulo Factsegur Internal Users

Numa vista pode-se definir o título, o formato com que os dados são apresentados<sup>98</sup>, os campos a apresentar, os filtros com os quais se pretende escolher quais serão os dados a apresentar e quais os critérios de ordenamento dos dados.

Além disso, fornece um outro conjunto de configurações avançadas tais como a utilização de filtros contextuais<sup>99</sup>, relações<sup>100</sup>, entre outras<sup>101</sup>.

Este módulo depende de um outro módulo externo, o *Chaos Tools*<sup>102</sup>.

98 Além da tabela geralmente utilizada neste projeto, também existem outras opções tal como grelha, lista HTML, etc.

99 Por exemplo, a utilização da informação do utilizador ligado nesse momento de modo a filtrar, ou id do nó onde um utilizador se encontra, de modo a filtrar os dados consoante esse valor.

100 Este aspeto permite o estabelecimento de relações entre dados contidos em diversas tabelas do mesmo ou de diferentes módulos, como por exemplo, o estabelecimento de uma relação que é efetuado entre os dados do módulo User do Drupal e o Factsegur Internal Users deste projeto..

101 Um outro aspeto relevante das Views, muito utilizado neste projeto de estágio, é a capacidade de *refresh* da página por AJAX.

102 <https://www.drupal.org/project/ctools>

### 6.4.3.6 Views Dynamic Fields

Este módulo adiciona um tipo de filtro extra às *Views*, permitindo que o utilizador escolha que campos mostrar numa vista renderizada, fornecendo uma visualização personalizada para cada utilizador, conforme as suas preferências [84].

### 6.4.3.7 Views PHP

Este é mais um módulo que amplia as capacidades das *Views*, fornecendo a possibilidade de adicionar *PHP* ao output de uma determinada célula de uma tabela, possibilitando a realização de operações mais complexas no interface de uma *view*, como por exemplo a adição de funcionalidades que não estariam previstas originalmente [85].

No entanto, conforme é indicado na página do próprio módulo, é aconselhado o uso de *handlers* e outros *plugins* quando possível, pois o uso de *PHP* na filtragem e ordenamento de uma *view*, tem um impacto considerável na performance.

Por este motivo, existe um objetivo futuro de retirar das dependências deste projeto este módulo.

## 6.4.4 Módulos independentes

Para além do conjunto de módulos relatados anteriormente, é possível ainda ter um conjunto de módulos que, em vez de ampliar as funcionalidades do sistema e oferecer novas funcionalidades de programação, funcionem como uma característica do sistema por si só<sup>103</sup>.

Integrado neste projeto está o módulo **Drupal Chat**, aparecendo na parte inferior direita do ecrã, como se pode ver na imagem 45.

---

<sup>103</sup> Neste momento existe apenas um módulo com tais características, mas é possível que no futuro sejam adicionados mais.

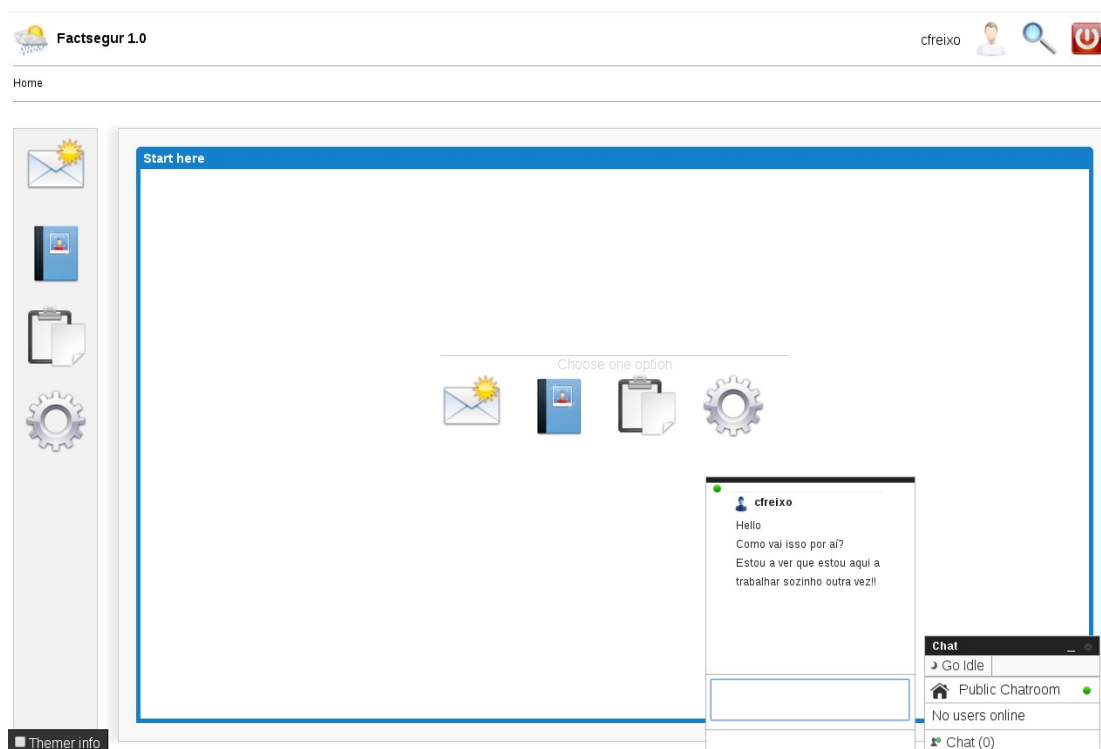


Imagem 45: Drupal Chat

Este módulo permite que os utilizadores troquem mensagens entre si num sistema de *chat* privado, ou então em salas públicas.

## 6.5 Tema

Esta secção diz respeito ao tema, ou seja, a forma como a informação será apresentada ao utilizador e ao seu aspeto gráfico.

Não se pretende aqui uma descrição exaustiva do sistema de *theming* do *Drupal*, mas sim apresentar um conjunto de tópicos sobre as questões fundamentais deste sistema e os passos necessários para a sua implementação e integração neste projeto.

Assim sendo, para que melhor se compreendam as implementações efetuadas e de modo a que os conceitos e as implementações não apareçam misturadas, foi efetuada uma pequena secção introdutória.

### 6.5.1 Conceitos introdutórios

#### 6.5.1.1 Ficheiro .info

Tal como para os módulos, também existe um ficheiro que identifica as principais

características do tema para o Drupal e, tal como nos módulos, é identificado com `<nome_do_tema>.info`.

Como se pode ser na imagem 46, em primeiro lugar são declarados o nome do tema e a sua descrição, e para que versão do Drupal foi feito.

```

name = Conflux Theme
description = The default theme for the FactSegur project
core = 7.x

; Additional info
package = FactSegur
version = "0.1"
project = "Factsegur"

; Stylesheets info
;stylesheets[all][] = css/regions.css
;stylesheets[all][] = css/layout.css

; Scripts info
scripts[] = js/functions.js

; Regions info
regions[login_container] = Login container
regions[page_top] = Page top
regions[page_bottom] = Page bottom
regions[vertical_left] = Vertical left
regions[vertical_right] = Vertical right
regions[header_top] = Header top
regions[header_bottom] = Header bottom
regions[sidebar_left] = Sidebar left
regions[content] = Content

```

Imagem 46: Exemplo do conteúdo do ficheiro `.info` de um tema

Além disso, são declarados ficheiros de *scripting* e de *css* que façam parte do tema, características<sup>104</sup> que o *Drupal* disponibiliza que estão presentes neste módulo, definições e outros conteúdos.

Algo importante a definir dentro deste ficheiro são as regiões, visíveis na imagem 47. Isso irá permitir dividir o interface do *Drupal* em secções e atribuir a cada uma delas um ou mais *items* que se pretendam colocar no seu interior.

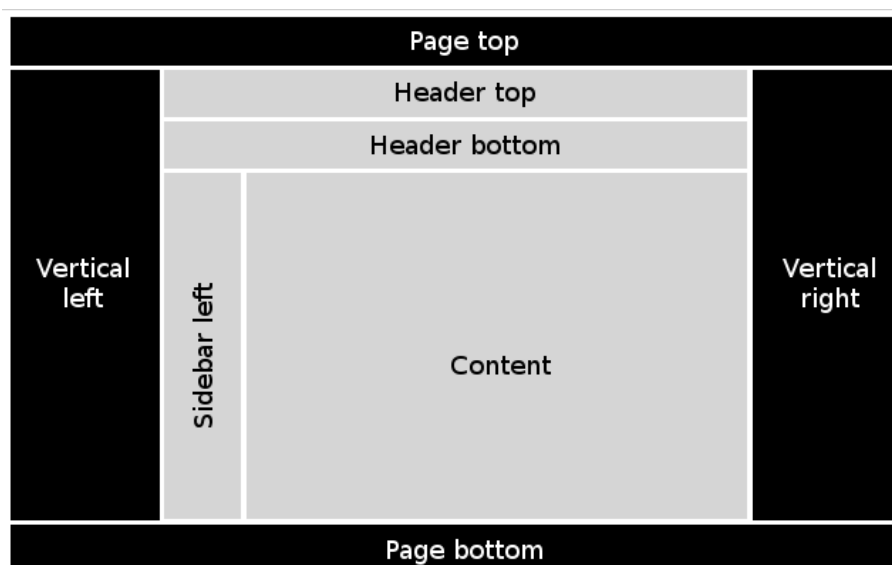


Imagem 47: Regiões definidas para o tema deste projecto

<sup>104</sup> Features.

As regiões apresentadas foram declaradas no ficheiro *.info* mas, no entanto, o seu formato e conteúdo só poderão ser definidos com recurso aos ficheiros de *template* do *Drupal*.

### 6.5.1.2 Ficheiros *.tpl.php*

O *Drupal* implementa um sistema que reconhece automaticamente pelo nome ficheiros que indicam como será o aspeto da página ou de uma determinada secção desta, de entre os ficheiros que se encontram no tema.

Esses ficheiros têm uma extensão têm o formato *<nome>.tpl.php* e contêm o código PHP que irá assistir na geração do conteúdo de uma página do *Drupal*.

Para melhor compreensão deste conceito, atente-se na imagem 48.

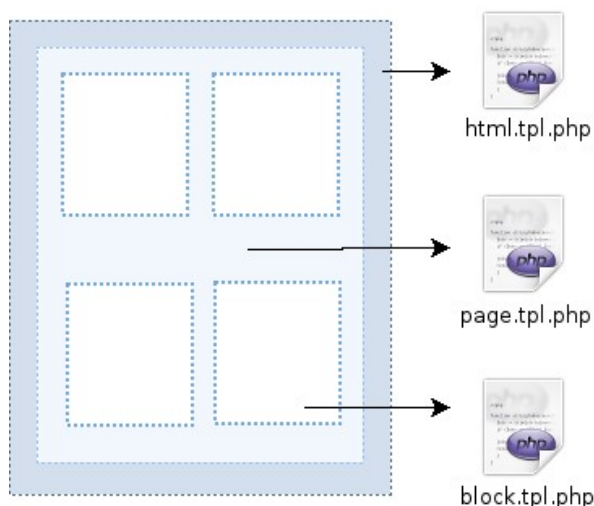


Imagem 48: Exemplo de aplicação dos ficheiros *.tpl.php*

Temos aqui um exemplo abstrato de uma página que seria construída através do uso de três *templates* em *PHP*, definindo cada um destes uma região/secção distinta<sup>105</sup>.

Quando um destes *templates* é usado pelo *Drupal* para construir conteúdo, é-lhe fornecido um *array* de variáveis relativo ao conteúdo correspondente. É então a partir desse *array* de variáveis da sua utilização dentro dos *templates* que surge a construção dinâmica das páginas do *Drupal*.

Na imagem 49 é apresentado um excerto do código do ficheiro *page.tpl.php* contido neste tema deste projeto.

<sup>105</sup> Com o template *block.tpl.php* a ser chamado 4 vezes.

```

<?php if($page['vertical_right']): ?>
  <div id="vertical-right">
    <?php print render($page['vertical_right']); ?>
  </div>
<?php endif ; ?>

<?php if(strlen($messages)>0): ?>
  <div id="dialog-message" title="Message">
    <p>
      <span class="ui-icon ui-icon-circle-check" style="..."></span>
      <?php print $messages; ?>
    </p>
  </div>
<?php endif ; //if(strlen($messages)>0)?>
:~php else : //if ($logged_in) else ?>
<?php /*if the user is logged out, two things can happen:
  - is presented the login block
  - the recover password form is presented
  Right now there's no third option. In case there will be one, it must be included here */
$_SESSION['message_login'] = TRUE;
?>

```

Imagem 49: Excerto do ficheiro page.tpl.php

Nele podemos ver a indicação do sítio onde estará o conteúdo da região “*Vertical right*”, quando é passada à função *render* a variável `$page['vertical-right']`.

### 6.5.1.3 *Template.php*

Este ficheiro contém as funções necessárias para manipular o output por defeito do *Drupal*. É um dos ficheiros mais úteis aquando da criação ou modificação de temas de *Drupal*. Com este ficheiro são possíveis três ações [86]:

- Modificar qualquer variável de *hook* de tema ou adicionar as próprias, usando funções de pré-processamento ou processamento.
- Fazer *override* de qualquer função de tema, ou seja, substituir um função do tema por defeito por outra.
- Chamar funções *hook\_\*\_alter* que permitem a alteração de várias partes do funcionamento interno do *Drupal*, incluindo a renderização de elementos de *forms*.

Como se pode ver na imagem 50, poderão ou não existir um conjunto de funções de tema definidas.

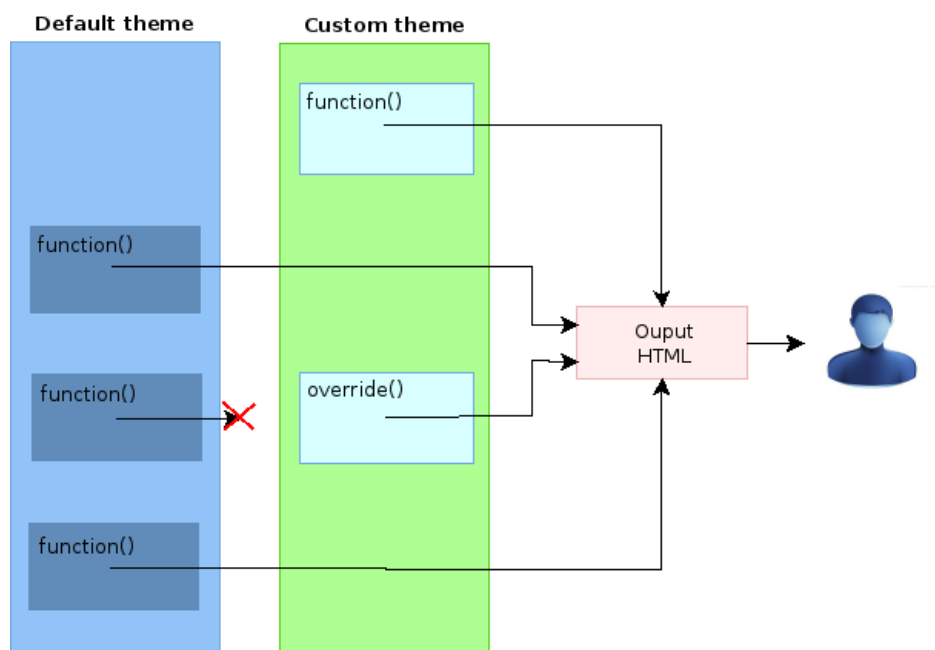


Imagem 50: Override de funções no Drupal

O possível papel do tema que está nesse momento definido será o de criar novas funções para definição do aspeto visual e ainda fazer *override* às funções pretendidas para modificação.

## 6.5.2 Implementação do tema

Após a apresentação secções anteriores, passa-se a explicar os pormenores de implementação dos componentes que constituem o tema do *Drupal*.

Em primeiro lugar foi necessário construir um novo ficheiro *page.tpl.php* que comportasse o aspeto idealizado para a aplicação, e fosse compatível com as novas regiões definidas. Nele foram ainda definidos onde se iriam encontrar outros elementos, tais como as *breadcrumbs*, logótipo, título da página, mensagens de erro, menus e conteúdo.

No ficheiro *template.php* foi implementada uma função de pré-processamento à página de modo a fazer inclusão condicional de *CSS*, conforme o utilizador estivesse autenticado ou não. Além disso, também foram incluídas bibliotecas de *jQuery UI*<sup>106</sup> para que as mensagens do *Drupal*<sup>107</sup> fossem mostradas em *dialogs*.

Para que o interface de *login* e requisição de nova password aparecessem de acordo com o definido nos requisitos, foi necessário criar uma função que “anunciase” o *override* às funções previamente utilizadas por eles. Também foi necessário utilizar o mesmo método para o aspeto do *Captcha*.

De modo a permitir a inclusão de imagens nos itens dos menus, também foi feito

<sup>106</sup> <https://jqueryui.com/>

<sup>107</sup> Quer fossem mensagens de status, avisos ou erros.



*override* às funções geradoras do menu primário e do menu secundário, sendo que, para que um determinado item possa estar num dos menus, terá de obedecer a um conjunto de requisitos tais como ter uma imagem<sup>108</sup> ou ter uma tag que o identifique como sendo um módulo Factsegur<sup>109</sup>.

Todo o restante aspeto foi implementado e modificado através do uso de CSS<sup>110</sup>.

## 6.6 Conclusão

Nesta capítulo foram explicadas de que maneira foram construídos os módulos e o tema do *Drupal*. Como nota final, é necessário referir ainda que foram tidos todos os cuidados para que os módulos fossem compatíveis com outros módulos do *Drupal*, podendo coexistir com eles.

Relativamente ao tema, não se passa o mesmo. Este foi construído a pensar no conjunto de módulos Factsegur e, caso sejam adicionados mais módulos a esta aplicação, terão de ter em conta a sua especificidade<sup>111</sup>.

Estando concluída a fase de implementação, pode-se finalmente passar à fase de testes e inquirir se os requisitos idealizados foram devidamente cumpridos.

---

108 Para o menu principal.

109 Para o menu secundário.

110 Era suposto que tivessem sido produzidos templates para todos os forms do site. No entanto, o estagiário deparou-se com um problema para o qual não conseguiu encontrar solução, embora todas as indicações em fóruns e na própria documentação do Drupal indicassem que a sua implementação estava correta. Quando era implementadas outras funções, de forma similar às que foram implementadas para os forms de login e requisição de password, o site ficava extremamente lento, ou chegava mesmo a dar timeout. Após ter investido uma grande quantidade de tempo a tentar resolver uma situação que funcionava perfeitamente para outros dois forms, foi decidido, por motivos temporais, descartar esta aproximação e modificar o aspeto dos restantes forms apenas por CSS.

111 Note-se que esta especificidade é propositada. Dados que os menus do Drupal não tinham capacidade para comportar imagens nos seus items, foi necessário implementar uma maneira que o permitisse.



# Capítulo 7

## Testes

Na fase de testes é um processo avalia-se o sistema e os seus componentes com o intuito de verificar se foram cumpridos ou não os requisitos definidos. Por outras palavras, testar é executar um conjunto de passos sobre um sistema de modo a identificar quaisquer falhas, erros ou requisitos em falta [87].

No caso do modelo de desenvolvimento em cascata, a fase de testes tem o seu início após a fase de implementação. No entanto, ao longo deste projeto, foram conduzidos testes unitários após a finalização de cada fase de modo a verificar que todas as funcionalidades estavam implementadas e que estavam de acordo com os requisitos definidos.

Nesta fase de testes foram ainda realizados testes funcionais, de modo a validar os resultados obtidos, e testes não funcionais, de modo a validar os requisitos não funcionais.

### 7.1 Testes funcionais

São um tipo de *black-box testing*, baseados nas especificações do software que irá ser testado. A aplicação é testada, sendo-lhe fornecidos os dados de entrada, e os resultados são examinados a para ver se estão em conformidade com a funcionalidade que era pretendida. Este tipo de testes são conduzidos num sistema completo e integrada, de forma a avaliar conformidade com os requisitos especificados<sup>112</sup> [88].

#### 7.1.1 Testes unitários

Este tipo de testes são efetuados pelos programadores antes de a equipa de testes executar formalmente os casos de teste. São realizados nas unidades individuais que foram desenvolvidas, sendo que os casos de teste utilizados são diferentes dos casos de teste da equipa de *quality assurance* [88].

---

<sup>112</sup> Com a exceção dos testes unitários conduzidos no final da implementação de cada um dos módulos,

No final de cada uma das fases de implementação de um módulo, foram idealizados um conjunto de testes de modo a verificar o seu correto funcionamento. Para estes testes foram definidos um conjunto de dados de entrada e estipulados os dados de saída pretendidos. Quando os dados obtidos não correspondiam ao desejado, eram efetuadas modificações ao código de modo a corrigir a situação.

## 7.1.2 Testes de integração

Testes de integração são um conjunto de testes efetuados nas partes combinadas de uma aplicação, de modo a determinar se apresentam um funcionamento correto. Podem ser efetuados através de duas aproximações diferente: testes de integração *bottom-up*<sup>113</sup> e *top-down*<sup>114</sup> [88].

No caso deste projeto optou-se por uma aproximação *bottom-up* onde, após a finalização da implementação e realização de testes unitários, era testada a integração entre os módulos efetuados. Tal como para os testes unitários, foram definidos um conjunto de parâmetros de entrada e definidas as respostas esperadas. Quando não se verificavam os comportamentos adequados, eram efetuadas modificações de modo a corrigir a situação.

## 7.1.2 Testes de sistema

Este tipo testes, testam o sistema como um todo. Assim que todos os componentes estão integradas, a aplicação como um todo é rigorosamente testada de modo a garantir que verifica os requisitos anteriormente especificados [88].

Estes testes são de extrema importância pois é o primeira vez que a aplicação é testada como um todo, sendo testada de modo a verificar que se encontra de acordo com as especificações funcionais e técnicas. É importante que a aplicação seja testada num ambiente o mais próximo possível com o ambiente de produção, de modo a verificar que estão reunidas todas as condições para que esta seja utilizada num ambiente real [88].

Para fazer esta validação foram compilados um conjunto de funcionalidades pretendidas para cada requisito e o resultado pretendido. Para tal, efetuou-se uma tabela com os campos descritos na tabela 10.

---

113 Este tipo de testes é iniciado com testes unitários, seguidos de testes cada vez mais alto-nível [88].

114 Neste tipo de testes, os elementos de mais alto-nível (por exemplo módulos) são testados em primeiro lugar e progressivamente são testados aspetos mais particulares de cada um destes

<b>ID teste</b>	Identificador unívoco do teste a ser realizado. Para efetuar correspondência com o caso de uso análogo, será apenas necessário substituir o prefixo TC no ID por UC <sup>115 116</sup> .
<b>Requisito</b>	Descrição do requisito de acordo com o caso de uso correspondente.
<b>Condições</b>	Estado do utilizador e do sistema aquando da realização das ações.
<b>Dados de entrada</b>	Input realizado pelo utilizador, dados ou ações.
<b>Resultado esperado</b>	Resultado esperado relativamente às ações realizadas pelo utilizador.
<b>E. (Resultado)</b>	Estado do resultado. Em caso de falha é marcado com uma “cruz” e adicionada uma nota de rodapé com informação relativa à falha. Em caso de sucesso, é marcado com um “visto”.

Tabela 10: Tabela para o registo de testes de sistema

A realização destes testes permitiu a validação do correto funcionamento do sistema. Quando foram encontradas falhas nas funcionalidades pretendidas, o teste foi marcado como falhado.

## 7.2 Testes não funcionais

Testes não funcionais envolvem o teste dos requisitos não funcionais de um determinado software, estando entre eles a performance, segurança, interface do utilizador, etc [88].

Foram escolhidos para esta fase de testes, aqueles que apresentavam maior relevância para o que será o cenário utilização real do sistema.

Antes de mais é necessário saber qual a resposta do sistema às condições definidas para a sua utilização, num escritório de pequena ou média dimensão de agentes de regulação de sinistros.

Além disso, dado que no escritório em questão poderão utilizar um ou mais browser e sendo que o browser utilizado em cada em cada escritório pode variar, é necessário efetuar um estudo da variação do aspeto do interface gráfico nos

<sup>115</sup> Por exemplo, o TC-001 corresponde ao UC-001.

diferentes softwares onde é apresentado <sup>117</sup>.

Finalmente, dado que este software lida com informação confidencial, foi testada a sua segurança para um conjunto de ataques conhecidos.

## 7.2.1 Testes de performance

Este tipo de testes é usado mais para identificar estrangulamentos ou problemas de performance do que para encontrar bugs no software. Sendo que existem muitas causas que podem causar impacto na performance de um software, este é um teste obrigatório. O teste de performance encontra-se dividido em várias sub-tipos, sendo um deles o **teste de carga**.

Antes de efetuar o teste de carga foi necessário efetuar uma análise para verificar qual seria a ferramenta mais adequada para esta tarefa. Assim sendo, foi realizada uma pesquisa e optou-se pela escolha do Apache JMeter[89] para a realização dos testes<sup>118</sup>. Além de permitir a definição das páginas que se pretende testar, este software permite ainda verificar qual a utilização de memória e cpu no lado do servidor, dados que podem ser importantes para futuras otimizações e comparação com resultados anteriores.

Foram considerados 4 cenários, cada um com 1, 2, 5 e 10 utilizadores. Para cada um destes cenários, foram executados 5 testes, variando o número de pedidos que cada utilizador faz em 2, 4, 6, 8 e 10. A imagem 51 apresenta um conjunto de resultados.

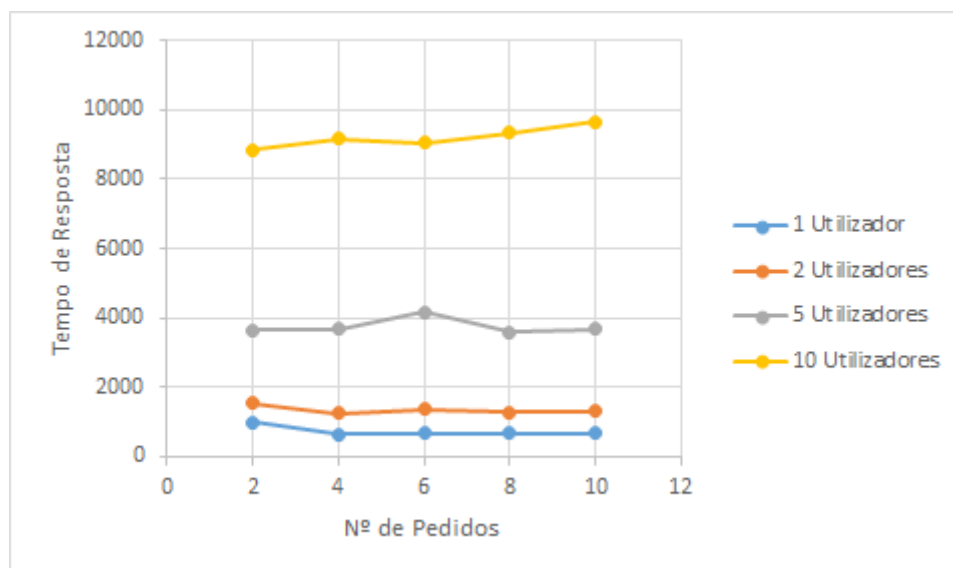


Imagem 51: Teste de carga à adição de um processo

<sup>116</sup> As condições de falha são identificadas através do sufixo “\_FA<n>”, de acordo com o representado anteriormente nos requisitos.

<sup>117</sup> Foram escolhidos para efetuar o teste os browsers indicados pelo W3Schools como os mais utilizados.

<sup>118</sup> Embora houvesse outras opções mais satisfatórias, tal como o Blazemeter, com integração no Drupal, devido a constrangimentos financeiros optou-se por esta solução gratuita.

Conforme se pode ver, de forma genérica, conforme vão aumentando o número de utilizadores a fazer pedidos, o tempo médio de resposta da página também vai aumentar. Pode-se verificar essa tendência também na imagem 52.

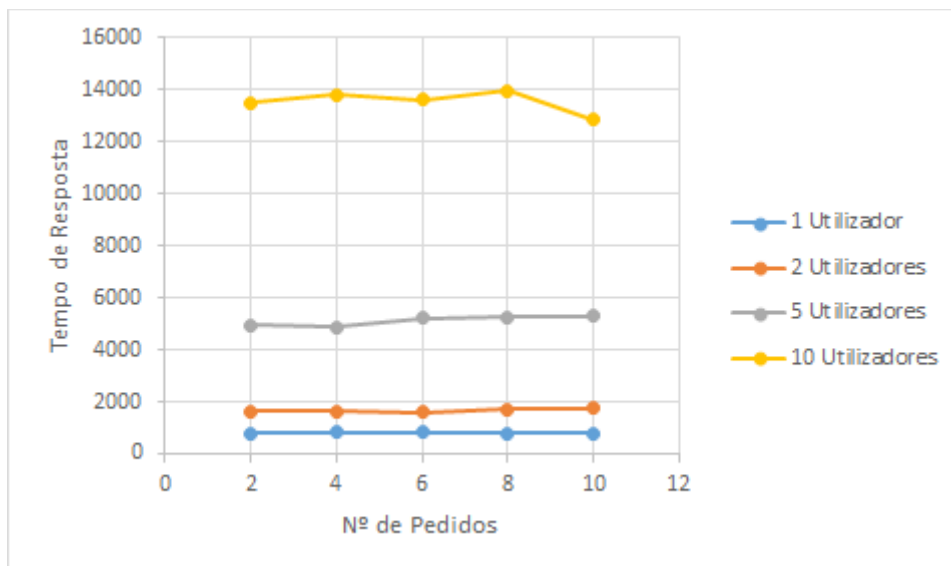


Imagem 52: Teste de carga para a visualização de listagens de processos.

Como se pode ver, com 10 utilizadores, o tempo médio de resposta, para 8 pedidos por utilizador, chegou aos 14 segundos. Na tabela 53, mais uma vez confirma-se o comportamento anterior.

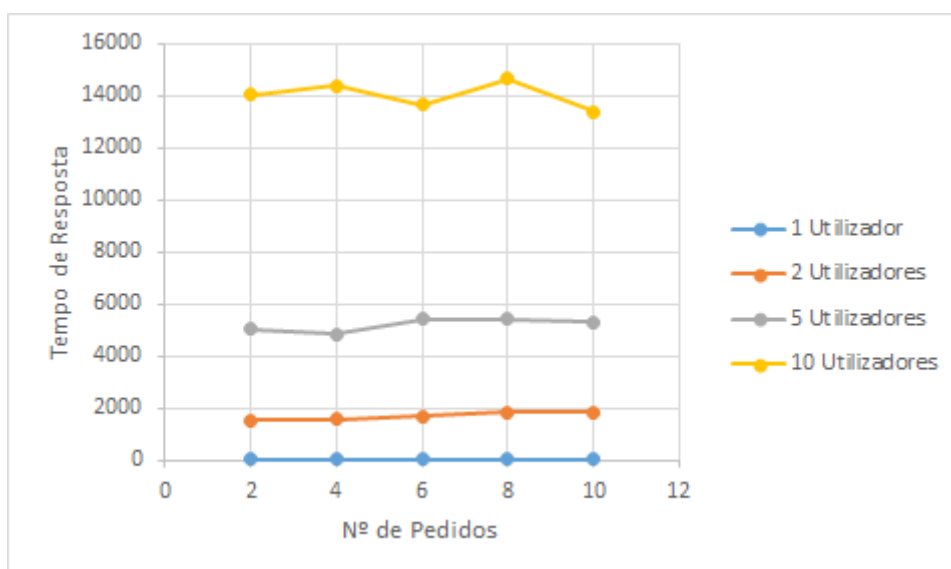


Imagem 53: Teste de carga para a visualização de um determinado processo.

Os resultados detalhados destes testes de carga são apresentados no anexo 9.

É necessário visualizar estes resultados com sentido crítico. Embora sejam francamente maus, o cenário de teste inclui uma máquina virtual com 512 de RAM a usar um processador de um Intel Core Duo T2250<sup>119</sup>, ou seja, um servidor com muito pouca capacidade de resposta para um ambiente um pouco mais exigente.

No entanto, estes testes servem de base para se poder testar um conjunto de otimizações que se podem efetuar no Drupal e no Apache. Além disso, tendo já o teste preparado, será mais simples a experimentação de outras máquinas de modo a acessar as necessidades do cliente em termos de hardware, caso o cenário se aplique, ou em termos de necessidade de capacidade de processamento na *cloud*.

## 7.2.2 Testes de segurança

Neste processo estão envolvidos testes efetuados ao software de forma a identificar quaisquer falhas ou lacunas sob o ponto de vista de segurança e da vulnerabilidade do sistema.

Dado que Drupal possui mecanismos de forma a prevenir ataques de SQL-injection, e esse cuidado foi tido aquando da implementação, não haveria motivos para preocupações acerca deste aspeto.

No entanto, de modo a obter alguma confirmação que todas as medidas tinham sido tomadas e que os mecanismos do Drupal estavam a funcionar, foram realizados um conjunto de testes sobre campos dos *forms*, não tendo sido detectado qualquer problema.

## 7.2.3 Conclusão

Um processo de testes termina apenas por constrangimentos temporais e/ou financeiros e, mesmo após a finalização desta etapa, não se pode dizer que a aplicação esteja livre de bugs ou que não é necessário realizar mais testes sobre o sistema.

Sempre que for necessário, é essencial executar um conjunto de testes pré-definidos sobre o sistema de modo a verificar se novas funcionalidades não introduziram bugs ou se algum requisito deixou de ser cumprido

---

119 <http://www.notebookcheck.net/Intel-Core-Duo-Notebook-Processor.12513.0.html>



# Capítulo 8

## Conclusão

Este projeto foi efetuado com o objetivo de se construir um software para a área de regulação de sinistros, sendo que este estágio em particular estava focado no aspeto de gestão de processos.

Numa fase inicial foi efetuado o estudo do estado da arte relativamente a esta área e quais os softwares que poderiam ser úteis nesta área e fazer assim o levantamento do conjunto de funções que poderiam ser relevantes neste software.

Em seguida foi definida uma metodologia para o desenvolvimento deste projeto e feito o levantamento dos requisitos. A partir daqui foi escolhida a *framework* a utilizar e, após realizar o estudo da sua arquitetura, foi iniciada a sua implementação.

Neste espaço de tempo foram construídos módulos que permitiam a gestão de utilizadores internos, efetuar pesquisa sobre os dados contidos na base de dados, enviar mensagens, gerir tarefas, visualização de logs e, principalmente, gerir processos de peritos de seguros.

Numa última fase, foram efetuados testes ao sistema de modo a verificar o correto funcionamento de acordo com o que havia sido definido nos requisitos e na arquitetura.

Embora tenham sido encontrados problemas pelo caminho, estes foram sendo resolvidos e, na opinião do estagiário, foram cumpridos os objetivos estipulados. Este foi um enorme processo de aprendizagem onde o aluno sente que tem ainda um longo caminho pela frente, mas que foram lançadas as bases para que o conhecimento se consolide de forma duradoura.

Embora não tenha sido efetuada a aplicação para dispositivos móveis para acesso ao sistema, torna-se agora evidente que tal seria impossível no espaço temporal estipulado.

### 8.1 Trabalho futuro

Embora muito tenha sido feito, muito fica ainda por fazer.

Terá de se efetuar ainda a validação do software com o cliente, de modo que se ouça a sua opinião e se façam ajustes ao sistema caso necessário.

Outro aspeto importante são os ficheiros de ajuda que, embora implementados no Drupal, tenham de ser ainda criados.

Um outro aspeto relevante é a troca de mensagens com suporte de email. Não foi possível utilizar o módulo que se tinha idealizado devido aos problemas encontrados, mas no futuro esta função terá de ser implementada.

O aspeto da otimização do Drupal também é um aspeto de relevo. Embora os testes tenham mostrado uma resposta lenta, isso é facilmente explicável devido ao uso de uma máquina virtual de capacidades reduzida. No entanto, existe ainda um conjunto de mecanismos que podem melhorar a resposta do Drupal, e que serão devidamente estudadas tais como Memcache<sup>120</sup> e APC<sup>121 122</sup>.

---

120 <http://php.net/manual/en/book.memcache.php>

121 <http://php.net/manual/en/book.apc.php>

122 Além de um conjunto de outros aspetos que se podem melhorar: <https://www.drupal.org/node/1722250> .

## Referências

- [1] RESI - Regulação de Sinistros, Lda [Online] [Visualizado em: 29 de Julho de 2015] Disponível em: <http://www.resi.pt/>
- [2] Jopema Reguladora de Sinistros [Online] [Visualizado em: 2 de Março de 2015] Disponível em: <http://www.jopema.com.br/v2/sobre.php?op=1&q=RegulacaoSinistros>
- [3] Portal das Finanças (Portugal) [Online] [Visualizado em: 3 de Março de 2015] Disponível em: [http://info.portaldasfinancas.gov.pt/pt/apoio\\_contribuinte/CertificacaoSoftware.htm](http://info.portaldasfinancas.gov.pt/pt/apoio_contribuinte/CertificacaoSoftware.htm)
- [4] Portal das Finanças (Portugal) [Online] [Visualizado em: 3 de Março de 2015] Disponível em: <http://www.portaldasfinancas.gov.pt/pt/Out/consultaProgCertificadosM24.action>
- [5] Capterra, the smart way to find business software [Online] [Visualizado em: 3 de Março de 2015] Disponível em: <http://www.capterra.com/claims-processing-software/>
- [6] Capterra, the smart way to find business software [Online] [Visualizado em: 3 de Março de 2015] Disponível em: <http://www.capterra.com/sem/document-management-software#infographic>
- [7] Garvin Allen - Advanced Insurance Systems [Online] [Visualizado em: 10 de Março de 2015] Disponível em: <http://www.garvin-allen.com/ais/index.html>
- [8] ALYCE Claims Management System [Online] [Visualizado em: 10 de Março de 2015] Disponível em: <http://www.brightworkinc.com/alyce.html>
- [9] Claimloop.com, Complete Claims Management Platform for Restoration Companies, Insurance Adjusters and Property Owners [Online] [Visualizado em: 10 de Março de 2015] Disponível em: <http://www.claimloop.com/>
- [10] Claim Ruler - Industrial Strength P&C Claims Management Software [Online] [Visualizado em: 12 de Março de 2015] Disponível em: <https://www.claimruler.com>
- [11] ClickClaims, Claims Management Software [Online] [Visualizado em: 10 de Março de 2015] Disponível em: <http://www.clickclaims.com/product.html>
- [12] Cloudworks, Software à medida, Software na Cloud [Online] [Visualizado em: 11 de Março de 2015] Disponível em: <http://cloudworks.pt/>
- [13] Cloudworks: software e consultadoria Lda. | Linkedin [Online] [Visualizado em: 11 de Março de 2015] Disponível em: <https://www.linkedin.com/company/cloudworks-pt>
- [14] Claims Management System, Claims Adjusting Software for independent adjuster handling P&C, Auto Appraisal, CAT and TPA claims. [Online] [Visualizado em: 11 de Março de 2015] Disponível em: <http://www.onlinereportinginc.com/>
- [15] [www.mrinformatica.com](http://www.mrinformatica.com) [Online] [Visualizado em: 13 de Março de 2015] Disponível em: <http://www.mrinformatica.com/>
- [16] Periagro, Peritagens agrícolas e avaliações fundiárias S.A. [Online] [Visualizado em: 12 de Março de 2015] Disponível em: <http://www.periagro.pt/home.html>
- [17] Periava , Lda. [Online] [Visualizado em: 12 de Março de 2015] Disponível em: <http://periava.pt/>
- [18] SGLSistemas [Online] [Visualizado em: 13 de Março de 2014] Disponível em: <http://www.sglsistemas.com.br/sistemas/>

- [19] Virtual Claims Adjuster - Virtual Claims Adjuster - Claims Management Software [Online] [Visualizado em: 13 de Março de 2015] Disponível em: <http://www.virtualclaimsadjuster.com/>
- [20] Alfresco (software) - Wikipedia, the free encyclopedia [Online] [Visualizado em: 13 de Março de 2015] Disponível em: [http://en.wikipedia.org/wiki/Alfresco\\_%28software%29](http://en.wikipedia.org/wiki/Alfresco_%28software%29)
- [21] Open Source Licensing - alfrescowiki [Online] [Visualizado em: 13 de Março de 2015] Disponível em: [https://wiki.alfresco.com/wiki/Open\\_Source\\_Licensing](https://wiki.alfresco.com/wiki/Open_Source_Licensing)
- [22] Modern Enterprise Content Management with Alfresco One ECM | Alfresco [Online] [Visualizado em: 13 de Março de 2015] Disponível em: <http://www.alfresco.com/products/one>
- [23] Creating and managing workflows | Alfresco Documentation [Online] [Visualizado em: 13 de Março de 2015] Disponível em: <http://docs.alfresco.com/4.2/topics/wf-howto.html>
- [24] Installing and configuring Google Docs integration | Alfresco Documentation [Online] [Visualizado em: 13 de Março de 2015] Disponível em: <http://docs.alfresco.com/5.0/concepts/googledocs-intro.html>
- [25] Portugal Portuguese Language Pack | Alfresco Add-ons - Alfresco Customizations [Online] [Visualizado em: 13 de Março de 2014] Disponível em: <https://addons.alfresco.com/addons/portugal-portuguese-language-pack>
- [26] Asite: Overview | LinkedIn [Online] [Visualizado em: 13 de Março de 2014] Disponível em: <https://www.linkedin.com/company/asite>
- [27] Document Management | Corporate Collaboration in the cloud - asite.com [Online] [Visualizado em: 13 de Março de 2015] Disponível em: <http://www.asite.com/adoddle/corporate-collaboration/document-management>
- [28] Contentverse: Document Management Software | Digital Filing Software [Online] [Visualizado em: 13 de Março de 2015] Disponível em: <http://www.contentverse.com/>
- [29] Cloud Document Management Software - docSTAR eclipse [Online] [Visualizado em: 13 de Março de 2015] Disponível em: <http://www.docstar.com/document-management-systems/cloud>
- [30] Electronic Document and File Management Software [Online] [Visualizado em: 13 de Março de 2015] Disponível em: <http://www.efilecabinet.com/>
- [31] filedepot | Drupal.org [Online] [Visualizado em: 13 de Março de 2015] Disponível em: <https://www.drupal.org/project/filedepot>
- [32] Drupal - Wikipedia, the free encyclopedia [Online] [Visualizado em: 13 de Março de 2015] Disponível em: <http://en.wikipedia.org/wiki/Drupal>
- [33] FileHold | Document Management Software, Best Records Management Software [Online] [Visualizado em: 13 de Março de 2015] Disponível em: <http://www.filehold.com/>
- [34] LogicalDOC Document Management System | Open Source Software [Online] [Visualizado em: 13 de Março de 2015] Disponível em: <http://logicaldoc.com/>
- [35] Secure Business File Sharing & FTP Hosting | SmartFile [Online] [Visualizado em: 13 de Março de 2015] Disponível em: <https://www.smartfile.com/>
- [36] Samba - opening windows to a wider world [Online] [Visualizado em: 28 de Janeiro de 2016] Disponível em: <https://www.samba.org/>

- [37] Waterfall model - Wikipedia, the free encyclopedia [Online] [Visualizado em: 31 de Março de 2015] Disponível em: [http://en.wikipedia.org/wiki/Waterfall\\_model](http://en.wikipedia.org/wiki/Waterfall_model)
- [38] Requirements analysis - Wikipedia, the free encyclopedia [Online] [Visualizado em: 31 de Março de 2015] Disponível em: [http://en.wikipedia.org/wiki/Requirements\\_analysis](http://en.wikipedia.org/wiki/Requirements_analysis)
- [39] Software architecture - Wikipedia, the free encyclopedia [Online] [Visualizado em: 31 de Março de 2015] Disponível em: [http://en.wikipedia.org/wiki/Software\\_architecture](http://en.wikipedia.org/wiki/Software_architecture)
- [40] Requirements Analysis Document Template [Online] [Visualizado em: 14 de Abril de 2015] Disponível em: <http://www.cs.fsu.edu/~lacher/courses/COP3331/rad.html>
- [41] MoSCoW method - Wikipedia, the free encyclopedia [Online] [Visualizado em: 15 de Junho de 2015] Disponível em: [https://en.wikipedia.org/wiki/MoSCoW\\_method](https://en.wikipedia.org/wiki/MoSCoW_method)
- [42] Web application - Wikipedia, the free encyclopedia [Online] [Visualizado em: 8 de Outubro de 2015] Disponível em: [https://en.wikipedia.org/wiki/Web\\_application](https://en.wikipedia.org/wiki/Web_application)
- [43] CloudFlare - Wikipedia, the free encyclopedia [Online] [Visualizado em: 5 de Julho de 2015] Disponível em: <https://en.wikipedia.org/wiki/CloudFlare>
- [44] Affordable advanced DDoS protection and mitigation | CloudFlare | The web performance & security company [Online] [Visualizado em: 5 de Julho de 2015] Disponível em: <https://www.cloudflare.com/ddos>
- [45] CloudFlare Review 2015 - TopTenREVIEWS [Online] [Visualizado em: 5 de Julho de 2015] Disponível em: <http://website-security-and-performance-review.toptenreviews.com/cloudflare-review.html>
- [46] Nagios - Wikipedia, the free encyclopedia [Online] [Visualizado em: 5 de Julho de 2015] Disponível em: <https://en.wikipedia.org/wiki/Nagios>
- [47] Application Monitoring - Nagios [Online] [Visualizado em: 5 de Julho de 2015] Disponível em: <https://www.nagios.com/solutions/application-monitoring>
- [48] PostgreSQL - Wikipedia, the free encyclopedia [Online] [Visualizado em: 5 de Julho de 2015] Disponível em: <https://en.wikipedia.org/wiki/PostgreSQL>
- [49] NoSQL for the Enterprise | EnterpriseDB [Online] [Visualizado em: 5 de Julho de 2015] Disponível em: <http://www.enterprisedb.com/nosql-for-enterprise>
- [50] Why I Choose PostgreSQL Over MySQL/MariaDB - Slashdot [Online] [Visualizado em: 5 de Julho de 2015] Disponível em: [http://posulliv.github.io/images/second\\_query\\_response\\_time.png](http://posulliv.github.io/images/second_query_response_time.png)
- [51] Apache HTTP Server - Wikipedia, the free encyclopedia [Online] [Visualizado em: 5 de Julho de 2015] Disponível em: [https://en.wikipedia.org/wiki/Apache\\_HTTP\\_Server](https://en.wikipedia.org/wiki/Apache_HTTP_Server)
- [52] The Drupal overview | Drupal.org [Online] [Visualizado em: 5 de Julho de 2015] Disponível em: <https://www.drupal.org/getting-started/before/overview>
- [53] Dries Buytaert - Wikipedia, the free encyclopedia [Online] [Visualizado em: 5 de Julho de 2014] Disponível em: [https://en.wikipedia.org/wiki/Dries\\_Buytaert](https://en.wikipedia.org/wiki/Dries_Buytaert)
- [54] Drupal - Wikipedia, the free encyclopedia [Online] [Visualizado em: 5 de Julho de 2015] Disponível em: <https://en.wikipedia.org/wiki/Drupal>
- [55] CMS comparison - WordPress vs Joomla vs Drupal / WebsiteSetup.org [Online] [Visualizado em: 5 de Julho de 2015] Disponível em: <http://websitesetup.org/cms-comparison-wordpress-vs-joomla-drupal/>

- [56] Explaining architectural tiers in Drupal - robknight.org.uk [Online] [Visualizado em: 5 de Julho de 2015] Disponível em: <http://robknight.org.uk/blog/2011/02/explaining-architectural-tiers-drupal/>
- [57] MVC vs. PAC | GarfieldTech [Online] [Visualizado em: 5 de Julho de 2015] Disponível em: <http://www.garfieldtech.com/blog/mvc-vs-pac>
- [58] How Drupal Works: An Architect's Overview | DrupalCon San Francisco 2010 [Online] [Visualizado em: 11 de Outubro de 2015] Disponível em: <http://sf2010.drupal.org/conference/sessions/how-drupal-works-architects-overview.html>
- [59] Cross-site scripting - Wikipedia, the free encyclopedia [Online] [Visualizado em: 12 de Outubro de 2015] Disponível em: [https://en.wikipedia.org/wiki/Cross-site\\_scripting](https://en.wikipedia.org/wiki/Cross-site_scripting)
- [60] Batch processing - Wikipedia, the free encyclopedia [Online] [Visualizado em: 12 de Outubro de 2015] Disponível em: [https://en.wikipedia.org/wiki/Batch\\_processing](https://en.wikipedia.org/wiki/Batch_processing)
- [61] Headless Drupal. Why & how a RESTful API in Drupal? | Blog OpenLucius [Online] [Visualizado em: 19 de Outubro de 2015] Disponível em: <http://blog.openlucius.com/en/blog/headless-drupal-why-how-restful-api-drupal>
- [62] Deployment environment - Wikipedia, the free encyclopedia [Online] [Visualizado em: 18 de Janeiro de 2016] Disponível em: [https://en.wikipedia.org/wiki/Deployment\\_environment](https://en.wikipedia.org/wiki/Deployment_environment)
- [63] Conceitos: Ambiente de Desenvolvimento [Online] [Visualizado em: 18 de Janeiro de 2016] Disponível em: [http://www.wthreex.com/rup/process/workflow/environm/co\\_deven.htm](http://www.wthreex.com/rup/process/workflow/environm/co_deven.htm)
- [64] <http://www.sitepoint.com/best-php-ide-2014-survey-results/> [Online] [Visualizado em: 12 de Janeiro de 2016] Disponível em: <http://www.sitepoint.com/best-php-ide-2014-survey-results/>
- [65] Why cant someone just make a good IDE for PHP? — explosive web programming — MODERN CODE TACTICS [Online] [Visualizado em: 12 de Janeiro de 2016] Disponível em: <http://www.jbplotus.com/2012/06/29/why-cant-someone-just-make-a-good-ide-for-php/>
- [66] Getting comfy with PhpStorm - one of the best IDEs so far! - The Storyteller [Online] [Visualizado em: 12 de Janeiro de 2015] Disponível em: <http://hasin.me/2012/01/03/getting-comfy-with-phpstorm/>
- [67] How To Install Drupal 7 on Ubuntu Server 12.04 - Ubuntu Server Guide [Online] [Visualizado em: 13 de Janeiro de 2016] Disponível em: <http://ubuntuserverguide.com/2012/08/how-to-install-drupal-7-on-ubuntu-server-12-04.html>
- [68] hook\_schema | system.api.php | Drupal 7 | Drupal API [Online] [Visualizado em: 12 de Janeiro de 2016] Disponível em: [https://api.drupal.org/api/drupal/modules%21system%21system.api.php/function/hook\\_schema/7](https://api.drupal.org/api/drupal/modules%21system%21system.api.php/function/hook_schema/7)
- [69] Writing module .info files (Drupal 7.x) | Drupal.org [Online] [Visualizado em: 12 de Janeiro de 2016] Disponível em: <https://www.drupal.org/node/542202>
- [70] form\_set\_error | form.inc | Drupal 7 | Drupal API [Online] [Visualizado em: 19 de Janeiro de 2016] Disponível em: [https://api.drupal.org/api/drupal/includes%21form.inc/function/form\\_set\\_error/7](https://api.drupal.org/api/drupal/includes%21form.inc/function/form_set_error/7)
- [71] How does Drupal 7 Work? Part 4: Modules and Menus | a denizen's

- entertainment [Online] [Visualizado em: 18 de Janeiro de 2016] Disponível em: <http://deninet.com/blog/1021/how-does-drupal-7-work-part-4-modules-and-menus>
- [72] hook\_menu | system.api.php | Drupal 7 | Drupal API [Online] [Visualizado em: 18 de Janeiro de 2016] Disponível em: [https://api.drupal.org/api/drupal/modules%21system%21system.api.php/function/hook\\_menu/7](https://api.drupal.org/api/drupal/modules%21system%21system.api.php/function/hook_menu/7)
- [73] Wildcard usage and core's wildcards | Drupal.org [Online] [Visualizado em: 18 de Janeiro de 2016] Disponível em: <https://www.drupal.org/node/209056>
- [74] About Views handlers | views.api.php | Views 7 | Drupal API [Online] [Visualizado em: 18 de Janeiro de 2016] Disponível em: [https://api.drupal.org/api/views/views.api.php/group/views\\_handlers/7](https://api.drupal.org/api/views/views.api.php/group/views_handlers/7)
- [75] hook\_form\_alter | system.api.php | Drupal 7 | Drupal API [Online] [Visualizado em: 13 de Janeiro de 2016] Disponível em: [https://api.drupal.org/api/drupal/modules%21system%21system.api.php/function/hook\\_form\\_alter/7](https://api.drupal.org/api/drupal/modules%21system%21system.api.php/function/hook_form_alter/7)
- [76] watchdog | bootstrap.inc | Drupal 7 | Drupal API [Online] [Visualizado em: 13 de Janeiro de 2016] Disponível em: <https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/watchdog/7>
- [77] PHPOffice/PHPWord: A pure PHP library for r... - GitHub [Online] [Visualizado em: 17 de Janeiro de 2016] Disponível em: <https://github.com/PHPOffice/PHPWord>
- [78] CAPTCHA - Wikipedia, the free encyclopedia [Online] [Visualizado em: 17 de Janeiro de 2016] Disponível em: <https://en.wikipedia.org/wiki/CAPTCHA>
- [79] CAPTCHA | Drupal.org [Online] [Visualizado em: 17 de Janeiro de 2016] Disponível em: <https://www.drupal.org/project/captcha>
- [80] Field Group | Drupal.org [Online] [Visualizado em: 18 de Janeiro de 2016] Disponível em: [https://www.drupal.org/project/field\\_group](https://www.drupal.org/project/field_group)
- [81] Date | Drupal.org [Online] [Visualizado em: 18 de Janeiro de 2015] Disponível em: <https://www.drupal.org/project/date>
- [82] Date Popup Timepicker | Drupal.org [Online] [Visualizado em: 18 de Janeiro de 2015] Disponível em: [https://www.drupal.org/project/date\\_popup\\_timepicker](https://www.drupal.org/project/date_popup_timepicker)
- 83: Views | Drupal.org, , 18 de Janeiro de 2016
- [84] Views Dynamic Fields | Drupal.org [Online] [Visualizado em: 18 de Janeiro de 2016] Disponível em: [https://www.drupal.org/project/views\\_dynamic\\_fields](https://www.drupal.org/project/views_dynamic_fields)
- [85] Views PHP | Drupal.org [Online] [Visualizado em: 18 de Janeiro de 2016] Disponível em: [https://www.drupal.org/project/views\\_php](https://www.drupal.org/project/views_php)
- [86] template.php | Drupal.org [Online] [Visualizado em: 19 de Janeiro de 2016] Disponível em: <https://www.drupal.org/node/1728096>
- [87] 37:43 min - Estágio - InfoInov. InfoInov • Summary Report • Toggl [Online] [Visualizado em: 25 de Janeiro de 2016] Disponível em: [http://www.tutorialspoint.com/software\\_testing/software\\_testing\\_overview.htm](http://www.tutorialspoint.com/software_testing/software_testing_overview.htm)
- [88] Software Testing - Levels [Online] [Visualizado em: 26 de Janeiro de 2016] Disponível em: [http://www.tutorialspoint.com/software\\_testing/software\\_testing\\_levels.htm](http://www.tutorialspoint.com/software_testing/software_testing_levels.htm)
- [89] <http://jmeter.apache.org/> [Online] [Visualizado em: 26 de Janeiro de 2016] Disponível em: <http://jmeter.apache.org/>