

Mestrado em Engenharia Informática
Relatório Final de Estágio



**Departamento de Engenharia Informática, Faculdade de
Ciências e Tecnologia da Universidade de Coimbra**

**Estágio FactSegur – Criação de um Sistema de
Informação específico para a área de Peritagens de
Seguros**

Aluno B, sistema geral e módulo de faturação

4 de Fevereiro de 2016

Estágio Plurianual 2014-2015

**Relatório de estágio submetido para o grau de Mestre em Engenharia
Informática**

Aluno: Marco André C. R. Pedrosa, nº 2006126653, mpedrosa@student.dei.uc.pt

Estágio FactSegur – Criação de um Sistema de Informação específico para a área de Peritagens de Seguros

Aluno B, sistema geral e módulo de faturação

4 de Fevereiro de 2016

Estágio Plurianual 2014-2015

**Relatório de estágio submetido para o grau de Mestre em Engenharia
Informática**

Aluno: Marco André C. R. Pedrosa, nº 2006126653, mpedrosa@student.dei.uc.pt

Orientador (DEI): Professor Doutor Paulo Rupino da Cunha

Júri Arguente: Professor Doutor Álvaro Rocha

Júri Vogal: Professor Doutor Tiago Batista

Resumo

FactSegur é um projeto desenvolvido por dois estagiários a terminar o curso de Mestrado em Engenharia Informática, que irá resultar num sistema de informação que permita a gestão dos processos de uma empresa da área de Peritagens/Regulação de Sinistros, bem como também resolver as suas necessidades ao nível da faturação, tendo assim de cumprir os requisitos legais Portugueses para tal.

O Aluno B, correspondente a este relatório irá estar encarregue do módulo de faturação. O outro estagiário estará encarregue do módulo de gestão de processos. Ambos os estagiários irão estar encarregues do sistema geral, que envolve o motor da aplicação e sistema de gestão de utilizadores.

Ambos os estagiários irão passar pelas mesmas fases de desenvolvimento, trabalhando cada um em áreas diferentes do projeto, e os seus relatórios irão conter o trabalho relativo à sua própria parte do projeto.

No primeiro semestre foi feita a análise do estado da arte, bem como a especificação das funcionalidades necessárias na fase de análise de requisitos. Foi também iniciada a fase de especificação da arquitetura, mas esta não chegou a ser concluída, continuando o seu progresso no segundo semestre.

No segundo semestre foi completada a arquitetura, passando de seguida à implementação, e por final decorreu a fase de testes, onde foi validado que o que foi implementado corresponde ao que foi definido na análise de requisitos.

Palavras-Chave

Certificado, FactSegur, Faturação, Gestão, Online, Peritagens, Peritos de Seguros, PME, Regulação de Sinistros, "Sistema de Informação"

Índice

Índice de Tabelas.....	11
Índice de Figuras.....	13
Acrónimos.....	15
Glossário.....	17
1.Introdução.....	19
1.1.Entidade Auxiliar.....	19
1.2.Objetivos.....	20
1.3.Planeamento.....	21
1.3.1.Reuniões de Estágio.....	21
1.3.2.Planeamento.....	21
1.3.3.Desvios ao Planeamento.....	25
1.4.Estrutura do Documento.....	26
2.Estado da Arte.....	27
2.1.Introdução à Área de Peritagens de Seguros.....	28
2.2.Requisitos de Certificação de Software de Faturação em Portugal.....	29
2.2.1.Isenção da necessidade de Certificação.....	29
2.2.2.Assinatura Digital de Documentos Comerciais com chave RSA.....	30
2.2.3.Saft-T (PT).....	30
2.2.4.Outros requisitos para o Programa.....	33
2.3.Soluções já existentes de faturação.....	35
2.4.Alternativas a uma implementação de raiz (GnuCash).....	40
2.5.Critérios de Comparação.....	41
2.6.Análise Comparativa.....	43
2.7.Conclusão.....	44
3.Metodologia de Desenvolvimento.....	45
3.1.Método em Cascata.....	45
3.2.Método SCRUM (desenvolvimento ágil).....	46
3.3.Metodologia escolhida.....	46
4.Análise de Requisitos.....	47
4.1.Levantamento de Requisitos.....	48
4.2.Núcleo Principal.....	49
4.3.Casos de Uso.....	50
4.4.Navegação no programa.....	62
4.5.Módulo de Contactos.....	63
4.6.Módulo de Faturação.....	64
4.7.Atributos de qualidade.....	66
4.8.Conclusão.....	67
5.Especificação da Arquitetura.....	69
5.1.Estrutura geral do sistema.....	70
5.1.1.CloudFlare.....	71
5.1.2.Nagios.....	72
5.1.3.PostgreSQL.....	72
5.1.4.Apache HTTP Server.....	72
5.2.Garantia dos atributos de qualidade.....	72
5.3.Drupal.....	74

5.3.1. Padrão Arquitetural do Drupal.....	74
5.3.2. Perspetiva de Fluxo do Drupal.....	76
5.3.3. Sistema de Renderização de Dados.....	77
5.3.4. Sistema de Hooks do Drupal.....	78
5.4. Vista de Módulos da aplicação FactSegur.....	79
5.5. Descrição dos Módulos.....	80
5.5.1. Módulo de Gestão de Documentos Comerciais.....	81
5.5.2. Módulo de Gestão de Produtos.....	81
5.5.3. Módulo de Gestão de Impostos.....	82
5.5.4. Módulo de Criação de Ficheiros SAF-T(PT).....	83
5.5.5. Módulo de Mapas de Faturação.....	83
5.5.6. Módulo de Contactos (Sistema geral).....	84
5.5.7. Módulo de Gestão de Utilizadores (Sistema geral).....	85
5.5.8. Módulo de Definições (Sistema geral).....	86
5.5.9. Módulo de Pesquisa (Sistema geral).....	86
5.5.10. Módulo de Logs (Sistema geral).....	87
5.5.11. Tema do Drupal (Sistema geral).....	87
5.6. Modelo de Dados.....	87
5.6.1. Tabela “users”.....	89
5.6.2. Tabela “users_internal”.....	89
5.6.3. Tabela “users_external”.....	90
5.6.4. Tabela “contacts_enterprise”.....	90
5.6.5. Tabela “contacts_single”.....	91
5.6.6. Tabela “product”.....	91
5.6.7. Tabela “sourcedocs”.....	91
5.6.8. Tabela “linessourcedocs”.....	93
5.6.9. Tabela “taxtable”.....	93
5.6.10. Tabela “taxtype”.....	94
5.6.11. Tabela “taxcode”.....	94
5.6.12. Tabela “taxcountryregion”.....	94
5.6.13. Tabela “sourcedocdefaults”.....	94
5.6.14. Tabela “sourcedoctype”.....	95
5.6.15. Tabela “taxexemptionreason”.....	95
5.6.16. Tabela “paymentmechanism”.....	95
5.7. Conclusão.....	96
6. Implementação.....	97
6.1. Ambiente de desenvolvimento.....	97
6.2. Módulos do Drupal.....	100
6.2.1. Ficheiro de Dependências do Módulo.....	100
6.2.2. Ficheiro de Instalação do módulo.....	101
6.2.3. Ficheiro de definição do módulo.....	101
6.2.4. Form API.....	103
6.2.5. Validação de Forms.....	105
6.2.6. Submissão de Forms.....	105
6.2.7. Módulo Views.....	105
6.3. Módulo de Utilizadores Externos.....	106
6.4. Módulo de Contactos.....	109
6.5. Módulo de Permissões.....	111
6.6. Módulo de Faturação.....	114
6.7. Módulo de Produtos.....	122

6.8.Módulo de Impostos.....	124
6.9.Módulo de Definições.....	125
6.10.Conclusão.....	126
7.Testes.....	127
7.1. Testes Funcionais.....	127
7.2. Testes Não Funcionais.....	128
7.3. Conclusão.....	134
8. Conclusão.....	135
8.1. Trabalho Futuro.....	135
9.Referências Bibliográficas.....	137
Anexos.....	141

Índice de Tabelas

Tabela 1: Requisitos de Certificação – Programa deve Possuir.....	33
Tabela 2: Requisitos de Certificação – Programa deve Assegurar.....	34
Tabela 3: Requisitos de Certificação – Programa não pode Permitir.....	35
Tabela 4: Análise Comparativa de várias soluções de faturação.....	43
Tabela 5: Lista total de Casos de Uso e sua priorização.....	51
Tabela 6: Atributos de Qualidade e sua priorização.....	66
Tabela 7: Módulo de Gestão de Documentos Comerciais.....	81
Tabela 8: Módulo de Gestão de Produtos.....	82
Tabela 9: Módulo de Gestão de Impostos.....	82
Tabela 10: Módulo de Criação de ficheiros SAF-T(PT).....	83
Tabela 11: Módulo de Mapas de Faturação.....	83
Tabela 12: Módulo de Gestão de Contactos (Sistema geral).....	85
Tabela 13: Módulo de Gestão de Utilizadores (Sistema geral).....	86
Tabela 14: Tabela “users”.....	89
Tabela 15: Tabela “users_internal”.....	90
Tabela 16: Tabela “users_external”.....	90
Tabela 17: Tabela “contacts_enterprise”.....	90
Tabela 18: Tabela “contacts_single”.....	91
Tabela 19: Tabela “product”.....	91
Tabela 20: Tabela “sourcedocs”.....	92
Tabela 21: Tabela “linessourcedocs”.....	93
Tabela 22: Tabela “taxtable”.....	93
Tabela 23: Tabela “taxtype”.....	94
Tabela 24: Tabela “taxcode”.....	94
Tabela 25: Tabela “taxcountryregion”.....	94
Tabela 26: Tabela “sourcedocdefaults”.....	95
Tabela 27: Tabela “sourcedoctype”.....	95
Tabela 28: Tabela “taxexemptionreason”.....	95
Tabela 29: Tabela “paymentmechanism”.....	96
Tabela 30: Páginas definidas no módulo de Utilizadores Externos.....	109
Tabela 31: Páginas definidas no módulo de Contactos.....	111
Tabela 32: Páginas definidas no módulo de Permissões.....	114
Tabela 33: Páginas definidas no módulo de Faturação.....	122
Tabela 34: Páginas definidas no módulo de Produtos.....	124
Tabela 35: Páginas definidas no módulo de Definições.....	126
Tabela 36: Estrutura dos testes de sistema.....	128

Índice de Figuras

Figura 1: Planeamento inicial do projeto de estágio.....	22
Figura 2: Resultado na altura do relatório intermédio, e planeamento do 2º semestre do estágio.....	22
Figura 3: Resultado final do cronograma, ao terminar o 2º semestre do estágio.....	23
Figura 4: Processo envolvido numa Peritagem de Seguros.....	28
Figura 5: O processo do Método em Cascata (fonte: wikipédia[37]).....	45
Figura 6: O processo do Método SCRUM (fonte: wikipédia[38]).....	46
Figura 7: Diagrama de Casos de Uso:Figura 8: Diagrama de Casos de Uso:	51
Figura 7: Diagrama de Casos de Uso:Figura 8: Diagrama de Casos de Uso:	51
Figura 9: Diagrama de Casos de Uso: Documentos Comerciais.....	52
Figura 10: Protótipo: Definições.....	62
Figura 11: Protótipo: Contactos.....	63
Figura 12: Protótipo: Pesquisa de documentos.....	64
Figura 13: Vista de Topo do Sistema.....	70
Figura 14: Sistema Vulnerável a ataques DDOS.....	71
Figura 15: Sistema protegido por CloudFlare.....	71
Figura 16: Sistematização da arquitetura do Drupal (fonte: http://robknight.org.uk/blog/2011/02/explaining-architectural-tiers-drupal).....	75
Figura 17: Perspetiva de Fluxo do Drupal (fonte: https://www.drupal.org/getting-started/before/overview).....	76
Figura 18: Mecanismo de obtenção de Página do Drupal (fonte: https://www.drupal.org/node/10858).....	77
Figura 19: Vista de Módulos da aplicação FactSegur.....	80
Figura 20: Modelo Entidades-Relacionamento das tabelas sobre os utilizadores.....	88
Figura 21: Modelo Entidades-Relacionamento das tabelas de faturação.....	89
Figura 22: Infra-estrutura do ambiente de desenvolvimento.....	98
Figura 23: Activação dos módulos no Drupal.....	99
Figura 24: Exemplo de um ficheiro .info de um módulo.....	100
Figura 25: Exemplo 1 de um ficheiro .install de um módulo.....	101
Figura 26: Exemplo 2 de um ficheiro .install de um módulo.....	101
Figura 27: Exemplo 1 de um ficheiro .module de um módulo.....	102
Figura 28: Exemplo de um método _load() no .module.....	103
Figura 29: Exemplo da definição de elementos num form.....	104
Figura 30: Exemplo da configuração de uma vista do Drupal.....	106
Figura 31: Diagrama de Fluxo do form de Utilizadores Externos.....	108
Figura 32: Diagrama de Fluxo do form de Contactos.....	110
Figura 33: Diagrama de Fluxo do form de Permissões.....	113
Figura 34: Form de Gestão de Faturação e alterações aos dados gerais.....	115
Figura 35: Form de Gestão de Faturação e alterações à tabela de produtos.....	116
Figura 36: Gravação do Form de Gestão de Faturação.....	117
Figura 37: Esquema da exportação para formato SAF-T(PT).....	119
Figura 38: Esquema da Geração de Documentos.....	121
Figura 39: Diagrama de Fluxo do form de Gestão de Produtos.....	123

Figura 40: Diagrama de Fluxo da Página de Definições.....	125
Figura 41: Tempo de resposta para página de Inserir Documentos Comerciais.....	129
Figura 42: Tempo de resposta para página de Inserir Processos.....	130
Figura 43: Tempo de resposta para página de Consultar Documento Comercial.....	130
Figura 44: Tempo de resposta para página de Consultar Processo.....	131
Figura 45: Tempo de resposta para página de Consultar Listagem de Documentos Comerciais.....	131
Figura 46: Tempo de resposta para página de Consultar Listagem de Processos.....	132
Figura 47: Recursos do servidor usados para 1 utilizador em simultâneo.....	132
Figura 48: Recursos do servidor usados para 5 utilizadores em simultâneo.....	133
Figura 49: Recursos do servidor usados para 15 utilizadores em simultâneo.....	133

Acrónimos

API – “Application Programming Interface”

AT – Autoridade Tributária (Portugal).

DGCI – Direção Geral de Contribuintes e Impostos (Portugal).

NIF – Número de Identificação Fiscal, também chamado número de contribuinte.

PME – Pequena-Média Empresa.

Glossário

CRM – “Customer Relationship Manager”, funcionalidade de apoio à gestão das relações comerciais com os clientes da empresa em questão.

ERP – “Enterprise Resource Planning”, Software de gestão e planeamento de um negócio.

IVA – Imposto sobre o valor acrescentado. Imposto aplicado em Portugal que incide sobre a despesa ou consumo efetuadas pelo contribuinte.

RSA – Algoritmo de criptografia de dados, usado para obter as chaves assimétricas (Pública/Privada) necessárias à funcionalidade de assinatura digital de documentos.

SAFT-T(PT) – Ficheiro normalizado (em formato xml) de exportação de dados de faturação, contabilísticos, e outros dados e documentos de relevância fiscal.

SHA-1 – “Secure Hash Algorithm”. Algoritmo de encriptação de dados.

XML – “Extensible Markup Language”. Linguagem que define regras para criar documentos num formato capaz de ser interpretado tanto por humanos ou máquinas.

XSD – “XML Schema Definition”. Descreve formalmente os elementos de um ficheiro XML. Pode ser usado para verificar a correção dos vários elementos deste.

1. Introdução

Este relatório surge no âmbito da cadeira de estágio curricular do Mestrado em Engenharia Informática, área de Engenharia de Software, na Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Este estágio é auto-proposto, sem ser relativo a uma empresa (se bem que existe uma empresa que se disponibilizou a participar na fase de levantamento de requisitos), e no projeto estão envolvidos dois estagiários.

O projeto FactSegur envolve a criação de um software específico para a área de Peritagem de Seguros, também chamada área de Regulação de Sinistros. Este software envolve uma parte de gestão de processos que apoie o modelo de negócios de uma empresa da área, incluindo a gestão de variados dados relativos a cada processo, e a ajuda na criação de relatórios de Regulação e outros documentos, que é a principal atividade das empresas desta área. Este projeto envolve também uma segunda parte que é relativa às necessidades de faturação da empresa, e envolve requisitos legais pois o software de faturação em Portugal necessita de certificação por parte da DGCI.

Um dos estagiários esteve responsável pela parte relativa à gestão de processos, e o outro estagiário (ao qual se refere este relatório) esteve responsável pela parte relacionada com faturação. Ambos passaram pelas mesmas fase de desenvolvimento de projeto, mas ambos produzindo documentos/trabalho relativo a partes diferentes do projeto. Cada aluno apresenta no seu relatório os documentos relevantes à sua parte do projeto.

Este projeto é de grande utilidade porque existem poucos softwares específicos para a área em questão, e menos ainda com funcionalidades de faturação e certificados. A empresa da área foi a própria a fazer esta afirmação, e mencionou pertencer a uma associação de empresas da área, sendo o consenso entre as empresas desta o não existir software para a área que resolva as suas necessidades. A minha própria pesquisa no portal das finanças, bem como na Internet usando variados motores de pesquisa, como o google, capterra, entre outros, também encontrou um conjunto muito limitado de software para a área, conforme apresentado mais à frente no estado da arte.

Um programa de apoio ao negócio é uma grande mais valia, podendo otimizar imensamente o funcionamento de uma empresa, e o facto de conter no mesmo programa a faturação da empresa, em vez de ter de serem usados vários programas é também um ponto importante, ajudando a organização e o funcionamento desta.

A orientação deste estágio esteve a cargo do Professor Doutor Paulo Rupino, da Universidade de Coimbra, tendo tido início a 9 de Fevereiro de 2015, e sendo previsto terminar a 4/5 de Fevereiro de 2016.

1.1. Entidade Auxiliar

Resi – Regulação de Sinistros Lda é uma empresa que realiza a actividade de Regulação de Sinistros ao nível dos vários ramos da Indústria Seguradora. Esta empresa não é uma seguradora, mas trabalha como terceira-parte contratada pelas empresas Seguradoras. A empresa foi constituída a 29 de Dezembro de 1998, e exerce actividade desde essa altura até ao momento, trabalhando principalmente ao nível da zona centro do País.

A Resi – Regulação de Sinistros Lda mostrou interesse num sistema de informação integrado que faça a gestão dos seus dados processuais, e também dos seus dados de faturação, visto de momento ter que usar um programa que só tem a parte de faturação. Usam um outro programa para a parte de gestão de processos, mas este tem várias falhas e está bastante obsoleto, e já não pode ser usado para questões de faturação porque não têm certificação.

Esta empresa não emprega Engenheiros Informáticos, logo este estágio não poderia ser proposto pela empresa, sendo assim um estágio auto-proposto pelos alunos envolvidos, sem ser um estágio formal numa empresa.

A empresa comprometeu-se a contextualizar os estagiários nesta área de negócio específica, e a participar na fase de levantamento e validação de requisitos de modo aos estagiários poderem contar com a sua ajuda para desenvolver um produto que seja útil a empresas do mesmo ramo, com características que apoiem o processo de negócio, e tornem mais expedito e organizado o seu funcionamento.

1.2. Objetivos

No projecto FactSegur composto por 2 estágios pretende-se desenvolver uma aplicação específica para a área de peritos de seguros. Esta profissão possui um conjunto de tarefas a serem desenvolvidas por cada profissional desta área. 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 por si recolhida, e facturar o serviço que prestou.

Neste processo, o 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 facturação (certificado) de modo a poder passar as facturas 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 actividades, 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 aspecto que deixa a desejar, como veremos em capítulos posteriores, um componente que estava em falta ou então um preço demasiadamente caro para pequenas e médias empresas que, neste contexto económico, lutam contra a adversidade em que o país se encontra. Sendo assim, o objectivo principal deste projecto 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 projecto, que serão postas em prática no capítulo 3, com a reunião dos requisitos fundamentais deste projecto.

1.3. Planeamento

1.3.1. Reuniões de Estágio

Foram efetuadas reuniões mensais com o orientador do DEI, o Professor Doutor Paulo Rupino, com ambos os estagiários envolvidos neste projeto. Nestas reuniões foi discutido o trabalho desenvolvido desde a reunião anterior, de modo a dar a conhecer o progresso do estágio. O orientador ofereceu regularmente as suas sugestões para enriquecimento, que ajudaram a organizar o progresso do projeto, e respondeu a dúvidas dos estagiários sempre que tal foi solicitado. Foram também escritos relatórios 15-5 semanais (15 minutos a escrever e 5 minutos a ler), onde foi dado a conhecer periodicamente o progresso ao orientador.

1.3.2. Planeamento

O planeamento do estágio foi efetuado começando por definir inicialmente fases correspondentes aos principais componentes do estágio. Foi tido como ponto de partida a intenção de trabalhar nos variados tipos de documentação no 1º semestre do estágio, e tratar da implementação e validação durante o 2º semestre.

Foram atribuídas datas de início e fim para cada tarefa de acordo com o esforço estimado para cada tarefa, e assim dividido o tempo do estágio pelas tarefas.

O calendário assim obtido foi representado num diagrama de Gantt.

Este diagrama foi usado como referência ao longo do projeto de modo a verificar o progresso do estágio, se havia atrasos, e fazer o planeamento de como se iria responder a esses atrasos.

Vai ser apresentado o calendário inicial com a estimativa feita no início do estágio, e o calendário existente na altura da entrega do relatório intermédio, que foi corrigido de modo a mostrar o resultado final das datas que as fases realmente envolveram, e o reescalamento da calendarização do planeamento futuro até ao fim do estágio.

É apresentado o resultado final do calendário do estágio, de modo a poder ser comparado com os planeamentos na altura do início do estágio, e na altura da defesa intermédia.

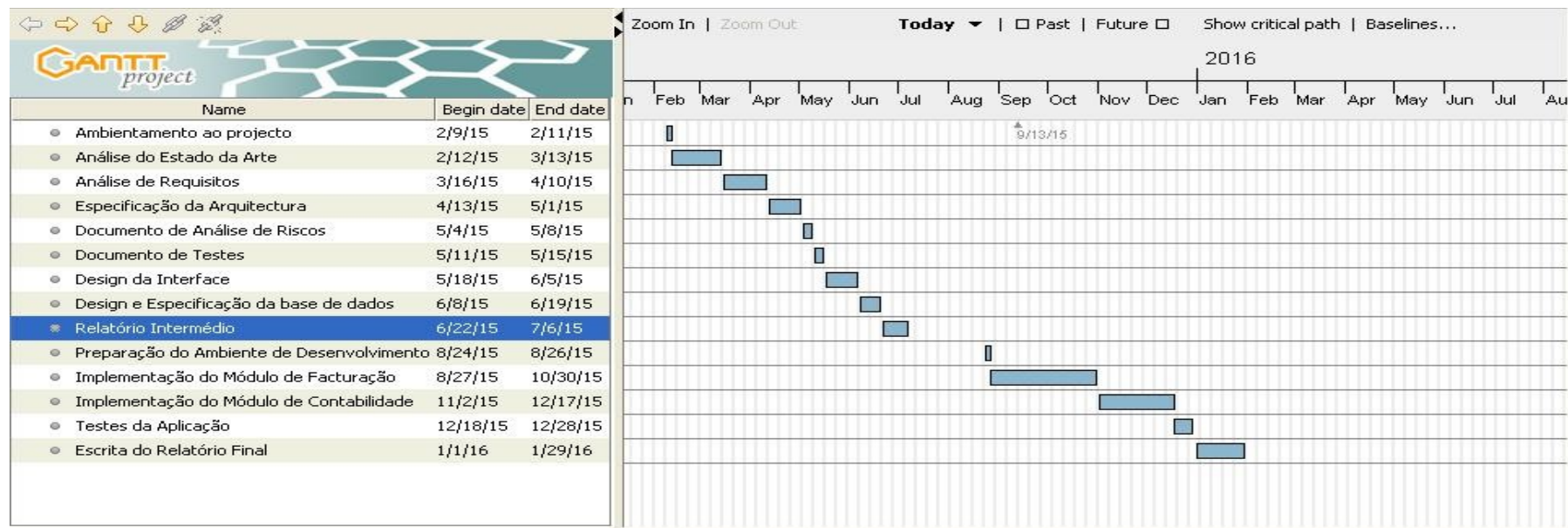


Figura 1: Planeamento inicial do projeto de estágio

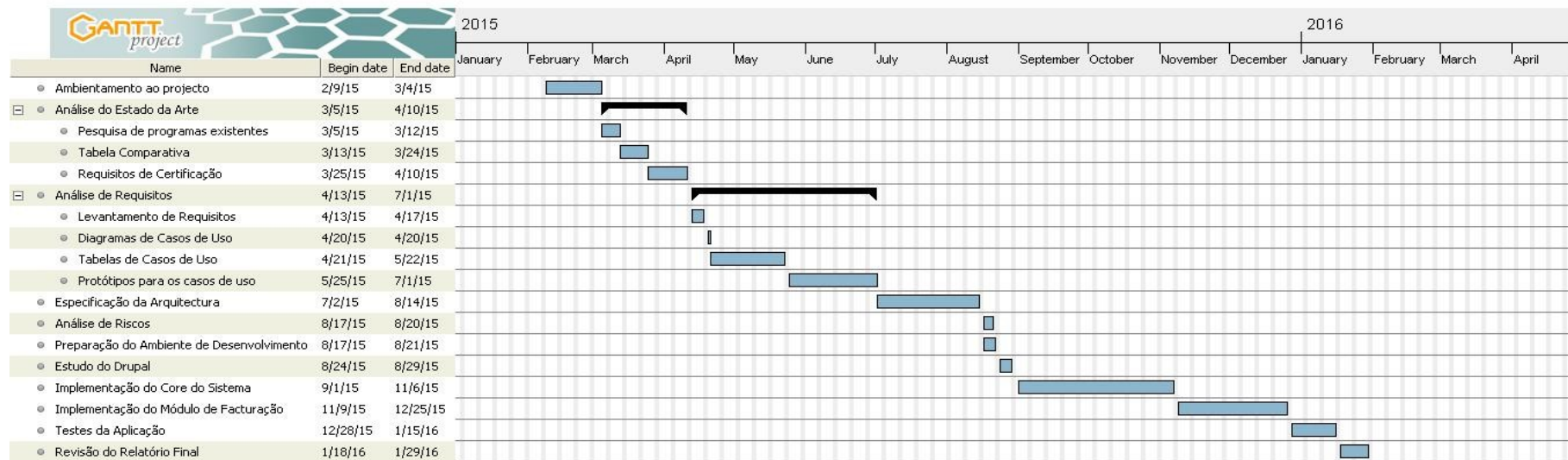


Figura 2: Resultado na altura do relatório intermédio, e planeamento do 2º semestre do estágio

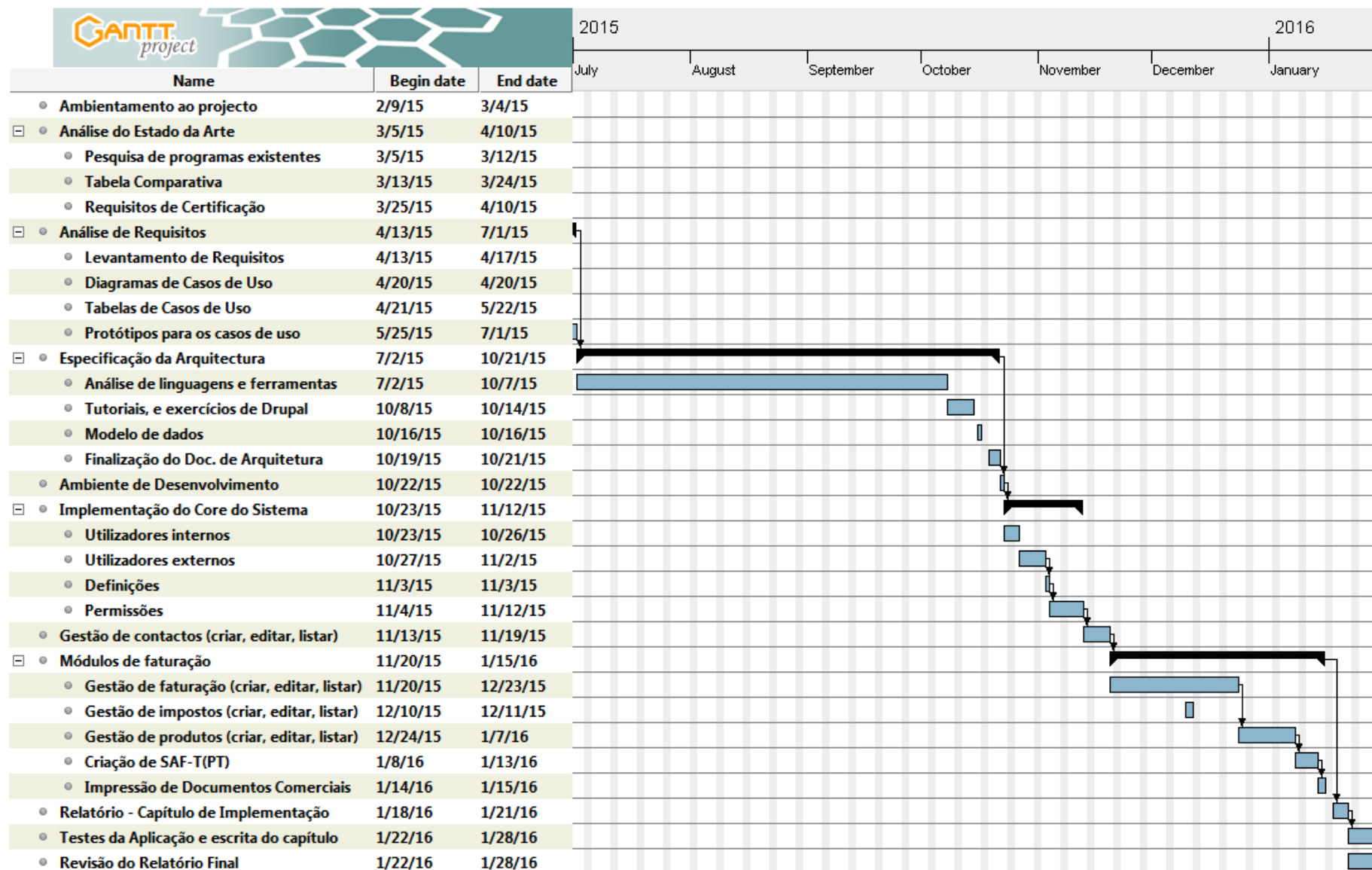


Figura 3: Resultado final do cronograma, ao terminar o 2º semestre do estágio

1.3.3. Desvios ao Planeamento

Logo no início podemos ver que a fase de ambientação ao projeto demorou um mês em vez de demorar 2 dias. A razão para isto foi o facto de o estágio ter sido começado com um mês de atraso, devido a motivos profissionais, por causa de um projeto que estava na altura a ser acabado. Sendo assim este período de ambientação ao projeto representa na sua maioria um período de inatividade.

A fase relativa ao estudo do de Estado da Arte ficou perto do estimado, sendo que a fase de Análise de Requisitos demorou bastante mais do que planeado, um total de 79 dias, em vez dos 24 planeados. Uma razão para isto foi a existência de uma fase de Design da Interface que estava planeada para depois da Especificação da Arquitetura, com uma duração de 18 dias, ter sido efetuada durante esta fase, por meio da elaboração de Protótipos dos casos de uso que já servem como Design da Interface.

Esta fase ocupou efetivamente a grande maioria do primeiro semestre, sendo que houve alguma subestimação do trabalho envolvido no planeamento inicial, mas também as 2 disciplinas de mestrado que o aluno estava a ter ocuparam um pouco mais de tempo que o esperado. Além disso, devido a uma reestruturação do mestrado em Engenharia Informática estas cadeiras foram acrescentadas como obrigatórias e estavam a ser dadas pela primeira vez este ano, sendo que as cargas de esforço para os alunos ainda não estavam muito balanceadas. Acabou por ser utilizado tempo a mais com os trabalhos dessas disciplinas, tendo sido dedicado menos tempo ao estágio por isso.

A fase de Especificação da Arquitetura, com um total de 18 dias na estimativa acabou por demorar bastante mais do que o planeado. Este atraso levou a alguma preocupação, levando a um maior controlo no planeamento.

As tarefas de implementação por foram ordenadas por ordem de importância, de modo a que a implementação ocorresse de acordo com esta, de modo a garantir que as tarefas mais importantes estariam completadas, e caso fosse necessário, as tarefas menos importantes poderiam ser abandonadas, caso não houvesse tempo para estas.

A fase de implementação decorreu de acordo com a estimativa para esta, mas não conseguiu recuperar do atraso que já vinha das fases anteriores. Devido a isto tiveram de ser abandonados alguns casos de uso, relativos ao módulo de impostos e ao módulo de mapas de faturação. A fase de testes no final decorreu normalmente, sendo que não houve muito tempo para efetuar a revisão do relatório final.

1.4. Estrutura do Documento

Este relatório está estruturado em 8 capítulos principais.

Este primeiro capítulo visa dar uma ideia do contexto onde o projeto se insere, uma explicação simples da área para onde este vai estar direcionado, uma apresentação da empresa do ramo que irá auxiliar os estagiários de modo a estes perceberem as necessidades de uma empresa do mesmo ramo, e também quais são os objetivos a serem atingidos. É também indicado como foi dividido o planeamento do estágio, e apresentadas as divergências ao planeado.

O segundo capítulo é dedicado ao estudo do Estado da Arte, que introduz a área em questão, e visa saber o estado do mercado relativo a programas que possam ser concorrentes, e fazer uma comparação entre estes. Também é efetuado um estudo relativo aos requisitos de certificação para software de faturação em Portugal.

Podemos encontrar no terceiro capítulo o método de desenvolvimento escolhido e quais as razões de utilização deste, e uma explicação do que ele envolve.

De seguida o quarto capítulo contém o estudo da Análise de Requisitos, onde são explicadas as funcionalidades que o sistema vai conter relativamente ao nível da faturação, e um estudo dos requisitos não funcionais para o sistema como um todo.

O quinto capítulo representa a documentação da Arquitetura que foi planeada para o sistema.

No sexto capítulo está detalhado o percurso da implementação do sistema, e as particularidades das soluções encontradas para os casos de uso definidos.

Existe um sétimo capítulo que indica como foram planeados e conduzidos os testes e a validação dos requisitos.

Por fim o capítulo oito contém uma conclusão onde é ponderado o projeto e o estágio em si, e uma reflexão sobre qual será o futuro esperado do projeto, o que pode ser melhorado nele, as coisas que podem ser acrescentadas, e as perspetivas para este.

2. Estado da Arte

As aplicações de apoio ao negócio são de extrema importância para o funcionamento de uma empresa, e aumentam de importância com a dimensão da empresa, porque quanto maior a empresa, maior a tarefa de gestão associada ao seu funcionamento [1].

O exemplo mais conhecido destas aplicações são os Sistemas ERP (“Enterprise Resource Planning”). Estes sistemas são importantes para as empresas, porque permitem gerir eficientemente uma empresa, tanto ao nível da camada de tomada de decisão empresarial, como ao nível da gestão financeira, bem como o acompanhamento e apoio ao processo de negócio de uma empresa [1].

A partir de 2011 os sistemas de faturação e contabilidade em Portugal passaram a necessitar de uma certificação de software por parte da DGCI, tendo de cumprir um conjunto de requisitos destinados a garantir a inviolabilidade da informação registada neles, e a sua transmissão regular à DGCI para efeitos de controlo fiscal [2]. Esta necessidade de certificação forçou muitas empresas a abandonar os programas de faturação que usavam por não serem certificados, e a adotarem uma solução diferente. Isto abriu muitas oportunidades de negócio na área de software de faturação.

Neste projeto queremos abordar a área de Peritagens de Seguros que não tem quase nada a nível de software específico, e menos ainda com faturação integrada no mesmo sistema que faz a gestão do processo de negócio da empresa.

Esta fase inicial do projeto consistiu numa análise de mercado, de forma a perceber quais as possibilidades a nível de software que existem para empresas da área adotarem na gestão do seu processo de negócio, bem como uma ideia simples dos custos envolvidos em adotar essas soluções, e os modelos de negócio existentes das empresas que fornecem soluções.

Este estudo tinha como objetivo conhecer quais os produtos possivelmente concorrentes, bem como a possibilidade e rentabilidade de usar uma solução externa de faturação, e fazer a integração desta com uma solução nossa de sistema de informação para gestão do negócio de uma empresa da área de Peritagens. Este estágio foca-se na parte de faturação, logo a pesquisa será relativa a produtos que contenham faturação, quer sejam da área de negócio específica, ou genéricos, sendo que o estágio do outro aluno será mais focado na pesquisa de produtos que façam a gestão dos processos e gestão do negócio, mesmo que não contenham funcionalidades de faturação.

É inicialmente apresentada uma introdução à Área específica a que nos dirigimos, incluindo um breve resumo do seu modelo de negócio.

De seguida é apresentada informação genérica sobre algumas soluções já existentes.

Depois é feita uma reflexão sobre a possibilidade de adotar uma solução de faturação já existente e integrá-lo num sistema de informação novo.

O próximo passo foi explicar sucintamente quais foram os critérios de comparação e o seu significado, seguido pela tabela de comparação de várias soluções.

O passo seguinte foi apresentar um resumo dos requisitos necessários atualmente (podem mudar com novos decretos-lei) para a certificação de software de faturação em Portugal.

2.1. Introdução à Área de Peritagens de Seguros

Este sistema de informação é específico para a área de Peritagens de Seguros. Esta área pode ser identificada também como atividade Regulatória de Sinistros, bem como atividade de Avaliação de bens e Danos, que é o código das finanças para empresas deste ramo.

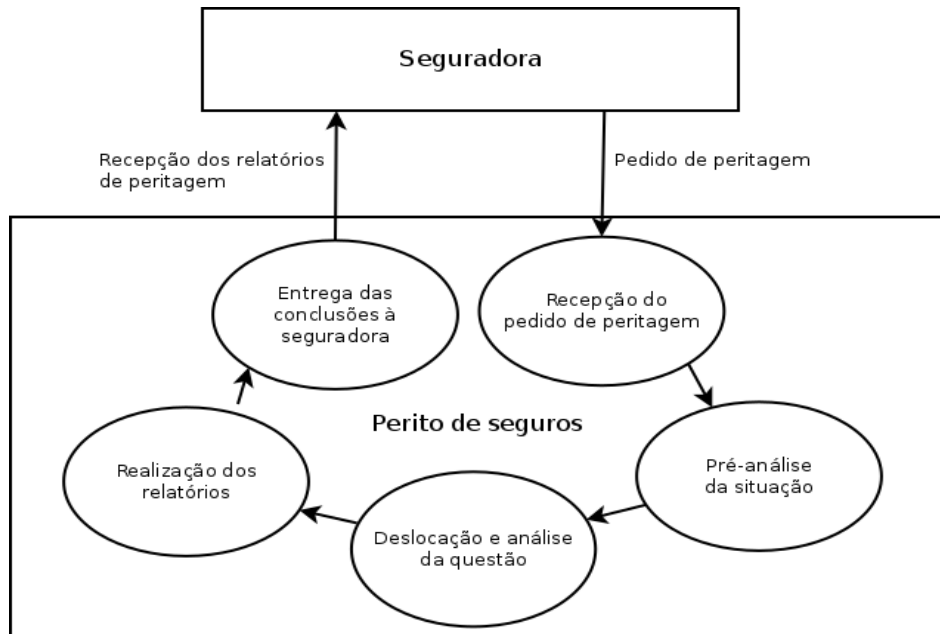


Figura 4: Processo envolvido numa Peritagem de Seguros

As empresas deste ramo realizam serviços para as Seguradoras (os seus clientes), que consistem na realização de um relatório de avaliação de danos sobre um certo acidente. A empresa de Peritagem por meio de Peritos especializados tem de identificar variados dados relativos ao processo e partes envolvidas, contactar com a entidade segurada (e por vezes outras entidades envolvidas, como outras pessoas presentes no acidente, polícia, bombeiros, entre outras), ir ao local do acidente e outros locais envolvidos para verificar e documentar o local do acidente e objetos envolvidos, por meio de fotografias do dano ou outros elementos. Depois deve ser fixado um valor regulatório para os danos associados ao acidente, que será descrito no relatório de Peritagem, que depois de finalizado é entregue à seguradora. A seguradora irá usar os resultados desse relatório para fixar o valor definitivo da indemnização que é atribuída à entidade segurada.

De acordo com o portal nacional de empresas em Portugal existem 947 empresas registadas para esta área em específico, sendo este o nosso mercado alvo [3].

No fim do processo, em que já existe um relatório com os resultados da Peritagem, a empresa de Peritagem terá de emitir faturas e recibos em nome da Seguradora, de modo a receber desta o pagamento pelos seus serviços. Estes documentos normalmente irão conter certos dados do processo (especialmente a identificação do processo na empresa de Peritagens, da empresa seguradora, e a identificação da pessoa segurada) para além dos dados fiscais. Devido a este fator é conveniente que seja o mesmo programa a gerir o processo e a emitir os documentos, de modo a facilitar a passagem destes dados de um lado para o outro (se for o mesmo programa podem ser importados automaticamente). Sendo o mesmo programa, também é possível na parte de processos existir uma gestão

que permita o controle de quais os processos que ainda não foram faturados, os quais os processos que já foram pagos. Isto poderá reduzir a possibilidade de erros como por exemplo o esquecimento de faturar um processo, e este ficar por pagar.

Em relação a programas de faturação, de acordo com a lista de programas certificados disponibilizada no portal das finanças existem de momento 2358 programas certificados pela DGCI [4].

2.2. Requisitos de Certificação de Software de Faturação em Portugal

Os requisitos para software de faturação certificados são alterados com alguma frequência, tendo até ao momento sido alterados pelo menos 2 vezes por ano, tipicamente existindo em cada ano uma alteração ao formato Saf-T(PT) e outra às portarias que definem os variados requisitos de certificação [2][5]. Os produtores de software são obrigados a garantir a atualização do seu software de modo a este continuar a cumprir os requisitos quando eles são alterados, não sendo necessário voltar a certificar o mesmo software quando este muda de versão.

A mudança de versão no formato Saf-T(PT) pode ser verificada usando a página acerca deste na secção de Apoio ao Contribuinte do portal das Finanças [5]. A mudança de outros requisitos de certificação pode ser verificada usando a página sobre a Certificação de Software de Faturação também na secção de Apoio ao Contribuinte do portal das Finanças [2]. Verificando a data dos decretos-lei, portarias, e ofícios técnicos existentes nessas páginas facilmente é verificado se houve alguma alteração relativamente aos Requisitos de Certificação.

Quando os requisitos mudam as empresas produtoras de softwares já certificados irão necessitar de os atualizarem de maneira a continuarem a cumprir com estes requisitos, ou correm o risco de perder a certificação.

Os principais requisitos que se aplicam neste projeto são resumidos e descritos nas secções seguintes, fazendo um resumo dos pontos relevantes nos decretos-lei e portarias envolvidas.

2.2.1. Isenção da necessidade de Certificação

Segundo a portaria n.º 363/2010, de 23 de Junho [6] existiam algumas situações de exclusão à necessidade de certificação para software de faturação. Um situação que se destacava, devido à sua utilidade para o desenvolvimento de novos softwares, era a exclusão de certificação para software produzido internamente ou por uma empresa do mesmo grupo económico, do qual fossem detentores dos direitos de autor. De acordo com as alterações a esta portaria através da portaria n.º22-A/2012, de 24 de Janeiro [7], e a portaria n.º340/2013, de 22 de Novembro a maioria das situação de exclusão foi revogada. De momento é sempre necessário a certificação, exceto quando no período de tributação anterior, o volume de negócios foi inferior ou igual a 100.000 euros, ou para documentos emitidos através de aparelhos de distribuição automática ou prestações de serviços, em que seja habitual a emissão de talão, bilhete de ingresso ou de transporte, senha ou outro documento pré-impresso ao portador comprovativo do pagamento.

2.2.2. Assinatura Digital de Documentos Comerciais com chave RSA

Os programas informáticos de faturação devem assinar todos os documentos emitidos com eficácia externa, com exceção dos recibos (Faturas, Documentos Retificativos, Orçamentos, Guias de Transporte, etc). Esta assinatura digital deverá ser criada através de um par de chaves RSA Pública/Privada e encriptação usando o algoritmo SHA-1. A chave privada a usar na encriptação deverá pertencer à empresa produtora do software, e deverá ser de conhecimento exclusivo desta, e a chave pública correspondente à chave privada deverá ser dada a conhecer às Finanças pela empresa, e poderá ser usada para validar a autenticidade e integridade do conjunto de dados no ficheiro Saf-T(PT).

Os detalhes para assinatura digital de documentos estão descritos no documento “Especificação dos requisitos técnicos”, Despacho nº8632/2014 de 03 de Julho, do Diretor-geral da AT [8].

Esta assinatura é criada com o algoritmo SHA-1 encriptando uma linha de texto contendo a data do documento, a data de entrada no sistema, a identificação do documento, o total com IVA, e a assinatura digital do documento anterior do mesmo tipo e da mesma série. Desta assinatura são obtidos 4 caracteres vindos de posições específicas da assinatura (1ª, 11ª, 21ª, 31ª). Os documentos assinados devem conter estes caracteres, uma expressão identificando o número de certificação do programa que originou o documento, e a identificação única do documento. O ficheiro Saf-T(PT) irá conter a assinatura na sua totalidade, e não apenas os caracteres presentes no documento. A assinatura só deverá ser gerada na altura de emissão do documento, e guardada na base de dados a partir desse momento. Também deve ser guardado o número de versão da chave privada usada para assinar.

Usando a chave pública e o algoritmo SHA-1 pode ser validado o ficheiro Saf-T(PT) verificando se os dados de cada documento nesse ficheiro correspondem aos dados com os quais a assinatura foi criada. Caso não haja correspondência isto indicará que os dados existentes na aplicação não são consistentes com os dados existentes no ficheiro Saf-T(PT), indicando por exemplo que os dados foram alterados. Visto este ficheiro ser um simples documento de texto este poderá facilmente ser alterado depois de ser exportado pelo programa.

Conforme explicado em cima, a assinatura de cada documento digital vai estar ligada ao documento anterior, logo caso algum documento seja apagado do ficheiro Saf-T(PT), ou da base de dados diretamente sem passar pelo programa, as assinaturas digitais dos documentos a partir desse deixariam de estar válidas, porque falhará a validação da assinatura do documento seguinte, por a assinatura não corresponder ao esperado. (No programa um documento nunca poderá ser apagado, apenas anulado, e os seus dados tal como a assinatura continuam disponíveis, logo este problema não se verificaria)

2.2.3. Saft-T (PT)

A norma Saft-T(PT) (“Standard audit file for Tax Purposes - Portuguese Version”) vai definir o modelo para um ficheiro de dados normalizado em formato XML.

Este modelo tem como objetivo permitir uma exportação fácil, e em qualquer altura, de um conjunto de registos contabilísticos, de faturação, de documentos de transporte, e recibos emitidos, num formato legível e comum, independentemente do programa utilizado, da sua estrutura interna de dados e funcionalidade.

Esta normalização facilita então a extração e tratamento da informação, e facilita a recolha em formato eletrónico dos dados fiscais relevantes por parte dos

inspetores/auditores tributários, que podem assim ter um meio mais eficaz de combate à evasão fiscal.

As empresas são atualmente obrigadas a comunicar mensalmente os dados das suas faturas e contabilidade através do envio do ficheiro Saf-T(PT) exportado mensalmente pelo programa de faturação certificado, submetendo este ficheiro no site e-fatura do portal das finanças [9]. Alternativamente, estes dados podem ser transmitidos em tempo real pelo programa através de um Web-Service disponibilizado pela AT, por inserção direta dos dados no portal das finanças, ou por via eletrónica através da submissão do modelo oficial de declaração para a comunicação dos elementos das faturas aprovado pela Portaria 426-A/2012, de 28 de Dezembro.

O formato de um ficheiro Saf-T(PT) está definido na portaria nº321-A/2007 de 26 de Março [10], e foi alterada para novas versões pela portaria nº1192/2009 de 08 de Outubro [11], pela portaria nº160/2013 de 23 de Abril [12], e pela Portaria nº274/2013 de 21 de Agosto [13].

No portal das finanças a AT fornece aplicações que podem ser usadas para validação de ficheiros Saft-T(PT)[5], e também um ficheiro XSD de estrutura de dados que pode ser usado para validar um ficheiro Saft-T(PT) depois da sua criação.

De seguida apresento um texto produzido no estágio de modo a explicar a estrutura da norma Saf-T(PT).

Em cada ficheiro Saf-T(PT) irá existir uma tabela de cabeçalho com os dados da empresa que criou o ficheiro, bem como alguns dados da empresa que produziu o software de faturação usado por esta. Existirá outra tabela com a lista de clientes e seus dados, e outra tabela que define os tipos de impostos utilizados, e outra tabela com os recibos emitidos.

Deve existir uma tabela de fornecedores, uma tabela de produtos e serviços, uma tabela de documentos comerciais, uma tabela de movimento de mercadorias, e uma tabela de documentos de conferência de entrega de mercadorias ou de prestação de serviços.

A tabela de cabeçalho contém dados como a versão do Saf-T(PT) do ficheiro, a identificação do registo comercial da empresa e número de contribuinte desta, nome e morada da empresa, data de início e fim do período do qual este ficheiro foi emitido (normalmente será de um mês), data de criação do ficheiro, identificação do programa certificado com o qual o ficheiro foi emitido, seu número de certificação e informação sobre o produtor do programa.

A tabela de códigos de contas é a prevista pelo sistema de normalização contabilística e outras disposições legais para o respetivo sector de atividade. Irá primariamente definir o tipo e hierarquia de conta, bem como os valores de Saldo de abertura e Saldo de encerramento a crédito e a crédito da conta do plano de contas.

A tabela de clientes irá conter os variados dados relativos a cada cliente, especialmente o código do cliente, o seu número de identificação fiscal, o seu nome e morada, bem como a morada para onde os produtos são expedidos, e se o cliente tem acordo de auto faturação.

A tabela de fornecedores irá conter os dados relativos aos fornecedores da empresa, tendo praticamente os mesmo campos que a tabela de clientes.

A tabela de produtos deverá conter para cada produto a identificação do tipo de produto (Produto/Serviço/Outro/Imposto), o código, família, descrição, e valor do código de barras.

A tabela de impostos irá conter os vários regimes fiscais de IVA aplicáveis nas linhas dos documentos. Para cada elemento da tabela deverá existir um código do tipo de imposto (IVA, IS, NS), identificação do país ou região do imposto, código que identifica o imposto no programa, descrição do imposto, e qual a percentagem ou valor fixo aplicável a este imposto.

A tabela de movimentos contabilísticos contém o registo de movimentos contabilísticos correspondentes ao período de exportação a que diz respeito o ficheiro Saf-T(PT). Irá conter o número de registo de movimentos contabilísticos, bem como os valores totais de débito e crédito de todos os movimentos do período selecionado para o ficheiro. Cada movimento conterá uma entrada de diário, que entre outras coisas identifica o código e descrição da entrada de diário, a chave única do movimento contabilístico, a data do documento, a identificação do utilizador que registou o movimento, bem como do cliente e/ou fornecedor, e uma ou mais linhas do diário onde estarão indicados a chave da linha no diário, o código da conta associada ao movimento, a identificação do documento comercial relacionado com a linha, a data de entrada no sistema, a descrição da linha, e o valor a débito ou crédito desta.

A tabela de documentos comerciais a clientes deve conter todos os documentos de venda e retificativos emitidos pela empresa, incluindo os anulados. Esta tabela começa por indicar o número total de documentos, bem como os valores totais de débito e de crédito (sem imposto). Para cada documento deverá existir uma identificação única do documento (relaciona série, tipo de documento e numeração sequencial para estes), informação sobre o estado atual do documento, data hora e utilizador responsável pela última modificação, motivo de anulação caso aplicável, origem do documento, dados da assinatura digital, data da emissão do documento, tipo do documento, código do utilizador que criou o documento, informação sobre a entrega de mercadoria se aplicável, e linhas indicando os variados produtos e/ou serviços e impostos. Cada linha conterá o número de linha, dados sobre o documento de origem (fatura no caso de documentos retificativos como notas de crédito e débito), identificador do produto na tabela de produtos e sua descrição, quantidade, preço unitário, data de envio de mercadoria ou prestação de serviço, descrição da linha, valor a débito ou crédito, dados da taxa correspondentes a uma taxa na tabela de impostos. No fim de cada documento deverá existir informação sobre os totais do documento e dados sobre o pagamento, bem como os dados sobre a retenção na fonte caso aplicável. Os dados sobre o pagamento e retenção na fonte são para documentos do tipo fatura-recibo, visto não existir um recibo com essa informação.

A tabela de movimentação de mercadorias deve conter a informação sobre os documentos que sirvam como documentos de transporte, nomeadamente guias de transporte ou de remessa. Documentos que já façam parte da tabela de documentos comerciais a clientes que também sirvam como documento de transporte não deverão constar desta tabela. Os campos de dados são praticamente os mesmos que os da tabela de documentos comerciais a clientes, identificando o número de documentos, total de quantidades, e para cada documento identificar o seu código, data, tipo de documento, assinatura digital, cliente, utilizador que criou o documento e utilizador que fez a última alteração, identificação do local de origem e destino dos bens, bem como as datas e horas para o início e fim do transporte. Em cada linha do documento é identificado o produto, a sua quantidade, preço unitário, imposto, e outros campos.

A tabela de documentos de conferência de entrega de mercadorias ou de prestação de serviços deve conter a exportação de documentos suscetíveis de apresentação ao cliente para conferência de entrega de mercadorias ou da prestação de serviços, mesmo que

objeto de faturação posterior. Orçamentos e as faturas pró-forma são exemplos de documentos deste tipo. Não devem ser exportados documentos que já constem da tabela de documentos comerciais a clientes ou a tabela de movimentação de mercadorias. Mais uma vez a tabela terá os mesmos campos que a tabela de documentos comerciais a clientes exceto não conter informação sobre transporte porque neste documento ainda não existe venda de produtos ou prestação de serviços, o número total de documentos, somas do total de débito e crédito sem imposto, e para cada documento o seu estado, identificação do cliente, assinatura digital, e linhas identificando os vários produtos, quantidades vendidas, valores e taxas de imposto. No fim do documento é apresentado a soma total de crédito e débito do documento, bem como o total de imposto a pagar.

A tabela de recibos emitidos deve conter a exportação dos recibos emitidos. A informação será a mesma que a existente na tabela de documentos comerciais a clientes, até porque como no caso da fatura-recibo esses documentos podem servir de recibo, os dados próprios de recibo têm de estar representados também nessa tabela. A maior diferença é que em cada produto/serviço o recibo tem que indicar o documento que deu origem a cada entrada para fazer a ligação entre o recibo e a fatura(s) ou documento(s) que deram origem a esse produto/serviço no recibo. Outra diferença é que os recibos não necessitam de assinatura digital.

2.2.4. Outros requisitos para o Programa

Estes requisitos estão enunciados no Despacho nº8632/2014 de 03 de julho, do Diretor-geral da AT, que contém a Especificação dos requisitos técnicos. [8]

Foram selecionados a partir desse documento os requisitos relevantes ao nosso projeto, estes foram separados nestas três categorias representadas nas três tabelas seguintes, e cada requisitos seguinte corresponde a um ponto da especificação, ou um resumo/explicação deste efetuada pelo estagiário. Outros requisitos existem, mas referem-se a outras áreas ou conteúdos que este projeto não irá conter, como por exemplo a existência de contabilidade, em vez de somente faturação.

Programa deve possuir
Adequados controlos de acessos ao sistema informático, obrigando a autenticação de cada utilizador, e devendo obrigar o utilizador a mudar a palavra passe no primeiro acesso.
Implementada uma política de cópias de segurança de periodicidade obrigatória de forma a minimizar o volume de dados a recuperar em caso de corrupção da base de dados e/ou a manutenção de duas ou mais bases de dados simultâneas para que quando uma se corrompa a(s) outra(s) assegure(m) a continuidade da faturação.
Controlo direto ou indireto da base de dados que utiliza e o registo do nº de reposições de cópias de segurança (backup) efetuadas, por exemplo através de sistemas de controlo que validem a base de dados no fecho e arranque da aplicação, de forma a evidenciar eventuais manipulações ou alterações de dados na base de dados, por outra via que não a aplicacional.
Controlo direto ou indireto da base de dados que utiliza e o registo do nº de reposições de cópias de segurança (backup) efetuadas, por exemplo através de sistemas de controlo que validem a base de dados no fecho e arranque da aplicação, de forma a evidenciar eventuais manipulações ou alterações de dados na base de dados, por outra via que não a aplicacional.
Proteção eficaz para a chave privada, incluindo durante o processo de assinatura dos documentos.
Não dispor de qualquer função que, no local ou remotamente, permita alterar, direta ou indiretamente, a informação de natureza fiscal, sem gerar evidência desta alteração agregada à informação original.
Capacidade de inserir no programa os dados de faturas manuais impressas em tipografia autorizada para uma série própria. (estas só poderão ser emitidas em caso de inoperacionalidade do programa).
Capacidade de inserir no programa recuperação de documentos em falta para uma série própria. (por exemplo documentos que já tinham sido emitidos que se perderam numa reposição da base de dados)

Tabela 1: Requisitos de Certificação – Programa deve Possuir

Programa deve assegurar
Que todos os tipos de documentos, identificados através das respetivas designações, deverão ser emitidos cronologicamente em uma ou mais séries, convenientemente referenciadas, de acordo com as necessidades comerciais, devendo ser datados e numerados de forma progressiva e contínua, dentro de cada série.
Que na identificação de documentos não devem ser utilizados caracteres que violem o esquema de validação ou possam ser interpretados como operadores de XML. Não pode constar da sequência numérica qualquer outra informação (como por exemplo o ano) que, a existir, deverá sempre constar da identificação da série.
Que o código identificador das séries deve ser específico de cada um dos estabelecimentos e/ou programas, e nunca pode ser repetido no mesmo contribuinte, para o mesmo tipo de documento, de modo a identificar univocamente cada documento emitido, mesmo que os documentos sejam emitidos por mais do que um programa de faturação.
A garantia que não existe mais do que um documento ativo proveniente da recolha do mesmo documento manual, ou documento integrado através de duplicados que não integram a cópia de segurança quando houver necessidade de reposição de dados por inoperacionalidade do sistema.
Que documentos emitidos em modo de formação por equipamentos ou programas de faturação não certificados devem conter menção expressa desse facto.
Que documentos que não sejam faturas ou documentos retificativos de fatura devem conter de forma evidente a sua natureza, e conter a expressão "Este documento não serve de fatura".
Que todos os documentos (com exceção dos recibos) quando na impressão tiverem mais do que uma página devem mostrar os totais por página a transportar e transportado, nº de cada página, nº total de páginas, sendo que os totais globais e de impostos apenas podem ser impressos na última página.
Que a expressão "Processado por programa certificado nº ..." não pode ser a primeira nem a última linha impressa no documento, de modo a que não exista a possibilidade (em caso de problema com a impressora) de não ser impresso.
Que apenas produtores de aplicações podem parametrizar ou desenhar os impressos dos documentos, ou então têm que garantir/validar que os impressos modificados/criados por outros estão de acordo com a lei, por exemplo através de assinatura digital.
Que nenhum documento em estado de preparação ou pré-visualização possa ser impresso num momento anterior à sua finalização e assinatura.
Que os documentos impressos pelo programa de faturação não devem conter valores negativos. Quando necessário, serão utilizados documentos retificativos de faturas (notas de crédito e notas de débito), como documentos de correção de operações de compra e venda, cuja forma, conteúdo e finalidade devem ser respeitados. Os valores negativos apenas poderão ser impressos nos casos de anulação de registos que já integram o documento ou para acerto de estimativas nas prestações de serviços continuadas. O valor negativo nunca poderá ser superior ao valor positivo da mesma rubrica ou serviço em cada fatura. Caso o acerto, por rubrica, seja superior ao valor positivo, estamos perante uma regularização que obriga a emissão da respetiva nota de crédito.
A utilização, para efeitos de cálculos, de valor com mais do que 2 casas decimais para evitar erros de arredondamento, motivadas por descontos, preços unitários inferiores ao cêntimo, quantidades fracionadas, taxas de câmbio, ou pela emissão de documentos em que o preço tem o imposto incluído.
Que no ficheiro Saf-T(PT) os valores do campo "Total do documento com impostos" devem ser exportados com o mesmo valor que foi considerado na assinatura, isto é, arredondado a duas casas decimais.
A exigibilidade ao utilizador do motivo de não apuramento do imposto, quando tal se verificar.

Tabela 2: Requisitos de Certificação – Programa deve Assegurar

Programa não pode permitir
Ser o utilizador a definir quais os tipos de documentos que são assinados e/ou exportáveis para o Saf-T(PT), especialmente os que foram criados ou modificados por aquele.
O processamento de qualquer cálculo sobre documentos recolhidos ou resultantes de integração de outros sistemas.
A reutilização de códigos de utilizador após o respetivo utilizador ter procedido à realização de movimentos fiscalmente relevantes.
A alteração do NIF numa ficha de cliente já existente e com documentos emitidos.
A alteração do nome e morada numa ficha de cliente já existente e com documentos emitidos, mas cujo NIF não esteja identificado.(NIF de consumidor final 999999999) Nesse caso poderá ser averbado o NIF em falta, e depois na ficha de cliente já poderá ser alterado o nome e morada, mas não o NIF.
A alteração numa ficha de produto já existente e com documentos emitidos, do campo Descrição do produto ou Serviço.
A criação de notas de crédito relativas a documentos anulados ou já totalmente retificados.
A anulação de documentos sobre os quais já tenha sido emitido documento retificativo, ainda que parcial, sem a prévia anulação dos documentos retificativos.

Tabela 3: Requisitos de Certificação – Programa não pode Permitir

2.3. Soluções já existentes de faturação

Não existem muitos sistemas de informação que forneçam funcionalidades específicas para a área de Peritagens de Seguros, tal como gestão de processos, e menos ainda com soluções de faturação integradas. Sendo que este estágio está focado no módulo de faturação, o meu estudo estará focado nesta área, sendo que o outro estagiário envolvido no projeto irá apresentar em mais detalhe as soluções com funcionalidades de apoio ao negócio específicas para a área.

Neste estudo aparecem não só os sistemas de informação e ERP que contenham soluções específicas para esta área, mas também software genérico de faturação que as empresas podem usar independentemente, usando outra solução para controlo do seu modelo de negócio que não tenha faturação.

Foram encontradas 2 empresas com software de faturação certificado próprio para a área que utilizam uma solução própria certificada com o nome da empresa. Estas foram encontradas a partir da lista de programas certificados [4] disponibilizada no portal das finanças. Estas empresas são a PeriAva, Lda e a PeriAgro, Peritagens Agrícolas e Avaliações Fundiárias SA. Não havendo informação disponível na Internet, as empresas foram contactadas, e foi descoberto que estas soluções funcionam apenas para uso da própria empresa, não sendo comercializadas para o público em geral.

Em relação a software de faturação que não seja específico da área existem imensos no mercado, e foi feito um estudo comparativo que visava obter uma ideia da gama de preços praticados, bem como as capacidades de integração destes com soluções externas.

CLOUDWORKS

Foi encontrada uma solução chamada CloudWorks [14] para a área de Peritagens, encontrada por ter publicitado na sua página da Internet [14], e numa página sua do LinkedIn, o facto de ter experiência no desenvolvimento de soluções para o sector segurador/Peritagens [15].

A empresa foi contactada por email de modo a saber mais informação, visto a página da Internet apenas conter breves detalhes.

Esta solução contém uma plataforma de Gestão de Peritagens, mas que ainda não está a ser anunciada publicamente pela empresa, estando provavelmente em fase de preparação para o seu lançamento num futuro próximo. Esta é específica para a área de Peritagens, com todas as capacidades para a gestão do processo de negócio de uma empresa desta área. Ela não contém software de faturação próprio certificado em seu nome, mas integrou na sua plataforma um software de faturação de outra empresa, sendo que a plataforma tem também portanto as funcionalidades de faturação integradas, e pode operar a nível de faturação em Portugal. O preço foi indicado como sendo 75€ por utilizador e por mês.

Esta solução será portanto um concorrente direto, a solução mais próxima do âmbito do nosso projeto que foi encontrada.

MRINFORMÁTICA

Existe um software específico para a área de Peritagens chamado MRInformática [16],

pertencente a uma empresa Angolana, fundada por dois especialistas informáticos portugueses, que trabalham no ramo desde 1998.

Após um contacto por email foi descoberto que este software contém gestão de utilizadores/clientes, gestão de processos, gestão documental.

Este software não contém a certificação necessária para operar em Portugal a nível de faturação, mas indicaram serem parceiros Primavera, e poderem integrar a solução deles com o Primavera de modo a poder fazê-lo, visto o Primavera conter certificação. A empresa também não quis disponibilizar informação sobre preços, até porque o preço iria depender do trabalho de integração com o Primavera, limitando-se a dizer uma pequena lista das funcionalidades específicas para a área da sua solução.

ERP (SAP, PRIMAVERA, PHC, SAGEERP)

Sistemas ERP como SAP [17], Primavera [18], PHC [19], SageERP [20] não têm um valor fixo para os seus produtos, principalmente porque não vendem simplesmente os módulos. Existe uma forte componente de personalização, que envolve não só a seleção destes módulos, como também a configuração e personalização destes para se adaptar aos processos de negócio da empresa alvo. Isto implica também um estudo sobre esses processos de negócio, e em alguns casos existe mesmo a necessidade da empresa mudar o seu processo de negócio de maneira a usar um processo mais otimizado, ou que se adapte melhor à utilização do sistema ERP [1]. Existem assim vários custos associados à implementação inicial do ERP na empresa, para além do custo do Sistema em si.

No caso específico do ERP SAP foi verificado que este tem um módulo específico para a área, chamado SAP Claims Management [21]. Isto torna o SAP bastante atrativo para empresas da área de Peritagens.

Em outras soluções ERP não foram encontrados módulos específicos para a área, sendo que estes poderiam ser personalizados de maneira a dar resposta às necessidades de uma empresa até certo ponto, mas não existiria à partida um sistema pronto a funcionar com a gestão do negócio da empresa, logo não seriam tão úteis e intuitivos como a solução do SAP, e possivelmente iriam acarretar mais custos na tarefa de personalização do ERP.

PRIMAVERA EXPRESS

O Primavera Express [22], ao contrário do ERP do mesmo grupo (Primavera), é uma solução gratuita (com a limitação que o volume de negócio da empresa seja inferior a 30000€ anuais) que permite cumprir com as necessidades de faturação. Esta permite a criação de documentos comerciais como faturas e recibos, a gestão de stocks, a gestão de contas correntes, a exportação do ficheiro SAF-T(PT), acesso a dados estatísticos de clientes e produtos, e a mapas de apoio à decisão.

Não contém nada a nível da área de Peritagens, mas sendo certificado e gratuito, esta é uma opção de faturação bastante atrativa para uma empresa de pequena dimensão. Ao contrário da versão ERP, esta não tem API para integração externa, logo não a podemos integrar com o nosso projeto de modo a esta cumprir com as necessidades de faturação.

SAGEONE

Mais uma vez sem elementos ao nível da área de Peritagens, mas só de faturação, SageOne [23] é uma solução económica do grupo Sage.

Este apenas permite a utilização por 1 utilizador em simultâneo e um máximo de 20 utilizadores, sendo esta uma limitação bastante inconveniente, visto apenas uma pessoa poder passar faturas na empresa ou ter . Ele contém funcionalidades de emissão de

documentos comerciais como faturas, bem como gestão de clientes e fornecedores, e a criação de relatórios sobre os ganhos e perdas. Este contém a certificação necessária para operar em Portugal, e um preço de 59.88€ por ano.

Este não pode ser integrado com o nosso projeto, porque não contém uma API que o permita, ao contrário da solução ERP.

PHCFX

O PHC FX [24] é uma solução económica do grupo PHC, só com faturação.

Este contém funcionalidades de faturação, contendo emissão de documentos comerciais, gestão de clientes e artigos, gestão de contas, e também emissão de mapas de faturação, bem como módulos de gestão de relações com clientes (CRM - Customer Relationship Manager), módulo de gestão de serviços que gere pedidos por parte de clientes, módulo de gestão de equipas, e um módulo de gestão documental. Este contém a certificação necessária para operar em Portugal, e um preço de 470,76€ por ano, pagando um extra de 16,80€ por cada utilizador a partir do primeiro.

Esta solução contém uma API para integração externa, logo poderia ser usada para cumprir com as necessidades de faturação do nosso projeto.

PROJECTO COLIBRI

O Projecto Colibri [25] é um software de faturação baseado em Java (mais uma vez sem ligação à área de Peritagens) com sede em Delães, Famalicão.

Este contém a certificação segundo a legislação, e funcionalidades como a emissão de documentos comerciais, gestão de stocks, gestão de produção, gestão de vendas, gestão de compras, gestão de contas correntes, registo de artigos e códigos de barras, criação de documentos com assinatura digital e a sua comunicação eletrónica. Este também permite a utilização em simultâneo por vários empregados.

Em relação ao preço, este custa 365€ por ano, e um extra de 180€ por cada atualização de versão. Caso seja necessário apoio telefónico devido a problemas, esta pode custar até 30€ por hora.

Este contém um esforço considerável no desenvolvimento de uma API para integração com outras soluções bastante documentado.

Este projeto também usa BIRT (Business Intelligence and Reporting Tools), um projeto de software open-source da Eclipse Foundation, que cria visualizações de dados e relatórios que podem embebedos em outras aplicações com Java. O BIRT pode ser usado em produtos comerciais gratuitamente, desde que o seu código fonte não seja alterado, segundo a Eclipse Public License [26].

INVOICEXPRESS

O InvoiceXpress [27] é um software de faturação (sem ligação à área de Peritagens) pertencente a uma empresa chamada Rupeal com base em Lisboa.

Este permite a emissão de documentos comerciais eletrónicos (em pdf), bem como a emissão e envio periódico de documentos recorrentes, a emissão de relatórios para consulta dos dados de faturação, permite o uso de outras moedas (sem ser só o euro), permite o envio automático dos documentos de faturação e transporte para a Autoridade Tributária, não sendo assim necessário o envio manual do ficheiro SAF-T(PT) por parte da empresa. Contém um sistema que emite documentos com referência multibanco, e que assinala automaticamente as faturas como pagas quando o cliente pagar através dessa

referência. Ele contém a certificação necessária para operar em Portugal.

Sobre o preço, este fica a 300€ por ano com um número máximo de 1000 documentos comerciais por mês, ou 450€ com um número máximo de 10000 documentos por mês. Para uma empresa do ramo os 1000 documentos por mês deverão ser suficientes.

Também têm uma limitação de um máximo de 8 utilizadores, mas o seu apoio é gratuito.

Contém uma API para integração com outras aplicações, podendo ser integrado com o nosso projeto.

DATAGEN

O Datagen [28] é um software de faturação simples e genérico, sem ligação à área de Peritagens.

Contém apenas as funcionalidades mais simples, como emissão de documentos comerciais, gestão de clientes, gestão de produtos e stocks, e emissão de relatórios sobre os dados de faturação, bem como a emissão do ficheiro SAF-T(PT).

Este tem um preço de 200€ por ano, sem outras limitações como pagamento extra por atualizações, e com apoio gratuito.

Este contém a certificação para operar em Portugal, mas não contém API para integração com outros programas externos.

WISEDAT

O WiseDat [29] é um software de faturação, sem ligação à área de Peritagens, pertencente a uma empresa com sede em Mira.

As suas funcionalidades são a emissão de documentos comerciais, gestão de clientes e fornecedores, gestão de produtos e stocks com códigos de barras, gestão de compras, vendas e pagamentos, consulta de variados relatórios e mapas de faturação. Os dados fiscais da empresa podem ser comunicados automaticamente à Autoridade Tributária, não necessitando o envio manual do ficheiro SAF-T(PT) por parte da empresa, podendo este ser emitido se necessário. Contém a certificação necessária para operar em Portugal.

O seu uso tem um preço de 100€ por ano com um posto de trabalho, mas um extra de 75€ pago pela instalação em cada posto de trabalho extra, e 35€ anuais por ano por cada posto de trabalho extra.

Não existe uma API para integração com outras aplicações externas.

KEYINVOICE

O KeyInvoice [30] é um software de faturação, sem ligação à área de Peritagens, pertencente a uma empresa com sedes em Lisboa, Madeira e Londres.

Este contém funcionalidades de emissão de documentos comerciais, controle de conta corrente de clientes e fornecedores, a possibilidade de uso de outras moedas sem ser o euro, o uso de referências multibanco nos seus documentos, gestão de artigos e stocks, e acesso via pc, tablet e smartphone. Também contém um sistema de gestão de relações com os clientes (CRM), e um sistema de gestão documental.

Também permite o envio dos dados fiscais da empresa automaticamente à Autoridade Tributária, não necessitando o envio manual do ficheiro SAF-T(PT) por parte da empresa, podendo este ser emitido se necessário. Contém a certificação necessária para operar em Portugal.

O seu preço é de 72€ por ano, com um número ilimitado de documentos e utilizadores.

Não existe uma API para integração com outras aplicações externas.

GESTIX

O Gestix [31] é um software de faturação, sem ligação à área de Peritagens, pertencente a uma empresa com sede em Lisboa.

As suas funcionalidades são a emissão de documentos comerciais, controle de custos de produção, rastreio de produtos por meio de número de série, gestão de clientes e relação com estes, uso de outras moedas sem ser o euro, gestão de orçamento para a empresa, trocas eletrónicas de documentos, automatização de opções de pagamento automático por referências multibanco ou outras opções, e opções de contabilidade.

Também permite o envio dos dados fiscais da empresa automaticamente à Autoridade Tributária, não necessitando o envio manual do ficheiro SAF-T(PT) por parte da empresa, podendo este ser emitido se necessário. Contêm a certificação necessária para operar em Portugal.

O preço é de 276€ por ano caso só seja preciso um utilizador em simultâneo, ou 292€ com 3 utilizadores em simultâneo, ou 476€ com um máximo de 10 utilizadores em simultâneo.

Esta solução contém uma API para integração com aplicações externas.

MOLONI

O Moloni [32] é um software de faturação, sem ligação à área de Peritagens, pertencente a uma empresa com sede em Lisboa.

Este permite a emissão de documentos de compra e venda, gestão de artigos e stocks, gestão de clientes e fornecedores, consultas de faturação e contas correntes, calendários de eventos e avenças, encomendas, compras e gestão documental. Pode ser utilizado por dispositivos móveis, e permite o envio automático dos dados fiscais da empresa automaticamente à Autoridade Tributária, não necessitando o envio manual do ficheiro SAF-T(PT) por parte da empresa, podendo este ser emitido se necessário. Contêm a certificação necessária para operar em Portugal.

O seu custo de utilização é 118,80€ por ano com 2 utilizadores em simultâneo, ou 178,80€ com um máximo de 5 utilizadores em simultâneo. O apoio técnico é gratuito, tal como as actualizações.

Esta solução contém uma API para integração com aplicações externas.

ECOMERCIO

O Ecomercio [33] é um software de faturação, sem ligação à área de Peritagens, pertencente a uma empresa com o nome MicroPlus com sede em Matosinhos.

As suas funcionalidades são a emissão de documentos de faturação (em formato PDF), a gestão de encomendas e stocks, gestão de clientes e fornecedores, a gestão de produtos e serviços, gestão da evolução do negócio, uma loja Online pronta a usar, compatibilidade com dispositivos móveis, bem como ferramentas integradas de Marketing utilizando o Facebook e Google. Também contém sistemas de apoio ao pagamento Multibanco, por transferência bancária, Paypal, Visa e Mastercard. Contêm a certificação necessária para operar em Portugal.

Contêm um custo de 100€ de ativação, e um custo anual de 350€. Caso sejam necessários mais de 10 utilizadores em simultâneo pode ser pago uma adição de 5€ por ano por cada utilizador extra. As actualizações e o suporte são gratuitos.

Não existe uma API para integração com outras aplicações externas.

FATURAVIRTUAL

A FaturaVirtual [34] é um software de faturação, sem ligação à área de Peritagens, pertencente a uma empresa com sede em Lisboa.

Contêm funcionalidades de emissão e gestão de faturas e outros documentos comerciais, gestão de clientes, produtos e serviços, emissão de relatórios variados, possibilidade de uso de outras moedas sem ser o euro, backups diários, agendamento de pagamentos, e possibilidade de acesso por dispositivos móveis.

Faz cópias de segurança diárias automaticamente, contém suporte gratuito por telefone e por email, e actualizações gratuitas.

Contêm a certificação necessária para operar em Portugal, incluindo a emissão do ficheiro SAF-T(PT).

O seu custo de utilização é 180€ por ano com 7 utilizadores em simultâneo, e com um número máximo de 500 documentos por mês. Também pode ser subscrito a 360€ por ano com 10 utilizadores em simultâneo, e um número ilimitado de documentos.

Não existe uma API para integração com outras aplicações externas.

EVARISTO

O Evaristo [35] é um software de faturação em Java, sem ligação à área de Peritagens, pertencente a uma empresa com sede em São João das Lampas, perto de Lisboa.

Inicialmente o Evaristo era um programa de faturação gratuito e de código aberto (antes da necessidade de certificação e dos requisitos extra que isso envolve), sendo que passou a ser comercializado com uma licença paga.

Contêm a emissão e gestão de documentos comerciais, a gestão de entidades como clientes e fornecedores, gestão de contas correntes, gestão de produtos, envio de documentos por email, e geração de variados relatórios de faturação como listas de compras, vendas, e mapas de IVA.

Contêm a certificação necessária para operar em Portugal, incluindo a emissão do ficheiro SAF-T(PT), mas não contém uma API para integração com outros programas externos.

Tem um custo de utilização um pouco elevado, 150€ pela instalação de um servidor, 50€ por cada posto de trabalho, e ainda custos anuais de 250€ pela licença, 150€ pela manutenção do servidor, e 50€ pela manutenção de cada cliente.

Logo com um posto de trabalho teremos um custo de instalação de 200€ e um custo anual de 450€. Cada posto de trabalho adicional adiciona 50€ de instalação e 50€ anuais.

O suporte também é pago à parte, a um custo de 50€ extra por 5 horas de suporte técnico, sendo as actualizações gratuitas.

Não existe uma API para integração com outras aplicações externas.

2.4. Alternativas a uma implementação de raiz (GnuCash)

Em vez de optar por implementar um sistema completamente de raiz, poderia ser usado um programa de faturação externo já certificado, e integrar este num sistema de informação criado por nós através de uma API. Poderíamos assim optar por uma solução

“Best-Of-Breed” onde fosse adotado um programa de faturação já existente, e juntássemos este a um sistema de informação específico para a área que apoiasse o funcionamento das empresas a gerir o seu processo de negócio da melhor maneira, de acordo com as necessidades do seu ramo de negócio.

Esta opção foi considerada não aceitável devido aos custos dos softwares de faturação encontrados que continham interfaces API para poderem ser integrados na nossa aplicação (custos podem ser consultados na tabela comparativa, nos produtos marcados com API para integração). O custo de usar um programa de faturação externo iria limitar a margem de lucro substancialmente.

Outra opção seria o uso de um programa de faturação não certificado gratuito e de código aberto, cuja licença permitisse a sua modificação e alteração de forma a usar no nosso projeto. Assim, seria preciso fazer as alterações ao programa necessárias de modo a este cumprir com todos os requisitos da certificação de software pela DGCI. Depois seria preciso fazer a integração deste programa com o resto do sistema de informação.

Foi encontrado um software que seria adequado para este efeito, o GNUCash [36].

A principal vantagem desta opção seria o uso gratuito do GNUCash, sendo apenas necessário realizar as alterações de modo a cumprir os requisitos de certificação. O GNUCash já contém um conjunto enorme de funcionalidades ao nível de faturação e Tesouraria, logo a nossa solução teria mais funcionalidades que uma solução implementada de raiz durante a duração deste estágio.

Existe, contudo, uma questão que impossibilita o uso desta solução devido a uma questão de licenças. O GNUCash usa a licença “GNU GPL” (“GNU General Public License”), que implica que trabalhos derivados só podem ser distribuídos nos mesmos termos. Isto implica que a versão certificada do GNUCash que nós iríamos produzir teria também de ser “GNU GPL”, e ser portanto gratuita e software aberto.

Outra desvantagem seria o facto de ser mais complicada a modificação de modo a tornar possível a certificação de software do que a implementação de raiz já tendo em conta as necessidades desta, visto que poderão existir problemas fundamentais na arquitetura que vão contra essas necessidades, e problemas de arquitetura são complicados de resolver depois da implementação já estar efetuada. Além disso a vantagem de o GNUCash já ter um conjunto enorme de funcionalidades é também uma desvantagem, pois aumenta a quantidade de funcionalidades que é preciso reescrever de modo ao software cumprir os requisitos de certificação.

2.5. Critérios de Comparação

Foram escolhidos alguns pontos comuns a várias soluções encontradas de modo a poder saber quais as mais valias de cada um, e qual o potencial destes enquanto produto concorrente.

Um dos principais pontos era a capacidade de concorrência a nível monetário, visto um produto demasiado caro estar num nível diferente do nosso, como seria o caso dos ERP. Outro ponto era a capacidade de usar essa solução para cumprir com a faturação do nosso projeto, logo se continha as ferramentas para isso na forma de uma API para integração. Também era importante a informação sobre se eram cumpridos os requisitos de certificação da legislação de modo a poderem operar em Portugal.

Um critério de comparação é o custo do uso do software, pois esta informação irá

automaticamente estar relacionada com a nossa margem de lucro potencial. Também é necessário analisar as restrições à utilização destes e custos adicionais.

O segundo parâmetro é se o programa tem funcionalidades de faturação, sendo que anteriormente ao falar de cada programa aparece um resumo destas funcionalidades. Sendo uma área de prestação de serviços não é necessário grande complexidade a nível de funcionalidades de faturação, basta poder passar faturas, recibos, ver relatórios com valores totais com e sem IVA, e exportar dados no ficheiro SAF-T(PT). Qualquer programa de faturação satisfaz as necessidades de uma empresa desta área.

O terceiro parâmetro indica se o programa tem funcionalidades de tesouraria. Estas funcionalidades de tesouraria implicam requisitos extra a nível da certificação em relação a programas só de faturação, logo muitos programas contêm faturação mas não tesouraria, ou vendem as funcionalidades de tesouraria como um pacote à parte. No nosso caso, as empresas da área de Peritagens de Seguros tendem a ter uma dimensão que implica que esta não contém um Técnico Oficial de Contas no quadro da empresa, que é quem precisa das funcionalidades de Tesouraria. As empresas normalmente contratam um serviço externo de Tesouraria, e estes contêm o seu próprio programa de Tesouraria, sendo que têm de importar os dados de faturação dos seus clientes. Esta importação hoje em dia é feita por meio do SAF-T(PT).

De seguida é indicado se o programa é certificado pela DGCI. Caso não seja teria de ser possível a sua modificação por nós de modo a cumprir os requisitos de certificação, e feita a certificação por nós.

O campo “CRM e Gestão de Stock” são duas características comuns a muitos programas, mas que não são fundamentais, logo em muitas soluções poderão não existir. Estas são mais valias, especialmente o CRM, uma vez que as empresas do ramo de Peritagens são de prestação de serviços, não tendo produtos físicos, mas sim serviços que prestam. Logo o CRM pode permitir a gestão dos seus clientes, de modo a otimizar o seu funcionamento.

A coluna “Solução específica para a área” indica se o programa tem já funcionalidades específicas para a área, de forma a permitir a gestão dos processos e modelo de negócio de uma empresa do ramo de Peritagens de Seguros. Um exemplo seria a capacidade de auxiliar a criação de avaliações de danos para um certo tipo de acidente, de maneira a importar posteriormente para um relatório estes dados, ou apoio na criação de um relatório de Peritagens e a inserção dos dados relativos a este.

Por fim a coluna “API para integração externa” indica se o programa fornece uma interface que torne possível a sua integração com outros projetos, incluindo o que vamos produzir. Caso fosse desejado integrar o nosso programa com uma solução de faturação, de modo a não termos de implementar os requisitos de faturação mas usar uma solução externa já certificada, este seria o nosso ponto de partida para a escolha de um parceiro.

2.6. Análise Comparativa

Nome Solução	Custo, restrições	Faturação	Tesouraria	Certificação	CRM e Gestão de Stocks	Solução Específica para a Área	API para Integração externa
CloudWorks	75€ por utilizador por mês	✓	✗	✓	✗	✓	✗
MRInformática	Não disponibilizada informação	✓	✗	✗	✗	✓	✗
SAP	Preço caso a caso, muito investimento para a implementação inicial	✓	✓	✓	✓	✓	✓
Primavera	Preço caso a caso, muito investimento para a implementação inicial	✓	✓	✓	✓	✗	✓
Primavera Express	Gratuito (max 30000€ de volume negócio anual)	✓	✗	✓	✓	✗	✗
SageERP	Preço caso a caso, muito investimento para a implementação inicial	✓	✓	✓	✓	✗	✓
SageOne	59.88€/ano, só 1 utilizador em simultâneo	✓	✓	✓	✗	✗	✗
Phc	Preço caso a caso, muito investimento para a implementação inicial	✓	✓	✓	✓	✗	✓
PhcFX	470€76/ano + 16€80 por utilizador extra(POS, gestão de produtos e stocks, loja online, etc)	✓	✗	✓	✓	✗	✓
Projecto Colibri	365/ano + 180€ por cada nova versão	✓	✗	✓	✓	✗	✓
InvoiceXpress	300€ até 1000 documentos/mês 450€/ano até 10000 documentos/mês	✓	✗	✓	✗	✗	✓
Datagen	200€/ano	✓	✗	✓	✓	✗	✗
Wisedat	100€/ano + 75€ pela instalação de cada posto + 35€/posto/ano	✓	✓	✓	✓	✗	✗
KeyInvoice	72€/ano	✓	✗	✓	✓	✗	✓
Gestix	276€/ano 1 utilizador 476€/ano 10 utilizadores	✓	✓	✓	✓	✗	✓
Moloni	118€80/ano 2 utilizadores 178€80/ano 5 utilizadores	✓	✗	✓	✓	✗	✓
Ecomercio	350€/ano + 100€ ativação (POS, gestão de produtos e stocks, loja online, etc)	✓	✗	✓	✓	✗	✗
FaturaVirtual	180€/ano max 500 docs 360€/ano docs ilimitados	✓	✗	✓	✗	✗	✗
Evaristo	450€/ano (licença e manutenção) + 200€ instalação, só 1 posto de trabalho, 50€ anuais e 50€ de instalação por cada posto de trabalho extra	✓	✗	✓	✗	✗	✗
Gnucash	Gratuito (Open Source)	✓	✓	✗	✗	✗	✓

Tabela 4: Análise Comparativa de várias soluções de faturação

2.7. Conclusão

Conforme pode ser verificado na tabela da Análise Comparativa, existem variadas soluções para faturação genéricas (e existem ainda muitas mais, conforme indicado na introdução deste capítulo), mas são praticamente inexistentes as soluções que contenham faturação integrada num sistema de gestão de negócio específico da área de Peritagem de Seguros.

O ERP SAP tem uma solução específica para esta área, que vai conter faturação integrada, mas o custo e esforço de implementação de um ERP vai fazer com que este não seja nosso concorrente direto, especialmente porque as empresas de Peritagens são normalmente de pequena dimensão.

Existe um concorrente direto na forma da solução Cloudworks, e este quando for lançado para o público em geral parece conter as mesmas funcionalidades que o nosso projeto, apesar de este ainda não estar no mercado. Conforme apresentado, o seu preço também não é muito elevado, desde que não seja pedida uma licença para muitos utilizadores (75€ por utilizador por mês).

Foram analisados os requisitos técnicos para a certificação de software de faturação em Portugal, de modo a que esta informação possa ser considerada e tomada em conta durante a Especificação dos Requisitos do nosso projeto, na fase seguinte.

Existe bastante complexidade acrescentada por esta certificação em relação ao período antes de esta existir, mas também veio abrir oportunidades de negócios, na medida em que alguns produtos que eram usados tiveram de ser abandonados, por estarem descontinuados, e o seu produtor não os ir atualizar de modo a cumprirem os requisitos de certificação. Logo as empresas que os utilizavam viram-se forçadas a procurar novas soluções.

3. Metodologia de Desenvolvimento

A metodologia de desenvolvimento escolhida é extremamente importante para um projeto, sendo um fator de considerável relevância em relação às suas probabilidades de sucesso. Existem variadas metodologias, e a metodologia escolhida deve ter uma relação clara com as necessidades de cada projeto, sendo que existem vários fatores a considerar para a escolha desta.

No nosso caso deste projeto foi considerada a estratégia de desenvolvimento tradicional, chamada método de Cascata[37] (Waterfall Method) e foi considerada um dos métodos de desenvolvimento ágil, o Scrum[38]. Foram escolhidos estes métodos porque eram métodos com os quais o estagiário já era familiar, e sendo que o Scrum foi escolhido de forma a representar a utilização dos métodos ágeis, em vez do tradicional método em Cascata. Ao considerar mais do que uma opção tentou-se chegar a uma solução mais adequada a este caso em específico.

3.1. Método em Cascata

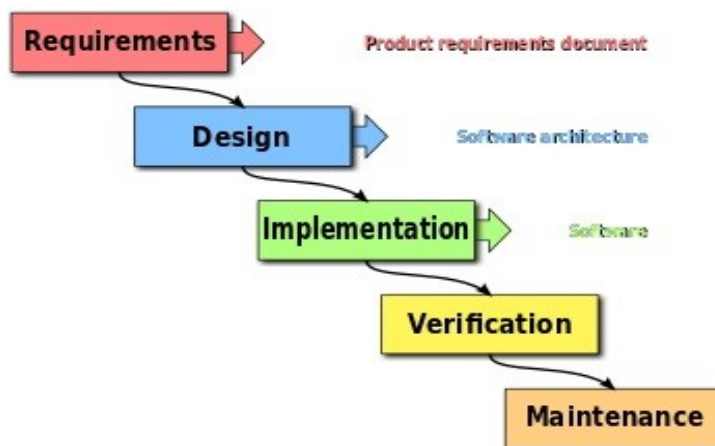


Figura 5: O processo do Método em Cascata (fonte: wikipédia[37])

O método em Cascata envolve um processo sequencial com fases rígidas e bem definidas, que ocorrem sempre pela mesma ordem, onde o progresso pode ser visto a fluir como uma cascata (esta é a origem do nome), começando por uma fase de Análise de Requisitos, e seguindo por fases de Design e arquitetura, Implementação, Testes e Verificação, Instalação e Manutenção. Devido à rigidez das fases, e o esquema sequencial, esta metodologia é bastante simples de utilizar e de gerir.

A 1ª fase é a Análise de Requisitos e envolve a documentação dos requisitos de forma a definir as necessidades a que o projeto necessita de responder, e o que este irá conter a nível de funcionalidades. Esta fase irá tentar que o projeto responda corretamente ao que o cliente necessita. De seguida a 2ª fase trata do Design e Arquitetura, onde são estudados os requisitos, e são especificados os requisitos de hardware e software necessários bem como a definição da arquitetura do sistema e a comunicação e integração das várias partes deste. A fase seguinte é a implementação, onde o projeto é dividido em partes mais pequenas e estas são implementadas e integradas. O próximo passo é a fase de Verificação e Testes, onde são feitos os testes de modo a minimizar o número de erros existentes no projeto, e a aceitação do projeto por parte do cliente que o vai utilizar. A última fase vai corresponder à Instalação e Manutenção, que envolve a instalação do projeto para uso do(s) cliente(s) e a manutenção de modo a garantir que o projeto

continue a funcionar corretamente sem problemas, e a responder às necessidades esperadas dele.

3.2. Método SCRUM (desenvolvimento ágil)

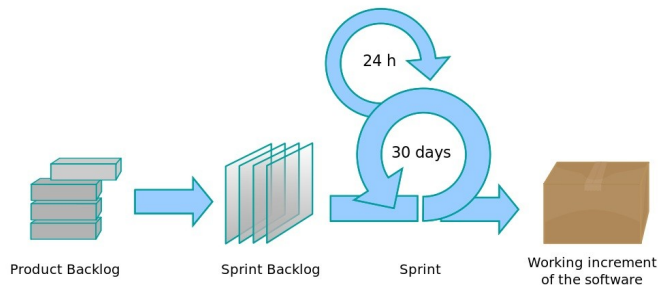


Figura 6: O processo do Método SCRUM (fonte: wikipédia[38])

O método Scrum é um método de desenvolvimento ágil que funciona de maneira iterativa e incremental. Alguns dos princípios fundamentais do Scrum são a assunção de que os requisitos podem mudar a qualquer altura, e a assunção de que o problema não pode ser completamente definido inicialmente, logo foca-se em maximizar a capacidade de desenvolver pequenas partes (incrementos) que são integradas e entregues em períodos pequenos de tempo, em vez de ser desenvolvido logo o projeto como um todo. Isto permite responder rapidamente à alteração de requisitos. O Scrum envolve a criação de um artefacto chamado Product Backlog que contém a lista de requisitos e que está em constante evolução ao longo do projeto. Elementos deste product backlog são selecionados para implementação numa fase chamada sprint, que normalmente dura 1 mês e envolve reuniões diárias com a equipa, e é produzido no fim do Sprint um incremento ao projeto que pode ser integrado no projeto, e pode ser validado com o cliente e outras partes interessadas.

3.3. Metodologia escolhida

No nosso caso escolhemos usar o Método de Desenvolvimento em Cascata.

Uma das razões principais foi o facto de não ser expectável a existência de alterações aos objetivos e funcionalidades ao longo do projeto, sendo possível efetuar uma especificação completa e detalhada logo desde o início. Este fator aponta claramente para o método em Cascata, visto o SCRUM estar pensado para o caso oposto, em que os requisitos são difíceis de definir e estão em constante mudança. Outra razão principal foi o tamanho da equipa, constituída por 2 elementos, sendo fácil a sincronização entre estes. Outra razão principal foi a necessidade de não interferir com o trabalho do cliente (a entidade auxiliar que colaborou no processo de análise de requisitos), e portanto minimizar a necessidade de contacto constante com essa entidade. Mais uma vez se nota a preferência para o método em Cascata, pois as metodologias ágeis (entre elas o SCRUM) necessitam de acesso frequente ao cliente e outras partes interessadas, de modo a obter ao longo do projeto informação acerca de como está a decorrer o progresso, e se corresponde ao esperado.

No caso de um estágio este está dividido por semestres, em que no primeiro semestre são esperados os conteúdos correspondentes às fases iniciais do Método em Cascata (Requisitos e Design e Arquitetura), e o segundo semestre corresponde normalmente às fases seguintes (Implementação e Verificação) podemos verificar que este se adapta melhor que o SCRUM, se bem que não existiria problema em usar essa metodologia.

4. Análise de Requisitos

A fase de Análise de Requisitos é de extrema importância para o sucesso de um projeto. É nesta fase que vamos identificar quais as questões que o sistema precisa de resolver, e quais as funcionalidades necessárias para este ter o máximo de valor possível para o seu público alvo.

Nesta fase vamos tentar que o sistema que estamos a conceber seja o sistema que os clientes precisam, em vez de criar algo que realmente não resolve os seus problemas e necessidades.

A interação com o cliente e a sua participação nesta fase é portanto desejável e importante, de modo a que o sistema esteja de acordo com as suas necessidades, e que a sua forma de utilização seja acessível, de modo a que o cliente o consiga utilizar de maneira eficaz e intuitiva na medida do possível.

Os requisitos devem ser bem documentados, e serem mensuráveis, verificáveis, rastreáveis, relacionados com os objetivos de negócio ou oportunidades, e descritos com um nível de detalhe suficiente para o sistema em questão [39]. Neste estágio os requisitos foram documentados na forma de casos de uso.

A especificação de requisitos também vai ajudar a perceber o que é necessário fazer, e ter uma ideia do esforço necessário para isso.

Sendo assim esta fase vai ser a base de trabalho para o projeto inteiro, e requisitos importantes por identificar, requisitos mal identificados ou mal detalhados, ou outras falhas nesta fase irão influenciar negativamente o projeto inteiro, podendo aumentar imenso o tempo necessário para desenvolver o projeto com sucesso.

Este capítulo começa com esta introdução, que visa indicar qual a importância da fase de análise de requisitos. De seguida é explicado o processo seguido no levantamento e análise de requisitos deste estágio. Lá são indicados os passos seguidos, a forma de formalização e representação, e como foi articulado com a empresa esta fase.

O próximo ponto vai apresentar um resumo do núcleo de funcionamento do programa, no qual está inserido o sistema de autenticação que garante o acesso ao sistema só depois de uma autenticação correta, bem como o sistema de navegação dentro do programa.

De seguida passamos a explicar o módulo que gere a identificação de contactos, que podem ser contactos individuais (pessoas) ou contactos de empresa. Estes contactos são depois utilizados no módulo de gestão de processos e no módulo de faturação.

O módulo seguinte a ser explicado é o módulo de faturação, e é explicado como se faz pesquisa nos variados tipos de documentos comerciais, uma breve explicação do que consiste um documento comercial, e um resumo das funcionalidades do módulo.

Podemos depois encontrar a lista total de casos de uso, a prioridade de cada um, e uma breve explicação sobre estes. É também apresentado um diagrama de casos de uso que têm como intenção mostrar graficamente a relação entre alguns casos de uso, bem como alguma informação sobre níveis de administração do programa.

Por fim é efetuada uma conclusão que faz uma breve reflexão sobre o capítulo.

4.1. Levantamento de Requisitos

O levantamento foi feito realizando entrevistas com uma empresa do ramo (RESI, apresentada anteriormente no capítulo de estado da arte), que especificou como funciona atualmente o seu trabalho, e os detalhes do seu modelo de negócio e fluxo de trabalho. As entrevistas foram realizadas com o seu dono, com a secretária que trata da documentação como emissão de faturas e recibos, e com dois peritos, a quem são atribuídos processos para emissão de relatórios sobre estes.

A empresa já continha um programa (não certificado) que usa para guardar os dados da empresa e fazer a gestão dos seus processos e dados destes, e outro para efeitos de faturação.

O programa usado para fazer a gestão processual chama-se Peritagens, e está descontinuado, sendo que a empresa indicou que o produtor da aplicação já nem pode ser contactado, tendo deixado de exercer atividade. Esta aplicação foi feita para Windows XP, e tem graves problemas de compatibilidade com sistemas mais recentes, e o seu instalador não fornece algumas dependências necessárias, sendo que não é possível instalar em computadores novos sem a aplicação perder algumas das suas funcionalidades. Este também funciona com uma base de dados em Microsoft Access, que além de ser pouco seguro, tem tendências a ficar com a base de dados corrupta, e perder alguma informação, tendo de restaurar a base de dados através de cópias de segurança, e voltar a repor os dados que faltarem. Este programa também contém funcionalidades de faturação e tesouraria, mas devido a estar descontinuado não tem a certificação necessária a software de faturação (nem cumpre os requisitos), logo a empresa já não pode usar este programa para esse efeito.

O software certificado usado para faturação é o Datagen, que faz parte do capítulo de estado da arte como sendo um dos softwares analisados. Este não contém funcionalidades a nível de tesouraria, mas tal não é necessário visto a empresa delegar a um contabilista externo que trata da contabilidade. Este apenas precisa dos dados dos documentos (faturas e recibos), e trata da contabilidade nos seus próprios programas. Esses dados podem ser importados a partir do ficheiro SAF-T(PT).

A empresa indicou que não era desejável usarem um programa para gerir os processos e outro para gerir as questões de faturação, e que os problemas do programa de gestão de processos já discutidos em cima lhes motive o interesse em usar um outro programa que cumpra as suas necessidades.

Foi procedido então à recolha de requisitos junto da empresa, de modo a identificar as funcionalidades que eram necessárias ao nosso projeto como um todo, e caso deste estágio, do módulo de faturação. O outro estagiário tratou de recolher também as funcionalidades gerais, e as necessárias à parte dele, relativas a gestão de processos.

Estes requisitos foram enumerados, e foram atribuídas prioridades a cada um. Foram recolhidos os dados que os estagiários acharam necessários para a documentação dos requisitos. Posteriormente estes requisitos foram desenvolvidos e formalizados em casos de uso, de modo a preencher para cada um uma tabela de casos de uso (que pode ser consultada no Anexo A - Documento de Casos de Uso). À medida que novas questões surgiram no desenvolvimento dos casos de uso, estas questões foram apontadas, e foram realizados contactos pontuais de forma a esclarecer detalhes dos requisitos de acordo com as necessidades.

Na maioria das tabelas de casos de uso pode ser encontrado um protótipo (mockup) que ilustra a interface do nosso projeto que vai fazer a interação com o utilizador, de modo a

facilitar a perceção de como irá funcionar cada funcionalidade.

A cada caso de uso foi definido uma prioridade, de modo a que a ordem de implementação siga uma ordem de importância, para as principais funcionalidades poderem ficar prontas, e poderem ser validadas com o cliente o mais rápido possível. Ao mesmo tempo é minorado o risco de chegar à data de fim do projeto e não existir um projeto minimamente viável, porque requisitos secundários foram implementados, e requisitos que eram absolutamente necessários para o projeto poder ser utilizado pela empresa não chegaram a ser acabados.

A prioridade definida para cada caso de uso foi determinada segundo o Método “MoSCoW”[40][41], que é um método utilizado em projetos de desenvolvimento de software para chegar a um entendimento com os vários intervenientes sobre a importância de cada requisito. Neste método por ordem da maior prioridade para a menor temos os valores M (Must Have), S (Should Have), C (Could Have), W (Won't Have).

Este documento só irá conter a lista de casos de uso, sendo que as tabelas e as imagens para caso de uso podem ser consultados em detalhe no Anexo A – Documento de Casos de Uso.

4.2. Núcleo Principal

A aplicação vai estar dividida por módulos, de modo a poderem ser inseridos módulos que introduzam novas funcionalidades de forma eficaz, tentando reduzir dependências que impliquem a alteração dos módulos já existentes de modo a manter a coesão do sistema. Por exemplo este projeto inicialmente não irá conter módulo de tesouraria, mas eventualmente poderá ser adicionado um novo módulo contendo estas funcionalidades.

Os módulos planeados até ao momento estão divididos pelos 2 estagiários a trabalhar neste projeto.

O estagiário A (Carlos Freixo) a trabalhar na parte de gestão de processos está responsável por um módulo de mensagens e alertas, que permite a comunicação e passagem de informação entre utilizadores dentro do sistema, bem como um sistema de alerta de prazos e tarefas para cada utilizador. Outro módulo para este é o módulo de gestão de processos, que permite guardar os dados de processos, guardar imagens e informação associada a cada processo, criar e armazenar relatórios para cada processo. Os clientes da empresa de seguros poderão ter uma autenticação limitada, com credenciais que lhes permitam entrar no sistema e aceder apenas a dados dos processos nos quais eles são intervenientes. Assim estes poderão assim através do sistema obter os relatórios diretamente do sistema, sem ter a empresa de seguros de mandar estes por email ou outra forma.

O estagiário B (Marco Pedrosa) a que se refere este relatório está encarregue do módulo de faturação, que trata da criação e gestão de documentos comerciais, e da visualização de variadas informações de faturação necessárias ao funcionamento da empresa.

Vai existir um módulo de contactos, que irá conter a informação relativa a pessoas individuais e a empresas, e esta informação é necessária tanto no módulo de faturação para identificar os clientes dos documentos, como nos módulos em cima referidos para o outro estagiário para identificar os clientes dos processos, logo este irá ser da responsabilidade de ambos.

O sistema de autenticação e gestão de utilizadores também irá ser da responsabilidade de

ambos os estagiários, sendo criado em conjunto, e existirá configurações de apoio na parte de definições, que estará dividido em configurações de módulo, da responsabilidade do autor do módulo e definições gerais, da responsabilidade de ambos os estagiários.

4.3. Casos de Uso

Na tabela seguinte podemos ver a lista de todos os casos de uso encontrados na análise de requisitos, bem como qual é a prioridade atribuída a cada um. Esta pode ser uma informação para definir a ordem de implementação, bem como quais as funcionalidades a retirar no caso do calendário estar atrasado, e no tempo alocado para o projeto não ser possível implementar todas as funcionalidades.

A Lista não está ordenada por ordem de prioridade, mas sim por área funcional do projeto, agrupando primeiro os casos de uso relativos à gestão de utilizadores (UC001 a UC009), depois casos de uso relativos a gestão de documentos de faturação (UC010 a UC017), de seguida casos de uso relativos a listagens de faturação (UC018 a UC020), contendo no fim casos de usos que configuram outros elementos necessários para o funcionamento do programa, especialmente na parte de faturação (UC021 a UC034).

A seguir à tabela com a Lista de Casos de Uso existe uma breve descrição de cada um, mas o Anexo A – Documento de Casos de Uso irá conter a descrição detalhada de cada caso de uso, e imagens do protótipo que representa a interface de comunicação com o utilizador.

ID do Caso de Uso	Nome do Caso de Uso	Prioridade
Gestão de Utilizadores		
UC001	Autenticação de Utilizador	M (Must Have)
UC002	Terminar Sessão de Utilizador	M (Must Have)
UC003	Recuperação de Palavra-Chave de Utilizador	M (Must Have)
UC004	Mudança de dados de Utilizador	M (Must Have)
UC005	Criação de novo Utilizador	M (Must Have)
UC006	Mudança das permissões de um Utilizador	M (Must Have)
UC007	Criação de Nova Empresa	C (Could Have)
UC008	Mudança dos Dados Gerais de uma Empresa	M (Must Have)
UC009	Mudança dos Módulos disponíveis para uma Empresa	C (Could Have)
Gestão de Documentos de Faturação		
UC010	Inserir Novo Documento Comercial	M (Must Have)
UC011	Procurar Documento Comercial	M (Must Have)
UC012	Ver Lista de Últimos Documento Comercial Emitidos	S (Should Have)
UC013	Ver Documentos em Aberto (por pagar e/ou receber total ou parcialmente)	S (Should Have)
UC014	Consultar Documento Comercial	M (Must Have)
UC015	Imprimir Documento Comercial	M (Must Have)
UC016	Anular Documento Comercial	M (Must Have)
UC017	Emitir Documento a partir do Atual	S (Should Have)
Listagens de Faturação		
UC018	Ver Lista de Vendas Líquidas	C (Could Have)
UC019	Ver Volume de Faturação	C (Could Have)
UC020	Ver Taxas de Iva para um Período	C (Could Have)

Outros elementos relativos à faturação		
UC021	Exportar ficheiro SAF-T(PT)	M (Must Have)
UC022	Inserir/Modificar produto	M (Must Have)
UC023	Procurar Contactos Individuais	M (Must Have)
UC024	Ver Contacto Individual	M (Must Have)
UC025	Adicionar/Editar Contacto Individual	M (Must Have)
UC026	Procurar Contactos de Empresas	M (Must Have)
UC027	Ver Contacto de Empresa	M (Must Have)
UC028	Adicionar/Editar Contacto de Empresa	M (Must Have)
UC029	Apagar Contacto (Individual ou de Empresa)	M (Must Have)
UC030	Procurar Impostos	M (Must Have)
UC031	Adicionar/Editar Imposto	M (Must Have)
UC032	Apagar Imposto	S (Should Have)
UC033	Alteração dos dados do programa	S (Should Have)

Tabela 5: Lista total de Casos de Uso e sua priorização

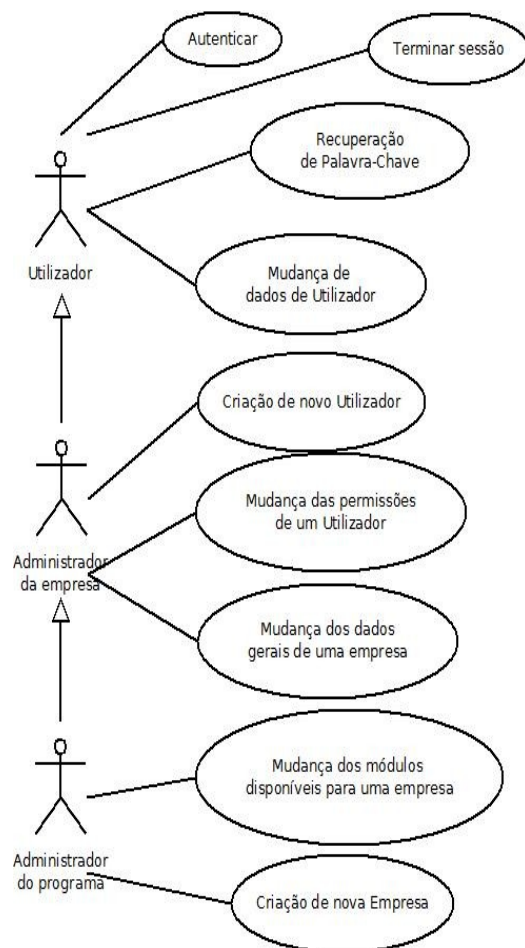


Figura 7: Diagrama de Casos de Uso:

Gestão de Utilizadores

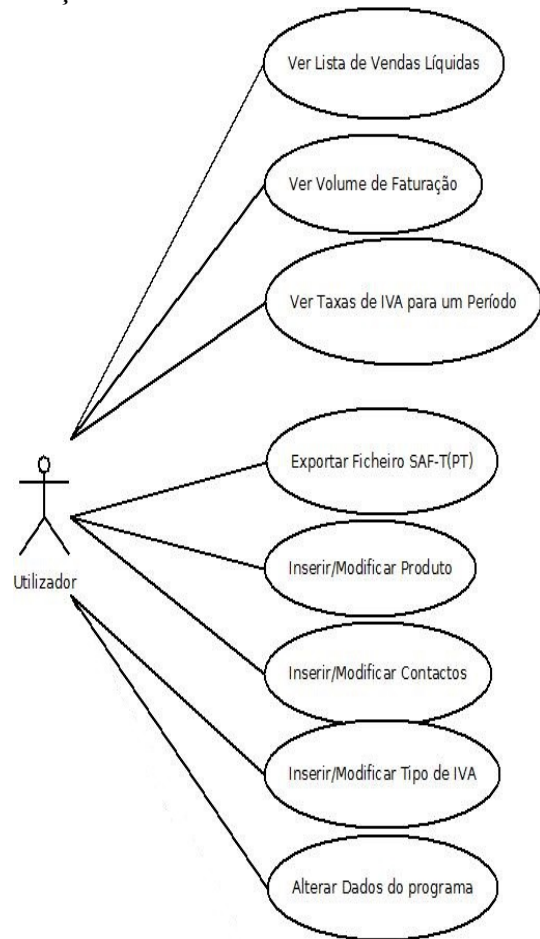


Figura 8: Diagrama de Casos de Uso:

Outros

No diagrama de caso de uso relativo à gestão de utilizadores podemos facilmente ver que existem 3 níveis de utilizadores base. O primeiro é o Administrador de empresa, que vai poder criar novas empresas (que serão os clientes do nosso programa, não confundir com

os contactos de empresa que cada empresa contém). Ao criar a empresa este pode configurar quais os módulos a quais esta tem acesso. Isto quer dizer que uma empresa poderia querer só a parte de gestão de processos e não lhe interessar a faturação, logo o programa poderia ser vendido só com a gestão de processos ativado. O Administrador do programa também pode configurar utilizadores para cada empresa, sendo o mais relevante a configuração do administrador de empresa inicial, que poderá depois adicionar novos utilizadores à sua empresa conforme desejar. O Administrador do Programa também terá acesso a todas as funcionalidades a que o Administrador de Empresa tem acesso.

O Administrador de Empresa pertence a uma empresa que está a utilizar o programa, e pode configurar os utilizadores da sua empresa, adicionar novos, e restringir as funcionalidades a que cada grupo de utilizador da sua empresa tem acesso, bem como atualizar os dados da sua empresa. O administrador de Empresa tem acesso a todas as funcionalidades de um utilizador normal.

O Utilizador é o nível mais básico, pertence a um grupo de utilizadores segundo o definido pelo Administrador de Empresa, e o grupo de utilizador indica as funcionalidades a que este tem acesso.

No caso de uso: Outros podemos ver variadas funcionalidades relativas a faturação que não são relativas a um documento comercial único, mas listagens e mapas de dados, exportação do ficheiro SAF-T(PT) e também configuração de tabelas de apoio como a listagem de tipos de imposto e de produtos, e as configurações do programa.

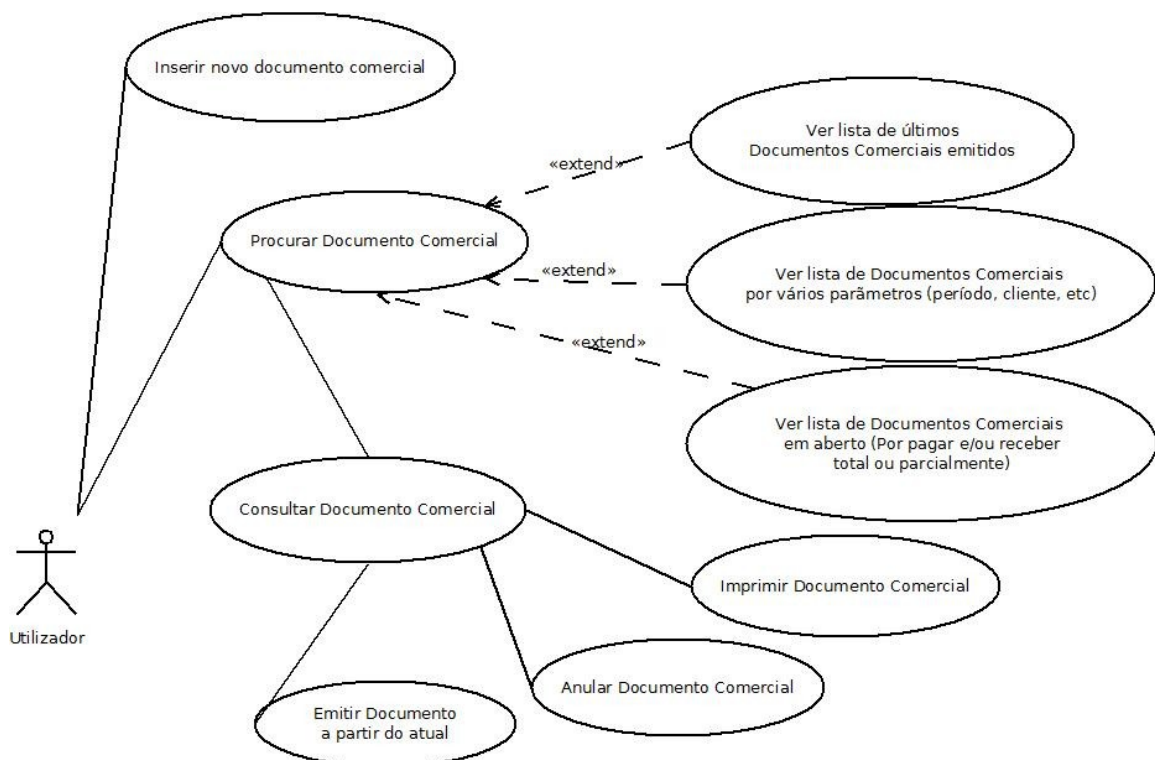


Figura 9: Diagrama de Casos de Uso: Documentos Comerciais

No diagrama de caso de uso relativo aos documentos comerciais podemos ver como estes casos de uso se relacionam entre si. Basicamente podem ser inseridos novos documentos, mas estes não podem ser modificados ou apagados, apenas anulados. A procura de documentos pode ser efetuada de diferentes maneiras, e ao selecionar um documento

podemos ver os seus detalhes, e efetuar ações como a sua impressão, a sua anulação e usar os dados de um documento (cliente, processo, produtos/serviços) para criar um novo documento com esses dados já preenchidos. Isto é de bastante importância porque por exemplo um recibo é normalmente feito com os mesmos dados que a fatura equivalente, a não ser que sejam usados vários recibos para cada fatura, ou fazer usar o mesmo recibo para várias faturas. Logo os dados da fatura podem (e devem) ser usados para preenchimento automático do recibo, quando for a altura de passar recibo (que simboliza o pagamento da fatura).

UC001 - AUTENTICAÇÃO DE UTILIZADOR

Este caso de uso representa a autenticação de um utilizador no sistema. Este já deve ter uma conta registada e ativada. Caso utilizador não esteja autenticado não pode aceder à maioria do conteúdo do sistema.

UC002 – TERMINAR SESSÃO DE UTILIZADOR

Este caso de uso representa um utilizador a terminar a sua sessão no sistema. Isto cancela a sua autenticação e faz com que deixe de ser possível aceder a páginas que não sejam públicas. Para voltar a poder aceder deverá autenticar-se outra vez.

UC003 – RECUPERAÇÃO DE PALAVRA-CHAVE DE UTILIZADOR

Este caso de uso representa a recuperação da Palavra-Chave de um Utilizador. Isto vai envolver um processo que envia um email para o endereço de Email já configurado nos dados do utilizador com um endereço que ao ser seguido permitirá a mudança da Palavra-Chave para um novo valor. Este endereço só estará válido durante 24 horas.

UC004 – MUDANÇA DE DADOS DE UTILIZADOR

Este caso de uso representa um utilizador a mudar os seus dados pessoais. Estes dados são o nome, morada, endereço de email e contactos como telemóvel, opcionalmente a sua foto, bem como nível de permissões para esse utilizador (se é administrador da empresa, e quais as opções do sistema às quais este tem acesso).

Estes dados também poderão ser alterados com a mesma interface por um administrador do programa ou administrador da empresa. (O administrador do programa pode gerir todos os utilizadores do programa inteiro, e o administrador da empresa pode apenas gerir os utilizadores da sua empresa, cada utilizador pode mudar os seus próprios dados).

UC005 – CRIAÇÃO DE NOVO UTILIZADOR

Este caso de uso representa um Administrador a criar um novo utilizador. Pode ser um administrador geral do sistema que vai adicionar um utilizador novo a uma empresa específica, ou ser um administrador de empresa a adicionar um utilizador novo à sua empresa.

Ao adicionar um utilizador devem ser preenchidos o máximo possível dos seus dados pessoais. Estes dados são o nome, morada, endereço de email e contactos como telemóvel, opcionalmente a sua foto, bem como nível de permissões para esse utilizador. (se é administrador da empresa, e quais as opções do sistema às quais este tem acesso).

UC006 – MUDANÇA DAS PERMISSÕES DE UM UTILIZADOR

Este caso de uso representa um Administrador a mudar as permissões de um utilizador através de um sistema de níveis de acesso que controlam os acessos de um grupo de utilizadores a várias funcionalidades (vários utilizadores podem partilhar este nível de acesso, logo mudamos as permissões para todos os utilizadores com esse nível de acesso). Na página de alteração dos dados de cada utilizador pode ser configurado o nível de acesso ao qual este pertence.

Pode ser um administrador geral do sistema que vai mudar as permissões de um utilizador de uma empresa específica, ou ser um administrador de empresa a mudar as permissões de um utilizador da sua empresa.

Não devem ser possível atribuir permissões que o utilizador que esteja a fazer esta alteração não possua (um administrador de empresa não pode atribuir permissão de administrador de sistema a um utilizador).

UC007 – CRIAÇÃO DE NOVA EMPRESA

Este caso de uso representa um Administrador do sistema a criar uma nova empresa.

O sistema deve criar bases de dados e/ou tabelas próprias para essa empresa, e devem ser preenchidos os dados iniciais da empresa (que podem ser alterados posteriormente pelo administrador da empresa).

UC008 – MUDANÇA DOS DADOS GERAIS DE UMA EMPRESA

Este caso de uso representa um Administrador que vai atualizar os dados gerais da empresa.

Estes dados são usado para os cabeçalhos de alguns documentos, bem como para preenchimentos dos dados da empresa no cabeçalho do ficheiro SAF-T(PT). Os dados podem já ter sido preenchidos na altura da criação da conta da empresa no programa (UC007).

Estes dados são o nome da empresa, o NIF da empresa, a morada, código postal e localidade da empresa, nº de telefone e nº de fax da empresa.

UC009 – MUDANÇA DOS MÓDULOS DISPONÍVEIS PARA UMA EMPRESA

Este caso de uso representa um Administrador que vai mudar o acesso de uma empresa a módulos que constituem partes do sistema. Pode ser adicionado o acesso a novos módulos, bem como retirar o acesso a módulos a que a empresa tivesse acesso.

Esta funcionalidade poderá suportar um sistema de negócio em que módulos sejam vendidos em separado, e cada empresa apenas precise de comprar os módulos que lhe interessarem.

Notam-se semelhanças ao mecanismo de seleção de funcionalidades disponíveis a cada nível de acesso para os utilizadores, mas aqui é selecionado se a empresa ou não tem acesso a cada módulo. Se a empresa não tiver acesso ao módulo (ou este acesso for cancelado), todos os níveis de acesso da empresa perdem o acesso às funcionalidades do módulo.

UC010 – INSERIR NOVO DOCUMENTO COMERCIAL

Este caso de uso representa um utilizador que irá inserir um novo documento comercial. Este documento poderá ter vários tipos, incluindo Fatura-recibo, Fatura, Recibo, Nota de Crédito, Nota de Débito, Orçamento, Guia de Transporte.

Deverão ser preenchidos dados relativos ao documento, e depois deverão ser adicionadas linhas relativas aos produtos e/ou serviços prestados que irão constar do documento.

Sempre que necessário deve ser indicado relações com outros documentos para o documento inteiro e/ou em cada uma de suas linhas individualmente, bem como ser indicado motivo de isenção linhas nas quais haja isenção de IVA.

Também existe possibilidade de definir um processo associado ao documento (informação sem relevância fiscal), e o documento poderá indicar qual o processo da empresa a que este documento se refere.

No Protótipo relativo à consulta de documentos podem ser consultados os dados relativos ao transporte de mercadorias, que são inseridos também neste caso de uso, sendo estes a identificação do veículo, observações, dados do local de origem e dados do local de destino. Tanto o local de origem como destino são identificados por um nome, morada,

código postal e país. Tal como as outras janelas, a inserção destes dados é extremamente semelhante a essa janela de visualização.

UC011 – PROCURAR DOCUMENTO COMERCIAL

Este caso de uso representa um utilizador que irá procurar por um determinado documento comercial. Existem vários campos pelos quais o documento pode ser procurado, por exemplo todos os documentos relacionados com certo processo, ou um certo cliente, ou por uma certa data de emissão, ou contendo um determinado produto. No fim existem campos não editáveis onde é possível consultar totais para todos os documentos encontrados, desde o número total de documentos, valor total de crédito com IVA, valor total de débito com IVA, e total pago. O valor total de crédito com IVA corresponde à soma dos valores dos documentos cujo tipo é nota de crédito, que indica o total faturado que foi cancelado por meio de notas de crédito. O valor total de débito corresponde à soma dos valores dos outros tipos de documento. O total pago indica o valor que já foi pago, porque é possível que o valor seja faturado, mas ainda não ter sido efetuado o pagamento.

UC012 – VER LISTA DE ÚLTIMOS DOCUMENTOS COMERCIAIS EMITIDOS

Este caso de uso representa um utilizador que irá pesquisar quais foram os documentos comerciais emitidos. Estes aparecem ordenados por data de emissão, e mostram vários dados como a sua identificação (que diz o número do processo, o seu tipo, e a sua série), a data de emissão, qual o seu cliente, qual o valor total, se está anulado ou não. Este caso de uso seria igual ao caso de uso UC011 sem restrições e com o resultado a aparecer ordenado por data de emissão, sendo simplesmente um atalho para não ter de ser configuradas as opções de procura a nível de restrições. As restrições podem depois ser alteradas e acrescentadas de modo a criar uma nova pesquisa baseada na pesquisa base deste caso de uso.

UC013 – VER DOCUMENTOS EM ABERTO (POR PAGAR E/OU RECEBER TOTAL OU PARCIALMENTE)

Este caso de uso representa um utilizador que irá pesquisar quais os documentos comerciais em aberto, que representam documentos sobre os quais precisa de ser feito pagamento ou recebimento. É possível parametrizar de modo a só procurar os documentos em aberto em relação a um cliente específico. Estes aparecem ordenados por data de emissão, e mostram vários dados como a sua identificação (que diz o número do processo, o seu tipo, e a sua série), a data de emissão, qual o seu cliente, qual o valor total, quais o valor já regularizado, se está anulado ou não.

Tal como o anterior, este caso de uso representa um atalho para uma procura de documentos com certas definições de procura que serão utilizadas frequentemente, não tendo que selecionar os campos de procura constantemente. Essas definições podem no entanto ser alteradas de modo a refinar a procura, tal como acontece no caso de uso de procura de documentos (UC011).

UC014 – CONSULTAR DOCUMENTO COMERCIAL

Este caso de uso representa um utilizador a consultar os dados de um documento em específico. O documento foi selecionado a partir de uma das variadas páginas de pesquisa de documentos existentes no programa. Esta página mostra vários dados acerca do documento. Os dados são apresentados de forma semelhante a quando foi preenchido o documento no UC010, mas agora os dados não podem ser alterados, apenas visualizados.

Conforme acontecia na inserção existe uma janela para apresentar os dados gerais, com a identificação do documento, dados do cliente e dados do processo associado (quando existir). Existe outra janela com os dados de transporte, que consistem na identificação

do veículo, dados do local de origem e dados do local de destino. Existe uma janela com os dados das linhas de produtos e serviços do documento, bem como os totais de valores, e por último uma janela relativa a dados de pagamento.

A partir desta janela o utilizador pode também imprimir o documento usando um botão para o efeito, pode anular o documento, ou pode criar um novo documento relacionado com este (o novo documento terá as suas linhas de produto e variadas outras informações preenchidas automaticamente com os dados existentes neste documento de origem, e as linhas de produto terão automaticamente a referência para este documento original).

Em relação à janela referente aos dados de pagamento do documento, esta vai permitir inserir e alterar os dados relativos ao pagamento. Normalmente esses dados são colocados automaticamente quando fazemos o recibo de uma fatura, colocando o pagamento como sendo em numerário por defeito. Caso o tipo de documento seja fatura-recibo em vez de fatura, como este não é pago emitindo um outro documento (emitir o recibo para pagar a fatura) temos de inserir os dados de pagamento aqui na página de consulta do documento manualmente. No caso de existir um recibo, os dados do recibo são inseridos automaticamente como pagamento nas faturas que lhe deram origem, usando as referências existentes nas linhas de produtos/serviços. Pode existir pagamento parcial do documento, e fazer com que este seja pago por mais que um recibo, logo esta janela tem isso em conta, podendo haver mais do que um pagamento. Certos documentos não irão conter dados de pagamento, como será o caso de guias de transporte e orçamentos.

UC015 – IMPRIMIR DOCUMENTO COMERCIAL

Este caso de uso representa um utilizador a realizar a impressão de um documento. O utilizador deverá estar numa página de consulta de documento. A página de consulta do documento vai conter um botão para realizar a impressão do documento.

O documento original só poderá ser impresso uma vez, sendo que depois só poderão ser impressas segundas vias. A exceção é quando o documento é indicado como sendo mal impresso, deverá ser permitido indicar que o documento foi mal impresso, bem como a razão, e nesse caso já poderá voltar a ser impresso o documento original.

Quando o documento é impresso, é descarregado para o computador um documento PDF que pode ser usado para imprimir o documento comercial. Aparece então na janela de consulta um popup que pergunta se a impressão foi efetuada com sucesso ou não. Caso seja indicado que não aparece uma caixa de texto onde é pedido para identificar a razão de não ter sido impresso com sucesso. Nessa altura o sistema restaura o estado de impressão como não tendo sido ainda impresso, e volta a permitir impressões do original. Caso o sistema tenha indicado que o original já foi impresso, a partir daí o botão de impressão do documento vai oferecer uma 2ª via em vez do documento original.

UC016 – ANULAR DOCUMENTO COMERCIAL

Este caso de uso representa um utilizador a realizar a anulação de um documento. O utilizador deverá estar numa página de consulta de documento. A página de consulta do documento vai conter um botão para realizar a anulação do documento.

Aparece uma janela de popup a pedir a confirmação da anulação, e o utilizador é obrigado a indicar uma razão para a anulação, escrevendo em texto essa razão, e ficará identificado no documento qual o utilizador que o anulou e qual a razão de anulação.

Um ficheiro só poderá ser anulado se não tiverem sido emitidos sobre estes documentos retificativos, ou se estes documentos retificativos já tiverem sido anulados. Caso documento não possa ser anulado aparece uma janela de popup a indicar que o documento não pode ser anulado e a razão.

UC017 – EMITIR DOCUMENTO A PARTIR DO ATUAL

Este caso de uso representa um utilizador que irá criar um novo documento, mas quer realizar o pré-preenchimento deste com os dados de outro documento. Isto acontece frequentemente no caso do recibo, em que os seus dados são normalmente os mesmos da fatura que lhe deu origem. Outros exemplos são os Orçamentos e Guias de Transporte, onde vamos querer criar uma fatura ou fatura-recibo para faturar estes elementos, sendo que a maioria dos dados da fatura são os mesmos do orçamento ou guia de transporte. Este caso de uso vai envolver ir à página de consulta do documento original, e carregar no botão de emitir documento associado. Caso estejamos numa fatura ou fatura-recibo é aberta a página de criação de documentos e é pré-preenchido o tipo como recibo, e são preenchidos os dados da fatura ou fatura-recibo, como o cliente, processo, produtos, quantidades e preços. Caso estejamos num recibo não é permitido criar novos documentos associados. Caso estejamos em Orçamento ou Guia de Transporte é preenchido o tipo como fatura. Caso queiramos criar um documento de um tipo diferente, por exemplo de um orçamento criar uma guia de transporte em vez de uma fatura podemos na página de criação de documentos mudar o tipo de documento conforme acontecia no caso de uso de criar novo documento. Aliás, depois do preenchimento automático dos dados do documento original este caso de uso passa a ser o caso de inserir novo documento, simplesmente vários dados são preenchidos automaticamente. Os dados preenchidos automaticamente no novo documento a partir do original são no caso dos dados gerais os dados do cliente e do processo, e os dados das linhas de produtos e serviços. Estes dados das linhas de produtos e serviços irão conter em cada linha a referência ao documento original. Os dados do transporte do documento anterior já não são relevantes, tal como os dados do pagamento, visto não ser efetuado pagamento a orçamentos e guias de transporte, apenas à fatura associada a estes.

UC018 – VER LISTA DE VENDAS LÍQUIDAS

Este caso de uso representa a determinação do valor de vendas líquidas num determinado período de tempo. Podem ser configurados parâmetros que refinam o espaço de procura de documentos, como só procurar documentos relativos a um certo cliente ou produto. Este caso de uso é semelhante ao caso de uso de procurar documentos comerciais, mas aqui são procurados valores totais de receita e despesa, em vez de documentos. Cada produto/serviço tem definido na sua ficha de produto um valor de custo, e esta janela permite somar os valores dos documentos a nível de receita líquida e despesa/custo, e com esse valor de custo vão ser calculados valores totais de receita líquida e despesa, de modo a poder visualizar a margem de lucro da empresa relativa a clientes e/ou determinados produtos. Podemos organizar os valores por clientes, por tipos de custo, ou por ambos, fazendo a relação entre os valores totais de cada cliente e valores totais de cada produto/serviço. São apresentados o número de documentos, valores líquidos, valores de custo, margem bruta, e percentagem de margem bruta.

UC019 – VER VOLUME DE FATURAÇÃO

Este caso de uso representa a soma de toda a faturação num determinado ano (Para todos os clientes ou para um cliente em específico). Este valor faturado não contém os valores de faturas anuladas, e contém os valores já retificados de acordo com as notas de débito e de crédito.

Deve ser escolhido um ano, e pode ser escolhido um cliente.

É apresentado o número total de processos, o número total de faturas, o valor total faturado nesse ano sem contar com o IVA, bem como estes valores por cada mês, de modo a ver a progressão da faturação em cada mês.

UC020 – VER TAXAS DE IVA PARA UM PERÍODO

Este caso de uso representa a visualização do IVA a pagar e a haver num certo período de tempo, tanto total como por documento, bem como a data de cada documento, a sua identificação, e qual o cliente.

No ficheiro SAF-T (PT) aparece o valor total faturado sem IVA, e cada documento tem indicado o valor de IVA que deve ser pago relativo a este, mas neste ficheiro não é calculado o valor total de IVA que deve ser pago pela empresa. Logo esta funcionalidade permite mostrar a estimativa do valor de IVA que a empresa tem a haver e a pagar num certo período de tempo, de modo a poder ser obtida facilmente esta informação.

UC021 – EXPORTAR FICHEIRO SAF-T(PT)

Conforme indicado no capítulo dos requisitos de certificação de software de faturação, a exportação do ficheiro normalizado de dados SAF-T(PT) é um dos principais requisitos. Este ficheiro deve também obrigatoriamente ser entregue mensalmente pelas empresas no portal das finanças, de modo a comunicar às finanças regularmente a situação financeira desta.

Esta janela vai permitir seleccionar o período de tempo (data de início e de fim), e gerar um ficheiro SAF-T(PT) para um ficheiro no computador do utilizador. Este ficheiro pode depois ser consultado, ou entregue diretamente às finanças através do portal das finanças.

UC022 – INSERIR/MODIFICAR PRODUTO

Este caso de uso representa a gestão dos produtos que aparecem nas linhas de produto dos ficheiros comerciais. Alguns dos dados das linhas de produtos podem ser configurados na altura, mas esta tabela permite preencher alguns outros dados e definir valores por defeito para os produtos/serviços/etc.

Podem representar não só produtos, como serviços, tipos de imposto e outros, existindo uma variável que indica qual o tipo do elemento. Também é definido a descrição do produto, qual é a unidade de medida (porque esta informação é necessária para a exportação do ficheiro SAF-T(PT)), qual é a taxa de IVA deste produto (pode ser alterada caso a caso em cada linha deste produto, logo este só diz o valor por defeito), qual é o seu preço (com e sem IVA, e este é só o valor por defeito também, porque o preço unitário pode ser alterado individualmente em cada linha de produto), e qual é o seu preço de custo (com e sem IVA). O preço de custo só serve como estimativa de modo a fazer certos mapas de faturação que calculem o lucro da empresa subtraindo a despesa interna com cada elemento, e sendo assim este preço de custo não irá ter nenhum impacto na inserção de documentos de faturação.

É também possível adicionar observações à ficha desse produto.

Este caso de uso define a funcionalidade de procura de produtos, bem como a funcionalidade de edição de produtos.

UC023 – PROCURAR CONTACTOS INDIVIDUAIS

Este caso de uso representa a procura de um determinado contacto individual, ou simplesmente a visualização da lista de contactos individuais. Depois pode ser seleccionado um certo contacto individual para proceder à sua visualização, edição ou apagamento.

Um contacto individual é uma pessoa representante de uma empresa, podendo pertencer à empresa atual, sendo um empregado desta, ou pertencer a um dos seus clientes ou fornecedores.

Cada contacto Individual é representado por um nome, uma data de nascimento, um endereço de email, um número de telefone, um número de telemóvel, a identificação da empresa a que pertence, dados complementares, e opcionalmente uma fotografia.

UC024 – VER CONTACTO INDIVIDUAL

Este caso de uso representa a visualização dos detalhes de um determinado contacto individual. No fundo da janela existem atalhos para proceder à edição ou apagamento deste contacto.

UC025 – ADICIONAR/EDITAR CONTACTO INDIVIDUAL

Este caso de uso representa a edição dos detalhes de um determinado contacto individual. No fundo da janela podem ser confirmadas ou canceladas as alterações efetuadas.

UC026 – PROCURAR CONTACTOS DE EMPRESA

Este caso de uso representa a procura dos detalhes de uma determinada empresa, ou simplesmente a visualização da lista de contactos de empresas. Depois pode ser selecionado um certo contacto de empresa para proceder à sua visualização, edição ou apagamento.

Um contacto de empresa representa um cliente da empresa, ou um fornecedor, ou até mesmo uma pessoa (que irá contar como empresa), caso os documentos comerciais sejam passados em nome desta, em vez de ser em nome de uma empresa.

Os contactos individuais são as pessoas representantes destas empresas, ou empregados delas, e não podem ter documentos comerciais passados em seu nome.

O programa já contém um contacto de empresa (cliente) por defeito, o Consumidor Final, para ser usado em documentos nos caso em que o cliente não necessita de ser identificado no documento (este não pediu documento com o seu número de Contribuinte).

Cada contacto de Empresa é representado por um nome de empresa, morada, código postal, localidade, número de telefone, número de fax, endereço de email, NIF (número de contribuinte), e dados complementares.

UC027 – VER CONTACTO DE EMPRESA

Este caso de uso representa a visualização dos detalhes de um determinado contacto de Empresa. No fundo da janela existem atalhos para proceder à edição ou apagamento deste contacto.

UC028 – ADICIONAR/EDITAR CONTACTO DE EMPRESA

Este caso de uso representa a edição dos detalhes de um determinado contacto de Empresa. No fundo da janela podem ser confirmadas ou canceladas as alterações efetuadas.

UC029 – APAGAR CONTACTO (INDIVIDUAL OU DE EMPRESA)

Este caso de uso representa o apagamento de um contacto. Este funciona de maneira semelhante quer seja contacto individual ou contacto de empresa.

Não existe uma página específica para proceder ao apagamento de contactos, podendo apagar o contacto a partir de várias páginas, incluindo as páginas de procura de contactos (individuais e de empresa), as páginas de visualização de um contacto, e as páginas de adição/modificação de um contacto (Se estivermos a adicionar um contacto o botão de apagar vai apenas cancelar a edição do contacto que estamos a preencher).

UC030 – PROCURAR IMPOSTOS

Este caso de uso representa a procura dos detalhes das tabelas de Impostos existentes no programa. Os Impostos definidos nessa tabela indicam os impostos que podem ser

selecionados para cada linha de produto/serviço em cada documento comercial.
No caso de existir Isenção de Imposto deve ser selecionado um imposto próprio para isso, e depois na própria linha de produto/serviço deve ser indicada a razão de isenção (existem várias razões de isenção definidas em artigos da lei)
Cada Imposto contém uma descrição do imposto, um tipo de imposto, um código de imposto, um valor de percentagem, e um país e região associado.

UC031 – ADICIONAR/EDITAR IMPOSTO

Este caso de uso representa a inserção ou edição dos detalhes de um determinado Imposto. No fundo da janela podem ser confirmadas ou canceladas as alterações efetuadas.

Os Impostos definidos servem para indicar para cada linha de produto/serviço em cada documento comercial qual o imposto que deve ser aplicado a cada linha.

No caso de existir Isenção de Imposto deve ser selecionado um imposto próprio para isso, e depois na própria linha de produto/serviço deve ser indicada a razão de isenção (existem várias razões de isenção definidas em artigos da lei)

Cada Imposto contém uma descrição do imposto, um tipo de imposto, um código de imposto, um valor de percentagem, e um país e região associado.

A descrição do imposto deve ter um texto que permita facilmente a identificação dos detalhes do imposto (principalmente tipo de imposto e o seu código), e no caso de ser um imposto de selo deverá ser preenchido com o valor da verba respetiva. O tipo de imposto poderá ser IVA, Imposto de Selo ou Não Sujeição a IVA ou Imposto de Selo. O código de um Imposto de IVA deverá conter Taxa Normal, Taxa Reduzida, Taxa Intermédia, Isenta ou outra de acordo com o necessário a aplicar em cada produto/serviço devido à legislação que regula qual o tipo de imposto a aplicar a estes. O código de um Imposto de Selo deverá conter o código da verba respetiva, o código de um Imposto Não Sujeito a IVA ou Imposto de selo deverá conter NS ou Não Sujeição. A percentagem do imposto deverá ter o valor a aplicar para esse código de imposto em percentagem normalmente, caso se trate de uma verba fixa de imposto do selo, terá o valor não em percentagem mas logo em moeda. O país e região do imposto indica se o imposto é relativo a Portugal Continental ou a uma das Regiões Autónomas, podendo ser preenchido com Portugal Continental, Região Autónoma dos Açores e Região Autónoma da Madeira.

UC032 – APAGAR IMPOSTO

Este caso de uso representa o apagamento de um imposto. Não é possível apagar um imposto se já existirem documentos comerciais com linhas de produto que usem este imposto.

Não existe uma página específica para proceder ao apagamento de contactos, podendo apagar o imposto a partir de várias páginas, incluindo as páginas de procura de impostos, e as páginas de adição/modificação de um imposto (Se estivermos a adicionar um imposto o botão de apagar vai apenas cancelar a edição do imposto que estamos a preencher).

UC033 – ALTERAÇÃO DOS DADOS DO PROGRAMA

Este caso de uso representa a configuração de certos atributos do programa relativos ao módulo de faturação. Estes atributos permitem configurar o valor por defeito de vários campos do programa. Será possível definir o texto a colocar nos cabeçalhos dos documentos comerciais, o texto a colocar no rodapé dos documentos comerciais, o texto a colocar no campo de observações, qual o tipo de documento comercial por defeito (que

normalmente será fatura ou fatura-recibo), qual o imposto (normalmente será taxa de IVA normal), qual a razão de isenção do IVA por defeito quando um produto/serviço é adicionado como isento, qual a unidade de medida de um novo produto/serviço, qual o prazo de pagamento por defeito, e o tipo de pagamento normalmente efetuado (Numerário, cheque, Transferência Bancária, etc).

AUTENTICAÇÃO E GESTÃO DE UTILIZADORES

Sendo que é um programa que irá conter bastantes informações confidenciais e cuja utilização vai ser pela Internet num Browser, é importante restringir a sua utilização com um sistema de utilizadores com credenciais (nome de utilizador e palavra-chave). Logo sem existir autenticação apenas será possível aceder à página de autenticação e à página de recuperação de palavra-chave (que será enviada para o email).

Cada utilizador terá um conjunto de detalhes associados (como o seu nome e palavra chave). Cada utilizador irá pertencer a um grupo (definido pelo Administrador da empresa), e a cada grupo de utilizador pode ser definido um conjunto de permissões que definem quais as funcionalidades do programa a que esse grupo tem acesso.

Por exemplo um grupo de utilizadores poderá apenas visualizar dados de processos ou dados de faturas, enquanto outros poderão fazer alterações só em certas funcionalidades. Um grupo de utilizadores são os utilizadores externos (clientes da empresa) que poderão ter credenciais de acesso ao sistema, e apenas poderão consultar (e não modificar) dados de processos e faturas relacionadas com a empresa cliente da qual esse utilizador faz parte.

Normalmente o Administrador de Empresa é o grupo de utilizador ao qual são permitidas todas as operações, e que tem a possibilidade de configurar quais as funcionalidades a que cada grupo de utilizador tem acesso.

4.4. Navegação no programa

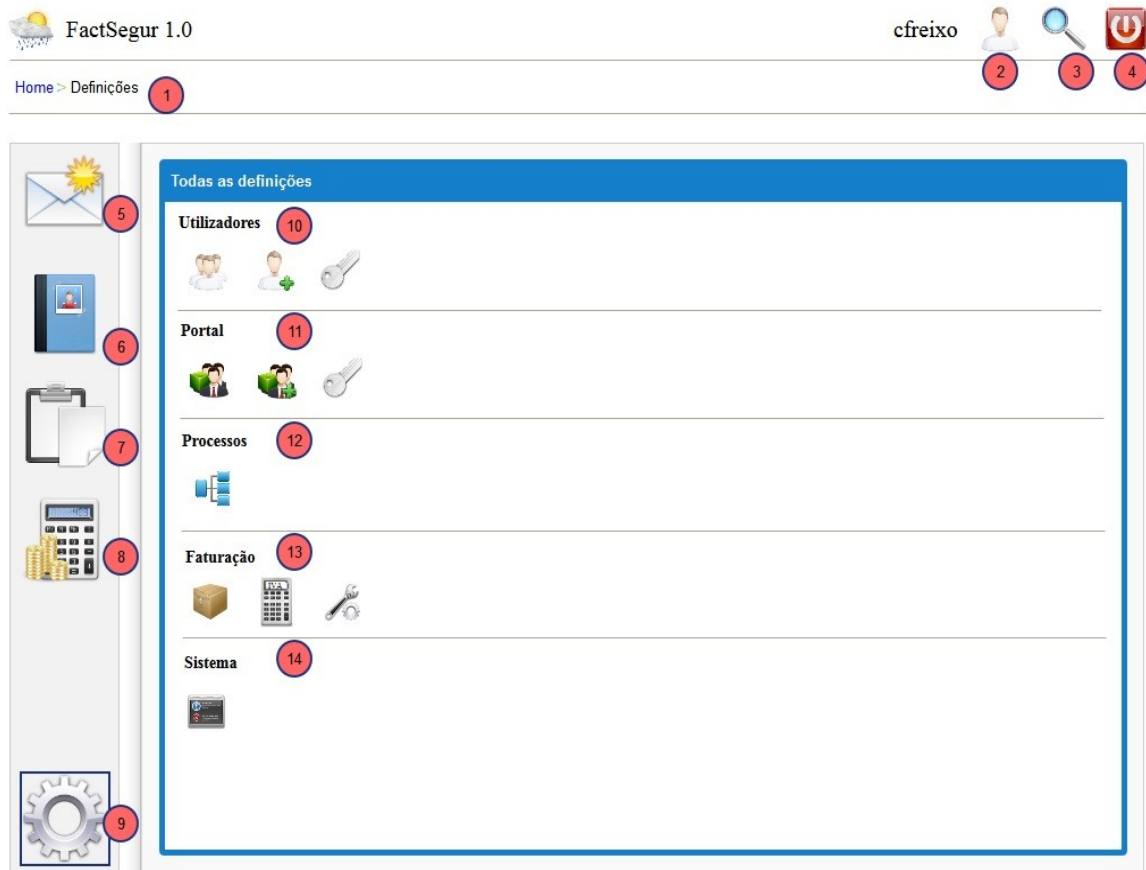


Figura 10: Protótipo: Definições

NAVEGAÇÃO NO PROGRAMA

Ao longo de toda a página pode ser encontrado um cabeçalho que identifica o utilizador que está autenticado e o atalho para mudar as suas definições (Protótipo: Definições 2), bem como a funcionalidade de procura genérica do programa (Protótipo: Definições 3), e o botão para cancelar a sua autenticação (Protótipo: Definições 4).

Por baixo deste cabeçalho pode ser encontrado uma área que contém um elemento de navegação chamado *breadcrumb*, que mostra qual é a área do programa onde estamos atualmente, e permite navegar para trás, para as páginas hierarquicamente anteriores (subsecções da mesma área). No canto direito desta área, por baixo dos botões 2 a 4 (que na imagem estava vazio) por vezes podem ser encontrados ícones de atalho para outras funcionalidades da mesma área ou módulo. Por exemplo no caso dos contactos pode ser encontrados os botões que trocam entre estar a ver contactos individuais e contactos de empresa.

No lado esquerdo do programa pode ser encontrada uma barra de navegação que permite trocar entre módulos, abrindo a página principal do módulo selecionado (Protótipo: Definições 5 a 9). No fundo desta barra pode ser acedidas às definições do programa que contém várias definições de cada módulo e definições do programa em geral, como a gestão de utilizadores. No caso da faturação permite configurar os tipos de produto que aparecem nas linhas dos documentos, os impostos disponíveis como o IVA, e outras definições como os valores por defeito de certos campos da criação de documentos comerciais.

4.5. Módulo de Contactos

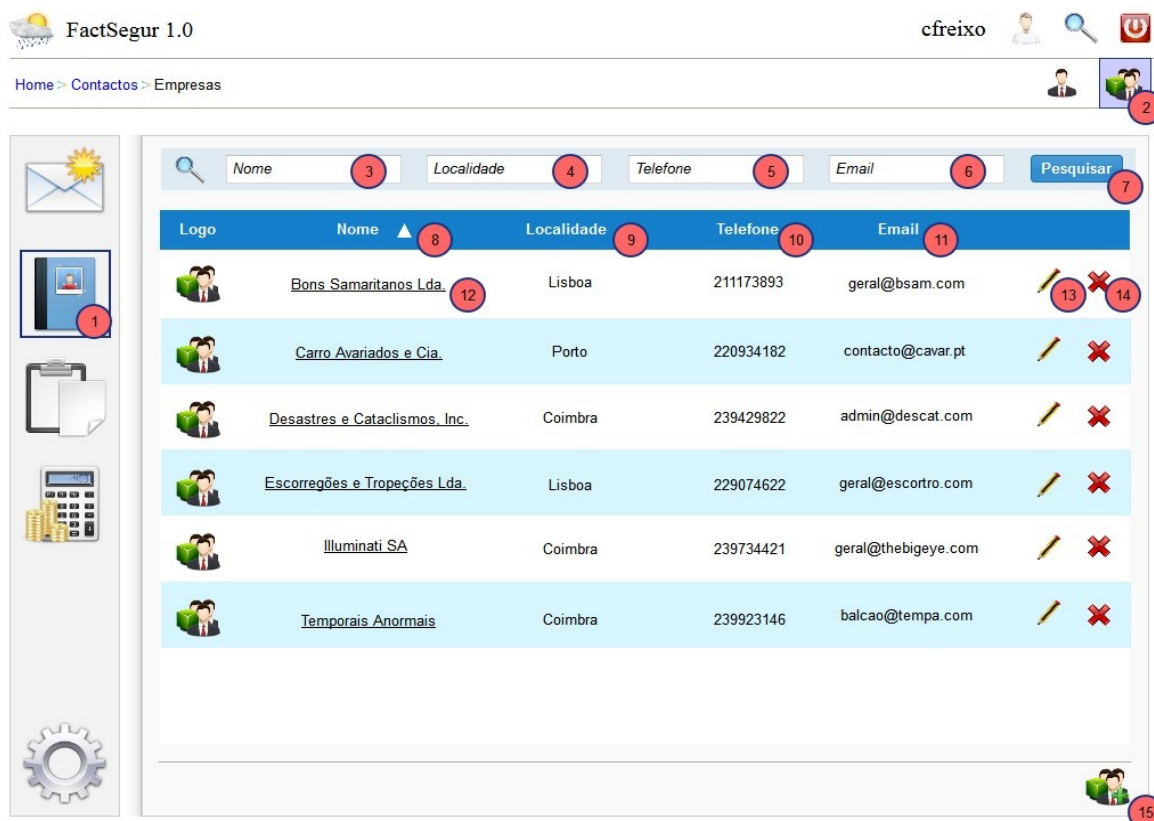



Figura 11: Protótipo: Contactos




Os contactos estão divididos em duas partes, os contactos individuais e contactos de empresa. Os contactos individuais são pessoas que fazem parte de empresas que existam na parte de contactos de empresa. Os contactos de empresa são as empresas que serão os clientes dos processos e dos documentos comerciais. Estes contactos de empresa terão portanto obrigatoriamente a identificação do nome de empresa, morada e NIF.

A maneira de utilização é bastante similar para um tipo de contactos e o outro, apenas mudando os campos existentes para cada um. Podem ser filtrados os contactos que aparecem por variados campos (Protótipo: Contactos 3 a 6) e só irão aparecer os contactos que cumpram com as restrições apresentadas. Pode também ser selecionado um contacto para ver todos os seus dados (Protótipo: Contactos 12), modificar os seus dados (Protótipo: Contactos 13), ou proceder ao apagamento do contacto (Protótipo: Contactos 14). Devido a requisitos de certificação não é possível apagar contactos se esse tiver documentos emitidos em seu nome.






Pode também ser adicionado um novo contacto (Protótipo: Contactos 15).

4.6. Módulo de Faturação

 FactSegur 1.0

cfreixo   

Home > Pesquisa avançada



Campos a pesquisar

Selecione módulo

Faturação - Procurar Documento

Selecione campos de pesquisa

Selecionar...

Tipo de Documento

Fatura

Empresa

a

Pesquisar

Data	Documento	Empresa ▲	Valor Sem Iva	Pago
01-05-2015	Fatura 32/1	Temporais Anormais	100.00 €	
05-05-2015	Fatura 36/1	Carro Avariados e Cia.	200.00 €	
03-05-2015	Fatura 35/1	Bons Samaritanos Lda.	100.00 €	
12-04-2015	Fatura 22/1	Escorregões e Tropeções Lda.	500.00 €	
01-06-2015	Fatura 45/1	Desastres e Cataclismos, Inc.	100.00 €	
21-03-2015	Fatura 15/1	Illuminati SA	250.50 €	

Figura 12: Protótipo: Pesquisa de documentos

A opção genérica de pesquisa do programa pode ser acedida no botão da barra superior (a lupa). Dentro da pesquisa pode ser definido o módulo do qual o elemento a pesquisar está inserido (neste caso estamos a fazer uma pesquisa de faturação). Depois podem ser inseridas variadas restrições (campos de pesquisa), que são diferentes de acordo com o módulo ou elemento selecionado anteriormente. No lado direito aparecem os resultados, e pode ser selecionado o elemento para ver ou alterar os seus detalhes (neste caso podemos carregar no nome do documento para ver a sua página, não podemos alterar os detalhes porque documentos não podem ser alterados, mas podemos visualizar, podemos anular o documento, imprimir o documento para pdf, e outras operações). Podemos também ordenar a pesquisa pelos vários campos, neste exemplo podemos verificar que os resultados estão ordenados pelo nome da empresa.

A nível de faturação, o mais importante é a criação de documentos comerciais. Por causa de questões fiscais e de certificação de software, estes depois de criados não podem ser alterados. Podem ser anulados, sendo que os dados do documento são mantidos, simplesmente este passa a contar como anulados, podendo o documento ser consultado à mesma. Os documentos mais relevantes, as faturas e faturas-recibo também podem ser corrigidas por meio de notas de crédito e notas de débito (retirar elementos das suas linhas de produtos/serviços, ou acrescentar novos elementos e suas quantidades).

A nível de documentos comerciais, qualquer que seja o tipo de documento que o nosso programa suporta (fatura-recibo, fatura, nota de crédito, nota de débito, orçamento, guia de transporte), estes irão conter os mesmos dados. Estes dados irão ser a identificação do documento (qual é o tipo deste e o seu número e série), a data de emissão e data de

pagamento, qual será o cliente (pode ser consumidor final, que é o consumidor genérico que não apresenta o seu número de contribuinte, ou então um cliente identificado por nome, morada, código postal e número de contribuinte, destes é obrigatório ou o número de contribuinte ou o nome, morada e código postal), os detalhes sobre a carga e descarga de mercadorias (esta informação é facultativa exceto nas guias de transporte), observações e notas, e dados do processo (estes dados são relativos ao processo de negócio da empresa, dá jeito a estes para identificação, podem constar dos documentos comerciais como as faturas mas não são relevantes fiscalmente logo não são obrigatórios). Estes dados do processo incluem a identificação do processo na empresa de Peritagem, bem como a identificação interna da empresa de Seguros, o número da apólice de seguros, bem como outros dados.

Cada Documento Comercial ainda vai conter a informação dos produtos/serviços envolvidos em linhas de documento. Cada linha identifica um produto/serviço, valores unitários, quantidade, imposto a pagar e razão de isenção quando necessário, percentagem de desconto, e a informação se é um produto ou um serviço. Também deve ser possível associar cada linha a outros documentos comerciais, por exemplo para indicar nas notas de crédito qual o documento que cada linha está a retificar (a mesma nota de crédito pode retificar mais que uma fatura).

Certos documentos podem ser criados a partir de outros, tendo os mesmos dados, e podemos assim usar um documento para preencher o próximo. Um exemplo disso será um orçamento, que depois a fatura do documento terá o mesmo cliente, e as mesmas linhas de produto, e o recibo dessa fatura também terá esses mesmos dados. Logo teremos de fazer com que seja possível realizar o pré-preenchimento de documentos a partir de outros documentos de forma a minimizar a probabilidade de erros ao passar os dados de um documento para outros, e também como forma de facilitar o funcionamento do programa. No caso do nosso programa ao consultar o documento existe uma opção que permite criar um novo documento que fica pré-preenchido com os dados do documento que estamos a consultar.

A consulta dos documentos ainda permite aceder a opções de gestão do documento como proceder à sua anulação, e impressão deste, bem como consulta dos seus dados. O programa só deve oferecer uma vez o documento original (se bem que em pdf, este depois poderá ser impresso mais de uma vez), e ao voltar a imprimir os documentos subsequentes devem vir com a informação de que se trata de uma 2ª via.

Ainda é extremamente relevante anotar que a faturação irá conter uma funcionalidade para exportar o ficheiro SAF-T(PT) para um determinado período. Este ficheiro deve ser entregue todos os meses no portal das finanças, segundo a legislação. Para esta funcionalidade deve ser indicado o período (data de início e data de fim), e o programa devolve o ficheiro xml SAF-T(PT), que a empresa pode depois consultar e/ou submeter no portal das finanças. O programa têm de garantir que exporta todos os dados necessários conforme definido na estrutura definida nas portarias para esse efeitos, já discutidas no capítulo do estado da arte.

Vão existir também variados tipos de Listagens de faturação que permitem oferecer totais de valor faturado, Imposto devido num período de tempo, entre outros tipos de informação.

4.7. Atributos de qualidade

Nesta secção são indicadas as principais preocupações a nível de atributos de qualidade, e na criação da arquitetura irão ser definidas estratégias para garantir estes atributos.

Os requisitos não funcionais são gerais para todo o sistema, logo a listagem destes foi elaborada em conjunto entre os dois estagiários.

Atributo de Qualidade	Preocupação	Importância (High/Medium/Low)
Segurança	Autenticação obrigatória para a maioria das funcionalidades	High
Segurança	Proteção de dados armazenados (confidencialidade), assegurando o acesso controlado à base de dados, com palavras passe seguras, e restringindo o acesso à base de dados a partir do computador que esta serve.	High
Segurança	Proteção das comunicações entre o cliente e o servidor (confidencialidade) usando o mecanismo protocolo https.	High
Segurança	Proteção do sistema a ataques informáticos, com uso de ferramentas que controlem ataques DNS, uso adequado de firewalls e mecanismo anti-brute force de autenticação.	High
Modificabilidade	Separação em módulos e redução das inter-dependências	High
Modificabilidade	Inserção de novas funcionalidades em módulos e novos módulos sem existirem dependências, módulos devem conseguir trabalhar independentemente	High
Disponibilidade	Deteção e Recuperação de falhas (automática quando possível)	High
Auditoria	Todas as operações que efetuem mudanças nos dados devem ser registados num sistema de Logging	Medium
Desempenho	Tempo de Resposta inferior a 3 segundos em qualquer página.	Medium
Usabilidade	Utilização deve apresentar uma baixa curva de aprendizagem, em que ao fim de uma semana já trabalhe intuitivamente, sem ter que pensar onde é que tem que aceder para executar uma operação.	Low

Tabela 6: Atributos de Qualidade e sua priorização

Conforme pode ser verificado na tabela iremos precisar de ter em conta questões de segurança, modificabilidade, disponibilidade, auditoria, desempenho e usabilidade.

SEGURANÇA

Devido ao facto do sistema ir conter bastantes informações confidenciais, como é o caso dos dados do processo, bem como os dados de faturação da empresa, a preocupação com todas as questões de segurança devem ser da máxima importância. As questões principais neste ponto envolvem a segurança do servidor onde vão estar os dados, bem como a proteção da comunicação entre o servidor e o computador do utilizador. Também é importante a autenticação dos utilizadores de modo a só poder aceder ao programa utilizadores com credenciais válidas, e mecanismos que dificultem ou impossibilitem a obtenção de credenciais por parte de atacantes, ou acesso direto ao sistema e serviços e componentes do servidor.

MODIFICABILIDADE

O sistema deve de estar preparado para a inserção de novas funcionalidades ao longo do tempo, de modo a que a inserção de uma nova funcionalidade não implique alterações nas partes do sistema já existentes de modo a manter a coesão do sistema. Sendo assim deve ser optado por um mecanismo de modularização e tentar manter ao mínimo as dependências entre módulos, de modo a que a inserção ou alteração de módulos não

implique alterações em outros módulos, facilitando o desenvolvimento futuro da aplicação.

DISPONIBILIDADE

Devem existir mecanismos de deteção de falhas e erros, e quando possível usar técnicas de recuperação de falhas de forma automática, sem necessitar a intervenção de um administrador. Quanto tal for impossível, deverá ser avisado um administrador de modo a que este possa resolver o imprevisto o mais rápido possível. Estas falhas podem ser algo como o servidor web estar em baixo, o servidor ou servidores estarem em baixo, os servidores não terem ligação à internet, ou um problema de corrupção no sistema e seus dados.

AUDITORIA

Devem existir um módulo de logging que registre as operações efetuadas ao servidor e aos seus dados (normalmente não será registada informação sobre acesso a dados, mas será guardada informação quando dados são alterados ou inseridos). Este registo irá identificar o utilizador autenticado que realizou a alteração, o seu endereço de IP, qual foi a operação, e qual foi o resultado desta.

Deve também existir um mecanismo que verifique a consistência de dados críticos, por exemplo que garanta que os dados de uma fatura não desapareçam da base de dados visto faturas não poderem ser apagadas, apenas anuladas.

DESEMPENHO

A nível de desempenho, a resposta ao utilizador deve ser feita em tempo útil sempre que possível, normalmente com um máximo de 3 segundos. Como o nosso foco principal é a segurança, que entra em conflito com o desempenho, o desempenho irá ter uma prioridade média. A razão para esse conflito é normalmente não ser possível garantir os 2 atributos ao mesmo tempo, porque as verificações e mecanismos necessários a manter a segurança irão reduzir o desempenho.

USABILIDADE

De modo a ter utilidade para o cliente a aplicação deve ter um nível de usabilidade que implique uma baixa curva de aprendizagem, e que permita efetuar eficazmente as operações que eles necessitam de efetuar, de modo a reduzir o tempo que demora a efetuar cada operação, com vista a reduzir a eficácia do seu trabalho. Apesar de existir alguma importância neste aspeto, comparado com os outros atributos este foi indicado como sendo o de menor importância.

4.8. Conclusão

Foi nesta fase que tomou forma o que o nosso programa iria ser. Foi estudado o processo de negócio da empresa da área, de modo a saber o que era realmente necessário a esta, e a maneira de responder às suas necessidades da forma mais eficaz, permitindo poupar tempo a esta empresa otimizando o funcionamento deste seu sistema de apoio.

Foi decidido optar pela modularidade de forma a obter capacidade de resposta a novas funcionalidades posteriores à data deste estágio, tendo em vista a expansão futura do programa a novos mercados, e a novas áreas de negócio. Foram tidos em conta bastantes outras considerações a nível de atributos de qualidade, especialmente a nível de segurança, visto este programa ir conter bastante informação confidencial.

Foi muito importante a participação da empresa que nos ajudou a perceber as suas necessidades, e o que era para eles relevante, e o que não era, o que nos ajudou a perceber o que teríamos de implementar no nosso programa.

A representação em Tabelas de Casos de uso ajudou a identificar pontos que precisavam de ser mais detalhados, porque ao preencher os campos específicos destas ia acontecendo que precisavam de ser respondidas questões (e ser definidas coisas) de maneira a chegar ao nível de detalhe pretendido.

Foi então muito positivo o saldo, e no fim desta fase ficou-se com uma ideia bastante mais clara do que iria constar neste produto, e de como este iria funcionar.

5. Especificação da Arquitetura

Uma boa arquitetura é fundamental de modo a garantir que o projeto vai ser implementado de maneira ordeira, que os seus componentes comunicam entre si de uma maneira bem definida (e da maneira esperada), e que as relações entre componentes estão bem identificadas de modo a saber como é que funcionará o sistema como um todo.

De acordo com a proposta de estágio, e com a análise de requisitos efetuada, o objetivo é a criação de uma Aplicação para a Internet, que funcione em qualquer Web-Browser no computador do cliente. Para isto apenas precisamos de ter um Servidor Web a providenciar o serviço, e o cliente não precisará de instalar e atualizar nenhuma aplicação cliente, podendo utilizar a aplicação num Web-Browser em qualquer computador com acesso à Internet.

Neste capítulo primeiro é feita uma apresentação da arquitetura, primeiro num nível mais alto, ao nível dos variados componentes que vão estar presentes, sendo que a maioria destes componentes apenas precisarão de ser configurados, visto que a nossa implementação estará só dentro do Drupal, que foi a framework de desenvolvimento escolhida.

De seguida é apresentada uma introdução ao Drupal e a sua maneira de funcionamento, e de seguida são indicados os módulos a implementar e o que estes vão conter, apresentando por último o modelo de dados que indica como os dados necessários ao funcionamento da aplicação serão representados na base de dados.

As razões de escolha do PHP como linguagem e do Drupal como framework estão definidas e podem ser consultadas no AnexoB- Análise e estudo das linguagens e ferramentas a utilizar, e este foi escolhido por variadas razões, entre estas a sua grande utilização na Internet, o tamanho da sua comunidade, o número de elementos de apoio ao desenvolvimento, e o já ter sido utilizado variadas vezes para resolver questões de natureza semelhante, como a implementação de um ERP open source, o ERPal, bem como um sistema famoso e bastante conhecido de E-Commerce, o “Drupal commerce”, e também variados sistemas de faturação. O ERPal em especial, sendo um projeto open source e desenhado para empresas de prestação de serviços, como é o caso da área a que o nosso projeto se refere, tem um valor inestimável como referência, visto poder ser verificado como este produto fez para resolver certas questões de implementação que nós também vamos ter de tratar. Um exemplo destas questões é o método de criação de documentos pdf com os dados existentes na nossa aplicação.

Este produto estará a funcionar num servidor linux dedicado para este efeito possuído pelos estagiários, com uma ligação à Internet sem limites de tráfego constantemente ligada. Caso seja necessário, é possível adicionar novos servidores web e novos servidores de base de dados de modo a poder escalar a solução. Para isso poderá ser usado o mecanismo master-slave de replicação de base de dados de modo a poder distribuir a carga por mais bases de dados e a adição de novos servidores web apache.

5.1. Estrutura geral do sistema

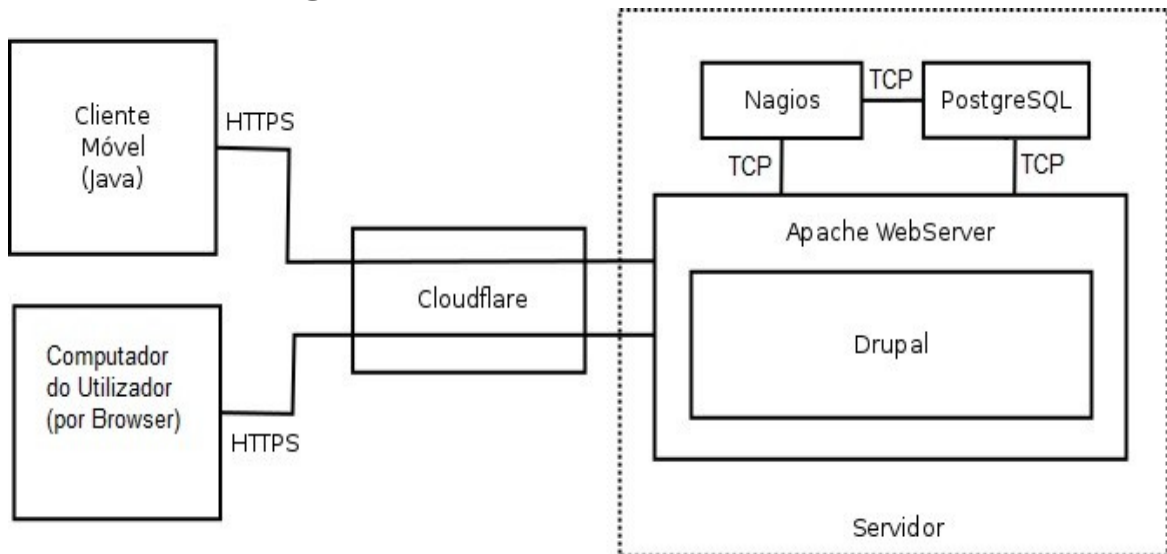


Figura 13: Vista de Topo do Sistema

Começa-se por apresentar aqui uma a arquitectura geral do sistema.

Pode-se visualizar através da vista de topo do sistema na figura anterior os principais componentes.

As ligações, tanto do cliente *web* como do cliente *mobile*, serão primeiramente encaminhadas pelo CloudFlare [42]. Estas ligações estarão encriptadas por SSL com uso do protocolo HTTP Secure (HTTPS).

A comunicação ir-se-á efectuar por HTTPS, tendo como destino o servidor Apache [43], que irá ter acesso às páginas pretendidas, utilizando para isso o CMS Drupal.

Por sua vez, os dados relativos à aplicação estarão guardados numa base de dados gerida pelo PostgreSQL [44], gerindo a fornecimento e armazenamento dos mesmos.

Para fazer a monitorização dos componentes que se encontram no servidor, atrás referidos, optou-se pela utilização do Nagios [45].

De seguida é apresentada uma explicação breve de cada um dos componentes (exceto o Drupal que sendo um componente de maior dimensão será apresentado em maior detalhe mais à frente).

5.1.1. CloudFlare

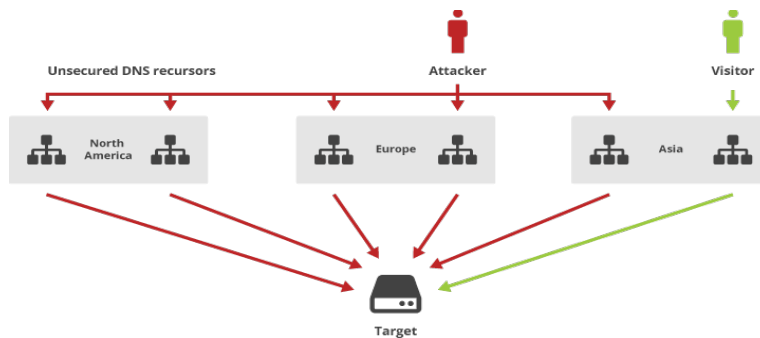


Figura 14: Sistema Vulnerável a ataques DDOS

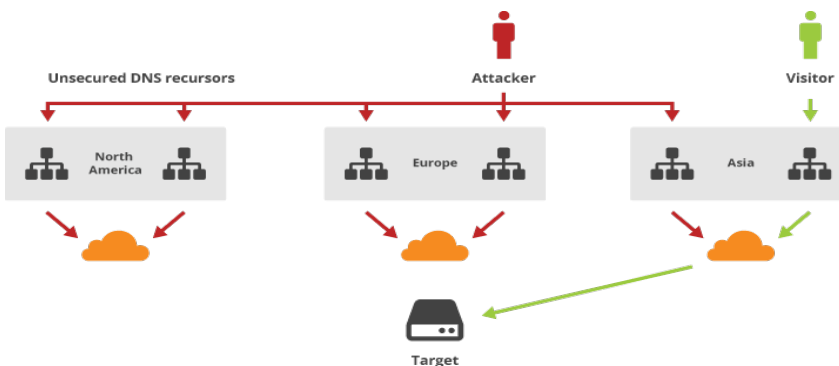


Figura 15: Sistema protegido por CloudFlare

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 [46].

A sua utilização permite a protecção contra ataques DDNS, o que, caso sucedam, podem afectar 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[47].

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.

Além disso, fornece protecção contra ataques SMURF, ACK e LAYER 7.

Pelo facto deste serviço disponibilizado ser gratuito e pelas excelentes críticas que recebeu [48], tornou-se aliciante a sua inserção neste projecto.

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 coisa erra ocorre e quando o problema foi resolvido [49].

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 [50] (conforme pode ser visualizado na vista de topo da arquitectura). 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 facto 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 objecto-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 companhias e contribuidores individuais. É gratuito e open-source, sob a PostgreSQL Licence, uma licença de software-livre permissiva [51].

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 directos [52][53].

Tendo todos estes aspectos 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) .

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 projecto [54].

5.2. Garantia dos atributos de qualidade

Em relação aos atributos de qualidade começaremos por indicar questões relativas à escalabilidade.

Segundo as informações obtidas na análise de requisitos, as empresas do ramo são de pequena dimensão, tendo no máximo 50 trabalhadores, logo é expectável que nunca

iremos ter mais do que 30 pedidos simultâneos por parte destes. Adicionando a possibilidade dos clientes se conectarem ao sistema para consultar dados dos seus processos, nunca seriam mais de 50 pedidos simultâneos por empresa. Assim sendo, este nível de escalabilidade é possível ser cumprido com a infraestrutura indicada em cima, e caso no futuro seja necessário escalar de modo a suportar mais empresas, tal como indicado em cima podem ser acrescentados novos servidores, e novas bases de dados, em que uma base de dados será a principal(master) e outras serão secundárias(slaves).

A nível de segurança, está contemplado o uso da Firewall do Linux no servidor para bloquear o acesso ao servidor a portas que não sejam necessárias à aplicação (como a porta SSH, só podendo ligar ao servidor localmente), e o uso do protocolo HTTPS para garantir a encriptação da informação através da Internet. O Drupal já tem o mecanismo de autenticação de utilizadores implementado, e este será usado para garantir a autenticação dos clientes. Também a nível de segurança a API de comunicação com a base de dados do Drupal já garante a proteção contra SQL Injection, e as API do Drupal contêm soluções para gestão de sessões, Cross Site Scripting, Cross Site Request Forgeries, entre outras [55].

A disponibilidade (deteção e recuperação de falhas) será garantida pelo Nagios, este permitirá aos administradores dos servidores saberem rapidamente por meio de uma mensagem SMS que um elemento do sistema está em baixo e tomar passos para o repor o mais rápido possível. Será necessário manter um nível de disponibilidade de pelo menos 95%, visto a aplicação ser indispensável ao trabalho do cliente. Existirão cópias de segurança da base de dados de periodicidade diária de modo a efetuar a reposição dos dados caso ocorra algum problema que não possa ser resolvido de outra forma. Estas cópias de segurança estarão encriptadas pelo motor de base de dados através do utilizador administrativo deste.

O tempo de resposta entre o pedido de uma página por parte do utilizador e a visualização da página correspondente deverá ser inferior a 5 segundos, sendo os mecanismos de caching do Drupal para criação de páginas deverão ajudar neste aspeto, e como não existe trabalho de processamento intensivo nas nossas páginas este tempo de resposta deverá estar assegurado.

A auditoria, mais propriamente o registo de operações relevantes estará garantida pelo sistema de Logs do Drupal, sendo que cada módulo a implementar estará responsável por adicionar mensagens ao Log, sempre que a operação o justifique. Deverá estar sempre identificada a operação, a data, e o utilizador que a efetuou.

A modificabilidade, tanto a nível de redução de inter-dependências entre funcionalidades e a facilidade de inserção de novas funcionalidades serão garantidas pelo sistema de módulos e hooks do Drupal. Este vai permitir que as funcionalidades estejam contidas no módulo, sendo que cada alteração apenas necessite de alterar o código desse módulo, não precisando de efetuar alterações nos outros porque estes são independentes tanto dos outros módulos como das funcionalidades Core do sistema.

Sendo uma página web, e utilizando opções padrão de interface aos quais os utilizadores já estão habituados, vamos garantir a usabilidade, e a diminuição da curva de aprendizagem, sendo que o utilizador já deve conseguir usar a aplicação intuitivamente, sem ter de andar à procura das funcionalidades, ao fim de um período de 3 a 5 dias.

5.3. 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 objectivo de ser multi funcional dentro do contexto da Web. Um dos seus maiores atractivos reside no facto de um programador poder utilizar a ferramenta com total liberdade desde os projectos mais simples, como uma Web page ou um blog, até aos mais complicados, como uma rede social, etc [56].

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 [57]. 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 [58].

Sendo o Drupal utilizado para diversos fins, tal como já foi referido, a gama de entidades que o utilizam vão desde a Casa Branca, até Zynga e o Ikea, entre outras [58].

Apesar de apresentar sérias barreiras para novos utilizadores, o *Drupal* é considerado por muitos o melhor CMS do mercado. Detém uma grande performance (apresenta as páginas mais rapidamente do que os outros), é tecnicamente mais avançado, customizável 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 [59].

Em suma, o Drupal apresenta-se como uma excelente escolha para utilização neste projecto de estágio. No entanto, dado que este CMS apresenta uma arquitectura própria, o software irá, por conseguinte, herdá-la. Convém portanto efectuar um estudo mais aprofundado da arquitectura do Drupal, a qual iremos apresentar na secção seguinte.

5.3.1. Padrão Arquitetural do Drupal

O Drupal utiliza o padrão arquitectural Presentation–Abstraction–Control (PAC) [60].

O PAC é um padrão arquitectural de software orientado à interacção, similar ao padrão Model-View-Controller (MVC). Em ambos os padrões, existe um sistema interactivo constituído por três componentes que são responsáveis por diferentes aspectos 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 [61].

As principais características do PAC são a utilização de uma estrutura hierárquica de agentes. Cada agente consistido por uma tríade (apresentação, abstracção e controlo), sendo que, a componente de apresentação está isolada da de abstracção, 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 abstracção ter

seja totalmente inicializada [60].

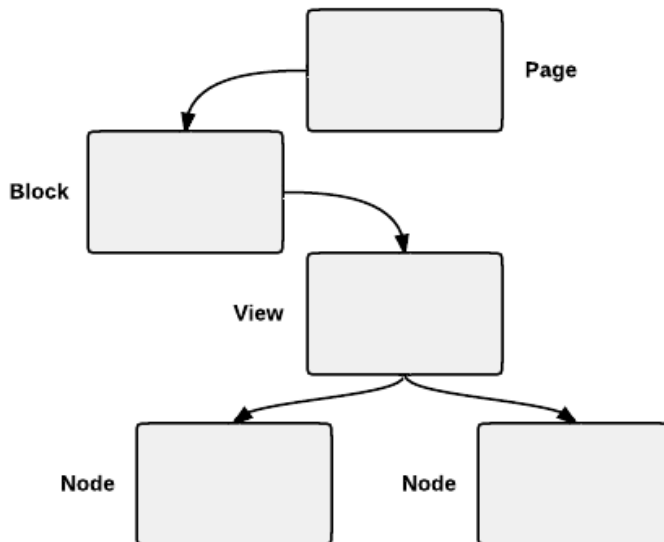


Figura 16: Sistematização da arquitetura do Drupal (fonte: <http://robknight.org.uk/blog/2011/02/explaining-architectural-tiers-drupal>)

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 nodos é devolvida e apresentada nos blocos são utilizadas *Views* [56].

Todos os conteúdos num *Website Drupal* estão armazenados e são tratados como nós (nodes). Cada um destes representa um conteúdo individual. Tratando todos os conteúdos como nodos, permite uma flexibilização da criação de novos tipos de conteúdos e facilita a aplicação de novas características e alterações a todos os nodos de um tipo [56].

A arquitectura 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.

5.3.2. Perspetiva de Fluxo do Drupal

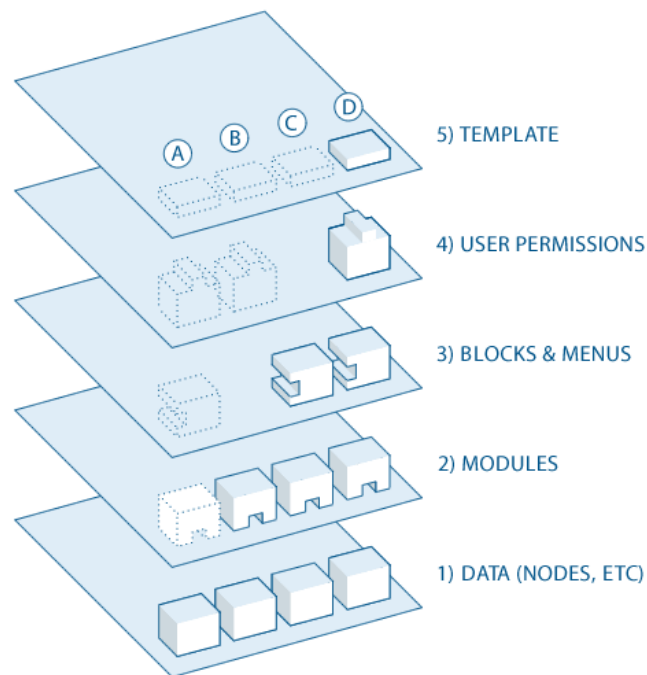


Figura 17: Perspetiva de Fluxo do Drupal (fonte: <https://www.drupal.org/getting-started/before/overview>)

Segundo a documentação do Drupal [62] a camada base (1) é a camada de dados, onde os dados são guardados na forma de nós na base de dados de acordo com uma API própria do Drupal para comunicação com a base de dados (formato próprio, não é simplesmente usar comandos SQL).

De seguida a camada de módulos (2) permite adicionar funcionalidades, manipular os campos de dados existente nos tipos nós, programaticamente ordenar e mostrar conteúdo (com filtros por exemplo), e muitas outras coisas. É nesta camada que são obtidos e manipulados os dados que irão passar para as páginas no futuro (normalmente em forma de arrays), e que a maioria do conteúdo é organizado. Existem módulos que fazem parte integral do Drupal, módulos implementados pela comunidade do Drupal e publicados no Website deste para poderem ser usado por qualquer pessoa na construção dos seus próprios websites, e qualquer pessoa pode criar módulos novos de modo a cumprir com as suas necessidades.

A camada de Blocos e Menus (3) é a seguinte. Os blocos correspondem ao conteúdo que os módulos devolvem (a maioria das vezes sem o aspeto definido, só terá aspeto ao passar pela camada de Tema), sendo possível configurar quais as páginas (poderão ser várias) ou utilizadores onde esse bloco aparece. Os Menus são os navegadores do Drupal, e permitem definir o conteúdo que aparece em cada página (URL).

A camada de Permissões de Utilizador (4) vai permitir definir níveis de acesso (Roles), e definir quais as páginas ou conteúdo que esse nível pode visualizar. Depois a cada utilizador é atribuído um nível de acesso, e podemos assim restringir o acesso a páginas e conteúdos de modo a só certos utilizadores terem acesso a este.

Finalmente a camada de Tema (5) vai pegar no conteúdo definido e vai transformá-lo em páginas HTML com o aspeto pretendido que irão ser mostradas ao utilizador. É a camada de tema que pega no conteúdo dos blocos (dados em forma de array), o vai ordenar, e inserir as tags php, código CSS, HTML e XHTML de forma a construir a página que vai ser mostrada ao utilizador. Existem temas que vêm com o Drupal, e podem ser encontrados muitos mais temas implementados pela comunidade, podem ser implementados temas de acordo com as necessidades.

5.3.3. Sistema de Renderização de Dados

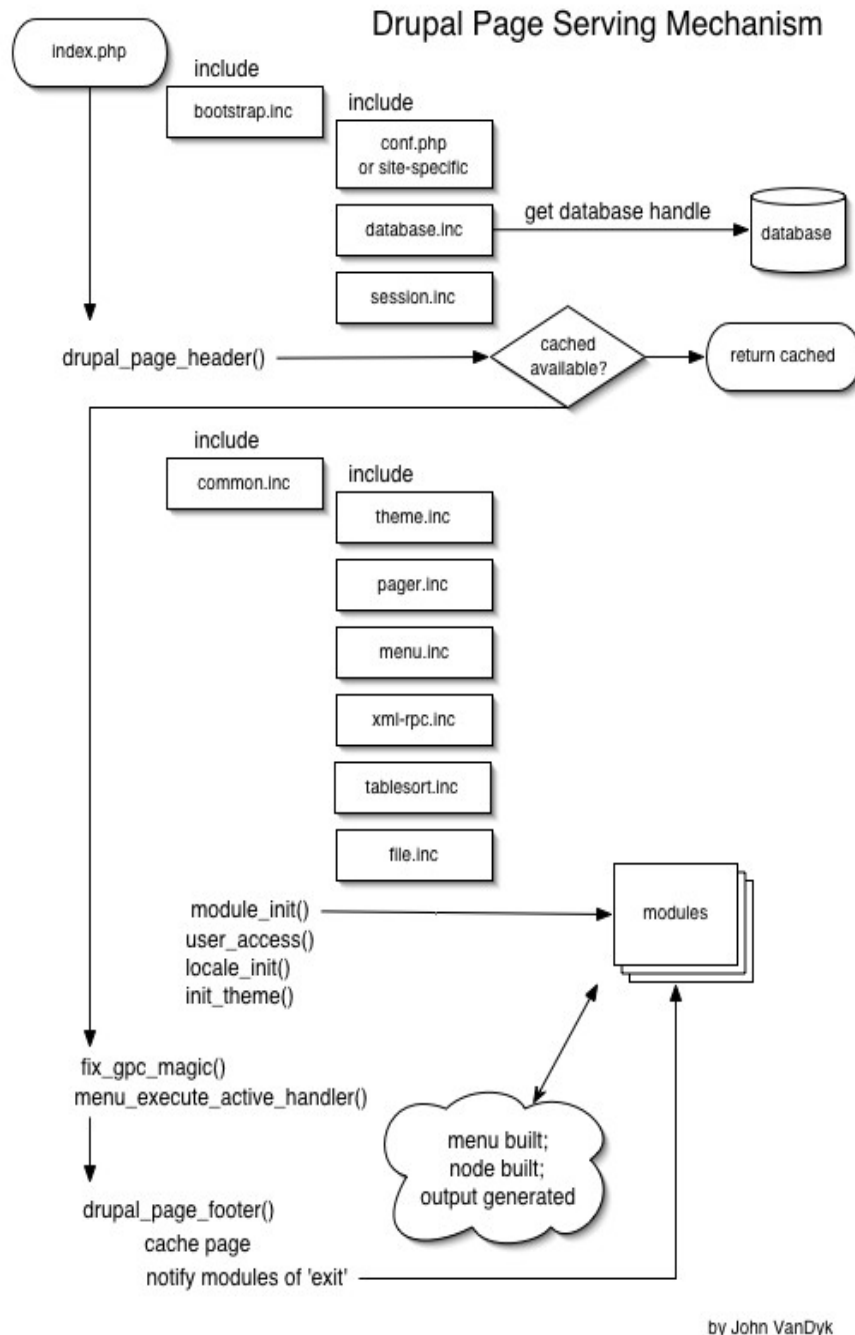


Figura 18: Mecanismo de obtenção de Página do Drupal (fonte: <https://www.drupal.org/node/10858>)

Embora o esquema anterior diga respeito à versão 4.6 do Drupal, não foi encontrada nenhuma versão mais recente relativamente a este aspecto, e é referido num site como o modelo vigente [63].

Como se pode ver, inicialmente é carregada a página `index.php`, onde estão incluídos ficheiros que estão encarregados de acesso a diversos componentes do sistema, tal como a base de dados (`database.inc`) e dados de sessão (`session.inc`).

Caso a página já se encontre em cache, são utilizados esses dados e são chamados os métodos respectivos para efectuar essa construção, havendo depois a notificação para os módulos que a página terminou a sua execução (lado esquerdo da imagem).

Por outro lado, se a página não se encontrar em cache, são carregados ficheiros cujo objectivo é determinar a forma como a página irá ser apresentada. Entre eles temos o `theme.inc` (encarregado do tema da página, onde aparece cada um dos componentes), `pager.inc` (com funções que auxiliam na apresentação de resultados na base de dados como um conjunto de páginas), `menu.inc` (funções para construção dos menus de acesso aos diversos componente), etc.

Em seguida são chamados os módulos que dizem respeito a esta página, cada um deles dependendo das permissões dos utilizadores (`user_access`), definições e preferências de localização (`locale_init`) e carregamento do tema específico em relação aquele módulo.

O seus componentes são então construídos num conjunto de arrays internos, e o output é gerado de acordo com esses mesmos objetos.

5.3.4. Sistema de Hooks do Drupal

O Drupal é constituído por diversos módulos, desde módulos internos do Drupal, a imensos módulos criados pela comunidade Drupal e podem livremente ser utilizados em websites, e módulos criados especificamente para websites.

Para efectuar esta comunicação entre os módulos, o Drupal implementa um sistema inteligente de comunicação entre os módulos: o *hook system*. De uma forma bastante resumida, quando é executada a visualização de uma página, o Drupal procura, nos módulos que estejam activados, “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.

Este sistema de hook implica que uma página pode ser construída por variados módulos, que vão adicionando conteúdo conforme o definido nestes para criar a página final resultante; em vez do esquema tradicional, onde existe uma página html real, e o conteúdo dinâmico é acrescentado à página usando código php. Aqui não existe uma página real que corresponda ao URL, mas sim uma página "virtual" que é construída a partir de todos os módulos que implementem hooks para esse URL. Logo uma página pode ter o seu conteúdo distribuído por vários módulos.

Sendo assim, o Drupal é constituído por um conjunto de módulos que comunicam entre si e que, através da adição de um módulo suplementar, as funcionalidades dentro de um determinado módulo podem ser aumentadas de uma forma simples.

Dentro dos módulos o conteúdo é transportado na forma de arrays encadeados (Nested Arrays). Um exemplo particular dos Nested Arrays são os Render Arrays que são arrays

que aderem aos standards de estrutura de dados definidos pela camada de renderização de temas do Drupal. Estes contêm os dados encadeados, bem como propriedades que indicam como deverá ser feita a sua renderização na camada de tema [64].

Logo os módulos (um ou mais) vão adicionar conteúdo de forma a construir os nested arrays com o conteúdo da página, e no fim a camada de tema vai pegar nestes arrays e construir a página final com o aspeto adequado de acordo com as propriedades definidas, e com o tema atual.

Outro aspeto importante é a existência de um sistema de Callbacks, que permite a inserção de nomes de funções dentro dos nested arrays, de forma a estas serem chamadas posteriormente no momento adequado. Isto vai permitir por exemplo inserir funções relativas à renderização, mas adiar execução destas até à chegada do conteúdo à camada de tema, permitindo que todos os módulos tenham a oportunidade de alterar o conteúdo da página. Outro exemplo da utilização de callbacks é a adição de uma função de validação a um form.

5.4. Vista de Módulos da aplicação FactSegur

O Drupal subdivide-se em vários módulos, que afectam o sistema, aumentam-no e comunicam entre si de modo a gerar novos conteúdos facilmente. Alguns desses módulos fazem parte integral do Drupal(Core), alguns módulos são implementados pela comunidade e publicados no website do Drupal, fornecendo imensas funcionalidades já implementadas, e cada utilizador pode implementar os seus próprios módulos de modo a implementar as funcionalidades necessárias de acordo com as suas necessidades.

Sendo assim, de modo a documentar a arquitectura do sistema proposto por este estágio na sua componente Web, fará sentido apresentar uma vista de módulos, e mostrar a forma como eles colaboram entre si de modo a apresentar os seus próprios conteúdos e aumentar as capacidades de outros módulos já presentes.

Por baixo é apresentada a figura de vista de módulos, deixando de parte os módulos que estão já presentes no seu *core* e concentra-se nos módulos que dizem respeito a esta aplicação, que serão os principais visados e fomentadores das funcionalidades do sistema. Os módulos que só dizem respeito à parte de gestão de processos (do outro estagiário) aparecem a azul na vista de módulos mas não são elaborados neste relatório, sendo que os módulos que influenciam os dois estágios são apresentados, e são da responsabilidade de ambos os estagiários, que irão dividir o trabalho relativo a estes entre si. Os módulos identificados como fazendo parte do Sistema Geral são módulos relativos a API's já existentes próprias do Drupal, e que portanto terão de ser adaptados para as nossas necessidades utilizando os mecanismos do Drupal para tal. Isto será feito sem alterar as API's do Drupal em si, mas sim adicionando conteúdos a estas por meio de hooks em módulos nossos.

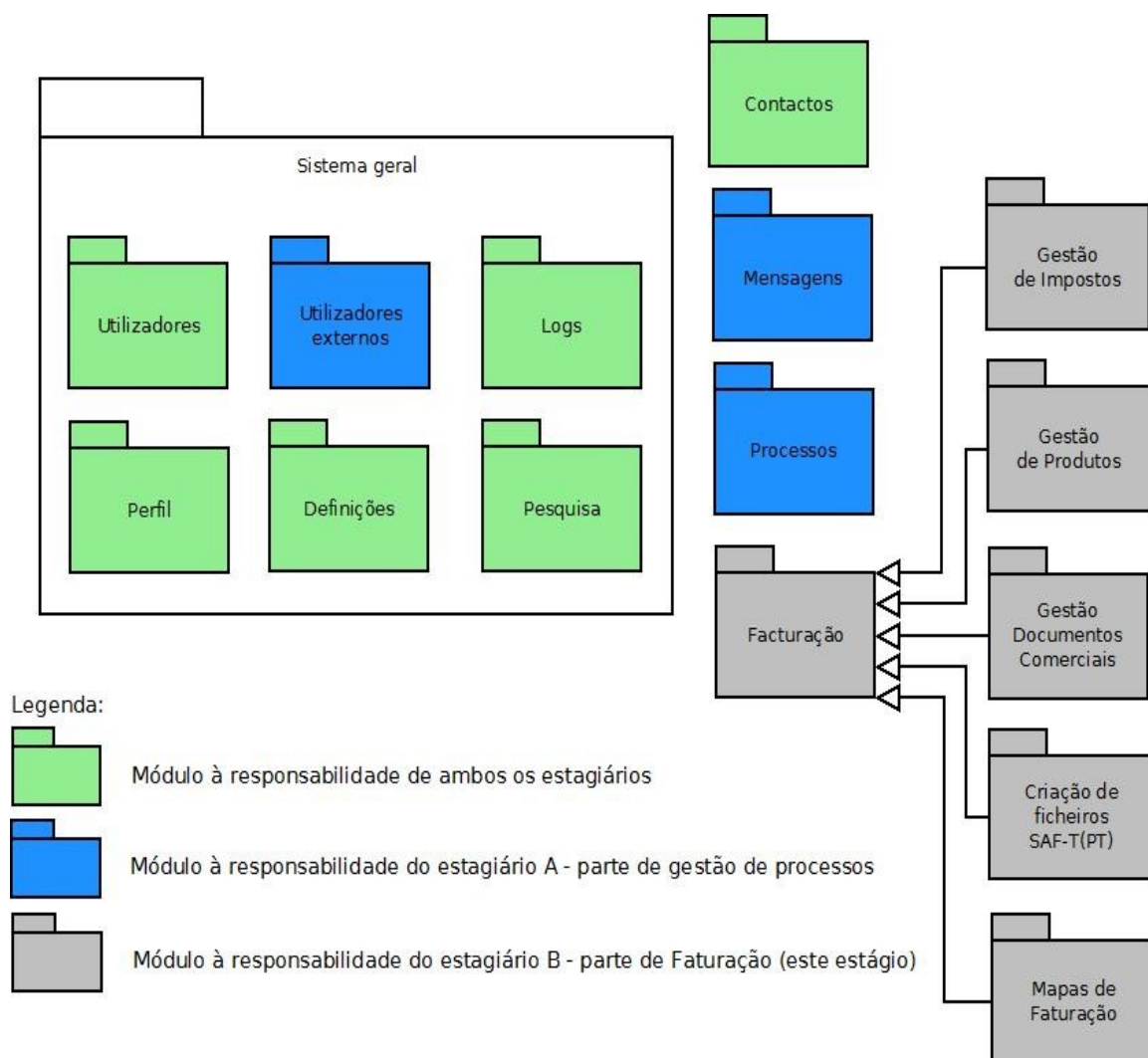


Figura 19: Vista de Módulos da aplicação FactSegur

5.5. Descrição dos Módulos

Neste capítulo vamos especificar o que vai existir em cada módulo, sendo que normalmente os módulos irão conter a implementação de hooks para tratar do processamento de certos URL (páginas envolvidas) que vão conter uma certa funcionalidade da aplicação.

Nas tabelas em cada módulo aparece a cinzento uma breve descrição do módulo e informação relevante a este, depois aparece nas linhas da tabela cada funcionalidade que este implementa, o nome desta, a sua descrição e um breve resumo dos mecanismos usados na criação da página, e o nome da página ou páginas envolvidas com essa funcionalidade.

5.5.1. Módulo de Gestão de Documentos Comerciais

Nome da funcionalidade	Descrição	Página(s) Envolvida(s)
Este módulo permitirá a criação, edição, apagamento e listagem dos documentos comerciais.		
Listar Documentos comerciais	Serão utilizadas vistas (Drupal View API) de modo a construir a lista de documentos existentes na base de dados cujo utilizador atual tem permissões para ver, e as vistas incluem opções de filtragem para facilitar a navegação.	/invoice/viewlist
Adicionar documento	Os dados a preencher serão representados usando o sistema de forms do Drupal. Os forms serão depois encaminhados para a camada de tema do drupal de modo a transformar o array representando o form e seus campos em código html com o aspeto desejado. O sistema de validação da API de forms também será extensivamente utilizado de modo a garantir que o documento ao ser submetido contém todos os dados necessários preenchidos, e que não existe nenhum conflito ou quebra de alguma regra de certificação ou de criação de documentos. Será usada a API própria do Drupal para comunicação com a base de dados, de modo a guardar os dados do novo documento depois de feita a validação. Neste fase é também criada a assinatura digital do documento através de código php e o openssl.	/invoice/add
Ver Documento	Os dados do documento comercial são obtidos da base de dados através da API de comunicação com base de dados do Drupal, e estes são adicionados ao array de conteúdo. O array de conteúdo é então enviado para a API de Tema para ser transformado numa página HTML com determinado aspeto.	/invoice/view/{docnumber}
Apagar Documento	Existirá uma função para anular documentos, visto estes não poderem ser apagados definitivamente. A razão de anulação deverá ser fornecida. Deverá ser acrescentado o botão de anulação do documento nas outras páginas deste módulo, não existindo assim uma página própria para apagar documentos, apenas atalhos para anulação noutras páginas.	Em: → /invoice/viewlist/{docnumber} → /invoice/view/{docnumber}

Tabela 7: Módulo de Gestão de Documentos Comerciais

5.5.2. Módulo de Gestão de Produtos

Nome da funcionalidade	Descrição	Página Envolvida
Este módulo permitirá a criação, edição, apagamento e listagem dos variados produtos, que irão ser utilizados nos documentos comerciais nas linhas de produtos. Essas linhas de produtos representam os produtos que estão envolvidos no documento comercial, bem como os seus detalhes, como unidade de medida, preço, desconto, entre outros.		
Listar Produtos	Serão utilizadas vistas (Drupal View API) de modo a construir a lista de produtos existentes cujo utilizador atual tem permissões para ver, e as vistas incluem opções de filtragem para facilitar a navegação.	/product/viewlist
Adicionar Produto	Os dados a preencher serão representados usando o sistema de forms do Drupal. Os forms serão depois encaminhados para a camada de tema do drupal de modo a transformar o array representando o form e seus campos em código html com o aspeto desejado. O sistema de validação da API de forms será utilizado de modo a garantir que a ficha de produto ao ser submetido contém todos os dados necessários preenchidos, e que não existe nenhum problema com este. Será usada a API própria do Drupal para comunicação com a base de dados, de modo a guardar os dados da nova ficha de produto depois de feita a validação.	/product/add
Editar Produto	Os dados do produto existente serão obtidos através da API de base de dados do Drupal, depois estes serão descritos num form do Drupal (em array). O form será depois enviado já com os dados para a API de Tema para ser transformado numa página HTML com o aspeto desejado. O utilizador poderá então alterar os campos necessários e submeter o form de modo a gravar as alterações, caso a validação seja concluída com sucesso.	/product/edit/{prodnumber}
Ver Produto	Os dados da ficha de produto são obtidos da base de dados através da API de comunicação com base de dados do Drupal, e estes são adicionados ao array de conteúdo. O array de conteúdo é então enviado para a API de Tema para ser transformado numa página HTML com determinado aspeto.	/product/view/{prodnumber}

Apagar Produto	Existirá uma função para apagar produtos. Deverá ser acrescentado o botão para apagar (que chama essa função), nas outras páginas deste módulo, não existindo assim uma página própria para apagar documentos, apenas atalhos para anulação noutras páginas. Não deverá ser possível apagar um produto que tenha documentos associados não anulados.	Em: → /product/ viewlist → /product/edit/ {prodnumber} → /product/ view/ {prodnumber}
----------------	--	--

Tabela 8: Módulo de Gestão de Produtos

5.5.3. Módulo de Gestão de Impostos

Nome da funcionalidade	Descrição	Página Envolvida
Este módulo permitirá a criação, edição, apagamento e listagem das variadas taxas de imposto, que irão ser seleccionadas em cada elemento das linhas de produto, existentes nos documentos comerciais. Caso o produto seja isento de imposto, deve ser colocado taxa de imposto como isento, e indicada a razão de isenção.		
Listar Taxas de Imposto	Serão utilizadas vistas (Drupal View API) de modo a construir a lista de taxas de imposto existentes cujo utilizador atual tem permissões para ver, e as vistas incluem opções de filtragem para facilitar a navegação.	/tax/viewlist
Adicionar Taxa de Imposto	Os dados a preencher serão representados usando o sistema de forms do Drupal. Os forms serão depois encaminhados para a camada de tema do drupal de modo a transformar o array representando o form e seus campos em código html com o aspeto desejado. O sistema de validação da API de forms será utilizado de modo a garantir que a ficha de imposto ao ser submetido contém todos os dados necessários preenchidos, e que não existe nenhum problema com este. Será usada a API própria do Drupal para comunicação com a base de dados, de modo a guardar os dados da nova ficha de imposto depois de feita a validação.	/tax/add
Editar Taxa de Imposto	Os dados da ficha de imposto existente serão obtidos através da API de base de dados do Drupal, depois estes serão descritos num form do Drupal (em array). O form será depois enviado já com os dados para a API de Tema para ser transformado numa página HTML com o aspeto desejado. O utilizador poderá então alterar os campos necessários e submeter o form de modo a gravar as alterações, caso a validação seja concluída com sucesso.	/tax/edit/ {taxnumber}
Ver Taxa de Imposto	Os dados da ficha de imposto são obtidos da base de dados através da API de comunicação com base de dados do Drupal, e estes são adicionados ao array de conteúdo. O array de conteúdo é então enviado para a API de Tema para ser transformado numa página HTML com determinado aspeto.	/tax/view/ {taxnumber}
Apagar Taxa de Imposto	Existirá uma função para apagar taxas de imposto. Deverá ser acrescentado o botão para apagar (que chama essa função), nas outras páginas deste módulo, não existindo assim uma página própria para apagar documentos, apenas atalhos para anulação noutras páginas. Não deverá ser possível apagar uma taxa que tenha documentos associados não anulados.	Em: → /tax/viewlist → /tax/edit/ {taxnumber} → /tax/view/ {taxnumber}

Tabela 9: Módulo de Gestão de Impostos

5.5.4. Módulo de Criação de Ficheiros SAF-T(PT)

Nome da funcionalidade	Descrição	Página Envolvida
Este módulo permitirá a criação, edição, apagamento e listagem das variadas taxas de imposto, que irão ser seleccionadas em cada elemento das linhas de produto, existentes nos documentos comerciais. Caso o produto seja isento de imposto, deve ser colocado taxa de imposto como isento, e indicada a razão de isenção.		
Listar Taxas de Imposto	Será inicialmente criado um form que permita ao utilizador seleccionar a data de início e data de fim através de calendários. Os calendários aparecem automaticamente com a data de início sendo o início do mês anterior, e a data de fim o fim do mês anterior. Depois o form é enviado à API de tema do Drupal de modo a transformar este numa página html para o utilizador. O Utilizador define as datas e submete o form, e caso seja validado que as datas estão corretas, é gerado um ficheiro xml usando o módulo do Drupal Views Data Export, e importando os dados relevantes através da API de comunicação com a base de dados do Drupal. A página xml resultante aparece então ao utilizador para este descarregar.	/saftpt/generate

Tabela 10: Módulo de Criação de ficheiros SAF-T(PT)

5.5.5. Módulo de Mapas de Faturação

Nome da funcionalidade	Descrição	Página Envolvida
Este módulo permitirá a visualização de variadas consultas relativas a faturação, ver os documentos emitidos por data, procurar só documentos ainda não pagos, ver a lista de vendas agrupada por certos parâmetros (cliente, produto, etc), ver o volume de vendas num mês num ano, e ver o total de IVA a pagar num determinado período de tempo.		
Ver Lista de últimos Documentos Emitidos	Serão utilizadas vistas (Drupal View API) de modo a construir a lista de documentos existentes cujo utilizador atual tem permissões para ver, organizando por data, e as vistas incluem opções para customizar a pesquisa, permitindo procurar só dentro de alguns parâmetros.	/invoice/viewlast
Ver Documentos em aberto	Serão utilizadas vistas (Drupal View API) de modo a construir a lista de documentos existentes cujo utilizador atual tem permissões para ver, que ainda não estão pagos, logo cujo recibo ainda não tenha sido passado, ou tenha sido pago apenas parcialmente, e as vistas incluem opções para parametrizar a pesquisa, permitindo procurar só dentro de alguns parâmetros.	/invoice/viewopen
Ver Lista de Vendas Líquidas	Serão utilizadas vistas (Drupal View API) de modo a construir a lista de vendas com o valor total vendido por produto e por cliente, e as vistas incluem opções para parametrizar a pesquisa, permitindo procurar só dentro de alguns parâmetros. Pode ser agrupado só o valor faturado por cliente, só por produto, ou por ambos, de modo a obter os totais desejados de faturação.	/invoice/viewgrouped
Ver Volume de Faturação	Serão utilizadas vistas (Drupal View API) de modo a construir a lista de vendas com o valor total vendido por mês dentro de um ano, e as vistas incluem opções para parametrizar a pesquisa, permitindo procurar só dentro de alguns parâmetros. Pode ser mostrado só o valor faturado relativo a um cliente específico.	/invoice/viewvolume
Ver Taxas de Iva para um Período	Serão utilizadas vistas (Drupal View API) de modo a construir a lista de documentos, e o valor de imposto a pagar em cada um, bem como os totais de imposto a pagar, e as vistas incluem opções para parametrizar a pesquisa, permitindo procurar só dentro de alguns parâmetros.	/invoice/viewtaxes

Tabela 11: Módulo de Mapas de Faturação

5.5.6. Módulo de Contactos (Sistema geral)

Nome da funcionalidade	Descrição	Página Envolvida
Este módulo está a cargo de ambos os estagiários, sendo que o trabalho relativo a estes será dividido pelos dois, e permitirá a criação, edição, apagamento e listagem de contactos individuais, que representam contactos de pessoas pertencentes a uma empresa, ou contactos de empresa, que correspondem a um cliente a quem são passados os documentos comerciais. Isto indica que existirá um contacto de empresa (cliente) por defeito que é o consumidor final, o utilizador que não indicou nome, morada, nem número de contribuinte, e também que um contacto de empresa pode ser uma empresa ou mesmo um indivíduo, se a fatura for passada em nome dessa pessoa em vez de ser em nome de uma empresa.		
Listar Contactos Individuais	Serão utilizadas vistas (Drupal View API) de modo a construir a lista de contactos individuais existentes cujo utilizador atual tem permissões para ver, e as vistas incluem opções de filtragem para facilitar a navegação.	/contact/viewlist
Adicionar Contacto Individual	Os dados a preencher serão representados usando o sistema de forms do Drupal. Os forms serão depois encaminhados para a camada de tema do drupal de modo a transformar o array representando o form e seus campos em código html com o aspeto desejado. O sistema de validação da API de forms será utilizado de modo a garantir que a ficha de imposto ao ser submetido contém todos os dados necessários preenchidos, e que não existe nenhum problema com este. Será usada a API própria do Drupal para comunicação com a base de dados, de modo a guardar os dados da nova ficha de contacto individual depois de feita a validação.	/contact/add
Editar Contacto Individual	Os dados da ficha de contacto individual existente serão obtidos através da API de base de dados do Drupal, depois estes serão descritos num form do Drupal (em array). O form será depois enviado já com os dados para a API de Tema para ser transformado numa página HTML com o aspeto desejado. O utilizador poderá então alterar os campos necessários e submeter o form de modo a gravar as alterações, caso a validação seja concluída com sucesso.	/contact/edit/{contnumber}
Ver Contacto Individual	Os dados da ficha de contacto individual são obtidos da base de dados através da API de comunicação com base de dados do Drupal, e estes são adicionados ao array de conteúdo. O array de conteúdo é então enviado para a API de Tema para ser transformado numa página HTML com determinado aspeto.	/contact/view/{contnumber}
Apagar Contacto Individual	Existirá uma função para apagar fichas de cliente individual. Deverá ser acrescentado o botão para apagar (que chama essa função), nas outras páginas deste módulo, não existindo assim uma página própria para apagar documentos, apenas atalhos para anulação noutras páginas.	Em: → /contact/viewlist → /contact/edit/{contnumber} → /contact/view/{contnumber}
Listar Contactos de Empresa	Serão utilizadas vistas (Drupal View API) de modo a construir a lista de contactos de empresa (clientes de faturação) existentes cujo utilizador atual tem permissões para ver, e as vistas incluem opções de filtragem para facilitar a navegação.	/customer/viewlist
Adicionar Contacto de Empresa	Os dados a preencher serão representados usando o sistema de forms do Drupal. Os forms serão depois encaminhados para a camada de tema do drupal de modo a transformar o array representando o form e seus campos em código html com o aspeto desejado. O sistema de validação da API de forms será utilizado de modo a garantir que a ficha de imposto ao ser submetido contém todos os dados necessários preenchidos, e que não existe nenhum problema com este. Será usada a API própria do Drupal para comunicação com a base de dados, de modo a guardar os dados da nova ficha de contacto de empresa (clientes de faturação) depois de feita a validação. Não deverá ser possível o uso de um número de contribuinte já existente na base de dados.	/customer/add
Editar Contacto de Empresa	Os dados da ficha de contacto de empresa (cliente de faturação) existente serão obtidos através da API de base de dados do Drupal, depois estes serão descritos num form do Drupal (em array). O form será depois enviado já com os dados para a API de Tema para ser transformado numa página HTML com o aspeto desejado. O utilizador poderá então alterar os campos necessários e submeter o form de modo a gravar as alterações, caso a validação seja concluída com sucesso.	/customer/edit/{custnumber}
Ver Contacto de Empresa	Os dados da ficha de contacto de empresa (clientes de faturação) são obtidos da base de dados através da API de comunicação com base de dados do Drupal, e estes são adicionados ao array de conteúdo. O array de conteúdo é então enviado para a API de Tema para ser transformado numa página HTML com determinado aspeto.	/customer/view/{custnumber}

Apagar Contacto de Empresa	Existirá uma função para apagar fichas de cliente de empresa (cliente de faturação). Deverá ser acrescentado o botão para apagar (que chama essa função), nas outras páginas deste módulo, não existindo assim uma página própria para apagar documentos, apenas atalhos para anulação noutras páginas.	Em: → /customer/ viewlist → /customer/edit/ {custnumber} → /customer/ view/ {custnumber}
----------------------------------	---	--

Tabela 12: Módulo de Gestão de Contactos (Sistema geral)

5.5.7. Módulo de Gestão de Utilizadores (Sistema geral)

Nome da funcionalidade	Descrição	Página Envolvida
Este módulo está a cargo de ambos os estagiários, sendo que o trabalho relativo a estes será dividido pelos dois. Este módulo contém dois tipos de utilizadores, os utilizadores internos, que são os trabalhadores da empresa que gere a sua faturação e gestão de processos com este programa. Também existem os utilizadores externos, que são os membros de um cliente da empresa, cujos membros podem ter um utilizador para entrar dentro do programa e poder ver apenas os seus processos. O sistema de permissões do Drupal deve estar configurado para bloquear o acesso a estes utilizadores externos a todo o outro conteúdo, como os dados de faturação da empresa, e acesso/visualização de processos relativos a outros clientes. Estas permissões são atribuídas utilizando um sistema de níveis de acesso (roles) em que em cada utilizador será definido um nível de acesso, e este indica as suas permissões.		
Listar Utilizadores Internos	Serão utilizadas vistas (Drupal View API) de modo a construir a lista de utilizadores internos existentes cujo utilizador atual tem permissões para ver, e as vistas incluem opções de filtragem para facilitar a navegação.	/user/viewlist
Adicionar Utilizador Interno	Os dados a preencher serão representados usando o sistema de forms do Drupal. Será obtido o form por defeito do Drupal já existente para adicionar utilizadores, e a este por meio de um hook implementado num módulo nosso (sem mudar o código desse form no sistema core do Drupal), e iremos adicionar novos campos de acordo com as nossas necessidades. O form continuará a passar pelo sistema de validação de forms do drupal (um nosso, já não será a validação de utilizadores do Drupal) e pela camada de Tema de modo a passar de um form a uma página html.	/user/add
Editar Utilizador Interno	Será obtido o form por defeito do Drupal relativo à edição de utilizadores, e serão acrescentados os dados extra adicionados por nós através da API de base de dados do Drupal e o uso de tabelas criadas por nós ligadas à tabela de utilizadores própria do Drupal a partir do id do utilizador que o Drupal atribuiu. O form será depois enviado já com os dados para a API de Tema para ser transformado numa página HTML com o aspeto desejado. O utilizador poderá então alterar os campos necessários e submeter o form de modo a gravar as alterações, caso a validação seja concluída com sucesso.	/user/edit/ {usernumber}
Apagar Utilizador Interno	Existirá uma função para apagar utilizadores, usando os mecanismos do drupal para remoção de utilizadores, e será possível fazê-lo a partir de várias páginas.	Em: → /user/ viewlist → /user/edit/ {usernumber} → /user/view/ {usernumber}
Listar Utilizadores Externos	Serão utilizadas vistas (Drupal View API) de modo a construir a lista de utilizadores internos existentes cujo utilizador atual tem permissões para ver, e as vistas incluem opções de filtragem para facilitar a navegação.	/company/ viewlist
Adicionar Utilizador Externo	Os dados a preencher serão representados usando o sistema de forms do Drupal. Será obtido o form por defeito do Drupal já existente para adicionar utilizadores, e a este por meio de um hook implementado num módulo nosso (sem mudar o código desse form no sistema core do Drupal), e iremos adicionar novos campos de acordo com as nossas necessidades. O form continuará a passar pelo sistema de validação de forms do drupal (um nosso, já não será a validação de utilizadores do Drupal) e pela camada de Tema de modo a passar de um form a uma página html.	/company/add
Editar Utilizador Externo	Será obtido o form por defeito do Drupal relativo à edição de utilizadores, e serão acrescentados os dados extra adicionados por nós através da API de base de dados do Drupal e o uso de tabelas criadas por nós ligadas à tabela de utilizadores própria do Drupal a partir do id do utilizador que o Drupal atribuiu. O form será depois enviado já com os dados para a API de Tema para ser transformado numa página HTML com o aspeto desejado. O utilizador poderá então alterar os campos necessários e submeter o form de modo a gravar as alterações, caso a validação seja concluída com sucesso.	/company/edit/ {compnumber}

Apagar Utilizador Externo	Existirá uma função para apagar utilizadores, usando os mecanismos do drupal para remoção de utilizadores, e será possível fazê-lo a partir de várias páginas.	Em: → /company/ viewlist → /company/ edit/ {compnumber} → /company/ view/ {compnumber}
Listar Nível de Acesso	Será usada uma Vista (View) com a listagem de níveis de acesso (roles) definidos no sistema. O Drupal já tem o mecanismo de níveis de acesso implementado, logo apenas precisamos de configurar os roles que necessitamos.	/role/viewlist
Adicionar Nível de Acesso	Será obtido o form do Drupal já existente para adicionar novos níveis de acesso (roles), e este será alterado de acordo com as necessidades, dentro de um módulo nosso, sem alterar esse form no código interno do Drupal.	/role/add
Editar Nível de Acesso	Será obtida a lista de permissões para um determinado nível de acesso através do mecanismo para isso já existente no drupal, e a este faremos as alterações necessárias de acordo com as nossas necessidades, dentro de um módulo nosso, sem alterar esse form no código interno do Drupal.	/role/edit
Apagar Nível de Acesso	Existirá uma função para apagar níveis de acesso, usando os mecanismos do drupal para remoção de níveis de acesso, e será possível fazê-lo a partir de várias páginas.	/role/delete/ {rolenumber}

Tabela 13: Módulo de Gestão de Utilizadores (Sistema geral)

5.5.8. Módulo de Definições (Sistema geral)

Não existirá um módulo que indique os dados existentes na página de definições. Como as definições existentes dependem dos variados módulos (pois cada módulo pode ou não precisar das suas próprias definições), cada módulo poderá definir dentro de si um Hook, que irá inserir na página de definições uma nova secção onde podem ser configuradas os campos que este necessita. Assim sempre que um módulo é adicionado ou alterado, não é necessário adicionar/alterar na página de definições as mudanças (pois esta nem existe), mas apenas configurar no hook desse módulo de modo a adicionar as secções relativas a este. Os hooks relativos aos elementos de definições de outros módulos permanecem assim inalterados. Em relação à parte de faturação, as definições irão conter os elementos definidos no módulo de gestão de impostos e gestão de produtos, bem como uma página com definição de valores pré-definidos para novos documentos comerciais (equivalente à tabela “sourcedocdefaults” no modelo de dados.

5.5.9. Módulo de Pesquisa (Sistema geral)

Tal como acontece no módulo de definições as pesquisas não estão definidas num único sítio. As pesquisas no Drupal são Vistas (Views), e podem ser acrescentadas ao Drupal por meio do sistema de Hooks, que notificam o módulo de Vistas que contém vistas de pesquisa, e estas podem ser utilizadas. Cada módulo pode implementar as suas próprias vistas, e assim os dados relativos a estes podem ser pesquisados e acedidos. Logo não necessitamos de implementar as vistas todas numa zona do software, mas sim em cada módulo implementar só as vistas relativas a este. Mais uma vez isto favorece a independência entre o módulo e o resto do programa.

5.5.10. Módulo de Logs (Sistema geral)

O Drupal já contém um módulo de Logs, sendo que cada módulo vai ter de gerar os dados que quer inserir no Log (a partir de dados na base de dados ou de acordo com o pedido efetuado ou problema encontrado na altura). Esses dados são enviados para o sistema de Logs do Drupal, e caso necessário a este pode ser dado um aspeto diferente do aspeto básico do Drupal utilizando a API de temas do Drupal, de modo a oferecer o aspeto definido na análise de requisitos.

5.5.11. Tema do Drupal (Sistema geral)

O Drupal contém um conjunto de Temas já implementados por defeito, que oferecem aspetos base de como os elementos são apresentados na página. Também já existem variados temas gratuitos construídos pela comunidade do Drupal, sendo estes temas independentes das páginas dos websites, oferecendo um aspeto genérico, sem ser preciso em cada página ter as suas próprias regras de renderização.

No módulo poderiam ser definidos em cada página templates de apresentação que substituíssem o aspeto definido no tema atual do Drupal para a página, podendo adequar um form a um aspeto específico (posição dos elementos, css, etc.), diferente do aspeto genérico que um form iria ter. No nosso caso os módulos irão passar os dados que querem apresentar (normalmente será em forms), e utilizaremos um tema genérico para definir a transformação destes elementos em código HTML, mantendo o seu aspeto consistente entre todas as páginas.

5.6. Modelo de Dados

É necessário definir o modelo de dados, pois este vai indicar a representação dos dados na base de dados, e a forma como estes podem ser guardados e obtidos, por meio da API do Drupal de comunicação com a base de dados.

Seguem-se a lista de tabelas existentes, bem como a descrição desta.

Em cada campo da tabela pode ser visto o nome deste, a informação do que este contém, e por fim o tipo de dados, se é único (unique, não pode ser repetido noutras linhas da mesma tabela), e se é uma chave primária da tabela (PK), ou se é uma chave secundária que efetua a ligação a outra tabela e o nome da tabela a que este está ligado (FK nome).

Em primeiro lugar convém explicar o mecanismo de gestão de utilizadores e de contactos porque é o mais complexo, envolvendo várias tabelas.

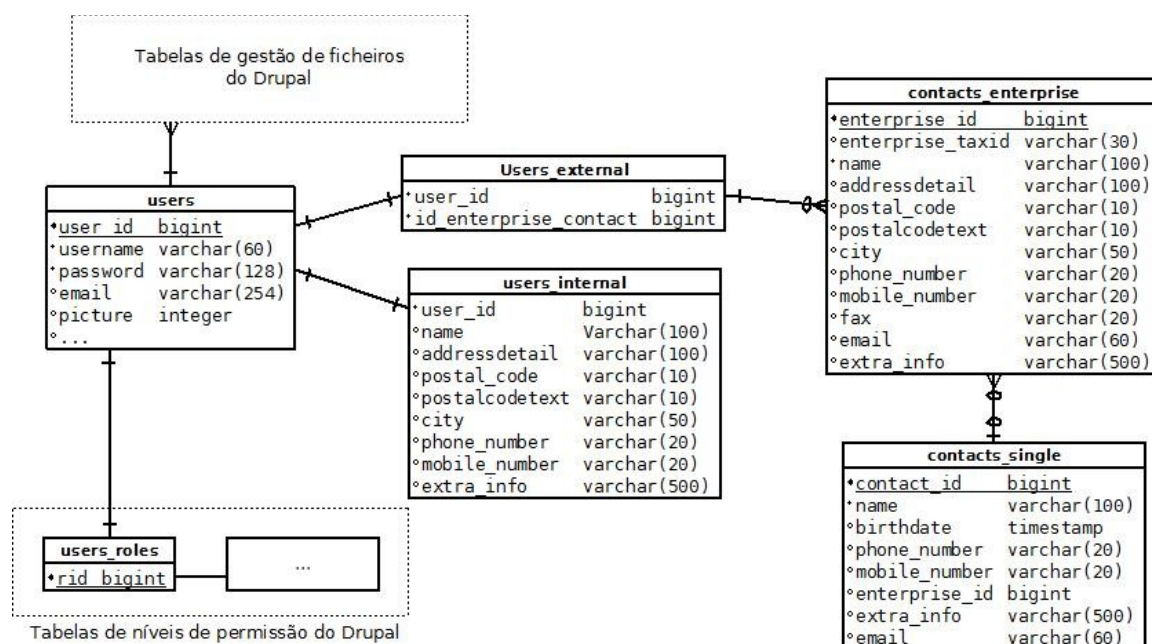


Figura 20: Modelo Entidades-Relacionamento das tabelas sobre os utilizadores

O Drupal já contém tabelas próprias com dados sobre os utilizadores que podem efetuar login no programa (Tabela Users). Não queremos modificar esta tabela (porque poderíamos inserir erros no funcionamento do Drupal), mas queremos adicionar novos campos de dados aos utilizadores. Para isso iremos usar uma nova tabela chamada tabela users_internal, que vai usar o mesmo id nesta que o id do utilizador na tabela Users, fazendo assim a ligação (Foreign Key) entre uma tabela e outra. Os dados extra do utilizador serão então guardados na tabela users_internal, sem alterar a tabela base do Drupal.

Existe também o caso de utilizadores com login no programa que não pertencem à empresa, mas sim a clientes que podem entrar no programa e ver apenas os dados dos seus processos, e obter documentos relativos à sua empresa. A empresa a que estes utilizadores pertencem está definida na tabela contacts_enterprise, e a ligação entre o utilizador e a empresa é feita através da tabela users_external, que faz a relação entre o id do utilizador na tabela users (Foreign Key), e o id da empresa a que este pertence na tabela users_external (Foreign Key). Deste modo será possível saber a qual empresa os utilizadores pertencem, e restringir o acesso destes para poderem só consultar os processos e documentos relativos a esta.

De grande importância é também a tabela sourcedocs, porque contém todos os dados dos documentos comerciais, e a maioria do trabalho do módulo de faturação estará centrado no uso desta tabela. A tabela linesourcedocs representa em cada linha o detalhe sobre um produto/serviço existente em um documento comercial, sendo ligado a este onde coincidir ao mesmo tempo o número, série e tipo do documento (Foreign keys).

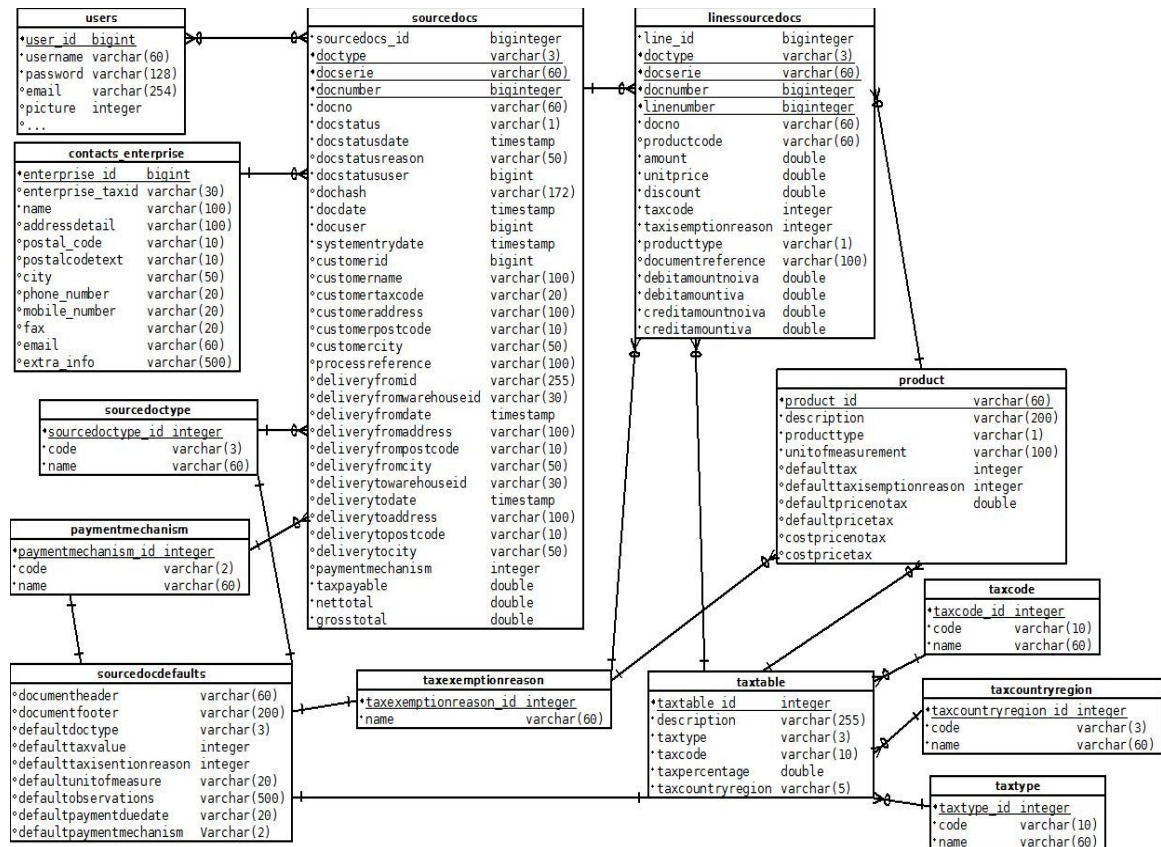


Figura 21: Modelo Entidades-Relacionamento das tabelas de faturação

Nas secções seguintes vai ser apresentado com maior detalhe as tabelas deste modelo de dados, e o que cada campo da tabela representa, de modo a saber os dados que estes podem conter.

5.6.1. Tabela “users”

Nome do Campo	Descrição	Tipo de dados
Esta tabela pertence ao Drupal, e não vamos efetuar a ela nenhuma alteração, sobre o risco de inserir erros no funcionamento do Drupal. Aqui são mostrados os dados relevantes à nossa aplicação.		
user_id	Código numérico que identifica o utilizador.	Bigint, unique, PK
username	Nome com que o utilizador efetua login no Drupal.	Varchar(60)
password	Palavra passe com que o utilizador efetua login no Drupal.	Varchar(128)
email	Email do utilizador.	Varchar(254)
picture	Chave que identifica a imagem do utilizador numa tabela do Drupal para gestão de ficheiros.	Integer, FK (file_usage)

Tabela 14: Tabela “users”

5.6.2. Tabela “users_internal”

Nome do Campo	Descrição	Tipo de dados
Esta tabela permite guardar os dados relativos aos utilizadores com login no programa. Esta tabela vai permitir configurar para cada utilizador dados extra, adicionando dados novos para além dos dados que já existem no drupal para cada utilizador (numa tabela à parte, própria do Drupal chamada Users).		
user_id	Código numérico que identifica o utilizador. Este é o mesmo id que o utilizador tem na tabela users.	Bigint, unique, PK, FK (users)
name	Nome do utilizador.	Varchar(100)

addressdetail	Morada do utilizador.	Varchar(100)
postalcode	Localidade do utilizador.	Varchar(10)
postalcodetext	Caso não exista código postal na lista auxiliar para autopreenchimento, podemos preencher à mão.	Varchar(10)
city	Localidade do utilizador.	Varchar(50)
phone_number	Número de telefone do utilizador.	Varchar(20)
mobile_number	Número de telefone do utilizador.	Varchar(20)
fax	Número de fax do utilizador.	Varchar(20)
email	Endereço de email do utilizador.	Varchar(60)
otherinformation	Texto com dados complementares sobre o utilizador.	Varchar(500)

Tabela 15: Tabela “users_internal”

5.6.3. Tabela “users_external”

Nome do Campo	Descrição	Tipo de dados
Esta tabela permite identificar utilizadores que não pertencem à empresa que comprou o software, mas sim a trabalhadores de empresas clientes desta, que têm acesso ao sistema para poder consultar dados da sua empresa, como os processos relativos à sua empresa. Esta tabela permite fazer a ligação entre um utilizador do drupal e qual é a empresa associada a este.		
user_id	Código numérico que identifica o utilizador.	Bigint, unique, PK, FK (contacts_enterprise)
id_enterprise_contact	Código numérico que identifica a empresa a que o utilizador pertence.	Bigint, FK (contacts_enterprise)

Tabela 16: Tabela “users_external”

5.6.4. Tabela “contacts_enterprise”

Nome do Campo	Descrição	Tipo de dados
Esta tabela permite guardar os dados relativos aos clientes do programa. Estes dados são usados para preenchimento dos dados do cliente nos documentos comerciais (faturas, recibos, etc).		
enterprise_id	Código numérico que identifica o cliente/empresa.	Varchar(30), unique, PK
enterprise_taxid	Nº de contribuinte do cliente, consumidor final será “999999990”.	Varchar(20), unique
name	Nome do cliente / empresa.	Varchar(100)
addressdetail	Morada do cliente / empresa.	Varchar(100)
postalcode	Localidade do cliente / empresa.	Varchar(10)
postalcodetext	Caso não exista código postal na lista auxiliar para autopreenchimento, podemos preencher à mão.	Varchar(10)
city	Localidade do cliente / empresa.	Varchar(50)
phone_number	Número de telefone do cliente / empresa.	Varchar(20)
mobile_number	Número de telefone do cliente / empresa.	Varchar(20)
fax	Número de fax do cliente / empresa.	Varchar(20)
email	Endereço de email do cliente / empresa.	Varchar(60)
extra_info	Texto com dados complementares sobre o cliente /empresa.	Varchar(500)

Tabela 17: Tabela “contacts_enterprise”

5.6.5. Tabela “contacts_single”

Nome do Campo	Descrição	Tipo de dados
Esta tabela permite guardar os dados relativos aos contactos do programa. Estes representam pessoas (não empresas) que fazem parte das empresas (contacts_enterprise).		
contact_id	Código numérico que identifica o contacto individual.	Bigint, unique, PK
name	Nome do contacto individual.	Varchar(100)
birthdate	Data do aniversário do contacto individual.	Timestamp
email	Endereço de email do cliente / empresa.	Varchar(60)
telephone	Número de telefone do cliente / empresa.	Varchar(20)
mobilephone	Número de telemóvel do cliente / empresa.	Varchar(20)
enterprise_id	Identificação da ficha de cliente / empresa ao qual este contacto individual pertence. Caso esteja vazio o contacto não faz parte de nenhuma.	Bigint, (FK contacts_enterprise)
extra_info	Texto com dados complementares sobre o cliente /empresa.	Varchar(500)

Tabela 18: Tabela “contacts_single”

5.6.6. Tabela “product”

Nome do Campo	Descrição	Tipo de dados
Esta tabela permite guardar os dados relativos aos produtos do programa. Estes dados são usados para preenchimento dos dados do produto nos documentos comerciais (faturas, recibos, etc).		
product_id	Código que identifica o produto.	Varchar(60), unique, pk
description	Nome/Descrição do produto.	Varchar(200), unique
producttype	Código identificativo do tipo de produto (P=Produtos, S=serviços, O=Outros).	Varchar(1)
unitofmeasurement	Unidade de Medida do produto, poderá ser unidade, hora, quilograma, metro, pode ser escrito qualquer coisa desde que identifique a unidade...	Varchar(100)
defaulttax	Imposto que aparece pré-seleccionado para este produto.	Int, (FK taxtable)
defaulttaxisemptionreason	Quando existe isenção de imposto neste produto, esta é a razão de isenção que aparece pré-seleccionada.	Int, (FK taxexemptionreason)
defaultpricenotax	Preço por defeito para o produto sem imposto.	Double
defaultpricetax	Preço por defeito para o produto com o imposto.	Double
costpricenotax	Preço de custo do produto à empresa sem o imposto.	Double
costpricetax	Preço de custo do produto à empresa com o imposto.	Double

Tabela 19: Tabela “product”

5.6.7. Tabela “sourcedocs”

Nome do Campo	Descrição	Tipo de dados
Esta tabela permite guardar os dados relativos aos documentos comerciais, como faturas, recibos, e todos os outros. Cada linha representando um produto numa fatura estará representado na tabela sourcedoclines, logo cada linha desta tabela estará ligada a uma ou mais linhas dessa tabela, dependendo do número de produtos existentes no documento.		
sourcedocs_id	Código que identifica o produto (composto pelo tipo de documentos, série e número).	Bigint, unique
doctype	Código identificativo do tipo de documento comercial (FT=Factura, RC=Recibo, etc – ver tabela sourcedoctype).	Varchar(3), PK , (FK sourcedoctype)
docserie	Código identificativo da série do documento comercial. Faz parte da identificação do documento comercial.	Varchar(60), PK
docnumber	Número do documento. Este número deverá ser único dentro da sua série e do seu tipo de documento.	Bigint, PK

docno	Identificação do documento, composta pelo tipo, série e número.	Varchar(60), unique
docstatus	Estado do documento (N=Normal, A=Documento Anulado).	Varchar(1)
docstatusdate	Data da última alteração ao estado do documento.	Timestamp
docstatusreason	Razão da última alteração ao estado do documento. Caso estado seja documento anulado, a razão deve sempre estar preenchida.	Varchar(50)
docstatususer	Utilizador responsável pela última alteração ao estado do documento.	Int, (FK users)
dochash	Assinatura digital com a chave privada do produtor de software segundo os termos da Portaria 363/2010, de 23 de Junho e suas alterações. O campo deve ser preenchido com 0, caso exista isenção da certificação de software.	Varchar(172)
docdate	Data da emissão do documento comercial	timestamp
docuser	Utilizador responsável pela criação do documento.	Int, (FK users)
systementrydate	Data da criação do documento	timestamp
customerid	Código de identificação do cliente na tabela customers.	Int, (FK contacts_enterprise)
customername	Nome que identifica o cliente na altura de criação do documento. Pode ser diferente do existente na tabela customers se desde essa altura o nome foi alterado.	Varchar(100)
customertaxcode	Número de contribuinte do cliente na altura de criação do documento. Pode ser diferente do existente na tabela customers se desde essa altura este elemento foi alterado.	Varchar(20)
customeraddress	Morada do cliente na altura de criação do documento. Pode ser diferente do existente na tabela customers se desde essa altura este elemento foi alterado.	Varchar(100)
customerpostcode	Código postal do cliente na altura de criação do documento. Pode ser diferente do existente na tabela customers se desde essa altura este elemento foi alterado.	Varchar(10)
customercity	Localidade do cliente na altura de criação do documento. Pode ser diferente do existente na tabela customers se desde essa altura este elemento foi alterado.	Varchar(50)
processreference	Referência ao processo associado a este documento. Neste campo poderá ser indicada a relação do documento comercial aos processos configurados na parte de gestão de processos deste projeto. Alguns campos serão preenchidos na fatura de acordo com o existente no processo (V(Processo, N° Apólice, Ramo, Segurado).	Varchar(100)
deliveryfromid	Identificação do meio de transporte, poderá incluir matrículas de veículos e outros dados.	Varchar(255)
deliveryfromwarehouseid	Identificação do armazém de origem.	Varchar(30)
deliveryfromdate	Data de início do transporte, a partir do armazém de origem.	timestamp
deliveryfromaddresses	Morada do armazém de origem.	Varchar(100)
deliveryfrompostcode	Código postal do armazém de origem.	Varchar(10)
deliveryfromcity	Localidade do armazém de origem.	Varchar(50)
deliverytowarehouseid	Identificação do armazém de destino.	Varchar(30)
deliverytodate	Data esperada para a chegada do transporte ao armazém de destino.	timestamp
deliverytoaddress	Morada do armazém de destino.	Varchar(100)
deliverytopostcode	Código postal do armazém de destino.	Varchar(10)
deliverytocty	Localidade do armazém de destino.	Varchar(50)
taxpayable	Valor total de imposto a pagar relativo a este documento.	double
netttotal	Valor total de crédito(income), descontando o valor do IVA.	double
grossttotal	Valor total de crédito(income), sem descontar o valor do IVA.	double

Tabela 20: Tabela “sourcedocs”

5.6.8. Tabela “linessourcedocs”

Nome do Campo	Descrição	Tipo de dados
Esta tabela permite guardar os dados relativos a produtos existentes nos documentos comerciais, como faturas, recibos, e todos os outros. Cada linha representa um produto num documento, podendo existir várias linhas relativas ao mesmo documento.		
line_id	Código numérico que identifica uma linha.	Bigint, unique
doctype	Código identificativo do tipo de documento comercial (FT=Factura, RC=Recibo, etc – ver tabela sourcedoctype).	Varchar(3), PK , (FK sourcedoctype)
docserie	Código identificativo da série do documento comercial. Faz parte da identificação do documento comercial.	Varchar(60), PK
docnumber	Número do documento. Este número deverá ser único dentro da sua série e do seu tipo de documento.	Int, PK
linenumber	Estado do documento (N=Normal, A=Documento Anulado).	Int, PK
docno	Identificação do documento, composta pelo tipo, série e número.	Varchar(60), unique
productcode	Código que identifica o produto na tabela de produtos.	Varchar(60), (FK product)
amount	Quantidade do produto.	double
unitprice	Preço do produto por unidade.	double
discount	Desconto existente em percentagem, quando aplicável.	double
taxcode	Imposto que aparece pré-selecionado para este produto.	Int, (FK taxtable)
taxisemptionreason	Quando existe isenção de imposto neste produto, aqui é indicada a razão de isenção selecionada.	Int, (FK taxexemptionreason)
producttype	Código identificativo do tipo de produto (P=Produtos, S=serviços, O=Outros).	Varchar(1)
documentreference	Referência a outros documentos, indicando de que outro documento esta linha está relacionada. Em notas de crédito e de crédito cada linha deve referenciar qual o documento que estamos a retificar nessa linha.	Varchar(100)
debitamountnoiva	Valor total de débito da linha sem o imposto.	double
debitamountiva	Valor total de débito da linha com o imposto.	double
creditamountnoiva	Valor total de crédito da linha sem o imposto.	double
creditamountiva	Valor total de crédito da linha com o imposto.	double

Tabela 21: Tabela “linessourcedocs”

5.6.9. Tabela “taxtable”

Nome do Campo	Descrição	Tipo de dados
Esta tabela permite guardar os tipos de imposto a usar na aplicação.		
taxtable_id	Código numérico que identifica o tipo de imposto.	int, unique, PK
description	Nome/Descrição do tipo de imposto, no caso de imposto do selo deve ser preenchido com a descrição da verba respectiva.	Varchar(255), unique
taxtype	Tipo do Imposto (IVA, IS, NS).	Varchar(3), (FK taxtype)
taxcode	Código do tipo de imposto, preenchido com um dos elementos existentes na tabela de apoio taxcode.	Varchar(10), (FK taxcode)
taxpercentage	Valor numérico da percentagem da taxa de imposto, ou do montante caso seja uma verba fixa de imposto de selo.	Double
taxcountryregion	Deve ser preenchido com o nome do país de acordo com a norma ISO 3166-1-alpha2 (PT=Portugal Continental).	Varchar(5), (FK taxcountryregion)

Tabela 22: Tabela “taxtable”

5.6.10. Tabela “taxtype”

Nome do Campo	Descrição	Tipo de dados
Tabela de apoio que identifica qual o tipo de imposto a pagar. (IVA=Imposto sobre o valor acrescentado, IS=Imposto do Selo, NS=Não sujeição a IVA ou IS).		
id	Código numérico que identifica o tipo de imposto.	int, unique, PK
code	Código de duas a três letras correspondente ao tipo de imposto.	Varchar(3), unique
name	Tipos de imposto em formato legível (Fatura, Fatura-Recibo, etc).	Varchar(60)

Tabela 23: Tabela “taxtype”

5.6.11. Tabela “taxcode”

Nome do Campo	Descrição	Tipo de dados
Tabela de apoio que identifica qual o código do tipo de imposto a pagar. (RED=Taxa Reduzida, INT=Taxa Intermédia, NOR=Taxa Normal, ISE=Isenta, OUT=Outros, aplicável para os regimes especiais de IVA, NS=Caso de Não sujeição).		
taxcode_id	Código numérico que identifica o código do tipo de imposto.	int, unique, PK
code	Código de duas a três letras correspondente ao código do tipo de imposto.	Varchar(10), unique
name	Código do tipo de imposto em formato legível (Taxa Reduzida, Taxa Normal, etc).	Varchar(60)

Tabela 24: Tabela “taxcode”

5.6.12. Tabela “taxcountryregion”

Nome do Campo	Descrição	Tipo de dados
Tabela de apoio que identifica a região ou o país do imposto. (PT=Portugal Continental, PT-AC=Espaço fiscal da Região Autónoma dos Açores, PT-MA=Espaço fiscal da Região Autónoma da Madeira).		
taxcountryregion_id	Código numérico que identifica o tipo de documento.	int, unique, PK
code	Código de duas letras correspondente ao tipo de documento comercial, ou 3 letras caso seja tipo de reposição de documento.	Varchar(3), unique
name	Tipos de pagamento de documentos comerciais em formato legível (Fatura, Fatura-Recibo, etc).	Varchar(60)

Tabela 25: Tabela “taxcountryregion”

5.6.13. Tabela “sourcedocdefaults”

Nome do Campo	Descrição	Tipo de dados
Esta tabela permite guardar os valores por defeito de variados campos dos documentos comerciais como faturas, recibos, etc.		
documentheader	Código numérico que identifica o produto.	Varchar(60), unique, PK
documentfooter	Nome/Descrição do produto.	Varchar(200), unique
defaultdoctype	Tipo por defeito dos novos documentos comerciais criados (FT, FR, NC, ND, RC, OR, GT).	Varchar(3), (FK sourcedoctype)
defaulttaxvalue	Valor por defeito de imposto num documento, representando a descrição de um elemento de imposto existente na tabela de imposto.	integer, (FK taxable)

defaulttaxisentionreason	Caso valor do imposto seja 0 (isento), deve ser indicado qual a razão por defeito para a existência de isenção.	integer, (FK taxexemptionreason)
defaultunitofmeasure	Valor por defeito para a unidade de medida, caso esteja não esteja definida na tabela de produtos, ou na criação de um novo produto.	Varchar(20)
defaultobservations	Valor por defeito para as observações ou comentários que aparecem na criação de novos documentos comerciais.	Varchar(500)
defaultpaymentdue date	Número de dias até o vencimento do documento.	Varchar(20)
defaultpaymentmechanism	Valor por defeito para o tipo de pagamento, numerário (NU), cheque (CH), cartão de crédito (CC), cartão de débito (CD), multibanco (MB), etc.	Varchar(2), (FK paymentmechanism)

Tabela 26: Tabela “sourcedocdefaults”

5.6.14. Tabela “sourcedoctype”

Nome do Campo	Descrição	Tipo de dados
<p>Tabela de apoio que contém os tipos disponíveis de documentos comerciais existentes. Os tipos cujo código tenha 3 letras são tipos de reposição de documentos que foram perdidos por motivo de reposição da base de dados, e têm de voltar a ser inseridos desta maneira.</p> <p>(FT=Fatura, FR=Fatura-Recibo, NC=Nota de Crédito, ND=Nota de Débito, RC=Recibo, RG=Outros Recibos, sem ser no âmbito do Regime de IVA de Caixa, FTD=Reposição de Fatura, FRD=Reposição de Fatura-Recibo, NCD=Reposição de Nota de Crédito, NDD=Reposição de Nota de Débito, RCD=Reposição de Recibo, RGD=Reposição de Outros Recibos).</p>		
sourcedoctype_id	Código numérico que identifica o tipo de documento.	int, unique, PK
code	Código de duas letras correspondente ao tipo de documento comercial, ou 3 letras caso seja tipo de reposição de documento.	Varchar(3), unique
name	Tipos de pagamento de documentos comerciais em formato legível (Fatura, Fatura-Recibo, etc).	Varchar(60)

Tabela 27: Tabela “sourcedoctype”

5.6.15. Tabela “taxexemptionreason”

Nome do Campo	Descrição	Tipo de dados
Razão de isenção de Imposto, caso taxa de imposto seja 0%(Isento).		
taxexemptionreason_id	Código numérico que identifica a razão de isenção de imposto.	int, unique, PK
name	Razão de isenção de imposto em formato legível, artigo 9º do CIVA, etc).	Varchar(60)

Tabela 28: Tabela “taxexemptionreason”

5.6.16. Tabela “paymentmechanism”

Nome do Campo	Descrição	Tipo de dados
<p>Tabela de apoio que contém os tipos disponíveis de pagamento de documentos comerciais</p> <p>(CC=cartão de crédito, CD=Cartão de Débito, CH=Cheque Bancário, CO=Cheque ou Cartão Oferta, CS=Compensação de Saldos em Conta Corrente, DE=Dinheiro Electrónico, LC=Letra Comercial, MB=Referência Multibanco, NU=Numerário, OU=Outros Meios, PR=Permuta de Bens, TB=Transferência Bancária ou débito direto, TR=Ticket Restaurante).</p>		
paymentmechanism_id	Código numérico que identifica o mecanismo de pagamento.	int, unique, PK
code	Código de duas letras correspondente ao mecanismo de pagamento de documento comercial (NU, CC, etc.).	Varchar(2), unique

name	Nome do mecanismo de pagamento de documentos comerciais em formato legível (Numerário, Cartão de Crédito, etc).	Varchar(60)
------	---	-------------

Tabela 29: Tabela “paymentmechanism”

5.7. Conclusão

Nesta fase de arquitetura ficaram definida os elementos que vão compor a nossa aplicação, e como estes vão trabalhar em conjunto de modo a podermos cumprir com o definido na análise de requisitos.

Foi feito um estudo do Drupal, visto ter sido escolhido para plataforma (framework) de desenvolvimento, e foi indicado como a nossa aplicação se vai inserir dentro do Drupal.

Existe uma definição do modelo de dados, já podendo saber quais as tabelas que vamos precisar de utilizar e o que é representado em cada uma delas, e também existe um detalhe dos módulos que acrescentam as funcionalidades, e o que será necessário implementar para cada um deles. Isto vai permitir a organização do trabalho, e vai permitir saber em que parte do código fonte encontrar a implementação de certa funcionalidade.

De seguida o próximo passo será passar à implementação.

6. Implementação

Este capítulo visa descrever como foram implementados os requisitos e arquitetura descritos nos capítulos anteriores.

É indicado o ambiente de desenvolvimento escolhido, o processo de desenvolvimento, a forma de usar o Drupal, de uma maneira específica os seus principais mecanismos. É indicado cada módulo, e para cada um destes um resumo do que consiste, quais foram os maiores desafios encontrados, e como estes foram superados.

Para cada módulo desenvolvido são apresentados diagramas ou esquemas exemplificativos da forma de funcionamento deste, uma tabela com as páginas (endereço) que esse módulo tem disponíveis de modo a oferecer funcionalidades, e também uma explicação em texto do seu funcionamento, bem como a indicação das maiores desafios encontrados, e a forma encontrada de os resolver.

Em relação a cooperação durante o desenvolvimento, de modo a organizar o trabalho, os módulos genéricos que seriam comuns a ambos foram divididos entre os dois estagiários, de forma a que todos os módulos referidos neste relatório de estágio foram exclusivamente implementados pelo estagiário ao qual este documento se refere.

Os módulos comuns desenvolvidos pelo estagiário do documento atual foram o módulo de Utilizadores Externos, o Módulo de Contactos, o Módulo de Permissões, e o Módulo de Definições. O outro estagiário ficou responsável pelo Módulo de Utilizadores Internos, o Módulo de Pesquisa Avançada, o Módulo de Logs, e também a camada de temas (que não é um módulo, é o que define o aspeto da aplicação). Tendo sido implementados pelo outro estagiário, esses módulos e o tema não são explicados no capítulo de implementação deste relatório.

6.1. Ambiente de desenvolvimento

O ambiente de desenvolvimento é de extrema importância, porque visa apoiar a implementação de modo a que esta decorra de forma organizada e agilizada. Este é composto principalmente pela infra-estrutura necessária, as ferramentas a utilizar, bem como todos os outros itens necessários para desenvolver e implantar o sistema, como por exemplo guias de programação, guias de teste, e outro tipo de documentação [65].

Na Figura 22 é chamada a atenção para o ambiente de desenvolvimento, onde cada estagiário trabalha no seu próprio computador, mas submete as alterações para um servidor que está a executar o Drupal, comum a ambos.

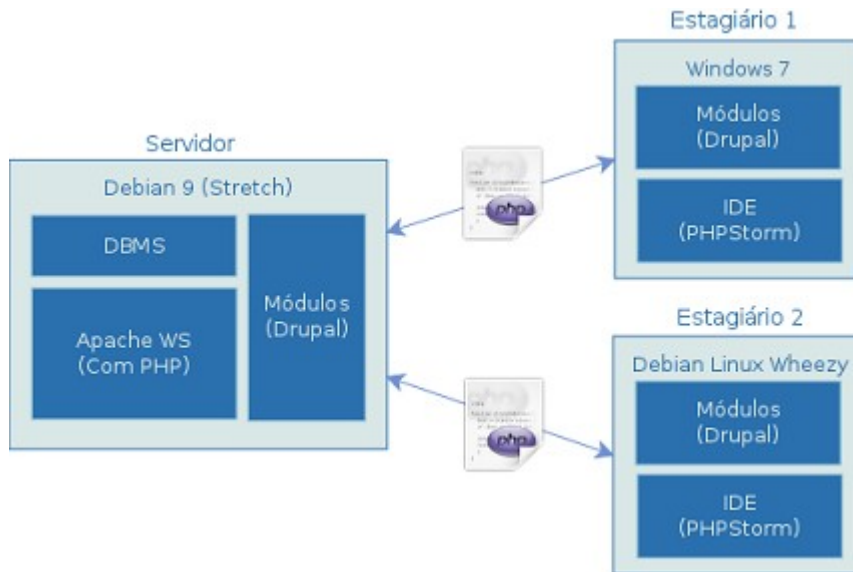


Figura 22: Infra-estrutura do ambiente de desenvolvimento

Neste caso, sendo uma aplicação baseada na Internet, de modo a aproximar o ambiente de desenvolvimento do ambiente real (produção), foi preparado um computador fixo dedicado para este efeito, com sistema operativo Linux (Debian), de modo a fornecer o serviço, através de um Servidor Web Apache, com PHP (versão 5), e uma base de dados PostgreSQL (versão 9.4.5).

Sendo que este estágio tem ligação a outro estágio a decorrer (em que este estágio é referente à parte de faturação, e o outro referente à parte de gestão de processos) foi necessário um ambiente comum de desenvolvimento, onde não existissem posteriormente conflitos relativos a ter sido usado versões diferentes do servidor web, ou base de dados, ou qualquer outra ferramenta. Neste esquema mencionado em cima ambos os estagiários executam o seu código no mesmo servidor, segundo as mesmas condições, assegurando a compatibilidade entre os seus módulos.

De modo a existir independência e minimizar conflitos relativos ao motor do Drupal durante o desenvolvimento foram instalados no mesmo servidor web 2 cópias do Drupal (com a mesma versão), e ambos os estagiários teriam a sua própria cópia. Essas cópias eram sincronizadas quando eram terminados módulos, e era testado a ver se não surgiam conflitos entre estes. A sincronização era feita simplesmente copiando a pasta do módulo que tinha sido trabalhado para a outra cópia do Drupal, visto os estagiários sempre terem trabalhado em módulos diferentes, mesmo os módulos do sistema geral foram divididos entre estes, de modo a cada um trabalhar um módulo diferente.

No lado dos estagiários, o desenvolvimento é feito no seu portátil utilizando o IDE PhpStorm, e o código é submetido pelo próprio PhpStorm para o servidor, sendo testado (executado) nesse servidor através de rede local e do navegador da Internet no seu computador (Google Chrome).

O PhpStorm foi escolhido por ambos os estagiários, por funcionar tanto em Windows

como em Linux, conter funcionalidades interessantes como upload do código para um servidor (como é necessário no nosso caso), autocomplete, code highlighting, ser bastante popular [66], e por ter licença gratuita para estudantes [67].

Depois de ser instalado o Apache com PHP e o motor de base de dados conforme indicado em cima, de seguida foi instalado o Drupal. A instalação deste é automatizada e explicada na documentação do Drupal [68]. Durante esta é principalmente pedido ao utilizador para este indicar o nome pretendido para o website drupal, e também os dados necessários para a ligação deste à base de dados (endereço IP, nome de utilizador, palavra passe).

A instalação da nossa aplicação no Drupal apenas envolve a cópia do código do website criado por nós para a pasta onde o Drupal foi instalado na diretoria “/sites/all”, e entrar na página do Drupal como administrador e ativar os módulos relevantes na lista de módulos do Drupal (que irá conter os módulos base do Drupal, bem como os módulos novos existentes nos ficheiros que copiámos). A ativação dos módulos faz automaticamente ocorrer a sua instalação, adicionando as tabelas de base de dados necessárias a estes, através do definido no ficheiro .install de cada módulo, não sendo necessário nenhum conhecimento destes por parte do utilizador. Isto implica que a nossa aplicação fica logo pronta a funcionar, não sendo necessário executar separadamente scripts de instalação e preenchimento de base de dados, ou outros tipos de configurações. Quaisquer configurações, dados iniciais e valores predefinidos nas tabelas são automaticamente inseridos pelo módulo durante a sua ativação.

Os módulos a ativar no nosso caso são todos os módulos da categoria Factsegur Modules.

A imagem em baixo 23 mostra a interface do Drupal para a ativação de módulos, em que apenas é necessário selecionar o(s) módulos que queremos ativar ou desativar, e o Drupal ativa esses módulos, bem como quaisquer outros módulos necessários ao funcionamento destes (dependências). Basta portanto carregar no checkbox do lado esquerdo, e o módulo está ativado.

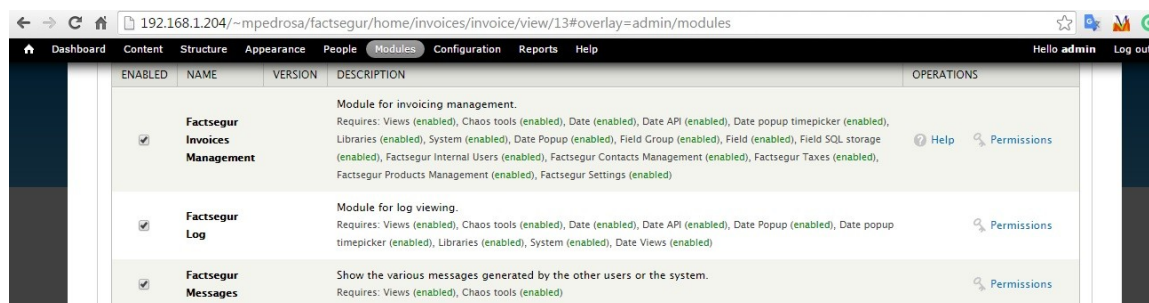


Figura 23: Ativação dos módulos no Drupal

6.2. Módulos do Drupal

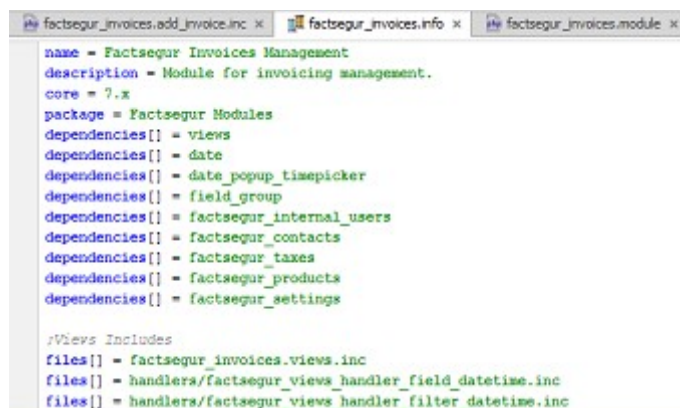
Aqui vamos dar uma pequena introdução à maneira de funcionamento dos módulos do Drupal, de modo a explicar os mecanismos do Drupal utilizados na implementação.

Cada módulo consiste numa pasta com um conjunto de ficheiros (e/ou outras pastas) que deve ser copiado para a pasta de instalação do Drupal, no caminho “drupal_instalacao/sites/all/modules”. A partir daí o Drupal procura automaticamente nas pastas existentes nesse caminho por ficheiros .info, e ao encontrar permite a ativação do módulo correspondente no seu interface de configuração (já explicado anteriormente).

Para além da pasta “modules” que irá conter dentro outras pastas, cada uma com o seu módulo, também existe a pasta “libraries”, que poderá conter bibliotecas externas (no nosso caso só foi necessária uma “library” chamada “timepicker”), e também existe a pasta “themes” que irá conter o(s) tema(s), que é o que contém as regras de definição do aspeto (apresentação) do Website, caso não sejam usados os temas por defeito do Drupal. No menu de configuração de administrador do Drupal dá para configurar qual o tema que está a ser utilizado, podendo mudar para outro, e o Website passa a funcionar com o mesmo conteúdo, mas com um aspeto visual diferente.

6.2.1. Ficheiro de Dependências do Módulo

O ficheiro info (neste caso o módulo é chamado factsegur_invoices, logo este ficheiro é o factsegur_invoices.info) contém informação que vai indicar ao Drupal o nome do módulo, a descrição com que este aparece no interface do Drupal para ativação de módulos, a versão do drupal para o qual este foi desenvolvido, a categoria onde este aparece nessa interface, as dependências (outros módulos que precisam de estar ativos para este funcionar), bem a indicação do caminho para ficheiros necessários. Conforme indicado anteriormente ao ativar um módulo, outros módulos do quais este dependa são automaticamente ativados pelo Drupal. De seguida na Figura 24 podemos consultar um exemplo de um ficheiro, onde podemos ver no meio do ecrã imensas dependências definidas (os campos dependencies[]).



```
factsegur_invoices.add_invoice.inc x factsegur_invoices.info x factsegur_invoices.module x
name = Factsegur Invoices Management
description = Module for invoicing management.
core = 7.x
package = Factsegur Modules
dependencies[] = views
dependencies[] = date
dependencies[] = date_popup_timepicker
dependencies[] = field_group
dependencies[] = factsegur_internal_users
dependencies[] = factsegur_contacts
dependencies[] = factsegur_taxes
dependencies[] = factsegur_products
dependencies[] = factsegur_settings

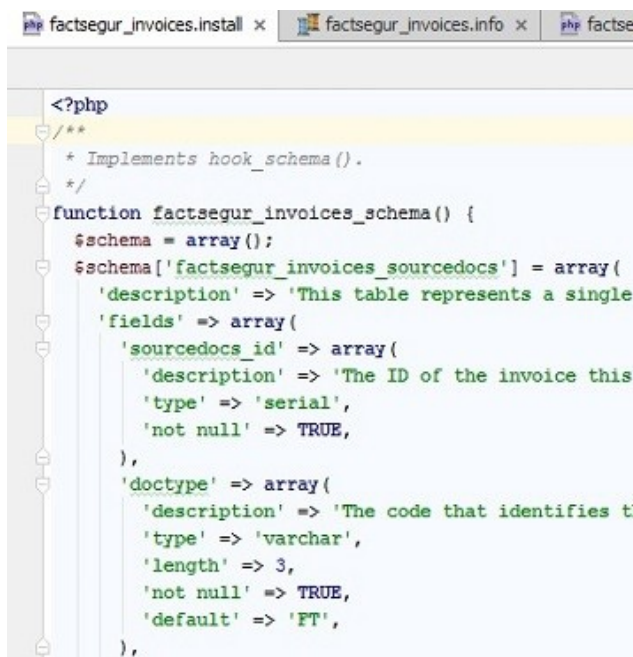
;Views Includes
files[] = factsegur_invoices.views.inc
files[] = handlers/factsegur_views_handler_field_datetime.inc
files[] = handlers/factsegur_views_handler_filter_datetime.inc
```

Figura 24: Exemplo de um ficheiro .info de um módulo

6.2.2. Ficheiro de Instalação do módulo

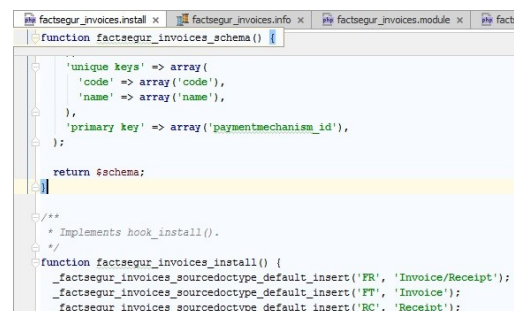
O ficheiro `.install` de um módulo vai conter um método chamado `<{nome_módulo}_schema()` que define o esquema da base de dados. Pode ser verificado na Figura 25 o início da definição de um esquema, com as primeiras duas colunas de uma tabela, a primeira contendo um serial identificando unicamente cada linha, e de seguida um campo de texto. No esquema podem ser definidos o tipo de dados de cada coluna, bem como valores por defeito, tamanho, e se podem ou não ser nulos. O esquema de um módulo pode conter a definição de várias tabelas. O esquema poderá também indicar restrições como chaves primárias, e chaves únicas (indicando que um valor nessa coluna ou lista de colunas não pode ser repetido noutras linhas).

Conforme pode ser verificado na Figura 26, no fundo da imagem pode ser encontrado o método `<{nome_módulo}_install()`, com código PHP que vai ser executado automaticamente na altura da instalação do módulo. No caso deste estágio esta função é utilizada frequentemente para preencher vários dados necessários ao funcionamento da faturação, como por exemplo preencher a lista de tipos de documento suportados pelo programa, ou as taxas de Imposto configuradas por defeito.



```
<?php
/**
 * Implements hook_schema().
 */
function factsegur_invoices_schema() {
  $schema = array();
  $schema['factsegur_invoices_sourcedocs'] = array(
    'description' => 'This table represents a single',
    'fields' => array(
      'sourcedocs_id' => array(
        'description' => 'The ID of the invoice this',
        'type' => 'serial',
        'not null' => TRUE,
      ),
      'doctype' => array(
        'description' => 'The code that identifies the',
        'type' => 'varchar',
        'length' => 3,
        'not null' => TRUE,
        'default' => 'FT',
      ),
    ),
  );
}
```

Figura 25: Exemplo 1 de um ficheiro `.install` de um módulo



```
function factsegur_invoices_schema() {
  // ...
  return $schema;
}

/**
 * Implements hook_install().
 */
function factsegur_invoices_install() {
  _factsegur_invoices_sourcedocstype_default_insert('FR', 'Invoice/Receipt');
  _factsegur_invoices_sourcedocstype_default_insert('FT', 'Invoice');
  _factsegur_invoices_sourcedocstype_default_insert('RC', 'Receipt');
}
```

Figura 26: Exemplo 2 de um ficheiro `.install` de um módulo

6.2.3. Ficheiro de definição do módulo

O ficheiro `.module` de um módulo vai conter um método chamado `<{nome_módulo}_menu()` onde são definidas as páginas que o Drupal vai aceitar, bem como as permissões para o uso dessas páginas, e qual o método ou form que vai construir a página em si.

A imagem da Figura 27 permite ver a definição de 2 páginas no drupal, a primeira envolve a criação de um documento comercial, e a segunda a emissão de um recibo a partir dos dados de outro documento.

```
/**
 * Implements hook_menu().
 */
function factsegur_invoices_menu() {
  $items = array();

  //add new invoice
  $items['home/invoices/invoice/add'] = array(
    'title' => t('Add Invoice'),
    'description' => t('Add invoice to the system'),
    'page callback' => 'drupal_get_form',
    'page arguments' => array('factsegur_invoices_add_invoice_form'),
    'access callback' => 'user_access', // user_access('access conten
    'access arguments' => array('factsegur invoices add invoice'),
    'file' => 'factsegur_invoices.add_invoice.inc',
    'type' => MENU_NORMAL_ITEM, // when it's menu normal item it's no
  );

  //add new invoice from previous document
  $items['home/invoices/invoice/add/%factsegur_invoices_invoice'] = a
    'title' => t('Add Invoice'),
    'description' => t('Add invoice to the system'),
    'page callback' => 'drupal_get_form',
    'page arguments' => array('factsegur_invoices_add_invoice_form',
    'access callback' => 'user_access', // user_access('access conten
    'access arguments' => array('factsegur invoices add invoice'),
    'file' => 'factsegur_invoices.add_invoice.inc',
    'type' => MENU_NORMAL_ITEM, // when it's menu normal item it's no
  );
};
```

Figura 27: Exemplo 1 de um ficheiro .module de um módulo

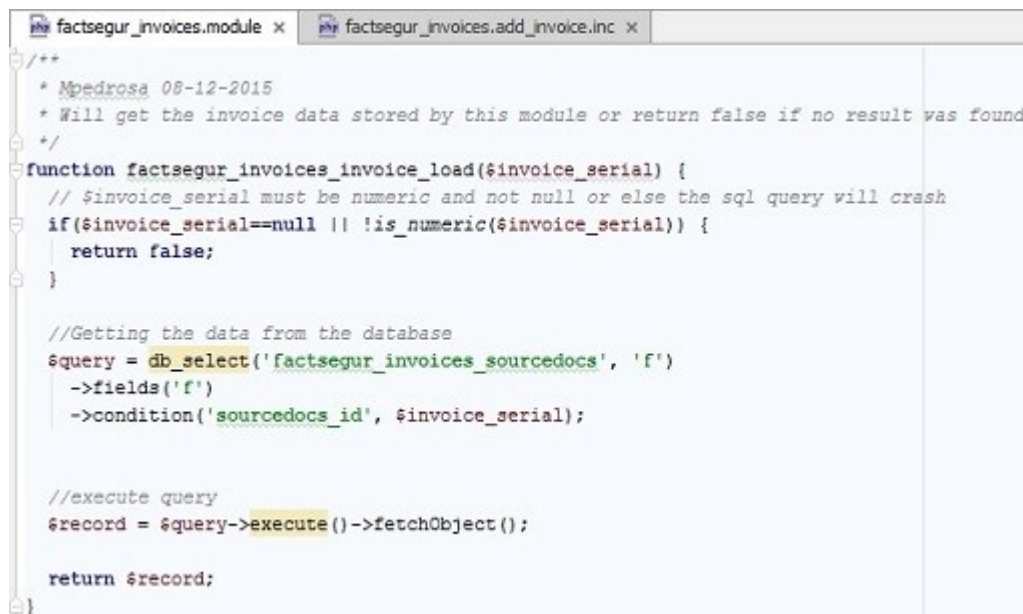
É importante referir segunda definição de página contém um elemento dinâmico (wildcard) que indica um elemento existente no Drupal. Por exemplo se chamarmos a página “/home/invoices/invoice/add/1”, sendo que o 1 era o wildcard “%factsegur_invoices_invoice” iremos obter a página resultante apenas se existir o elemento, nesse caso se existir uma fatura com o código de linha (serial) 1. Caso não exista o próprio Drupal mostra uma mensagem a dizer que a página não foi encontrada, em vez de criar a página segundo o “form” da página.

Sendo assim é o método <{nome_modulo}_menu()> de cada módulo que vai indicar quais são as páginas disponíveis no Drupal, e o método com a definição do conteúdos destas, e não ficheiros html ou ficheiros php com o código para criação destas.

O ficheiro .module vai conter também o método {nome_modulo}_permission(), que vai servir para o Drupal saber quais são as permissões que o módulo utiliza, que normalmente serão as identificações usadas em cada elemento do _menu() no campo “access_arguments”, conforme pode ser verificado na Imagem de Exemplo 1. Este método _permission contém a identificação da permissão, e qual a sua descrição, que aparece no menu de configuração de permissões (em vez da identificação, ou o “nome de máquina”).

O ficheiro .module vai conter outros métodos importantes para o funcionamento das

páginas/forms definidas pelo módulo, podendo até ser chamados por outros módulos, desde que este módulo esteja ativo. Um conjunto de métodos importantes são os métodos que fazem a ligação com a base de dados, contendo o código para procurar, ler, inserir, atualizar e apagar elementos da base de dados gerida por esse módulo. Esses métodos costumam ter a identificação `<{nome_módulo}_{nome_elemento}_load()` caso seja o método para carregar um ou vários elementos para objecto, existindo também o `_save()` para atualização e gravação, e o `_delete()` para apagamento. O acesso à base de dados nestes não é através de código SQL, mas através de mecanismos próprios do Drupal, que vai permitir a abstração do motor de base de dados, fazendo com que o mesmo código funcione em variados motores de base de dados. Na Figura 28 podemos ver como efetuar no Drupal uma pesquisa a uma tabela (select).



```
factsegur_invoices.module x factsegur_invoices.add_invoice.inc x
/**
 * Mpedrosa 08-12-2015
 * Will get the invoice data stored by this module or return false if no result was found
 */
function factsegur_invoices_invoice_load($invoice_serial) {
  // $invoice_serial must be numeric and not null or else the sql query will crash
  if($invoice_serial==null || !is_numeric($invoice_serial)) {
    return false;
  }

  //Getting the data from the database
  $query = db_select('factsegur_invoices_sourcedocs', 'f')
    ->fields('f')
    ->condition('sourcedocs_id', $invoice_serial);

  //execute query
  $record = $query->execute()->fetchObject();

  return $record;
}
```

Figura 28: Exemplo de um método `_load()` no `.module`

6.2.4. Form API

No nosso trabalho utilizamos frequentemente o “Form API” do Drupal [69]. Este permite a definição do conteúdo de uma página sem necessitar de criar código HTML diretamente. Usando o “Form API” os campos são definidos como objetos, e a conversão deste objetos para código HTML é tratada internamente pelo Drupal, e definido o aspeto através da camada de tema. Isto significa que não temos de abrir e fechar tags html, inserção de css inline ou classes CSS. Fica bastante mais fácil definir o conteúdo da página do que com o uso de templates nos quais se insere conteúdo dinâmico com código PHP.

Na Figura 29 podemos consultar a definição de um campo de texto (Textfield), e uma lista dos seus parâmetros, incluindo uma definição de auto preenchimento (o parâmetro #autocomplete_path na 10ª linha).

```
// name of the client, should import other data when filled
$form['tab_general_data']['client']['name'] = array(
  '#type' => 'textfield',
  '#title' => t('Name'),
  '#default_value' => isset($form_state['values']['name']) ? $form_state['values']['name'] : NULL,
  '#size' => 60,
  '#maxlength' => 100,
  '#required' => TRUE,
  '#disabled' => $form_state['storage']['settings']['is_view_document'] ? TRUE : FALSE,
  '#autocomplete_path' => 'factsegur-contacts/autocomplete-enterprise', // Dont forget you need th
  '#ajax' => array(
    'callback' => 'factsegur_invoices_add_invoice_ajax_changed_client_name_callback',
    'wrapper' => 'client_fieldset',
    'event' => 'autocompleteSelect', //calls the ajax when the enter is pressed, but loses the inf
    // 'keypress' => TRUE, //calls the ajax when the enter is pressed, but loses the information se
  ),
);
```

Figura 29: Exemplo da definição de elementos num form

Na Imagem de Exemplo da definição de elementos num form podemos ver a definição de um campo que irá aparecer no form. Podemos verificar pela linha `<$form['tab_general_data']['client']['name']>` que este campo aparecerá dentro do fieldset 'tab_general_data', e ainda dentro do fieldset 'client' (que por sua vez pertence ao primeiro fieldset). O campo será identificado pelo nome 'name'. Podemos ver por baixo as suas propriedades, o tipo (#type), a label (#title, texto que aparecerá a identificar o campo na página), o valor por defeito(#default_value), o tamanho máximo (#size), se é obrigatório (#required, a validação obrigará a ter um valor preenchido), se o seu valor pode ser alterado (#disabled, campo aparece a cinza caso não possa). Existem ainda outras propriedades, sendo que o campo do tipo de campo é necessário, mas qualquer outra pode ser omitida, sendo definida por defeito de acordo com o Drupal.

Qualquer propriedade pode ser alterada mais tarde ao definir outros objetos, por exemplo para fazer com que o campo tenha comportamento diferente de acordo com o conteúdo de outro campo qualquer (coisa que acontece frequentemente no módulo de faturação, por exemplo não precisamos de definir razão de isenção de IVA caso o produto não seja isento). Depois da página já ter sido construída a partir do form, e enviada ao utilizador estas alterações poderão ser efetuadas através de AJAX, também integrado com o “Form API”. No exemplo em cima conseguimos ver nas últimas 7 linhas a definição de uma chamada AJAX.

O “Form API” também contribui para a segurança, de modo a que cada vez que um form é gerado é-lhe atribuído um identificador e um token, e guardado o seu estado na base de dados. Quando o cliente responde ao form (com um submit), o Drupal pode saber se o pedido veio de um form existente ou não, podendo manter relação entre o form original e as suas respostas. Isto vai fazer com que o cliente não possa programaticamente inserir o submit de um form sem ter passado pela geração do form original (workflow normal da página). Esta propriedade permite também guardar variáveis internas de estado nos dados do form (numa variável do form a que o Drupal chama \$form_state), que não passam para a página resultante, e que estão acessíveis para processamento só para esse form, e

quando a resposta for recebida num submit subsequente. Isto permite manter variáveis importantes para processamento interno escondidas do utilizador (não podem ser vista no código fonte da página)

6.2.5. Validação de Forms

Quando o utilizador preenche um form e realiza uma ação (normalmente carregar num botão), o Drupal recebe o pedido de submissão, reconhece o form (segundo a identificação em cima explicada), e em vez de apresentar o form original, ele vai tentar validar os campos. Esta validação é feita através do definido originalmente em cada objecto, através das propriedades de tamanho (#size), e da obrigatoriedade deste (#required), mas também através do definido no método `<{nome_form}_validate()>`. Caso este método exista, o Drupal “Form API” executa este método quando recebe submits para efetuar validações mais complexas que as que são definidas no objeto. É possível criar as verificações necessárias, apontando os erros e o objeto onde o erro ocorreu. No fim caso tenham sido encontrados erros o Drupal volta a enviar ao utilizador o form como ele estava, indicando as mensagens de erro, e assinalando a vermelho os campos onde os erros ocorreram. Caso não tenha ocorrido erros o Drupal procede com a submissão (submit) executando o método próprio para este efeito.

6.2.6. Submissão de Forms

Caso o submit de um form tenha passado na validação (não terem sido encontrados erros nenhuns), o Drupal “Form API” vai então executar o método `<{nome_form}_submit()>`. Conforme indicado em cima, caso tenham sido encontrados erros, o Drupal cancela o submit (não executa o método), e vai mostrar os erros encontrados ao utilizador, enviando o form de volta para ser corrigido. Este método de submissão é que vai conter o código para executar as alterações referidas nesse form, nomeadamente inserções ou alterações de dados. Este método vai tentar efetuar as alterações usando os métodos de apoio definidos nos ficheiros .module, e vai no fim indicar ao utilizador se o pedido foi efetuado com sucesso ou não.

6.2.7. Módulo Views

Este é um módulo existente na comunidade Drupal que se tornou importante para este projeto. Ele permite automatizar a criação de listagens de dados. Em vez de serem criados métodos para obter e mostrar dados, podemos criar no Drupal uma vista. As vistas são configuradas sem precisar de escrever código nenhum, só selecionando as colunas que queremos mostrar graficamente, podemos selecionar quais os elementos que queremos nas colunas (Fields), e quais os filtros que queremos para os dados (Filter Criteria, para colocar condições), e podemos mesmo, automaticamente, colocar esses filtros na vista, de modo ao utilizador poder alterar as condições e ver os dados que este precisa. Isto torna fácil certas operações, como por exemplo filtrar a lista de clientes por nome de cliente, não tendo que alterar o método que obtém os dados para contar com essa condição. É tudo tratado pela vista. Também podem ser configurados facilmente mecanismos de paginação, para apresentar só um certo número de resultados por cada página da vista.

Na Figura 30 podemos ver o exemplo da configuração onde no lado esquerdo ao meio podemos encontrar os elementos definidos para os FIELDS. Cada um destes elementos é uma coluna que aparece na vista, e caso quiséssemos retirar a coluna era só carregar no elemento e depois no botão de remover. Caso quiséssemos adicionar uma nova coluna era carregar no botão de adicionar fields (no meio da figura) e seleccionar o campo desejado a partir de uma lista de campos da tabela cuja vista está a consultar.

The screenshot shows the Drupal Views configuration page for 'Factsegur Products - List Products'. The interface is divided into two main panels: the left panel for field and criteria configuration, and the right panel for block settings.

Left Panel Configuration:

- TITLE:** Factsegur Products - List Products
- FORMAT:** Table | Settings
- FIELDS:** A list of available fields including 'Factsegur Products: Serial (Serial)', 'Factsegur Products: Description (Description)', 'Factsegur Product Type: Product Type Description (Type)', 'Factsegur Products: Unit Of Measurement (Measurement)', 'Factsegur Taxes: Tax Table Percentage (Tax)', 'Factsegur Products: Default Price Without Tax (Price Without Tax)', 'Global: PHP (Price Without Tax)', 'Factsegur Products: Cost Price Without Tax (Cost Price Without Tax)', 'Global: PHP (Cost Price Without Tax)', 'Global: PHP', and 'Global: Custom text'. An 'Add' button is at the bottom.
- FILTER CRITERIA:** A list of available filter criteria including 'Factsegur Product Type: Product Type Description (exposed)', 'Factsegur Product Type: Product Type Description (exposed)', and 'Factsegur Products: Default Price Without Tax (exposed)'. An 'Add' button is at the bottom.
- SORT CRITERIA:** A list of available sort criteria including 'Factsegur Products: Description (asc)'. An 'Add' button is at the bottom.

Right Panel (BLOCK SETTINGS):

- Block name:** Factsegur Products - Lis...
- Access:** Permission | View the list of Products of the system.
- HEADER:** An 'Add' button is present.
- FOOTER:** An 'Add' button is present.
- PAGER:** Use pager: Full | Paged, 10 items. More link: No.

Figura 30: Exemplo da configuração de uma vista do Drupal

Para usar uma vista temos que criar um ficheiro {nome_modulo}.views.inc que define quais os elementos que temos na nossa base de dados, qual o tipo destes (texto, número, etc.), e o handler para manipulação destes. O handler é que define as regras de comparação para os filtros, e de tratamento dos dados, por exemplo contém as regras de tratamento de datas de modo a poder converter um texto com um timestamp para um formato reconhecido pela base de dados. Já existem vários handlers pré-definidos, mas podem ser criados novos por nós, caso necessitemos de algo mais elaborado.

Tal como o resto, as vistas são instaladas no Drupal a partir de código quando o módulo é instalado, não tendo de ser criada ou configurada as vistas definidas nos módulos em cada instalação de Drupal de modo a esta funcionar nos websites.

6.3. Módulo de Utilizadores Externos

O utilizador externo representa um utilizador que pode entrar dentro do sistema (tem nome de utilizador e palavra passe), mas que tem o acesso (permissões) bloqueado à maioria deste. Este tem acesso aos dados da sua própria empresa, o que significa que é necessário poder indicar qual é essa empresa, mas só um utilizador autorizado é que o pode fazer. Sendo assim, aos dados básicos do utilizador de Drupal teria de ser possível seleccionar uma de entre uma lista de empresas (Módulo de Contactos, contactos de

empresa). Teria de ser garantido que o próprio utilizador não poderia alterar esta informação, de modo a não poder editar os seus dados de modo a indicar que pertencia a outra empresa, pois passaria a poder ver dados relativos a esta, o que seria uma falha de segurança.

O Drupal já contém mecanismos de gestão de utilizadores, e houve uma preocupação de manter o funcionamento do Drupal sempre que possível, de modo a poder aproveitar ao máximo as funcionalidades já existentes. Sendo assim, o módulo de Utilizadores Externos representa uma adição a um utilizador do Drupal.

Este requisito foi superado através do mecanismo de permissões, criando acessos separados para editar os dados de utilizador, e para alterar a identificação da empresa. Só um Utilizador Interno com a permissão certa poderia alterar a informação de qual empresa o utilizador externo faz parte. Este utilizador pode alterar o resto dos seus dados, mas não é possível alteração da identificação da empresa (nem é possível ver a lista de empresas para seleccionar outra, o combobox fica desativado).

Outra tarefa foi manter o funcionamento dos utilizadores do Drupal, e a estes adicionar a nova informação. Em termos de base de dados isto foi conseguindo adicionando uma tabela nova na base de dados, independente da tabela onde são guardados os dados dos utilizadores do drupal. Esta nova tabela irá fazer a relação entre o id interno (Serial) que o Drupal atribui ao utilizador no módulo de utilizadores, e um outro id (serial) que liga esse utilizador a um contacto de empresa (Enterprise Contact) existente, relativo ao módulo de contactos. Foi também estudada a maneira de manter a coerência entre os elementos, para não gravar o utilizador do Drupal normalmente, mas ocorrer um erro na gravação da parte específica deste módulo, logo seria um utilizador válido, mas os dados relativos a ser um utilizador externo não estariam definidos.

Ainda relativamente a manter o funcionamento dos utilizadores Drupal, este contém um mecanismo que pode alterar forms já existentes noutros módulos ou mesmo no core do Drupal. O mecanismo de alteração de forms consiste em implementar no novo módulo um método com o nome do form, seguido de `_alter`. Sendo assim, o Drupal contém um form chamado `user_profile_form`, e foi implementado o `user_profile_form_alter()`. Caso esteja ativo o módulo, este método recebe o form básico, proveniente do Drupal, e são efetuadas alterações de modo a alterar o seu conteúdo e/ou adicionar novos elementos. Os elementos do form original estariam presentes (foram mantidos todos os campos que o drupal já tinha na gestão de utilizadores), mas foram adicionados a estes os campos novos necessários. No caso dos utilizadores externos apenas foi necessário adicionar um novo campo, que permite seleccionar qual é a empresa à qual o utilizador pertence. Esta empresa é um contacto, relacionado com o módulo de contactos. Isto implica ter um campo 'Select'(combobox em java e html) onde aparece a listagem de todos os contactos de empresa, e pode ser seleccionado um deles. Foi também acrescentado um novo método de submissão (para além do método de validação já existente no `user_profile_form`) e um novo método de submissão.

A Figura 31 seguinte representa o fluxo que controla a inserção de um novo utilizador externo, incluindo a integração com o módulo de utilizadores do Drupal através do `drupa_get_form()`, na parte esquerda da imagem, e no canto inferior direito pode ser visto o fluxo de submissão, incluindo o mecanismo que garante a coerência entre os utilizadores externos e um utilizador Drupal que lhe corresponda.

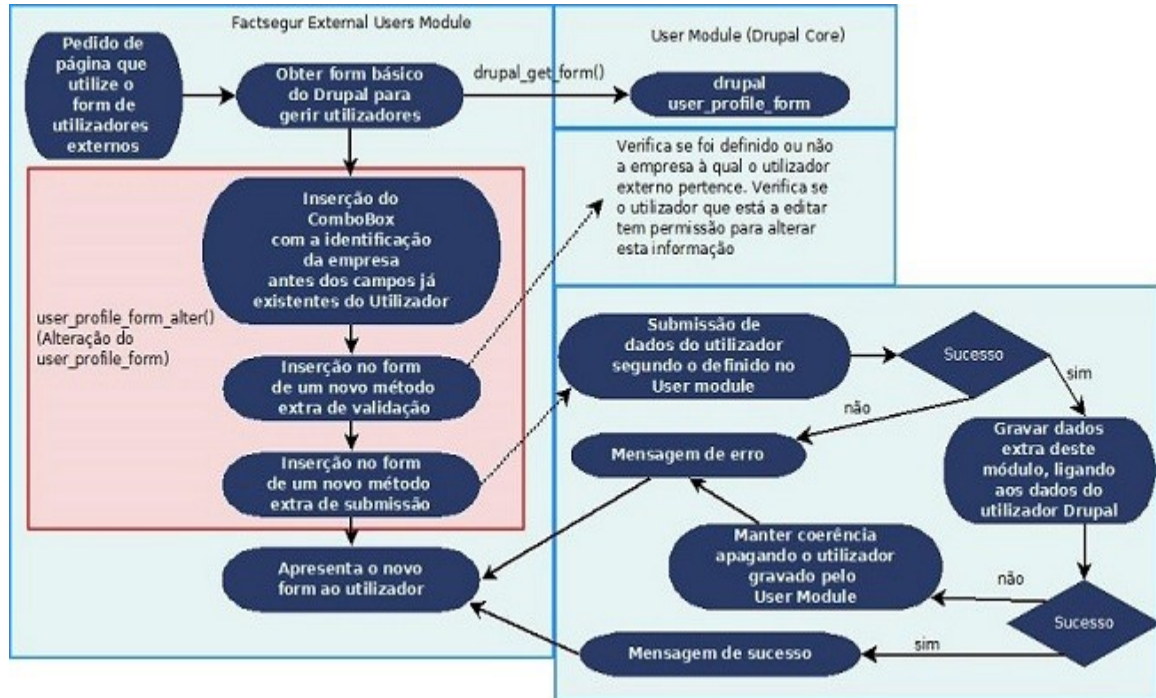


Figura 31: Diagrama de Fluxo do form de Utilizadores Externos

Foi também criada uma vista (através do módulo de vistas já explicado) que mostra a listagem de utilizadores externos, permite a filtragem por certos campos, e que contém o atalho para a adição de novos utilizadores externos, bem como podendo visualizar, editar e apagar os utilizadores externos apresentados.

Podemos ver de seguida na tabela 30 a listagem de todas as páginas definidas no módulo de utilizadores externos, e a descrição do que cada página contém, e quais as funcionalidades que esta oferece.

Páginas definidas no Módulo de Utilizadores Externos	
home/definitions/externalusers/profile	Página contendo o form com os dados do Utilizador Externo. Permite a edição do utilizador externo atualmente autenticado.
home/definitions/externalusers/profile/edit/%external_user	Página contendo o form de dados de qualquer Utilizador Externo, permite a edição dos dados relativos a um certo identificador (serial id no fim do endereço).
home/definitions/externalusers/profile/delete/%external_user	Página com o interface para a remoção de um Utilizador Externo, respetivo a um certo identificador (serial id no fim do endereço). É pedida uma confirmação.
home/definitions/externalusers/add	Página contendo o form de dados de qualquer Utilizador Externo, permite a adição de um novo utilizador no Drupal, contendo também os dados novos do módulo de Utilizadores Externos.
home/definitions/externalusers/profile/view/%external_user	Página de consulta dos dados de um Utilizador Externo. Este interface apenas mostra os dados, não permite a sua edição.

home/definitions/externalusers/ list/external	Página que mostra uma listagem de Utilizadores Externos, pode ser filtrada por certos dados, tais como nome de utilizador, e nome de empresa associada. Construída através do Módulo Views.
--	---

Tabela 30: Páginas definidas no módulo de Utilizadores Externos

6.4. Módulo de Contactos

O módulo de contactos tem a particularidade de conter 2 tipos bastante diferentes, os Contactos Individuais, e os Contactos de Empresa. Estes contêm dados separados, e os contactos individuais podem ou não ser identificados como pertencentes a uma empresa. Sendo assim, foram criados os tipos de contactos completamente separados, com tabelas diferentes, e interfaces diferentes para consulta e para edição. Sendo assim, a edição de contactos individuais é feita no seu “form” próprio, e a edição de contactos de empresa num outro “form” semelhante, mas diferente, com outros dados definidos. Também a consulta das suas listagens envolve 2 vistas separadas, uma para os contactos individuais, outra para os contactos de empresa.

Quando o utilizador preenche os dados do Contacto que está a ser adicionado (ou editado) no form e carrega no botão de gravar as alterações, é então feita a validação dos conteúdos (formatos de dados, e regras próprias que são necessário cumprir, como a verificação sobre se esse contacto de empresa tem documentos emitidos, faturas ou outros), e caso não seja encontrado nenhum problema, fazendo as alterações respetivas na base de dados e avisando o utilizador do sucesso ou erro da operação.

Houve uma preocupação extra com os contactos de empresa, visto estes serem necessários para o módulo de faturação, e para este ser necessário estar definido o número de contribuinte (que passou a ser obrigatório, não pode ser adicionado ou alterado um contacto de empresa sem estar definido este dado). Também não é permitido apagar o contacto ou alterar os dados principais (especialmente o nome e número de contribuinte) depois de existirem documentos emitidos (faturas e outros tipos) nos quais o cliente seja esse Contacto de Empresa.

Também foi implementado nos contactos de empresa a utilização deste em campos de Preenchimento Automático (por nome do cliente ou por número de contribuinte), através do mecanismo do Drupal para este efeito. Este preenchimento automático será depois utilizado no módulo de faturação para preencher os dados do cliente ao escrever o seu nome ou o seu número de contribuinte nos campos respetivos para este efeito. Ao utilizar o preenchimento automático não só o campo no qual foi procurado é preenchido, mas todos os campos relativos a esse Contacto são importados a partir do módulo de contactos de empresa, preenchendo automaticamente os campos respetivos na página de emissão de documento através de AJAX.

A Figura 32 representa o fluxo que controla a gestão dos contactos, tanto para inserção como para edição, envolvendo identificar se é um contacto individual ou de empresa, de seguida verificar se a informação preenchida é válida, e por fim tentar gravar esses dados.

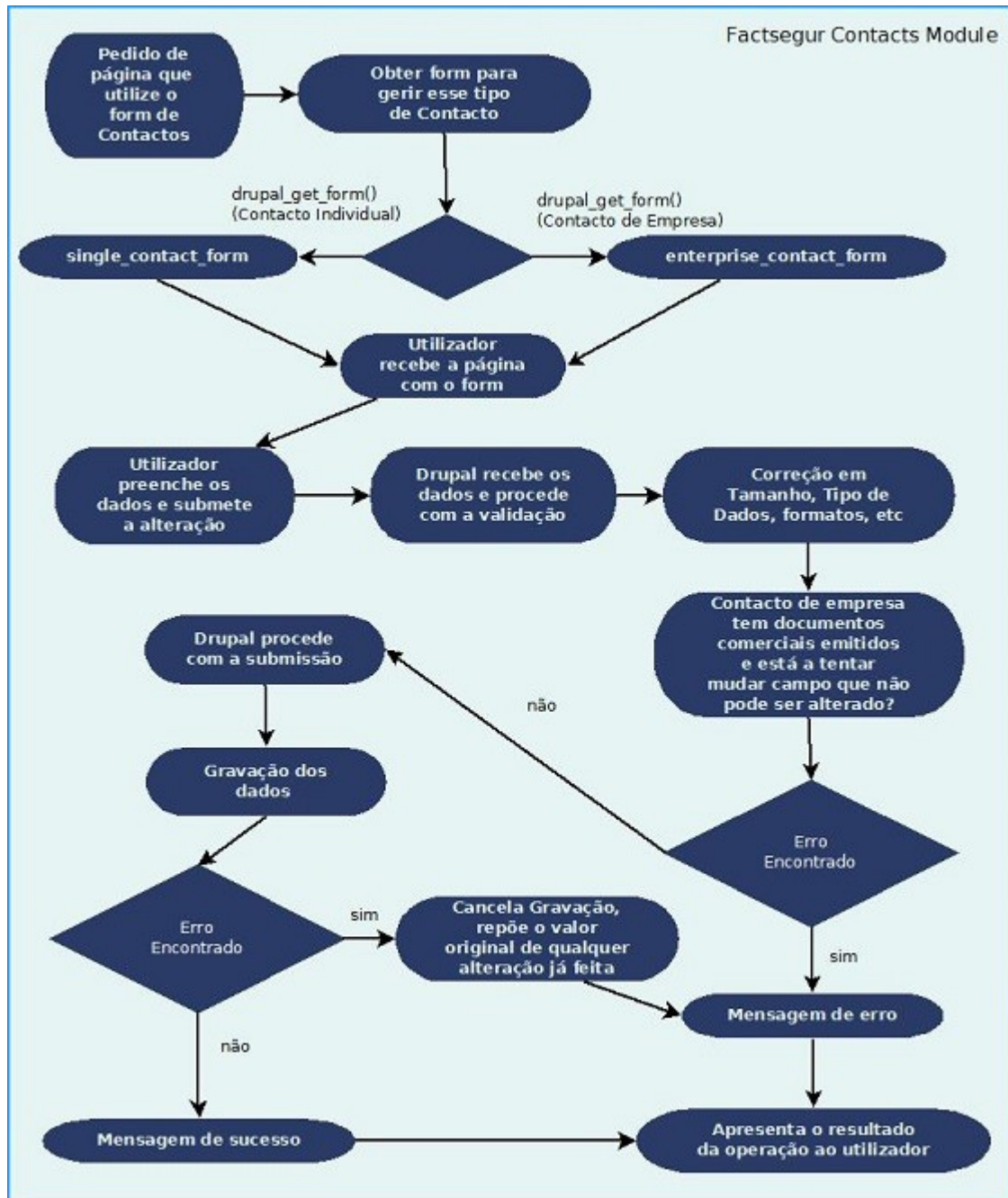


Figura 32: Diagrama de Fluxo do form de Contactos

Os contactos de empresa são utilizados como clientes nos documentos do módulo de Faturação (faturas, etc.), sendo possível adicionar no documento de uma fatura um novo contacto de empresa, sem ter de o criar nas páginas do módulo de contactos antes de poder emitir esse documento. Nesse caso o próprio módulo de faturação vai submeter um pedido de criação de nova ficha de cliente (sem necessidade de preencher o form de gestão de contactos de empresa), sendo criada automaticamente uma ficha de cliente para o novo cliente, a partir os dados definidos no documento de faturação. Este é um caso

especial de contacto, porque precisa de existir de modo a aparecer no ficheiro SAF-T(PT), mas poderia interferir com a lista de clientes gerida pelo utilizador, aparecendo uma lista de contactos de empresa demasiado extensa, em que alguns deles só seriam potencialmente usados em um documento específico. Sendo assim optou-se por esconder por defeito estas fichas de cliente geradas automaticamente da lista de contactos genérica, que é mostrada quando entramos na página inicial da categoria de contactos, carregando no botão do menu principal no lado esquerdo (página home/contacts/single/list).

Podemos ver de aqui na tabela 31 a listagem de todas as páginas definidas no módulo de contactos, e a descrição do que cada página contém, e quais as funcionalidades que esta oferece.

Páginas definidas no Módulo de Contactos	
home/contacts/single/add	Página contendo o form de dados para gestão de um Contacto Individual, permite a adição de um novo contacto.
home/contacts/enterprise/add	Página contendo o form de dados para gestão de um Contacto de Empresa, permite a adição de um novo contacto.
home/contacts/single/edit/ %singlecontact	Página contendo o form com os dados para gestão de qualquer Contacto Individual, permite a edição dos dados relativos a um certo identificador (serial id no fim do endereço).
home/contacts/enterprise/edit/ %enterprisecontact	Página contendo o form com os dados para gestão de qualquer Contacto de Empresa, permite a edição dos dados relativos a um certo identificador (serial id no fim do endereço).
home/contacts/single/delete/ %singlecontact	Página com o interface para a remoção de um Contacto Individual, respetivo a um certo identificador (serial id no fim do endereço). É pedida uma confirmação.
home/contacts/enterprise/delete/ %enterprisecontact	Página com o interface para a remoção de um Contacto de Empresa, respetivo a um certo identificador (serial id no fim do endereço). É pedida uma confirmação.
home/contacts/single/list	Página que mostra uma listagem de Contactos Individuais, pode ser filtrada por certos dados, tais como nome do contacto, e nome de empresa associada, entre outros. Criada através do Módulo Views.
home/contacts/enterprise/list	Página que mostra uma listagem de Contactos de Empresa, pode ser filtrada por certos dados, tais como nome de empresa, e endereço de email, entre outros. Criada através do Módulo Views.
factsegur-contacts/ autocomplete-enterprise	Endereço não usado diretamente, mas internamente para obter os dados relativos ao Preenchimento Automático por nome de empresa.
factsegur-contacts/ autocomplete-enterprise-taxcode	Endereço não usado diretamente, mas internamente para obter os dados relativos ao Preenchimento Automático por número de contribuinte.

Tabela 31: Páginas definidas no módulo de Contactos

6.5. Módulo de Permissões

O Drupal já contém mecanismos de permissões para os seus utilizadores. Este usa categorias de utilizadores (chamados roles), e em cada um dos roles pode ser configurada cada permissão individual, definida nos módulos. Ao editar os utilizadores pode ser atribuído a cada utilizador um ou vários roles. Foi criado um novo interface de edição de

roles e permissões que não mostre as permissões do Core do Drupal, mas apenas as permissões que necessitamos nos nossos módulos. Ao submeter as alterações no nosso interface, são usados os mecanismos (métodos) próprios do Drupal para adicionar, editar e apagar os roles e permissões, em vez de os alterarmos manualmente numa base de dados nossa, ou acedendo às bases de dados internas do Drupal. Isto significa que se usarmos as interfaces genéricas do Drupal para ver e editar os roles e permissões, os nossos aparecem lá corretamente (bem como as permissões internas do Drupal que não aparecem no nosso interface). A integração é total, e não foi convertido o funcionamento do Drupal, este permanece inalterado, apenas o interface de consulta e gestão foi modificado.

A maior preocupação deste módulo era ser independente dos outros módulos, e não precisar de saber quais outros módulos existem, e quais as suas permissões, de modo a que a adição de um novo módulo ou de uma nova permissão não envolvesse atualizar o módulo de permissões com informação sobre estes.

A maneira como isto foi conseguido foi o form de permissão não conter ele próprio a lista de módulos e permissões, mas implementar um método que inserisse dinamicamente categorias de permissões (nome de um módulo), e permissões novas dentro dessa categoria(em checkboxes). Outros módulos iriam implementar o `form_alter` sobre o form de permissões (conforme explicado nos utilizadores externos), e iriam inserir as suas próprias permissões dentro do form de permissões através desse método. Devido ao uso do método preparado para o efeito, este módulo apenas precisaria de chamar esse método dentro do `form_alter`, não sendo necessário ao módulo novo conhecer o funcionamento interno do módulo de permissões.

Na figura 33 pode ser encontrado o fluxo que indica como o módulo de permissões comunica com os outros módulos, esperando que estes publiquem as suas próprias listas de permissões dinâmica mente, conforme pode ser visto nos módulos vistos em cima, que vão publicar os seus dados de junto do módulo de permissões, que está preparado para receber destes essa informação.

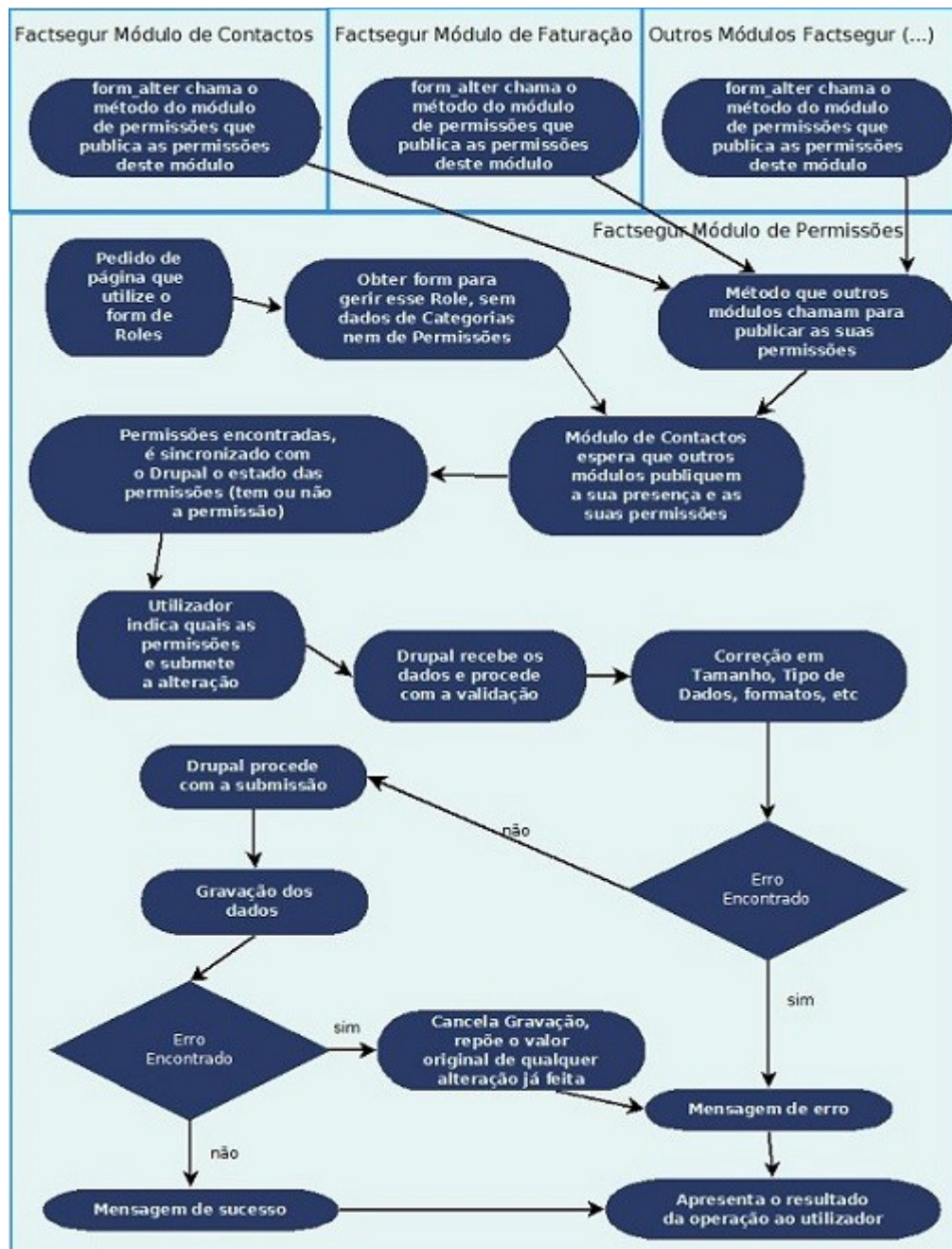


Figura 33: Diagrama de Fluxo do form de Permissões

Conforme já foi explicado em cima, o módulo de permissões de cada vez que apresenta a página de edição de permissões esta vai ser gerada vazia, e fica à espera que os outros módulos publiquem a sua informação de permissões, de modo a poder preencher a página.

De seguida na tabela 32 pode ser consultada a lista de todas as páginas definidas pelo módulo de permissões, e o que consiste cada página.

Páginas definidas no Módulo de Permissões	
home/definitions/permissions/addrole	Página contendo o form de dados para a adição de um novo papel de utilizador (role), permite a adição de um novo Role.
home/definitions/permissions/edit/%role	Página contendo o form com os dados de qualquer Role, permite a edição dos dados relativos a um certo identificador (serial id no fim do endereço).
home/definitions/permissions/delete/%role	Página com o interface para a remoção de um Role, respetivo a um certo identificador (serial id no fim do endereço). É pedida uma confirmação.
home/definitions/permissions/list	Página que mostra uma listagem de Roles, pode ser filtrada por certos dados, tais como nome do contacto, e nome de empresa associada, entre outros. Criada através do Módulo Views.

Tabela 32: Páginas definidas no módulo de Permissões

6.6. Módulo de Faturação

O maior desafio na implementação deste módulo consistiu no cumprimento dos requisitos de certificação para software de faturação. Isto implica que este necessita da capacidade de exportação dos seus dados para o formato SAF-T(PT), e para isto, o seu modelo de dados precisaria de suportar todos os dados e informações que seriam precisas para esta exportação. Outra necessidade seria a criação de uma assinatura digital (de acordo com o especificado nos requisitos de certificação de software) na altura da gravação de cada documento através da sua data, valores, e assinatura da linha anterior. Caso fossem eliminados ou alterados dados de documentos diretamente na base de dados, a assinatura dessa linha e linhas posteriores deixariam de estar válidas, porque não iria existir correspondência entre os dados existentes e as assinaturas.

Também seria necessário bloquear variados tipos de operações, como por exemplo a criação de recibos sobre elementos que não constassem em faturas, ou verificar a obrigatoriedade de associar o documento que está a ser retificado(Fatura) em Notas de Crédito e Notas de Débito, bem como o documento sobre qual está a ser emitido um recibo nesse caso.

Na Figura 34 podem ser consultados os fluxos da página de inserção de documentos relativos às alterações aos dados gerais, incluindo os auto preenchimentos de campos por AJAX, e a atualização das datas de pagamento de acordo com a seleção de data de documento.

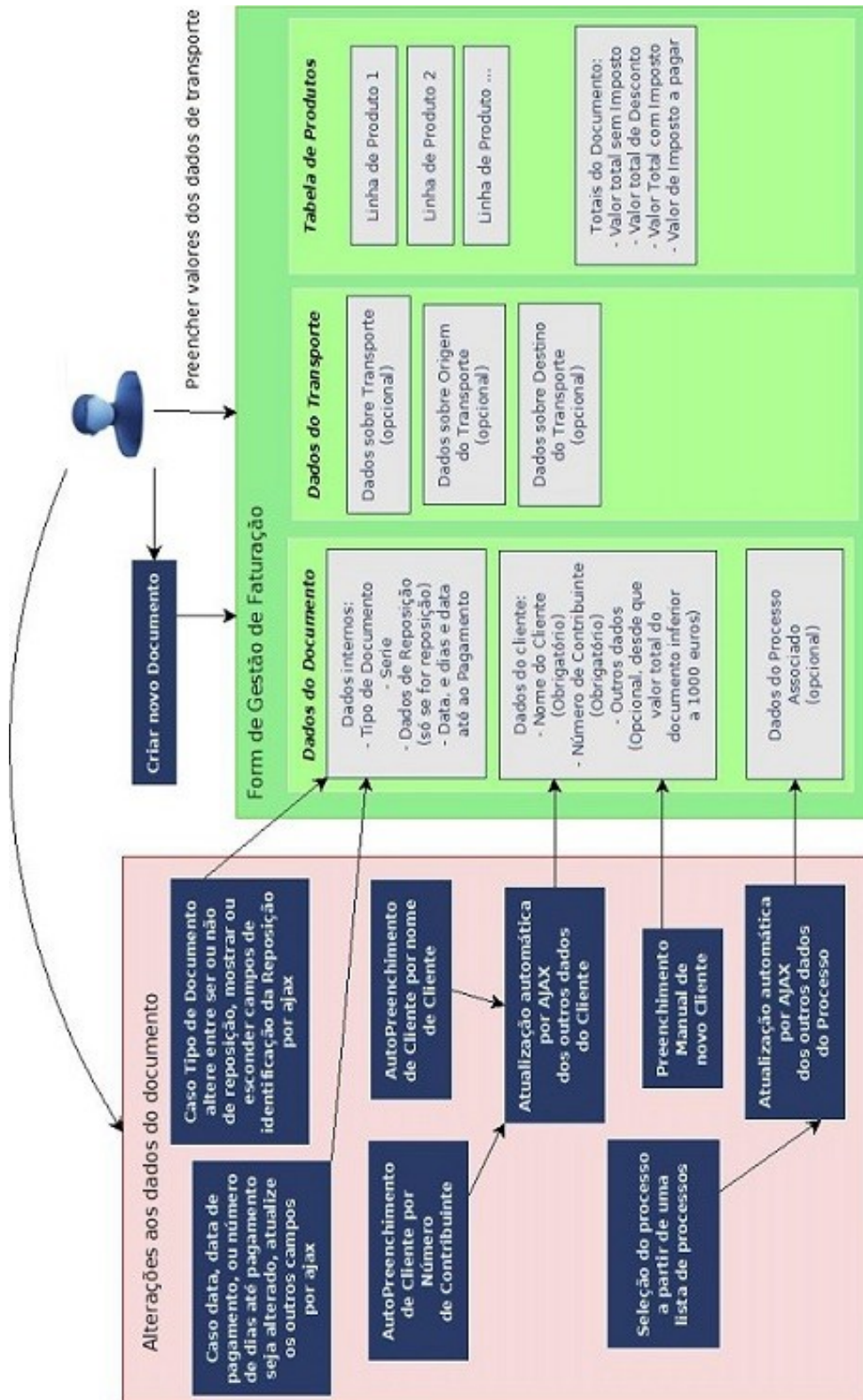


Figura 34: Form de Gestão de Faturação e alterações aos dados gerais

Na figura 35 podemos consultar novamente um fluxo relativo à inserção de documentos, mas relativo à alteração das linhas de produto nas tabelas. Estas têm de permitir uma forte dinâmica de modo a ser fácil o seu preenchimento, e seja possível apagar ou alterar dados inseridos caso existam erros. Também é necessário garantir que os dados inseridos são válidos, pois a linha contém bastantes restrições que é necessário cumprir, como o não poder conter produtos em recibos que não existam na sua fatura, ou não referenciar documentos que não existam, entre muitas outras restrições.

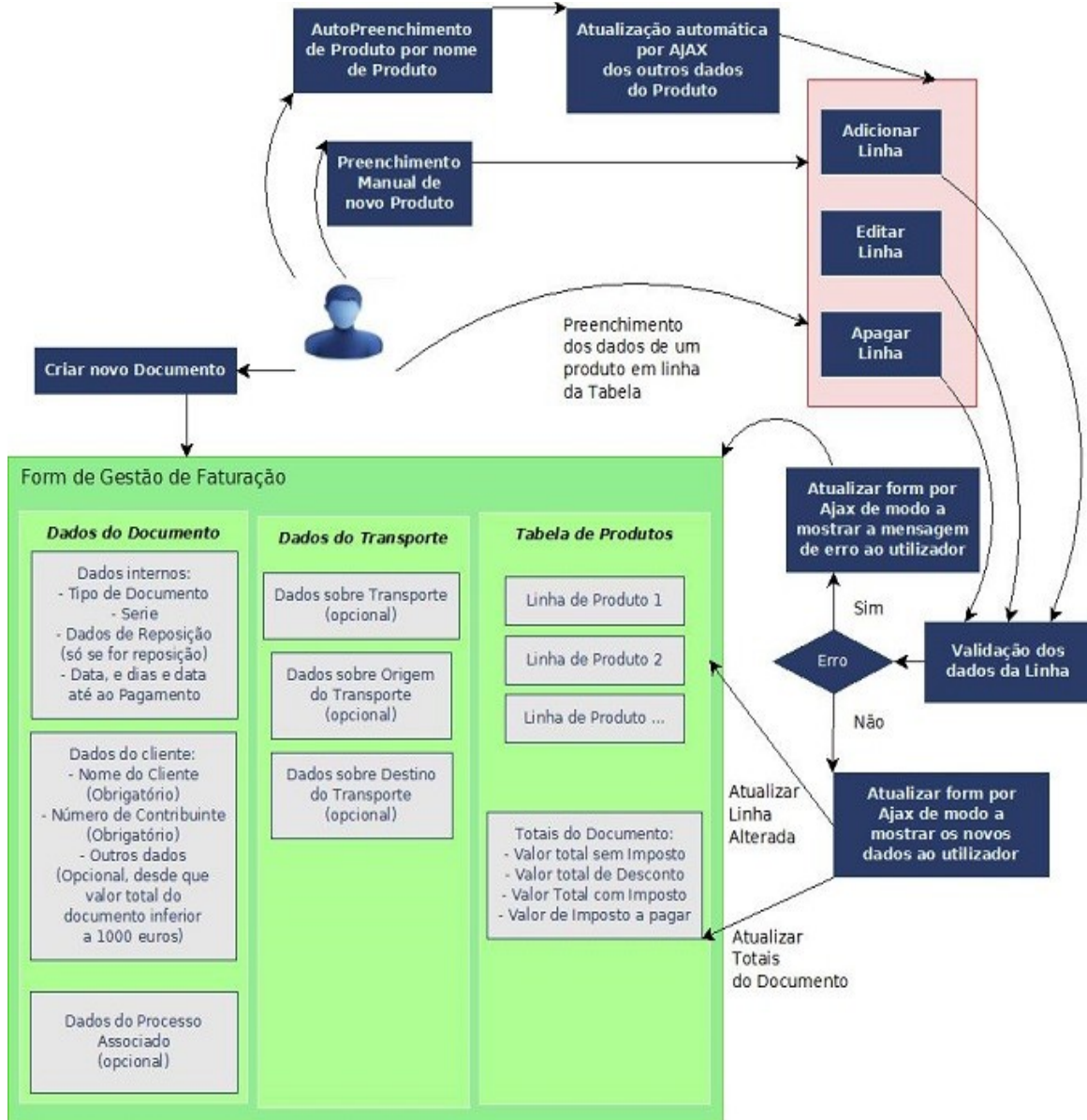


Figura 35: Form de Gestão de Faturação e alterações à tabela de produtos

Finalmente a Figura 36, também sobre o fluxo da criação de novos documentos comerciais trata da última parte, depois do documento estar preenchido, os passos seguidos para a sua validação como um todo, a sua gravação, e a emissão da assinatura digital necessária para a certificação de software. É também necessário a garantia de que não é gravada apenas metade da informação, e caso ocorra um erro toda a operação seja cancelada.

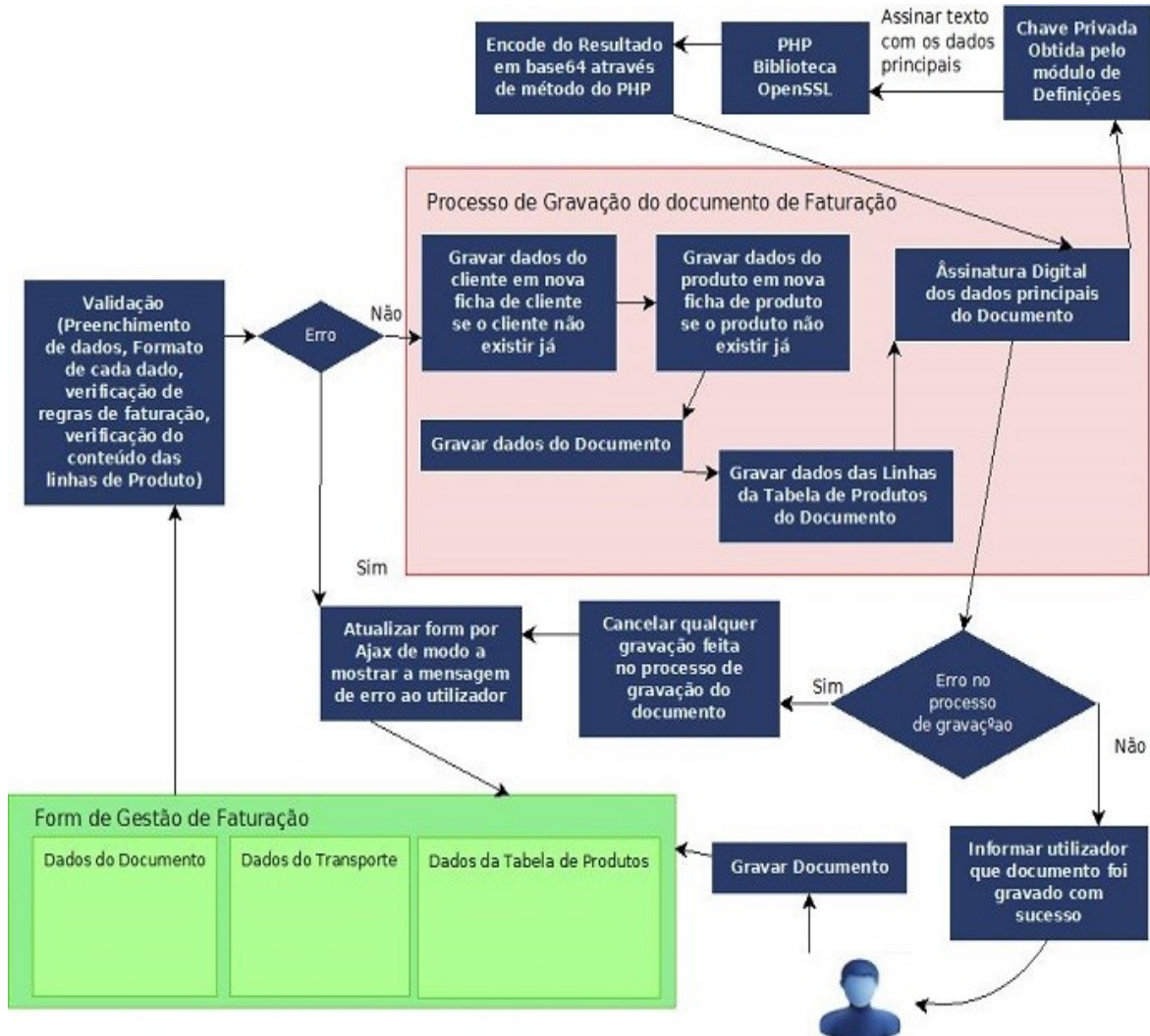


Figura 36: Gravação do Form de Gestão de Faturação

Começou por ser preparada a interação com o utilizador relativamente à adição de documentos comerciais (faturas, etc...) conforme o definido nos casos de uso, usando os forms do Drupal. Foi também necessário usar AJAX de modo a manter a coordenação entre vários elementos relacionados, por exemplo certos campos que só devem aparecer em certas condições, por exemplo caso o tipo de documento seja de Reposição de documentos (por documentos já emitidos terem sido perdido num restauro de base de dados) deverão aparecer novos campos de modo a poder identificar a série e o número do documentos que está a ser recuperado. Também a data de pagamento deverá manter a relação com o campo de dias até ao pagamento (ao mudar qualquer um destes campos o outro deverá ser atualizado de modo a manterem a consistência). Também o preenchimento dos dados dos clientes deverá ser feito com um mecanismo de auto-preenchimento por AJAX de acordo com os dados definidos na base de dados com os clientes, e o auto-preenchimento deverá ser feito por nome do cliente e por número de

contribuinte. Ao ativar o auto-preenchimento também deverá por AJAX ser obtidos o resto dos dados de modo a preencher todos os dados do cliente. Também para os dados sobre processos deverá poder ser selecionado um processo, e por AJAX ocorrer o preenchimento automáticos de dados acerca deste. Todas estas chamadas AJAX tiveram de ser organizadas de modo a não interferirem com o mecanismo de validação e submissão do form em si.

Outro desafio de interação foi a tabela com a lista de produtos e serviços, porque não existia nenhum elemento no Form API que oferecesse o nível de interação que necessitava, tendo de ser sido implementada manualmente, cada comportamento individualmente selecionado. A lista de pagamentos, que só aparece em documentos do tipo Fatura-Recibo ou Recibo e funciona de maneira semelhante à lista de produtos e serviços. Estas tabelas necessitavam de poder ser alteradas dinamicamente, permitindo não só a adição de linhas, mas também a edição e apagamento de linhas já inseridas. Isto foi tudo conseguido por AJAX, tendo de existir a garantia de que os dados inseridos eram válidos. A verificação destes elementos foi bastante complexa, pois tinha de ser verificado em cada campo da linha o tipo de dados, o formato, o número de casas decimais em valores, e existiam variadas regras referentes a faturação que tiveram de ser cumpridas, por exemplo certos tipos de documento como notas de crédito precisarem de referir quais as faturas que estavam a retificar, ou não estarem a referir documentos que não existiam, ou mesmo acrescentarem aos recibos produtos que não existiam nas faturas. Existem muitas mais validações que tiveram de ser feitas de modo a poder aceitar a correção de cada linha.

A assinatura digital é feita na altura da criação de um novo documento, e para este fim foram obtidos certificados através de OpenSSL, e com estes ficheiros de chave pública e chave privada. A chave privada é obtida temporariamente da base de dados, e os dados necessários para ser assinados são retirados da base de dados (data do documento, valor do documento, valor da assinatura do último documento do mesmo tipo e série), e é usada a biblioteca de OpenSSL para PHP[70] para assinar os dados com a chave privada. É também necessário realizar a codificação do resultado em base64[71] de forma a completar a assinatura, segundo o definido na legislação sobre a certificação de software. Depois o valor assinado é gravado na base de dados, no documento que está a ser criado. A validade desta assinatura foi confirmada através de uma aplicação de validação mencionada mais em baixo quando é mencionada a validação da exportação no formato SAF-T(PT) [72].

A exportação dos dados no formato SAF-T(PT) foi também uma tarefa complexa, pois é necessário cumprir rigorosamente o formato definido para este na Portaria 274/2013 [73], e garantir que todos os dados necessários são exportados, e com o valor e formato correto. Para este efeito foi possível efetuar a validação com ferramentas fornecidas pelo Ministério das Finanças. Uma destas ferramentas é uma aplicação baseada na Internet [74] (accedida por navegador da Internet), que verifica a estrutura do formato SAF-T(PT), indicando os erros encontrados. Esta permite, então, identificar campos necessários que estão em falta, campos com formato incorreto, etc... Outra ferramenta é parecida, mas funciona como uma aplicação executável localmente [72] (em computadores Windows), e depois de indicar o ficheiro SAF-T(PT) e o ficheiro de chave pública, permite a verificação em cada documento envolvido da assinatura digital mencionada anteriormente, e se os dados exportados correspondem aos dados assinados. Sendo assim

esta ferramenta permite assegurar que a assinatura digital é feita de acordo com as especificações dos requisitos de certificação de software.

A criação do ficheiro XML no formato SAF-T(PT) foi feita utilizando a biblioteca DOM do PHP, obtendo e processando os dados existentes na base de dados de acordo com o necessário, e adicionando ao objeto DOM os campos necessários. De acordo com o volume de informação necessária, acabou por ser uma tarefa consideravelmente grande, devido a diferenças entre exportar documentos de um tipo e outro, e campos que são necessários numa condições, e noutras não devem existir.

Na Figura 37 pode ser encontrados os passos necessários à criação do ficheiro SAF-T(PT), envolvendo a obtenção dos dados da base de dados, o seu processamento conforme o necessário, e a publicação desta informação num objeto da biblioteca DOM do PHP, de modo a obter a partir desta um ficheiro XML que é depois enviado ao utilizador.

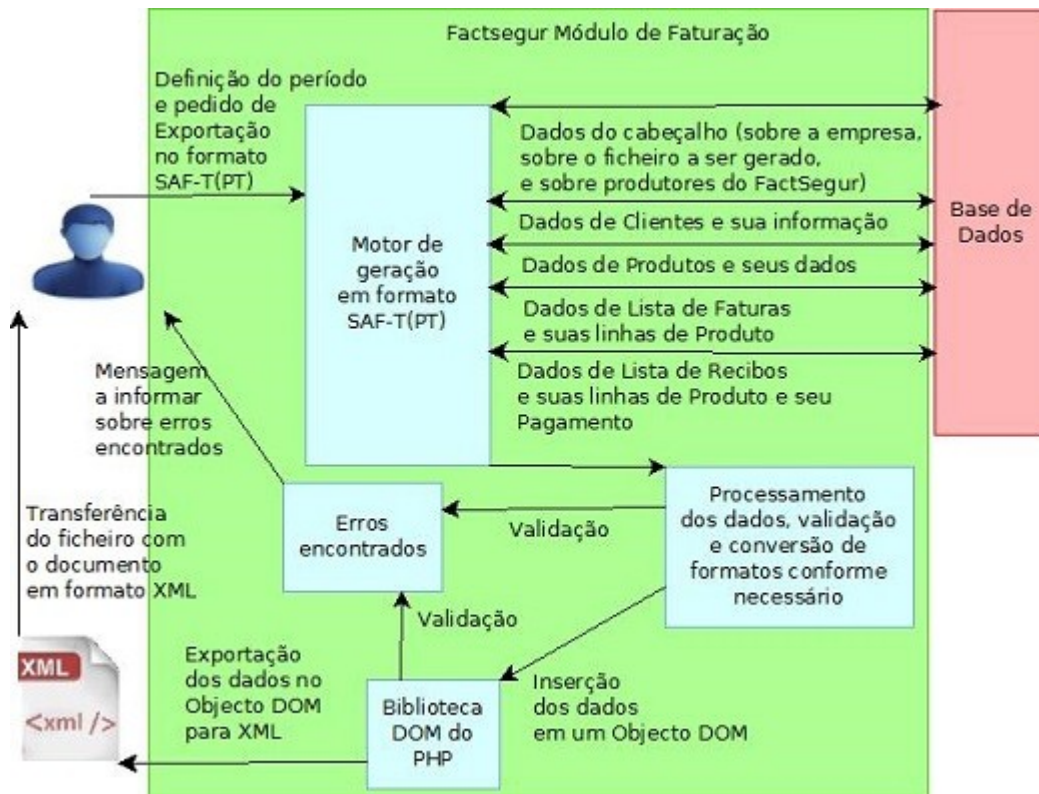


Figura 37: Esquema da exportação para formato SAF-T(PT)

Outra questão para a qual foi necessário obter soluções foi a geração de documentos, de modo ao utilizador poder receber um documento PDF com a fatura (ou outro tipo de documento), que possa posteriormente ser impresso. Foi encontrada uma ferramenta que cria documentos docx a partir de documentos de estrutura (Templates). Esta ferramenta chama-se PHPWord [75], e é uma biblioteca para PHP que permite a atribuição de variáveis em documentos de Template, e posteriormente na altura da criação de um novo documento, é possível definir no processamento do PhpWord o valor que essas variáveis devem ter, e este substitui essas variáveis pelo valor desejado, criando um novo documento docx com o resultado. Esta ferramenta também tem integração com outras ferramentas que convertem docx para PDF, mas estas foram testadas e são extremamente

limitadas, fazendo com que o documento perca a formatação, e o PDF resultante não ficasse com o aspeto nem com toda a informação necessária, perdendo conteúdo. Foi resolvida esta questão gerando o documento docx através do PHPWord e usando um serviço do LibreOffice próprio para converter o documento docx para o formato PDF.

Sendo assim foi possível criar os Templates em formato docx com a estrutura e formatação desejada para os nossos ficheiros de fatura, obter os dados a partir da base de dados e substituir os valores para estes com o PHPWord. Foi necessário existir também um mecanismo que apresentasse um código relativo à assinatura digital mencionada anteriormente (4 caracteres em posições específicas da assinatura do documento na base de dados), bem como a existência de uma tabela com os totais de todas as linhas por valor de IVA no fim do documento.

Também na geração de documentos existiu um grande desafio a nível de paginação. O PHPWord consegue facilmente inserir novas linhas numa tabela, e colocar tantas linhas quantas necessárias, mesmo fazendo a tabela trocar de página, mas não tem nenhum mecanismo que saiba o momento em que a tabela muda de uma página para outra, nem mecanismo para colocar no fim de cada página uma linha de totais. No nosso caso um dos requisitos de certificação impunha que sempre que a tabela com a lista de produtos passe de uma página para outra seja acrescentado no fim da página o valor a transportar (valor sem imposto definido até essa página), e no início da página seguinte o valor a importar (que deverá ser igual ao valor a transportar da página anterior). Isto revelou-se difícil porque caso fossem inseridas as linhas na mesma tabela já não existiria forma de colocar os valores de transporte na quebra de página da mesma tabela. O problema foi resolvido fixando o número máximo de linhas que cada página poderia suportar (de acordo com o tamanho máximo de caracteres que a linha poderia conter), e criando tabelas diferentes para cada página, com os valores de transporte entre elas. Esta solução envolveu bastante mais esforço na geração da página, pois foi necessário processar diferentemente as variáveis dependendo de quantas páginas existissem, e calcular exatamente em que ponto acabava uma página, e os valores relativos a esta, e qual a página atual, em comparação a colocar linhas novas sempre na mesma tabela, caso não fossem necessários os valores de transporte.

Na figura 38 podem ser encontrados os passos necessários para obter o ficheiro PDF, que envolve a obtenção dos dados, o cálculo das páginas que vão ser necessária de modo a inserir os dados de acordo, para as linhas de produto irem para a página certa, a inserção destes dados num documento docx através do PHPWord e um mecanismo de templates, e finalmente a utilização do libreoffice para converter o documento docx num PDF, que será enviado ao utilizador.

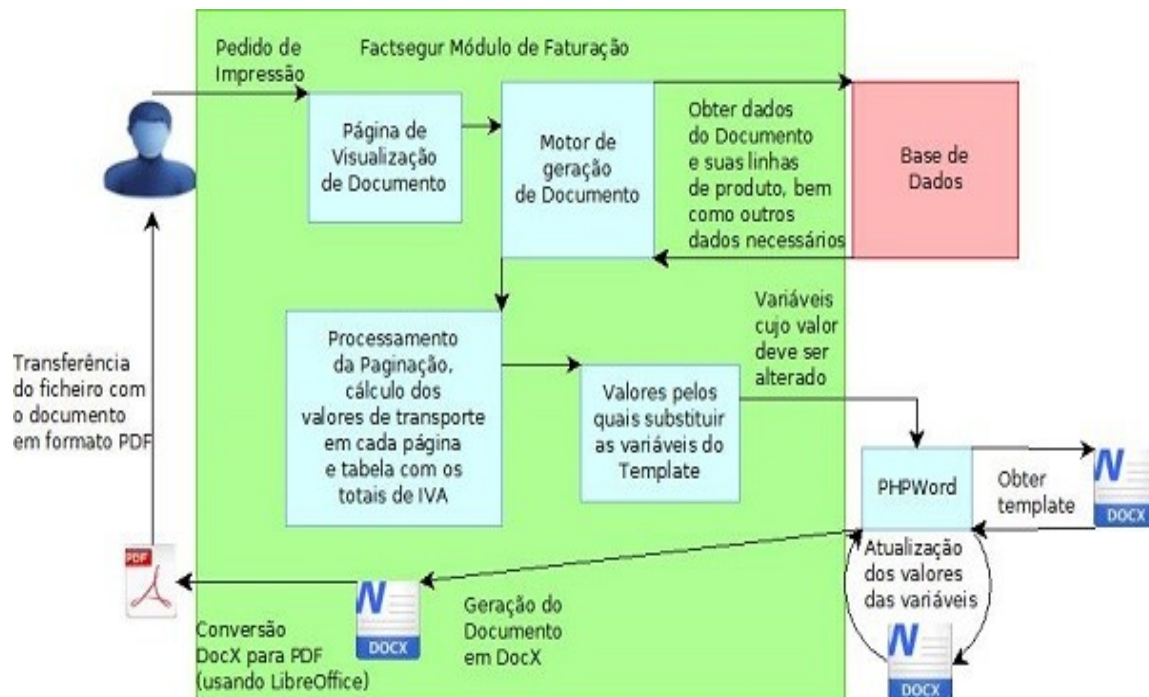


Figura 38: Esquema da Geração de Documentos

De seguida na tabela 33 pode ser consultada a lista de todas as páginas definidas pelo módulo de faturação, e a explicação do que consiste cada página.

Páginas definidas no Módulo de Faturação	
home/invoices/invoice/add	Página contendo o form relativo a documentos comerciais(faturas, recibos, etc). Permite a adição de um novo documento.
home/invoices/invoice/add/%factsegur_invoices_invoice	Página contendo o form relativo a documentos comerciais(faturas, recibos, etc). Permite a adição de um novo documento, mas preenchendo automaticamente os dados dos produtos e cliente com os dados de um documento já existente. Utilizado principalmente para emissão de um recibo a partir de uma fatura, sem ter de preencher os dados manualmente.
home/invoices/invoice/view/%factsegur_invoices_invoice	Página contendo o form relativo a documentos comerciais(faturas, recibos, etc). Permite a consulta de um documento, bem como a sua impressão para formato PDF, e a emissão de outros documentos baseado nos seus dados (ver definição da página anterior). Também permite associar dados relativos ao pagamento do documento.
home/invoices/invoice/delete/%factsegur_invoices_invoice	Página com o interface para a anulação de um Documento Comercial, respetivo a um certo identificador (serial id no fim do endereço). É pedida uma confirmação. Os dados não são apagados, podendo o documento continuar a ser consultado depois da anulação.

home/invoices/invoice/list	Página que mostra uma listagem de Documentos Comerciais, pode ser filtrada por certos dados, tais como tipo de documento, data, e cliente. Pode ser selecionado um documento para visualização. Contém informação sobre o número de documentos, e a soma dos valores importantes, como o total de imposto, e valor total de débito. Construída através do Módulo Views.
home/invoices/invoice/print/%factsegur_invoices_invoice	Página que permite obter um ficheiro PDF com os dados de um documento, de modo a poderem ser impressos os dados desse documento.
home/invoices/saftpt/generate	Página que permite a exportação dos dados relativos a um período de tempo no formato SAF-T(PT). Depois de definido esse período é possível obter o ficheiro XML com os dados nesse formato.

Tabela 33: Páginas definidas no módulo de Faturação

6.7. Módulo de Produtos

Os Produtos são usados para já só no módulo de faturação, para preencher os dados das linhas de produto dos documentos. Posteriormente talvez possa ser adicionadas funções de gestão de stocks e outras funções, mas para já apenas serve para definir um conjunto de definições que são depois importadas para as linhas de produto dos documentos comerciais.

Tal como os contactos de empresa, os produtos podem ser preenchidos automaticamente nos documentos de faturação pelo seu nome para as linhas da tabela de produtos/serviços, importando para os outros campos os dados relevantes como o tipo de produto, custo unitário, valor de imposto, entre outros. O custo unitário por defeito e o imposto podem ser alterados para outro valor, continuando a referir o mesmo produto. Tal como acontecia no módulo de contactos, também é possível adicionar manualmente um produto novo (que não existisse já anteriormente) preenchendo os seus dados no interface de adição de novos documentos de faturação. O módulo de faturação irá então criar automaticamente uma nova ficha de produto de acordo com os dados definidos na linha de produto desse documento.

Tal como acontecia para os contactos de empresa, depois de existirem documentos comerciais relativos a esse produto, este já não pode ser apagado, e alguns dos seus dados (o seu nome e tipo) já não podem ser alterados.

Tal como em outros módulos, existe uma vista (Módulo Views) que permite a visualização da lista de produtos, com um sistema de paginação com 10 itens por página, e esta lista de produtos pode ser filtrada por alguns campos, como o nome de produto, tipo de produto, e outros. Na vista pode ser selecionado um produto para visualização, edição, e apagamento.

Na Figura 39 pode ser encontrado o fluxo relativo à inserção de produtos, especialmente a validação que precisa de garantir que os dados estão corretos, e todos os dados estão preenchidos, mas também que não é permitido a alteração do nome e tipo de produtos se já existir documentos com estes preenchidos.

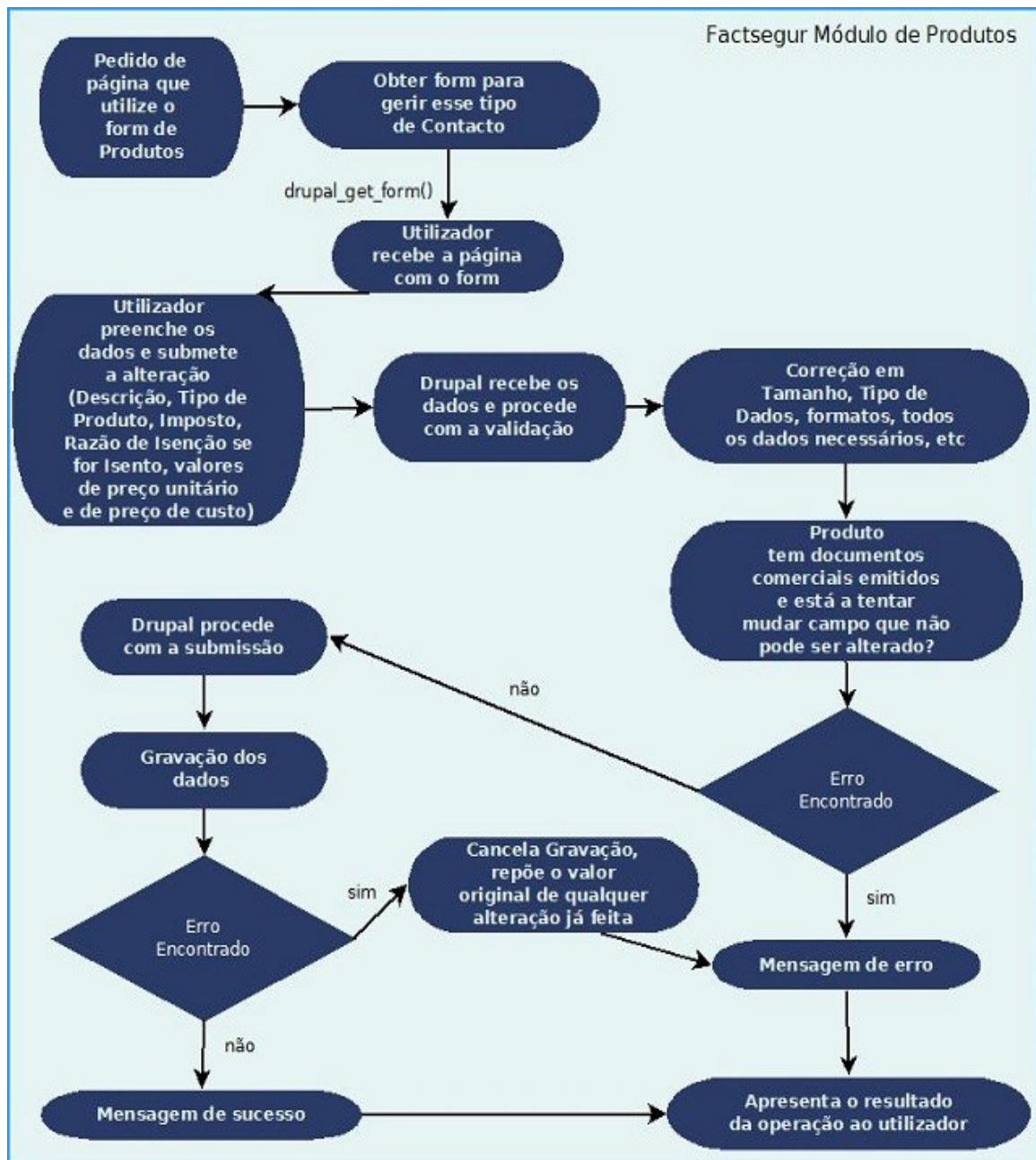


Figura 39: Diagrama de Fluxo do form de Gestão de Produtos

Na tabela 34 pode ser consultada a lista de todas as páginas definidas pelo módulo de produtos, e a explicação do que consiste cada página.

Páginas definidas no Módulo de Produtos	
home/products/add	Página contendo o form de dados para gestão de um Produto, permite a adição de um novo Produto.
home/products/edit/ %factsegur_products_product	Página contendo o form para gestão de um Produto, permite a edição dos dados relativos a um certo identificador (serial id no fim do endereço).
home/products/view/ %factsegur_products_product	Página contendo o form para gestão de um Produto, apenas permite a visualização dos dados relativos a um certo identificador (serial id no fim do endereço). A edição dos campos está bloqueada.
home/products/delete/ %factsegur_products_product	Página com o interface para a remoção de um Produto, respetivo a um certo identificador (serial id no fim do endereço). É pedida uma confirmação.
home/products/list	Página que mostra uma listagem de Produtos, pode ser filtrada por certos dados, tais como nome do produto, tipo de produto, entre outros. Criada através do Módulo Views. Pode ser selecionado um produto para visualização, edição ou apagamento.
factsegur-products/ autocomplete-product	Endereço não usado diretamente, mas internamente para obter os dados relativos ao Preenchimento Automático por Nome de Produto.

Tabela 34: Páginas definidas no módulo de Produtos

6.8. Módulo de Impostos

Este módulo de Impostos contém a definição das taxas de Imposto que podem ser selecionadas para os produtos das linhas de produto do módulo de faturação. Isto envolve não só a taxa de Imposto em si, mas também uma listagens de razões de isenção de imposto, que deverão ser indicadas, mas apenas caso o produto seja isento de imposto (valor de Imposto de 0%).

Como a informação deste módulo é necessária para o módulo de faturação foram criadas todas as tabelas relativas a este módulo, e foram construídos os métodos necessários para inserção dos dados relativos a estas, bem como os métodos para obtenção destes dados para o funcionamento dos outros módulos que dependem deste (faturação e produtos). Foram também preenchidos com os valores das taxas de imposto e razões de isenção existentes por defeito, que vêm instaladas com o programa, contendo todas as taxas de imposto definidas atualmente pelo Ministério das Finanças. Sendo assim o módulo contém o necessário para o funcionamento dos outros módulos, mas não foi preparado o interface necessário à configuração de novas taxas de imposto.

Devido a restrições temporais não foi possível completar o interface de visualização e edição das taxas de imposto, que seria semelhante às páginas existentes para edição e configuração dos produtos.

6.9. Módulo de Definições

O módulo de definições consiste inicialmente numa página onde podem ser consultadas as várias páginas de definições/configurações necessárias para o funcionamento de todos os módulos. De modo a permitir mostrar os dados de definições de outros módulos, mas sem dependerem um do outro foi criado um mecanismo onde os outros módulos poderiam facilmente publicar os endereços das suas páginas de configuração, semelhante ao criado para a publicação de permissões.

Sendo assim a página inicial com a lista de definições (accedida a partir do menu lateral esquerdo) representa um form vazio caso não estejam instalados nenhuns módulos (ou estes não conterem páginas de configuração), e à medida que são publicadas as páginas de definição dos módulos, aparecem neste interface as novas categorias (equivalentes ao nome dos módulos), e em cada categoria a lista das suas páginas de definição.

Os outros módulos (nas caixas de cima na Figura 40) publicam as suas página de definições neste módulo (método com o qual comunicam por baixo), não só com as páginas de configuração, mas também por vezes com páginas de adição de elementos desse módulo (como novos produtos, ou novos grupos de utilizadores/roles). Também existem atalhos para a visualização dos elementos dos módulos através das vistas, que normalmente também podem ser consultados através do menu lateral esquerdo).

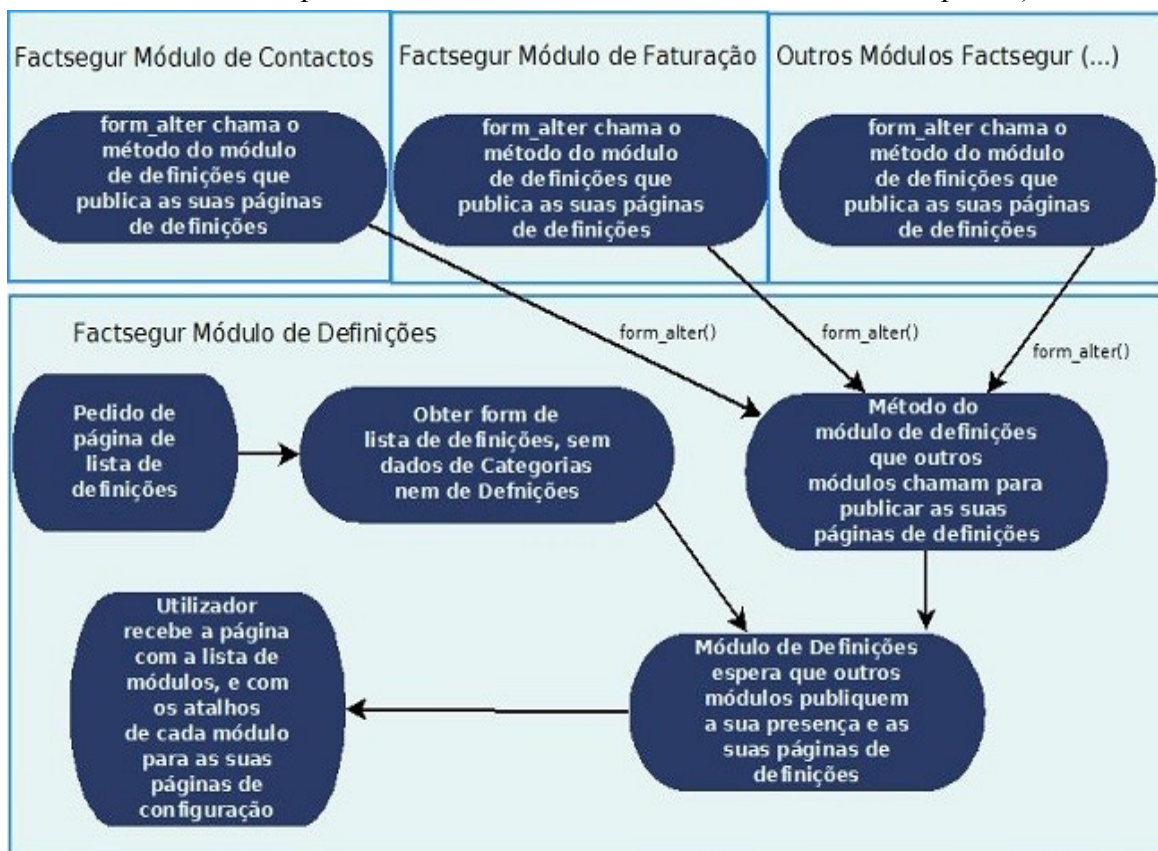


Figura 40: Diagrama de Fluxo da Página de Definições

O módulo de Definições apenas contém uma página, e pode ser verificada a sua descrição e função na tabela 35.

Páginas definidas no Módulo de Definições	
home/definitions/list	Página contendo as categorias(módulos com páginas de definições) e o atalho para as páginas de definições publicadas pelos módulos.

Tabela 35: Páginas definidas no módulo de Definições

6.10. Conclusão

Neste capítulo é explicado o que foi implementado, como foi implementado, e quais foram os maiores desafios encontrados. A implementação decorreu subordinada ao que foi definido na análise de requisitos, correspondendo aos casos de usos existentes nesta, cumprindo com o modelo de dados e especificação definidos na arquitetura.

Foi também explicado como foram usados os mecanismos do Drupal, visto ter existido algum esforço de adaptação ao Drupal, cuja forma de utilização era praticamente desconhecida. O Drupal é considerado uma mais valia, tendo oferecido um conjunto muito grande de funcionalidades que puderam ser aproveitadas para integrar com o funcionamento do programa de modo a facilitar o desenvolvimento, e a contribuir para a robustez e segurança, aderindo aos mecanismos próprios do Drupal, e aproveitando as características positivas deste.

De seguida irá ser procedida a fase de testes, onde será validado o cumprimento dos requisitos, e a robustez do produto resultante da implementação.

7. Testes

Os testes têm extrema importância para um projeto, porque é nesta fase que se verifica que a aplicação está a funcionar correctamente, dentro dos parâmetros esperados. Muitas vezes nesta fase é possível encontrar situações que não são facilmente detetadas, descobrindo erros que passaram despercebidos durante a implementação.

Durante a fase de implementação, as páginas e funcionalidades foram sendo testadas, de modo a verificar que o seu comportamento era o esperado, e que cumpriam com o definido nos casos de uso. Para além destes testes unitários, periodicamente (quando um módulo era completado) foram efetuados testes de integração que verificavam se a junção dos variados módulos (especialmente os módulos provenientes do trabalho do outros estagiário), não gerava conflitos entre estes. No nosso caso os mecanismos do Drupal promovem a independência entre os módulos, sendo que não houve erros encontrados relativos à integração entre os módulos produzidos pelos estagiários, se bem que foi encontrado um conflito com um dos módulos próprios do Drupal, cujo uso inibia o funcionamento do mecanismo de paginação. Este problema foi resolvido através de uma correção automática existente para este efeito na comunidade do Drupal.

Na altura final do projeto foram efetuados testes de desempenho, que simulavam vários níveis de carga do sistema, de modo a verificar o seu comportamento relativo a estas situações, analisando a variação do tempo de resposta de acordo com o número de utilizadores, bem como a existência de falhas na obtenção de pedidos de página. Também foram realizados alguns testes relativos a segurança, especialmente a nível de proteção contra Injeção de código SQL, e inserção programática de forms com conteúdo malicioso.

7.1. Testes Funcionais

Conforme mencionado anteriormente, durante o desenvolvimento foram efetuados testes unitários durante a implementação de cada página, e testes de integração quando cada módulo era concluído. Segundo o método de desenvolvimento adoptado, o método em Cascata, depois da fase de desenvolvimento ocorreu a fase de testes, e nestes a nível de testes funcionais foi realizado um conjunto de testes de sistema. Estes testes estão definidos no Anexo C-Documento de Testes, que irão estar representados em tabelas, organizadas por módulos, nas quais cada linha representará um caso de teste. É explicado de seguida na tabela 36 qual a estrutura destes, de modo a saber em cada coluna a informação que pode ser encontrada.

Cabeçalho (Coluna)	O que este irá conter
ID Teste	Identificação do teste. Obtido por juntar TS(teste de sistema), a identificação do caso de uso, e o número do teste relativo ao caso de uso. O teste 1 do Caso de Uso UC001 será TS001.1, o teste 2 será TS001.2, e assim por diante.
Descrição	Uma breve descrição sobre o teste.



Condições de entrada	Indica as condições necessárias para realizar o teste, o que é preciso fazer, e como o fazer.
Resultado esperado	Apresenta qual o comportamento e resultados que são esperados por parte do sistema.
Estado	Se teste passou (resultado correspondeu ao esperado) ou falhou (resultado diferente do esperado).  = passou  = falhou
Caso de uso	Identificação do caso de uso, de forma a manter a rastreabilidade com o documento de casos de uso, podendo identificar facilmente o caso de uso associado.

Tabela 36: Estrutura dos testes de sistema

Estes testes de sistema permitiram conferir se o sistema se comportava de acordo com o definido nos casos de uso, e a confirmação das funcionalidades e restrições.

7.2. Testes Não Funcionais

Foram realizados testes de carga de modo a testar o tempo de resposta, e a capacidade de servir múltiplos utilizadores em simultâneo.

Sendo que este projeto está direccionado a empresas de pequena dimensão, tipicamente com 10 a 30 empregados, nunca irá ser necessário que o programa suporte múltiplos pedidos em simultâneo, tendo uma carga normal de 1 a 5 pedidos em simultâneo. Foram variados o número de pedidos em simultâneo, e o número de pedidos efetuados, de modo a obter informação sobre como varia o tempo de resposta de acordo com a mudança destes dados.

Ao fim de uma pesquisa onde foram experimentadas algumas ferramentas indicadas como adequadas para a nossa situação, foi escolhido Apache Jmeter [76], e foram implementados planos de teste neste, tendo sido criados scripts de realização de um plano de testes específico para o nosso servidor. Os testes foram criados de maneira a serem testadas as páginas identificadas como prioritárias. Estas páginas foram as páginas relativas à inserção, edição e consulta de listagem de processos, e inserção, edição e consulta de documentos comerciais (faturas, recibos etc).

Os testes foram efetuados numa máquina virtual que estava instalada com um processador Intel Core Duo T2250¹, e com 512MB de RAM.

Os testes foram executados em conjunto relativos às páginas dos dois estágios, porque no sistema real os utilizadores estarão simultaneamente a trabalhar em processos e a emitir faturas, bem como outras funções, logo faz sentido que os nossos testes de esforço contenham acessos simultâneos a vários tipos de página, não apenas testes individuais independentes de cada página. Sendo assim o plano de testes é executado várias vezes, alterando o número de utilizadores em simultâneo (1, 2, 5 e 10 utilizadores), e fazendo

¹ <http://www.notebookcheck.net/Intel-Core-Duo-Notebook-Processor.12513.0.html>

variar também o número de pedidos a cada página por utilizador (2, 4, 6, 8, 10 pedidos). Os resultados são obtidos de acordo com cada página, e representam principalmente o número de erros obtidos, o tempo de resposta de cada pedido, entre outros dados. Esses resultados foram transferidos para tabelas, que podem ser consultadas no Anexo C- Documento de Testes, e a partir desses dados foram criados gráficos que medissem a variação do tempo de resposta nestas diferentes condições. Os gráficos serão apresentados de seguida.

Em termos de uso de recursos na máquina do servidor, foi monitorizado o uso o desempenho do CPU, do input/output do disco, e da memória, utilizando um plugin para o Jmeter chamado Perfmon, de modo a tentar perceber quais as maiores limitações deste que limitavam o desempenho.

Na Figura 41 vai ser mostrado o gráfico do tempo de resposta relativo ao número de utilizadores e número de pedidos para a página de inserção de documentos comerciais.

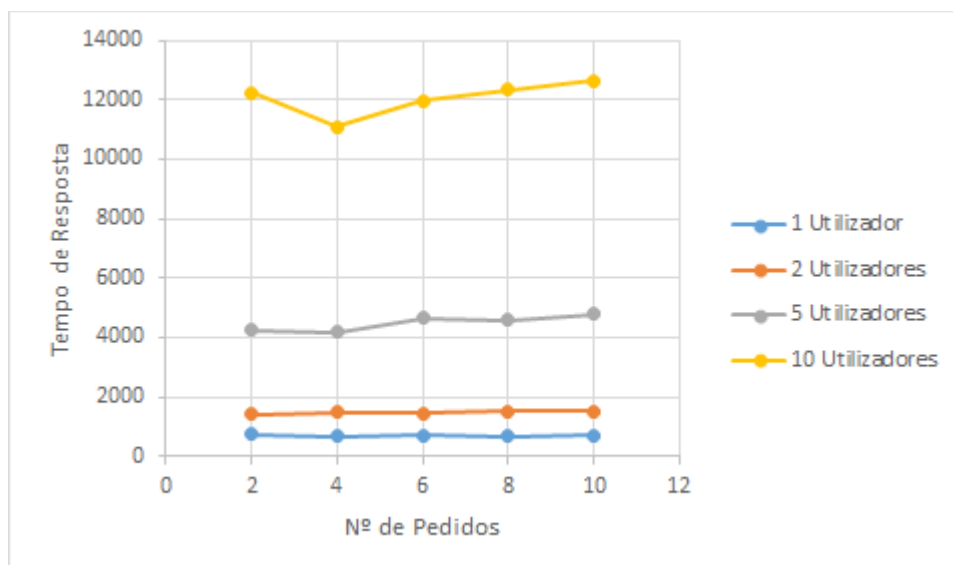


Figura 41: Tempo de resposta para página de Inserir Documentos Comerciais

Podemos verificar um valor médio por volta dos 0.9 segundos para 1 utilizador, um valor de 1.7 segundos para 2 utilizadores, 4.5 segundos para 5 utilizadores, e 12 segundos para 10 utilizadores.

É portanto verificado que o efeito da adição de mais utilizadores em simultâneo nota-se de uma forma extremamente acentuada.

De modo a poder existir uma comparação, vai ser mostrado de seguida Figura 42, com o tempo de resposta da página equivalente relativa à inserção de processos, cuja página foi criada pelo outro estagiário.

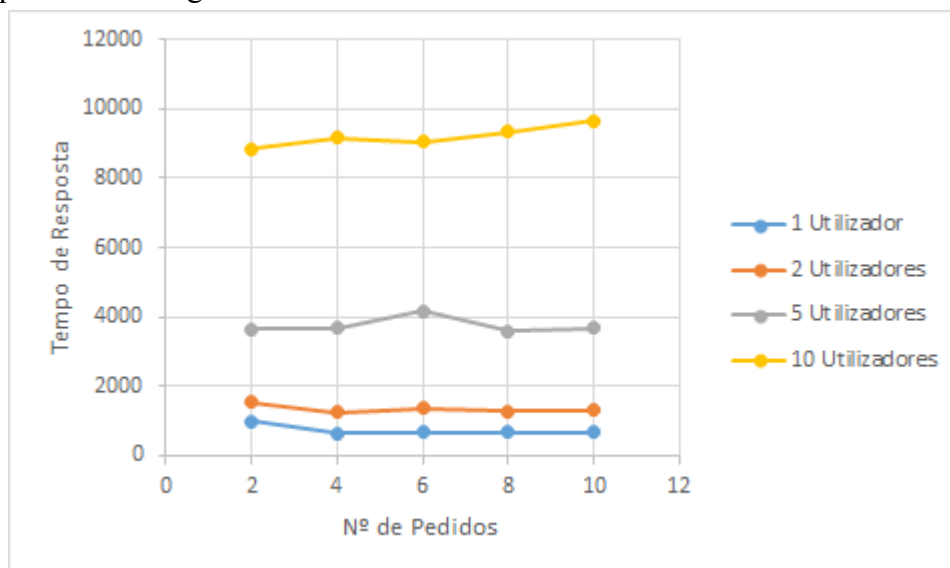


Figura 42: Tempo de resposta para página de Inserir Processos

Podemos verificar que existe o tempo de resposta aumenta consideravelmente à medida que são adicionados mais utilizadores a efetuar pedidos em simultâneo, mas que os resultados para uma página e outra não diferem muito. De seguida serão analisadas outras páginas.

Agora vamos analisar um tipo de página um pouco diferente, a página de consulta de documentos já existentes, onde é necessário carregar a página com os dados de um processo já existente na base de dados.

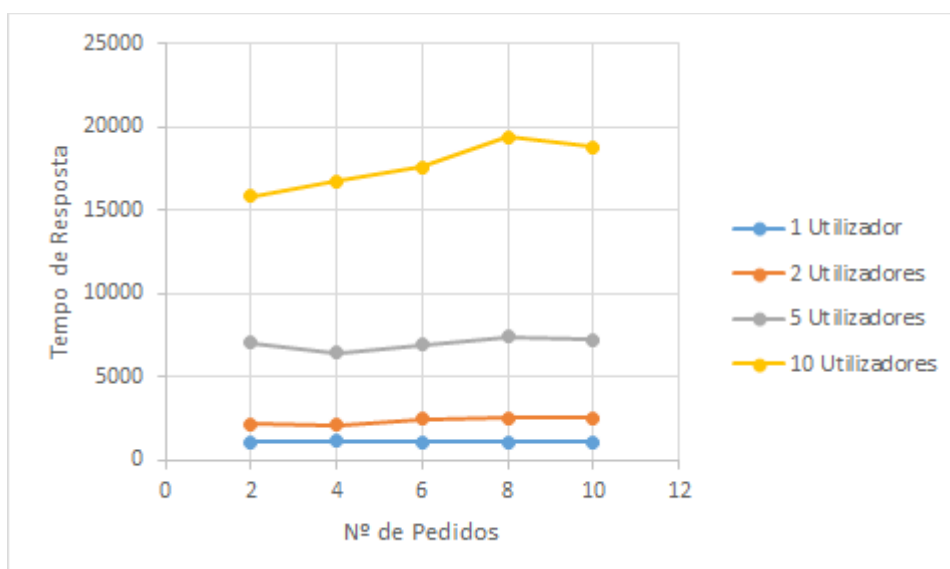


Figura 43: Tempo de resposta para página de Consultar Documento Comercial

Tal como foi feito anteriormente, vai ser mostrada uma página equivalente, a de consulta de um processo na Figura 44.

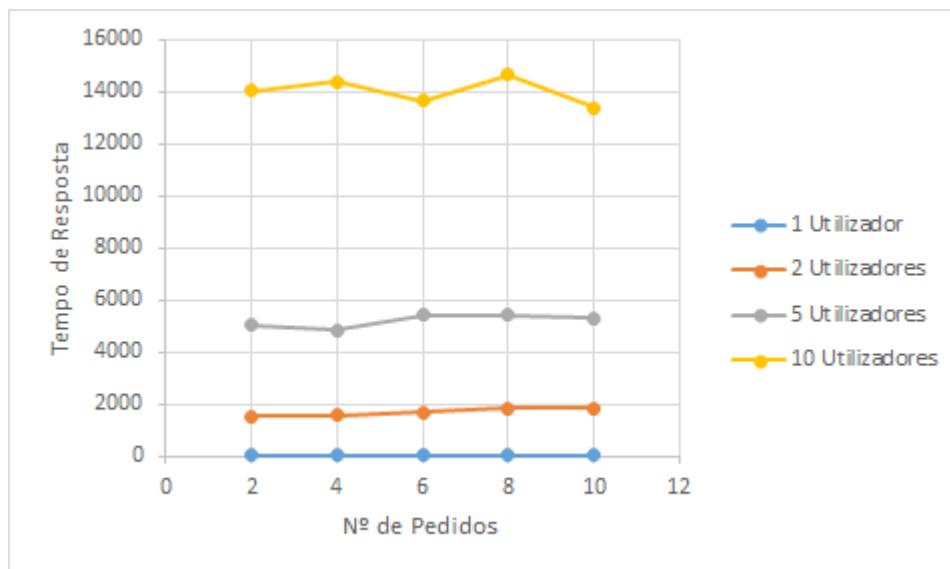


Figura 44: Tempo de resposta para página de Consultar Processo

Podemos verificar que mais uma vez os resultados estão mais ou menos parecidos, mas nota-se perfeitamente que estas janelas têm um tempo de carga superior aos da página anterior.

Agora vamos fazer a última comparação relativa ao tempo de resposta, onde são comparados os resultados para páginas que contenham as vistas do Drupal com as listagens de documentos comerciais e processos (Módulo Drupal Views).

Tal como anteriormente, começamos por ver na Figura 45 o gráfico relativo ao tempo de resposta da página de Consulta de Listagem de Documentos Comerciais.

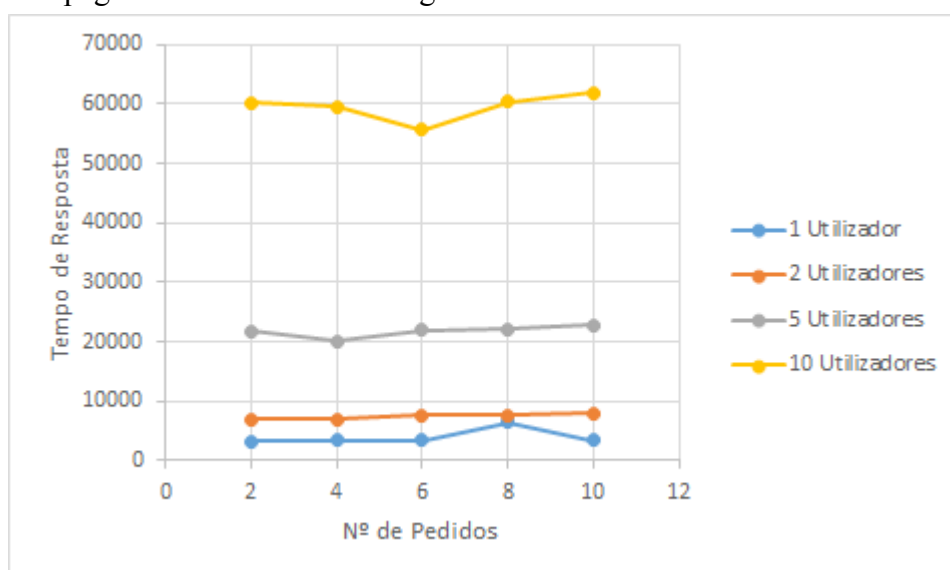


Figura 45: Tempo de resposta para página de Consultar Listagem de Documentos Comerciais

Agora vamos ver a página de Consultar Listagem de Processos na Figura 46.

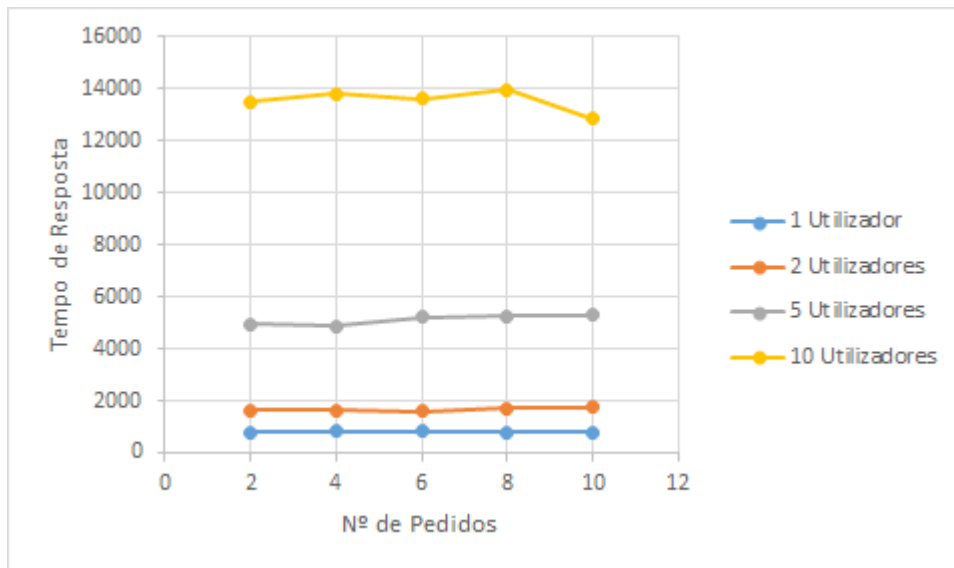


Figura 46: Tempo de resposta para página de Consultar Listagem de Processos

Podemos verificar que nesta a página de Consultar Listagem de Processos tem um desempenho bastante melhor, e também podemos verificar comparando os três gráficos que as páginas com melhor desempenho são as de inserir novo documento comercial e novo processo, seguido das páginas de consultar documentos comerciais e processos, e as páginas com desempenho pior são as páginas de listagens, especialmente a de faturação, provavelmente porque calcula valores de modo a poder atualizar somatórios de forma a poder indicar a soma de certos valores de todos os documentos que aparecem nas listagens.

De seguida apresentamos o gráfico da figura 47, que vai mostra a carga sobre os recursos do servidor durante a execução dos testes. De seguida vão ser mostrados os gráficos da carga para valores de pedidos em simultâneo, de modo a poder verificar a diferença no consumo de recursos.

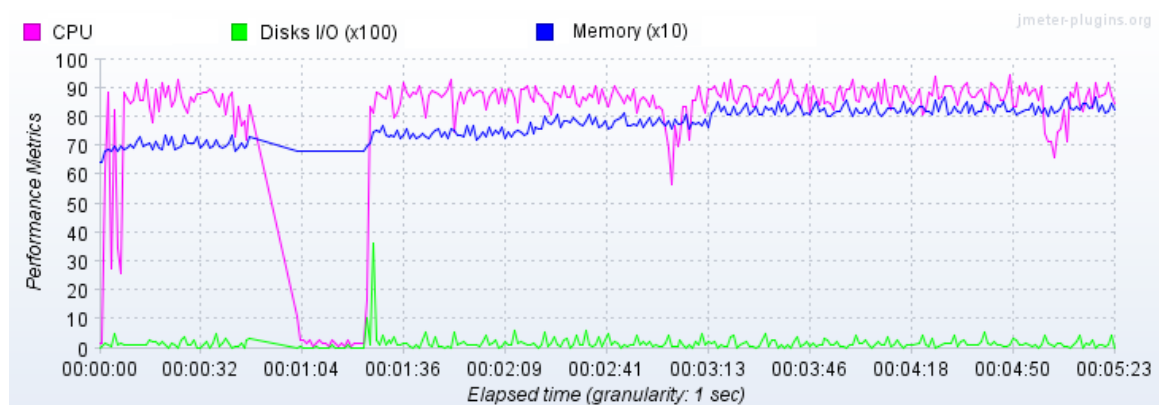


Figura 47: Recursos do servidor usados para 1 utilizador em simultâneo

Conforme podemos ver, mesmo quando os pedidos são efetuados em separado, conseguimos ver que o processador está em uso não muito intensivo, com picos. A memória aparece multiplicada por 10x, logo a sua utilização anda entre os 6.5 e 8.5

Megabytes. Finalmente podemos verificar que a curva de input/output do disco é relativamente estável.

Agora, no gráfico da figura 48 vamos poder ver a diferença entre passar de uma carga de 1 utilizador em simultâneo para 5 utilizadores em simultâneo.

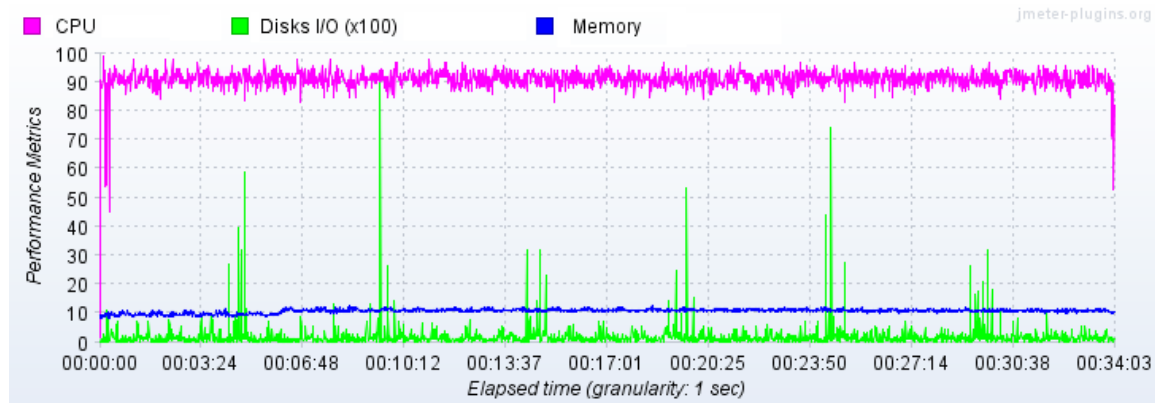


Figura 48: Recursos do servidor usados para 5 utilizadores em simultâneo

A nível de uso do processador podemos verificar que continua a andar que apesar de andar normalmente à volta dos 85% a 97% de utilização, verificamos que os picos estão muito mais compactos, acusando uma maior utilização do processador. A nível da memória o seu uso está estável por volta dos 10 Megabytes, utilizando bastante mais do que no exemplo anterior. Podemos ver também que os picos de input/output do disco estão muito mais acentuados.

Por último, no gráfico da figura 49 vamos ver o gráfico com a carga dos recursos do servidor para 15 utilizadores em simultâneo e analisar os seus picos de esforço.

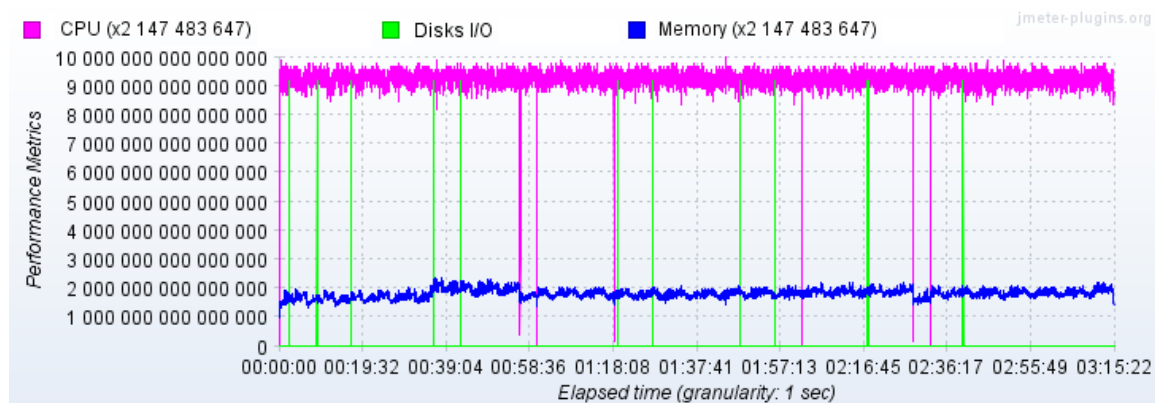


Figura 49: Recursos do servidor usados para 15 utilizadores em simultâneo

Devido aos picos enormes de Input/output do disco, a utilização de cpu e de memória estão com multiplicadores enormes de modo a aparecer os seus dados. O CPU está a variar entre valores semelhantes ao gráfico anterior (88% a 98%), e a memória já está a variar entre 10 Megabytes de utilização ao início, e 20 Megabytes para o fim, o dobro do que era usado no gráfico anterior.

De acordo com o apresentado nos gráficos podemos concluir que nesta máquina virtual o

maior limitador é o processador, cujo processador trabalha quase ao máximo logo no caso em que só é efetuado um pedido de cada vez. O Input/output do disco também é um fator que parece ter relevo, visto a acentuação dos picos deste parece ter uma relação forte com o incremento do tempo de resposta ao aumentar o número de pedidos em simultâneo. A memória não parece ser um fator muito limitador, porque estando a usar até 20 Megabytes de memória, a máquina ainda tinha bastante memória disponível.

Os resultados parecem francamente não muito animadores, mas é preciso notar que a máquina que estava disponível para testes era francamente bastante inferior ao que seria pedido a uma máquina que fosse efetuar a tarefa de servidor num ambiente de carga, sendo normal que esta não consiga um tempo de resposta muito bom. No entanto o teste já está definido, e pode voltar a ser executado posteriormente num outro ambiente, e existe a possibilidade de efetuar o teste numa máquina mais competente, e efetuar optimizações a nível do Drupal (pois estamos a correr este com a configuração base, sem optimizadores de cache, nem nenhum outro tipo), e também ao nível do Apache podem ser configurados melhoramentos.

Em relação a segurança, foi verificado que não era possível existir Injeção de SQL (especialmente em forms), porque o Drupal usa um motor interno de gestão de bases de dados, que internamente executa os comandos sql com código pré-compilado.

Também a nível de segurança, foi assegurado o acesso a dados confidenciais pelos mecanismos de permissões do Drupal, de maneira a que só utilizadores autenticados pudessem aceder a estes dados. A autenticação em si também já estava implementada pelo Drupal de modo a ocorrer de forma segura.

Finalmente, tendo sido construído a maioria do projeto usando o Form Api do Drupal, foi verificado a segurança destes, e o próprio Form API contém mecanismos importantes de validação que asseguram a correção destes. Um importante de mencionar é um mecanismo que impede a mudança programática de elementos do form no lado do utilizador (por exemplo a inserção de dados novos maliciosos num objeto SELECT antes da submissão do Form). O Drupal guarda o estado dos forms quando os envia para o utilizador, e a resposta que recebe deste é comparada, de forma a validar se não existiu alterações indevidas a este. Outro mecanismo associado a este vai atribuir uma identificação (id e token) individual a cada Form, garantindo que não é possível a submissão de forms falsos ao Drupal gerados programaticamente, pois este reconhece que essa submissão não corresponde a nenhum Form que este tenha criado inicialmente.

7.3. Conclusão

Esta fase de testes decorreu com normalidade, tendo sido encontrado alguns testes de sistema cujo resultado foi diferente do esperado, sendo que muitos destes correspondem a casos de uso que não foram implementados, ou cujo âmbito deixou de corresponder ao original. Os testes de desempenho ficaram um pouco aquém das expectativas, mas isso foi provavelmente devido ao terem sido realizadas numa máquina sem capacidade para tratar um cenário deste tipo.

8. Conclusão

Este documento reflete o trabalho desenvolvido ao longo do projeto de estágio plurianual do ano 2015/1016. O trabalho realizado no primeiro semestre começou pela consolidação de conhecimento sobre a área em questão, e a análise de mercado, de modo a obter alguma informação sobre o meio envolvente em que nos iremos inserir. De seguida foi tentado perceber junto da empresa do Ramo o que seria útil para estes e que os ajudaria no seu negócio, culminando esta investigação no documento de casos de usos que foi definido e que serviu de guia mestre para o trabalho que foi desenvolvido.

O passo seguinte foi a especificação da arquitetura que decorreu de forma a criar as bases para a construção do que estava definido nos casos de uso.

No segundo semestre decorreu a fase de implementação do projeto. Nesta começou por serem adquiridos conhecimentos práticos sobre a plataforma Drupal, e a aquisição de métodos que levassem à sua forma correta de utilização. Seguiu-se a fase de implementação, onde foi desenvolvido o trabalho de acordo com o que estava definido no planeamento, e nos documentos de casos de uso e arquitetura. A fase de implementação foi a que ocorreu com mais normalidade, sendo que as anteriores foram marcadas por alguns atrasos no planeamento. No final ocorreu uma fase de testes, para a qual não foi atribuído tanto tempo como inicialmente desejado e planeado, mas acho que o resultado final foi satisfatório, e que o processo de desenvolvimento foi positivo.

8.1. Trabalho Futuro

Este projeto ainda tem bastante espaço para crescimento, sendo que um primeiro passo será a revisão no Ministério das Finanças, de modo a ser obtida a certificação enquanto software de faturação. Também é importante a implementação dos módulos que ficaram por concretizar (impostos e mapas de faturação), e a adição de novas funcionalidades, de modo a trazer mais valias ao projeto, sendo possível assim atrair novos clientes.

Ficaram algumas funcionalidades por implementar, que não são necessárias para obter a certificação, mas que são mais valias na sua operação.

Também poderá ser necessário trabalhar a nível de optimizações, porque de acordo com o teste de carga é agora imperativo testar de novo o projeto numa máquina com capacidade suficiente para servir um projeto desta dimensão, de modo a verificar se nessas condições já é possível obter resultados melhores para os testes de carga.

Este trabalho está pensado como sendo um ponto de partida, e que possa ser enriquecido posteriormente, adicionado a este novas capacidades, e possivelmente expansão para novos tipos de mercado, oferecendo suporte a outros tipos de negócio.

9. Referências Bibliográficas

- [1] Lopes, Vasco Manuel Dias. Implementação de um sistema ERP numa PME: estudo de um caso. Dissertação de Mestrado em Gestão da Informação em DGEI, Universidade de Aveiro, 2003. [Online] [Citação: 20 de Março de 2015] Disponível em: <https://ria.ua.pt/handle/10773/10863>
- [2] Certificação de Software de Faturação, Portal das Finanças de Portugal. [Online] [Citação: 24 de Março de 2015] Disponível em: http://info.portaldasfinancas.gov.pt/pt/apoio_contribuinte/CertificacaoSoftware.htm
- [3] Portal Nacional. [Online] [Citação: 13 de Março de 2015] Disponível em: <http://portalnacional.com.pt/empresas/seguros/peritagens/>
- [4] Lista de Programas Certificados, Portal das Finanças de Portugal. [Online] [Citação: 13 de Março de 2015] Disponível em: <http://www.portaldasfinancas.gov.pt/pt/Out/consultaProgCertificadosM24.action>
- [5] SAF-T PT (Standard Audit File for Tax purposes) - Versão Portuguesa, Portal das Finanças de Portugal. [Online] [Citação: 24 de Março de 2015] Disponível em: http://info.portaldasfinancas.gov.pt/pt/apoio_contribuinte/news_saf-t_pt.htm
- [6] Portaria n.º 363/2010 de 23 de Junho. [Online] [Citação: 22 de Julho de 2015] Disponível em: <https://dre.pt/application/dir/pdf1s/2010/06/12000/0222102223.pdf>
- [7] Portaria n.º 22-A/2012 de 24 de Janeiro. [Online] [Citação: 22 de Julho de 2015] Disponível em: <http://www.desafioinformatico.pt/DocumentosCertificacao2012/portaria22A2012.pdf>
- [8] Despacho n.º 8632/2014, de 3 de Julho, Requisitos técnicos dos programas de faturação. [Online] [Citação: 10 de Abril de 2015] Disponível em: http://info.portaldasfinancas.gov.pt/NR/rdonlyres/89DB70CE-7BB5-417B-B13E-C72A912FF66E/0/Despacho_n_8632_2014_03_07.pdf
- [9] e-fatura - Página Inicial. [Online] [Citação: 5 de Julho de 2015] Disponível em: <https://faturas.portaldasfinancas.gov.pt/>
- [10] Portaria n.º 321-A/2007, de 26 de Março, Estrutura de dados do modelo normalizado de exportação de dados. [Online] [Citação: 10 de Abril de 2015] Disponível em: <http://info.portaldasfinancas.gov.pt/NR/rdonlyres/9725D3A5-DA36-45C2-BF45-22DF6AD65E6B/0/Portaria321A2007AuditFile.pdf>
- [11] Portaria n.º 1192/2009, de 08 de Outubro, Formato de ficheiro normalizado de exportação de dados SAF-T(PT), 1ª alteração ao anexo da Portaria n.º 321-A/2007, de 26 de Março. [Online] [Citação: 10 de Abril de 2015] Disponível em: http://info.portaldasfinancas.gov.pt/NR/rdonlyres/15D18787-8AA9-4060-90D5-79F168A927A4/0/Portaria_11922009.pdf
- [12] Portaria n.º 160/2013, de 23 de Abril, Formato de ficheiro normalizado de exportação de dados SAF-T(PT), 2ª alteração ao anexo da Portaria n.º 321-A/2007, de 26 de Março. [Online] [Citação: 10 de Abril de 2015] Disponível em: http://info.portaldasfinancas.gov.pt/NR/rdonlyres/4CA657D7-EEA6-4078-9157-8E837DFF4AE1/0/Portarian160_2013_23_04.pdf
- [13] Portaria n.º 274/2013, de 21 de Agosto, Formato de ficheiro normalizado de exportação de dados SAF-T(PT), 3ª alteração ao anexo da Portaria n.º 321-A/2007, de 26 de Março. [Online] [Citação: 10 de Abril de 2015] Disponível em:
- [14] CloudWorks: Software à medida Software na Cloud. [Online] [Citação: 27 de Março de 2015] Disponível em: <https://cloudworks.pt>

- [15] CloudWorks - Software e Consultadoria, Lda: Overview | LinkedIn. [Online]
[Citação: 5 de Julho de 2015] Disponível em:
<https://www.linkedin.com/company/cloudworks-pt>
- [16] MR Informática | Informatizamos padrões de Negócio (1998-2013). [Online]
[Citação: 5 de Julho de 2015] Disponível em:
<http://www.mrinformatica.pt/asp/Home.aspx>
- [17] SAP Software & Solutions | Technology and Business Applications. [Online]
[Citação: 5 de Julho de 2015] Disponível em: <http://go.sap.com/index.html>
- [18] Catálogo de Produtos - PRIMAVERA BSS. [Online] [Citação: 5 de Julho de 2015]
Disponível em: <http://pt.primaverabss.com/pt/catalogo/software-de-gestao/professional/professional/>
- [19] Solução de Gestão e ERP. [Online] [Citação: 5 de Julho de 2015] Disponível em:
<http://www.phc.pt/porta1/e/tsolgestao.aspx>
- [20] ERP | Sage. [Online] [Citação: 5 de Julho de 2015] Disponível em:
<http://www.sage.pt/solucoes/erp>
- [21] SAP Claims Management as Part of SAP ERP 6.0. [Online] [Citação: 27 de Março de 2015] Disponível em: <http://help.sap.com/insurance-cm>
- [22] Catálogo de Produtos - PRIMAVERA BSS. [Online] [Citação: 5 de Julho de 2015]
Disponível em: <http://pt.primaverabss.com/pt/catalogo/software-faturacao/express-faturacao/express/>
- [23] Sage One. [Online] [Citação: 5 de Julho de 2015] Disponível em:
<http://www.sage.pt/solucoes/faturacao/startups-e-empresendedores/sage-one>
- [24] Preço - PHC FX. [Online] [Citação: 5 de Julho de 2015] Disponível em:
<http://pt.phcfx.com/suites/phc-fx-bill/preco/>
- [25] Preços - Projecto Colibri. [Online] [Citação: 5 de Julho de 2015] Disponível em:
<http://www.projectocolibri.com/precos>
- [26] Eclipse Pubic License - Version 1.0. [Online] [Citação: 5 de Julho de 2015]
Disponível em: <https://www.eclipse.org/legal/epl-v10.html>
- [27] Planos e Preços do Software de Facturação | InvoiceXpress. [Online] [Citação: 5 de Julho de 2015] Disponível em: <https://invoicexpress.com/planos-precos>
- [28] Software - Facturação e Stocks. [Online] [Citação: 5 de Julho de 2015] Disponível em: <http://www.datagen.eu/index.php?it=13&sit=15&l=0>
- [29] WISEDAT Software Solutions - Software Certificado Comercial - Facturação e POS. [Online] [Citação: 5 de Julho de 2015] Disponível em:
<http://www.wisedat.pt/Produtos/COM.aspx>
- [30] Software Facturação online programa de facturação certificado. [Online] [Citação: 5 de Julho de 2015] Disponível em: <https://www.keyinvoice.com/index.php#planos>
- [31] ERP/CRM Management Solutions. [Online] [Citação: 5 de Julho de 2015]
Disponível em: <http://www.gestix.com/go/solutions/enterprise/?showPrice=0#prices>
- [32] Moloni - Facturação Online Grátis. [Online] [Citação: 5 de Julho de 2015]
Disponível em: <https://www.moloni.com/index.php>
- [33] eComércio - Conheça a Solução Ideal para a sua Empresa. [Online] [Citação: 5 de Julho de 2015] Disponível em: <https://www.ecomercio.pt/modalidades.aspx>
- [34] FaturaVirtual - Faturação Grátis Online. [Online] [Citação: 5 de Julho de 2015]
Disponível em: https://www.faturavirtual.com/product/pricing/?show_all=1&period=year&goto=plan-base-year
- [35] Evaristo - Memória Persistente. [Online] [Citação: 5 de Julho de 2015] Disponível em: <https://t5.m16e.com/wiki/pageviewer/view/6>
- [36] Free Accounting Software | GnuCash. [Online] [Citação: 5 de Julho de 2015]
Disponível em: <http://www.gnucash.org/>

- [37] Waterfall model. [Online] [Citação: 4 de Julho de 2015] Disponível em: https://en.wikipedia.org/wiki/Waterfall_model
- [38] Scrum (software development). [Online] [Citação: 4 de Julho de 2015] Disponível em: [https://en.wikipedia.org/wiki/Scrum_\(software_development\)](https://en.wikipedia.org/wiki/Scrum_(software_development))
- [39] Requirements Analysis. [Online] [Citação: 11 de Junho de 2015] Disponível em: http://en.wikipedia.org/wiki/Requirements_analysis
- [40] Clegg, Dai; Barker, Richard. Case Method Fast-Track: A RAD Approach. em Addison-Wesley, 1994. [Online] [Citação: 6 de Maio de 2015] Disponível em:
- [41] MoSCoW Prioritisation. [Online] [Citação: 6 de Maio de 2015] Disponível em: <http://www.dsdm.org/content/10-moscow-prioritisation>
- [42] Home | CloudFlare | The Web Performance & Security Company. [Online] [Citação: 5 de Julho de 2015] Disponível em: <https://www.cloudflare.com/>
- [43] Welcome! - The Apache HTTP Server Project. [Online] [Citação: 5 de Julho de 2015] Disponível em: <http://httpd.apache.org/>
- [44] PostgreSQL: The World's most advanced open source database. [Online] [Citação: 5 de Julho de 2015] Disponível em: <http://www.postgresql.org/>
- [45] Nagios - The Industry Standard in IT Infrastructure Monitoring. [Online] [Citação: 5 de Julho de 2015] Disponível em: <https://www.nagios.org/>
- [46] CloudFlare - Wikipédia, the free encyclopedia. [Online] [Citação: 5 de Julho de 2015] Disponível em: <https://en.wikipedia.org/wiki/CloudFlare>
- [47] Affordable advanced DDoS protection and Mitigation. [Online] [Citação: 5 de Julho de 2015] Disponível em: <https://www.cloudflare.com/ddos>
- [48] CloudFlare Review 2015. [Online] [Citação: 5 de Julho de 2015] Disponível em: <http://website-security-and-performance-review.toptenreviews.com/cloudflare-review.html>
- [49] Nagios - Wikipédia, the free encyclopedia. [Online] [Citação: 5 de Julho de 2015] Disponível em: <https://en.wikipedia.org/wiki/Nagios>
- [50] Application Monitoring - Nagios. [Online] [Citação: 5 de Julho de 2015] Disponível em: <https://www.nagios.com/solutions/application-monitoring>
- [51] PostgreSQL - Wikipédia, the free encyclopedia. [Online] [Citação: 5 de Julho de 2015] Disponível em: <https://en.wikipedia.org/wiki/PostgreSQL>
- [52] NoSQL for the Enterprise. [Online] [Citação: 5 de Julho de 2015] Disponível em: <http://www.enterprisedb.com/nosql-for-enterprise>
- [53] Why I Choose PostgreSQL Over MySQL/MariaDB - Slashdot. [Online] [Citação: 5 de Julho de 2015] Disponível em: http://posulliv.github.io/images/second_query_response_time.png
- [54] Apache HTTP Server - Wikipédia, the free encyclopedia. [Online] [Citação: 5 de Julho de 2015] Disponível em: https://en.wikipedia.org/wiki/Apache_HTTP_Server
- [55] Is Drupal Secure?. [Online] [Citação: 21 de Outubro] Disponível em: <https://www.drupal.org/documentation/is-drupal-secure>
- [56] The Drupal Overview | Drupal.org. [Online] [Citação: 5 de Julho de 2015] Disponível em: <https://www.drupal.org/getting-started/before/overview>
- [57] Dries Buytaert - Wikipédia, the free encyclopedia. [Online] [Citação: 5 de Julho de 2015] Disponível em: https://en.wikipedia.org/wiki/Dries_Buytaert
- [58] Drupal - Wikipédia, the free encyclopedia. [Online] [Citação: 5 de Julho de 2015] Disponível em: <https://en.wikipedia.org/wiki/Drupal>
- [59] CMS Comparison - WordPress vs Joomla vs Drupal / WebsiteSetup.org. [Online] [Citação: 5 de Julho de 2015] Disponível em: <http://websitesetup.org/cms-comparison-wordpress-vs-joomla-drupal/>

Disponível em: <http://robknight.org.uk/blog/2011/02/explaining-architectural-tiers-drupal/>

[61] MVC vs. PAC. [Online] [Citação: 5 de Julho de 2015] Disponível em: <http://www.garfieldtech.com/blog/mvc-vs-pac>

[62] The Drupal Overview. [Online] [Citação: 1 de Outubro] Disponível em: <https://www.drupal.org/node/265726>

[63] php - How Drupal Works?. [Online] [Citação: 5 de Julho de 2015] Disponível em: <http://stackoverflow.com/questions/1068556/how-drupal-works>

[64] Render Arrays in Drupal 7. [Online] [Citação: 21 de Outubro] Disponível em: <https://www.drupal.org/node/930760>

[65] Conceitos: Ambiente de Desenvolvimento. [Online] [Citação: 18 de Janeiro de 2016] Disponível em: http://www.wthreex.com/rup/process/workflow/environm/co_deven.htm

[66] Best PHP IDE in 2014 – Survey Results. [Online] [Citação: 18 de Janeiro de 2016] Disponível em: <http://www.sitepoint.com/best-php-ide-2014-survey-results/>

[67] Free for students: Professional developer tools from JetBrains. [Online] [Citação: 18 de Janeiro de 2016] Disponível em: <https://www.jetbrains.com/student/>

[68] Installation Guide | Drupal.org. [Online] [Citação: 18 de Janeiro de 2016] Disponível em: <https://www.drupal.org/documentation/install>

[69] Form API. [Online] [Citação: 19 de Janeiro de 2016] Disponível em: <https://www.drupal.org/node/37775>

[70] PHP: OpenSSL - Manual. [Online] [Citação: 21 de Janeiro de 2016] Disponível em: <http://php.net/manual/en/book.openssl.php>

[71] RFC 4648 - The Base16, Base32, and Base64 Data Encodings. [Online] [Citação: 21 de Janeiro de 2016] Disponível em: <https://tools.ietf.org/html/rfc4648>

[72] Certificação de Software de Faturação : Aplicação de Validação. [Online] [Citação: 21 de Janeiro de 2016] Disponível em: http://info.portaldasfinancas.gov.pt/pt/apoio_contribuinte/CertificacaoSoftware.htm

[73] Portaria n.º 274/2013 de 21 de Agosto. [Online] [Citação: 21 de Janeiro de 2016] Disponível em: http://info.portaldasfinancas.gov.pt/NR/rdonlyres/BA9FB096-D482-445D-A5DB-C05B1980F7D7/0/Portaria_274_2013_21_09.pdf

[74] Validador Saft. [Online] [Citação: 21 de Janeiro de 2016] Disponível em: <http://info.portaldasfinancas.gov.pt/apps/saft-pt03/>

[75] Welcome to the homepage of PHPWord!. [Online] [Citação: 21 de Janeiro de 2016] Disponível em: <https://phpword.codeplex.com/>

[76] Apache JMeter. [Online] [Citação: 26 de Janeiro de 2016] Disponível em: <http://jmeter.apache.org/>

Anexos