

Mestrado em Engenharia Informática

Estágio

Relatório Final

# Modelação de processos em OpenSource

Luis Filipe Semedo Castanheira

pipe@student.dei.uc.pt

Orientador DEI:

Prof. Dr. Fernando Boavida

Orientadores GSIIC:

Eng<sup>o</sup> Mário Bernardes

Eng<sup>o</sup> Jorge China

Data: Setembro 2014



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

*"When it comes to software, I much prefer free software, because I have very seldom seen a program that has worked well enough for my needs, and having sources available can be a life-saver."*

Linus Torvalds

Criador do Linux e impulsionador do software livre

*"It's amazing what you can get on open source now if you actually use the right search engines to find the material."*

Sir Gus O'Donnell

Economista, Cabinet Secretary entre 2005 e 2012

## **Resumo**

A necessidade de implementar serviços acessíveis ao comum dos cidadãos, com interfaces de qualidade, fornecendo métricas de desempenho e mecanismos claros de auditoria e acompanhamento, tem motivado as organizações públicas e privadas no sentido de dotar os principais processos inerentes à sua atividade com os mecanismos capazes de responder a estes requisitos.

Neste contexto, muitas organizações têm optado pela utilização de produtos comerciais para implementação dos seus principais processos de negócio. No entanto, esta aproximação não está ao alcance de muitas organizações em resultado dos custos de licenciamento e manutenção dos produtos utilizados no desenvolvimento.

O objetivo deste estágio consiste na avaliação da viabilidade da utilização por parte da Universidade de Coimbra da solução de código aberto OpenERP para a modelação dos seus processos organizacionais, e a sua integração com sistemas aplicativos existentes. Esta avaliação permitirá eventualmente libertar esta instituição das obrigações contratuais que a ligam atualmente às soluções proprietárias que contratou para a execução dessas mesmas tarefas, levando assim a uma previsível redução de custos.

Para efetuar essa avaliação, utilizou-se o OpenERP para replicar a implementação de processos desta instituição, comparando os resultados obtidos com os resultados apresentados pela solução proprietária existente.

## **Palavras chave**

Administração Pública, Customer Relationship Management, Enterprise Resource Planning, Modelação de Processos, OpenERP, Open Source, Software Livre

## **Agradecimentos**

À Rute, por ser a melhor companheira do mundo, pela imensa paciência e pelo seu infindável apoio. Sem ela não teria sido possível chegar aqui.

À minha família, por estar sempre a meu lado, e por sempre me ter apoiado, em qualquer circunstância.

Aos colegas da DSI, pelo fantástico ambiente de trabalho proporcionado.

Por último, mas não menos importantes, aos meus orientadores, pelo apoio dado ao longo do projeto.

A todos, muito obrigado.

## Lista de Acrónimos

|           |  |
|-----------|--|
| AGPL      | Afferro General Public Licence   |
| BPMN      | Business Process Model and Notation  |
| CMS       | Content Management System  |
| DEI       | Departamento da Universidade de Coimbra  |
| ERP       | Enterprise Resource Planning   |
| FCTUC     | Faculdade de Ciências da Universidade de Coimbra   |
| GIMP      | GNU Image Manipulation Program   |
| GNU       | GNU is not Unix  |
| GRH       | Grupo Recursos Humanos   |
| GSIIC-DSI | Divisão de Sistemas de Informação do serviço de Gestão de Sistemas e Infraestruturas de Informação e Comunicação |
| GTK       | The GIMP Toolkit   |
| HTTP      | Hypertext Transfer Protocol  |
| HTTPS     | Hypertext Transfer Protocol Secure   |
| JSON      | JavaScript Object Notation   |
| LDAP      | Lightweight Directory Access Protocol  |
| MVC       | Model-View-Controller  |
| ORM       | Object Relational Mapping  |
| OSS       | Open Source Software   |
| PoS       | Point of Sale  |
| RH        | Recursos Humanos   |
| RPC       | Remote Procedure Call  |
| SIGE      | Sistemas de Integrados de Gestão Empresarial   |
| SQL       | Structured Query Language  |
| TCO       | Total Cost of Ownership  |
| UC        | Universidade de Coimbra  |
| UCPA      | Portal de Acesso da Universidade de Coimbra  |
| XML       | eXtensible Markup Language   |

## Índice de Conteúdos

|          |   |    |
|----------|---|----|
| 1.       | Introdução .....  | 1  |
| 1.1.     | Motivação .....   | 1  |
| 1.2.     | Objetivos .....   | 2  |
| 1.3.     | Contribuições .....   | 3  |
| 1.4.     | Plano de estágio.....   | 4  |
| 1.4.1.   | Planeamento de tarefas para o primeiro semestre.....                      | 4  |
| 1.4.1.1. | Cronograma.....   | 5  |
| 1.4.2.   | Planeamento de tarefas para o segundo semestre .....                      | 5  |
| 1.4.2.1. | Cronograma.....   | 6  |
| 1.5.     | Estrutura do documento.....   | 7  |
| 2.       | Estado da arte .....  | 8  |
| 2.1.     | Apresentação da plataforma <i>OutSystems</i> .....                        | 8  |
| 2.1.1.   | Custos inerentes à utilização da plataforma <i>OutSystems</i> .....       | 9  |
| 2.2.     | Custos associados ao <i>software Open Source</i> .....                    | 10 |
| 2.2.1.   | Definição de Custo Total de Propriedade .....                             | 11 |
| 2.3.     | Custos inerentes à utilização do <i>OpenERP</i> .....                     | 12 |
| 2.4.     | Custo do <i>OutSystems</i> vs custo do <i>OpenERP</i> .....               | 13 |
| 2.5.     | Conclusões .....  | 15 |
| 3.       | <i>OpenERP</i> : abordagem teórica.....                                   | 16 |
| 3.1.     | Formação teórico-prática.....   | 16 |
| 3.2.     | Arquitetura .....   | 16 |
| 3.2.1.   | Padrão <i>Model-View-Controller</i> .....                                 | 18 |
| 3.2.1.1. | Object Relational Mapping.....  | 19 |
| 3.2.2.   | Escalabilidade.....   | 21 |
| 3.3.     | Instalação de ambiente de desenvolvimento e testes.....                   | 21 |
| 3.4.     | Instalação de módulos .....   | 22 |
| 3.5.     | Cópias de segurança.....  | 22 |
| 3.5.1.   | Cópias de segurança automatizadas .....                                   | 24 |
| 3.6.     | Auditoria.....  | 25 |
| 3.6.1.   | Bugs encontrados no módulo <i>audit_trail</i> .....                       | 26 |
| 3.7.     | Autenticação centralizada.....  | 26 |
| 3.7.1.   | <i>Bugs</i> encontrados no módulo <i>ldap_populate</i> .....              | 28 |
| 3.8.     | Motor de <i>workflow</i> .....  | 28 |
| 3.9.     | Estudo da camada de apresentação .....                                    | 28 |
| 3.10.    | Comparação de funcionalidades do <i>OutSystems</i> e <i>OpenERP</i> ..... | 29 |

|   |    |
|---|----|
| 3.11. Conclusões .....  | 29 |
| 4. Exploração do <i>OpenERP</i> em contexto de trabalho .....                                 | 30 |
| 4.1. Tarefas executadas .....   | 30 |
| 4.1.1. Instalação e configuração do <i>PoS</i> .....  | 30 |
| 4.1.2. Reuniões com o cliente.....  | 30 |
| 4.1.2.1. Testes e correção de bugs.....   | 31 |
| 4.1.3. Produção de documentação .....   | 31 |
| 5. Implementação de processos no <i>OpenERP</i> .....   | 32 |
| 5.1. Identificação e estudo dos processos a implementar.....                                  | 32 |
| 5.2. Análise de requisitos transversais do <i>workflow</i> do <i>LUGUS – OutSystems</i> ..... | 32 |
| 5.2.1. Características transversais a todos os tipos de pedido .....                          | 33 |
| 5.3. Análise prática do motor de <i>workflow</i> do <i>OpenERP</i> .....                      | 33 |
| 5.4. Implementação do Processo A: Pedido genérico .....                                       | 35 |
| 5.4.1. Análise do processo, validação e adaptação da especificação .....                      | 35 |
| 5.4.2. Fase de desenvolvimento.....   | 37 |
| 5.4.2.1. Application Builder vs Escrita de Código.....  | 38 |
| 5.4.2.2. Dificuldades encontradas durante o desenvolvimento .....                             | 39 |
| 5.4.2.3. Integração com sistemas externos .....   | 42 |
| 5.4.3. Testes .....   | 43 |
| 5.5. Processo B: Pedido de deslocação em serviço .....  | 43 |
| 6. Análise comparativa das implementações .....   | 45 |
| 6.1.1. Análise funcional e desempenho.....  | 45 |
| 6.1.1.1. Usabilidade .....  | 46 |
| 6.1.1.2. Testes de carga .....  | 49 |
| 6.1.2. Esforço desenvolvimento .....  | 55 |
| 7. Conclusões .....   | 57 |
| 7.1. Trabalhos realizados .....   | 57 |
| 7.2. Principais dificuldades encontradas .....  | 58 |
| 7.3. Considerações finais .....   | 59 |
| 7.4. Trabalho futuro.....   | 60 |
| 8. Bibliografia .....   | 62 |
| 9. Anexos.....  | 64 |

## Índice de Figuras

|  |    |
|--|----|
| Figura 1 - Cronograma de tarefas para o primeiro semestre .....                                | 5  |
| Figura 2 - Cronograma de tarefas para o segundo semestre .....                                 | 7  |
| Figura 3 - Custo total de Propriedade - Custos visíveis vs custos invisíveis .....             | 11 |
| Figura 4 - Arquitetura do OpenERP.....   | 17 |
| Figura 5 - Padrão MVC .....  | 18 |
| Figura 6 - Interface de instalação de módulos - Screenshot da aplicação .....                  | 22 |
| Figura 7 - Mecanismo de cópias de segurança do OpenERP - Screenshot da aplicação.....          | 23 |
| Figura 8 - Caixa de Login OpenERP - Screenshot da aplicação .....                              | 23 |
| Figura 9 - Exemplo de regra de auditoria - Screenshot da aplicação .....                       | 25 |
| Figura 10 - Logs gerados pelo Audit Trail - Screenshot da aplicação .....                      | 26 |
| Figura 11 - Configuração LDAP do OpenERP - Screenshot da aplicação .....                       | 27 |
| Figura 12 - Fluxograma da implementação do pedido genérico em OutSystems .....                 | 36 |
| Figura 13 - Modelação do pedido genérico em notação BPMN.....                                  | 37 |
| Figura 14 - Application Builder - View Editor - Screenshot da aplicação.....                   | 38 |
| Figura 15 - Gráfico de análise da utilidade de respostas recebida – StackOverflow .....        | 40 |
| Figura 16 - Gráfico de análise da utilidade de respostas recebida - Fórum OpenERP .....        | 41 |
| Figura 17 - Gráfico de análise de respostas dadas pelo estagiário – StackOverflow.....         | 41 |
| Figura 18 - Gráfico de análise de respostas dadas pelo estagiário - Fórum OpenERP .....        | 42 |
| Figura 19 - Criar pedido no LUGUS - Screenshot da aplicação.....                               | 46 |
| Figura 20 - Criar pedido no OpenERP - Screenshot da aplicação .....                            | 47 |
| Figura 21 - Visualização dos dados de um pedido genérico no LUGUS - Screenshot da aplicação .  | 48 |
| Figura 22 - Visualização de dados do pedido genérico no OpenERP - Screenshot da aplicação..... | 49 |
| Figura 23 - Teste de carga - Cenário 1.....  | 52 |
| Figura 24 - Gráfico de tempo de resposta - Cenário 1 .....                                     | 53 |
| Figura 25 - Teste de carga - Cenário 2.....  | 53 |
| Figura 26 - Gráfico de tempo de resposta - Cenário 2 .....                                     | 54 |
| Figura 27 - Teste de carga - Cenário 3.....  | 54 |
| Figura 28 - Gráfico com tempo de resposta - Cenário 3.....                                     | 55 |



## **Índice de Tabelas**

|  |    |
|--|----|
| Tabela 1 - Componentes do cálculo do TCO .....   | 12 |
| Tabela 2 - Análise comparativa do custo de substituição do OutSystems pelo OpenERP ..... | 14 |
| Tabela 3 - Comparação entre funcionalidades do OutSystems e OpenERP .....                | 29 |
| Tabela 4 - Análise funcional comparativa - Pedido genérico - LUGUS vs OpenERP.....       | 45 |

# 1. Introdução

O presente estágio foi desenvolvido, no ano letivo 2013 / 2014, no âmbito da disciplina de Dissertação / Estágio inserida no Mestrado em Engenharia Informática do Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

O estágio decorreu nas instalações da Divisão de Sistemas de Informação do serviço de Gestão de Sistemas e Infraestruturas de Informação e Comunicação (GSIIC-DSI), no Pólo II da Universidade de Coimbra, sob orientação do Professor Doutor Fernando Boavida por parte do Departamento de Engenharia Informática da Universidade de Coimbra e dos Engenheiros Mário Bernardes (Diretor do Serviço) e Jorge China (Chefe de Divisão), por parte da instituição acolhedora do estágio.

A equipa de trabalho destacada para este projeto foi constituída apenas pelo estagiário, não sendo de menosprezar o importante feedback recebido não só pelos orientadores, mas também pela restante equipa da Divisão de Sistemas de Informação.

## 1.1. Motivação

No atual contexto da gestão empresarial, dado o aumento da competitividade, todos os dias surgem novos desafios que devem ser ultrapassados pelas instituições. Necessidades como a de implementar serviços acessíveis, com interfaces de qualidade, fornecendo métricas de desempenho e mecanismos de auditoria e acompanhamento, têm motivado as organizações públicas e privadas no sentido de dotar os principais processos inerentes à sua atividade com os mecanismos capazes de responder a estes requisitos.

Para ultrapassar estes desafios é necessário uma enorme cooperação entre funcionários, dirigentes, gestores e todos intervenientes nos processos organizacionais. Recorrendo às tecnologias de informação, encontram-se importantes ferramentas estratégicas que auxiliam a coordenação entre as diferentes áreas das instituições, levando a uma melhoria da eficiência e organização das mesmas, levando a uma conseqüente otimização de custos.

Uma forma de atingir estes objetivos é a utilização de ferramentas de planeamento de recursos, os Sistemas Integrados de Gestão Empresarial (SIGE), ou como também são conhecidos, *Enterprise Resource Planning* (ERP). Este tipo de ferramentas auxiliam as instituições na integração e sincronização de processos isolados, permitindo diminuir de forma drástica a complexidade dos mesmos.

Neste contexto, muitas organizações têm optado pela utilização de produtos comerciais para a implementação dos seus principais processos de negócio. Não sendo a Universidade de Coimbra

exceção, também ela adotou uma ferramenta de código proprietário, adotando como *ERP* a ferramenta *SAP*, complementada com uma solução cuja manutenção é atualmente assegurada *in house* - o *LUGUS* - utilizando a metodologia *Agile* da plataforma *OutSystems*. Esta solução integra com *SAP* e que é uma valiosa ferramenta de apoio à desmaterialização dos processos organizacionais desta instituição, sendo por isso utilizada diariamente pelos colaboradores desta instituição nas mais diversas situações.

No entanto, essa aproximação às soluções comerciais não está ao alcance de muitas organizações em resultado dos custos de licenciamento e manutenção dos produtos utilizados. Com o amadurecimento de algumas soluções de *software* livre (*Open Source*) percebe-se que muitas empresas já encaram essas soluções como alternativa viável às soluções de código proprietário.

Também aqui, a Universidade de Coimbra, tem estado na vanguarda, adotando muitas vezes soluções de código aberto para suprir as suas necessidades. Assim sendo, no seguimento da política de contenção orçamental que tem adotado, e com o intuito de se libertar do peso das obrigações contratuais que tem para com a empresa proprietária da tecnologia *OutSystems*, a Universidade de Coimbra pretende explorar uma solução de código aberto, no sentido de averiguar se esta é uma boa alternativa e pode ser utilizada como ferramenta de desenvolvimento rápido para implementação dos seus processos organizacionais.

## 1.2. Objetivos

Tendo sido identificada pela Universidade de Coimbra uma plataforma de código aberto, que tem tido um rápido crescimento e uma ampla aceitação no mercado, o objetivo deste estágio consiste na avaliação da viabilidade da utilização dessa plataforma para a implementação dos processos organizacionais da instituição, e a sua integração com sistemas aplicativos existentes, em alternativa à solução proprietária atualmente utilizada.

Este estudo permitirá avaliar se a solução identificada pela Universidade de Coimbra pode ser usada como complemento ou até como substituta da plataforma *OutSystems* levando, em qualquer um dos casos, a uma previsível redução de custos.

Neste âmbito, os trabalhos a desenvolver neste estágio compreendem uma avaliação da solução selecionada para o efeito. Para tal, deverão ser identificados os processos institucionais mais representativos e proceder à sua implementação, passando por todas as suas fases de desenvolvimento, desde o estudo, especificação, implementação e testes.

O estudo da plataforma deverá passar por diversas fases, começando na exploração e familiarização com a solução de código aberto e terminando nas conclusões finais acerca da viabilidade ou não da

sua utilização em substituição da plataforma proprietária. Assim sendo são objetivos deste estágio os seguintes pontos:

- Familiarização do estagiário com a plataforma
- Identificação e análise de processos identificativos
- Validação e adaptação da especificação desses processos
- Implementação e testes
- Análise comparativa das implementações
  - Análise funcional
  - Análise de desempenho
  - Análise esforço de desenvolvimento
- Conclusões
- Produção de documentação

### **1.3. Contribuições**

Com vista ao cumprimento dos objetivos traçados, foram obtidas do estagiário as contribuições que se enumeram de seguida.

Foi feito o estudo do estado da arte, sendo para tal lida diversa documentação sobre a plataforma OutSystems, nomeadamente sobre a sua arquitetura, tecnologias utilizadas, performance, escalabilidade e mecanismos de segurança. Foi ainda consultada literatura sobre conceitos como *software* de código aberto e estimativas do custo total de propriedade para este tipo de software.

Foi feito um estudo da plataforma OpenERP, tendo sido lida documentação sobre esta plataforma, relativamente à sua arquitetura, tecnologias utilizadas e mecanismos de segurança disponibilizados, foi efetuado um estudo comparativo entre as principais características entre esta plataforma e o *OutSystems*.

Procedeu à instalação e configuração de um ambiente de testes e produziu documentação sobre esse processo. O estagiário frequentou ainda uma formação teórico-prática e fez o acompanhamento de entrada em produção de um módulo de PoS para a Loja da UC. Durante este período participou em diversas reuniões de especificação e validação com o cliente. Participou na configuração do módulo de *PoS* de acordo com as necessidades específicas da Loja e produziu documentação relativa a estes procedimentos de configuração. Produziu ainda documentação de apoio a estes processos.

No segundo semestre o estagiário criou um módulo OpenERP com vista à implementação de dois processos organizacionais da UC. Neste processo avaliou a forma como estes são implementados na plataforma OutSystems, procedendo à análise e especificação dos mesmos. Implementou apenas

um destes processos, apoiando-se na consulta de documentação de apoio ao desenvolvimento em OpenERP. Elaborou e executou um plano de testes para validar o bom funcionamento da implementação. Procedeu à comparação entre a solução implementada e o OutSystems efectuando uma análise crítica sobre o processo de implementação em OpenERP. Produziu ainda documentação sobre o processo de implementação de um módulo OpenERP.

## **1.4. Plano de estágio**

O plano de estágio divide-se pelos dois semestres de forma distinta, sendo que no primeiro semestre é esperado do estagiário um estudo teórico da plataforma selecionada e uma familiarização prática com o ambiente de desenvolvimento da mesma, sendo integrado num projeto já existente, com início anterior ao início do estágio. No segundo semestre é esperado o planeamento e desenvolvimento de um módulo que implemente dois processos organizacionais da instituição e uma análise crítica dos resultados obtidos.

### **1.4.1. Planeamento de tarefas para o primeiro semestre**

Nesta secção do documento são apresentados as diversas tarefas esperadas para o primeiro semestre do estágio e o respetivo cronograma.

#### **Estudo do Estado da Arte**

O estagiário deverá familiarizar-se com a plataforma OutSystems, estudando a sua arquitetura, tecnologias utilizadas e custos inerentes à sua utilização.

Também será feito um estudo introdutório ao conceito de custo total de propriedade, permitindo assim ter uma noção do tipo de custos que podem estar associados à adoção da utilização de uma plataforma de *software* de código aberto. Devendo finalmente ser identificados quais os custos a ter em conta em caso de migração para a nova solução.

#### **Formação Teórico-prática**

O estagiário irá participar numa formação teórico-prática em *OpenERP*, contratada pela *GSIIC-DSI*, para os seus colaboradores. Essa formação é efetuada por uma empresa parceira oficial *OpenERP* em Portugal, nas instalações da instituição acolhedora, e tem a duração de uma semana.

#### **Instalação e configuração de ambiente de testes**

Além da instalação da plataforma, deverão também ser instalados os módulos adicionais necessários ao seu bom funcionamento. Deverá ainda ser produzida documentação relativa aos processos de instalação do *OpenERP*.

## Familiarização com *OpenERP*

O estagiário deverá familiarizar-se com a plataforma, e fazer um estudo teórico da nova plataforma, devendo estudar, entre outros aspeto a sua arquitetura; os mecanismos de segurança e auditoria que a mesma apresenta e possibilidade de integração com sistemas de autenticação existentes na instituição.

## Exploração do *OpenERP* em contexto de trabalho

Está previsto para a primeiro semestre a exploração do OpenERP em contexto real de trabalho. Nesse sentido, o estagiário deverá ter uma participação ativa no acompanhamento de entrada em produção de um módulo da plataforma.

### 1.4.1.1. Cronograma

As tarefas descritas acima deverão ser executadas de acordo com o seguinte cronograma

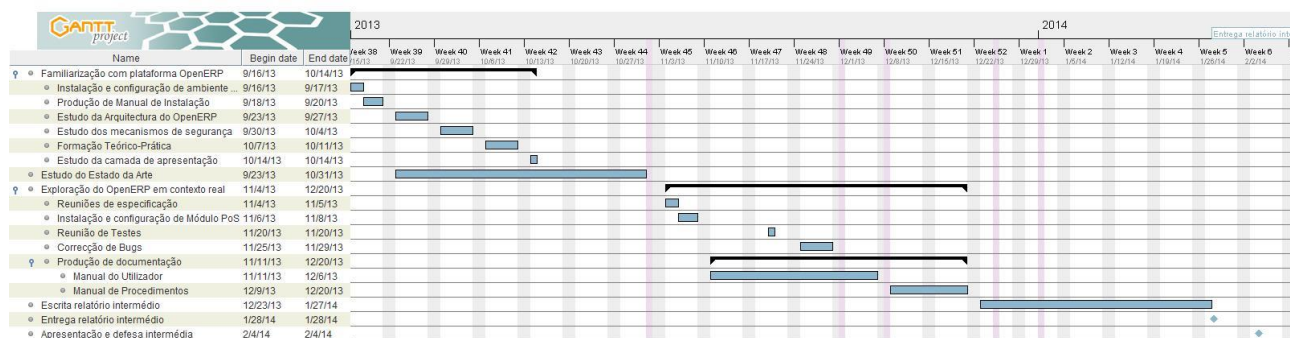


Figura 1 - Cronograma de tarefas para o primeiro semestre

O cronograma mais detalhado pode ser consultado no Anexo I deste documento.

## 1.4.2. Planeamento de tarefas para o segundo semestre

Nesta secção do documento são apresentados as diversas tarefas esperadas para o segundo semestre do estágio e o respetivo cronograma.

### Análise do motor de *workflow* do *OpenERP*

Antes de iniciar a implementação dos processos deverá ser feito um estudo e análise do motor de *workflow* do *OpenERP*. Deverão ser analisadas as ferramentas que a plataforma fornece, e quais as melhores abordagens para a sua construção.

### Análise de requisitos transversais do *workflow* do *LUGUS*

Também o modelo de *workflow* do LUGUS deverá ser objeto de análise rigorosa, por forma a entender todos os seus mecanismos, possibilitando a replicação dos processos no *OpenERP*. Deverão

ser analisados os modelos e permissões existentes no LUGUS e o modo como são feitas as interações.

### **Implementação dos processos**

Para cada processo a implementar deverá ser efetuada uma análise, durante a qual poderão ser realizadas diversas reuniões entre o estagiário e um elemento, destacado para o efeito, da equipa do projeto *LUGUS*. Estas reuniões servirão para familiarizar o estagiário com o processo em análise.

Numa fase posterior será efetuada pelo estagiário a validação e adaptação da especificação. Durante esta fase deverá ser produzida documentação de especificação do processo, nomeadamente relativa ao seu *workflow* e modelo de permissões. Deverão ser identificados quais os intervenientes e que ações lhes competem em cada etapa do processo.

Seguir-se-á uma fase de implementação propriamente dita, de acordo com as especificações elaboradas nas fases anteriores.

Após o que se seguirá uma fase de testes. Nesta fase dos trabalhos, o estagiário deverá elaborar um plano de testes detalhado, por forma a garantir que o maior número de cenários possíveis seja coberto. Neste plano deverão constar as ações a efetuar pelo utilizador e uma descrição detalhada dos resultados esperados, de acordo com a especificação. Deverão depois ser efetuados os testes, de acordo com o plano elaborado, e deverá ser feito um registo do seu resultado.

### **Análise comparativa das implementações**

Nesta fase do estágio será feita uma análise comparativa entre os resultados obtidos no *OpenERP* e do *OutSystems*. Esta análise será constituída por três tipos de análise: funcional; de desempenho e esforço de desenvolvimento.

Na análise funcional deverão ser realçados os pontos onde uma ou outra solução se destaque, quer pela positiva, quer pela negativa.

Na análise de desempenho deverá analisar-se o desempenho de ambas as soluções. Não se deverá dar enfoque apenas no desempenho – a nível de tempo de execução (até porque as soluções estão instaladas em ambientes díspares) – mas sobretudo ao nível da produtividade dos utilizadores, no que diz respeito à usabilidade da solução desenvolvida.

Na análise do esforço de desenvolvimento deverá fazer-se uma comparação entre o esforço necessário para a implementação de um processo no *OpenERP* e no *OutSystems*, por forma a comparar a rentabilidade da utilização desta nova plataforma.

#### **1.4.2.1. Cronograma**

As tarefas identificadas serão executadas de acordo com o seguinte cronograma.

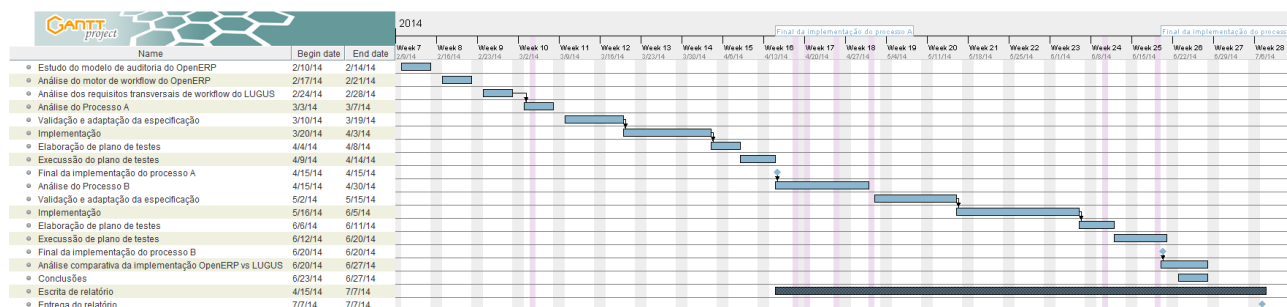


Figura 2 - Cronograma de tarefas para o segundo semestre

O cronograma mais detalhado pode ser consultado no Anexo I deste documento.

## 1.5. Estrutura do documento

Este relatório é composto por 7 capítulos onde é descrito o trabalho realizado durante este estágio, e ainda por um oitavo capítulo onde consta a bibliografia consultada e um nono capítulo, contendo diversos anexos, com informação adicional.

No primeiro capítulo deste relatório é feito um enquadramento geral do projeto, identificam-se os objetivos deste estágio e qual o seu planeamento.

O segundo capítulo é dedicado ao estado da arte. Nele é apresentada a plataforma *OutSystems* e quais as suas principais características. É ainda feita uma abordagem teórica ao conceito de custo total de propriedade, sendo identificados alguns custos que poderão estar envolvidos na adoção de uma nova solução.

No terceiro capítulo é feita uma abordagem teórica ao *OpenERP*, nomeadamente no que se refere à sua arquitetura e ferramentas disponibilizadas por esta plataforma.

No quarto constam as atividades desenvolvidas no primeiro semestre que permitiram ao estagiário a utilização de algumas funcionalidades da ferramenta, no contexto da entrada em produção de um módulo de Ponto de Venda.

O quinto capítulo documenta todo o processo de implementação e justifica algumas opções tomadas durante essa fase.

No sexto capítulo é feita uma análise comparativa entre a solução existente e a solução desenvolvida.

E, finalmente, no sétimo capítulo são apresentadas as conclusões retiradas da utilização do *OpenERP*, e trabalho futuro a ser desenvolvido sobre esta plataforma.



## 2. Estado da arte

Neste estado da arte será apresentada a solução atualmente em utilização pela Universidade de Coimbra, e os custos inerentes à sua utilização. Também irá ser apresentado uma introdução ao estudo do conceito de custo total de propriedade, permitindo assim ter uma noção dos custos associados à adoção de uma solução de *Software Open Source (Open Source Software – OSS)*. Com base nos conceitos definidos de custo total de propriedade, irá ser feita identificação dos custos a ter em conta no caso específico da adoção do OpenERP por parte da Universidade de Coimbra.

### 2.1. Apresentação da plataforma *OutSystems*

Aquando da reestruturação dos diversos serviços da Universidade de Coimbra, houve a necessidade de normalizar e centralizar o tratamento de diverso tipo de procedimentos que antes desta reestruturação eram implementados de forma mais ou menos independente por cada unidade orgânica desta instituição.

Nessa altura a Universidade de Coimbra estudou as melhores soluções informáticas que permitissem auxiliar à desmaterialização desses procedimentos e, nesse sentido, acabou por adotar a plataforma *OutSystems* dando então início ao projeto *LUGUS*. Esta plataforma utiliza tecnologia proprietária e é utilizada intensivamente por várias dezenas dos seus colaboradores, diariamente.

#### Arquitetura

A solução apresentada pela *OutSystems* é constituída por duas plataformas, uma de *BackOffice*, onde é feito todo o desenvolvimento e outra de *FrontOffice*, onde é instalada a aplicação desenvolvida no *BackOffice* e que é acedida pelos utilizadores finais.

Na plataforma de *BackOffice* é feita a gestão da aplicação, e é onde são efetuadas as configurações visuais da mesma. É esta plataforma que gera automaticamente o código, faz a verificação de erros de implementação e deteta eventuais inconsistências. E que, finalmente, disponibiliza e instala o código gerada no(s) servidor(es) de *FrontOffice*.

A plataforma de *FrontOffice* é onde é instalado a aplicação desenvolvida no *BackOffice* e que será acedido pelos utilizadores finais da mesma. Todas as funcionalidades de segurança, auditoria, *logging* e monitorização correm na plataforma de *FrontOffice*.

De seguida serão apresentadas duas características diferenciadoras desta plataforma.

#### TrueChange

A plataforma de *BackOffice* do *OutSystems* gera código *java* ou *aspx* otimizado e totalmente

documentado. Antes de gerar o código é feita uma verificação às alterações visuais efetuadas pelo utilizador, recorrendo à tecnologia *TrueChange*. Esta tecnologia, proprietária da *OutSystems*, permite detetar em tempo real inconsistências no código e potenciais implementações que venham a levar a problemas de escalabilidade ou performance, fornecendo ao utilizador sugestões para corrigir os erros detetados. Os problemas detetados pelo motor *TrueChange* são do mais variado tipo e englobam por exemplo:

- Grandes quantidades de informação armazenados em memória;
- Criação de base de dados com estruturas ineficientes;
- Acessos à base de dados feitos de forma ineficiente;
- Lógica implementada para processar mais quantidade de dados do que aqueles que são realmente necessários para efetuar determinada ação.

Desta forma, além de garantir que o código gerado não contém erros, é também garantido que este não irá levar a problemas de performance da aplicação.

### **1-Click-Publish**

O processo de entrada em produção do produto desenvolvido é muito simples, graças à tecnologia *1-Click-Publish*.

Quando o programador clica no botão *1-Click-Publish*, a plataforma

1. Valida o modelo recorrendo ao motor *TrueChange*;
2. Cria uma nova versão da aplicação e guarda-a no repositório;
3. O compilador *OutSystems* analisa o novo modelo e todas as suas dependências e gera código *java* ou *aspx*;
4. Imediatamente, a aplicação gerada, juntamente com todas as suas dependências, é instalada no(s) servidor(es) de *FrontOffice*, sendo feitas todas as alterações necessárias à base de dados de suporte à aplicação.

#### **2.1.1. Custos inerentes à utilização da plataforma *OutSystems***

Como já várias vezes foi afirmado, o *OutSystems* é uma plataforma proprietária e tem por isso mesmo custos de licenciamento. A política de habitual de licenciamento do *OutSystems* baseia-se numa licença de utilização anual cujo custo proporcional à utilização de *Software Units*. Esta unidade de medida está relacionada com a complexidade do código desenvolvido. O número de *Software Units* utilizado numa determinada aplicação é calculado tendo em conta, o número de páginas apresentadas, formulários, tabelas de base de dados, acessos a essas tabelas, etc.

No entanto o contrato de licenciamento vigente entre a Universidade de Coimbra e a *OutSystems* é distinto deste tipo de contrato, não havendo lugar a pagamento anual. A Universidade de Coimbra adquiriu uma licença que lhe garante, além do direito à utilização da plataforma e acesso a *upgrades* e *patches* de segurança que venham a ser lançados pela *OutSystems*. O direito de utilização contratualizado prevê um valor máximo de 600 000 *Software Units* e 52 transações concorrentes. Sendo ultrapassado um destes limites, a licença de utilização da plataforma pode ser invalidada o que, a acontecer, poderá impossibilitar a Universidade de Coimbra de proceder ao desenvolvimento novas funcionalidades.

As soluções desenvolvidas pela Universidade de Coimbra até ao momento levaram ao consumo da quase totalidade das *Software Units* contratadas. É por isso urgente encontrar solução para esta situação, pois uma eventual renegociação da licença de utilização levará a um aumento dos custos associados à utilização desta solução.

## 2.2. Custos associados ao *software Open Source*

Muitas vezes se chama, erradamente, "*Software Gratuito*" ao *Software Open Source* (*Open Source Software – OSS*). Esse erro comum poderá vir de uma tradução livre do termo inglês "*Free Software*". Neste caso, o termo "*free*" não se refere à ausência de pagamento, mas sim à liberdade que este tipo de *software* oferece. Na sua definição de *software* livre Richard M. Stallman escreveu o seguinte: "*Free software*" is a matter of liberty, not price. To understand the concept, you should think of "*free*" as in "*free speech*," not as in "*free beer*." [1]

Ainda de acordo com o mesmo autor, o princípio do *software* livre é uma questão de liberdade dos utilizadores em executar, copiar, distribuir, estudar, mudar e melhorar o *software*. Mais concretamente refere-se a quatro tipos de liberdades para os utilizadores [1]:

- Liberdade 0: A liberdade de executar o *software*, para qualquer propósito.
- Liberdade 1: A liberdade de estudar o seu funcionamento, e adaptá-lo às suas necessidades. (É por isso um pré-requisito o acesso ao código fonte)
- Liberdade 2: A liberdade de redistribuir cópias de modo a ajudar o próximo.
- Liberdade 3: A liberdade de melhorar o programa, e partilhar os seus melhoramentos com o público, de modo a que toda a comunidade possa deles beneficiar.

Com base nestas liberdades pode concluir-se que o modelo de custos no *OSS* é muito distinto do modelo proprietário, uma vez que todo o desenvolvimento é feito com base em mecanismos de partilha de conhecimento e de programação colaborativa. Os custos acabam por ser diminuídos porque, muitas vezes, um programa novo pode surgir da reutilização de código já existente. Também

a distribuição tem um custo menor, uma vez que é feita através da internet e com publicidade gratuita, feita pelos próprios utilizadores, nos fóruns de discussão da especialidade [2]. No entanto os custos existem e não devem ser menosprezados quando é chegada a hora de tomar a decisão de migrar de uma solução proprietária para uma solução *OSS*. Sejam estes custos diretos, advindos da utilização do *software* - como custos de licenciamento ou custos de atualização - ou sejam custos indiretos, como custos de formação dos utilizadores ou, até, do estudo das diversas ofertas e até à seleção do produto.

### 2.2.1. Definição de Custo Total de Propriedade

Tendo isto em conta poderemos tentar definir o custo total de propriedade (*Total Cost of Ownership* – *TCO*) para os *OSS*. Segundo Richard West e Stephen Daigle o *TCO* de um *ERP* deve ter em conta os 5 estágios do seu ciclo de vida: Aquisição, Implementação, Operação, Manutenção e Substituição [3].

Sendo alguns destes custos mais fáceis de identificar do que outros, alguns autores chegam até a comparar o *TCO* a um iceberg, sendo os custos mais difíceis de identificar muitas vezes bem maiores do que aqueles que mais facilmente são identificáveis [4].

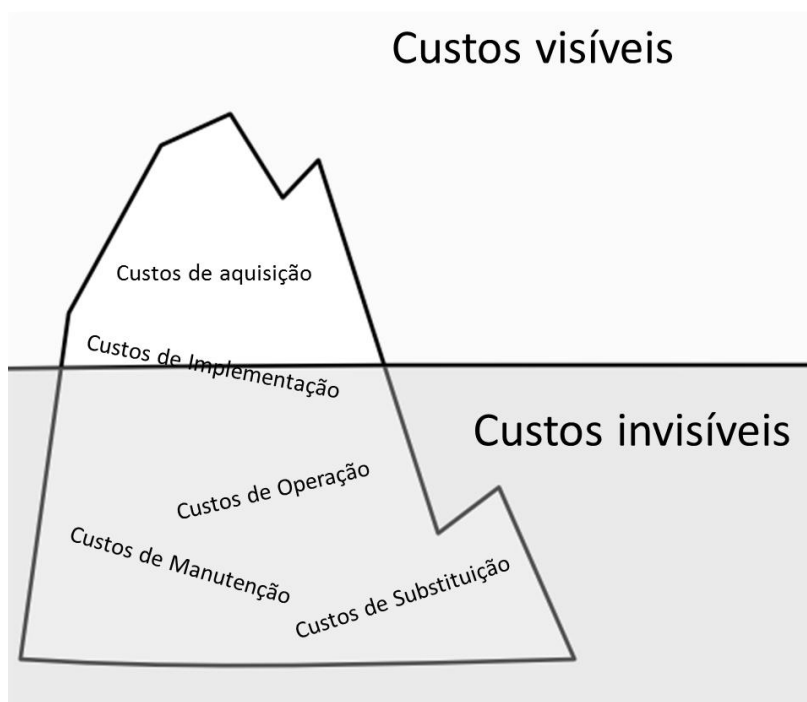


Figura 3 - Custo total de Propriedade - Custos visíveis vs custos invisíveis

Num relatório elaborado para o *Cabinet Office* do Reino Unido, Maha Shaikh e Tony Cornford

concluíram que, se os custos de licenciamento, manutenção, atualizações e contratos de suporte eram os custos mais facilmente identificados pelos decisores, os custos relacionados com a necessidade de adaptação organizacional e custos de retirada de *software* eram os que mais dificilmente eram identificados – e por isso de ter em conta quando era altura de escolher um *software* [5].

Ainda segundo estes autores as principais componentes do custo total de propriedade são as seguintes:

| <b>Categoria do custo</b> |  |
|---------------------------|--|
| <b>Pesquisa</b>           | Custo do estudo e avaliação da solução   |
|                           | Custo do estudo e previsão de implementação  |
| <b>Aquisição</b>          | Custo de licenciamento do <i>software</i>  |
|                           | Custo de parametrização de acordo com as necessidades do negócio                               |
|                           | Custo de integração com a plataforma atual   |
| <b>Integração</b>         | Custo de migração (tanto dados como utilizadores)  |
|                           | Custo com formação   |
|                           | Custo de adaptação aos novos processos e boas práticas   |
| <b>Utilização</b>         | Custo do suporte do serviço – <i>in house</i>  |
|                           | Custo do suporte do serviço – contratações externas  |
|                           | Custo de manutenção e <i>upgrades</i>  |
|                           | Custo de necessidades de escalar (eventual aumento do número de utilizadores ou de transações) |
| <b>Retirada</b>           | Custos de retirada (relativo ao hardware e software)   |
|                           | Custos de retirada (relativo à mudança e necessidade de novas formações)                       |

*Tabela 1 - Componentes do cálculo do TCO*

*Adaptado de "Total cost of ownership of open source software: a report for the UK Cabinet Office supported by OpenForum Europe"*

### **2.3. Custos inerentes à utilização do *OpenERP***

Tendo em conta o que foi dito na secção anterior deste documento, iremos tentar identificar as parcelas a ter em conta para o cálculo dos custos inerentes à utilização do *OpenERP* por parte da Universidade de Coimbra.

- Pesquisa
  - Não deverá ser de menosprezado o custo associado ao decorrer deste estágio, e menos ainda os custos associados ao tempo despendido antes do início do mesmo

para o estudo e seleção da plataforma a ser aqui utilizada;

- **Aquisição**
  - Se, no caso do OpenERP, o custo de licenciamento da plataforma é nulo, não serão de menosprezar as restantes componentes enumeradas em *Tabela 1 - Componentes do cálculo do TCO*, nomeadamente:
    - Custo com a parametrização da plataforma (nomeadamente os custos associados à formação dos colaboradores da instituição responsáveis pela *parametrização in house*);
    - Custo com a integração com a plataforma atualmente utilizada e com os restantes serviços em utilização na instituição;
- **Integração**
  - Não será necessário ter em conta os custos para migração de dados uma vez que estes deverão manter-se disponíveis na plataforma atual, não sendo migrados para o *OpenERP*;
  - Deverá ser tido em conta no entanto o custo da migração dos utilizadores, sobretudo na vertente de configuração de privilégios e permissões diferenciadas, utilizador a utilizador;
- **Utilização**
  - Nesta vertente, deverão ser tidos em conta os custos associados com a manutenção e contínuo melhoramento da plataforma por parte dos colaboradores da instituição afetos ao projeto *OpenERP*;
  - Também os custos de suporte contratados externamente para assegurar o bom funcionamento deverá ser tido em conta;
- **Retirada**
  - As boas práticas dizem que estes custos devem ser tidos também em conta, nomeadamente no que toca custos de migração para outras soluções quando estas foram abandonados.

## 2.4. Custo do *OutSystems* vs custo do *OpenERP*

Nesta secção irá ser apresentada uma comparação entre os custos associados à utilização do *OutSystems* vs custos associados à adoção do *OpenERP*.

| Categoria do custo |   | OutSystems | OpenERP     |
|--------------------|---|------------|-------------|
| Pesquisa           | Custo do estudo e avaliação da solução      | a)         | Relevante + |
|                    | Custo do estudo e previsão de implementação | a)         | Relevante + |

|                   |  |                    |                   |
|-------------------|--|--------------------|-------------------|
| <b>Aquisição</b>  | Custo de licenciamento do <i>software</i>  | b)                 | Nulo              |
|                   | Custo de parametrização de acordo com as necessidades do negócio                               | Relevante c)       | Relevante<br>++++ |
|                   | Custo de integração com a plataforma atual   | N/A                | Relevante         |
| <b>Integração</b> | Custo de migração (tanto dados como utilizadores)  | N/A                | Relevante d) +    |
|                   | Custo com formação   | N/A                | Relevante ++      |
|                   | Custo de adaptação aos novos processos e boas práticas   | N/A                | Relevante ++      |
| <b>Utilização</b> | Custo do suporte do serviço – <i>in house</i>  | Relevante          | Relevante +       |
|                   | Custo do suporte do serviço – contratações externas  | Relevante          | Relevante ++      |
|                   | Custo de manutenção e <i>upgrades</i>  | Relevante          | Relevante         |
|                   | Custo de necessidades de escalar (eventual aumento do número de utilizadores ou de transações) | Relevante          | Relevante         |
| <b>Retirada</b>   | Custos de retirada (relativo ao <i>hardware</i> e <i>software</i> )                            | Relevante e)<br>++ |                   |
|                   | Custos de retirada (relativo à mudança e necessidade de novas formações)                       | Relevante e)<br>++ |                   |

Tabela 2 - Análise comparativa do custo de substituição do *OutSystems* pelo *OpenERP*

- a) Não há neste momento custos associados à pesquisa para o *OutSystems* uma vez que esta é a solução atual.
- b) O investimento em licenciamento do *OutSystems* já foi efetuado pelo que não será relevante nesta comparação.
- c) Apesar do *OutSystems* ser já uma plataforma amadurecida e inteiramente parametrizada, de acordo com as necessidades da instituição, a solução mantém-se em melhoramento contínuo pela equipa do projeto *LUGUS*.
- d) Apesar de o custo com a migração dos utilizadores não ser considerado relevante, o custo com a configuração dos acessos e permissões desses utilizadores deverá ser considerado.
- e) O custo de retirada do *LUGUS* deverá ser tido em conta e deverá ser igual ou superior ao custo de pesquisa, aquisição e integração do *OpenERP*.

Na tabela acima foi adotada uma notação relativamente ao peso que determinado custo considerado relevante deve ter no cálculo do *TCO*. Devendo esse peso ser tanto maior quanto maior o número de + associadas à componente do custo.

## 2.5. Conclusões

Verifica-se que, apesar de a exploração de uma nova solução aparentar um investimento de alguma monta por parte da Universidade de Coimbra, manter a solução atual implicará também gastos significativos por parte desta instituição. Estes gastos poderão ser ainda mais elevados se, com a renegociação do contrato de licenciamento com a OutSystems, esta passar a aplicar o custo de licenciamento anual, como tem vindo a praticar com todos os seus clientes.

Se tivermos em conta que os custos de utilização do OpenERP (nomeadamente com suporte *in house*) podem ser amortizados com a migração dos recursos afetos ao *OutSystems* para o *OpenERP*, poderá equilibrar-se ou até anular o custo com a formação e parametrização do *OpenERP* com o custo de uma nova licença de utilização do *OutSystems*.

Desta forma, e tendo em conta que o preço mais baixo do licenciamento da plataforma *OutSystems* é de € 15,000.00 por ano [6], conclui-se que é razoável assumir nesta fase que haverá menos custos numa aposta no *OpenERP* como solução complementar ao *LUGUS*, passando esta plataforma a assumir alguns dos processos organizacionais e mantendo-se esta plataforma em funcionamento com as atuais funcionalidades operacionais ou até migrando alguns dos processos desta plataforma para o *OpenERP*.



## **3. *OpenERP*: abordagem teórica**

Neste capítulo será feita uma apresentação sobretudo teórica do *OpenERP*, nomeadamente no que se refere à sua arquitetura, mecanismos de segurança e auditoria, e por fim será feita uma comparação entre as principais categorias entre esta solução e o *OutSystems*.

### **3.1. Formação teórico-prática**

O início do período de estágio coincidiu com a realização de uma formação teórico-prática, contratada pela SGSIIC-DIS, no sentido de munir os seus colaboradores com dos conhecimentos necessários para a utilização do *OpenERP*, tendo o estagiário também estado nela presente.

Nessa formação foi feita uma introdução teórica aos procedimentos necessários para criação de novos módulos no *OpenERP* ou alteração de módulos existentes. Foi explicada a diferença entre os diversos tipos de vista e foi ainda feita uma introdução ao modelo de permissões, edição de *workflow* e produção de relatórios. Durante o período de formação foram executados exercícios práticos com o intuito de por em prática os conhecimentos adquiridos durante as introduções teóricas.

Esta formação foi de grande utilidade uma vez que permitiu, de uma forma abrangente, dar a conhecer o funcionamento do *OpenERP* e os mecanismos envolvidos na implementação de novas funcionalidades, munindo desta forma o estagiário dos conhecimentos básicos que lhe permitirão proceder às tarefas de implementação dos processos, agendados para o segundo semestre deste estágio.

### **3.2. Arquitetura**

O *OpenERP* tem uma arquitetura "*three-tier*" e baseia-se no paradigma "*Cliente-Servidor*" obedecendo ao padrão "*Model-View-Controller*".

Esta arquitetura torna independentes as camadas de dados, de lógica e de apresentação, podendo assim qualquer uma delas sofrer alterações não comprometendo obrigatoriamente o funcionamento das outras.

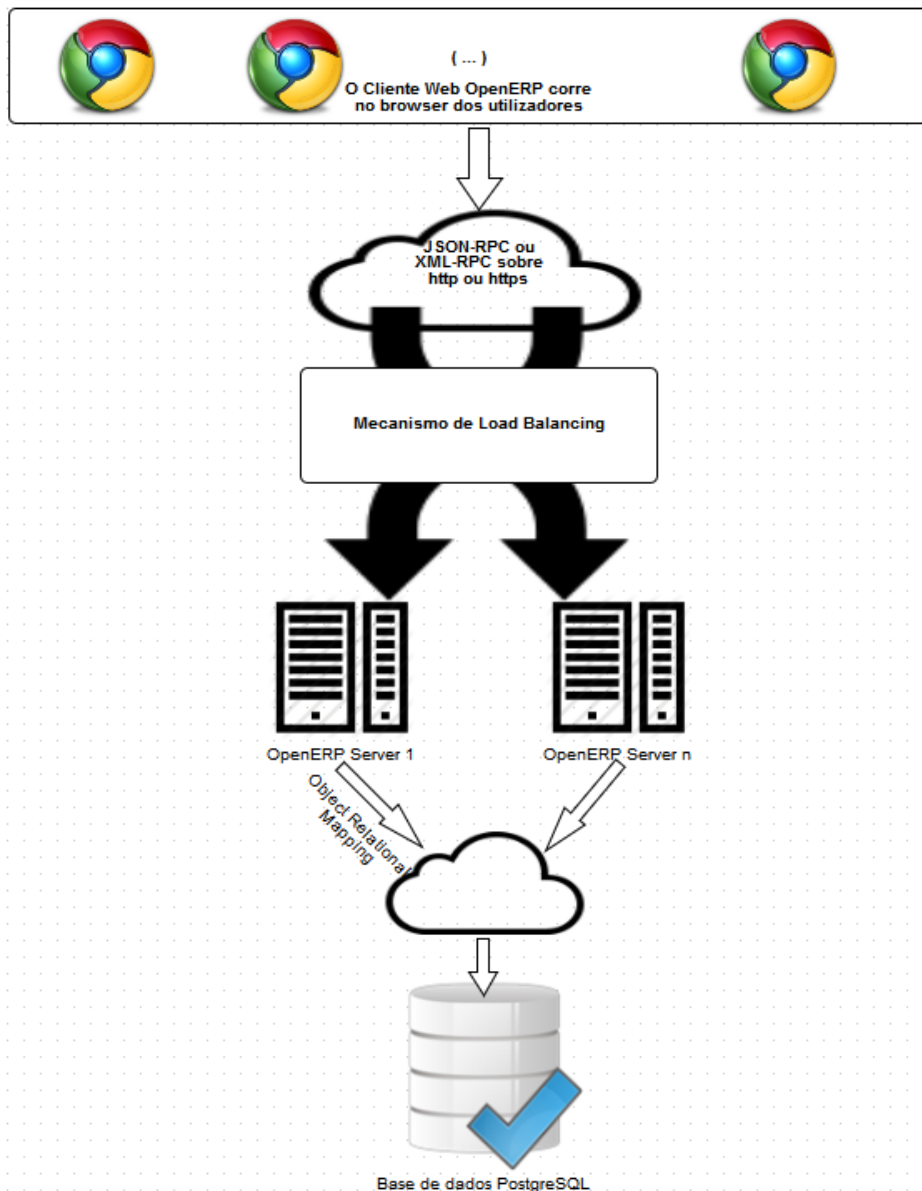


Figura 4 - Arquitetura do OpenERP

(Imagem gerada online em <http://draw.io> adaptado de [7])

Na imagem acima podemos ver claramente a separação entre as três camadas. Estas camadas podem correr no mesmo servidor, ou podem ser distribuídas por diversas máquinas, se houver requisitos de performance que assim o exijam.

Na camada de dados temos uma base de dados *PostgreSQL*, contendo todos os dados da aplicação e as suas principais configurações.

A camada de lógica é formada pelo núcleo original da aplicação, ao qual podem ser posteriormente acrescentados vários módulos (note-se que existem módulos chamados oficiais, que são desenvolvidos pela OpenERP S.A. e módulos de comunidade, sobre os quais a OpenERP S.A. não

toma qualquer responsabilidade), podendo desta forma ser parametrizado, de acordo com as necessidades específicas de cada instituição. Esta camada assegura ainda a comunicação com a camada de dados e a camada de apresentação.

A camada de apresentação é constituída pelo cliente-web, que corre como uma aplicação *JavaScript*, no *browser* dos utilizadores, e que comunica com o servidor usando o protocolo *JSON-RPC* sobre *HTTP* ou *HTTPS*.

Apesar de serem facilmente implementáveis, a utilização e suporte para clientes *ad-hoc* foi abandonada a partir da versão 7.0 do *OpenERP*. A vantagem destes clientes em relação ao cliente web era sua melhor performance quando estes se encontravam na mesma rede que o servidor. No entanto, face à comodidade de acesso a partir de qualquer ponto do globo ou a partir de qualquer computador ou terminal móvel, e à não necessidade de instalação (o utilizador pode aceder ao servidor utilizando o seu *browser* preferido!) e praticamente não necessidade de manutenção (as atualizações do *browser* são feitas pelo próprio utilizador) levou ao abandono do suporte dos clientes *GTK*, muito utilizados até à versão 6.x do *OpenERP*.

### 3.2.1. Padrão *Model-View-Controller*

Quando uma aplicação informática atinge um determinado grau de complexidade e tem necessidade de apresentar grande quantidade de dados ao utilizador, torna-se relevante a separação entre os dados e a apresentação dos mesmos. Desta forma, alterações aplicadas na interface gráfica não têm impacto sobre a manipulação de dados, e o mesmo se aplica no sentido inverso, podendo processar-se a uma reorganização dos dados sem necessidade de alterar o *layout*.

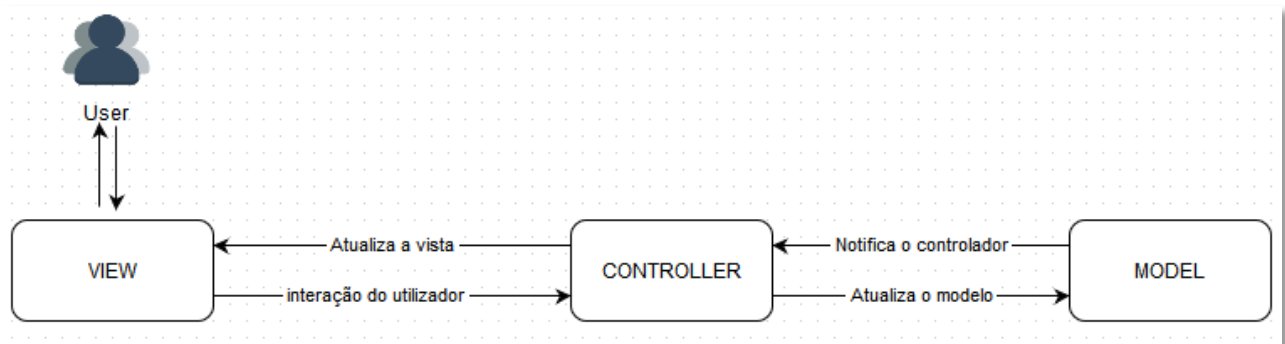


Figura 5 - Padrão MVC

(imagem gerada online em <http://draw.io/>)

O padrão *Model-View-Controller* resolve este problema através da separação das tarefas de acesso aos dados e lógica de negócio da lógica de apresentação e de interação com o utilizador, introduzindo um componente entre os dois: o controlador.

O *OpenERP* segue o padrão *MVC* da seguinte forma:

- *Model*: Tabelas da base de dados *PostgreSQL*
- *View*: Definidas nos ficheiros *XML* do *OpenERP*
- *Controller*: Os objetos do *OpenERP*

### 3.2.1.1. *Object Relational Mapping*

A camada de *ORM* (*Object Relational Mapping*) presente no servidor *OpenERP* é uma das suas principais características. Fazendo uso desta funcionalidade, os modelos de dados são descritos no código em *Python* e o *OpenERP* cria as tabelas na base de dados com base nessas descrições. Desta forma conseguem obter-se todos os benefícios de uma base de dados relacional somada à flexibilidade do *Python*, que permite acrescentar qualquer tipo de restrição adicional ao modelo.

Esta tecnologia permite a criação de interfaces sobre os objetos que permitem desenhar e implementar módulos eficientes. É necessário ter em conta a importância do *ORM* antes de tentar contorná-lo acedendo diretamente à base de dados através *queries SQL*. Só quando faz uso do *ORM* o *OpenERP* sabe que os dados acedidos são consistentes.

O *ORM* oferece vários métodos que permitem acessos à base de dados, tanto de escrita como de leitura. Os métodos disponíveis são os seguintes:

- *browse(cr, uid, ids, context=None)*  
Pesquisa e devolve os resultados como objectos. Permite usar pontos para aceder a dados armazenados em outras tabelas relacionadas (e.g. *owner\_id.name*);
- *check\_access\_rule(cr, uid, ids, operation, context=None)*  
Verifica se o utilizador tem permissões para efectuar uma determinada operação;
- *copy(cr, uid, id, default=None, context=None)*  
Duplica os dados de um registo com um determinado id;
- *copy\_data(cr, uid, id, default=None, context=None)*  
Semelhante ao anterior, mas apenas copia os dados, não criando nenhuma cópia;
- *create(cr, uid, vals, context=None)*  
Cria um novo registo com os valores passados no dicionário "*vals*";
- *default\_get(cr, uid, fields\_list, context=None)*  
Devolve os resultados definidos por defeitos para os campos contidos na lista "*fields\_list*";

- *export\_data(cr, uid, ids, fields\_to\_export, context=None)*  
Exporta os dados contidos nos campos definidos na lista *"fields\_to\_export"* para os registos correspondentes à lista de ids fornecida;
- *fields\_get(cr, uid, fields=None, context=None)*  
Obtém a descrição dos campos (de apenas alguns campos se especificado no campo *"fields"*);
- *fields\_view\_get(cr, user, view\_id=None, view\_type='form', context=None, toolbar=False, submenu=False)*  
Obtém a composição detalhada dos campos para um determinado tipo de vista;
- *get\_xml\_id(cr, uid, ids, \*args, \*\*kwargs)*  
Tenta determinar o *xml\_id* de uma lista de registos se estes tiverem sido definidos;
- *import\_data(cr, uid, fields, datas, mode='init', current\_module="", nouupdate=False, context=None, filename=None)*  
Permite importar dados para um determinado módulo;
- *name\_get(cr, user, ids, context=None)*  
Devolve tuples com o nome do(s) objeto(s) pedido(s);
- *name\_search(cr, user, name="", args=None, operator='ilike', context=None, limit=100)*  
Pesquisa registos pelo nome, de acordo com o domínio definido no parâmetro *args*;
- *perm\_read(cr, user, ids, context=None, details=True)*  
Devolve uma lista de dicionários com alguns atributos dos registos, nomeadamente quando foi criado o registo e por quem, quando e por quem foi alterado pela última vez;
- *read\_group(cr, uid, domain, fields, groupby, offset=0, limit=None, context=None, orderby=False)*  
Devolve uma lista de registos, agrupado de acordo com o especificado no parâmetro *groupby*;
- *search(cr, user, args, offset=0, limit=None, order=None, context=None, count=False)*  
Pesquisa por registo de acordo com um determinado domínio – definido no parâmetro *args*;
- *unlink(cr, uid, ids, context=None)*

Remove os registos definidos no parâmetro *ids*,

- *write(cr, user, ids, vals, context=None)*

Atualiza os registos cujo id estão definidos no parâmetro *ids* com os valores definidos no dicionário *vals*.

### 3.2.2. Escalabilidade

Em termos de escalabilidade, verifica-se que o desempenho do *OpenERP* poderá estar limitado por um eventual menor desempenho da sua base de dados *PostgreSQL*, sobretudo se esta se mantiver com as configurações por defeito. Algumas operações de *tuning* da base de dados podem ser necessárias para aumentar a performance do *OpenERP*, nomeadamente aumentar o número de acessos concorrentes [8].

Em casos de utilização extrema, a camada de lógica pode ser distribuída por diversos servidores, com um sistema de *load balancing*. Sabe-se que esta foi a solução foi adotada pelo ministério da educação para fazer face à grande quantidade de acessos concorrentes na sua plataforma de contratação dos docentes, desenvolvida sobre *OpenERP* pela *ThinkOpen*, parceiro oficial *OpenERP* em Portugal. Segundo este integrador, esta é a maior instalação *OpenERP* do mundo, com mais de 200.000 utilizadores e com picos de utilização que chegam a atingir os 50.000 acessos simultâneos. Este grau de escalabilidade foi atingido recorrendo à plataforma de virtualização *Open Source Proxmox* que permite aumentar ou diminuir o número de servidores distribuídos sem qualquer *downtime*. Não existe no entanto documentação oficial a que o estagiário tenha tido acesso para melhor estudar esta solução.

### 3.3. Instalação de ambiente de desenvolvimento e testes

Para instalação de uma instância do *OpenERP* é requisito mínimo uma máquina *Ubuntu 10* ou superior, com pelo menos 4Gb de espaço livre em disco e 1 *Gb* de memória.

Deverá começar por alterar-se a lista de fontes de atualização do sistema operativo, acrescentando o repositório *nightly* do *OpenERP* para futuras atualizações automáticas. Será ainda necessário instalar diversos pacotes, nomeadamente de *Python* e o *PostgreSQL*, para o funcionamento da plataforma.

A lista completa de pacotes a instalar e restante processo de instalação pode ser consultado no Anexo II deste documento.

### 3.4. Instalação de módulos

O *OpenERP* é conhecido pelo grande número de módulos que disponibiliza e que permite instalar. Sendo o processo de instalação extremamente simples, o utilizador pode escolher os módulos que mais se adequam às suas necessidades, permitindo-lhe parametrizar o *ERP* de acordo com as suas preferências.

De seguida será explicado, de uma forma breve, como proceder para instalar um módulo *OpenERP*. Aproveitando um caso prático segue-se a explicação da instalação do módulo de auditoria.

O primeiro passo passa por ir às configurações da aplicação, remover eventuais filtros que estejam presentes na caixa de pesquisa e introduzir o nome do módulo a instalar. Serão então mostrados todos os resultados que satisfaçam a pesquisa. Deverá seleccionar o módulo que pretende instalar clicando e clicar em em "Install".

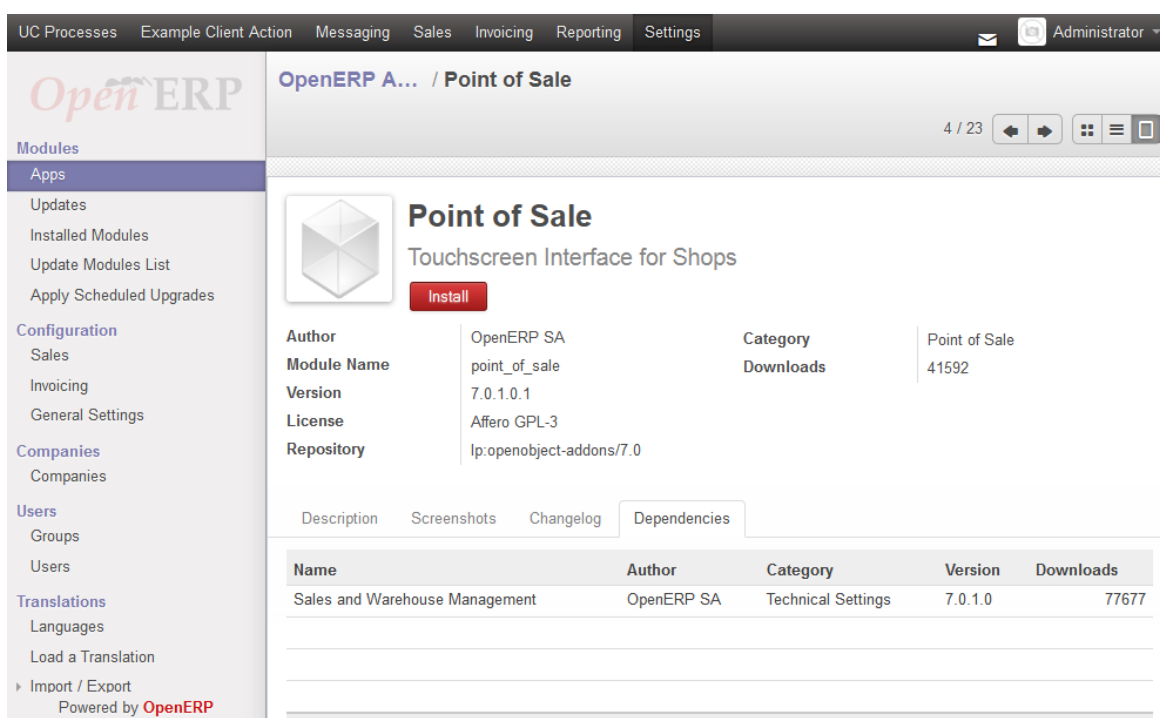


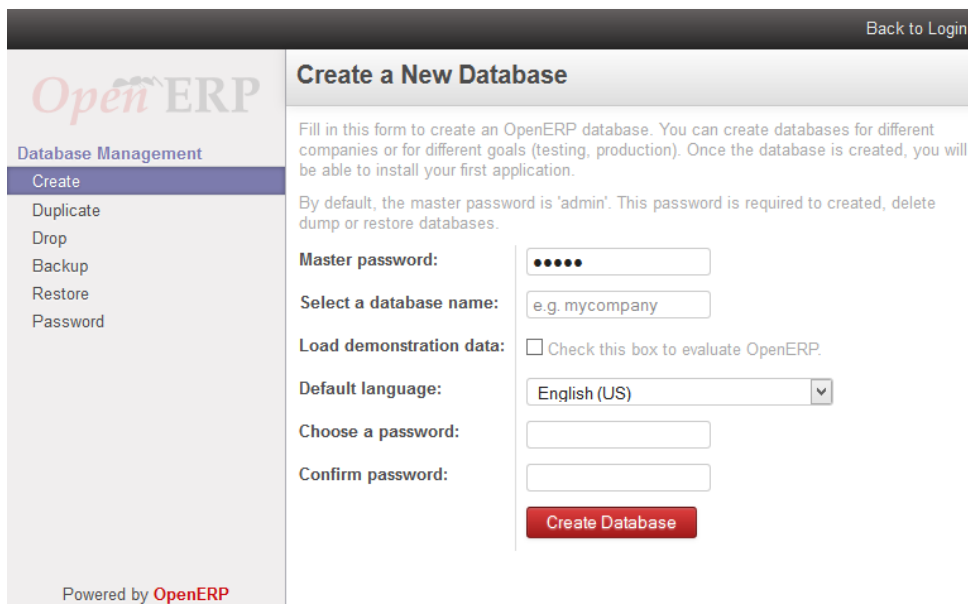
Figura 6 - Interface de instalação de módulos - Screenshot da aplicação

De notar ainda que, se o módulo tiver dependências, e estas não estiverem já instaladas, elas serão instaladas automaticamente. No caso representado na imagem acima, ao instalar o módulo *Point of Sale*, o módulo *Sales and Warehouse Management* será automaticamente instalado.

### 3.5. Cópias de segurança

Nos dias atuais, a informação é um dos bens mais valiosos de qualquer instituição. Por isso, é muito importante definir uma política de cópias de segurança. O *OpenERP* oferece nativamente um ótimo

sistema de cópias de segurança e restauro de bases de dados. Além de cópias de segurança, o utilizador tem ainda a possibilidade de restaurar uma base de dados a partir de uma cópia de segurança feita previamente; duplicar uma base de dados; descartar uma base de dados ou, criar de raiz uma nova base de dados, sem quaisquer dados ou com dados de demonstração.



The screenshot shows the 'Create a New Database' page in the OpenERP application. The page has a dark header with 'Back to Login' on the right. The main content area is titled 'Create a New Database' and contains instructions: 'Fill in this form to create an OpenERP database. You can create databases for different companies or for different goals (testing, production). Once the database is created, you will be able to install your first application.' Below this, it states: 'By default, the master password is 'admin'. This password is required to created, delete dump or restore databases.' The form includes the following fields and options: 'Master password:' (masked with dots), 'Select a database name:' (text input with 'e.g. mycompany' as a placeholder), 'Load demonstration data:' (checkbox, currently unchecked), 'Default language:' (dropdown menu set to 'English (US)'), 'Choose a password:' (text input), and 'Confirm password:' (text input). A red 'Create Database' button is at the bottom right. On the left, there is a sidebar with 'OpenERP' logo and 'Database Management' menu, where 'Create' is selected. At the bottom left, it says 'Powered by OpenERP'.

Figura 7 - Mecanismo de cópias de segurança do OpenERP - Screenshot da aplicação

Todas estas ações são feitas *online*, sem necessidade de conhecimentos técnicos e sem necessidade de parar serviços ou servidores. Para lhes aceder, basta clicar no *link* "Gerir bases de dados" que é apresentado no ecrã inicial do *OpenERP*, abaixo da caixa de *login*, como se pode ver na imagem abaixo.



The screenshot shows the login interface of the OpenERP application. At the top, the 'OpenERP' logo is displayed with the tagline 'OPEN SOURCE BUSINESS APPLICATIONS'. Below the logo is a dark grey login box with a red 'Log in' button. The box contains two input fields: 'Username' with the text 'admin' and 'Password' with masked characters. At the bottom of the login box, there is a red banner with the text 'Manage Databases | Powered by OpenERP'.

Figura 8 - Caixa de Login OpenERP - Screenshot da aplicação



Será então encaminhado para a zona de gestão de bases de dados onde poderá proceder às operações acima enumeradas.

### 3.5.1. Cópias de segurança automatizadas

Apesar da simplicidade do mecanismo de cópias de segurança oferecido pelo OpenERP, é de todo o interesse da instituição não deixar a responsabilidade da segurança dos seus dados nas mãos de um utilizador. É por isso aconselhada a adoção de um mecanismo automatizado de cópias de segurança. Neste caso criam-se dois *scripts bash*, um para fazer cópias de segurança das bases de dados, outro para apagar cópias com mais de 30 dias e adicioná-los à tabela de tarefas *cron jobs* [9].

```
#!/bin/sh
hostname=`hostname`

#####
## OpenERP Backup
## Backup databases: openerpdb1, openerpdb2
#####

# Stop OpenERP Server
/etc/init.d/openerp-server stop

# Dump DBs
for db in openerpdb1 openerpdb2
do
    date=`date +"%Y%m%d_%H%M"N`
    filename="/var/pgdump/${hostname}_${db}_${date}.sql"
    pg_dump -E UTF-8 -p 5433 -F p -b -f $filename $db
    gzip $filename
done

# Start OpenERP Server
/etc/init.d/openerp-server start

exit 0
```

*Conteúdo do ficheiro /var/scripts/dump\_db.sh*

```
#!/bin/sh
path=/var/pgdump
logfile=/var/log/$0

rm -f $logfile
for file in `find /var/pgdump/ -mtime +30 -type f -name *.sql.gz`
do
    echo "deleting: " $file >> $logfile
    rm $file
done

exit 0
```

*Conteúdo do ficheiro /var/scripts/housekeeping.sh*

```
# m h dom mon dow user  command
0 1 * * * postgres /var/scripts/dump_db.sh
0 5 * * * postgres /var/scripts/housekeeping.sh
```

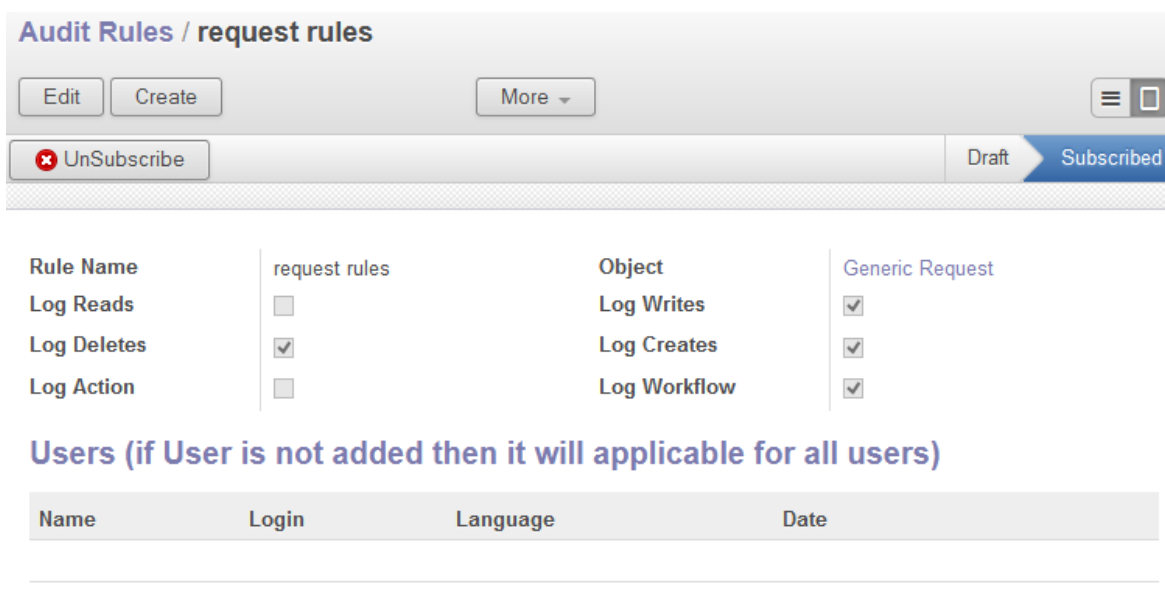
*Linhas a acrescentar ao ficheiro /etc/crontab*

Desta forma os dois *scripts* correrão diariamente, sendo que o *script* de *backup* fica agendado para a 1:00 e o que efectua a limpeza corre às 5:00.

Uma nota final para acrescentar que os restauros de dados feitos a partir destas cópias de segurança deverão ser feitos por linha de comando, uma vez que estas cópias não são compatíveis com o mecanismo de gestão de bases de dados disponibilizado pelo interface web do *OpenERP*.

### 3.6. Auditoria

O OpenERP fornece um módulo oficial de auditoria que permite guardar logs das atividades dos utilizadores sendo assim possível seguir todas as ações efetuadas sobre determinados objetos. Para tal deve instalar-se o módulo *Audit Trail*, um módulo oficial, desenvolvido pela OpenERP S.A. e configurar-se as regras de auditoria. Estas regras indicam ao módulo que objetos devem ser auditados e quais são as ações que devem ser alvo de auditoria. Depois de criada, a regra deve ser subscrita para se tornar ativa. Nas imagens abaixo apresenta-se um exemplo de regra de auditoria e os *logs* por ela gerada.



*Figura 9 - Exemplo de regra de auditoria - Screenshot da aplicação*

| <input type="checkbox"/> | Date                | Resource Name                                 | Object          | Method               | User               |
|--------------------------|---------------------|---|-----------------|----------------------|--------------------|
| <input type="checkbox"/> | 08/20/2014 19:57:12 | Pedido de aquisição de material de escritório | Generic Request | close_request        | User HR            |
| <input type="checkbox"/> | 08/20/2014 19:56:44 | Pedido de aquisição de material de escritório | Generic Request | take_request         | Administrator      |
| <input type="checkbox"/> | 08/20/2014 19:56:11 | Pedido de aquisição de material de escritório | Generic Request | auth_send_processing | Filipe Castanheira |
| <input type="checkbox"/> | 08/20/2014 19:55:16 | Pedido de aquisição de material de escritório | Generic Request | take_request         | User HR            |
| <input type="checkbox"/> | 08/20/2014 19:54:39 | Pedido de aquisição de material de escritório | Generic Request | treat_request        | Administrator      |
| <input type="checkbox"/> | 08/20/2014 19:54:37 | Pedido de aquisição de material de escritório | Generic Request | submit_request       | Administrator      |
| <input type="checkbox"/> | 08/20/2014 19:54:32 | Pedido de aquisição de material de escritório | Generic Request | write                | Administrator      |
| <input type="checkbox"/> | 08/20/2014 19:54:17 | Pedido de aquisição de lápis                  | Generic Request | create               | Administrator      |
| <input type="checkbox"/> | 08/20/2014 19:49:17 | dados   | Generic Request | treat_request        | Administrator      |

Figura 10 - Logs gerados pelo Audit Trail - Screenshot da aplicação

Note-se que uma atualização intensa da aplicação pode gerar muito tráfego e um grande volume de logs, pelo que não se aconselha a ativação do registo de todas as ações sobre o objeto (nomeadamente acessos de leitura) sob pena de poder vir a comprometer a sua performance.

### 3.6.1. Bugs encontrados no módulo *audit\_trail*

Apesar de se tratar de um módulo oficial, isto não implica que este viesse isento de *bugs*. A versão descarregada e instalada tinha um *bug* que impossibilitava a utilização do *OpenERP* quando se subscrevessem algumas regras criadas, nomeadamente quando nessas regras estivesse selecionada a opção de auditoria às alterações de *workflow*. Este *bug* foi identificado e reportado por diversos utilizadores que, inclusive chegaram a apontar várias soluções, não sendo no entanto disponibilizada até à data nenhuma versão oficial, corrigida, do módulo. Por forma a prosseguir com os trabalhos de desenvolvimento, o *bug* foi corrigido manualmente, alterando o código com uma das soluções apresentadas por um dos membros da comunidade.

## 3.7. Autenticação centralizada

Um estudo levado a cabo pela Microsoft em 2007 concluiu que cada utilizador tem, em média, 6.5 palavras-chave diferentes que utiliza em 25 *sites* distintos [10]. O mesmo estudo refere ainda que cada utilizador tem que introduzir a uma palavra-chave 8.11 vezes por dia. Com base nisto, facilmente se conclui que todos os mecanismos implementados no sentido de diminuir estes números são fortemente apreciados pelos utilizadores. Desta forma, torna-se útil encontrar uma solução de autenticação centralizada para aceder ao *OpenERP*. Essa solução é permitida por um módulo oficial, que permite aos utilizadores a autenticação com os dados da sua conta institucional, sendo a verificação das credenciais de autenticação efetuada na árvore *LDAP* existente na instituição.

## Funcionamento da autenticação via LDAP

Como o *OpenERP* suporta tanto utilizadores locais (o administrador, por exemplo) como utilizadores que se autenticam remotamente, a verificação no *LDAP* apenas ocorre quando a autenticação local falha. Sendo os utilizadores criados por defeito sem *password*, a autenticação local falhará sempre, sendo portanto feita a autenticação remota, no servidor *LDAP*.

## Módulos adicionais para popular a base de dados com os utilizadores

Conjuntamente com o módulo oficial de autenticação, foram instalados dois outros módulos, desenvolvidos por membros da comunidade, que permitem popular a base de dados com os utilizadores existentes na árvore *LDAP* da instituição. São eles os módulos *users\_ldap\_populate* e *users\_ldap\_mail*. O primeiro permite carregar na base de dados local os utilizadores, o segundo permite acrescentar dois campos adicionais que permitem popular o nome dos utilizadores e o seu endereço de correio eletrónico.

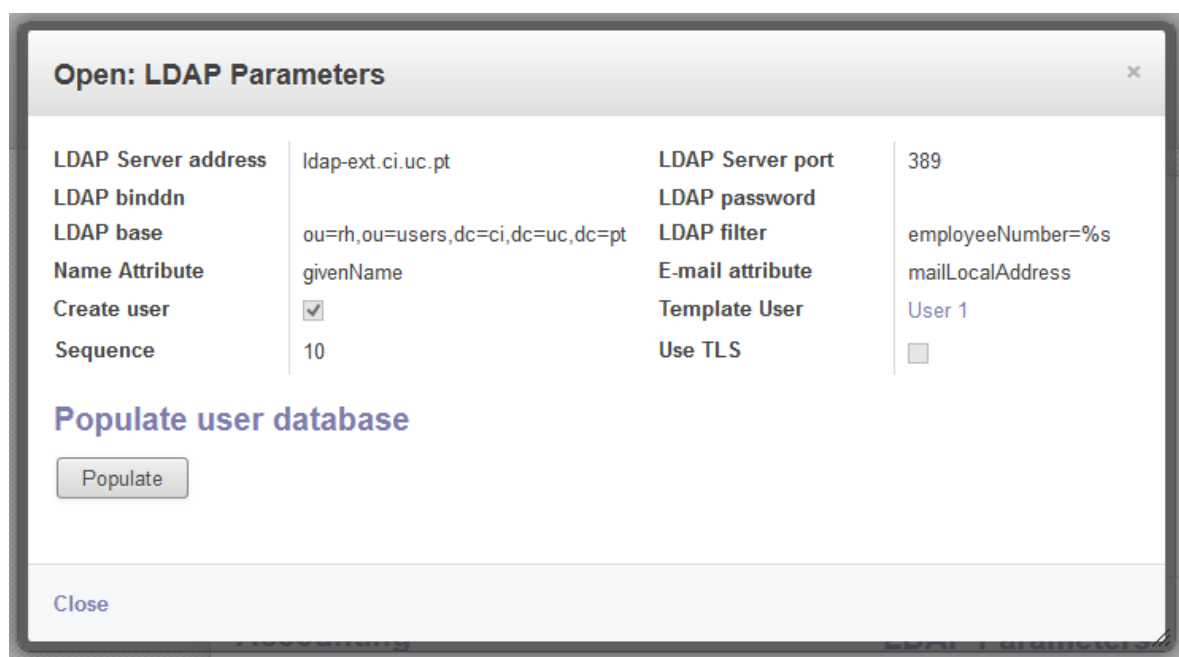


Figura 11 - Configuração LDAP do OpenERP - Screenshot da aplicação

De acordo com as configurações acima, o campo *login* será populado com os número mecanográfico do colaborador, sendo que os utilizadores devem autenticar-se utilizando esse número e a *password* da sua conta institucional. A seleção da *checkbox* "Create user" indica que, em caso de boa autenticação, se o utilizador não tiver sido já criado na base de dados do *OpenERP*, este deverá ser acrescentado utilizando o utilizador *template* indicado. No caso, os novos utilizadores serão criados com o perfil de utilizador normal, sem qualquer privilégio especial.

### **3.7.1. *Bugs encontrados no módulo `ldap_populate`***

Também neste caso, o módulo que vinha com a versão instalada do *OpenERP* tinha um bug que, apesar de não despoletar qualquer mensagem de erro, não permitia a sincronização com o *LDAP*. Tal como no caso do *bug* detetado no módulo de auditoria, também este foi identificado e reportado por vários utilizadores da comunidade sendo, também neste caso, apontadas diversas possíveis soluções.

A abordagem por parte do desenvolvedor deste módulo foi diferente da que foi a da *OpenERP S.A.* no caso descrito acima. Neste caso foi disponibilizada uma versão atualizada do módulo, com o *bug* corrigido. Desta forma, após atualização do módulo (e também do núcleo do *OpenERP*), e criando uma nova base de dados, foi possível importar os utilizadores do *LDAP* e popular a base de dados com os dados desses utilizadores.

## **3.8. Motor de *workflow***

O *Workflow* consiste na automatização de processos onde os documentos, informação ou tarefas são passadas entre os participantes, de acordo com um conjunto de regras definidas com vista atingir ou contribuir para um objetivo global do processo de negócio [11].

O *OpenERP* possui um potente motor de *workflow* totalmente parametrizável. Sendo este estágio sobre modelação de processos, é fundamental poder aceder a esse motor de *workflow* por forma a poder parametriza-lo de acordo com a lógica definida pela instituição. O motor fornecido nesta ferramenta é parametrizável visualmente, o que permite proceder a alterações nos procedimentos sem ser necessária a programação de novas funcionalidades e sem ter que escrever qualquer linha de código, tendo estas alterações efeito imediato sobre o comportamento da aplicação.

## **3.9. Estudo da camada de apresentação**

Foi solicitado ao estagiário que explorasse a possibilidade de se alterar o aspeto visual do cliente *web* do *OpenERP*, com o intuito de aproximar o mais possível o seu aspeto gráfico ao da imagem institucional da Universidade de Coimbra.

Deste estudo se concluiu que, apesar de todas as páginas do *OpenERP* serem parametrizáveis recorrendo à utilização de *CSS*, existem normas rígidas no que se refere aos nomes, logótipos e *links* do *OpenERP* que devem permanecer inalterados [12].

Adicionalmente, também não se vislumbrou uma forma de garantir que as parametrizações efetuadas no cliente *web* da aplicação não sejam esmagadas ao aplicar uma atualização automática da plataforma. Pelo que se abandonou esta abordagem, optando-se por manter a imagem original

do *OpenERP*.

### 3.10. Comparação de funcionalidades do *OutSystems* e *OpenERP*

| Funcionalidade                                   | OutSystems          | OpenERP                       |
|--|---------------------|-------------------------------|
| Ferramenta de desenvolvimento visual             | ✓                   | ✓                             |
| Alterações visuais produzem efeitos imediatos    | ✗                   | ✓                             |
| Geração de código através das alterações visuais | ✓                   | ✗                             |
| Validação automática das alterações              | ✓                   | ✗                             |
| Possibilidade de alterar código manualmente      | ✗                   | ✓                             |
| Gestão de workflow                               | ✓                   | ✓                             |
| Módulos pré-desenvolvidos                        | ✗                   | ✓                             |
| Possibilidade de criar relatórios                | ✓                   | ✓                             |
| Parametrização de processos                      | ✓                   | ✓                             |
| Monitorização da performance da aplicação        | ✓                   | ✗                             |
| Cliente web                                      | ✓                   | ✓                             |
| Ligação segura ao servidor                       | ✓                   | ✓                             |
| Autenticação centralizada                        | ✓                   | ✓                             |
| Controlo de acessos por perfil ou grupo          | ✓                   | ✓                             |
| Auditoria  | ✓                   | ✓                             |
| <b>Licença</b>                                   | <b>Proprietária</b> | <b>Open Source<br/>AGPLv3</b> |

Tabela 3 - Comparação entre funcionalidades do *OutSystems* e *OpenERP*

### 3.11. Conclusões

Com base neste estudo teórico pode afirmar-se que o *OpenERP* aparenta ser uma ferramenta suficientemente flexível e que apresenta características satisfatórias para poder ser adotada pela Universidade de Coimbra no desenvolvimento dos seus processos institucionais. Do estudo comparativo feito entre as funcionalidades oferecidas pelo *OutSystems* e pelo *OpenERP*, convém realçar a incapacidade deste último em converter as parametrizações visuais em código por forma a disponibilizar essas mesmas alterações em versões futuras do mesmo módulo.

## 4. Exploração do *OpenERP* em contexto de trabalho

No sentido de uma maior familiarização com a utilização da plataforma o estágio evoluiu no sentido da exploração da mesma num contexto específico e concreto. Com esse objetivo, o estagiário foi integrado na equipa que acompanhava o processo de entrada em produção do módulo de *PoS* (ponto de venda) para utilização em ambiente real, na Loja da Universidade de Coimbra.

De realçar que, dadas as especificidades das necessidades da referida loja, foi necessário implementar várias alterações ao módulo *PoS* oficial. Estas alterações foram desenvolvidos por um parceiro oficial *OpenERP* em Portugal, contratado pela Universidade de Coimbra para o efeito, estando as especificações dessas alterações a cargo da equipa na qual o estagiário foi integrado.

### 4.1. Tarefas executadas

Neste contexto foram realizadas várias tarefas pelo estagiário, sendo que algumas foram executadas individualmente, nomeadamente a correção dos relatórios de reimpressão de *tickets* de venda e de etiquetas de produto. Outras foram elaboradas em conjunto com outros elementos da equipa – como reuniões com o cliente, instalação e configuração do *PoS*, realização de testes, correção de bugs detetados e produção de documentação de suporte. Outras ainda, onde o estagiário teve apenas uma participação de observador, não sendo por isso relevante – nomeadamente o acompanhamento do processo de importação de produtos do *SAP* e exportação para *SAP* das vendas efetuadas no *PoS*.

#### 4.1.1. Instalação e configuração do *PoS*

Quando o estagiário foi integrado na equipa e começou a acompanhar o processo, já tinha sido feito a instalação do módulo, no entanto participou ativamente no processo de configuração do mesmo, nomeadamente configuração de *pricelists* e tipos de parceiro; configuração de meios de pagamento; criação de lojas e pontos de venda. Mais informação acerca destas tarefas pode ser consultada em documento anexo a este relatório (Anexo V).

#### 4.1.2. Reuniões com o cliente

O estagiário teve também participação ativa em reuniões com o cliente (neste caso na forma de representantes da Loja da Universidade de Coimbra e representantes da Gestão Financeira da Universidade de Coimbra) no sentido de proceder ao levantamento de necessidades e especificação e, posteriormente, à despistagem de erros ou anomalias.

#### **4.1.2.1. Testes e correção de bugs**

Foi fornecido ao cliente um *template*, com o modelo a seguir para a elaboração do plano de testes, e foi solicitado que elaborassem um plano de testes. Este plano que deveria conter o máximo de operações normais do dia-a-dia de um ponto de venda e também os processos de comunicação com a plataforma *SAP*. Os testes foram efetuados, na presença do cliente, e validada a passagem dos mesmos ou, no caso de falha, foram registados os erros ocorridos para futura correção dos bugs detetados.

O plano de testes e seus resultados estão anexados a este relatório (Anexos III e IV, respetivamente).

#### **4.1.3. Produção de documentação**

No seguimento deste acompanhamento foi produzida documentação diversa. Nomeadamente um Manual do Utilizador e do Administrador para o módulo *PoS*, e também um Manual de Configurações e procedimentos.

O primeiro documento tem como público-alvo os utilizadores do *PoS* configurado nesta fase dos trabalhos.

O segundo é um memorando, para utilização interna da Divisão de Sistemas de Informação do Serviço de Gestão de Sistemas e Infraestruturas de Informação e Comunicação da Universidade de Coimbra, para consulta aquando de novas instalações e configurações de instâncias do *PoS*.

Ambos os documentos estão anexos a este relatório (Anexos V e VI).



## 5. Implementação de processos no *OpenERP*

Neste capítulo serão descritas as tarefas desenvolvidas para a implementação dos processos identificados.

### 5.1. Identificação e estudo dos processos a implementar

Sendo o objetivo principal deste estágio o estudo da viabilidade da utilização de uma plataforma de código aberto, em alternativa à plataforma atualmente em utilização, de código proprietária, optou-se por implementar dois processos da instituição atualmente implementados em *OutSystems*. São eles, o pedido genérico e o pedido de deslocação em serviço.

O primeiro processo foi escolhido por ser o mais abrangente pois inclui lógica que é transversal a todos os processos da instituição. O segundo foi escolhido por ser mais complexo, por requerer a participação de mais intervenientes no seu funcionamento, e também por implicar a integração com outras plataformas (nomeadamente o *SAP*), pretendendo-se com esta escolha explorar a versatilidade de integração do *OpenERP* com outras plataformas. Durante esta fase foram realizadas reuniões/entrevistas entre o estagiário e um membro da divisão, destacada para o efeito dentro da equipa do projeto *LUGUS*. Estas reuniões serviram para familiarizar o estagiário com o processo em análise

### 5.2. Análise de requisitos transversais do *workflow* do *LUGUS – OutSystems*

O acesso à plataforma *LUGUS* é limitado aos utilizadores do ramo *RH* do *LDAP* da Universidade de Coimbra. Qualquer utilizador desse ramo pode autenticar-se na plataforma, tendo permissões básicas de acesso à mesma.

As permissões do *LUGUS* são hierárquicas, isto é, os proprietários de um pedido não são utilizadores em concreto mas sim grupos de utilizador. Daí existir o conceito de "grupos de *owner*".

#### Grupos de *Owner*

Ao ser criado um pedido, a propriedade desse pedido é automaticamente atribuída a um determinado grupo de utilizadores (por exemplo, os pedidos genéricos são propriedade do grupo "*request*" e os pedidos de deslocação em serviço são propriedade do grupo "*human resources*"). Desta forma, qualquer pessoa pertencente ao grupo proprietário do pedido tem acesso ao mesmo, podendo então reclamar o pedido para si, passando a ser o seu executante.

O pedido deixa de estar visível publicamente para os restantes membros do grupo (mas continua visível para consulta através da pesquisa). Apenas o executante do pedido pode passar o pedido

para outra pessoa (não é permitido a um utilizador tomar a execução de um pedido para si, mesmo que pertença ao mesmo grupo).

### 5.2.1. Características transversais a todos os tipos de pedido

Foram identificadas três características que são transversais a todos os tipos de pedido. São elas:

- **Ficheiros**

Esta funcionalidade permite anexar ficheiros, com informação adicional ao pedido. O *LUGUS* interage com a plataforma *KT* (plataforma de gestão documental), onde os ficheiros ficam armazenados, associados ao pedido.

- **Pareceres**

Esta funcionalidade permite pedir a emissão de pareceres a outros utilizadores da plataforma. O utilizador a quem é solicitado o parecer deverá ter acesso aos dados do pedido, podendo assim emitir um parecer.

- **Tracing**

Esta funcionalidade permite aos utilizadores visualizar o estado em que se encontra o pedido, sendo também possível visualizar um resumo do percurso do pedido, desde a sua criação, até ao estado atual.

## 5.3. Análise prática do motor de *workflow* do *OpenERP*

O *OpenERP* fornece um ágil motor de *workflow* que permite garantir a sequência lógica das tarefas a executar sobre um determinado registo.

Os *workflows* definem os diversos estágios por que um determinado processo deve passar, sendo que, de acordo com determinadas condições uma transição pode ou não ser efetuada.

Os *workflows* são definidos por atividades e transições entre essas mesmas atividades. A definição dessas atividades e transições é feita, no *OpenERP*, num ficheiro *XML*.

Nesse ficheiro constam:

- A definição do *workflow*.

Na definição consta o nome do *workflow*, o objeto sobre o qual este tem efeito e se deve ou não ser iniciado novo *workflow* aquando da criação de um novo registo.

**Exemplo de definição de *workflow* (só a definição):**

```
<record model="workflow" id="wkf_opinion">
  <field name="name">opinion.wkf</field>
  <field name="osv">opinion</field>
  <field name="on_create">True</field>
</record>
```

- A definição das atividades.

As atividades são os nós do *workflow*, ou seja, os estados possíveis que o processo pode ter. Na definição de cada atividade, há que ter o cuidado de associá-la ao *id* do *workflow* definido anteriormente.

**Exemplo de definição de atividades:**

```
<record model="workflow.activity" id="act_requested">
  <field name="wkf_id" ref="wkf_opinion" />
  <field name="flow_start">True</field>
  <field name="name">opinion_requested</field>
  <field name="kind">function</field>
  <field name="action">request_opinion()</field>
</record>
<record model="workflow.activity" id="act_reviewing">
  <field name="wkf_id" ref="wkf_opinion" />
  <field name="name">opinion_reviewing</field>
  <field name="kind">function</field>
  <field name="action">review_opinion()</field>
</record>
<record model="workflow.activity" id="act_issued">
  <field name="wkf_id" ref="wkf_opinion" />
  <field name="flow_stop">True</field>
  <field name="name">opinion_issued</field>
  <field name="kind">function</field>
  <field name="action">issue_opinion()</field>
</record>
```

- A definição das transições.

As transições são as ligações entre os nós do *workflow*, e serão tomadas sempre que uma ação seja tomada e devolva verdadeiro. Na definição das transições consta a atividade de origem da transição, a atividade de destino e o sinal que despoleta a transição.

**Exemplo de definição de transição:**

```
<record model="workflow.transition" id="opinion_t1">
  <field name="act_from" ref="act_requested" />
  <field name="act_to" ref="act_reviewing" />
  <field name="signal">review_opinion</field>
</record>
<record model="workflow.transition" id="opinion_t2">
  <field name="act_from" ref="act_reviewing" />
  <field name="act_to" ref="act_issued" />
  <field name="signal">issue_opinion</field>
</record>
```

- A definição de condições.

As condições não são mais do que transições que são tomadas de forma automática pelo *workflow* se estas forem satisfeitas. Ao contrário das transições definidas anteriormente -

que são tomadas apenas através de uma ação por parte do utilizador que envia um sinal ao *workflow* e indica que aquela transição deve ser tomada – as condições permitem passar ao estado seguinte, não parando num estado intermédio, quando a condição é satisfeita.

**Exemplo da definição de uma condição:**

```
<record model="workflow.transition" id="request_t102">
  <field name="act_from" ref="act_submit" />
  <field name="act_to" ref="act_confirm" />
  <field name="condition">check_responsible()</field>
</record>
```

No exemplo acima, o pedido passa imediatamente para o estado "confirmado" se o valor devolvido pela função *check\_responsible()* for verdadeiro.

**Exemplo de código de uma função de condição do workflow:**

```
def check_responsible(self, cr, uid, ids, context=None):
    return self.browse(cr, uid, ids, context)[0].is_responsible
```

No exemplo acima, a função de verificação da condição do workflow devolve verdadeiro ou falso, consoante o utilizador seja responsável ou não pelo pedido (um *booleano* guardado no campo *is\_responsible*).

## 5.4. Implementação do Processo A: Pedido genérico

Neste ponto do documento será descrito todo o processo que levou à implementação do pedido genérico, desde a sua análise, validação do documento e implementação propriamente dita. Serão justificadas as opções tomadas e identificadas as dificuldades encontradas.

### 5.4.1. Análise do processo, validação e adaptação da especificação

Durante esta fase do processo foram realizadas reuniões com o elemento da equipa LUGUS destacado para o efeito. Nessas reuniões foi analisado o fluxograma do processo e a forma como este foi implementado em *OutSystems*. Ainda durante estas reuniões procedeu-se ao levantamento dos requisitos para o funcionamento do processo no sentido de elaborar uma especificação funcional (Anexo X) e uma especificação técnica (Anexo XI).

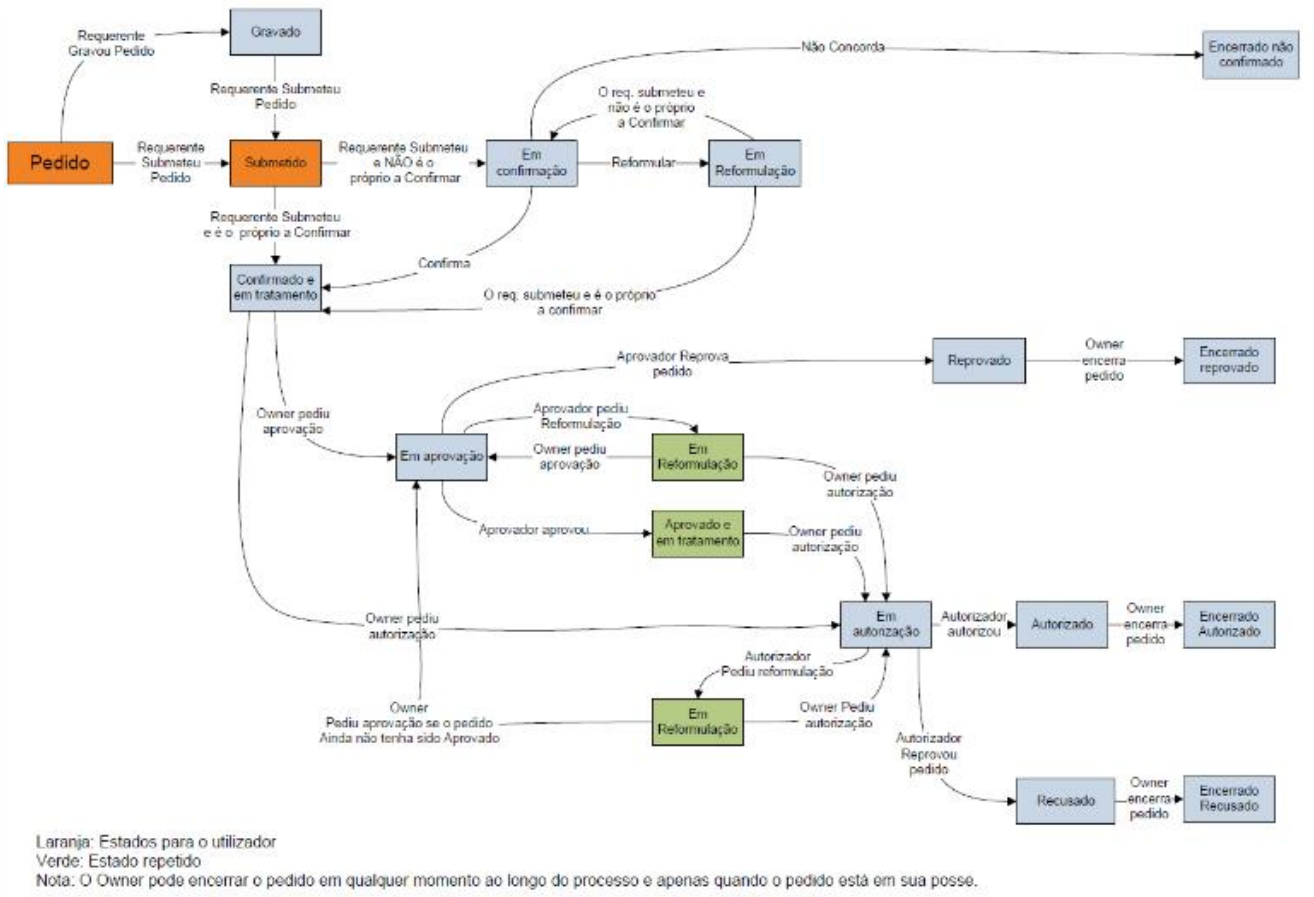


Figura 12 - Fluxograma da implementação do pedido genérico em OutSystems

(Diagrama fornecido pela equipa do projeto LUGUS – Anexo VII)

Com base no fluxograma fornecido pela equipa LUGUS, foi elaborado novo fluxograma, em notação *BPMN* para a implementação deste processo no *OpenERP* (ver anexos VIII e IX deste documento)

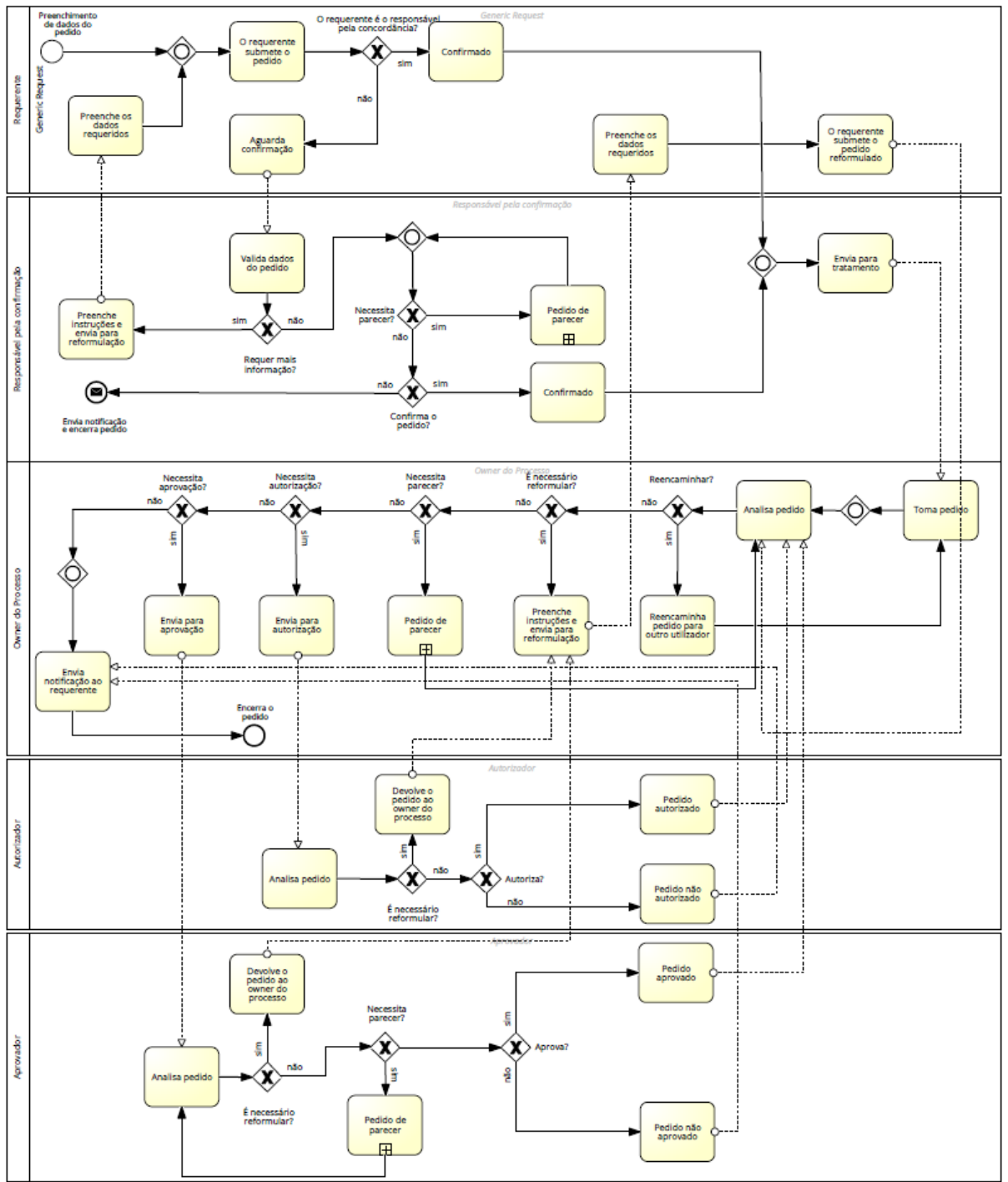


Figura 13 - Modelação do pedido genérico em notação BPMN

(Gerado online em <http://academic.signavio.com>)

### 5.4.2. Fase de desenvolvimento

Após os períodos de estudo do processo e respetiva especificação, deu-se início à fase de desenvolvimento e implementação do mesmo. Esta fase veio a tornar-se mais morosa do que o

inicialmente previsto pelos motivos que serão descritos ao longo desta secção do documento.

### 5.4.2.1. *Application Builder vs Escrita de Código*

O *OpenERP* fornece uma ferramenta gráfica de desenvolvimento rápido a *Application Builder*. Note-se que, apesar de ser afirmado muitas vezes que, graças a essa funcionalidade, se podem efetuar alteração e acrescentar funcionalidades ao *OpenERP* sem ter que se escrever uma única linha de código, isso não é totalmente verdade. Grande parte das funcionalidades implementadas através desta ferramenta implicam a escrita de código no interface gráfico, nomeadamente para a definição das vistas ou mesmo para definições de funções em *python* para as ações do workflow.

O editor visual de vistas dos objetos, por exemplo, pouco mais faz do que editar linha a linha uma espécie de "XML simplificado", como se pode ver na imagem abaixo.

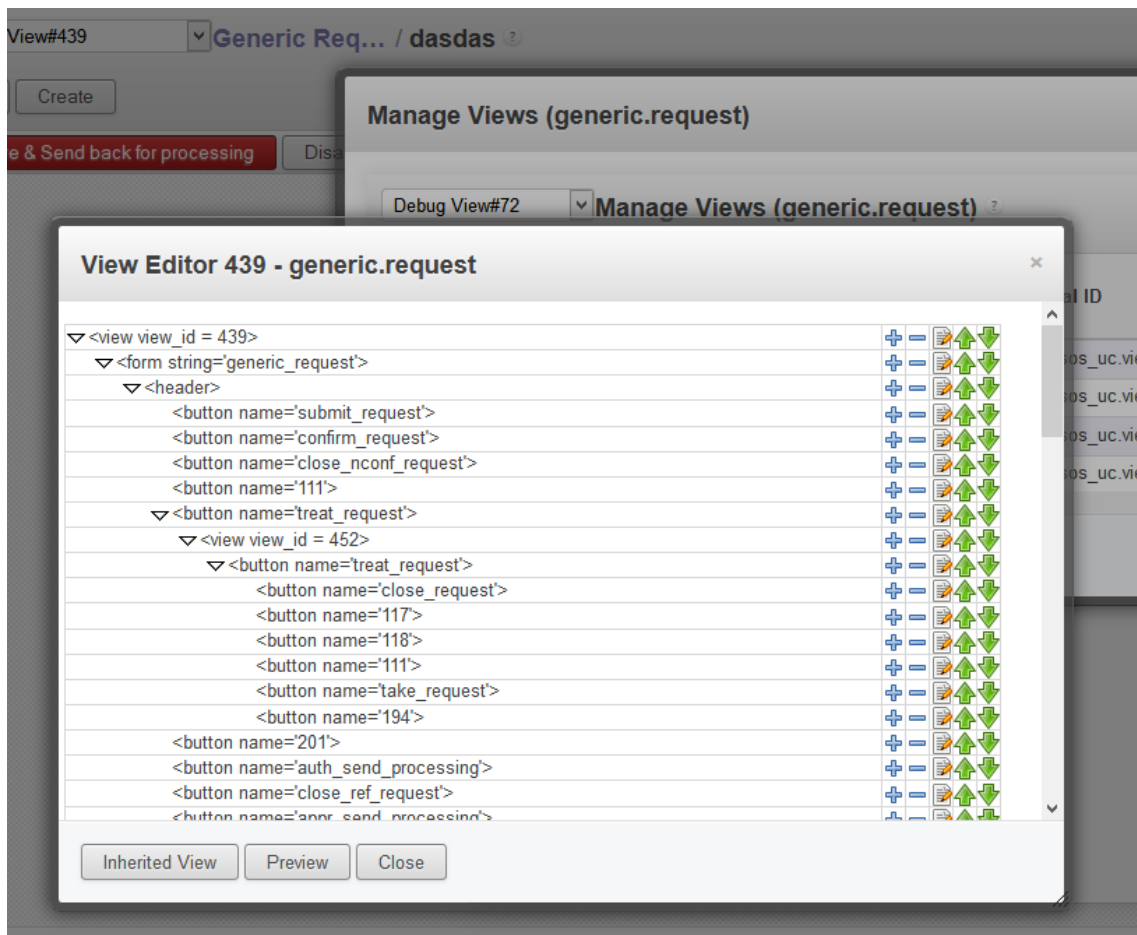


Figura 14 - Application Builder - View Editor - Screenshot da aplicação

Tendo em conta que foi incessantemente lembrado durante a formação teórico-prática tida no primeiro semestre que as alterações feitas diretamente (no código) de módulos existentes, ou feitas com auxílio do das ferramentas visuais podem ser esmagadas por uma atualização do módulo ou da

plataforma, optou-se por não utilizar a *Application Builder* para a implementação dos processos. Outro argumento válido para a não utilização das ferramentas gráficas fornecidas pela plataforma é o facto de as funcionalidades implementadas visualmente terem validade "local". Isto é, todas as parametrizações levadas a cabo visualmente são gravadas na base de dados do servidor e apenas têm repercussão nessa instância. Querendo isto dizer que uma eventual migração para outro servidor, ou a necessidade de instalação "fresca" do módulo (sem os dados de teste obrigatoriamente introduzidos durante o desenvolvimento) implica a perda total do trabalho de desenvolvimento.

Tendo tudo isto em mente, optou-se por implementar todo o processo em código. Esta opção leva a um maior tempo de desenvolvimento, no entanto há ganho de tempo em futuras reinstalações do módulo, uma vez que desta forma garantimos que sempre haja lugar a uma atualização do módulo este se encontra perfeitamente parametrizado, incluído o carregamento automatizado de dados, configurações de ações no servidor (como envio automatizado de emails), criação de grupos, permissões de acesso com base nesses grupos, etc.

#### *5.4.2.2. Dificuldades encontradas durante o desenvolvimento*

Serão de seguida elencadas algumas das maiores dificuldades encontradas no decorrer do estágio, durante a fase de desenvolvimento.

##### **Despistagem de bugs em módulos existentes**

Como foi já dito acima neste documento, alguns dos módulos instalados, necessários para o bom funcionamento da aplicação, não se encontravam em boas condições de funcionamento. Foi necessário identificar e encontrar solução para o problema de funcionamento do módulo de auditoria. E foi também necessário proceder a diversos testes e configurações para perceber qual o problema do módulo que viria a permitir popular a base de dados com os utilizadores do ramo *RH* do *LDAP* da Universidade de Coimbra. Estes dois bugs consumiram quase uma semana de trabalho e pesquisa.

##### **Fraca documentação e comunidade pouco participativa**

Todo o processo de implementação foi mais moroso do que o esperado, sobretudo pela falta de documentação de suporte ao produto. A documentação disponibilizada no *site* do produto é, quase sempre, vaga, escassa e algumas vezes até inexistente<sup>1</sup>. O que levou a ter de recorrer muitas vezes ao suporte da comunidade. Mas mesmo esse suporte nem sempre é o mais adequado pois percebe-se que as respostas dadas são também vagas e algumas vezes erradas ou nada têm a ver com o

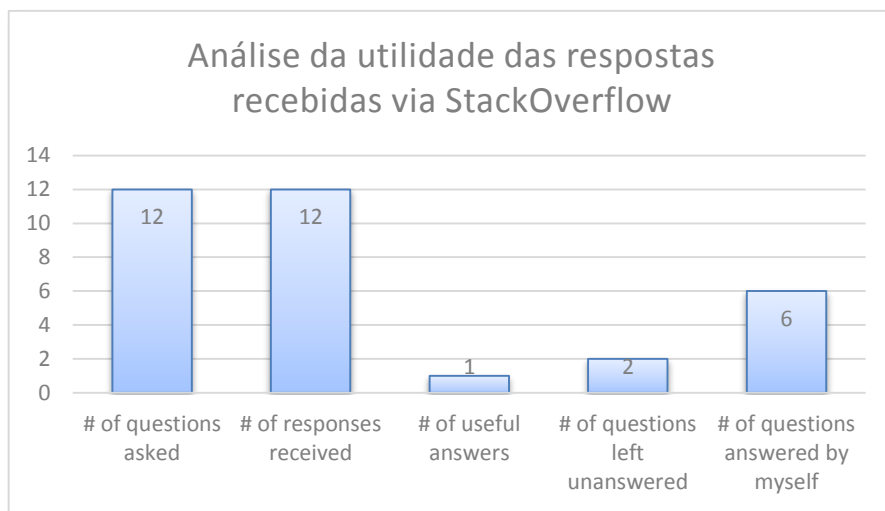
---

<sup>1</sup> Por exemplo esta página da documentação oficial, onde deveriam constar instruções de como configurar uma "Form View" em XML para a versão 7.0 do *OpenERP* encontra-se totalmente em branco [https://doc.odoo.com/7.0/usability\\_book/3.%20OpenERP%27s%20Views/Form%20Views/](https://doc.odoo.com/7.0/usability_book/3.%20OpenERP%27s%20Views/Form%20Views/)



problema em questão.

Apresentam-se de seguida gráficos de análise do grau de utilidade de dois dos fóruns de comunidade frequentados durante o decorrer deste estágio, são eles o *StackOverflow*<sup>2</sup> e o *Fórum OpenERP*<sup>3</sup>.



*Figura 15 - Gráfico de análise da utilidade de respostas recebida – StackOverflow*

Como se pode ver no gráfico acima, das 12 questões colocadas neste fórum, foi obtido um total de 12 respostas para 10 questões, sendo deixadas 2 questões sem qualquer resposta. No entanto, destas respostas, apenas 1 foi útil e levou à resolução do problema ou dissipou a dúvida. Seis das questões colocadas acabaram por ser resolvidas pelo estagiário, através de pesquisas em artigos publicados em *sites* e/ou *blogs* particulares de membros da comunidade ou então, por tentativa e erro.

---

<sup>2</sup> <http://stackoverflow.com/>

<sup>3</sup> <http://help.openerp.com/>

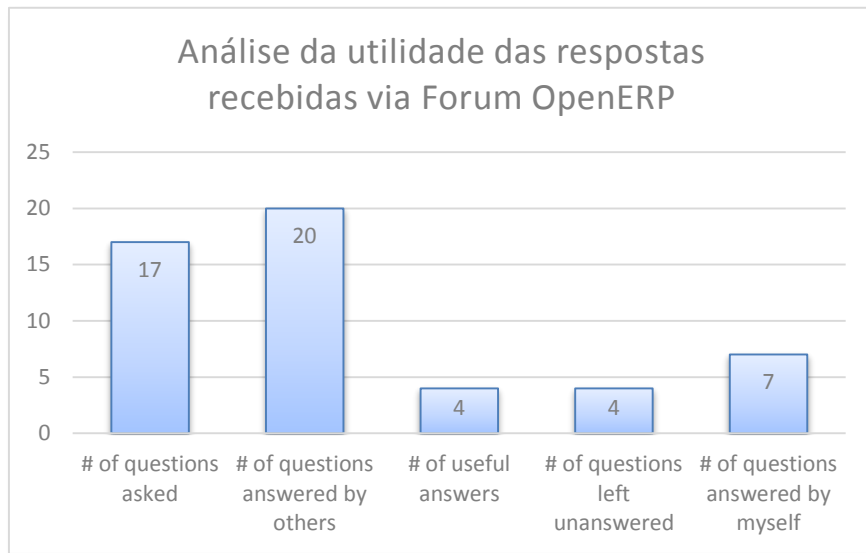


Figura 16 - Gráfico de análise da utilidade de respostas recebida - Fórum OpenERP

No fórum OpenERP foram colocadas 17 questões, sendo obtidas um total de 20 respostas, para 13 das questões colocadas, ficando 4 questões sem qualquer resposta. Das respostas recebidas, apenas 4 foram de alguma utilidade. Sete destas questões colocadas foram resolvidas pelo estagiário.

Fazendo uma análise crítica à capacidade de resposta destas duas comunidades conclui-se que, apesar de haver uma razoável taxa de resposta às questões colocadas (cerca de 80% das questões colocadas obtiveram resposta), o grau de utilidade das mesmas deixa a desejar numa comunidade que se espera ativa e prestável (15,6% de utilidade).

No que se refere à participação ativa do estagiário nestes fóruns, seguem abaixo gráficos onde são indicadas o número de respostas dadas, o número de respostas aceites e o número de respostas votadas positivamente.

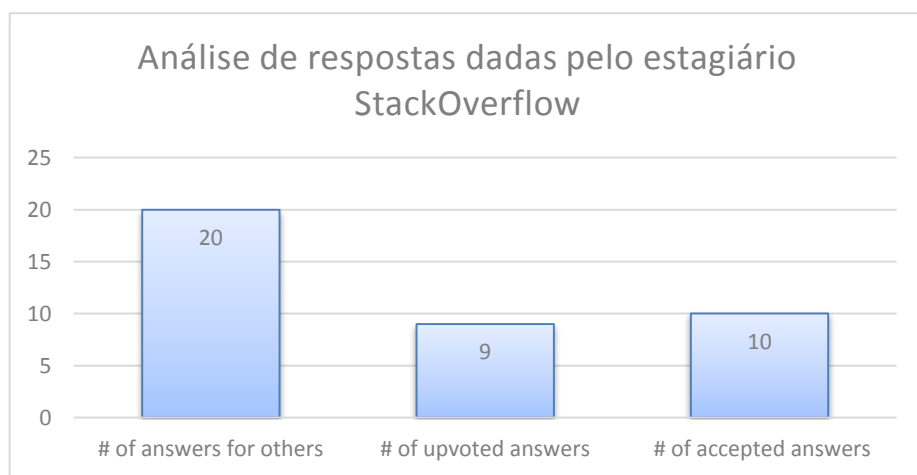


Figura 17 - Gráfico de análise de respostas dadas pelo estagiário – StackOverflow

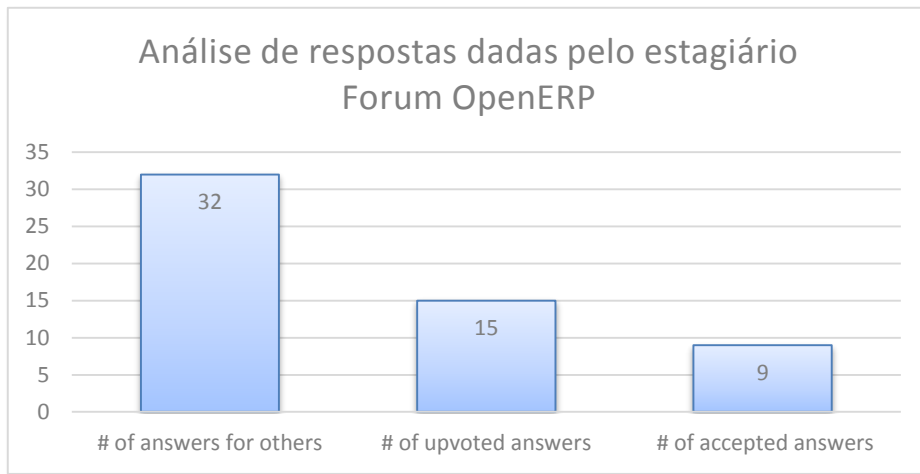


Figura 18 - Gráfico de análise de respostas dadas pelo estagiário - Fórum OpenERP

Sempre que tinha disponibilidade, o estagiário tentou ter uma participação ativa nos fóruns da comunidade, dando respostas claras, acompanhadas com exemplos de código. Como se pode ver nos gráficos acima, o estagiário tentou ter uma participação ativa nos fóruns da comunidade. Dando respostas sempre que possível, das 20 respostas dadas no *StackOverflow*, 10 foram aceites (consideradas totalmente certas portanto) e 9 foram votadas positivamente (consideradas portanto boas respostas e que levam à resolução da questão). No *fórum OpenERP* das 32 respostas dadas, 9 foram aceites e 15 foram votadas positivamente.

### Dificuldade em fazer *debug*

Outro problema frequentemente encontrado durante o desenvolvimento do módulo foi a dificuldade extrema em fazer *debug* no *OpenERP*, sempre que havia um problema com o desenvolvimento. Devido à falta de documentação e de feedback por parte da comunidade, foi várias vezes necessário recorrer ao método da tentativa e erro, com *debug* das variáveis, durante o decorrer deste estágio. Este método não é prático, nem aconselhado, em nenhuma situação e mostrou-se particularmente desafiante no *OpenERP*, que demonstrou ser uma aplicação onde é particularmente difícil perceber em tempo real qual o valor presente nas variáveis e até, por vezes, qual o seu formato.

### 5.4.2.3. Integração com sistemas externos

O *OpenERP* é apresentado como sendo uma ferramenta extremamente flexível e que é facilmente integrada com outros sistemas existentes na instituição. No entanto, isto apenas é verdade, se se entender como *integração com outros sistemas* o fornecimento de informação gerida por parte desta ferramenta aos sistemas externos.

O *OpenERP* funciona como um fornecedor de serviços, respondendo a chamadas remotas (o servidor responde aos pedidos *JSON-RPC* feitos pelo cliente web, e *XML-RPC* efetuadas por outros clientes).

No entanto, quando se trata enviar pedidos a serviços remotos e aguardar a sua resposta para proceder a novas ações, não foi encontrada qualquer documentação que indicasse que essa funcionalidade era suportada.

Uma das questões colocadas nos fóruns da comunidade foi precisamente esta: “como integrar o *OpenERP* com um sistema de *file hosting* externo, ficando os binários guardados no sistema externo e o identificador gerado pelo sistema externo e restantes *metadados* do lado do *OpenERP*?”.

Apesar do número de visualizações (mais de 300, no total dos dois fóruns), nenhum participante arriscou qualquer resposta. Desta forma, assume-se que este tipo de integração só é possível colocando a lógica de comunicação com o sistema externo do lado do cliente. Por isso mesmo, para possibilitar este tipo de integração seria necessário contruir um novo cliente web, ou reescrever o cliente existente, o que, no tempo deste estágio não foi de todo possível, pelo que não foi possível implementar esta funcionalidade.

### **5.4.3. Testes**

Ao longo do processo de implementação foram sendo efetuados testes funcionais para verificar o funcionamento da aplicação. Já com o processo praticamente implementado, foi elaborada uma bateria de testes que foram efetuados por forma a validar o funcionamento da aplicação, de acordo com o especificado.

Os testes e o resultado dos mesmos pode ser consultado em anexo a este documento (Anexos XII e XIII).

## **5.5. Processo B: Pedido de deslocação em serviço**

Devido às dificuldades encontradas no decorrer da implementação do processo do pedido genérico e às constantes derrapagens nos prazos daí resultantes, não houve possibilidade de, dentro dos *timings* deste estágio, proceder à implementação do segundo processo.

Foi efetuada apenas uma breve análise do referido processo através observação do seu workflow (Anexo VII) e de reunião com a equipa do projeto *LUGUS*. Desta análise se concluiu que se trata de um processo algo mais complexo que o primeiro, sobretudo no número de intervenientes no processo. Os intervenientes são, além do requerente, do confirmador, do executante do grupo (*GRH*) *owner* do processo e dos aprovador e/ou autorizador, intervêm neste processo a contabilidade a quem é atribuída a responsabilidade de cabimentar e pedir elegibilidade ou dotação, o gabinete de projetos ou orçamento e contas (dependendo a quem a contabilidade atribui a responsabilidade do processo) e finalmente a tesouraria que efetuará os adiantamentos ou pagamentos conforme tenha

sido solicitado pelo requerente.

Existem algumas semelhanças entre este processo e o pedido genérico, pelo que é possível reutilização de alguma da lógica implementada no pedido genérico. O processo de criação do pedido e confirmação e envio para tratamento tem o mesmo fluxo, e também os processos de aprovação e/ou autorização.

Uma das grandes características diferenciadoras deste processo em relação ao anterior tem a ver com a possibilidade descrição dos encargos inerentes à deslocação, no caso de estes encargos serem da responsabilidade da Universidade de Coimbra (não se tratando de reembolsos), deverá ser criado um novo processo de pedido de compra no *OpenERP*, que deverá ser associado ao pedido de deslocação em serviço que lhe deu origem. Este novo processo - o "pedido de compra" - é um processo independente, com workflow próprio, que deverá também ser implementado para garantir o bom funcionamento do processo de pedido de deslocação em serviço.

A integração que este processo oferece com *SAP* não deverá apresentar problemas de maior, uma vez que essa integração é unidirecional. Isto é, a dada altura do processo é apresentado um formulário ao *owner* do processo (*GRH*) e esses dados são enviados para *SAP*, sendo a responsabilidade do processamento dos dados aí inseridos da responsabilidade do *SAP* a partir daí, não sendo esperada qualquer resposta pelo *OpenERP*.

### **Tempo de desenvolvimento**

A estimativa de tempo dada para o desenvolvimento deste processo em *OutSystems* é de oito a nove semanas. Esta estimativa não inclui o desenvolvimento do pedido de compras.

Com base na experiência adquirida no desenvolvimento do pedido genérico, uma previsão do tempo de desenvolvimento para este pedido em *OpenERP* é de doze semanas, mais uma semana para especificação e outra para testes. Esta estimativa também não inclui o desenvolvimento do pedido de compras.

## 6. Análise comparativa das implementações

Nesta secção será feita uma análise comparativa entre os resultados obtidos no *OpenERP* e do *LUGUS*.

### 6.1.1. Análise funcional e desempenho

Nesta secção do documento apresentam-se algumas comparações entre funcionalidades e um eventual ganho de desempenho em utilizar uma ou outra solução.

Relativamente às funcionalidades, era objetivo deste estágio replicar o processo implementado em *OutSystems* pelo que as funcionalidades são idênticas com uma ou outra ligeira diferença.

| Funcionalidade   | OutSystems | OpenERP |
|--|------------|---------|
| Criação de pedidos por parte de qualquer utilizador do ramo RH                               | ✓          | ✓       |
| Preenchimento automático do campo responsável caso o utilizador seja responsável pelo pedido | ✓          | ✓       |
| Validação dos campos de preenchimento obrigatório  | ✓          | ✓       |
| Validação da consistência dos dados  | ✓          | ✓       |
| Acessos diferenciados por tipo de permissão  | ✓          | ✓       |
| Confirmação do pedido  | ✓          | ✓       |
| Propriedade do pedido por grupo  | ✓          | ✓       |
| Tomada de um pedido  | ✓          | ✓       |
| Reencaminhamento de um pedido  | ✓          | ✓       |
| Avocação de um pedido a outro utilizador dentro do grupo proprietário                        | ⊘          | ✓       |
| Pedido de parecer  | ✓          | ✓       |
| Emissão de parecer   | ✓          | ✓       |
| Pedido de aprovação / autorização do pedido  | ✓          | ✓       |
| Reformulação do pedido   | ✓          | ✓       |
| Registo de <i>tracing</i>  | ✓          | ✓       |
| Encerramento do pedido   | ✓          | ✓       |
| Notificações automáticas por <i>email</i>  | ✓          | ✓       |
| Possibilidade de anexar ficheiros ao pedido  | ✓          | ✓       |
| Integração com serviço externo de <i>file hosting</i>  | ✓          | ⊘       |

Tabela 4 - Análise funcional comparativa - Pedido genérico - LUGUS vs OpenERP

Na tabela acima podem ver-se as principais funcionalidades implementadas no OpenERP, sendo que esta solução fica a perder, relativamente ao LUGUS por não ter sido implementada a funcionalidade de integração com o serviço externo de *file hosting*.

### 6.1.1.1. Usabilidade

Relativamente ao desempenho de usabilidade, apresentam-se de seguida *screenshots* das duas aplicações, para as mesmas ações, que mostram um ligeiro ganho para o *OpenERP*, sobretudo monitores com baixas resoluções.

**Criar Pedido** Need Help ?

**Tipo de Pedido:** Genérico  
**Finalidade:** Para o qual não exista formulário específico

N.º do Requerente: lfc  
 Nome do Requerente: Filipe Castanheira

Lista de pedidos

Gravar Submeter Eliminar

Dados Gerais

**Identificação do responsável**

É o responsável?  ?

Nome do Responsável: Type or double-click for list ?

Orgânica do Responsável: ?

Orçamento de Gestão: - ?

**Pedido** 1024\*768 px

Tipo de Assunto: -- Selecione Assunto -- ?

Classificador: -- Selecione Classificador --

Breve Descrição do Assunto: ?

Descrição da Necessidade: \*

Fundamentação: \*

Observações: ?

Data esperada: YYYY-MM-DD \* ?

Gravar Submeter Eliminar

Figura 19 - Criar pedido no LUGUS - Screenshot da aplicação

Na imagem acima, pode verificar-se que um colaborador com um monitor com resolução 1024\*768px terá mais de metade do formulário fora do alcance imediato, estando cinco campo de preenchimento obrigatório fora do seu campo de visão.

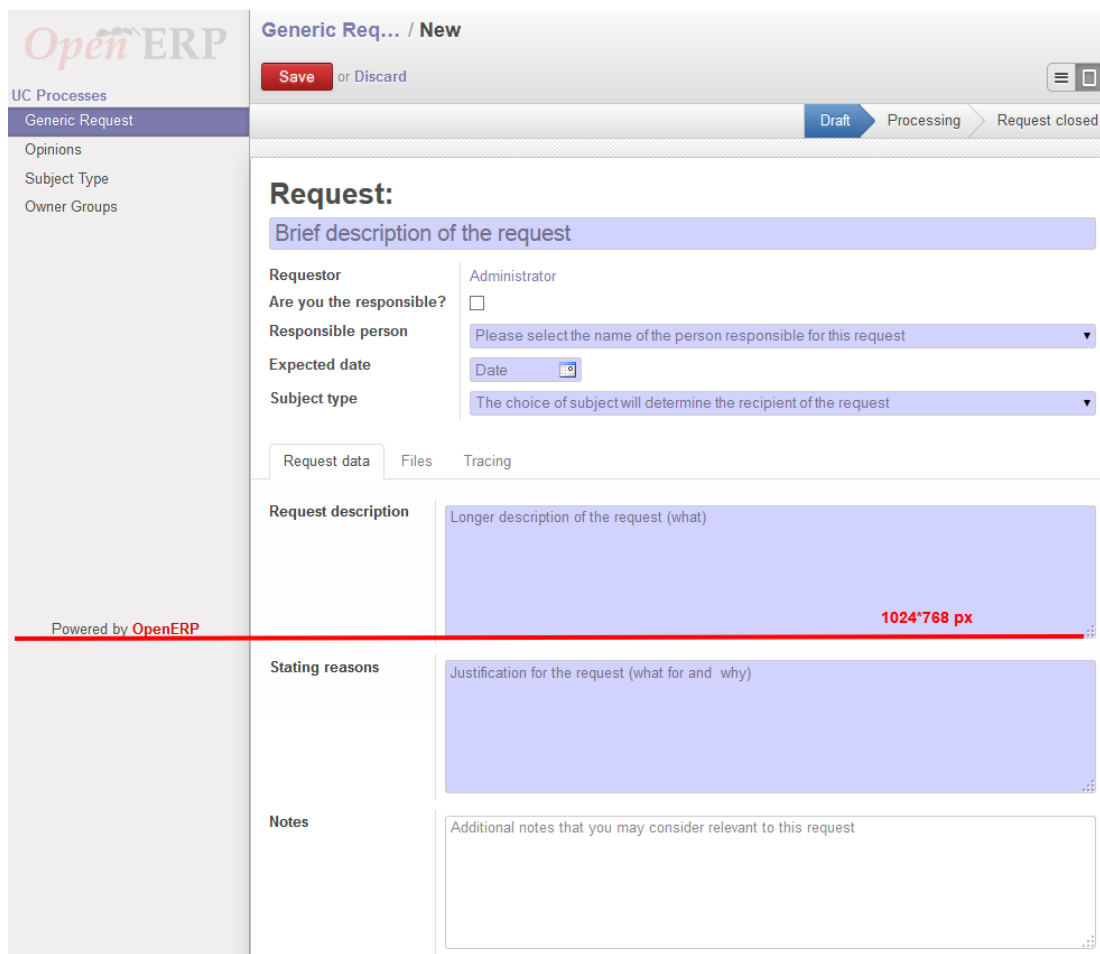


Figura 20 - Criar pedido no OpenERP - Screenshot da aplicação

Por outro lado, com uma melhor disposição da informação, o *OpenERP* tem apenas um campo de preenchimento obrigatório fora do ecrã, sendo que com resoluções superiores, todo o formulário fica visível.

A diferença entre a forma como é feita a apresentação dos dados do pedido apenas para consulta é ainda maior como poderemos ver nas imagens abaixo.



**Editar Pedido**  
 Tipo de Pedido: Genérico  
 Finalidade: Para o qual não exista formulário específico

|                    |                    |                  |         |
|--------------------|--------------------|------------------|---------|
| N.º do Requerente  | lfc                | Estado do Pedido | Gravado |
| Nome do Requerente | Filipe Castanheira | N.º do Pedido    | 3155    |
| Na posse de        | -                  |                  |         |

Lista de pedidos

Dados Gerais | Ficheiros Anexos | Pareceres | Tracing

**Identificação do responsável**

É o responsável?    
 Nome do Responsável: Filipe Castanheira   
 Orgânica do Responsável: Reitoria   
 Orçamento de Gestão: -

**Pedido**

Tipo de Assunto: [GSIIIC]   
 Classificador: -- Seleccione Classificador --   
 Breve Descrição do Assunto: Pedido de teste   
 Descrição da Necessidade: Descrição da necessidade   
 Fundamentação: Fundamentação do pedido   
 Observações: Observações   
 Data esperada: 2014-08-25

Figura 21 - Visualização dos dados de um pedido genérico no LUGUS - Screenshot da aplicação

O ecrã que permite a visualização dos dados do pedido é o mesmo que o que permite criar um novo pedido com a diferença deste ter os campos do formulário desativados. Mais uma vez, grande parte dos dados ficam fora com campo de visão imediato do utilizador, e não é claro como o utilizador deve proceder para alterar os dados do pedido ou para submetê-lo.



Figura 22 - Visualização de dados do pedido genérico no OpenERP - Screenshot da aplicação

Neste cenário, o *OpenERP* fica claramente a ganhar em relação ao *LUGUS* pois toda a informação do pedido é mostrada num único ecrã, sem necessidade de fazer *scroll down*. É dado grande destaque à informação de identificação do pedido – número e assunto. Também a informação de identificação do requerente e responsável está destacada da restante informação do pedido. É clara a forma de como o utilizador pode interagir com o pedido, com os botões "edit" e "submit" em destaque, estando mais destacado o botão de ação "submit", o que permite ao utilizador perceber que esta será a próxima ação lógica a executar sobre o pedido.

Assim sendo, conclui-se que, se em termos de funcionalidade, o *OpenERP* pode estar atrás do *LUGUS*, no que se refere ao desempenho dos utilizadores na sua utilização, este leva vantagem sobre o concorrente proprietário por permitir ecrãs mais limpos e havendo necessidade de menos cliques para efetuar operações sobre os pedidos.

### 6.1.1.2. Testes de carga

Foram realizados testes de carga para avaliar a capacidade de resposta do *OpenERP*. A ferramenta utilizada para estes testes foi o *JMeter*. O *JMeter* é uma aplicação *Open Source* cujo funcionamento permite simular o comportamento de um cliente web e possui várias ferramentas de monitorização do comportamento do servidor. Neste contexto, configurou-se o *JMeter* para proceder a três tipos de ação típicas da utilização do *OpenERP*: autenticação, leitura dos dados de um pedido e criação de um novo pedido. Estas ações são configuradas no *JMeter* e envia chamadas XML-RPC ao servidor.

Foi simulado um universo de 500 utilizadores e três cenários com diferentes níveis de acessos concorrentes à plataforma *OpenERP*, sendo para tal alterado o parâmetro "*Ramp Up*" na ferramenta de testes.

**Ramp Up** - É o tempo que o *JMeter* deve demorar para que todas as *threads* sejam iniciadas (cada *thread* corresponde a um utilizador). Exemplo: tendo 500 *threads* e um *ramp up* de 100 segundos, cada *thread* vai iniciar 0.2 segundos depois da anterior ter iniciado, resultando desta forma em média 5 acessos concorrentes.

### Scripts de configuração *JMeter*:

#### *XML-RPC Login*

```
<?xml version="1.0"?>
<methodCall>
  <methodName>login</methodName>
  <params>
    <param>
      <value>may_9</value>
    </param>
    <param>
      <value>admin</value>
    </param>
    <param>
      <value>admin</value>
    </param>
  </params>
</methodCall>
```

#### *XML-RPC Read Request*

```
<?xml version="1.0"?>
<methodCall>
  <methodName>execute</methodName>
  <params>
    <param>
      <!-- selecção da base de dados -->
      <value><string>may_9</string></value>
    </param>
    <param>
      <!-- uid do utilizador autenticado -->
      <value><int>1</int></value>
    </param>
    <param>
      <!-- login do utilizador autenticado -->
      <value><string>admin</string></value>
    </param>
    <param>
      <!-- identificação do objecto -->
      <value><string>generic.request</string></value>
    </param>
    <param>
      <!-- acção -->
      <value><string>read</string></value>
    </param>
  </params>
</methodCall>
```

```

    <param>
      <!-- lista de ids de pedidos a ler -->
      <array>
        <data> <int>1</int> </data>
        <data> <int>2</int></data>
        <data> <int>3</int> </data>
        <data> <int>4</int></data>
      </array>
    </param>
    <param>
      <!-- campos do pedido a devolver -->
      <array>
        <data><string>name</string></data>
        <data><string>request_description</string></data>
      </array>
    </param>
  </params>
</methodCall>

```

### XML-RPC Create Request

```

<?xml version='1.0'?>
<methodCall>
  <methodName>execute</methodName>
  <params>
    <param>
      <!-- selecção da base de dados -->
      <value><string>may_9</string></value>
    </param>
    <param>
      <!-- uid do utilizador autenticado -->
      <value><int>1</int></value>
    </param>
    <param>
      <!-- login do utilizador autenticado -->
      <value><string>admin</string></value>
    </param>
    <param>
      <!-- identificação do objecto -->
      <value><string>generic.request</string></value>
    </param>
    <param>
      <!-- definição da acção -->
      <value><string>create</string></value>
    </param>
    <param>
      <!-- valores a introduzir -->
      <value>
        <!-- indicação dos nome dos campos, identificação do tipo de variável e definição do valor da mesma -->
        <struct>
          <member>
            <name>name</name>
            <value><string>Pedido via JMeter</string></value>
          </member>
          <member>
            <name>subject_type</name>
            <value><int>1</int></value>
          </member>
        </struct>
      </value>
    </param>
  </params>
</methodCall>

```

```

        <member>
            <name>responsible_name</name>
            <value><int>4</int></value>
        </member>
        <member>
            <name>request_description</name>
            <value><string>Desc</string></value>
        </member>
        <member>
            <name>stating_reasons</name>
            <value><string>stating reasons</string></value>
        </member>
        <member>
            <name>expected_date</name>
            <value><string>08/31/2014</string></value>
        </member>
    </struct>
</value>
</param>
</params>
</methodCall>

```

Para cada um dos cenários testados é apresentado um resumo dos resultados do teste e um gráfico que apresenta os tempos de resposta do OpenERP aos pedidos ao longo do teste.

**Cenário 1:**

Número de utilizadores: 500

Ramp up: 100 (0.2 segundos entre o início de cada thread)

Acessos concorrentes: 5

| Label                    | # Samples | Average | Min | Max  | Std. Dev. | Error % | Throughput | KB/sec | Avg. B... |
|--------------------------|-----------|---------|-----|------|-----------|---------|------------|--------|-----------|
| Thread Group:XML-RPC ... | 500       | 40      | 11  | 1856 | 94,25     | 0,00%   | 5,0/sec    | 0,61   | 124,8     |
| Thread Group:XML-RPC ... | 500       | 35      | 10  | 610  | 46,53     | 0,00%   | 5,0/sec    | 5,98   | 1224,5    |
| Thread Group:XML-RPC ... | 500       | 253     | 112 | 2390 | 269,28    | 0,00%   | 5,0/sec    | 0,62   | 126,3     |
| TOTAL                    | 1500      | 109     | 10  | 2390 | 195,49    | 0,00%   | 14,9/sec   | 7,15   | 491,9     |

Figura 23 - Teste de carga - Cenário 1

Como se pode verificar na imagem acima, a totalidade dos 500 utilizadores efetuaram as suas ações com sucesso, sendo os tempos médios de resposta aos pedidos de 40 ms para o login, 35 ms para as leituras e 253 ms para a criação de pedidos.

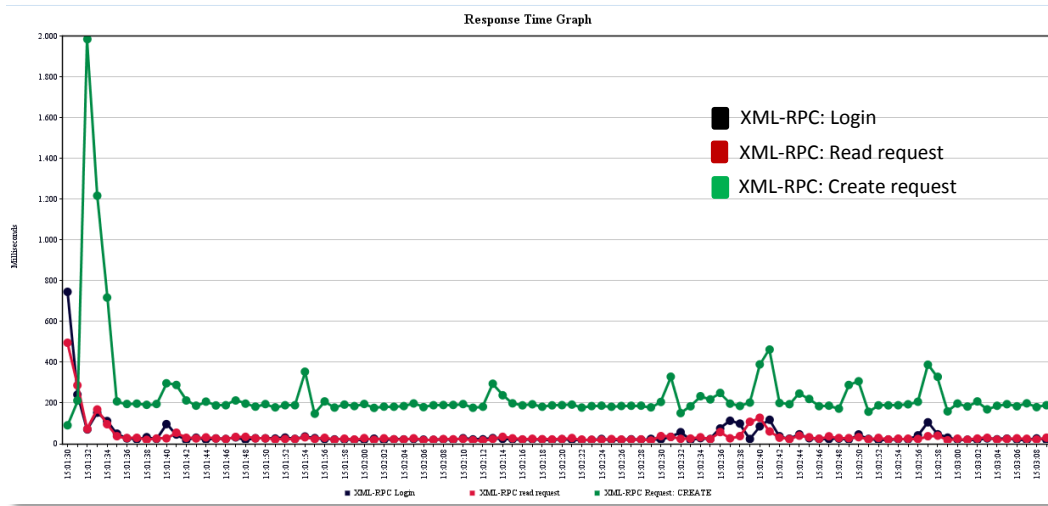


Figura 24 - Gráfico de tempo de resposta - Cenário 1

Os tempos de resposta mantiveram-se baixos ao longo de todo o teste. Os tempos mais elevados no início podem dever-se a algum processo que estivesse a ocupar recursos da máquina quando o teste teve início.

**Cenário 2:**

Número de utilizadores: 500

Ramp up: 50 (0.1 segundos entre o início de cada thread)

Acessos concorrentes: 10

**Summary Report**

Name: Summary Report

Comments:

Write results to file / Read from file

Filename:  Browse... Log/Display Only:  Errors  Successes

| Label                       | # Samples | Average | Min | Max   | Std. Dev. | Error % | Throughput | KB/sec | Avg. Byt... |
|-----------------------------|-----------|---------|-----|-------|-----------|---------|------------|--------|-------------|
| Thread Group:XML-RPC Log... | 500       | 2454    | 24  | 8055  | 1604,52   | 0,00%   | 8,7/sec    | 1,03   | 121,0       |
| Thread Group:XML-RPC rea... | 500       | 2650    | 24  | 7564  | 1698,89   | 0,00%   | 8,0/sec    | 9,66   | 1235,0      |
| Thread Group:XML-RPC Re...  | 500       | 5584    | 311 | 10975 | 2423,36   | 0,00%   | 7,1/sec    | 0,86   | 124,0       |
| TOTAL                       | 1500      | 3563    | 24  | 10975 | 2414,08   | 0,00%   | 21,2/sec   | 10,22  | 493,3       |

Figura 25 - Teste de carga - Cenário 2

Neste cenário, todos os utilizadores continuam a conseguir efetuar todas as ações com sucesso, aumentando no entanto os tempos de resposta médios para 2.45s para o login, 2.65 ms para a leitura do pedido e 5.58 ms para a criação de um novo pedido.

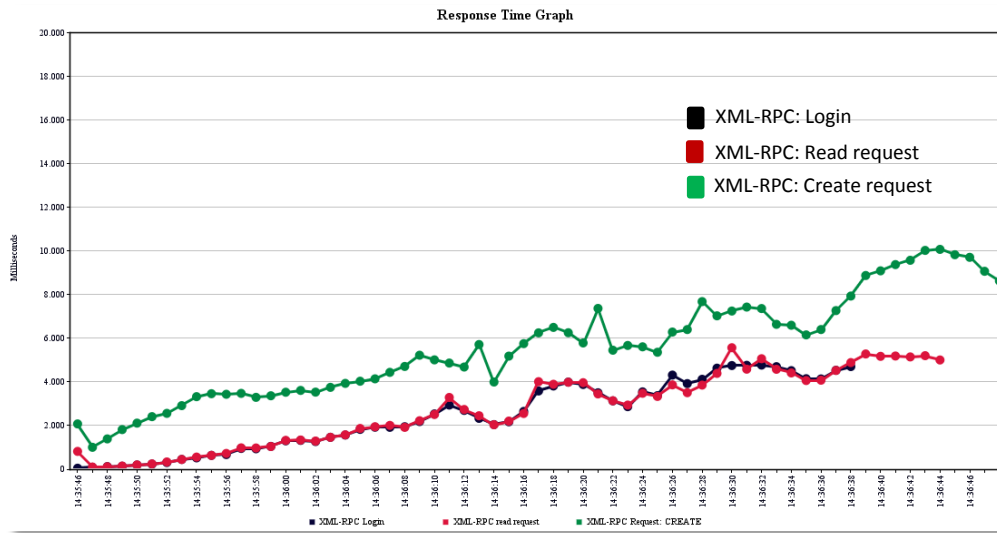


Figura 26 - Gráfico de tempo de resposta - Cenário 2

Na imagem acima, percebe-se que os tempos de resposta tiveram um aumento assinalável, à medida que a carga do servidor foi aumentando. Note-se que aumentando o tempo de resposta aumentam também o número de acessos concorrentes, tendo em conta que o tempo de inicialização de cada thread se mantém constante (uma thread a cada 0.1 segundos).

**Cenário 3:**

Número de utilizadores: 500

Ramp up: 10 (0.02 segundos entre o início de cada thread)

Acessos concorrentes: 50

**Summary Report**

Name: Summary Report

Comments:

Write results to file / Read from file

Filename:  Browse... Log/Display Only:  Errors  Successes

| Label                       | # Samples | Average | Min  | Max   | Std. Dev. | Error % | Throughput | KB/sec | Avg. Byt... |
|-----------------------------|-----------|---------|------|-------|-----------|---------|------------|--------|-------------|
| Thread Group:XML-RPC Log... | 500       | 6739    | 98   | 54332 | 7576,80   | 7,80%   | 8,1/sec    | 2,02   | 254,2       |
| Thread Group:XML-RPC rea... | 500       | 9400    | 130  | 48507 | 8150,33   | 6,80%   | 7,9/sec    | 9,84   | 1275,7      |
| Thread Group:XML-RPC Re...  | 500       | 14721   | 1659 | 57081 | 8213,78   | 8,60%   | 7,2/sec    | 1,90   | 271,0       |
| TOTAL                       | 1500      | 10287   | 98   | 57081 | 8647,57   | 7,73%   | 21,5/sec   | 12,60  | 600,3       |

Figura 27 - Teste de carga - Cenário 3

Neste cenário a performance do *OpenERP* foi posta seriamente posta em causa, havendo inclusive alguns utilizadores que não conseguiram efetuar as suas ações. O tempo de resposta médio também aumentou, passou para 6.73 s para o login (com taxa de erro de 7.8%), 9.4 s para a leitura de um pedido (com taxa de erro de 6.8%) e de 14.7 s para a criação de pedido (com taxa de erro de 8.6%).

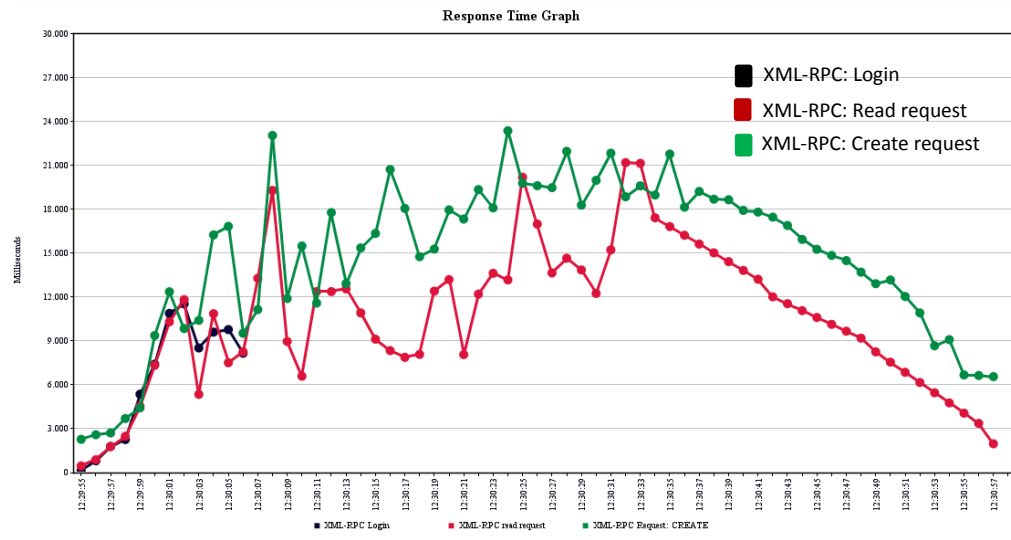


Figura 28 - Gráfico com tempo de resposta - Cenário 3

Neste cenário pode observar-se que, excetuando no início do teste (e também no final) os tempos de resposta com este nível de concorrência foram sempre bastante elevados.

### 6.1.2. Esforço desenvolvimento

Na última reunião com o membro da equipa do projeto *LUGUS*, foi pedido que fosse feita uma estimativa dos tempos para o desenvolvimento, de raiz, do pedido genérico, usando a tecnologia *Agile* da *OutSystems*. A estimativa dada foi de sensivelmente 6 semanas, incluindo uma semana para testes. No entanto, foi feita a ressalva que a mesma implementação demoraria mais duas semanas, se se assumisse que a pessoa responsável pela implementação do processo desconhecesse a tecnologia.

Ora, como se veio a perceber ao longo deste estágio, o período de aprendizagem do *OpenERP* é bastante mais longo do que as duas semanas estimadas para a aprendizagem e domínio da plataforma *OutSystems*, uma vez que a documentação de suporte ao *OpenERP* nem sempre é a adequada para as necessidades.

Aquando do planeamento das tarefas para o segundo semestre do estágio, foi feita uma estimativa otimista que previa um período de desenvolvimento para o pedido genérico de 4 semanas (incluindo testes). Ora esta estimativa veio mostrar-se completamente desajustada da realidade, tendo em conta que o desenvolvimento demorou cerca de 12 semanas (mais 4 semanas do que a pior estimativa de desenvolvimento para o mesmo módulo na plataforma *OutSystems*). Acrescendo do facto de a integração do *OpenERP* com o sistema externo de gestão de ficheiros não ter sido implementada.



Claro que, com o conhecimento adquirido ao longo deste período, o tempo de desenvolvimento seria agora reduzido. Com base nesse conhecimento, uma estimativa realista para o desenvolvimento deste mesmo módulo, a partir do nada seria de 7 a 8 semanas (incluindo testes). Ainda assim, portanto, superior à estimativa apresentada para a implementação em *OutSystems*. Ressalve-se que continua a não ser-se capaz de dar uma estimativa real para o tempo de desenvolvimento da integração do *OpenERP* com o serviço externo de *file hosting*.

## 7. Conclusões

Neste capítulo serão apresentadas algumas conclusões relativas ao cumprimento dos objetivos propostos, a forma como o estágio correu, considerações finais sobre a utilização da plataforma OpenERP e ainda conselhos para trabalho futuro.

### 7.1. Trabalhos realizados

No âmbito dos trabalhos que lhe foram atribuídos, o estagiário procedeu ao estudo da arte, estudando o funcionamento da solução existente e comparando as suas principais funcionalidades com as da plataforma a utilizar neste estágio, lendo para tal documentação diversa sobre uma e outra solução. Procurou ainda munir-se do conhecimento que lhe permitissem identificar as parcelas a ter em conta para fazer uma estimativa do custo total de propriedade da nova plataforma. Com base no conhecimento adquirido, o estagiário concluiu que apesar de a adoção de uma nova solução apresentar custos elevados, esse investimento aparenta ser menor do que os custos associados a uma eventual renegociação do contrato de licenciamento *OutSystems*.

Durante a segunda semana do estágio, adquiriu conhecimentos sobre o desenvolvimento em *OpenERP* participando numa formação contratada pelo GSIIC para os seus colaboradores. O estagiário acabou por ser um dos elementos mais participativos nesta formação, atingindo todos os objetivos traçados na mesma, tendo resolvido com sucesso todos os problemas propostos.

Integrou uma equipa formada dentro do seio da GSIIC-DSI para fazer o acompanhamento de entrada em produção de um módulo *PoS* do *OpenERP*, participando ativamente em diversas reuniões de especificação e de testes, dando mesmo algumas sugestões para a resolução de alguns problemas detetados. Esta integração permitiu aprofundar algum conhecimento adquirido na formação, nomeadamente no campo do *reporting*, chegando a proceder à alteração de alguns dos relatórios fornecidos e criar novos relatórios. Enquanto membro desta equipa de acompanhamento, o estagiário produziu documentação de suporte ao módulo em questão, juntamente com outro elemento desta equipa.

Paralelamente a esta atividade, foi feito um estudo teórico da plataforma, sendo feita instalação e configuração de um ambiente de testes, sendo testados alguns mecanismos de autenticação centralizada, segurança e auditoria oferecidos. Neste período foram identificados e corrigidos alguns bugs da aplicação. Foi produzida documentação sobre o processo de instalação e configuração do *OpenERP*.

No segundo semestre, o estagiário teve oportunidade de aplicar os conhecimentos adquiridos durante o primeiro semestre, conhecimentos que serviram sobretudo para este se familiarizar com

a ferramenta. Neste período criou um módulo para o OpenERP com vista à implementação dos processos identificados, aplicando os métodos habituais de engenharia para implementação de *software*. Procedendo a um levantamento dos requisitos, tendo para tal agendado reuniões com elemento do projeto *LUGUS* e feito uma análise da forma como o processo era implementado em *OutSystems*, com base neste estudo foi elaborada uma especificação funcional e uma especificação técnica para o primeiro processo identificado. Tendo em conta que o módulo foi implementado de raiz, estas especificações tiveram em conta todas as necessidades do projeto, desde a construção de modelo de dados do módulo, à disposição da informação e desenho dos interfaces de utilização.

Finda a implementação o estagiário procedeu a um estudo comparativo entre a solução implementada em *OpenERP* e a solução implementada em *OutSystems*, efetuou ainda uma comparação entre a apresentação dos dados, por parte de uma e outra soluções, e efetuou um teste de carga sobre o *OpenERP* para avaliar a capacidade de resposta deste servidor face a alturas de maior solicitação. Estas análises levaram o estagiário algumas conclusões, que serão expostas mais adiante, nas considerações finais.

Apesar de não ter conseguido implementar os dois processos a que se tinha proposto no início deste estágio, e não se tratando este de um estágio de desenvolvimento mas de um estudo da viabilidade de adoção de uma determinada solução com vista à substituição ou complemento de outra em utilização, o estagiário admite ter cumprido os seus objetivos. Este estágio permitiu estudar uma plataforma em ambiente de produção e identificar quais as suas potencialidades e limitações, sendo produzida documentação que permitirá ser o ponto de partida para equipa que vier a desenvolver soluções sobre esta plataforma.

O estágio permitiu ao estagiário consolidar os seus conhecimentos na linguagem *python* que praticamente desconhecia no início deste projeto e aprimorar a sua autonomia, uma vez que era o único constituinte da equipa deste projeto, não havendo dentro do seio da GSIIC-DSI nenhum elemento com conhecimentos de desenvolvimento em OpenERP.

## **7.2. Principais dificuldades encontradas**

Como foi dito acima, não foi possível implementar os dois processos inicialmente propostos. Este atraso deveu-se principalmente às dificuldades encontradas na implementação do primeiro processo, sobretudo por falta de documentação que sustentassem algumas ações um pouco mais “fora da caixa” e também por falta de feedback por parte da comunidade. Como foi apontado no capítulo deste relatório dedicado ao desenvolvimento, a documentação de apoio ao desenvolvimento é pouca, e a que existe está muitas vezes desatualizada, sendo para as versões 6.x do OpenERP. Também o feedback – não tanto em quantidade, mas sobretudo em qualidade - dado pela comunidade é

bastante abaixo do espectável.

A título de exemplo, foi necessário quase duas semanas de investigação, tentativa e erro, para perceber como seria possível criar registos automáticos na tabela de *tracing*, ligando estes registos ao pedido que lhe deu origem, na tabela *requests*, sempre que um utilizador efetua uma ação sobre o workflow desse pedido. E outro tanto tempo quando foi necessário registar o estado corrente de um pedido quando foi implementado a funcionalidade de pedido de parecer.

Sendo habitualmente este tipo de operações feito de forma automática pelo próprio OpenERP, além de não haver documentação sobre este tipo de implementação não há, nos fóruns da comunidade, quem aponte no sentido de solução ou arrisque sequer uma resposta.

Estes longos e frustrantes períodos de investigação levaram ao constante atraso no desenvolvimento e fizeram com que o estagiário se sentisse algumas vezes desmotivado. No entanto, o estagiário acabou por conseguir sempre implementar o comportamento esperado da funcionalidade.

A integração com uma plataforma externa de *file hosting* foi a exceção. Não foi possível encontrar qualquer documentação ou testemunho acerca de uma forma de integrar o OpenERP levando-o a guardar apenas metadados, sendo armazenamento do binário assegurado por um serviço externo. A ausência de documentação sobre este assunto leva o estagiário a crer que este tipo de integração só é possível colocando a lógica da integração do lado do cliente.

### **7.3. Considerações finais**

O *OpenERP* é uma ferramenta que provou ser versátil e que permite ser parametrizada de acordo com as necessidades mais específicas do cliente. No entanto, apesar de ser apresentado como uma ferramenta visual de desenvolvimento rápido, o *OpenERP* não o é, de facto. Esta plataforma fornece uma ferramenta visual – a *application builder* - que permite efetuar todas as parametrizações através de diversos interfaces gráficos. No entanto, esta solução não permite gerar código que permita guardar as alterações efetuadas visualmente. Por isso mesmo, se houver necessidade criar uma nova instância dos módulos desenvolvidos, ou reinstalar num novo servidor, ou até mesmo de reinstalar o módulo ou fazer uma atualização, todo o trabalho efetuado de parametrização feito através desse método visual poderá ser perdido e deverá ser feito novamente. Foi por esse motivo que, no decorrer deste estágio e mesmo arriscando não conseguir cumprir com o plano inicialmente traçado, se optou por desenvolver o módulo em questão recorrendo à escrita do código *python* para definir os modelos de dados e a lógica sobre esses modelos e ao *xml* para definir as vistas, menus, *workflow* e proceder ao carregamento automatizado de dados.

Também se pode verificar que a documentação de suporte a esta solução é escassa e, muitas vezes

até, inexistente. Também, o suporte dado pela comunidade é, muitas vezes, de pouca utilidade. E quando são dadas respostas pelos membros da comunidade são, quase sempre, tão vagas que levantam mais dúvidas ainda e raramente apontam para uma solução.

De salientar ainda o facto de que, durante o segundo semestre deste estágio, e sem qualquer aviso prévio aos seus clientes, a *OpenERP S.A.* mudou várias políticas relativas ao projeto *OpenERP*, nomeadamente o seu âmbito (que deixou de ser meramente um ERP, passando a ser também, entre outras coisas, *CMS* – nas vertentes de *Web Site Builder* e *Blog* - e plataforma de *e-commerce*). Esta mudança tem efeito em toda a imagem do produto e no próprio nome do projeto, que deixou de ser *OpenERP*, passando a adotar o nome *Odoo* em finais de maio de 2014. Esta é já a segunda vez que este tipo de viragem acontece desde a criação do projeto em 2005. O projeto tinha o nome *TinyERP* e mudou de nome e imagem aquando do lançamento da versão 5, em 2008, passando a adotar a denominação *OpenERP*. Este pode ser um sinal que a empresa proprietária não tem uma estratégia definida a longo prazo, podendo a qualquer altura descontinuar o produto.

Tendo tudo isto em conta é opinião do estagiário que esta ferramenta, apesar de tudo, tem um grande potencial e que por isso mesmo merece ser explorada no futuro. O *OpenERP* mostrou ter uma curva de aprendizagem bastante longa, pelo que os custos iniciais tidos com formação de recursos humanos poderão ser elevados. No entanto, ultrapassada essa fase inicial, e uma vez que a plataforma não tem custos de licenciamento, este investimento inicial em formação poderá ser amortizado com a sua utilização a longo prazo.

O facto de disponibilizar todas as suas funcionalidades através de serviços web leva a olhar para esta ferramenta com grande otimismo. Podendo ser criado um cliente web adaptado à imagem institucional da Universidade de Coimbra, integrando-o na zona privada dos utilizadores através da autenticação do Portal de Acessos da Universidade de Coimbra. A criação deste cliente, além de integrar o *OpenERP* na área privada do utilizador, disponibilizando-o como mais uma "*App UC.PT*", esta integração – que deverá obedecer às restrições descritas no ponto 3.9 deste documento - permitirá debelar o problema detetado aquando da implementação do processo de pedido genérico de integração com serviços externos, colocando a lógica de integração do lado do cliente.

## **7.4. Trabalho futuro**

No futuro, e se a Universidade de Coimbra vier a apostar nesta solução, é importante que seja dada uma sólida formação à equipa que vier a trabalhar neste projeto por forma a poder muni-la das ferramentas necessárias para o desenvolvimento de novas funcionalidades. Foi produzida documentação sobre a implementação de um módulo *OpenERP*, é aconselhada a sua leitura antes do início do processo de implementação.

É aconselhado que a equipa seja polivalente e que seja constituída por elementos com sólidos conhecimento na linguagem *python* e também com conhecimentos em *javascript* e *jQuery* e *Qweb* para o desenvolvimento de um cliente web independente do cliente web disponibilizado pelo *OpenERP* uma vez que esta solução mostrou ser indispensável para integração do OpenERP com sistemas externos.

## 8. Bibliografia

- [1] R. M. Stallman, *Free Software, Free Society: Selected Essays of Richard M. Stallman*, 1ª ed., GNU Press, 2002.
- [2] R. F. O. Badenes, H. G. Gómez e R. B. López, "Factores diferenciales entre los ERP de software libre (FSw ERP) y los ERP propietarios," *Dirección y Organización*, 2011.
- [3] R. West e S. L. Daigle, "Total Cost of Ownership: A strategic tool for ERP planning and implementation," *Educause - ECAR*, 2004.
- [4] New Zeland Government, "Total Cost of Ownership - An introduction to whole-of-life costing," Government Procurement Branch, Wellington, 2013.
- [5] M. Shaikh e T. Cornford, "Total cost of ownership of open source software: a report for the UK Cabinet Office supported by OpenForum Europe," London School of Economics and Political Science, London, 2011.
- [6] OutSystems S.A., "OutSystems Pricing, Licencing and Subscriptions," [Online]. Available: <http://www.outsystems.com/pricing-and-licensing>. [Acedido em 30 08 2014].
- [7] OpenERP S.A., "OpenERP Architecture," [Online]. Available: [https://doc.openerp.com/trunk/server/02\\_architecture/](https://doc.openerp.com/trunk/server/02_architecture/). [Acedido em 26 Janeiro 2014].
- [8] PostgreSQL Wiki, "Tuning your PostgreSQL server," [Online]. Available: [http://wiki.postgresql.org/wiki/Tuning\\_Your\\_PostgreSQL\\_Server](http://wiki.postgresql.org/wiki/Tuning_Your_PostgreSQL_Server). [Acedido em 26 Janeiro 2014].
- [9] OpenERP, "How to set up a regular PostgreSQL database backup," [Online]. Available: <http://help.openerp.com/question/4728/how-to-setup-a-regular-postgresql-database-backup/>. [Acedido em 25 Janeiro 2014].
- [10] D. Florêncio e C. Herley, "A Large-Scale Study of Web Password Habits," 2007. [Online]. Available: <http://research.microsoft.com/en-us/um/people/cormac/Papers/www2007.pdf>. [Acedido em 18 Janeiro 2014].
- [11] D. Hollingsworth, *The Workflow Reference Model*, Winchester, Hampshire: The Workflow Management Coalition, 1995.

[12] Licença Legal - OpenERP S.A., "Introduction to Licences," [Online]. Available: <https://doc.openerp.com/legal/license/>. [Acedido em 26 Janeiro 2014].



## **9. Anexos**

Anexo I – Planeamento de tarefas 1º e 2º semestre

Anexo II – Manual de Instalação OpenERP V7.0

Anexo III – Plano de testes PoS OpenERP

Anexo IV – Resultados e notas dos testes PoS OpenERP

Anexo V – Manual do Utilizador e Administrador do PoS OpenERP

Anexo VI – Manual de Configurações e Procedimentos PoS OpenERP

Anexo VII – Fluxogramas do pedido genérico e pedido de deslocação em serviços (documento fornecido pela equipa de desenvolvimento LUGUS)

Anexo VIII – Fluxograma do pedido genérico em notação BPMN

Anexo IX – Fluxograma do pedido de parecer em notação BPMN

Anexo X – Pedido genérico - Especificação funcional

Anexo XI – Pedido genérico – Especificação técnica

Anexo XII – Pedido genérico – Plano de testes

Anexo XIII – Pedido genérico – Resultado dos testes efetuados

Anexo XIV – Manual para o desenvolvimento de um módulo OpenERP

Anexo XV – Atas de reuniões de orientação