

Mestrado em Engenharia Informática
Dissertação/Estágio
Relatório Final

WISE – Web Information System Enterprise

Hugo Filipe Guedes Barrigas
hfguedes@student.dei.uc.pt

Orientador (DEI):
Prof. Dr. Paulo Rupino

Orientador (Critical Software):
Eng. Daniela Domingues

01 de Julho de 2014



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

Resumo:

O desenvolvimento de um sistema de informação é um processo contínuo, evoluindo juntamente com a organização ao longo do tempo. É comum serem necessárias reestruturações e novas implementações à medida que a organização evolui, retratando desta forma as alterações dos processos de negócio e novas necessidades da organização.

O estágio foi realizado no âmbito do projeto WISE, um sistema de informação empresarial desenvolvido pela Critical Software S.A. Os seus principais objetivos focam-se na implementação de um novo módulo de Recursos que integra com uma ferramenta externa através de Web Services, e ainda a reestruturação de um módulo existente, o módulo de Credenciações. Após um ciclo de desenvolvimento rigoroso, foram obtidas as soluções pretendidas contribuindo positivamente na gestão de recursos e credenciações.

Palavras-Chave:

Sistema de Informação, Hibernate, JDBC, J2EE, Integração de Aplicações Empresariais, Web Services, Recursos, Credenciações

Índice

1	Introdução.....	8
1.1.	Âmbito.....	8
1.2.	Motivação.....	9
1.3.	Objetivos do Estágio.....	10
1.4.	Estrutura do Documento.....	10
2	Estudo do Estado da Arte.....	12
2.1.	WISE – <i>Web Information System Enterprise</i>	12
2.2.	Módulos constituintes.....	13
2.2.1	Módulo de Recursos.....	14
2.2.2	Módulo de Credenciações.....	15
2.3.	Arquitetura.....	16
3	Gestão de Projeto.....	22
3.1.	Equipa.....	22
3.2.	Metodologia de Desenvolvimento.....	22
3.3.	Gestão de Riscos.....	24
3.4.	Planeamento.....	27
4	Desenvolvimento.....	31
4.1.	Módulo de Recursos.....	31
4.2.1	O meu Software.....	31
4.2.2	Lista de Software.....	32
4.2.3	Relatórios.....	34
4.2.4	Requisições de Software.....	36
4.2.5	Casos de Uso.....	44
4.2.6	Abordagem de Integração.....	47
4.2.7	Arquitetura.....	52
4.2.	Módulo de Credenciações.....	54
4.2.1	Credenciações Válidas.....	55
4.2.2	Credenciações Pendentes.....	56
4.2.3	Alertas.....	60
4.2.4	Pedido de Requisição.....	61
4.2.5	Relatórios.....	62

4.2.6	Gestão de Marca e Nível	65
4.2.7	Casos de Uso.....	66
4.2.8	Arquitetura.....	68
4.3.	Validação.....	70
5	Resultados do Desenvolvimento	71
5.1.	Módulo de Recursos	71
5.1.1	O Meu Software.....	72
5.1.2	Lista de Software	74
5.1.3	Relatórios	75
5.1.4	Requisições de Software	75
5.2.	Módulo de Credenciações	81
6	Conclusão.....	85
6.1.	Trabalho Futuro.....	85
Anexos	89

Índice de Tabelas

Tabela 1 – Tipos de Credenciação.....	15
Tabela 2 – Lista do meu Software.....	31
Tabela 3 – Libertar Licença.....	31
Tabela 4 – Notificação a DSI.....	32
Tabela 5 – Lista de Software.....	33
Tabela 6 – Filtros a aplicar na Lista de Software.....	33
Tabela 7 – Detalhes de um item da Lista de Software.....	33
Tabela 8 – Requisição de Software a partir de um item da lista de software.....	33
Tabela 9 – Relatório de Software.....	34
Tabela 10 – Filtros no relatório.....	35
Tabela 11 – Requisição de Software.....	36
Tabela 12 – Detalhes de Software.....	37
Tabela 13 – Detalhes de Aprovação.....	37
Tabela 14 – Secção de Utilização.....	37
Tabela 15 – Envio de Requisição por E-mail.....	38
Tabela 16 - Requisição de novo software.....	39
Tabela 17 – Detalhes de Requisição.....	40
Tabela 18 – Secção de Software.....	40
Tabela 19 – Secção de utilização.....	40
Tabela 20 – Detalhes de Aprovação.....	41
Tabela 21 – Mapa de estados.....	41
Tabela 22 – Requisições pendentes de Projeto.....	42
Tabela 23 – Requisições pendentes de Área.....	42
Tabela 24 – Requisições pendentes Gerais.....	42
Tabela 25 – Requisições pendentes do Colaborador.....	43
Tabela 26 – Hardware do colaborador.....	43
Tabela 27 - Funcionalidades do Web Service.....	48
Tabela 28 – Credenciações Válidas.....	55
Tabela 29 – Filtros da lista de Credenciações Válidas.....	56
Tabela 30 – Lista de Credenciações Pendentes.....	57
Tabela 31 – Rejeitar Credenciação Pendente.....	57
Tabela 32 – Filtros aplicáveis na lista de Credenciações Pendentes.....	58
Tabela 33 – Detalhes do processo de uma Credenciação Pendentes.....	59
Tabela 34 – Mapa de Estados do processo de Credenciação Pendente.....	59
Tabela 35 – Novos alertas do módulo de Credenciações.....	60
Tabela 36 – Detalhes dos alertas.....	60
Tabela 37 – Pedido de Credenciação.....	61
Tabela 38 – Secção Colaboradores do pedido de Credenciação.....	61
Tabela 39 – Secção Propriedades do pedido de Credenciação.....	61
Tabela 40 – Relatórios de Credenciações.....	62
Tabela 41 – Filtros a aplicar nos relatórios de Credenciações.....	63
Tabela 42 – Renovar Credenciação nos relatórios.....	63
Tabela 43 – Cancelar Credenciações nos relatórios.....	63
Tabela 44 – Visualizar histórico de Credenciações.....	63

Tabela 45 – Histórico de Credenciações.....	64
Tabela 46 - Gestão de Marca e Nível	65

Índice de Figuras

Figura 1 – Módulos mais relevantes do Sistema de Informação	13
Figura 2 – Arquitetura WISE – Nível 1	16
Figura 3 – Arquitetura WISE – nível 2.....	17
Figura 4 – Arquitetura WISE – Nível 3	18
Figura 5 – Mapeamento entre Evento e método.....	19
Figura 6- Utilização de tags Stripes	19
Figura 7 – Anotações Stripes para Validações	20
Figura 8 – Exemplos <i>@WICCoreEJB</i>	20
Figura 9 – Entidade com anotações Hibernate.....	21
Figura 10 – Fases de desenvolvimento - Waterfall.....	23
Figura 11 – Ciclo de vida da Gestão de Riscos	24
Figura 12 – Matriz de Risco	25
Figura 13 – Riscos mais significativos do projeto	26
Figura 14 – Diagram Gantt geral do projeto	27
Figura 15 – Diagrama Gantt do módulo de Credenciações.....	28
Figura 16 - Diagrama Gantt do módulo de Recursos	29
Figura 17 – Software do utilizador	32
Figura 18 – Lista de Software	34
Figura 19 – Relatórios de Software	35
Figura 20 – Processo de Requisição de Software	36
Figura 21 – Requisição de Software.....	38
Figura 22 – Requisição de novo software.....	39
Figura 23 – Detalhes de requisição	41
Figura 24 – Requisições pendentes.....	43
Figura 25 – Hardware do Colaborador	44
Figura 26 – Diagrama de casos de uso - <i>wisuserrole</i>	45
Figura 27 – Diagrama de casos de uso - Aprovadores.....	46
Figura 28 – Abordagem de integração	47
Figura 29 – Tecnologias utilizadas no WISE, GLPI, Camada de integração	49
Figura 30 – Exemplo WSDL para o método <i>getComputerInformationById</i>	50
Figura 31 – Arquitetura do módulo de Recursos	51
Figura 32 – Exemplo anotação <i>@EJB</i>	52
Figura 33 – Exemplos de utilização do Web Service	53
Figura 34 – Modelo ER do módulo Recursos.....	53
Figura 35 - Lista de Credenciações Válidas.....	55
Figura 36 - Filtros da lista de Credenciações Válidas	56
Figura 37 – Lista de Credenciações Pendentes	57
Figura 38 – Filtros aplicáveis na lista de Credenciações Pendentes.....	58
Figura 39 - Detalhes de uma Credenciação Pendente	59
Figura 40 – Pedido de Credenciação.....	62
Figura 41 – Relatórios de Credenciações.....	64
Figura 42 – Filtros aplicáveis nos relatórios	64
Figura 43 – Histórico de Credenciações	65
Figura 44 – Casos de uso <i>wisuserrole</i> , <i>wisclearancerole</i> , <i>wiserhrole</i> e <i>wisemrole</i>	66
Figura 45 – Casos de uso <i>isPM</i> , <i>wisemanagerrole</i> , <i>wisuserrole</i>	67

Figura 46 – Arquitetura do módulo de Credenciações.....	68
Figura 47 – Modelo ER antigo do módulo de Credenciações.....	69
Figura 48 – Novo modelo ER do módulo de Credenciações.....	69
Figura 49 – Página inicial do módulo de Recursos	71
Figura 50 – Diagrama de sequência para o Meu Software.....	72
Figura 51 – Exemplo de mensagens SOAP – <i>getLicenseInformation(Id)</i>	73
Figura 52 – Lista de Software – Solução final.....	74
Figura 53 – Exemplo de relatório.....	75
Figura 54 – Secção de Requisição – Solução final.....	76
Figura 55 – Diagrama de sequência para requisição de Software.....	76
Figura 56 – Histórico de requisições – Solução final.....	77
Figura 57 – Detalhes de requisição – Solução final.....	77
Figura 58 – Lista de requisições pendentes para Projeto	78
Figura 59 – Detalhes de uma requisição pendente - Aprovadores	78
Figura 60 – Diagrama de sequência para aprovação de requisição.....	79
Figura 61 – Diagrama de sequência para processo de sincronização.....	80
Figura 62 – Processo de Requisição completo.....	80
Figura 63 – Página inicial do módulo de Credenciações	81
Figura 64 – Página de credenciações pendentes alterada	81
Figura 65 – Página detalhes de credenciação pendente.....	82
Figura 66 – Secção de relatórios.....	83
Figura 67 – Histórico de Credenciações	84
Figura 68 – Secção de administração – Solução final.....	84

1 Introdução

Este documento apresenta o trabalho desenvolvido por Hugo Filipe Guedes Barrigas, no âmbito da sua Dissertação de Mestrado em Engenharia Informática, no Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

O estágio realizado encontrava-se dividido em duas partes. A primeira, que decorreu no primeiro semestre, focou-se em vertentes de planeamento e investigação, tendo como objetivos a definição de um plano de atividades concretas e ordenadas, a realização do estudo do estado da arte e a especificação do produto a ser desenvolvido. A segunda parte, focando-se em aspetos de desenvolvimento, decorreu durante o 2º semestre e envolveu a implementação do produto e respetiva validação.

O estágio teve início no dia 10 de Setembro de 2013 e está previsto terminar no dia 9 de Julho de 2014. A orientação do mesmo esteve a cargo da Eng. Daniela Domingues da Critical Software S.A. e do Dr. Paulo Rupino da Universidade de Coimbra.

1.1. Âmbito

O crescimento interno de uma empresa implica alterações a nível organizacional e estrutural, estando constantemente sujeito a novos desafios. A gestão de recursos e processos internos, apoio a decisões, mercados de software que implicam áreas de desenvolvimento específicas, garantir a aplicação de processos de qualidade e, suportar métodos de avaliação de desempenho ágeis, são alguns exemplos.

Perante esta situação, surge a necessidade de uma solução capaz de integrar todos os níveis organizacionais, de forma a centralizar a informação e garantir canais de comunicação flexíveis. A solução que apresenta as características necessárias é um *Sistema de informação empresarial*.

De forma geral, o termo é utilizado para referir não só a informação e meios de comunicação utilizados pela organização, mas também uma forma em que os colaboradores interagem com esta solução como suporte aos processos de negócio. O seu principal objetivo é agilizar a operação diária da organização, fornecendo suporte aos processos de negócio essenciais ao bom desempenho e sucesso da organização.

Assim, o estágio encontra-se no âmbito do projeto “WISE - *Web Information System Enterprise*”, um sistema de informação empresarial desenvolvido pela **Critical Software S.A.**

1.2. Motivação

A Critical Software, investiu no desenvolvimento de um sistema de informação Empresarial com o objetivo de fazer frente às crescentes necessidades de centralização e uniformização da informação.

O WISE, como sistema de informação empresarial, pretende agilizar o desempenho da Organização através da automatização dos processos de negócio, tornando-os, assim, mais eficientes [1]. Esta característica permite não só, reduzir o tempo de execução dos processos, mas também a probabilidade de erros durante a sua execução. Com isto podemos concluir que um sistema de informação é um elemento fundamental numa organização, provando ser uma vantagem estratégica a nível de negócio e organizacional.

Atualmente vários processos de negócio são executados através do WISE, mas a presença de lacunas é um cenário comum. Perante estas lacunas, certos módulos são incapazes de realizar o objetivo para os quais foram desenvolvidos. Na realidade, o que por vezes acontece, é que contribuem para o atraso dos processos e realização das respetivas atividades.

Devido às dificuldades apresentadas aos utilizadores, atrasos e ineficácia, os colaboradores optam por não utilizar esta ferramenta ou recorrem mesmo a ferramentas externas para realizar as suas tarefas. Esta situação é prejudicial para a empresa, contribuindo para despesas desnecessárias e módulos integrados no sistema que são obsoletos.

Tendo em conta o cenário aqui apresentado, surge a necessidade de (re)implementar dois módulos de forma a contribuírem positivamente para o funcionamento do sistema e da empresa. Desta forma, espera-se que as soluções definidas neste documento, auxiliem os colaboradores nas suas atividades diárias.

1.3. Objetivos do Estágio

O estágio curricular apresenta ao estagiário, ainda aluno, desafios complexos num ambiente profissional. Trata-se da transição do mundo académico para o profissional, onde o último é um ambiente extremamente competitivo e exigente. Pretende-se providenciar ao estagiário novos conhecimentos e, a oportunidade de solidificar conhecimentos previamente adquiridos durante o curso, na presença de ambientes maduros e profissionais seniores. Espera-se que o desafio seja resolvido através da aplicação de boas práticas e metodologias de engenharia, evidenciando capacidade de adaptação e raciocínio.

Assim, o presente estágio foca-se na reestruturação do módulo de Credenciações e uma nova implementação do módulo de Recursos. Os principais objetivos apresentados podem ser divididos em vários objetivos de menor dimensão. O primeiro refere-se ao planeamento e especificação das soluções esperadas, envolvendo os requisitos pretendidos, o seu aspeto gráfico, respetiva arquitetura e permissões de acesso. Com base nesta especificação, é possível definir normas e uma sequência lógica de atividades que devem ser seguidas na sua implementação. O segundo objetivo diz respeito à implementação das soluções definidas. É esperado que realizem todos os requisitos especificados e o seu aspeto visual deve seguir os padrões utilizados nos restantes módulos, contribuindo para um sistema coerente.

Outro objetivo é garantir que as soluções implementadas são robustas e tolerantes a falhas, evitando interrupções na experiência do utilizador. Um sistema robusto e tolerante a falhas significa suportar e tratar erros, reagindo em cenários de informação anómala introduzida pelo utilizador não comprometendo a restante informação e funcionamento do sistema.

A validação das soluções implementadas constitui outro objetivo. Para o efeito foram especificados e realizados testes de validação que garantem o seu funcionamento correto e esperado.

Por fim, o último objetivo inclui a integração de uma ferramenta no sistema de informação, de forma a partilhar informação útil e necessária ao WISE. Para garantir uma integração eficiente, é necessário definir uma arquitetura robusta e uma abordagem que minimize as situações de inconsistência na informação.

1.4. Estrutura do Documento

Esta secção tem como objetivo dar ao leitor uma breve descrição dos capítulos existentes e a estrutura do documento apresentado.

- No **capítulo 1** é feita uma introdução ao estágio, o âmbito em que se enquadra, a motivação que o originou e ainda os objetivos traçados para o mesmo.
- No **capítulo 2** é apresentado o estudo do estado da arte do sistema de informação, contextualizando o leitor no âmbito do projeto apresentando a sua arquitetura, respetivas tecnologias e os módulos que o constituem.
- O **capítulo 3** é dedicado à gestão de projeto. Ao longo das secções é descrita a metodologia de desenvolvimento adotada, a gestão de riscos

utilizada para identificar e mitigar os riscos que surgiram e, é apresentado o planeamento do projeto.

- O **capítulo 4** descreve o trabalho realizado ao longo do desenvolvimento do projeto. Encontra-se dividido em várias secções onde são apresentados os principais requisitos definidos para cada módulo, os casos de uso, as respetivas arquiteturas e, a abordagem utilizada na integração da ferramenta no sistema.
- No **capítulo 5** são apresentados os resultados obtidos. Descrevem-se as ações que o utilizador pode realizar no sistema e explicadas algumas decisões de design e implementação.
- No **capítulo 6**, são apresentadas as conclusões acerca do trabalho desenvolvido, bem como algumas sugestões que podem melhorar as soluções implementadas.

2 Estudo do Estado da Arte

Neste projeto em particular, o estado da arte indica o grau de desenvolvimento atingido até ao momento e o estado atual em que se encontra o sistema de informação.

2.1. WISE – *Web Information System Enterprise*

O processo de desenvolvimento de um sistema de informação envolve um conhecimento profundo da organização. É necessário ter uma visão clara de quais os objetivos pretendidos da solução, quais os processos de negócio que são relevantes retratar no sistema e, como serão disponibilizados ou executados. Inicialmente, um sistema de informação não é desenvolvido tendo em conta uma visão global da organização, mas sim uma fração de interesse em otimizar, automatizar ou, disponibilizar informação específica de forma fácil e eficaz. É um processo longo e incremental, onde são adicionadas novas funcionalidades à medida que o projeto cresce e evolui com o amadurecimento da organização e respetivos processos de negócio. Para que fique claro ao leitor, quando se fala em processos de negócio refere-se ao conjunto de atividades, sujeitas a regras ou procedimentos, que executadas produzem o resultado esperado pelo processo.

O WISE, sistema de informação utilizado atualmente na Critical Software S.A, é uma solução organizacional e de gestão desenvolvida internamente. Sendo uma parte integral da organização, desempenha um papel fundamental no dia-a-dia laboral dos colaboradores. Fornece suporte à tomada de decisões, o que permite obter decisões mais rápidas e precisas, aumenta a quantidade de informação disponibilizada em todos os departamentos, proporciona acessos flexíveis à informação disponibilizada, agiliza a execução dos processos de negócio e, facilita a sinergia entre os diferentes departamentos, *spin-offs* e subsidiárias da empresa. Seguindo uma abordagem modular, é constituído por vários módulos independentes que foram desenvolvidos com o intuito de resolver problemas concretos e específicos. Esta abordagem permitiu a construção de uma solução complexa com base em pequenos subsistemas que, embora desenvolvidos individualmente, operam como um conjunto integrado.

A razão pela qual a Critical Software optou por implementar um sistema de informação e evitar soluções comerciais (e.g: SAP), resume-se essencialmente à redução de custos. Foi realizado um estudo pela empresa onde foram exploradas as várias vantagens e desvantagens associadas à implementação de uma solução *in-house*. Com o estudo concluiu-se que o facto de optar por uma solução implementada de raiz permite, não só, obter um sistema de informação completamente personalizado à organização, evitando funcionalidades desnecessárias, mas também, a possibilidade de melhorar e alterar continuamente o sistema de informação, expandindo-o sempre que necessário sem ter que recorrer a entidades externas e ainda, a facilidade de manutenção da solução.

2.2. Módulos constituintes

Como consequência da evolução da organização emergiu a necessidade de centralizar informação acerca dos colaboradores. Com base nesta necessidade nasceu o WISE. Inicialmente constituído por funcionalidades básicas como a listagem de colaboradores, foi evoluindo até à versão atual composta por um vasto conjunto de funcionalidades distribuídas entre módulos e sub-módulos.

Sendo um dos grandes objetivos facultar suporte ao desenvolvimento de software, o módulo de projectos é desta forma o mais relevante na solução, juntamente com o módulo de gestão de recursos da empresa [1]. Apesar da gestão de projectos e recursos ser a base de funcionamento do WISE, este é muito mais vasto em termos de funcionalidades. Na Figura 1 são apresentados os módulos mais relevantes do sistema.

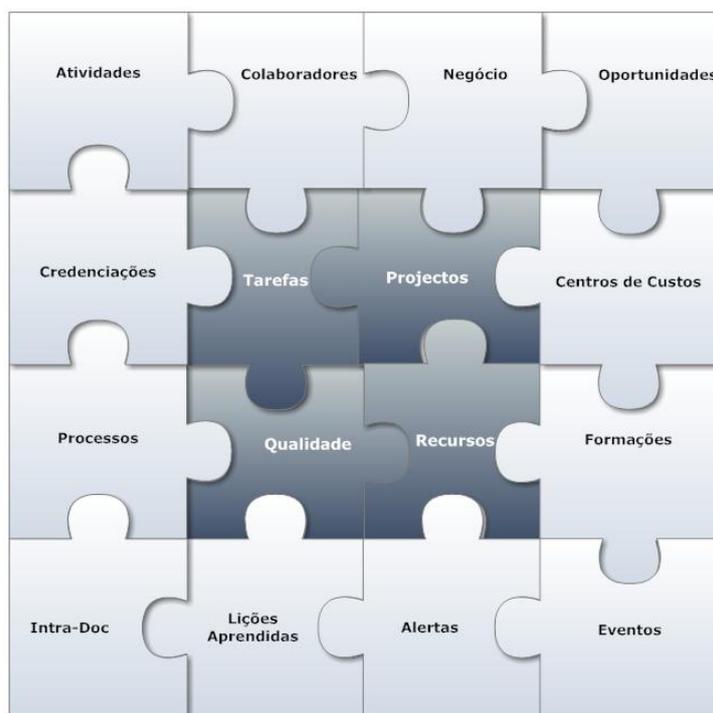


Figura 1 – Módulos mais relevantes do Sistema de Informação

Com a figura pretende-se transmitir a ideia de considerar o WISE como um quebra-cabeças constituído por várias peças que se complementam, os módulos. Combinando os vários módulos é possível obter um sistema de informação completo e robusto, capaz de resolver todas as necessidades da empresa.

No centro da imagem encontram-se os módulos mais relevantes; o módulo Tarefas permite alocar diariamente o esforço dos colaboradores nas respetivas atividades. Relaciona-se diretamente com o módulo Projetos através da informação utilizada nas várias tarefas de cada projeto. O módulo Qualidade assegura a qualidade dos produtos e projetos através da disponibilidade dos diversos documentos de qualidade. Por exemplo, garante acesso fácil e rápido aos vários processos através de listagens específicas e orientadas às funções dos colaboradores. Faculta também a funcionalidade de introduzir e registar documentos com identificadores únicos que, seguindo uma norma, contêm informação relevante ao âmbito do documento e permitem identificá-los de forma única

na empresa (Intra-Doc). De forma a ter uma aprendizagem contínua, no módulo Lições Aprendidas é possível registar observações úteis para projetos futuros ou melhorias para a infraestrutura da empresa.

O módulo Projetos tem como principal objetivo suportar os utilizadores na gestão e controlo dos seus projetos. Encontra-se integrado com aplicações externas ao sistema, como por exemplo o JIRA, que permite realizar a gestão de problemas no software, ou SVN e CVS, que permitem realizar controlo de versões [2]. O módulo de Negócio permite aos gestores avaliar o custo de cada projeto e identificar possíveis gastos desnecessários.

No anexo **A**, o leitor poderá encontrar uma breve descrição dos restantes módulos que constituem o WISE e que não foram apresentados. De seguida são detalhados os módulos alvo do estágio, Recursos e Credenciações.

2.2.1 Módulo de Recursos

A gestão de recursos é vital no seio de qualquer empresa, sendo necessário garantir que os colaboradores têm não só acesso ao material essencial para realizar as suas tarefas com sucesso e eficácia, mas também que a sua atribuição e gestão é realizada com a maior simplicidade e celeridade possível. Podemos encarar a gestão de recursos como o conjunto de atividades que têm como finalidade assegurar o fornecimento de ferramentas necessárias ao bom funcionamento da empresa, numa janela temporal aceitável, na quantidade necessária, qualidade requerida e, como seria expectável, pelo melhor preço possível. Para atingir este objetivo, são necessárias ferramentas que auxiliam no seu planeamento e controlo, bem como, utilizar processos de negócio que maximizam o seu fluxo entre os vários departamentos da organização.

O atual módulo de Recursos que integra no WISE inclui um vasto conjunto de funcionalidades. Permitem gerir e requisitar software, hardware e licenças, bem como consultar os recursos disponíveis. Embora o módulo contenha estas funcionalidades, revelou ser incapaz de resolver as crescentes necessidades do departamento de sistemas informáticos (DSI). Com base na informação retirada ao longo das reuniões com os responsáveis do departamento, concluiu-se que o módulo atual contém diversas lacunas a nível de informação, nomeadamente a falta de campos informativos, a presença de campos informativos irrelevantes ao utilizador e, a falta de um mecanismo que possibilita ao utilizador monitorizar as suas requisições a qualquer momento. Acrescentando às lacunas de informação, temos processos de negócio desatualizados como o processo de requisição de hardware retratado no sub-módulo equipamentos ou, funcionalidades que não cumprem o devido objetivo, como grande parte das funcionalidades de gestão que não retratam os processos necessários e não contêm informação essencial. Ainda, o aspeto visual do módulo é precário devido à sua implementação em tecnologias antigas que se tornaram obsoletas com a evolução do projeto.

Como resultado dos problemas identificados, o DSI foi forçado a utilizar uma nova ferramenta capaz de resolver todas as suas necessidades, o GLPI. Com esta nova ferramenta a gestão de recursos torna-se mais ágil e flexível, proporcionando todas as funcionalidades e informação essenciais.

Na presença de um módulo com lacunas e o DSI forçado a utilizar uma nova ferramenta, surgiu a necessidade de implementar um novo módulo de recursos que resolvesse os problemas identificados e que integrasse com a nova ferramenta para centralizar a informação.

2.2.2 Módulo de Credenciações

Para trabalhar em projetos confidenciais ou que requerem um grau de segurança superior ao normal, as organizações e os seus colaboradores precisam de possuir credenciações de segurança. Uma credenciação torna uma empresa elegível ao acesso de informação confidencial, garantindo ao cliente confiança e o nível de segurança necessário para desenvolver o projeto num ambiente rigoroso, controlado e protegido contra fugas de informação. A Tabela 1 contém os diferentes tipos de credenciações associadas a diferentes graus de segurança [3][4]. É importante salientar que o governo nacional é o único agente responsável por emitir credenciações.

Nome	Classificação
Confidencial	“A fuga de informação pode causar algum dano à segurança nacional”
Secreto	“A fuga de informação pode causar danos graves à segurança nacional”
Muito Secreto	“A fuga de informação pode causar danos muito graves à segurança nacional”

Tabela 1 – Tipos de Credenciação

A Critical Software e os seus colaboradores têm credenciações de segurança atribuídas pelo **GNS**¹ que promovem um ambiente seguro na organização, controlando os acessos a espaços privilegiados e garantindo a integridade de informação em projetos com elevada confidencialidade. O aumento gradual do número de colaboradores credenciados deu origem à necessidade de uma nova solução eficaz na gestão de credenciações. Foram pensadas soluções com diversos sistemas diferentes, no entanto, cada um tinha limitações e não satisfaziam as necessidades dos utilizadores. Pela flexibilidade da plataforma WISE, uma integração mais eficaz e manter a centralização da informação, a construção de um módulo de Gestão de Credenciações no WISE surge como a solução mais eficiente [5].

O módulo atual contém limitações que impossibilitam a sua utilização pelos responsáveis da gestão de credenciações. Estas limitações focam-se na falta de campos informativos necessários ao utilizador, definições de acesso ao módulo e funcionalidades mal configuradas, e incompleto no que toca ao conjunto de funcionalidades fornecidas. Nos capítulos 4 e 5 serão apresentadas com detalhe as alterações realizadas no módulo atual.

¹GNS - Gabinete Nacional de Segurança

2.3. Arquitetura

Como qualquer estrutura complexa, é essencial que o software seja construído sobre uma base sólida. A modelação arquitetural de um sistema é um processo rigoroso e ponderado, sendo necessário considerar requisitos chave e projetar refletindo em possíveis consequências a longo prazo. O descuido nestas atividades pode colocar a aplicação em risco, dando origem num software instável, incapaz de concretizar os requisitos pretendidos ou difícil de gerir num ambiente de produção [6].

Ao longo desta secção pretende-se apresentar a arquitetura do WISE recorrendo a diferentes níveis de profundidade, começando pelas várias ferramentas que integram o sistema contribuindo para um ambiente partilhado que centraliza toda a informação, passando posteriormente para o modelo arquitetural, e finalizando com as diferentes tecnologias utilizadas. De seguida, na Figura 2, é ilustrado o primeiro nível da arquitetura.

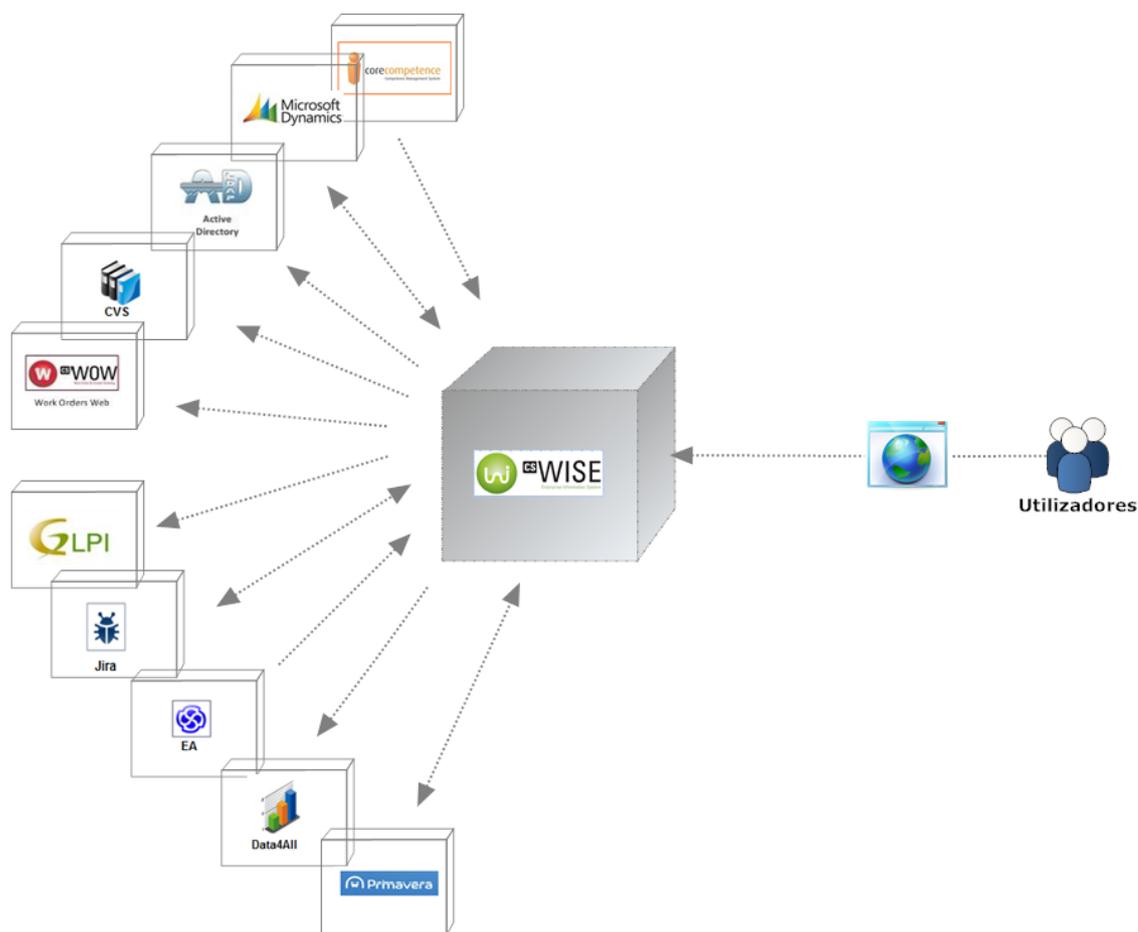


Figura 2 – Arquitetura WISE – Nível 1

Considerando o WISE como um sistema caixa preta, encontram-se na Figura 4 as várias aplicações que integram o sistema. Com vista a otimizar os seus processos de negócio, as diversas aplicações da organização devem ser capazes de interagir entre si partilhando informação e complementando-se mutuamente. Tipicamente, estas aplicações são implementadas nas mais variadas linguagens, completamente independentes, impossibilitando a comunicação entre si. Para ultrapassar esta limitação o WISE utiliza

dois métodos de integração, *Web Services*² e *Data Transformation Services*³. Estes serviços fornecem informação útil ao sistema, como por exemplo, informação relevante à alocação de esforço e tarefas existentes no JIRA, gestão e controlo de versões com o CVS e SVN, sincronização de informação das propostas e oportunidades com o Microsoft Dynamics, validação de credenciais dos colaboradores com o LDAP ou, informação acerca dos diversos recursos com o GLPI.

Aumentando o nível de profundidade, na Figura 3 é apresentado o modelo arquitetural.

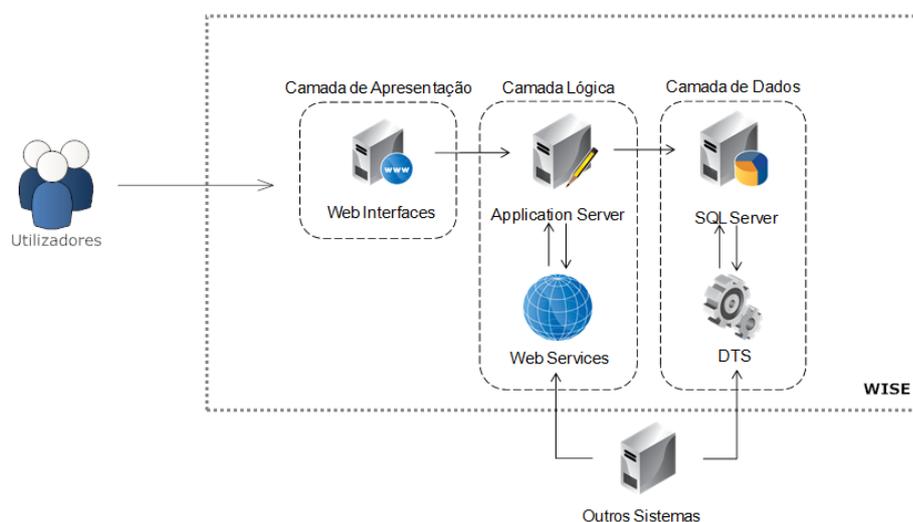


Figura 3 – Arquitetura WISE – nível 2

O modelo arquitetural do apresentado segue uma abordagem típica de três camadas, Apresentação, Lógica e Dados. Este modelo promove o desenvolvimento independente de cada camada, permitindo desta forma uma clara separação lógica. Com a utilização deste modelo encontram-se associadas um conjunto de vantagens [7][8]:

- **Escalabilidade:** Caso sejam necessárias mais camadas lógicas para garantir boa performance, bastará adicionar novas instâncias desta camada (e.g. novos servidores);
- **Fiabilidade:** Com múltiplas instâncias da camada lógica torna-se mais fácil aumentar a fiabilidade do sistema através de vários níveis de redundância;
- **Manutenção:** Com a separação lógica das camadas e devido à sua natureza independente, as alterações efetuadas numa camada não implicam necessariamente alterações nas restantes.
- **Reutilização:** Separar a aplicação em várias camadas maximiza a reutilização. Por exemplo, a mesma camada lógica pode ser utilizada por diferentes camadas de apresentação.

No que respeita ao papel que cada camada desempenha, a camada de apresentação é o primeiro nível do modelo, sendo o canal de comunicação entre o utilizador e o sistema. O

² **Web Services** - Mecanismos que permitem a interoperabilidade entre aplicações desenvolvidas em diferentes plataformas e linguagens. São caracterizados pela agilidade que trazem aos processos de negócio e eficácia na comunicação entre aplicações, tornando-se atrativos pela utilização de tecnologias padrão como o protocolo HTTP e XML [34].

³ **Data Transformation Services** -conjunto de objetos e serviços que permitem automatizar operações *Extract, Transform, Load* entre bases de dados.

seu objetivo é apresentar a informação ao utilizador e interpretar os seus pedidos, processando os dados introduzidos antes de serem enviados para a camada subjacente, lógica. Esta camada pode ser considerada o núcleo da aplicação e é responsável por todas as operações e processamento de informação. Por último, a camada de dados é responsável por armazenar toda a informação do sistema, sendo comum a utilização de base de dados para o efeito.

Atingimos assim o último nível de profundidade da arquitetura, Figura 4, onde são apresentados os vários componentes de cada camada e como interagem entre si. Encontram-se também presentes as tecnologias utilizadas em cada camada e respetivos componentes.

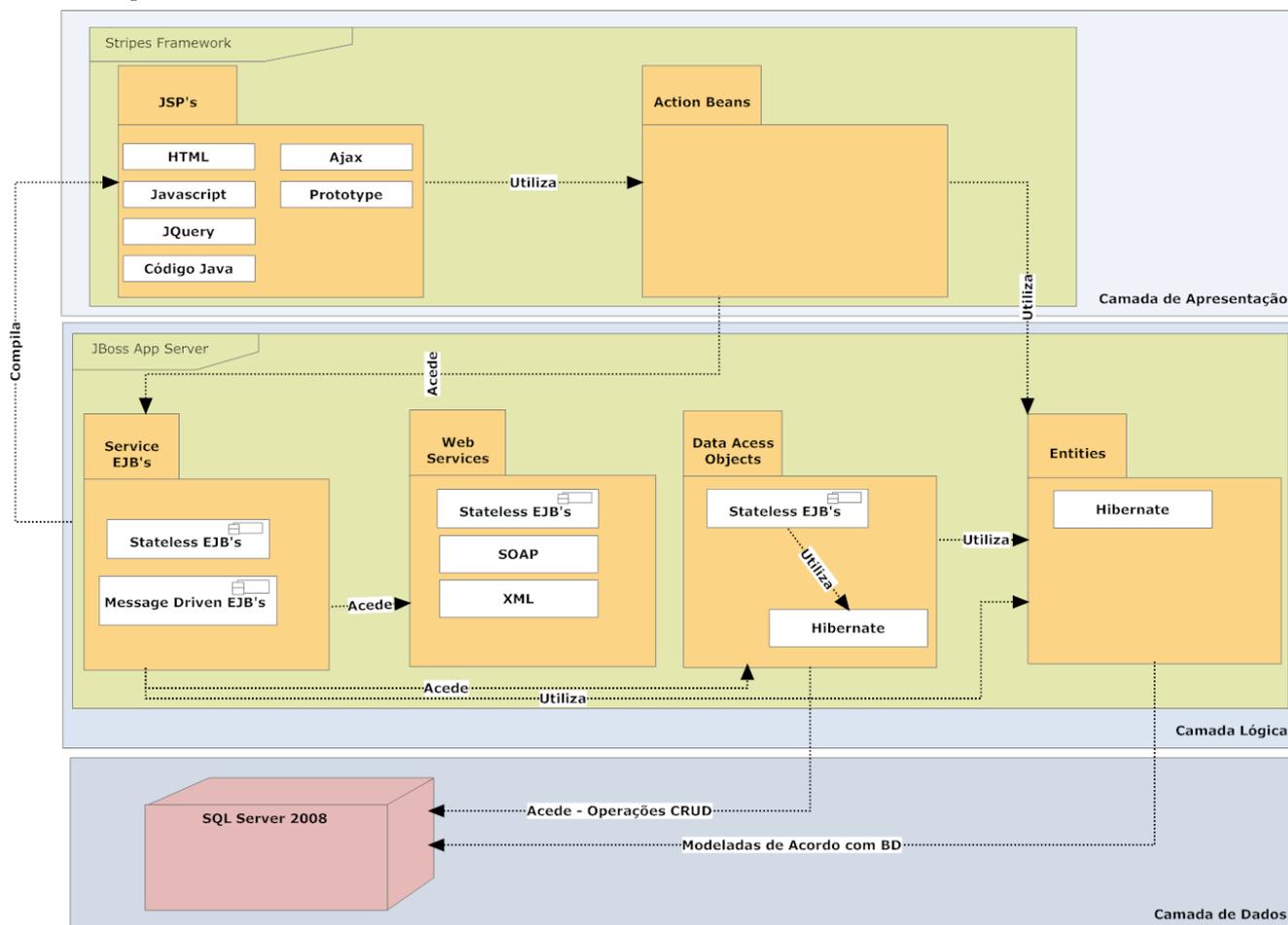


Figura 4 – Arquitetura WISE – Nível 3

A camada de apresentação é constituída por várias JSP's⁴ que, geradas dinamicamente, formam a interface gráfica da aplicação. Estas páginas utilizam tecnologias Web tradicionais que permitem o desenvolvimento de interfaces ricas em design e tratamento de eventos. As JSP's encontram-se integradas na framework *stripes* e diretamente relacionadas com os *Action Beans*.

Stripes é uma framework open-source utilizada no desenvolvimento de aplicações web em ambientes Java. Adoptando o modelo MVC⁵, permite o desenvolvimento rápido,

⁴ Java Server Pages

⁵ Model View Controller

fácil e eficaz de aplicações web. A framework é caracterizada por dois aspetos fundamentais: a criação de uma camada de abstração para a *servlet* e tratamento dos pedidos HTTP, e a redefinição de algumas tags HTML. A camada de abstração resulta da utilização de *Action Beans*, uma interface para todas as classes que respondem a pedidos. Através de um conjunto de anotações fornecidos pela interface, é possível mapear os vários pedidos aos respetivos *EventHandlers* [9][10]. A Figura 5 ilustra um exemplo.

```
@HandlesEvent("rejectRequisition")
public Resolution rejectRequisition() throws StripesAuthorizationException{
    WISERequisitionStateDefinition initialApproverState = this.newResourcesService.findStateByKey("initialApprover");

    try {
        this.newResourcesService.rejectRequisition(requisitionId, initialApproverState.getId(), this.getContext().getRequest().getRemoteUser(),
            null, this.getContext().getRequest().getLocale());
    } catch (SystemAlertException e) {
        this.getContext().getValidationErrors().addGlobalError(new LocalizableError("newresources.pendingrequisitions.details.reject.fail"));
        return new ForwardResolution(PendingRequisitionStepActionBean.class).addParameter("requisitionId", requisitionId);
    }

    this.getContext().getMessages().add(new LocalizableMessage("newresources.pendingrequisitions.details.reject.success"));
    return new ForwardResolution(PendingRequisitionStepActionBean.class).addParameter("requisitionId", requisitionId);
}
```

Figura 5 – Mapeamento entre Evento e método

Analisando o exemplo, verificamos que a anotação *@HandlesEvent* permite associar o nome do evento ao método pretendido. A Figura 6 complementa o exemplo anterior ilustrando como associar um *hyperlink* ao evento *rejectRequisition*.

```
<<:otherwise>
<display:column media="html" class="column-one-image noborder" headerClass="noborder">
    <stripes:link beanclass="com.criticalsoftware.wiseweb.stripes.action.newresources.PendingRequisitionsActionBean"
        event="rejectRequisition" class="rejectRequisition">
        <stripes:param name="requisitionId" value="${pendingRequisitions.id}"/>
        "
            title="<fmt:message key="newresources.common.rejectrequisition" />"/>
    </stripes:link>
</display:column>
</c:otherwise>
```

Figura 6- Utilização de tags Stripes

A framework disponibiliza um vasto conjunto de tags que simplificam a integração de JSP's com os respetivos Action Beans, e os vários pedidos com os respetivos eventos, tornando o desenvolvimento de aplicações web num processo simples e intuitivo. O principal objetivo dos Action Beans é realizar pré-processamento da informação proveniente da JSP, incluindo validação e tratamento de erros, antes de enviar o pedido para as restantes camadas. A validação pode incluir vários aspetos, como por exemplo, validar o formato da informação recebida, validar os acessos às funcionalidades com base nas permissões, ou validar que informação é disponibilizada ao utilizador. A validação pode ser realizada de duas formas, através de anotações específicas nos atributos do Action Bean, ou através de métodos específicos declarados com as devidas anotações. A Figura 7 apresenta um exemplo das duas formas possíveis.

```

@Validate(required = true, on = {"cancelClearance", "renewClearance"})
protected Long clearanceId;
@Validate(required = true, on = "cancelClearance")
protected String clearanceCreator;

/**
 * Validate start date and end date.
 */
@ValidationMethod(when = ValidationState.ALWAYS, on = { "clearanceReport" })
public void validateRequiredFields() {
    if ((this.startDate != null) && (this.endDate != null)) {
        // if start date after end date.
        if (this.startDate.after(this.endDate)) {
            this.getContext().getValidationErrors().add("startDate",
                new LocalizableError("workers.clearances.report.invalidDate"));
        }
    }
}
}

```

Figura 7 – Anotações Stripes para Validações

A camada subjacente, Lógica, é constituída por vários componentes de negócio e funciona como intermediário entre a camada de apresentação e a base de dados. Antes de discutir o papel que cada um desempenha é essencial introduzir a plataforma de desenvolvimento, o J2EE.

A plataforma J2EE é um conjunto de especificações padrão que descrevem componentes aplicativos para o desenvolvimento de aplicações empresariais. Este tipo de aplicação utiliza tipicamente componentes como *Enterprise Java Beans* e *Web Services* que, utilizados devidamente, fornecem características essenciais como segurança, controlo transaccional, fiabilidade, entre outras [11][12].

Nesta camada as entidades são criadas de acordo com a informação recebida e sofrem o processamento necessário antes de serem colocadas na base de dados ou enviadas para a camada de apresentação. Os Service EJB's, ou serviços, são utilizados para construir lógica de negócio e, dependendo da funcionalidade a executar, processam informação para enviar ao DAO ou para o Action Bean. Estes serviços são utilizados pelos Actions Beans e a comunicação é realizada através de *dependency injection* com a anotação *@WICCoreEJB* nos atributos do Action Bean. A Figura 8 ilustra exemplos da anotação mencionada.

```

@WICCoreEJB
protected UserService userService;
@WICCoreEJB
protected ClearanceService clearanceService;
@WICCoreEJB
protected DepartmentService departmentService;
@WICCoreEJB
protected AuthorizationService authorizationService;
@WICCoreEJB
protected CostCenterService costcenterService;
@WICCoreEJB
protected CompanyService companyService;
@WICCoreEJB
protected NewResourcesService newResourcesService;
@WICCoreEJB
protected ProjectService projectService;

```

Figura 8 – Exemplos *@WICCoreEJB*

Esta anotação realiza, de forma transparente, uma pesquisa ao JNDI⁶ pelo serviço correspondente. No contexto do WISE são utilizados dois tipos de EJB's, os *stateless*, que como o nome indica não têm estado durante o ciclo de vida da aplicação e a informação processada em cada pedido é independente dos restantes pedidos, e *message driven*, que são utilizados para o processamento de mensagens assíncronas, por exemplo o envio de E-mails.

⁶ Java Naming Directory Interface – Uma API java que permite às aplicações pesquisar objetos através do seu nome na diretoria de serviços.

Os *Data Access Objects* são o ponto de contato entre a base de dados e a camada lógica. A diferença entre os Service EJB's e DAO's é a camada com que interagem, Apresentação e Dados respetivamente. Nos DAO's são geridos os acessos à base de dados e criadas as *queries HQL (Hibernate Query Language)*, queries orientadas a objetos que serão executadas pelo Hibernate.

O Hibernate é uma framework que permite abstrair o programador da camada de dados. Simplifica a utilização de bases de dados através da normalização do mapeamento entre o seu modelo informação e os objetos java. As tabelas da base de dados são representadas através de classes Java, as *Entities* ou entidades utilizadas pelos vários componentes, e os seus atributos serão uma representação direta das colunas da tabela. A informação que se encontra nestas tabelas será instâncias das classes na aplicação [13][14]. A Figura 9 ilustra o exemplo de uma entidade com as respetivas anotações específicas ao Hibernate.

```
@Entity
@Table(name="NEWRESOURCESREQUISITIONS")
public class WISERequisitions implements Serializable{

    @Id
    @GeneratedValue
    @Column(name = "id", unique = true, nullable = false)
    private int id;

    @Basic
    @Column(name = "computerId", nullable = false)
    private int computerId;

    @Basic
    @Column(name = "login", nullable = false)
    private String login;

    @OneToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "login",referencedColumnName="login", insertable = false, updatable = false)
    private User user;

    @Basic
    @Column(name = "softwareId", nullable = false)
    private int softwareId;
```

Figura 9 – Entidade com anotações Hibernate

Por último, os *Web Services* são componentes que tipicamente fornecem funcionalidades que expõem informação útil de outros sistemas ou da própria aplicação. De características semelhantes aos EJB's e DAO's no que respeita às propriedades inerentes da plataforma J2EE e funcionalidades de consulta ou escrita de informação, acresce um conjunto de propriedades derivadas do suporte de tecnologias web e direcionadas à interoperabilidade entre sistemas. Veremos com maior detalhe este componente nos capítulos que se seguem.

Para finalizar este capítulo falta apenas discutir a camada de dados. Como referido anteriormente, o objetivo desta camada é armazenar a informação da aplicação. No contexto do WISE é utilizado o SQL Server 2008, uma base de dados relacional desenvolvida pela Microsoft.

3 Gestão de Projeto

A natureza de desenvolvimento de software requer o conhecimento de técnicas de gestão para a monitorização e controlo do processo de desenvolvimento. A gestão de projeto é a disciplina de planear, organizar, motivar e controlar recursos de forma a atingir objetivos específicos [15].

3.1. Equipa

A equipa deste projeto é constituída por três elementos, o estagiário Hugo Barrigas, responsável pelo desenvolvimento e análise do produto, Engenheira Daniela Domingues, desempenhando o papel de orientadora por parte da Critical e responsável pela validação de toda a documentação e produto desenvolvidos, e por último, Engenheira Diana Gomes, gestora do projeto WISE que durante o primeiro semestre auxiliou na validação da documentação produzida. Durante o segundo semestre o Engenheiro Daniel Fonseca forneceu sempre que necessário suporte no desenvolvimento.

3.2. Metodologia de Desenvolvimento

No desenvolvimento de software, uma atividade chave no seu sucesso é a escolha da metodologia adequada. Existe um vasto leque de metodologias que auxiliam a controlar e monitorizar o desenvolvimento. A escolha de uma em particular depende de vários fatores e características inerentes ao projeto.

Foram consideradas e estudadas as duas metodologias de desenvolvimento mais frequentes no desenvolvimento de software na Critical Software. *Waterfall* [16][18][19], uma metodologia tradicional que adota um processo sequencial e rigoroso constituído por um conjunto de fases bem definidas e, *Scrum* [17], uma metodologia ágil que se foca numa estratégia de desenvolvimento flexível e holística.

Os fatores que contribuíram na escolha de *Waterfall* como metodologia de desenvolvimento foram: o tamanho da equipa, constituída por apenas três elementos, sendo o estagiário responsável pela realização do projeto; o facto de as funcionalidades e visão do produto final serem claras e objetivas e, ainda, o estágio estar dividido em dois semestres. Esta metodologia é especialmente adequada a projetos onde é possível realizar uma especificação detalhada e completa dos requisitos *a priori*. Como o primeiro semestre foi dedicado a componentes de investigação, análise do problema e possíveis soluções, e, o segundo semestre dedicado à implementação das soluções definidas e respetiva validação, a escolha desta metodologia era a mais adequada.

O modelo **Waterfall** é por omissão, a metodologia adotada pelo departamento de qualidade da Critical Software. Trata-se de uma metodologia de desenvolvimento sequencial constituída por cinco fases: *Requirements Engineering*, *Design Engineering*, *Validation*, *Acceptance* e *Operations and Maintenance*. De forma a completar o ciclo de desenvolvimento é necessário concluir com sucesso cada uma das fases mencionadas. A Figura 10 ilustra as diferentes fases do modelo [18].

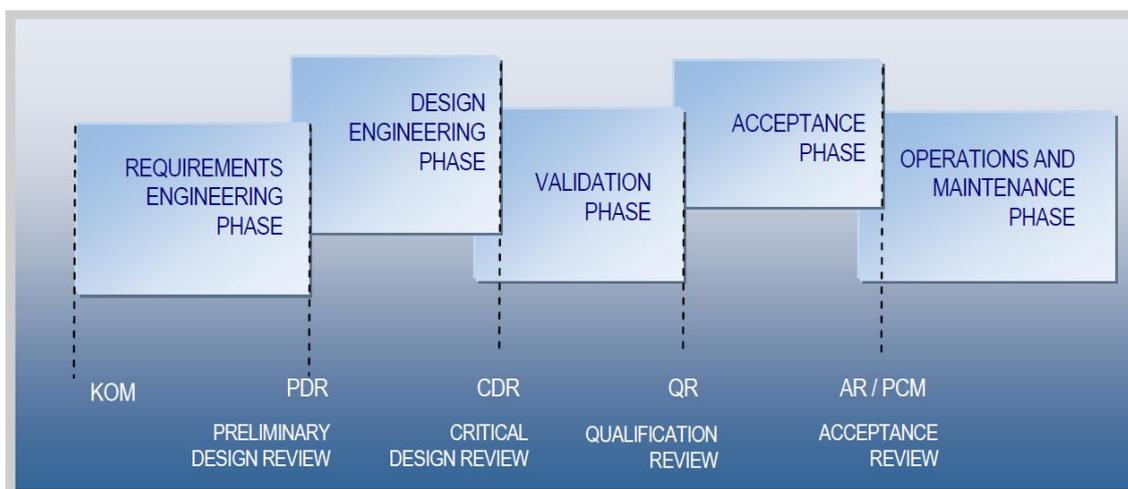


Figura 10 – Fases de desenvolvimento - Waterfall

Analisando a imagem verificamos que o processo não é inteiramente sequencial, mas que o fim de cada fase sobrepõe o início da próxima. Isto significa que durante o fim de cada fase é possível dar início à fase seguinte. Por exemplo, no fim da fase de requisitos é possível realizar paralelamente os protótipos rápidos ou, no fim da fase de *design*, alguma validação já poderá ser executada.

O início do projeto é marcado por uma *Kick-Off Meeting (KOM)*, onde é discutido o âmbito do projeto, a equipa é contextualizada e são apresentados os objetivos pretendidos. Uma vez realizada a *KOM*, é inicializado o processo começando pela fase *Requirements Engineering Phase*. Durante esta fase é realizada uma análise das necessidades do cliente e elaborado o documento técnico que responda às suas necessidades.

Quando o documento técnico está terminado, é realizada a *Preliminary Design Review (PDR)*. Esta reunião, que marca o fim da fase de requisitos e o início da fase de *design*, tem como objetivo validar a documentação elaborada durante a fase anterior. Durante a fase de *design*, é construído o produto definido e validado anteriormente, envolvendo todo o código necessário e os respetivos testes unitários.

Terminado o desenvolvimento é altura de realizar a *Critical Design Review (CDR)*. Durante esta reunião é analisado o código desenvolvido e verifica-se pela última vez se as funcionalidades implementadas estão de acordo com o esperado.

Após uma *CDR* bem sucedida começa a *Validation Phase*. Durante esta fase são realizados diversos testes de sistema na aplicação desenvolvida, verificando se as funcionalidades correspondem aos requisitos e especificações iniciais, não havendo qualquer erro.

Durante a fase seguinte, *Acceptance*, o sistema será novamente testado e validado pelo cliente, no ambiente de produção. O objetivo desta fase é garantir que o sistema testado no ambiente de desenvolvimento é igualmente válido no ambiente final. Após reunir com cliente e obter a sua aprovação final, é altura de lançar a primeira *release* do projeto. O ciclo de desenvolvimento do projeto poderá terminar nesta fase, avançando para a última fase de desenvolvimento apenas se necessário.

No anexo **B**, o leitor pode encontrar uma descrição mais detalhada de cada fase.

3.3. Gestão de Riscos

A gestão de riscos é um componente essencial na gestão de projetos. É uma abordagem sistemática e contínua, utilizada com o intuito de minimizar a vulnerabilidade a potenciais perdas na presença de situações adversas. De forma a classificar e ordenar os riscos, existe um conjunto de características mensuráveis, sendo estas:

- **Probabilidade** : Representa a exequibilidade de uma ameaça se concretizar;
- **Impacto**: Representa a perda que irá ocorrer caso a ameaça se concretize;
- **Magnitude**: Representa a vulnerabilidade a potenciais perdas com base na probabilidade e impacto.

O ciclo de vida da gestão de risco é constituído pelas três fases [20] representadas na Figura 11.



Figura 11 – Ciclo de vida da Gestão de Riscos

Durante a fase *Avaliar* são identificadas as situações, lacunas ou incertezas que podem revelar-se prejudiciais ao projeto e descritas sob a forma de riscos detalhados, distintos e tangíveis formalmente documentados. Ao longo desta fase é exetável que os riscos encontrados sejam definidos com a seguinte informação:

- **Título do risco**: Declaração sucinta e única do risco.
- **Probabilidade**;
- **Impacto**;
- **Magnitude**: Vulnerabilidade a potenciais perdas com base na probabilidade e impacto. Varia entre 1-muito baixa e 5-muito alta. Serve para ordenar os riscos de forma a identificar os mais importantes (≥ 3).
- **Descrição do risco**: Deve descrever claramente a ameaça e a consequência obtida caso a ameaça se concretize.
- **Tipo**: Pode tomar os seguintes valores:
 - I. Directo – O projecto tem controlo sobre o risco.
 - II. Indirecto – O projecto não tem controlo sobre o risco.
 - III. Recursos – Com origem organizacional, pessoal, tempo, fundos.
 - IV. Tecnológico – Dependências externas ou tecnológicas.

Na fase *Planear* é necessário determinar uma estratégia para resolver ou mitigar um risco, produzindo um plano para o realizar. Esta estratégia pode tomar os seguintes valores:

- **Aceitar:** Caso um risco se concretize, as suas consequências são toleráveis e não serão tomadas medidas.
- **Transferir:** O risco é transferido para outra entidade.
- **Evitar:** São tomadas medidas de forma a evitar que o risco aconteça.
- **Controlar:** São tomadas medidas de forma a controlar, ou conter o risco.

O plano desenvolvido para a abordagem tem que detalhar com precisão as atividades necessárias para ser executado com sucesso. É aconselhável estabelecer objetivos, identificar a(s) entidade(s) responsável para cada atividade, datas concretas que registam o início e o fim do plano/atividade(s) e, por fim, um plano de contingência. O plano de contingência pode ser considerado a última barreira face a um risco, especificando em concreto o que será feito caso o risco ocorra, detalhando as ações para corrigir e controlar o impacto que o risco causa.

A última fase do ciclo, *Mitigar*, tem como objetivo lidar com o risco através da execução da estratégia definida anteriormente. No caso de a estratégia definida não resultar será necessário executar o plano de contingência.

Na Figura 12, é ilustrada a matriz de risco utilizada ao longo do projeto.

Matriz de Risco		Impacto				
		1 Insignificante	2 Pequeno	3 Moderado	4 Urgente	5 Crítico
Probabilidade	5 Quase Certo				6,15,19,22	
	4 Provável		7	2,10,12,14,21	1,17	16
	3 Possível			4,2		8,9,11,13
	2 Pouco Provável				5,18	3
	1 Raro					

Figura 12 – Matriz de Risco

A matriz de risco é uma ferramenta muito útil na visualização rápida dos riscos, identificando facilmente os prioritários. Os identificadores numéricos dos riscos encontram-se distribuídos ao longo de três áreas diferentes que os classificam. A área vermelha identifica os riscos que apresentam cenários com elevada probabilidade de acontecerem e danificarem o desenvolvimento do projeto. Na zona amarela, situam-se os riscos que, embora não apresentem séria ameaça ao projeto, é necessário considerar e estar preparado na eventualidade de se agravarem. Por fim, na zona verde encontram-se

os riscos que normalmente não apresentam consequências significativas para o projeto, podendo aceitar-se as consequências sem graves danos ao projeto.

Na Figura 13 encontram-se os quatro riscos mais significativos que foram identificados ao longo do desenvolvimento do projeto.

ID	Data identificação (dd/mm/yyyy)	Título	Impacto	Probabilidade	Magnitude	Descrição	Tipo	Responsável	Estratégia de mitigação	Plano de contingência
6	11-12-13	Responsável pela validação dos documentos fora do País	4	5	4	O responsável pelos módulos de credenciações e recursos encontra-se fora do país. O atraso na validação dos documentos pode prologar-se ainda mais.	Indirecto	Daniela Domingues, Diana Gomes	Controlar: Caso o responsável pelos módulos não valide os documentos até ao final do mês de Dezembro será necessário marcar uma reunião de forma a validar pessoalmente os documentos por todos os envolvidos no projeto. Responsáveis do módulo e elementos da equipa.	Perguntar aos responsáveis dos módulos se é possível avançar com a implementação mesmo que os documentos apresentados não tenham sido validados formalmente.
15	10-4-14	Validação pendente dos requisitos do módulo de Gestão de Carreiras.	4	5	4	Até à data de hoje não existe qualquer feedback relativo ao documento de requisitos para o módulo de gestão de Carreiras. Esta situação pode causar um atraso significativo na implementação do módulo.	Indirecto	Hugo Barrigas, Daniela Domingues, Diana Gomes	CONTROLAR: Marcar uma reunião presencial para validar o documento de requisitos.	No caso de ser impossível marcar a reunião presencial e, atrasar ainda mais a validação do documento, implementar se possível a solução como está definido no documento entregue.
19	10-5-14	Validação pendente dos requisitos do módulo de Gestão de Carreiras.	4	5	4	Até à data de hoje não existe qualquer feedback relativo ao documento de requisitos para o módulo de gestão de Carreiras. Foi marcada uma reunião presencial para discutir o documento mas, os responsáveis do módulo voltaram a adiar. Esta situação provocou que, a implementação do módulo seja impraticável devido à falta de informação disponibilizada e, falta de tempo para alterar o documento se necessário, a sua validação, implementar e validar a solução.	Indirecto	Hugo Barrigas, Daniela Domingues	CONTROLAR: Marcar uma reunião presencial para validar o documento de requisitos, discutir todas as dúvidas e, debater as alterações na solução já definida.	Com base nesta reunião irá ser realizado um novo documento de requisitos que reflita os comentários da reunião. Se se verificar que implementar o módulo e, escrever o relatório final de estágio seja impraticável, só será definida a solução esperada via documento de requisitos.
22	10-5-14	Falta de Informação na BD do GLPI	4	5	4	A base de dados do GLPI tem falta de informação e informação mal introduzida. Para corrigir estas falhas foi disponibilizado acesso à aplicação como Admin. Esta tarefa irá consumir tempo necessário para na implementação.	Indirecto	Hugo Barrigas	Controlar: Corrigir e introduzir apenas a informação necessária para o módulo.	Caso esta tarefa consuma demasiado tempo será necessário comunicar com DSI para ajudar na correcção da BD.

Figura 13 – Riscos mais significativos do projeto

Como podemos ver nos exemplos ilustrados, os riscos são detalhados através da informação apresentada nas diferentes fases do ciclo de vida, sendo ordenados em ordem decrescente pela sua magnitude. O leitor poderá encontrar os restantes riscos identificados no anexo C.

3.4. Planeamento

De forma a controlar a execução do projeto e conseguir determinar em que estado se encontra, foram utilizados diagramas de Gantt. Os diagramas permitem acompanhar todo o processo de desenvolvimento através da sua decomposição em várias tarefas e/ou fases com início e fim bem definidos. Com base neste princípio é possível verificar o estado atual do projeto, isto é, se este se encontra atrasado, dentro do previsto ou até adiantado. Através destas observações é possível tomar medidas que mantenham o equilíbrio e contribuam para o sucesso do projeto. Na Figura 14 é apresentado o diagrama de Gantt geral utilizado no desenvolvimento do projeto.

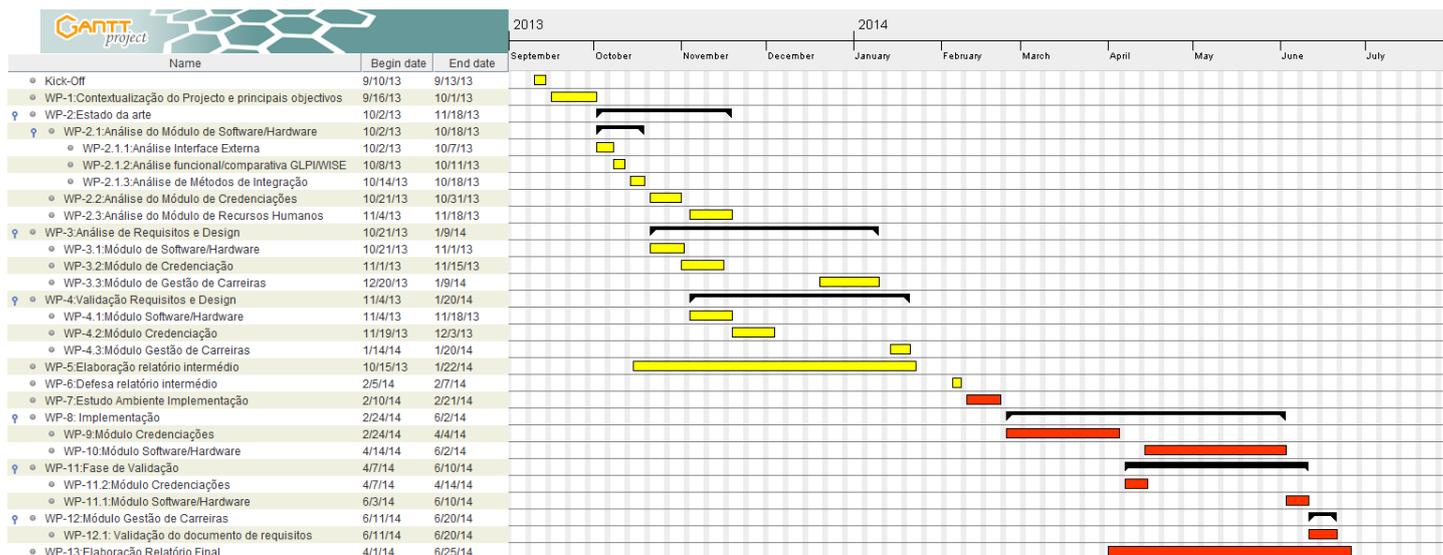


Figura 14 – Diagram Gantt geral do projeto

Analisando o diagrama verificamos a existência de tarefas com cores distintas. A cor amarela identifica as tarefas realizadas durante o primeiro semestre, enquanto, que a cor vermelha identifica as tarefas realizadas durante o segundo semestre.

O estágio começou por uma contextualização do projeto e a definição dos seus objetivos. Esta tarefa teve como objetivo perceber o que era realmente o sistema de informação WISE, as funções que desempenha e o impacto que causa no dia-a-dia da organização. Uma vez estabelecido o contexto do estágio foi realizado um estudo ao sistema de informação, envolvendo a sua arquitetura, o grau de desenvolvimento atingido até ao momento e, as ferramentas e tecnologias utilizadas. Durante esta tarefa foi também analisada a ferramenta a integrar no sistema com o intuito de averiguar se a integração seria praticável.

Terminado este estudo, realizaram-se reuniões com as entidades responsáveis dos módulos de forma a discutir as alterações necessárias no módulo de credenciações e, definir a nova solução para o módulo de recursos. Com base na informação extraída das reuniões foram elaborados os documentos técnicos que, através de requisitos funcionais, casos de uso e protótipos rápidos, especificam as soluções a implementar. Os documentos produzidos foram submetidos a várias sessões de validação pela orientadora e pelas entidades responsáveis dos módulos, permitindo uma visão final e concreta, partilhada por todos os intervenientes do projeto.

O segundo semestre iniciou com um estudo e preparação do ambiente de desenvolvimento. Durante esta fase foram também realizados alguns exercícios de preparação à implementação. Terminada a fase de adaptação, iniciou-se o desenvolvimento com o módulo de credenciações. No início desta fase foi desenvolvido o diagrama de Gantt ilustrado na Figura 15.

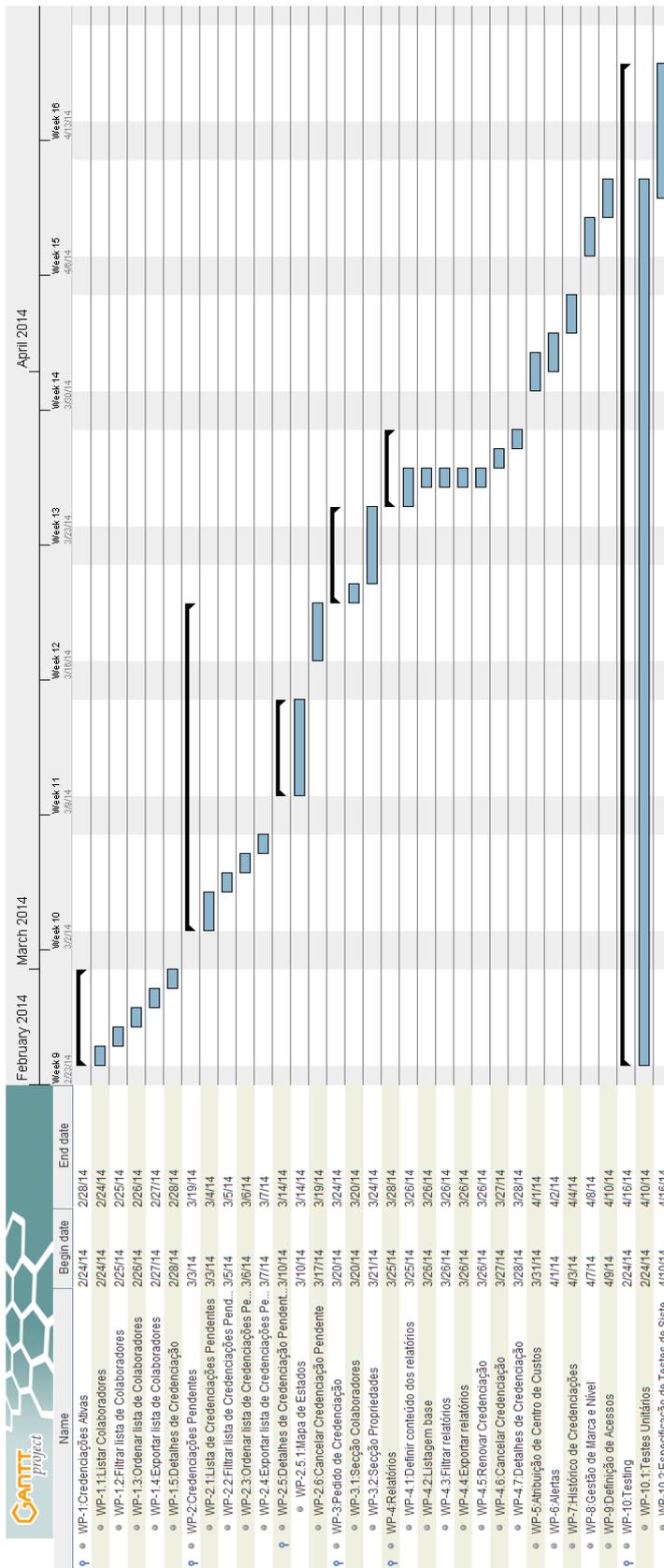


Figura 15 – Diagrama Gantt do módulo de Credenciações

O diagrama provou ser uma ferramenta extremamente útil no desenvolvimento, permitindo monitorizar constantemente o progresso e controlar o esforço dedicado. A produção de código foi acompanhada de testes unitários para a garantir a sua qualidade. No final da implementação, foram desenvolvidos vários testes de sistema específicos às funcionalidades criadas.

Assim que a reestruturação do módulo de credenciações terminou e a solução final ficou validada através dos testes definidos, partiu-se para a fase mais desafiante do estágio, a implementação do novo módulo de recursos e a integração da nova ferramenta no sistema.

Antes de iniciar o desenvolvimento do novo módulo, foi criado novamente um diagrama de Gantt dedicado. A Figura 16 apresenta este diagrama.

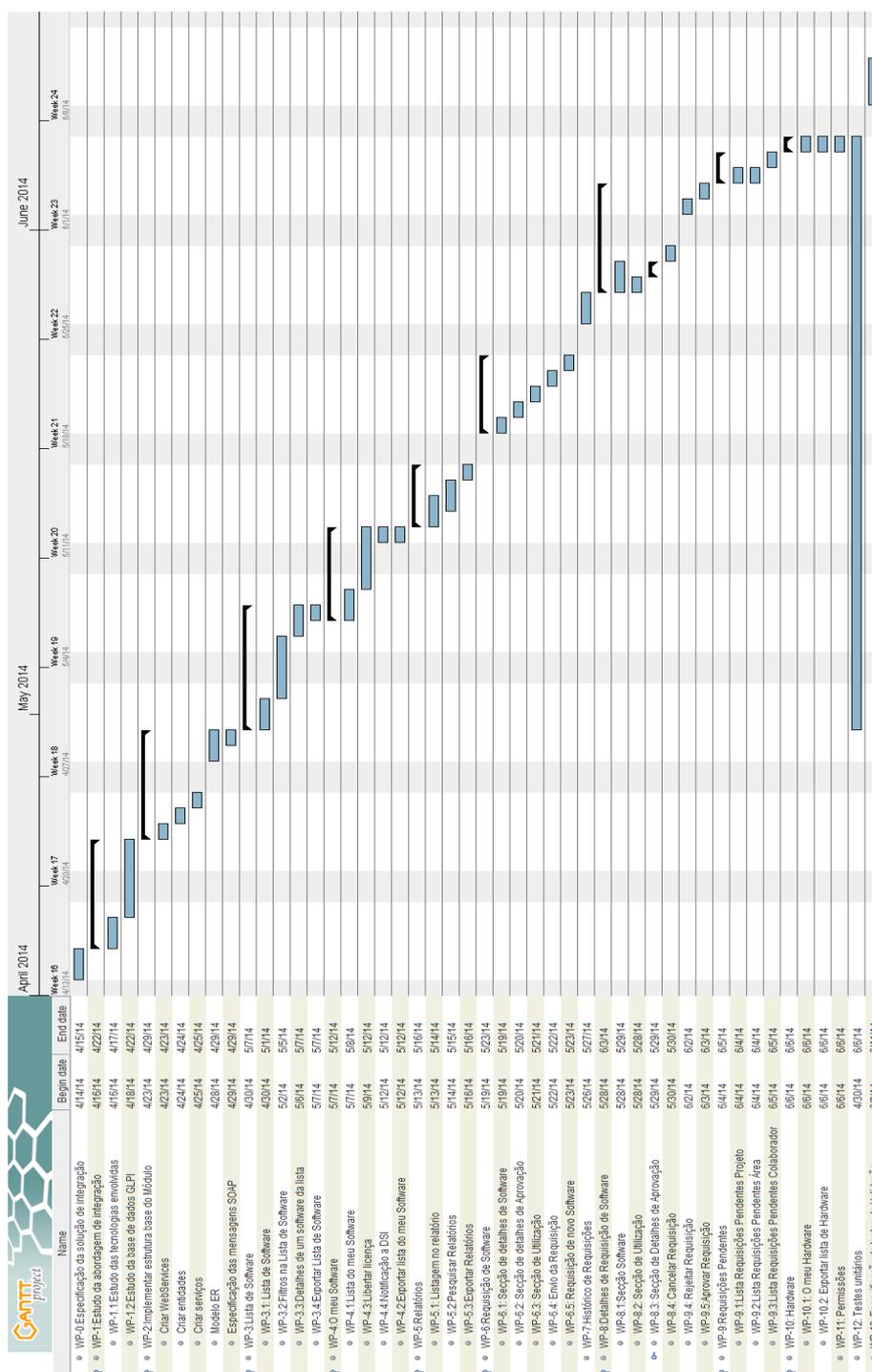


Figura 16 - Diagrama Gantt do módulo de Recursos

A implementação deste novo módulo arrancou com a especificação detalhada da solução de integração, incluindo a arquitetura do módulo, tecnologias a utilizar e o estudo da base de dados do GLPI. Uma vez definida e validada a abordagem de integração, avançou-se para a fase crucial no desenvolvimento, criar a base do módulo. A base do módulo resume-se na criação todos os seus componentes, serviços EJB, DAO's, Web Services, entidades, JSP's, ActionBeans e, estabelecer o fluxo de comunicação entre eles. Com a base criada e, juntamente com a experiência obtida durante a reestruturação do módulo de credenciações, o desenvolvimento revelou ser um processo fluído e claro.

A estratégia de implementação adotada envolveu dois passos: primeiro optou-se por implementar as funcionalidades que envolviam informação proveniente do GLPI, e só posteriormente seriam implementadas as funcionalidades associadas às requisições de software. O motivo desta estratégia foi isolar a informação externa da informação interna do módulo, preparando simultaneamente a informação que seria necessária para as requisições de software.

Tal como no módulo de credenciações o código produzido foi acompanhado de testes unitários, finalizando o processo de desenvolvimento com a especificação de testes de sistema que serão utilizados na fase de validação da solução implementada.

Por fim, durante o decorrer do projeto existiram alguns desvios ao plano apresentado e que estão relacionados com o último módulo Gestão de Carreiras e, uma tarefa no módulo de Recursos. Inicialmente estava previsto realizar também alterações no módulo de Recursos Humanos, no entanto, os responsáveis do módulo informaram que não tinham experiência suficiente na sua utilização para pedirem alterações concretas. Foi então decidido substituir estas alterações pela implementação de um novo módulo para gerir as carreiras dos colaboradores. Durante o primeiro semestre foi elaborado, e submetido para validação, o documento técnico que definia a solução com base na informação fornecida. A validação por parte dos responsáveis do módulo atrasou até ao dia 16 de Junho, data em que se realizou a segunda reunião para contextualizar o estagiário do que realmente era pretendido na gestão de carreiras, as várias fases do seu ciclo e qual seria a solução esperada. Devido a este atraso foi impossível realizar o novo documento técnico que retratasse a nova visão.

Outro desvio a considerar foi o facto de, durante o desenvolvimento do módulo de recursos existiu um atraso na tarefa *WP-2*, conseguindo no entanto, recuperar o tempo perdido durante a implementação das funcionalidades.

4 Desenvolvimento

Durante este capítulo será apresentado o trabalho desenvolvido durante o projeto. A apresentação segue uma ordem sequencial remetendo o leitor para a metodologia de desenvolvimento utilizada.

4.1. Módulo de Recursos

Uma das fases mais relevantes durante o desenvolvimento foi a especificação dos requisitos. Após analisar cuidadosamente a solução esperada, as funcionalidades pretendidas foram definidas na forma de requisitos. Estes requisitos podem ser divididos em quatro categorias: **O meu Software, Lista de Software, Relatórios, Requisições e Hardware**. De forma a complementar os requisitos são utilizados protótipos rápidos que contribuam na sua validação e, permitem visualizar qual será o seu aspeto gráfico. É importante esclarecer que os requisitos apresentados são os mais relevantes nas categorias mencionadas, para obter a lista completa o leitor deve dirigir-se ao anexo **D**.

4.2.1 O meu Software

Nesta categoria encontram-se os requisitos associados ao software do colaborador, definindo a informação que deve ser apresentada e as funcionalidades pretendidas.

SRS-RES-032-Lista do meu Software							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-10-13
Type:	Functional						
Deve ser possível listar todo o software associado ao(s) equipamento(s) do colaborador com os seguintes campos: <ul style="list-style-type: none"> ▪ Equipamento ▪ Nome do software ▪ Versão do software ▪ Número de Série ▪ Descrição ▪ Data de expiração – Data de expiração da licença. Caso já tenha expirado deverá aparecer a negrito. ▪ Libertar Licença - Esta opção estará disponível para cada linha da lista e irá libertar uma licença de software. 							

Tabela 2 – Lista do meu Software

SRS-RES-034-Libertar Licença							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-10-13
Type:	Functional						
Deve ser possível libertar uma licença de software associada a um colaborador. Esta funcionalidade será realizada através do WISE e será concretizada através do SRS-RES-035-Notificação a DSI.							

Tabela 3 – Libertar Licença

SRS-RES-035-Notificação a DSI							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-10-13
Type:	Functional						
<p>Deve ser enviado um E-Mail a DSI notificando que um utilizador deseja libertar a licença que possui para um determinado software. O E-Mail deve conter a seguinte informação:</p> <ul style="list-style-type: none"> ▪ Utilizador (login) ▪ Software (Fabricante + Produto + Versão) ▪ Identificador da licença (productkey ou equivalente) ▪ Equipamento ao qual estava atribuído. 							

Tabela 4 – Notificação a DSI

De seguida é apresentado o protótipo que ilustra o aspeto gráfico desejado para os requisitos anteriores.

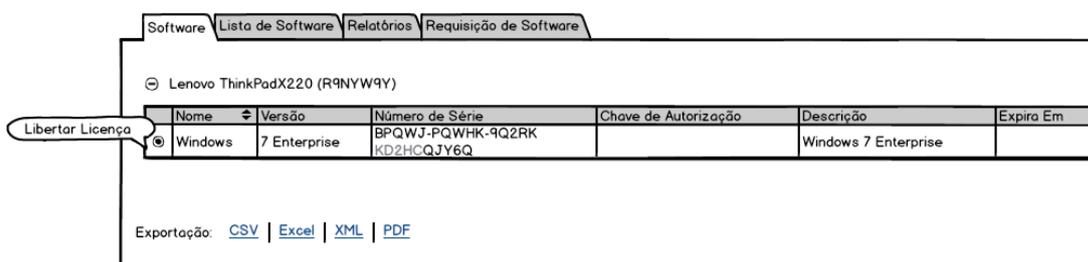


Figura 17 – Software do utilizador

Através da Figura 17 verificamos que o software associado ao colaborador é apresentado através de uma lista. Cada equipamento do utilizador terá a sua própria lista com o respetivo software. É importante salientar que a informação utilizada provém da ferramenta GLPI, no entanto, como existe a possibilidade de libertar licenças, é necessário armazenar informação no WISE de forma a identificar as licenças libertadas.

4.2.2 Lista de Software

Nesta categoria encontram-se os requisitos associados à lista de software, definindo as funcionalidades pretendidas e respetiva informação. A informação utilizada é novamente fornecida pela ferramenta GLPI.

SRS-RES-030-Lista de Software							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-10-13
Type:	Functional						
<p>Deve ser possível listar todo o software com os seguintes atributos:</p> <ul style="list-style-type: none"> ▪ Nome ▪ Versão ▪ Tipo de Software ▪ Tipo de licença ▪ Número de Licenças disponíveis ▪ Número total de Licenças 							

SRS-RES-030-Lista de Software

- Estado
- Pedir Software – Esta opção estará disponível para cada linha da lista e irá reencaminhar o utilizador para a secção de requisição de software.

Tabela 5 – Lista de Software

SRS-RES-031-Filtros na Lista de Software

Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-10-13
Type:	Functional						
Deve ser possível filtrar com os seguintes parâmetros:							
<ul style="list-style-type: none"> ▪ Nome ▪ Tipo de Software ▪ Tipo de Licença 							

Tabela 6 – Filtros a aplicar na Lista de Software

SRS-RES-034-Detalhes de um software da lista

Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-10-13
Type:	Functional						
Deve ser possível apresentar os detalhes de cada software presente na lista através dos seguintes campos:							
<ul style="list-style-type: none"> ▪ Nome ▪ Versão ▪ Tipo de Software ▪ Tipo de Licença ▪ Descrição ▪ Fabricante ▪ Estado ▪ Nº de licenças disponíveis ▪ Nº total de licenças 							

Tabela 7 – Detalhes de um item da Lista de Software

SRS-RES-035-Requisição de Software

Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-10-13
Type:	Functional						
Clicando num item da lista de software o utilizador deve ser reencaminhado para o separador Requisições, preenchendo automaticamente os detalhes de software com a informação do item seleccionado.							

Tabela 8 – Requisição de Software a partir de um item da lista de software

A Figura 18 ilustra o protótipo definido para a lista de software.

Software

The screenshot shows a web application interface for software management. At the top, there are navigation tabs: 'Software', 'Lista de Software', 'Relatórios', and 'Requisição de Software'. Below the tabs is a search form with fields for 'Nome', 'Tipo', and 'Tipo de Licença', and a 'Procurar' button. Below the search form is a table with 7 columns: 'Nome', 'Versão', 'Tipo de Software', 'Estado', 'Licenças', 'Tipo de Licença', and an action icon. The table contains three rows of data. Below the table is a details panel with fields for 'Nome', 'Versão', 'Tipo de Software', 'Tipo de Licença', 'Descrição', 'Fabricante', 'Estado', 'N° total de Licenças', and 'N° de licenças disponíveis'. At the bottom, there are export options: 'Exportação: CSV | Excel | XML | PDF'. A callout box on the right side of the table contains the text: 'Pedir Software. Clicando nos ícones de seta para a direita, é possível fazer requisição com 0 licenças.'

Figura 18 – Lista de Software

Todo o software que existe no GLPI será apresentado ao utilizador através de uma lista onde é possível aplicar uma série de filtros para facilitar a pesquisa. Na lista encontram-se também funcionalidades úteis ao utilizador, como requisitar o software e visualizar detalhes extra do software.

4.2.3 Relatórios

Nesta categoria encontram-se os requisitos associados aos relatórios de software, definindo as funcionalidades pretendidas e respetiva informação. A informação utilizada é novamente fornecida pela ferramenta GLPI.

SRS-RES-020-Relatórios							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-10-13
Type:	Functional						
Deve ser possível gerar um relatório com o software associado aos colaboradores constituído pelos seguintes campos: <ul style="list-style-type: none"> ▪ Colaborador ▪ Nome do Software ▪ Versão do Software ▪ Equipamento ▪ Empresa ▪ Área 							

Tabela 9 – Relatório de Software

SRS-RES-021-Pesquisar relatórios							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-10-13
Type:	Funcional						
<p>É possível aplicar os seguintes filtros nos relatórios:</p> <ul style="list-style-type: none"> ▪ Empresa – Lista de empresas que pertencem ao grupo Critical. Permite filtrar por colaboradores da empresa selecionada. Ao seleccionar uma opção, as listas de Área e Colaborador são atualizadas com áreas e colaboradores associados à empresa. No caso de não ser selecionada nenhuma empresa, as listas de Área e Colaboradores não contêm informação. ▪ Área – Lista de áreas associadas à empresa selecionada. Permite filtrar por colaboradores da Área selecionada. Ao seleccionar uma opção limita-se a informação do filtro Colaboradores a colaboradores que pertencem à Área selecionada. ▪ Colaborador – Lista de colaboradores que estão associados à Área e Empresa seleccionados. Limita a informação do relatório ao colaborador selecionado. No caso de nenhuma Área ou Empresa forem selecionadas, contém todos os colaboradores existentes para todas as áreas e empresas. ▪ Software - lista com todo o software ativo e disponível no momento. Limita a informação do relatório ao software selecionado. Esta lista provém do software existente no GLPI. <p>Caso nenhum filtro seja especificado, é obtida uma lista de software associado para todos os colaboradores.</p>							

Tabela 10 – Filtros no relatório

Software

Software | Lista de Software | Relatórios | Requisição de Software

Empresa: ▼
 Área: Todos os Colaboradores ▼
 Colaborador: ▼
 Software: ▼
 Procurar

Primeiro/Anterior | 1 2 3 4 5 6 7 8 | Próximo/Último

1649 registos encontrados, registos de 1 a 50.

Colaborador	Nome de Software	Versão	Equipamento
Abel Pinto	Fabricante + Nome	XP Professional SP3	Marca + Modelo (N° de Série)
Abel Pinto	Microsoft Windows	XP Professional SP3	Lenovo ThinkPadX220 (R9NYW9Y)

Exportação: [CSV](#) | [Excel](#) | [XML](#) | [PDF](#)

Figura 19 – Relatórios de Software

Os relatórios contêm informação acerca de todas as licenças de software atribuídas aos colaboradores. Nesta lista é possível aplicar uma série de filtros que permitem personalizar a informação do relatório.

4.2.4 Requisições de Software

As requisições de software permitem automatizar o processo de solicitar software necessário. As categorias anteriores utilizavam exclusivamente informação proveniente do GLPI, no entanto, o processo de requisição é constituído por um conjunto de etapas onde é possível realizar ações que afetam o seu estado. Assim, foi necessário dividir o processo em duas fases distintas em que cada uma contém objetivos específicos. A primeira, executada no contexto do WISE, inclui as etapas de submissão da requisição, respetiva aprovação ou rejeição por uma entidade superior e, o envio da requisição para o departamento de sistemas informáticos (DSI). A segunda fase, executada no contexto do GLPI, inclui as etapas de análise sobre a requisição e atribuição de uma licença de software. A Figura 20 ilustra o diagrama de estados do processo de requisição.

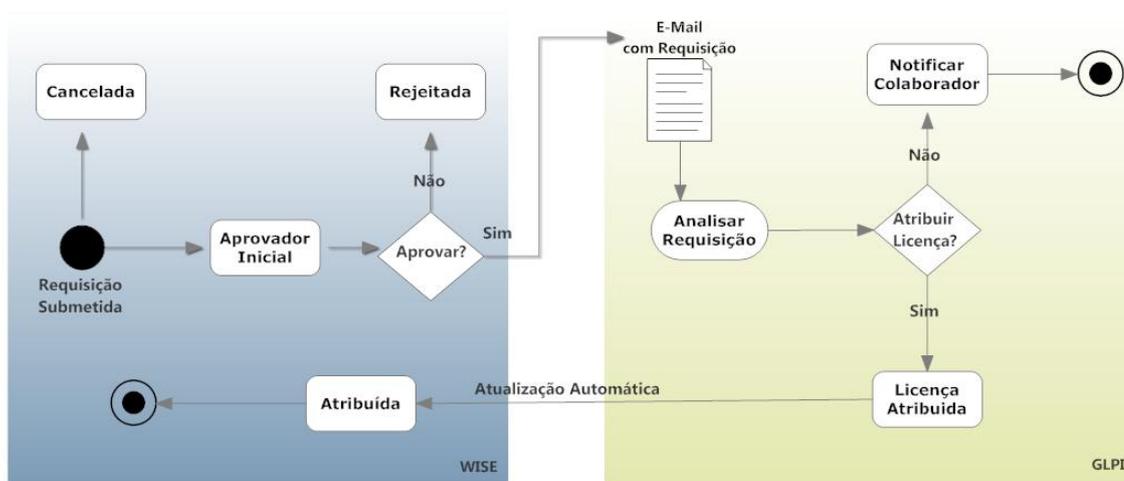


Figura 20 – Processo de Requisição de Software

O processo é inicializado através da submissão de uma requisição. Uma vez submetida, ficará pendente no estado *Aprovador Inicial* até que o responsável tome uma decisão. No caso de ser aprovada, será enviado um e-mail a DSI com os detalhes da requisição. Na segunda fase do processo é realizada uma análise da requisição e decidido se será atribuída a licença de software. Se eventualmente atribuírem uma licença no GLPI, a informação será transmitida automaticamente ao WISE, atualizando assim o estado da requisição, respetiva informação e, finaliza-se o processo.

De seguida são apresentados os principais requisitos desta categoria.

SRS-RES-040-Requisição de Software							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-10-13
Type:	Functional						
Deve ser possível efectuar o pedido de uma requisição de software. A requisição de software é constituída pelas seguintes secções: <ul style="list-style-type: none"> ▪ Secção de Detalhes de Software; ▪ Secção de Detalhes de Aprovação; ▪ Secção de Utilização. 							

Tabela 11 – Requisição de Software

SRS-RES-044-Secção de Detalhes de software							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-10-13
Type:	Functional						
<p>Esta secção é de preenchimento automático e contém os seguintes campos:</p> <ul style="list-style-type: none"> ▪ Nome ▪ Versão ▪ Descrição ▪ Tipo ▪ Fabricante <p>Toda a informação provém de uma ferramenta externa.</p>							

Tabela 12 – Detalhes de Software

SRS-RES-043-Secção Detalhes de Aprovação							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-10-13
Type:	Functional						
<p>Esta secção é composta pelos seguintes campos:</p> <ul style="list-style-type: none"> ▪ Notas iniciais – Caixa de texto para introduzir notas relevantes à requisição. ▪ Aprovador Inicial * – Lista que contém os aprovadores iniciais para a requisição. O conteúdo deste campo depende da escolha efectuada pelo colaborador nos campos Área/Projecto referidos no requisito SRS-RES-042. <p>Os campos marcados com * são de preenchimento obrigatório.</p>							

Tabela 13 – Detalhes de Aprovação

SRS-RES-042-Secção de Utilização							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-10-13
Type:	Functional						
<p>Esta secção é composta pelos seguintes campos:</p> <ul style="list-style-type: none"> ▪ Data de início – Data inicial da requisição. ▪ Tipo de Requisição * – Projecto/Área/Geral. ▪ Empresa * – Lista com todas as Empresas. ▪ Área – Lista com todas as áreas existentes na empresa seleccionada. ▪ Projecto – Lista com todos os projectos disponíveis na empresa seleccionada. ▪ Equipamento * – Lista que contém todos os computadores associadas ao colaborador. <p>Os campos marcados com * são de preenchimento obrigatório.</p>							

Tabela 14 – Secção de Utilização

SRS-RES-046-Envio de requisição							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-10-13
Type:	Funcional						
<p>Uma vez efectuada a requisição e, aprovada pelo Aprovador Inicial, a requisição será enviada para DSI que ficará encarregue de concluir o processo associando uma licença ao colaborador (o processo do DSI é efectuado através de uma ferramenta externa).O envio da requisição será realizado através de um E-mail que, contém a seguinte informação:</p> <ul style="list-style-type: none"> ▪ Nome do Colaborador ▪ Empresa do Colaborador ▪ Área do Colaborador ▪ Nome do Software ▪ Equipamento ▪ Versão ▪ Tipo ▪ Fabricante ▪ Data do pedido ▪ Tipo de Requisição – Projecto/Área/Geral ▪ Projecto ▪ Área ▪ Comentários do Aprovador Inicial ▪ Aprovador Inicial 							

Tabela 15 – Envio de Requisição por E-mail

Na Figura 21 ilustra-se o protótipo rápido definido para representar os requisitos especificados.

Software

Software | Lista de Software | Relatórios | Requisição de Software

Software:

Detalhes de Software

Nome:	AutoMate
Versão:	6
Descrição:	AutoMate 6 Professional v6.11
Tipo:	Tipo Software

Utilização:

Data de Início:

Tipo de Requisição*:

Empresa*:

Área:

Projecto:

Equipamento*:

Detalhes de Aprovação:

Notas Iniciais:

Aprovador Inicial*:

Figura 21 – Requisição de Software

A informação utilizada nas diferentes secções da requisição deriva de ambos os sistemas. Na secção de detalhes de software, a informação relativa ao software e respetiva versão provém do GLPI. Nas restantes secções, exceptuando a informação dos equipamentos associados ao colaborador, a informação utilizada encontra-se no WISE. Nesta página o utilizador poderá também requisitar software que não esteja presente na lista. O requisito da Tabela 16 define esta funcionalidade.

SRS-RES-047-Requisição de Novo Software							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-10-13
Type:	Functional						
<p>Se o colaborador seleccionar esta opção será aberta uma nova janela onde é possível enviar um email ao aprovador inicial. Esta nova janela contém os seguintes campos:</p> <ul style="list-style-type: none"> ▪ Nome Software * ▪ Tipo de Requisição * ▪ Área ▪ Projecto ▪ Comentários ▪ Aprovador Inicial * <p>Os campos marcados com * são de preenchimento obrigatório.</p>							

Tabela 16 - Requisição de novo software

Software

Figura 22 – Requisição de novo software

Como podemos verificar na Figura 22, seleccionando a opção de *New Software* é aberta uma nova janela com um formulário onde o utilizador, preenchendo corretamente os

campos, pode pedir o software que pretende. O envio da requisição é novamente realizado por e-mail contendo os campos especificados no requisito .

No momento que uma requisição é submetida, pode ser monitorizada pelo utilizador ao longo de todo o processo. Os próximos requisitos definem a informação e funcionalidades pretendidas para monitorizar e visualizar os detalhes de requisições submetidas.

SRS-RES-060-Visualização dos detalhes de Requisição							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-10-13
Type:	Functional						
Deve ser possível visualizar os detalhes de uma requisição através das seguintes secções:							
<ul style="list-style-type: none"> ▪ Secção Software; ▪ Secção Utilização; ▪ Secção Detalhes de Aprovação; 							

Tabela 17 – Detalhes de Requisição

SRS-RES-061- Secção Software							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-10-13
Type:	Functional						
Nesta secção serão apresentados os seguintes campos:							
<ul style="list-style-type: none"> ▪ Nome ▪ Versão ▪ Descrição ▪ Tipo ▪ Fabricante 							

Tabela 18 – Secção de Software

SRS-RES-063-Secção de Utilização							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-10-13
Type:	Functional						
Nesta secção serão apresentados os seguintes campos:							
<ul style="list-style-type: none"> ▪ Data de Inicio ▪ Utilização ▪ Equipamento ▪ Nome do requerente ▪ Empresa do requerente ▪ Departamento do requerente 							

Tabela 19 – Secção de utilização

SRS-RES-064-Secção Detalhes de Aprovação							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-10-13
Type:	Functional						
Nesta secção são apresentados os seguintes campos: <ul style="list-style-type: none"> ▪ Mapa de Estados conforme o Requisito SRS-RES-065 ▪ Notas 							

Tabela 20 – Detalhes de Aprovação

SRS-RES-065-Mapa de Estados							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-10-13
Type:	Functional						
Neste mapa será possível visualizar os estados que constituem uma requisição, sendo estes: Pedido submetido -> Cancelada** -> Aprovador Inicial ->Rejeitada* -> Enviada a DSI -> Atribuída Cada estado concluído deve apresentar o seu responsável, a data de conclusão e, se existirem, as notas associadas através de uma <i>tooltip</i> . Os estados concluídos deverão ser de cor verde, estados pendentes de cor cinza e, o estado atual de cor azul. Em alternativa, os estados poderão ter imagens intuitivas ao utilizador. *Só será apresentado caso essa requisição tenha sido rejeitada pelo aprovador inicial. **Só será apresentado no caso de o colaborador cancelar a requisição. Apenas o colaborador que fez a requisição poderá cancelar.							

Tabela 21 – Mapa de estados

De seguida é apresentado o protótipo definido para os requisitos apresentados.

Software

Software
Lista de Software
Relatórios
Requisição de Software

Software:

Nome:

Versão:

Descrição:

Tipo de Licença:

Fabricante:

Detalhes de Aprovação

Mapa de Estados:

Pedido Submetido → Aprovador Inicial → Enviado a DSI → Atribuída

Requerente + Data

Notas:

Utilização:

Data de Início: 22/08/2011

Utilização: Projecto-Sentinel-2
Central SW Development

Equipamento: Fabricante+Modelo+Nº de Série

Figura 23 – Detalhes de requisição

Nos detalhes de uma requisição pendente é possível visualizar toda a informação necessária para contextualizar o utilizador. A informação relativa ao software, versão e hardware deriva novamente do GLPI. A monitorização do processo é realizada através de um mapa de estados que permite informar ao utilizador o seu estado atual. No caso de o processo se encontrar no estado *Aprovador Inicial* e o utilizador respeitar as condições definidas, será possível aprovar ou rejeitar a requisição escolhida.

Outra funcionalidade esperada na solução final é a visualização de requisições pendentes. De seguida são apresentados os principais requisitos definidos para disponibilizar esta informação.

SRS-RES-050-Listar Requisições Pendentes de Projecto							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-10-13
Type:	Functional						
<p>Deve ser possível obter uma lista de todas as requisições pendentes de aprovação para todos os projectos onde o utilizador é <i>Project Manager</i> ou <i>Project Management Officer</i>. A lista é composta pelos seguintes campos:</p> <ul style="list-style-type: none"> ▪ Requerente ▪ Data da Requisição ▪ Nome do Software ▪ Tipo de Software <p>Esta funcionalidade encontra-se disponível apenas aos utilizadores com role <i>isPM</i> e <i>isPMO</i>.</p>							

Tabela 22 – Requisições pendentes de Projeto

SRS-RES-051-Lista de Requisições Pendentes de Área							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-10-13
Type:	Functional						
<p>Deve ser possível obter uma lista de todas as requisições pendentes de aprovação para todas as áreas onde o utilizador é <i>Director</i> ou <i>Gestor</i>. A lista é composta pelos seguintes campos:</p> <ul style="list-style-type: none"> ▪ Requerente ▪ Data da Requisição ▪ Nome do Software ▪ Tipo de Software <p>Esta funcionalidade encontra-se disponível apenas aos utilizadores com role <i>wisemanagerrole</i>, <i>wisedirectorrole</i>.</p>							

Tabela 23 – Requisições pendentes de Área

SRS-RES-056-Lista de Requisições Pendentes Gerais							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-10-13
Type:	Functional						
<p>Deve ser possível listar todas as requisições de tipo Geral. Esta lista é composta pelos seguintes campos:</p> <ul style="list-style-type: none"> ▪ Requerente ▪ Data da Requisição ▪ Nome do Software ▪ Versão do Software ▪ Tipo do Software <p>Esta funcionalidade encontra-se disponível apenas ao director e manager de DSI.</p>							

Tabela 24 – Requisições pendentes Gerais

SRS-RES-052-Lista de Requisições Pendentes do Colaborador							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-10-13
Type:	Functional						
<p>Deve ser possível listar todas as requisições efectuadas pelo colaborador. Esta lista é composta pelos seguintes campos:</p> <ul style="list-style-type: none"> ▪ Requerente ▪ Data da Requisição ▪ Nome do Software ▪ Versão do Software ▪ Tipo do Software ▪ Cancelar Requisição - Esta opção estará disponível para cada item da lista. 							

Tabela 25 – Requisições pendentes do Colaborador

Requisições Pendentes

Requerente	Data de Requisição	Nome do Software	Tipo do Software	Versão do Software
Nuno Carreiró Monteiro	19/08/2011	Highcharts	Tipo	Versão
Paulo Fernandes	06/06/2012	Microsoft Office	Tipo	Versão
Hugo Barrigas	12/04/2013	Fancy Software	Tipo	Versao

Clicando na linha pretendida somos reencaminhados para uma página onde se encontram os detalhes da requisição.

Figura 24 – Requisições pendentes

Com base na Figura 24 e nos requisitos apresentados, concluímos que as requisições pendentes encontram-se divididas em quatro secções distintas consoante o seu tipo. A secção *As minhas* é de acesso livre e contém todas as requisições pendentes submetidas pelo utilizador. As restantes são de acesso limitado e apresentam uma lista específica para cada tipo de requisição.

Por último, é esperado que o módulo apresente ao utilizador informação relativa aos equipamentos que lhe estão atribuídos. De seguida é apresentado o requisito definido para retratar esta informação.

SRS-RES-080-Lista de Hardware							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-10-13
Type:	Functional						
<p>Deve ser possível obter uma lista de hardware associado ao colaborador. A lista é composta pelos seguintes campos:</p> <ul style="list-style-type: none"> ▪ Associado a - Valores: itGrow / Critical Software ▪ Fabricante ▪ Modelo ▪ Número de Série ▪ Tipo ▪ Data de Atribuição ▪ Data de Renovação ▪ Notas ▪ Localização ▪ Estado 							

Tabela 26 – Hardware do colaborador

Hardware

Fabricante	Modelo	Número de Série	Tipo	Associado a	Data Atribuição	Data Renovação	Notas	Location	Estado
Lenovo	ThinkPad X220	R9NYW9y	Portátil	Área	31/05/2012	31/05/2016	type:4291-RM6	Coimbra	Atribuído

Exportação: [CSV](#) | [Excel](#) | [XML](#) | [PDF](#)

Figura 25 – Hardware do Colaborador

O utilizador poderá exportar a lista para formatos convenientes como *PDF*, *XML*, *CSV* e *Excel*, bem como, visualizar o software associado a cada um dos computadores presentes na lista. É importante mencionar que a informação utilizada é obtida diretamente do GLPI.

Uma vez terminada a especificação do documento técnico, foi iniciado um ciclo de validação com a orientadora e clientes responsáveis pelo módulo. Uma vez aprovado pela orientadora, a equipa reuniu novamente com os responsáveis do módulo para efetuar uma validação presencial. Após realizar as alterações discutidas durante a reunião, o documento foi aprovado por ambas entidades definindo assim, a visão final da solução partilhada por todos os intervenientes do projeto.

4.2.5 Casos de Uso

Os casos de uso são bastantes úteis para representar as interações entre os diferentes atores e o sistema. De forma a complementar os requisitos e protótipos apresentados anteriormente, foram elaborados dois diagramas de casos de uso que identificam não só os diferentes atores do módulo, mas também, as funcionalidades a que têm acesso.

A interação com o sistema é bastante simples pois cada caso de uso representa uma página com um formulário ou informação na qual é possível aplicar filtros e realizar operações. A transição de estado implica a ação de carregar num botão ou clicar num link para realizar uma operação. Um aspeto importante a considerar é a relação direta entre os casos de uso e os requisitos definidos.

No total existem sete atores diferentes que desempenham um papel específico no módulo. Cada um é identificado através de um *role*⁷ associado que representa um nível de acesso específico. O *wisuserrole* representa o nível de acesso básico e comum a todos os atores. Os restantes representam utilizadores com acessos e privilégios adicionais.

O diagrama da Figura 26 representa os casos de uso para o ator básico, *wisuserrole*.

⁷ Traduzindo à letra, é um “papel” atribuído aos colaboradores que os identificam e limitam as suas permissões/acessos no WISE.

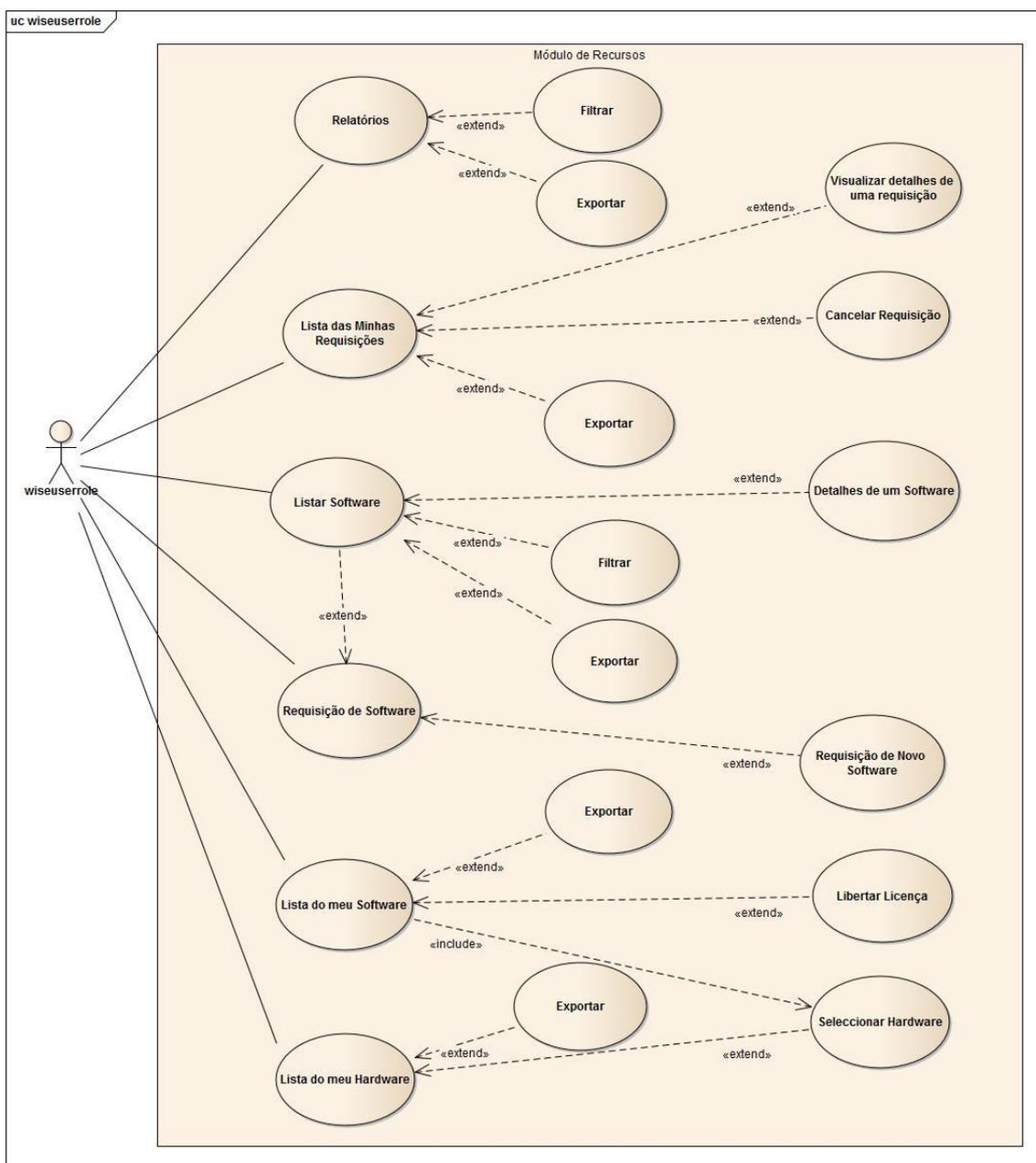


Figura 26 – Diagrama de casos de uso - wiseuserrole

Como mostra o diagrama, o ator pode realizar grande parte das funcionalidades definidas nas quatro categorias. Para tornar a usabilidade do módulo mais simples, optou-se por colocar as operações de cancelar requisição e libertar software, integradas nas listas *Minhas Requisições* e *O meu software*, respetivamente. Selecionando um software na *Lista de software*, o utilizador será redirecionado para a página de requisição, preenchendo automaticamente a secção de detalhes de software com o software selecionado. Verificamos também que a escolha de um hardware na *Lista do meu hardware* influencia a informação que será apresentada na *Lista do meu Software*.

Na Figura 27 encontra-se o diagrama de casos de uso para os restantes atores do módulo.

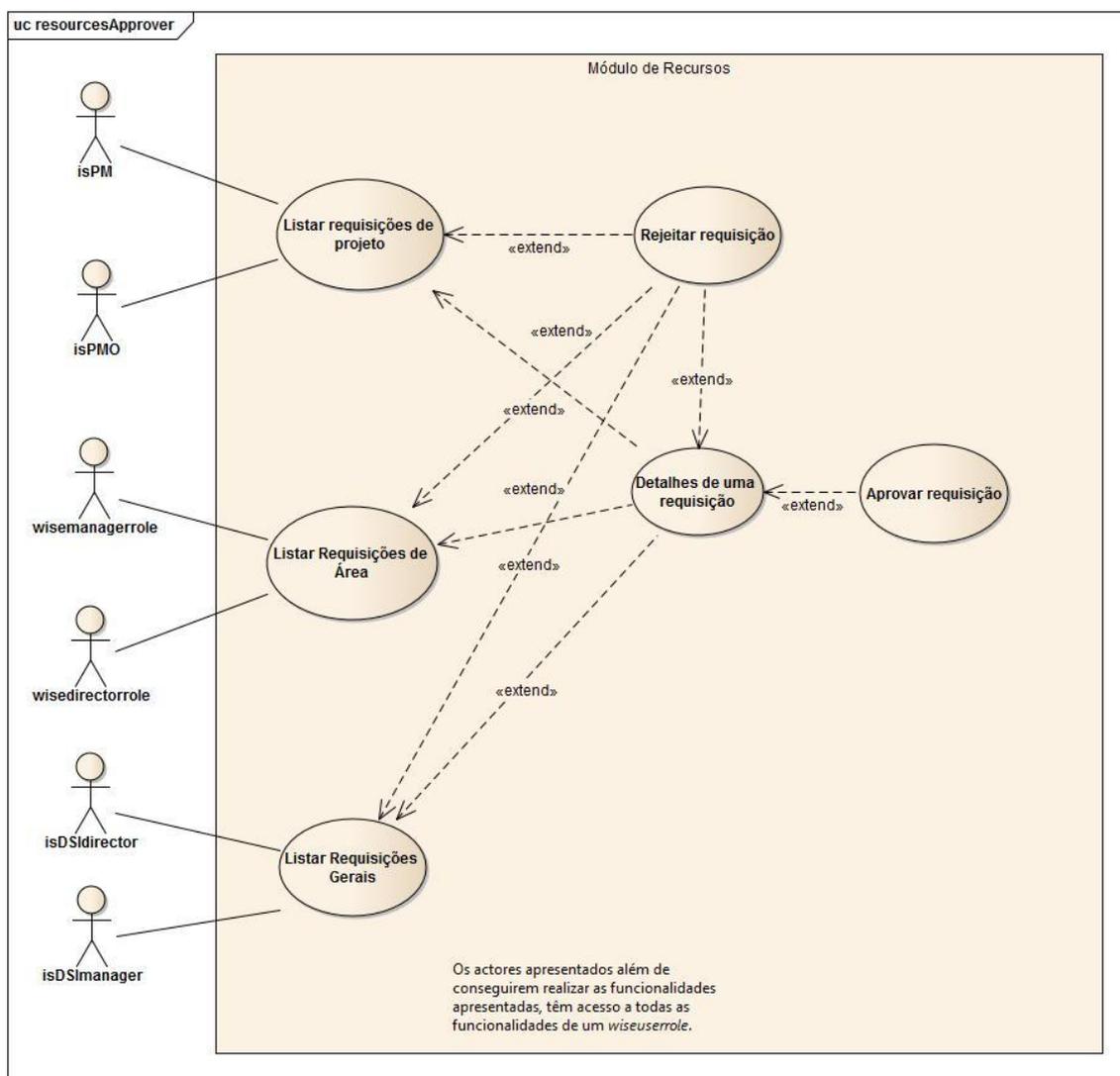


Figura 27 – Diagrama de casos de uso - Aprovadores

O diagrama identifica os atores com os privilégios necessários para ter acesso às listas de requisições pendentes. Os atores *isPM* e *isPMO* indicam utilizadores que desempenham um papel de gestão nos projetos. Apenas eles terão acesso à lista de requisições de projeto e respetivas aprovações ou rejeições. Os atores *wisemanagerrole* e *wisedirectorrole* identificam os utilizadores responsáveis pelas áreas da organização. Apenas eles terão acesso à lista de requisições de software submetidas no âmbito de área, ficando responsáveis pela sua aprovação ou rejeição. As requisições que não se enquadrem no âmbito de projeto ou área são do tipo geral, sendo apresentadas na lista Geral de requisições. Só o diretor e gestor de DSI, identificados pelos atores *isDSIdirector* e *isDSImanager*, poderão ter acesso a esta lista e poderão aprovar ou rejeitar as requisições.

4.2.6 Abordagem de Integração

Durante os requisitos verificou-se que a informação relacionada com software e hardware é facultada pelo GLPI. Nesta secção pretende-se detalhar a abordagem definida para integrar esta ferramenta no sistema de informação.

Dado que o cenário de integração é constituído por apenas duas aplicações, optou-se por utilizar uma abordagem Ponto-a-Ponto. O princípio desta abordagem é implementar um canal de comunicação direto entre os dois sistemas, estabelecendo um nível sofisticado de interoperabilidade com elevado controlo sobre a entrega e, processamento de informação [21]. O canal de comunicação foi estabelecido através da utilização de um Web Service, uma tecnologia com características muito úteis na integração de sistemas.

Uma vez definida a abordagem e o mecanismo para estabelecer o canal de comunicação, o passo seguinte foi decidir qual o protocolo de comunicação a utilizar. Foram considerados dois protocolos de comunicação tipicamente associados a Web Services, *Rest*⁸ e *SOAP*⁹. A escolha de uma em particular depende das necessidades do sistema e da equipa de desenvolvimento. Os fatores que influenciaram a escolha de *SOAP* como o protocolo de comunicação preferencial, foi o facto de já existir um conjunto de Web Services no sistema de informação que o utiliza e, permitir a definição de um contrato formal que normaliza a comunicação. Este contrato define informação essencial para utilizar o Web Service, como por exemplo, a sua localização, os métodos que disponibiliza, a informação que pretende receber e devolver, e, o formato das mensagens que são trocadas na comunicação. O contrato é descrito utilizando *WSDL*¹⁰, uma linguagem XML¹¹ que define a sua interface sintacticamente.

Na Figura 28 é ilustrada a abordagem de integração.

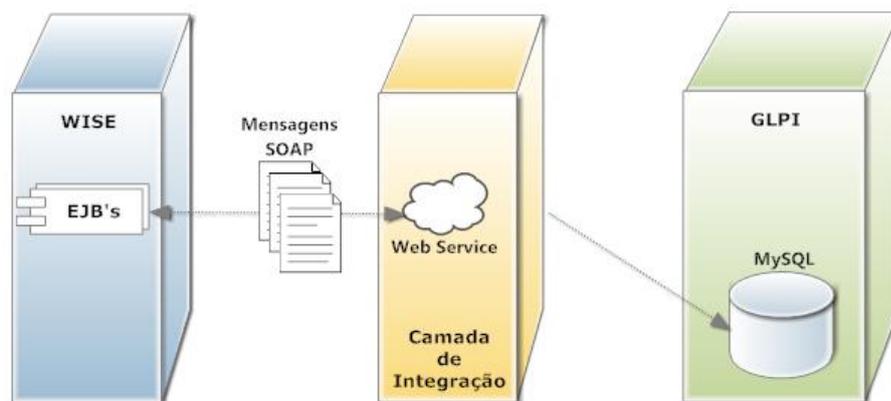


Figura 28 – Abordagem de integração

Através da figura verificamos que o cenário de integração é relativamente simples. O WISE (*consumer*) comunica com o Web Service (*provider*) através de mensagens SOAP. No Web Service é fornecido um conjunto de funcionalidades que através de pesquisas na base de dados do GLPI, expõe informação sobre software, hardware e licenças. A comunicação estabelecida é unilateral, isto é, o WISE é o único consumidor nesta relação com o GLPI.

⁸ Representational State Transfer

⁹ Simple Object Access Protocol

¹⁰ Web Service Definition Language

¹¹ eXtensible Markup Language

Na tabela 27 são listadas as funcionalidades que o Web Service dispõe.

Método	Descrição
getAllSoftware(String filterName, int softwaretype, int licensetype)	Envia mensagem com informação de todo o software gerido pelo GLPI.
getSoftwareInformation(int softwareId, int softwareVersionId)	Envia mensagem com informação de um software em particular.
getComputersByName(String filterName)	Envia mensagem com informação de todos os computadores associados a um colaborador.
getComputerInformationById(int id)	Envia mensagem com informação de um computador em particular.
getPhonesByName(String filterName)	Envia mensagem com informação de todos os telemóveis associados a um colaborador.
getMonitorsByName(String filterName)	Envia mensagem com informação de todos os monitores associados a um colaborador.
getPeripheralsByName(String filterName)	Envia mensagem com informação de todos os periféricos associados a um colaborador.
getSoftwareLicensesByComputer(int computerId)	Envia mensagem com informação de todas as licenças associadas a um computador em particular.
getAllSoftwareLicenses(int versionId,String login)	Envia mensagem com informação de todas as licenças geridas pelo GLPI.
getAllSoftwareTypes()	Envia mensagem com informação de todos os tipos de software existentes no GLPI.
getAllSimpleSoftware()	Envia mensagem com pouca informação de todo o software existente no GLPI.
getAllSoftwareLicenseTypes()	Envia mensagem com informação de todos os tipos de licença existentes no GLPI.
getLicenseInformation(int licenseId)	Envia mensagem com informação de uma licença em particular.

Tabela 27 - Funcionalidades do Web Service

Analisando as funcionalidades verificamos que se focam na consulta de informação. Nesta fase inicial de integração, o DSI pretende unicamente partilhar os dados relevantes, evoluindo futuramente para funções de escrita à medida que o processo evolui e amadurece.

Na Figura 29 são apresentadas as tecnologias utilizadas no WISE, GLPI e Camada de integração.



Figura 29 – Tecnologias utilizadas no WISE, GLPI, Camada de integração

As tecnologias utilizadas no projeto WISE já foram apresentadas anteriormente e por este motivo, aqui apenas serão focadas as tecnologias do GLPI (verde) e Camada de apresentação (amarelo).

De forma a comunicar com a base de dados MySQL do GLPI, o Web Service utiliza *JDBC*¹² para estabelecer a comunicação. O JAX-WS¹³ é uma tecnologia que permite construir Web Services e os respetivos clientes, comunicando entre si usando XML. Apesar de a utilização de mensagens SOAP tornar-se complexa em algumas situações, o uso desta tecnologia facilita o desenvolvimento, abstraindo o programador de operações como a sua criação e análise. As invocações ao Web Service e respetivas respostas são convertidas automaticamente para mensagens SOAP [24]. O JAX-B¹⁴ é uma API java que facilita a manipulação de ficheiros XML através da utilização de anotações nos objetos e funcionalidades úteis para gerar automaticamente hierarquias de estruturas de dados a partir de ficheiros XSD¹⁵ [23]. Com as estruturas de dados criadas, o uso das anotações automatiza a geração de instâncias com a informação da mensagem XML. Os ficheiros XSD descrevem formalmente os elementos de um ficheiro XML [25].

Na Figura 30 são ilustrados alguns exemplos do WSDL que define o Web Service criado.

¹² **Java Database Connectivity** – Uma API padrão para estabelecer comunicação entre a linguagem Java e um vasto conjunto de bases de dados [22].

¹³ **Java API for XML Web Services**

¹⁴ **Java Architecture for XML Binding**

¹⁵ **Xml Schema Definition**

```

- <definitions name="NewResourcesWebServiceImplService" targetNamespace="com.criticalsoftware.wiccore.wise.business.ws.newresources">
  - <types>
    - <xs:schema targetNamespace="com.criticalsoftware.wiccore.wise.business.ws.newresources" version="1.0">
      <xs:element name="getComputerInformationById" type="tns:getComputerInformationById"/>
      <xs:element name="getComputerInformationByIdResponse" type="tns:getComputerInformationByIdResponse"/>
    - <xs:complexType name="getComputerInformationById">
      - <xs:sequence>
        <xs:element name="arg0" type="xs:int"/>
      </xs:sequence>
    </xs:complexType>
    - <xs:complexType name="getComputerInformationByIdResponse">
      - <xs:sequence>
        <xs:element minOccurs="0" name="return" type="tns:hardware"/>
      </xs:sequence>
    </xs:complexType>
    - <xs:complexType name="hardware">
      - <xs:sequence>
        <xs:element minOccurs="0" name="attributionDate" type="xs:dateTime"/>
        <xs:element minOccurs="0" name="defaultType" type="tns:newResourcesHardwareDefaultTypeEnum"/>
        <xs:element minOccurs="0" name="entity" type="tns:hardwareEntity"/>
        <xs:element name="id" type="xs:int"/>
        <xs:element minOccurs="0" name="location" type="tns:location"/>
        <xs:element minOccurs="0" name="manufacturer" type="tns:manufacturer"/>
        <xs:element minOccurs="0" name="model" type="tns:hardwareModel"/>
        <xs:element minOccurs="0" name="notes" type="xs:string"/>
        <xs:element minOccurs="0" name="renewDate" type="xs:dateTime"/>
        <xs:element minOccurs="0" name="serial" type="xs:string"/>
        <xs:element maxOccurs="unbounded" minOccurs="0" name="softwareList" nillable="true" type="tns:software"/>
        <xs:element minOccurs="0" name="state" type="tns:state"/>
        <xs:element minOccurs="0" name="type" type="tns:hardwareType"/>
        <xs:element name="userId" type="xs:int"/>
      </xs:sequence>
    </xs:complexType>
  </types>
  - <message name="NewResourcesWebServiceImpl_getComputerInformationById">
    <part element="tns:getComputerInformationById" name="getComputerInformationById"/>
  </message>
  - <message name="NewResourcesWebServiceImpl_getComputerInformationByIdResponse">
    <part element="tns:getComputerInformationByIdResponse" name="getComputerInformationByIdResponse"/>
  </message>
  - <operation name="getComputerInformationById" parameterOrder="getComputerInformationById">
    <input message="tns:NewResourcesWebServiceImpl_getComputerInformationById"/>
    <output message="tns:NewResourcesWebServiceImpl_getComputerInformationByIdResponse"/>
  </operation>
  - <operation name="getComputerInformationById">
    <soap:operation soapAction=""/>
    - <input>
      <soap:body use="literal"/>
    </input>
    - <output>
      <soap:body use="literal"/>
    </output>
  </operation>
  - <service name="NewResourcesWebServiceImplService">
    - <port binding="tns:NewResourcesWebServiceImplBinding" name="NewResourcesWebServiceImplPort">
      <soap:address location="http://127.0.0.1:8080/NewResourcesWebServiceImplService/NewResourcesWebServiceImpl"/>
    </port>
  </service>

```

1 - Objetos

2 - Mensagem

3 - Operação

4 - Nome e localização do Web Service

Figura 30 – Exemplo WSDL para o método `getComputerInformationById`

A secção um do exemplo apresenta parte do XSD, incluído no WSDL, que define os elementos das mensagens utilizadas na invocação do método `getComputerInformationById`. Verificamos que existem elementos complexos para o pedido e resposta, sendo a resposta constituída por zero ou mais elementos hardware. Na secção dois encontra-se a definição das mensagens, incluído o nome da mensagem e o elemento que a constitui, neste caso, os elementos definidos na secção um. Na secção três é definida a operação SOAP para o método em questão, definindo as mensagens de entrada e saída previamente definidas. Por último, na secção 4, são definidas as propriedades do Web Service, nomeadamente o nome e a sua localização.

De seguida é apresentado em detalhe a arquitetura do módulo.

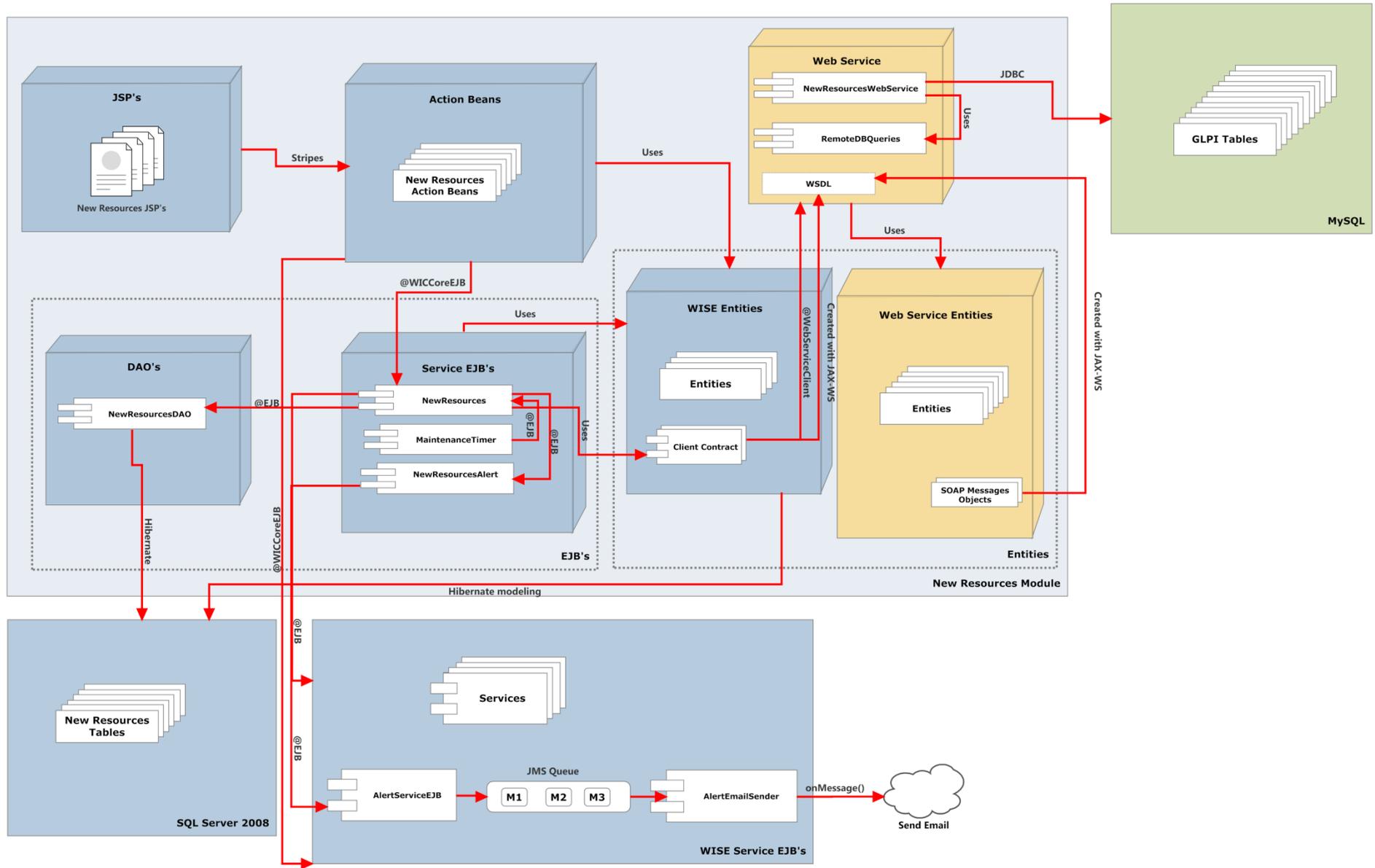


Figura 31 – Arquitetura do módulo de Recursos

4.2.7 Arquitetura

Nesta secção será detalhada a arquitetura do módulo incluindo os vários componentes e como interagem entre si. A sua leitura deve ser acompanhada da Figura 31 situada na página anterior.

Com a utilização da framework *stripes*, as JSP's encontram-se intimamente ligadas aos Action Beans. Como vimos anteriormente, os Action Beans recorrem à anotação `@WICCoreEJB` para comunicarem diretamente com os serviços. Com a implementação deste módulo foram criados três novos serviços, nomeadamente *NewResources*, *MaintenanceTimer* e *NewResourcesAlert*. Cada um destes serviços desempenha funções específicas no módulo. O *NewResourcesEJB* fornece toda a lógica computacional necessária para executar as funcionalidades pretendidas, sendo por isso o componente mais relevante. Utiliza não só os serviços do módulo, mas também, os restantes serviços disponíveis no WISE de forma a buscar informação necessária. A comunicação entre os serviços é feita através da anotação `@EJB` que permite referenciar serviços existentes através de *dependency injection*. A Figura 32 ilustra um exemplo desta anotação.

```
@EJB
private UserService userService;

@EJB
private CompanyService companyService;

@EJB
private DepartmentService departmentService;

@EJB
private NewResourcesDAO newResourcesDAO;

@EJB
private NewResourcesAlertService newResourcesAlertService;
```

Figura 32 – Exemplo anotação `@EJB`

O serviço *NewResourcesAlertEJB* é responsável pelo envio de alertas para a fila de mensagens (*JMS Queue*) no acontecimento de eventos alertáveis, por exemplo, a submissão de uma nova requisição. O serviço *MaintenanceTimer* foi criado de forma a comportar-se como um temporizador, que apenas irá realizar as suas operações de sincronização de informação entre os dois sistemas, uma vez por dia. Durante a sua execução são realizadas duas operações: verificar se existem licenças atribuídas no GLPI que correspondam a requisições pendentes de atribuição e, sincronizar as licenças atribuídas entre os dois sistemas.

Relativamente às entidades do módulo, existe uma separação clara entre as que são utilizadas no âmbito do WISE das que são utilizadas no Web Service. O WISE utiliza entidades que são uma representação direta das tabelas existentes no SQL Server e o contrato necessário para comunicar com o Web Service. O contrato é constituído por objetos que definem não só as mensagens de pedido e resposta, mas também permitem criar uma referência local que representa o Web Service (*proxy*). Este contrato foi gerado automaticamente utilizando a ferramenta *wsimport*¹⁶ disponível no JAX-WS.

As entidades utilizadas pelo Web Service incluem os objetos criados para representar a informação proveniente do GLPI, e os objetos que definem as mensagens de pedido e

¹⁶ Ferramenta utilizada para ler o WSDL e gerar os objetos necessários para comunicar com o Web Service.

resposta. Os objetos que definem as mensagens foram obtidos através da ferramenta *wsgen*¹⁷ disponível no JAX-WS.

A Figura 33 ilustra exemplos da utilização do Web Service.

```
private NewResourcesWebServiceImplService webservice;
private NewResourcesWebServiceImpl port;

@PostConstruct
void postConstruct(){
    webservice = new NewResourcesWebServiceImplService();
    port = webservice.getNewResourcesWebServiceImplPort();
}

private void sendSoftwareRequisitionNotification(int requisitionId,String initialApproverNotes,Locale locale) throws SystemAlertException{
    //Find DSI state
    WISERequisitionStateDefinition dsiState = this.findStateByKey("dsi");

    WISERequisitions requisition = this.findRequisitionById(requisitionId);
    Software softwareRequested = port.getSoftwareInformation(requisition.getSoftwareId(), requisition.getVersionId());
    Hardware usersHardware = port.getComputerInformationById(requisition.getComputerId());
}
```

Figura 33 – Exemplos de utilização do Web Service

Analisando os exemplos, podemos verificar que graças à utilização do JAX-B e JAX-WS, a utilização e manipulação de mensagens SOAP é abstraída do desenvolvimento. A comunicação com o Web Service é feita através do objeto local que representa o Web Service, no exemplo *port*. É possível realizar invocações ao Web Service e receber as respetivas respostas como se tratasse da utilização de um serviço disponível no WISE, ficando transparente ao programador o que realmente acontece por trás das invocações.

Por fim, o modelo de dados do módulo está retratado no diagrama ER da Figura 34.

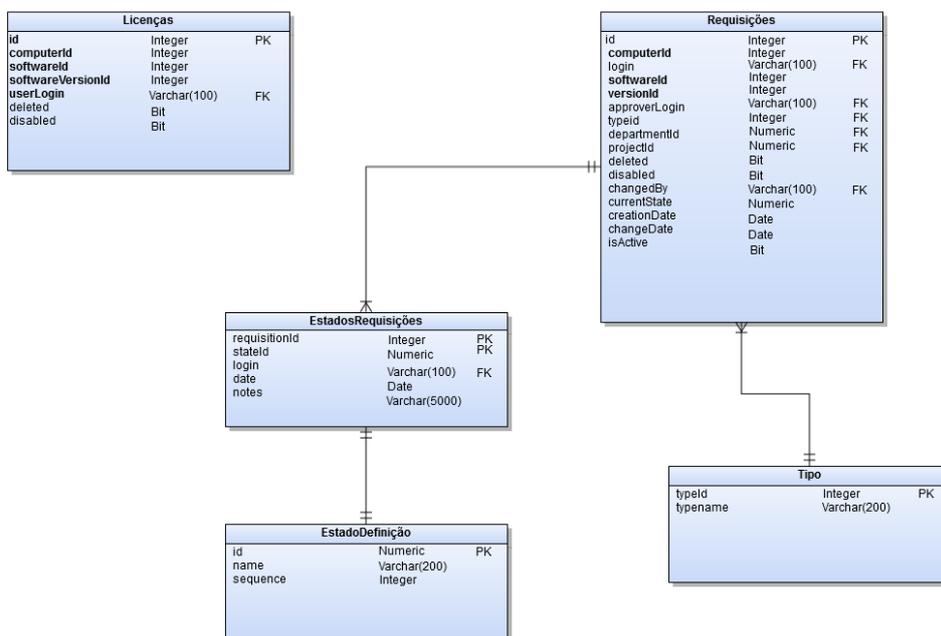


Figura 34 – Modelo ER do módulo Recursos

¹⁷ Ferramenta que utiliza a implementação do Web Service e gera os objetos necessários para realizar o seu *deploy*.

Essencialmente o módulo gere as requisições de software, armazenando a informação necessária na tabela de requisições. Nesta tabela encontram-se algumas chaves estrangeiras que indexam informação de outras tabelas, nomeadamente de projetos, áreas (departamento), colaboradores e tipo de requisição. Na tabela encontram-se realçados os campos que armazenam informação do GLPI, mais especificamente os identificadores numéricos do software, respetiva versão, e do computador ao qual se pretende atribuir a licença.

A tabela licenças permite armazenar informação de todas as licenças atribuídas aos computadores dos colaboradores. Esta informação, que deriva do GLPI, é utilizada para permitir aos utilizadores libertarem as suas licenças no WISE. Com isto evita-se mostrar licenças ainda válidas no GLPI mas já libertas no WISE. A informação consiste nos identificadores numéricos do software, versão, computador, utilizador e licença provenientes do GLPI, os restantes campos servem para designar a licença como válida ou inválida.

Os estados passados das requisições são armazenados na tabela *EstadosRequisições*, identificando-os através de uma chave primária composta pelos identificadores da requisição e estado. Por fim, as restantes tabelas contêm informação acerca dos vários estados possíveis que uma requisição pode tomar, *Submetida, Aprovador Inicial, Enviada a DSI, Atribuída, Cancelada, Rejeitada* e; os tipos de requisição existentes, *Geral, Projeto e Área*.

4.2. Módulo de Credenciações

O módulo de credenciações sofreu várias alterações com o intuito de acrescentar funcionalidades necessárias na gestão de credenciações e, adicionar campos informativos úteis para o utilizador. Assim, ao longo desta secção será apresentada a reestruturação que o módulo sofreu, começando por introduzir os requisitos especificados, os casos de uso definidos para os atores do módulo e concluindo com a sua arquitetura. É de salientar que as alterações realizadas devem ser consideradas como as lacunas que existiam no módulo anterior, e ao mesmo tempo, as soluções implementadas para as resolver.

Os requisitos definidos para este módulo podem ser divididos nas seguintes categorias: **Credenciações Válidas, Credenciações Pendentes, Pedido de Credenciação, Alertas, Relatórios, Histórico e Gestão de marca e nível**. De forma a complementar os requisitos são utilizados protótipos rápidos que permitem visualizar o aspeto gráfico desejado e, foram parcialmente utilizados na sua validação. É importante esclarecer que os requisitos apresentados são os mais relevantes nas categorias mencionadas, para obter a lista completa o leitor deve dirigir-se ao anexo **E**. Uma vez que se trata de uma reestruturação serão salientadas as alterações realizadas.

4.2.1 Credenciações Válidas

Nesta secção são encontram-se os requisitos associados às credenciações válidas, definindo a informação a apresentar e as funcionalidades pretendidas.

SRS-RES-10-Listar Colaboradores com Credenciações Válidas							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-11-01
Type:	Funcional						
Deve ser possível listar todos os colaboradores com credenciações válidas. A lista deve conter os seguintes campos: <ul style="list-style-type: none"> ▪ Nome do Colaborador ▪ Empresa ▪ Área ▪ Marca ▪ Nível ▪ Tipo - Corporativa/Individual ▪ Data do pedido ▪ Data de expiração 							

Tabela 28 – Credenciações Válidas

A Figura 35 ilustra o protótipo definido para a lista de credenciações válidas.

Nome	Empresa	Área	Marca	Nível	Tipo	Data do Pedido	Data de Expiração
José Veríssimo	Critical Software	Delivery - Engineering	Nacional	Secreto	Individual	13-10-2013	13-10-2014
	Critical Software	Delivery - Engineering	Nacional	Secreto	Corporativa	13-10-2013	13-10-2014

Figura 35 - Lista de Credenciações Válidas

A lista contém todas as credenciações válidas atribuídas aos colaboradores da organização. De forma a tornar a lista mais completa, foram adicionados os campos Empresa, Área, Data do Pedido e Tipo de credenciação. De seguida são apresentados os filtros que se podem aplicar na lista.

SRS-RES-011-Filtrar Lista de Colaboradores							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-11-01
Type:	Funcional						
<p>Deve ser possível filtrar os resultados obtidos na lista com base nos seguintes filtros:</p> <ul style="list-style-type: none"> ▪ Nome/Login do colaborador ▪ Empresa ▪ Área ▪ Marca ▪ Nível ▪ Tipo – Checkbox. Se estiver activa apenas credenciações corporativas serão listadas. ▪ Data de Pedido ▪ Data de Expiração 							

Tabela 29 – Filtros da lista de Credenciações Válidas

The screenshot shows a web form with the following elements:

- Nome: [Text input field]
- Empresa: [Text input field with dropdown arrow]
- Área: [Text input field with dropdown arrow]
- Marca: [Text input field with dropdown arrow]
- Nível: [Text input field with dropdown arrow]
- Data do Pedido: [Date input field with calendar icon]
- Data de Expiração: [Date input field with calendar icon]
- Credenciação Corporativa
- Procurar [Submit button]

Figura 36 - Filtros da lista de Credenciações Válidas

Os filtros permitem obter listas personalizadas que contêm apenas a informação desejada. Os filtros adicionados foram Empresa, Área, Credenciação Corporativa, Data do pedido e Data de expiração.

4.2.2 Credenciações Pendentes

Esta secção apresenta os requisitos associados às credenciações pendentes, definindo a informação a apresentar e as funcionalidades pretendidas.

SRS-RES-020-Listar Credenciações Pendentes							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-11-01
Type:	Funcional						
<p>Deve ser possível listar todos os colaboradores com credenciações pendentes. A lista deve conter os seguintes campos:</p> <ul style="list-style-type: none"> ▪ Nome do Colaborador ▪ Tipo – Corporativa / Individual ▪ Empresa 							

SRS-RES-020-Listar Credenciações Pendentes

- Área
- Marca
- Nível
- Estado actual
- Data do pedido

Tabela 30 – Lista de Credenciações Pendentes

SRS-RES-025-Rejeitar credenciação pendente

Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-11-01
Type:	Functional						

Deve ser possível rejeitar um pedido de credenciação pendente. Só é possível rejeitar o pedido nos seguintes estados:

- Pedido Submetido
- Aprovado pelo OS
- Documentação entregue ao colaborador
- Documentação recebida do colaborador
- Documentação validada

Apenas o responsável pelo pedido e utilizadores com role *wiseosrole* é que podem rejeitar.

Esta funcionalidade encontra-se presente numa coluna dedicada da tabela de listagem das Credenciações Pendentes.

Tabela 31 – Rejeitar Credenciação Pendente

Primeiro/Anterior 1 2 3 4 5 6 7 8 Próximo/Último X registos encontrados, registos de 1 a 50.

Nome	Tipo	Empresa	Área	Marca	Nível	Estado Actual	Data do Pedido	
José Verissimo	Individual	Critical Software	Delivery - Engineering	Nacional	Secreto	Aprovado pelo OS	13-10-2014	⊙
Hugo Barrigas	Individual	Critical Software	Delivery - Engineering	Nacional	Secreto	Documentação entregue ao colaborador	13-10-2013	⊙ Rejeitar
Daniela Domingues	Individual	Critical Software	Delivery - Engineering	Nacional	Secreto	Documentação Recebida do colaborador	13-10-2013	⊙
Diana Gomes	Individual	Critical Software	Delivery - Engineering	Nacional	Secreto	Pedido enviado ao GNS	13-10-2013	

Exportar: [CSV](#) | [Excel](#) | [XML](#) | [PDF](#)

Figura 37 – Lista de Credenciações Pendentes

A lista de credenciações pendentes contém todas as credenciações que têm um processo aberto. De forma a tornar a lista mais completa e útil aos utilizadores, foram adicionados os campos Tipo, Empresa, Área e Data do pedido. Através da lista é possível rejeitar credenciações e, abrir o processo da credenciação visualizando os seus detalhes. De seguida são apresentados os filtros que podem ser aplicados na lista.

SRS-RES-021-Filtrar lista de credenciações Pendentes							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-11-01
Type:	Functional						
<p>Deve ser possível filtrar a lista através dos seguintes campos:</p> <ul style="list-style-type: none"> ▪ Nome/Login do colaborador ▪ Empresa ▪ Área ▪ Marca ▪ Nível ▪ Estado actual ▪ Tipo – Checkbox. Se estiver activa apenas credenciações corporativas serão listadas. ▪ Data do Pedido 							

Tabela 32 – Filtros aplicáveis na lista de Credenciações Pendentes

The screenshot shows a search filter interface. It contains the following elements:

- Nome: [Text input field]
- Empresa: [Text input field with dropdown arrow]
- Área: [Text input field with dropdown arrow]
- Marca: [Text input field with dropdown arrow]
- Nível: [Text input field with dropdown arrow]
- Estado Actual: [Text input field with dropdown arrow]
- Data do Pedido: [Date input field with slashes and a calendar icon]
- Credenciação Corporativa
- Procurar button

Figura 38 – Filtros aplicáveis na lista de Credenciações Pendentes

Os filtros disponíveis permitem obter listas de credenciações pendentes personalizadas contendo só a informação desejada. Os filtros adicionados foram Empresa, Área, Credenciação Corporativa, Data do pedido e Estado atual. De seguida são apresentados os requisitos associados aos detalhes de uma requisição pendente.

SRS-RES-023-Detalhes da credenciação pendente							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-11-01
Type:	Functional						
<p>Clicando num item da lista deve ser possível apresentar os detalhes da credenciação pendente através de uma página dedicada. Esta página deve conter a seguinte informação:</p> <ul style="list-style-type: none"> ▪ Nome do Colaborador ▪ Área ▪ Empresa ▪ Marca 							

SRS-RES-023-Detalhes da credenciação pendente

- Nível
- Centro de Custos
- Submetido por – Colaborador responsável pelo pedido
- Tipo
- Mapa de estados

Tabela 33 – Detalhes do processo de uma Credenciação Pendentes

SRS-RES-024-Mapa de Estados

Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-11-01
Type:	Funcional						

Neste mapa será possível visualizar os estados efectuados num pedido de credenciação, que são os seguintes:
 Pedido Submetido -> Aprovado pelo OS -> Documentação entregue ao colaborador -> Documentação recebida do colaborador -> Documentação validada -> Pedido entregue ao GNS -> Credenciação aprovada pelo GNS

O estado atual será destacado do resto. Os estados passados serão cinzento-escuro, enquanto, que os estados ainda por visitar serão cinzento-claro. É necessário indicar o responsável por cada estado, a data em que terminou e notas caso existam. As notas são mostradas através de uma tooltip quando se passa em cima do estado.

No último estado, Credenciação aprovada pelo GNS, é necessário existir um campo obrigatório para adicionar a data de expiração da credenciação.

Tabela 34 – Mapa de Estados do processo de Credenciação Pendente

Detalhes Credenciação Pendente

Credenciações | Relatórios

Detalhes da Credenciação

Nome do Colaborador:
 Empresa:
 Área:
 Marca:
 Nível:
 Centro de Custos:
 Responsável:
 Tipo:

Mapa de Estados

Notas:

Pedido Submetido Aprovado pelo OS Documentação entregue ao colaborador Documentação recebida ao colaborador Documentação validada Pedido entregue ao GNS Credenciação aprovada pelo GNS

Paulo Lourenço ✉
13/10/2013

Associar centro de custo: _____

Centro de Custo: _____

Caixa auto-complete. Este campo só se encontra disponível no estado Aprovado pelo OS e a utilizadores com role wiseosrole.

Notas: _____

Figura 39 - Detalhes de uma Credenciação Pendente

Nos detalhes do processo de uma credenciação pendente foi necessário acrescentar os campos informativos Empresa, Área e Tipo; implementar uma nova funcionalidade que permite a utilizadores com a devida autorização alterar o centro de custos associado, a rejeição de uma credenciação foi limitada até ao estado Documentação validada e, por último, o aspeto gráfico do mapa de estados foi alterado.

4.2.3 Alertas

Nesta secção são apresentados os requisitos associados aos alertas do módulo, definindo que operações devem ser alertáveis e que informação deve ser apresentada nos respetivos alertas.

O módulo já continha um conjunto de alertas disponíveis, no entanto, existia a necessidade de transformar mais funcionalidades em eventos alertáveis para avisar os utilizadores da sua ocorrência. Assim, foram implementados os três novos alertas definidos na Tabela 35. A informação que deve constar em todos os alertas foi redefinida e encontra-se na Tabela 36.

SRS-RES-030-Alertas							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-11-01
Type:	Funcional						
<p>O módulo deve suportar os seguintes alertas:</p> <ul style="list-style-type: none"> ▪ Cancelamento: Um e-mail deve ser enviado aos colaboradores com role <i>wiseosrole</i> acerca de requisições ativas canceladas. ▪ Rejeição: Um e-mail deve ser enviado aos intervenientes do processo (Colaborador alvo e criador do processo) e colaboradores com role <i>wiseosrole</i> acerca de pedidos rejeitados. ▪ Renovação: Sempre que é efectuada uma renovação de credenciação deve ser enviado um e-mail para os colaboradores com role <i>wiseosrole</i>. 							

Tabela 35 – Novos alertas do módulo de Credenciações

SRS-RES-031-Detalhes dos Alertas							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-11-01
Type:	Funcional						
<p>O e-mail de alerta deve conter os seguintes parâmetros:</p> <ul style="list-style-type: none"> ▪ Nome do colaborador ▪ Empresa ▪ Área ▪ Marca ▪ Nível ▪ Tipo de credenciação – Individual/Corporativa ▪ Data de expiração ▪ Data do pedido ▪ Data de cancelamento/rejeição ▪ Estado do mapa de estados ▪ Responsável ▪ Notas 							

Tabela 36 – Detalhes dos alertas

4.2.4 Pedido de Requisição

Nesta secção encontram-se os requisitos associados ao pedido de uma nova credenciação.

SRS-RES-040-Pedido de Credenciação							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-11-01
Type:	Functional						
Deve ser possível efectuar o pedido de uma nova credenciação. Este pedido encontra-se dividido em duas secções, secção Colaboradores e secção Propriedades.							

Tabela 37 – Pedido de Credenciação

SRS-RES-041-Secção Colaboradores							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-11-01
Type:	Functional						
<p>Nesta secção deve ser possível seleccionar o(s) colaborador(es) afectado(s) pelo pedido. De forma a filtrar os utilizadores existem os seguintes campos:</p> <ul style="list-style-type: none"> ▪ Empresa – Lista que contém todas as empresas. ▪ Área – Lista que contém todas as áreas. Esta lista depende da empresa escolhida. <p>Após ser escolhida a empresa e área, devem ser listados os colaboradores correspondentes. Deve ser possível seleccionar os colaboradores da lista a que se pretende atribuir a credenciação.</p> <p>Caso a checkbox Credenciação Corporativa, referida no SRS-RES-043, esteja seleccionada, o campo Empresa é de preenchimento obrigatório. Caso contrário, embora os campos não sejam de preenchimento obrigatório, é obrigatório seleccionar no mínimo um colaborador.</p>							

Tabela 38 – Secção Colaboradores do pedido de Credenciação

SRS-RES-043-Secção Propriedades							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-11-01
Type:	Functional						
<p>Nesta secção é possível definir as propriedades do pedido. Estas propriedades são:</p> <ul style="list-style-type: none"> ▪ Marca* – NATO,NACIONAL,EU ▪ Nível* – Depende da marca. <ul style="list-style-type: none"> ✓ NATO:Confidential; Cosmic Top Secret; Cosmic Top Secret Atomal; Nato Secret ✓ NACIONAL: Secreto; MuitoSecreto ✓ EU: Secret; Top Secret ▪ Centro de Custos* – A lista depende dos acessos que o utilizador, que faz o pedido, tem aos centros de custos. ▪ Credenciação Corporativa – Checkbox. Se estiver activa indica que a credenciação é de empresa. ▪ Notas <p>Os valores base da Marca e Nível são configuráveis, consultar SRS-RES-080.</p> <p>Os campos com “*” são de preenchimento obrigatório.</p>							

Tabela 39 – Secção Propriedades do pedido de Credenciação

Colaboradores

Empresa:

Área:

Colaborador
 Colaborador
 Colaborador
 Colaborador
 Colaborador

<<
<
>
>>

Propriedades

Marca*:

Nível*:

Centro de Custos*:

Credenciação Corporativa:

Notas:

Figura 40 – Pedido de Credenciação

No formulário da Figura 40, o utilizador poderá realizar pedidos de novas credenciações. Foram realizados pequenos ajustes no formulário, nomeadamente a remoção do campo *Data Esperada* que não era utilizado e, acrescentada a possibilidade de criar credenciações corporativas através da *checkbox* Credenciação Corporativa.

4.2.5 Relatórios

Esta secção contém os requisitos associados aos relatórios de credenciações, definindo a informação necessária nos relatórios bem como as funcionalidades pretendidas para esta secção. É importante mencionar que esta secção foi completamente reestruturada.

SRS-RES-050-Relatórios							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-11-01
Type:	Functional						
Deve ser possível obter relatórios sobre credenciações com os seguintes campos: <ul style="list-style-type: none"> ▪ Nome do colaborador ▪ Área ▪ Empresa ▪ Marca ▪ Nível ▪ Centro de custo ▪ Data do pedido ▪ Data de expiração ▪ Estado ▪ Tipo 							

Tabela 40 – Relatórios de Credenciações

SRS-RES-051-Filtros dos Relatórios							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-11-01
Type:	Funcional						
<p>Os seguintes filtros podem ser aplicados nos relatórios:</p> <ul style="list-style-type: none"> ▪ Nome do colaborador ▪ Empresa ▪ Área ▪ Marca ▪ Nível ▪ Data de início ▪ Data de fim ▪ Estado – Validada, Expirada, Cancelada, Rejeitada ▪ Credenciação Corporativa – Checkbox. Se seleccionada lista apenas credenciações de empresa. ▪ Centro de Custos 							

Tabela 41 – Filtros a aplicar nos relatórios de Credenciações

SRS-RES-056-Renovar Credenciação							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-11-01
Type:	Funcional						
<p>Após obter o relatório será possível renovar credenciações listadas.</p> <p>Renovando uma credenciação o sistema cria automaticamente um processo idêntico ao de um novo pedido. A informação do pedido anterior é reutilizada na renovação. Neste processo será necessário cumprir o requisito SRS-RES-059.</p> <p>Esta funcionalidade encontra-se disponível a utilizadores com role <i>wiseosrole</i>, numa coluna dedicada na lista.</p> <p>Esta opção estará inactiva caso já exista um pedido de renovação para a credenciação em questão.</p>							

Tabela 42 – Renovar Credenciação nos relatórios

SRS-RES-057-Cancelar Credenciação							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-11-01
Type:	Funcional						
<p>Após obter o relatório será possível cancelar as credenciações ativas listadas. Esta funcionalidade encontra-se disponível ao utilizador responsável pela credenciação e utilizadores com role <i>wiseosrole</i>, numa coluna dedicada na lista.</p> <p>Credenciações corporativas não podem ser canceladas, apenas renovadas.</p>							

Tabela 43 – Cancelar Credenciações nos relatórios

SRS-RES-058-Visualizar histórico de Credenciações							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-11-01
Type:	Funcional						
<p>Clicando num colaborador da lista, deve ser possível reencaminhar o utilizador para a tab <i>Credenciações</i> no perfil do colaborador.</p>							

Tabela 44 – Visualizar histórico de Credenciações

Nome	Empresa	Área	Marca	Nível	Centro de Custos	Data do Pedido	Data de Expiração	Estado	Renovar	Cancelar
Verissimo	Critical Software	Delivery - Engineering	Nacional	Secreto	17598 - Credenciação Nato	13-10-2013	13-10-2014	Válida	⊕	⊖
Hugo Barrigas	Critical Software	Delivery - Engineering	Nacional	Secreto	17598 - Credenciação Nato	13-10-2013	13-10-2014	Cancela	⊕	⊖
Daniela Domingues	Critical Software	Delivery - Engineering	Nacional	Secreto	17598 - Credenciação Nato	13-10-2013	13-10-2014	Expirada	⊕	⊖

Exportar: [CSV](#) | [Excel](#) | [XML](#) | [PDF](#)

Figura 41 – Relatórios de Credenciações

Nome:

Empresa:

Área:

Marca:

Nível:

Data de início:

Data de fim:

Estado:

Credenciação Corporativa

Figura 42 – Filtros aplicáveis nos relatórios

Os relatórios contêm informação relevante de todas as credenciações existentes no sistema, facilitando desta forma a sua gestão e controlo. Nesta secção são disponibilizadas operações úteis ao utilizador, nomeadamente renovar e cancelar credenciações e, visualizar o histórico de credenciações de um colaborador. A informação presente no relatório pode ser personalizada utilizando os filtros disponíveis na Figura 42.

Na Tabela 45 encontra-se definido o requisito para o Histórico de Credenciações.

SRS-RES-070-Histórico de Credenciações							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-11-01
Type:	Functional						
<p>Na tab <i>Credenciações</i> do perfil do colaborador deve ser possível visualizar as suas credenciações actuais, bem como credenciações passadas (canceladas, expiradas ou rejeitadas). A informação apresentada é:</p> <ul style="list-style-type: none"> ▪ Marca ▪ Nível ▪ Data de expiração ▪ Tipo ▪ Estado 							

Tabela 45 – Histórico de Credenciações

Credenciação

1						2 registos encontrados	
Marca	Nível	Data de expiração	Tipo	Estado		Cancelar	
NACIONAL	Secreto	25/05/2014	Individual	Válida	⊗		
EU	Secret	25/05/2014	Individual	Válida	⊗		
1						2 registos encontrados	

Figura 43 – Histórico de Credenciações

O histórico de credenciações de um colaborador encontra-se numa secção específica do seu perfil. Para o efeito será utilizada uma lista que inclui todas as suas credenciações válidas, expiradas, rejeitadas ou canceladas. Ainda, para utilizadores com as devidas permissões, será possível cancelar as credenciações válidas.

4.2.6 Gestão de Marca e Nível

Antes de realizar a reestruturação do módulo, os valores utilizados na marca e nível associados às credenciações eram estáticos. De forma a adicionar ou editar estes valores, era necessário comunicar com a equipa de manutenção do WISE para alterarem a informação diretamente na base de dados. Assim, com o intuito de facilitar e agilizar a gestão desta informação, foi adicionada a secção de administração disponível apenas a utilizadores com as permissões necessárias. Nesta secção será possível alterar os valores da marca e nível utilizados. A Tabela 46 define o requisito para esta secção.

SRS-RES-080-Gestão de Marca e Nível							
Version:	1.0	Phase:	1.0	State:	Proposed	Date:	2013-11-01
Type:	Functional						
Deve ser possível adicionar e editar os valores base da Marca e Nível das credenciações numa tab dedicada no módulo.							
Esta tab só deve estar disponível a utilizadores com role <i>wiseosrole</i> .							

Tabela 46 - Gestão de Marca e Nível

O protótipo rápido para definir o aspeto gráfico desta secção não foi definido, dando liberdade ao estagiário no aspeto gráfico a utilizar desde que respeitasse o padrão utilizado nos restantes módulos.

4.2.7 Casos de Uso

Os diagramas desta secção têm como objetivo complementar os requisitos e protótipos apresentados anteriormente, identificando os diferentes atores e respetivos acessos às operações descritas. O módulo de credenciações tem visibilidade limitada e, por este motivo, apenas os atores indicados têm acesso.

Assim como no módulo de recursos, a interação com o módulo é bastante simples, cada caso de uso representa uma página com um formulário ou, informação na qual é possível aplicar filtros e realizar operações. A transição de estado implica a ação de carregar num botão ou, clicar num link para realizar uma operação. No total existem sete atores que identificam os diferentes utilizadores e respetivos privilégios no módulo.

A Figura 44 ilustra o diagrama de casos de uso para os atores *wiseosrole*, *wiseclearancerole*, *wiserhrole* e *wisemrole*.

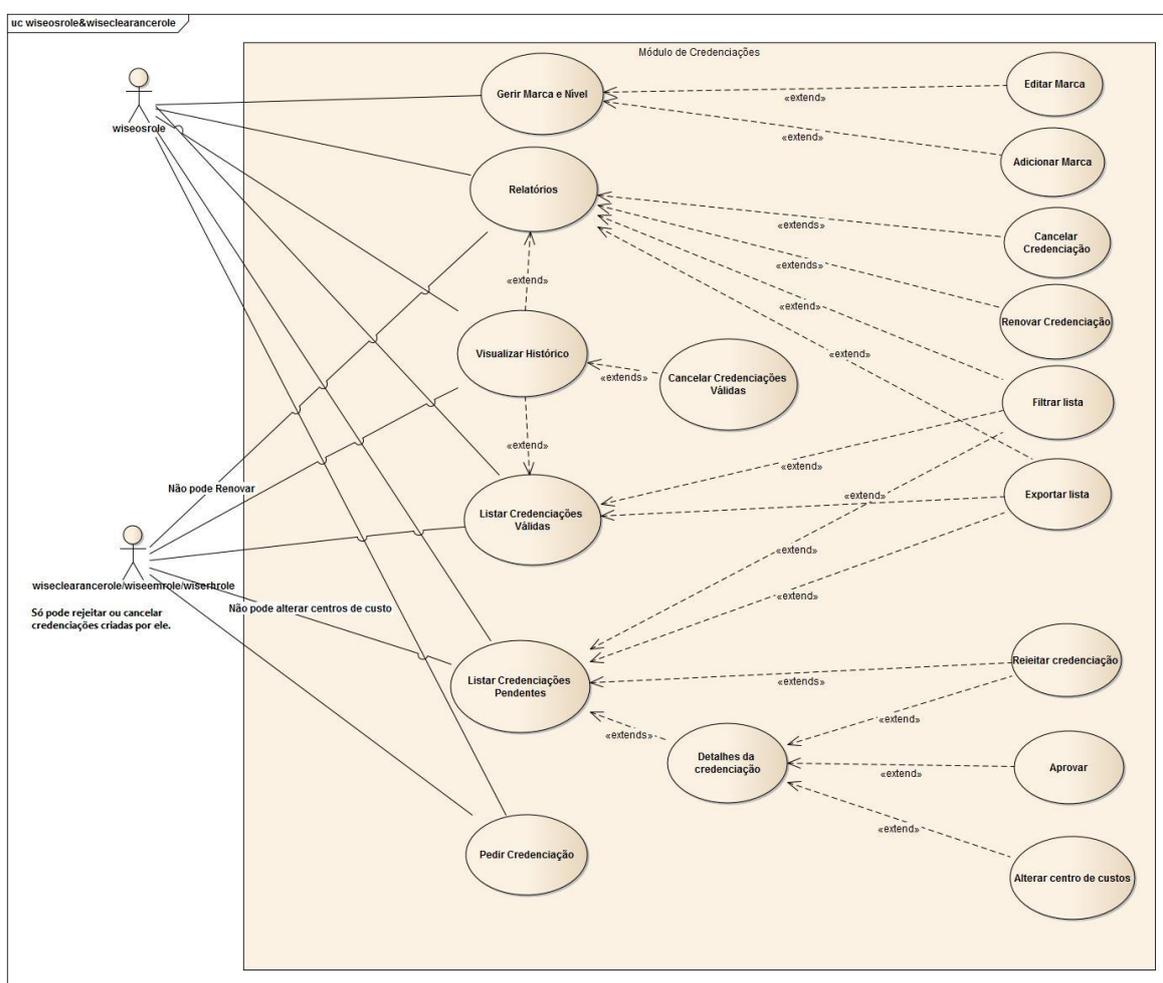


Figura 44 – Casos de uso *wiseosrole*, *wiseclearancerole*, *wiserhrole* e *wisemrole*

Estes atores são responsáveis pela gestão e controlo das credenciações existentes. O ator *wiseosrole* identifica o oficial de segurança e, tem acesso a todas as funcionalidades e informação do módulo. Os restantes atores têm acesso a grande parte das funcionalidades, no entanto, são impostas algumas limitações: não podem renovar credenciações nem aceder à secção de administração e, as operações de cancelar e rejeitar são limitadas às credenciações que eles próprios criaram. Nos detalhes de uma credenciação pendente,

apenas o oficial de segurança tem acesso ao formulário que permite alterar o centro de custos associado. De forma a simplificar a usabilidade do módulo, foram colocadas novamente as operações de cancelar, rejeitar e renovar, numa coluna dedicada das respetivas listas.

Na Figura 45 encontra-se o diagrama de casos de uso dos restantes atores do módulo.

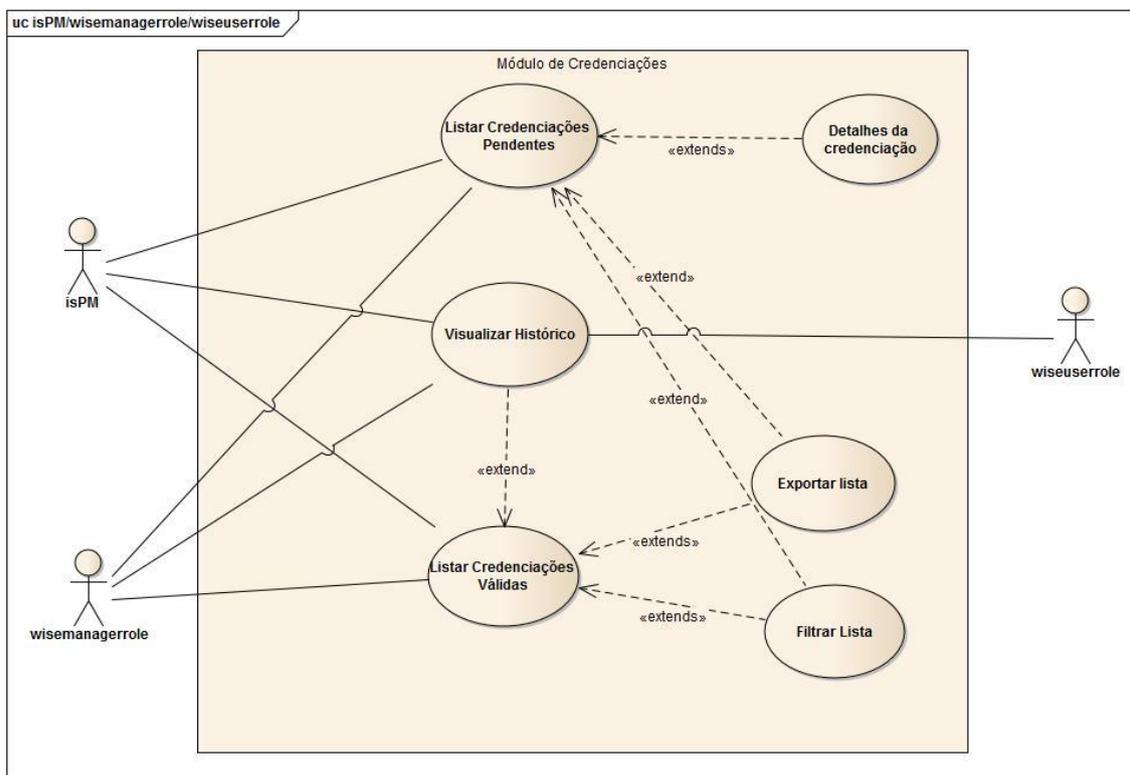


Figura 45 – Casos de uso isPM, wisemanagerrole, wiseuserrole

No diagrama encontram-se os atores que têm acesso ao módulo apenas para consultar a informação. O ator *isPM* identifica os utilizadores que desempenham a função de gestor de projetos, por isso, a consulta de informação pode ser útil caso o seu projeto implique a utilização de credenciações. O mesmo princípio pode ser aplicado ao ator *wisemanagerrole*, mas neste caso, o âmbito é área. Por último, o ator *wiseuserrole* só tem acesso à sua informação através do seu perfil.

4.2.8 Arquitetura

Na Figura 46 é apresentada a arquitetura do módulo incluindo os vários componentes e como interagem entre si.

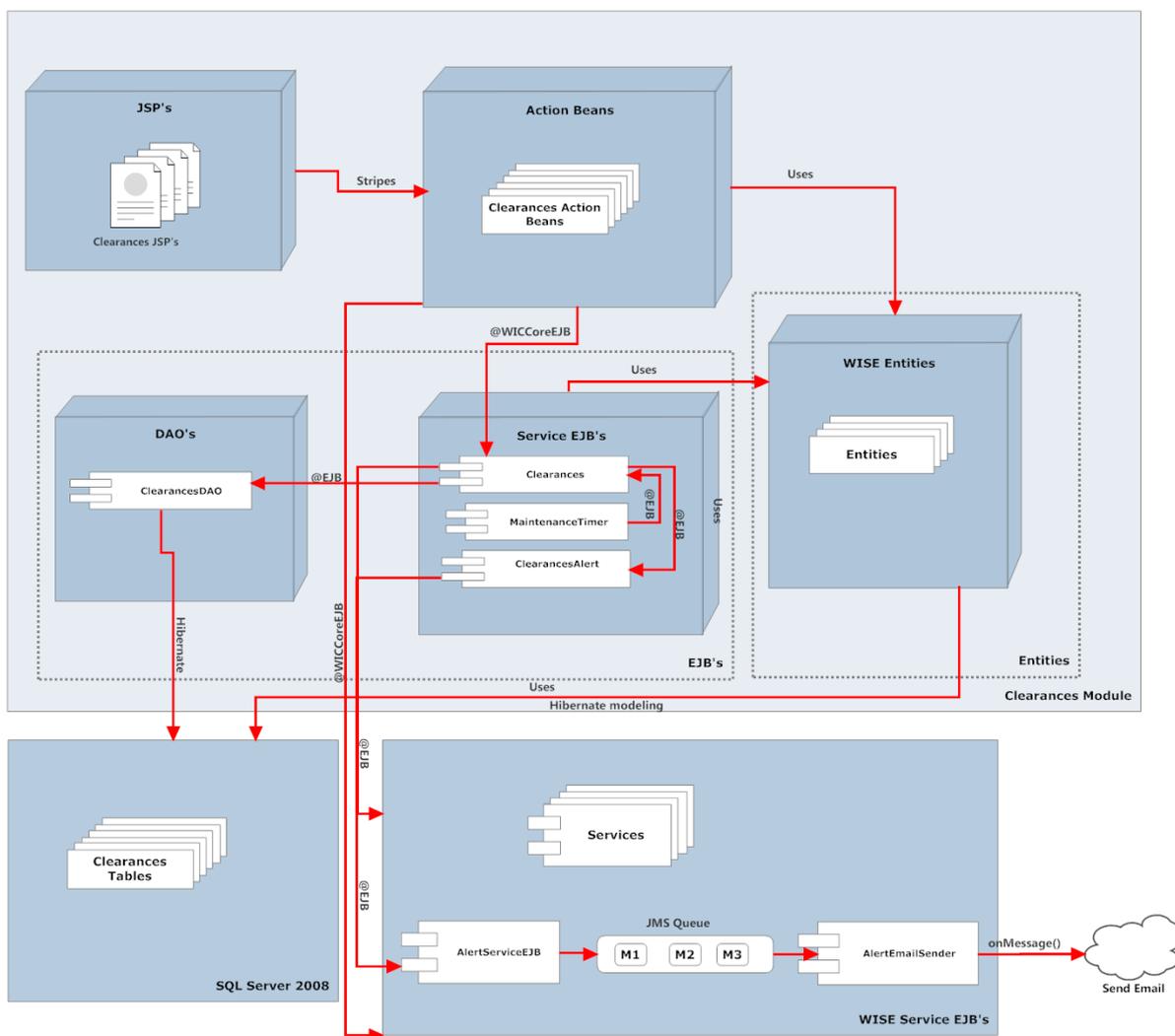


Figura 46 – Arquitetura do módulo de Credenciações

A arquitetura do módulo segue a abordagem de três camadas mencionada no capítulo 2. Analisando a figura verificamos que é muito semelhante à arquitetura do módulo de recursos, por isso, aplica-se o mesmo princípio à função dos componentes e como comunicam entre si.

Durante a reestruturação foi necessário alterar o modelo de dados utilizado no módulo. Durante a alteração foram acrescentados novos campos nas tabelas, criada uma nova tabela que armazena o tipo de credenciações e, modificada a relação existente entre a tabela de processos e credenciações. As Figuras 47 e 48 ilustram, respectivamente, o diagrama ER anterior e o atual.

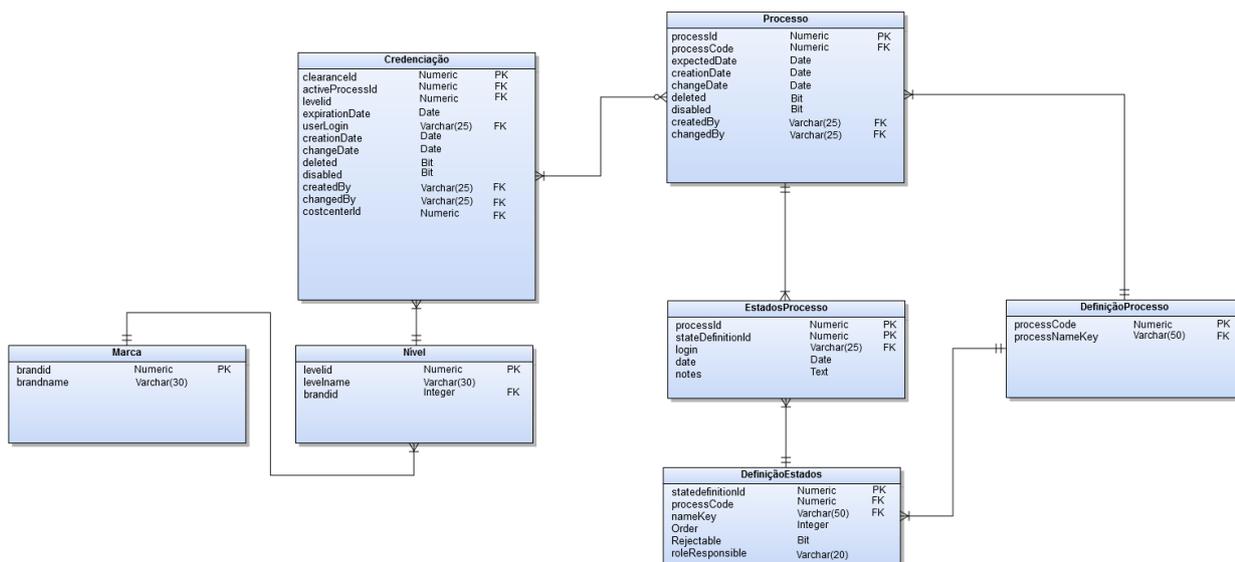


Figura 47 – Modelo ER antigo do módulo de Credenciações

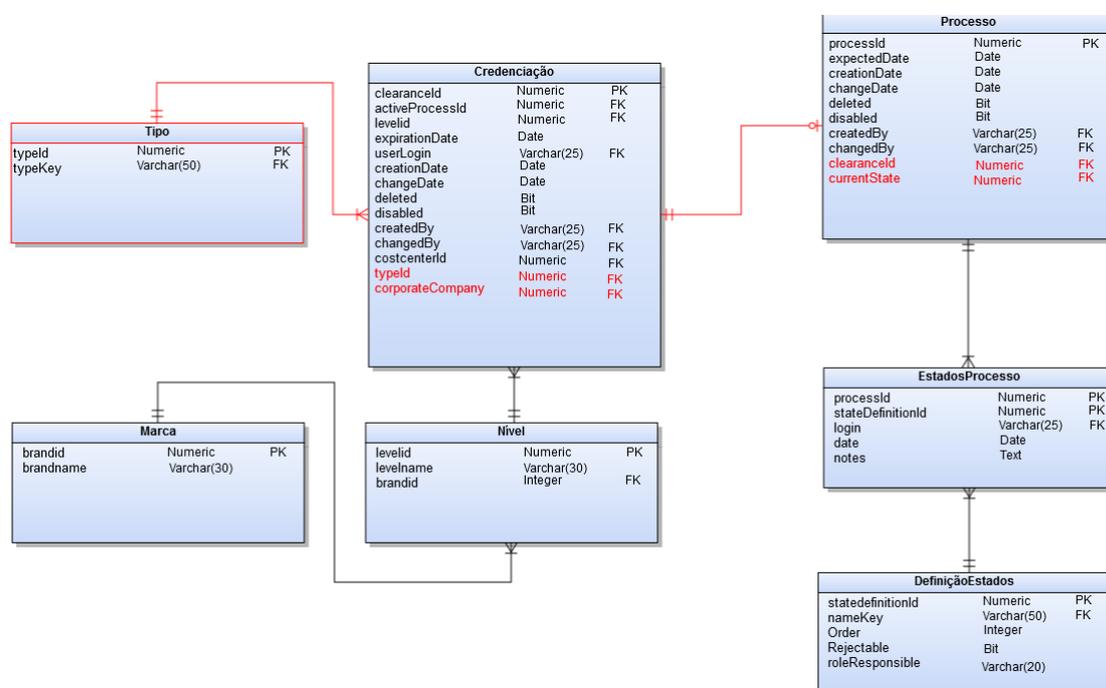


Figura 48 – Novo modelo ER do módulo de Credenciações

No modelo da Figura 46 são realçadas as alterações através da cor vermelha. A relação entre as credenciações e respetivos processos foi redefinida de *many-to-many* para *one-to-one*. Esta alteração foi realizada após concluir que, logicamente, a relação de um para um era a mais adequada dado que um processo deve ter apenas uma, ou zero, credenciações associadas. A informação relativa aos tipos de processo foi eliminada dado não ser necessário diferenciar entre pedidos e renovações de credenciações. Para retratar os tipos de credenciação, Individual e Corporativa, foi necessário adicionar uma tabela que se relaciona diretamente com as credenciações. Por último, foram também adicionados campos nas tabelas Credenciação e Processo.

Para adaptar a informação existente ao novo modelo definido, foram desenvolvidos dois scripts em *Transact-SQL* que automatizam o processo de migração. Caso o leitor pretenda analisar os scripts poderá encontrá-los no anexo F.

Devido às alterações realizadas foi necessário analisar todos os componentes do módulo e adaptá-los ao novo modelo de informação. Este processo envolveu a criação de novas queries, as existentes foram remodeladas consoante as necessidades, as entidades sofreram alterações para refletirem os novos campos e relações, a lógica computacional situada nos serviços foi reestruturada e alterada para refletirem as novas operações e, por último, as alterações concluíram nas JSP's e respetivos Action Beans.

4.3. Validação

Como mencionado no capítulo dois, à medida que a fase de desenvolvimento se aproxima do fim é iniciada a fase de validação. Antes de dar início a esta fase, os documentos de requisitos foram revistos e atualizadas de acordo com as alterações e decisões efetuadas durante o desenvolvimento. De seguida foram realizadas *Code Review Meetings* com o objetivo de validar o código desenvolvido e, verificar se segue os padrões estabelecidos para a implementação de aplicações na linguagem Java. Após alterar o código consoante os comentários da reunião, foram definidos testes de validação de acordo com os requisitos de software. Os testes especificados garantem que as soluções desenvolvidas são robustas e realizam todas as funcionalidades pretendidas. Como os testes foram especificados de acordo com o documento de requisitos, estão agregados conforme as suas categorias. No total foram desenvolvidos dois documentos de testes, um para cada módulo, e podem ser encontrados nos anexos **G** e **H**. Cada documento inclui uma matriz de rastreabilidade que permite verificar quais os testes que validam um requisito em específico.

Durante o desenvolvimento do Web Service foi utilizada a ferramenta *SOAP UI* para validar os métodos de consulta à medida que eram construídos. Desta forma isolou-se a implementação da camada de integração dos restantes componentes, permitindo validar que os métodos disponíveis estavam completamente funcionais antes de serem utilizados.

Graças a execução dos testes especificados nos documentos, foram encontrados erros que corrompiam a utilização dos módulos. Após resolver os problemas, voltou-se novamente a executar todos os testes para garantir que não só os problemas encontrados estavam resolvidos, mas também, que as soluções implementadas não corrompiam mais nenhuma funcionalidade. Uma vez terminada a execução dos testes, foram obtidas as soluções finais prontas para serem colocadas no ambiente de pré-produção. Neste ambiente é utilizada informação real e as entidades responsáveis dos módulos irão validar as soluções implementadas.

5 Resultados do Desenvolvimento

Durante este capítulo são apresentados os resultados obtidos, descrevendo em detalhe as ações que o utilizador pode realizar e o comportamento do sistema quando realizadas. É dada uma visão geral das operações através de diagramas de sequência, justificando também algumas diferenças que existem relativamente aos requisitos.

5.1. Módulo de Recursos

A visualização de software e hardware é um dos objetivos principais deste módulo. O layout definido permite visualizar de forma clara e intuitiva a informação destes componentes. Na Figura 49 é apresentada a página inicial do módulo.

TAREFAS
COLABORADORES
PROJECTOS
NEGÓCIO
TENDERING
QUALIDADE
RECURSOS
EVENTOS
VIAGENS
REUTILIZAÇÃO

Tem 69 novos alertas

Hardware

O Meu Hardware

Tipo :
 Computadores

1
Um item encontrado.

Modelo	Fabricante	Tipo	Associado a	Data Atribuição	Data Renovação	Notas	Localização	Estado
ThinkPad T530	Lenovo	Laptop	CSW	21/01/2014	30/11/2013	WO n.º 67345	Coimbra	Active

1
Um item encontrado.

Opções de exportação: [CSV](#) | [Excel](#) | [XML](#) | [PDF](#)

Software

O MEU SOFTWARE
LISTA DE SOFTWARE
RELATÓRIOS
REQUISIÇÃO DE SOFTWARE

O Meu Software

1
3 itens encontrados.

Equipamento	Nome Software	Versão Software	Número de Série	Comentários	Data Expiração
Lenovo ThinkPad T530 (PK1DNNC)	Enterprise Architect	Corporate	2008433018		
Lenovo ThinkPad T530 (PK1DNNC)	Windows Server	2012 Datacenter			
Lenovo ThinkPad T530 (PK1DNNC)	Visual Studio	Professional with MSDN 2012	BPQCR-24Q48-3QKTT-MF64K-YY3DV		

1
3 itens encontrados.

Opções de exportação: [CSV](#) | [Excel](#) | [XML](#) | [PDF](#)

Requisições Pendentes

AS MINHAS REQUISIÇÕES
PROJECTO

Requisições Pendentes

1
Um item encontrado.

Requerente	Data de requisição	Nome Software	Versão Software	Tipo Software
Diana Gomes	24/06/2014	Windows 7	Enterprise	

1
Um item encontrado.

Opções de exportação: [CSV](#) | [Excel](#) | [XML](#) | [PDF](#)

Figura 49 – Página inicial do módulo de Recursos

Através da figura verificamos que a página inicial é constituída por três partes, *Hardware*, *Software* e *Requisições pendentes*. Em cada uma, o utilizador pode consultar informação e realizar um conjunto de funcionalidades previamente definidas.

Na secção *Hardware*, o utilizador tem acesso à informação dos seus equipamentos. O objectivo inicial seria apenas apresentar os computadores atribuídos, mas quando se analisou melhor o modelo de informação do GLPI, verificou-se que existem outros tipos de hardware a serem geridos. Após confirmar com os responsáveis do módulo, foi decidido adicionar a possibilidade de consultar a informação de telefones, monitores, computadores e periféricos. Para seleccionar o tipo pretendido foi adicionada uma *combobox* que condiciona a informação da tabela. Se o tipo de hardware seleccionado for *Computadores*, a tabela contém uma coluna extra que permite seleccionar um computador em particular e visualizar o software que lhe está atribuído.

A secção *Software* é sem dúvida a mais relevante do módulo. Utilizando um mecanismo de separadores o utilizador consegue navegar facilmente entre as diferentes funcionalidades implementadas.

5.1.1 O Meu Software

Quando o utilizador entra no módulo, o separador ativo é, por omissão *O meu Software*. Nesta secção é apresentado o software atribuído à máquina seleccionada na tabela de hardware. No diagrama da Figura 50 encontram-se os passos realizados pelo sistema para disponibilizar esta informação.

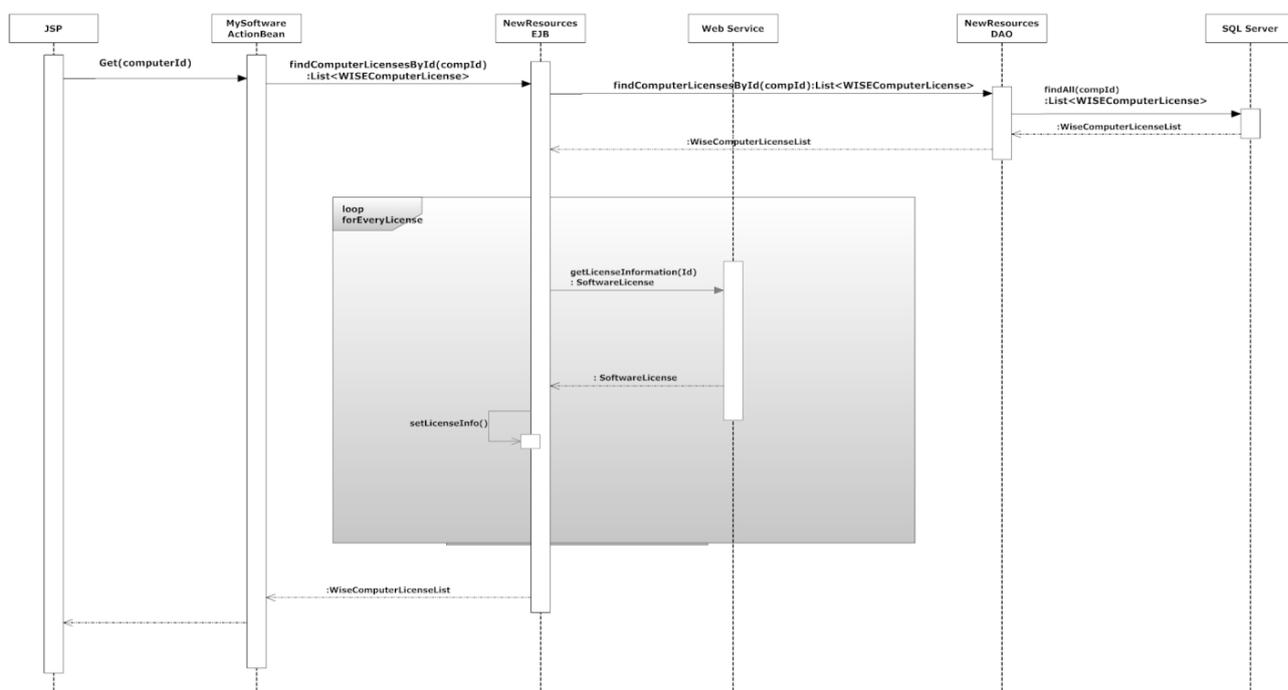


Figura 50 – Diagrama de sequência para o Meu Software

Quando o utilizador escolhe um computador é submetido um pedido para o ActionBean que, recorrendo ao serviço do módulo, busca a informação relevante. Dado que no WISE são unicamente armazenados os identificadores numéricos do software, versão, computador e licença, é necessário utilizar o Web Service para buscar a

informação necessária. As licenças da tabela podem ser libertadas a qualquer momento pelo utilizador. Esta ação provoca um *soft delete* na licença pretendida, isto é, o campo *deleted* é ativado e para efeitos aplicativos a licença fica apagada mas, na realidade, nunca é removida da base de dados.

Qualquer comunicação com o Web Service é feita através de mensagens SOAP. Na Figura 51 encontra-se um exemplo das mensagens trocadas entre o serviço e o Web Service.

<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:com="com.criticalsoftware.wiccore.wise.business.ws.newresources"> <soapenv:Header/> <soapenv:Body> <com:getLicenseInformation> <arg0>46</arg0> </com:getLicenseInformation> </soapenv:Body> </soapenv:Envelope></pre>	<pre><env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"> <env:Header/> <env:Body> <ns2:getLicenseInformationResponse xmlns:ns2="com.criticalsoftware.wiccore.wise.business.ws.newresources"> <return> <computer> <id>0</id> <manufacturer> <id>0</id> <name>Lenovo</name> </manufacturer> <model> <id>0</id> <name>ThinkPad T530</name> </model> <serial>PK1DNNC</serial> <userId>0</userId> </computer> <computerLicenseId>0</computerLicenseId> <id>0</id> <serial>BPQCR-24Q48-3QKTT-MF64K-YY3DV</serial> <software> <availableLicenses>0</availableLicenses> <id>0</id> <manufacturer> <id>0</id> <name>Microsoft</name> </manufacturer> <name>Visual Studio</name> <totalLicenses>0</totalLicenses> <version> <id>0</id> <name>Professional with MSDN 2012</name> </version> </software> <userLogin>dsgomes</userLogin> </return> </ns2:getLicenseInformationResponse> </env:Body> </env:Envelope></pre>
Pedido	Resposta

Figura 51 – Exemplo de mensagens SOAP – *getLicenseInformation(id)*

A mensagem do lado esquerdo é utilizada para invocar o método *getLicenseInformation(id)*. Neste caso em particular, pretende-se buscar a informação associada à licença com *id* 46. A resposta a este pedido, mensagem do lado direito, contém a informação necessária para preencher os campos da lista.

5.1.2 Lista de Software

O separador *Lista de Software* permite visualizar todo o software gerido pelo GLPI. Na Figura 52 é ilustrado o conteúdo desta secção.

Software

O MEU SOFTWARE LISTA DE SOFTWARE RELATÓRIOS REQUISIÇÃO DE SOFTWARE

Lista de Software

Nome:

Tipo de Software:

Tipo de Licença:

1 19 items encontrados.

Nome	Versão	Tipo de Software	Tipo de Licença	Estado	Nº Licenças	Licenças Disponíveis	
<input type="checkbox"/> AXURE RP PRO	Professional	Productivity		Active	1	0	
<input type="checkbox"/> Adobe Acrobat	Pro 11			Active	2	2	
<input type="checkbox"/> Adobe Acrobat	Std 11	Productivity		Active	1	0	
<input type="checkbox"/> Adobe Design Standard	CS6 MAC				1	0	
<input type="checkbox"/> Adobe Design Standard	CS6 Windows				1	1	
<input type="checkbox"/> Aptixia Ixload	Mail			Reserved	1	1	
<input type="checkbox"/> Aptixia Ixload	Plus			Reserved	1	1	
<input type="checkbox"/> Enterprise Architect	Corporate			Active	270	266	
<input type="checkbox"/> Mak-rti	Mak-rti Test Version			Reserved	5	4	
<input type="checkbox"/> Office	Professional Plus 2013		Microsoft Partner Network	Active	100	100	
<input type="checkbox"/> Office	Professional Plus 2010		Open License Agreement	Active	12	12	
<input type="checkbox"/> Project	Standard 2013	Productivity	Open License Agreement	Active	58	58	
<p>Nome:Project Versão:Standard 2013 Tipo de Software: Productivity Tipo de Licença: Open License Agreement Fabricante: Microsoft Descrição:OPEN 69500550ZZS1312 Estado:Active Nº Licenças:58 Licenças Disponíveis:58</p>							
<input type="checkbox"/> ReSharper	C# Edition	Productivity			8	8	

Requisitar software

Figura 52 – Lista de Software – Solução final

Como definido durante a fase de requisitos, a lista apresentada pode ser personalizada utilizando os filtros disponíveis. Por exemplo, se o utilizador desejar ver apenas softwares do tipo *Productivity*, deve seleccionar o tipo correto no filtro *Tipo de Software*. Os tipos disponíveis são obtidos diretamente do GLPI, dando liberdade aos gestores para controlar esta informação. O mesmo princípio é aplicado aos restantes filtros.

Caso o utilizador pretenda visualizar informação extra sobre o software, deve clicar no botão expandir, associado a cada elemento da lista. A disponibilidade de cada software é representada através de um campo colorido situado na última coluna. As cores utilizadas são intuitivas ao utilizador, a cor verde indica que o software tem licenças disponíveis, enquanto, que a cor vermelha indica que não existem licenças para atribuir. Independentemente da sua disponibilidade, o utilizador pode requisitar o software clicando no mesmo campo colorido, sendo reencaminhado para o separador *Requisição de Software*.

5.1.3 Relatórios

No separador *Relatórios* é possível visualizar todas as licenças de software atribuídas no GLPI. Na Figura 53 é ilustrado o resultado obtido após finalizar desenvolvimento.

Software

O MEU SOFTWARE LISTA DE SOFTWARE **RELATÓRIOS** REQUISICÃO DE SOFTWARE

Relatórios

Empresa : Critical Software
 Área : Delivery - Engineering
 Colaborador :
 Software :

Procurar

1 9 itens encontrados.

Colaborador	Empresa	Área	Software	Versão	Equipamento
Bruno Filipe Matos Rito	Critical Software	Delivery - Engineering	Versions	1.2.3	Apple MacBookPro9,2 (C1MKM0FSDTY4)
Diana Sofia Pereira Gomes	Critical Software	Delivery - Engineering	Enterprise Architect	Corporate	Lenovo ThinkPad T530 (PK1DNNC)
Diana Sofia Pereira Gomes	Critical Software	Delivery - Engineering	Windows Server	2012 Datacenter	Lenovo ThinkPad T530 (PK1DNNC)
Diana Sofia Pereira Gomes	Critical Software	Delivery - Engineering	Visual Studio	Professional with MSDN 2012	Lenovo ThinkPad T530 (PK1DNNC)
Padro Nuno Simplicio Messias	Critical Software	Delivery - Engineering	Enterprise Architect	Corporate	Apple MacBook Pro11,2 (C02LJ7VFFD56)
Rute Esteves de Frias	Critical Software	Delivery - Engineering	Enterprise Architect	Corporate	Apple MacBookPro9,2 (C1MKP2JQDTY4)
Rute Esteves de Frias	Critical Software	Delivery - Engineering	Versions	1.2.3	Apple MacBookPro9,2 (C1MKP2JQDTY4)
Rute Esteves de Frias	Critical Software	Delivery - Engineering	Adobe Design Standard	CS6 MAC	Apple MacBookPro9,2 (C1MKP2JQDTY4)
Rute Esteves de Frias	Critical Software	Delivery - Engineering	AXURE RP PRO	Professional	Apple MacBookPro9,2 (C1MKP2JQDTY4)

1 9 itens encontrados.

Opções de exportação: CSV | Excel | XML | PDF

Figura 53 – Exemplo de relatório

Analisando a figura verificamos que a secção de relatórios é relativamente simples. O utilizador além de conseguir visualizar todas as licenças atribuídas, pode exportar os relatórios para formatos mais convenientes como *PDF*, e personalizar a informação que pretende aplicando os filtros disponíveis. No exemplo, a informação é filtrada por empresa *Critical Software* e área *Delivery – Engineering*.

A informação utilizada nesta secção resulta de ambos os sistemas, o Web Service dá acesso aos dados sobre software, hardware e licenças, e os serviços são responsáveis pela informação de colaboradores, áreas e empresas. De facto, este princípio é aplicado em todo o módulo. Sempre que é necessário qualquer informação sobre recursos é usado o Web Service.

Uma das limitações encontradas durante o seu desenvolvimento, foi o facto de o GLPI não retratar os conceitos de Área e Empresa. Para ultrapassar esta limitação, os filtros relacionados com esta informação são aplicados no serviço quando é recebida a informação do Web Service.

5.1.4 Requisições de Software

Uma das grandes vantagens derivada da implementação do novo módulo de recursos, foi a automatização do processo de requisição de software conjugando a informação de ambos os sistemas. Nesta secção, ver Figura 54, o utilizador pode requisitar software gerido pelo GLPI.

Figura 54 – Secção de Requisição – Solução final

Selecionando uma opção da lista de software, a sua informação é apresentada automaticamente na secção de detalhes. Dependendo do tipo de requisição seleccionada, os campos *Área* ou *Projeto* são desbloqueados e preenchidos automaticamente, disponibilizando todas as áreas ou projetos da empresa seleccionada. O aprovador inicial está diretamente relacionado com a área ou projeto escolhido. Por exemplo, seleccionando um projeto da lista, o campo *Aprovador Inicial* é atualizado com os respetivos *PM* e *PMO*. O diagrama de sequência apresentado na Figura 55 retrata a submissão de uma requisição.

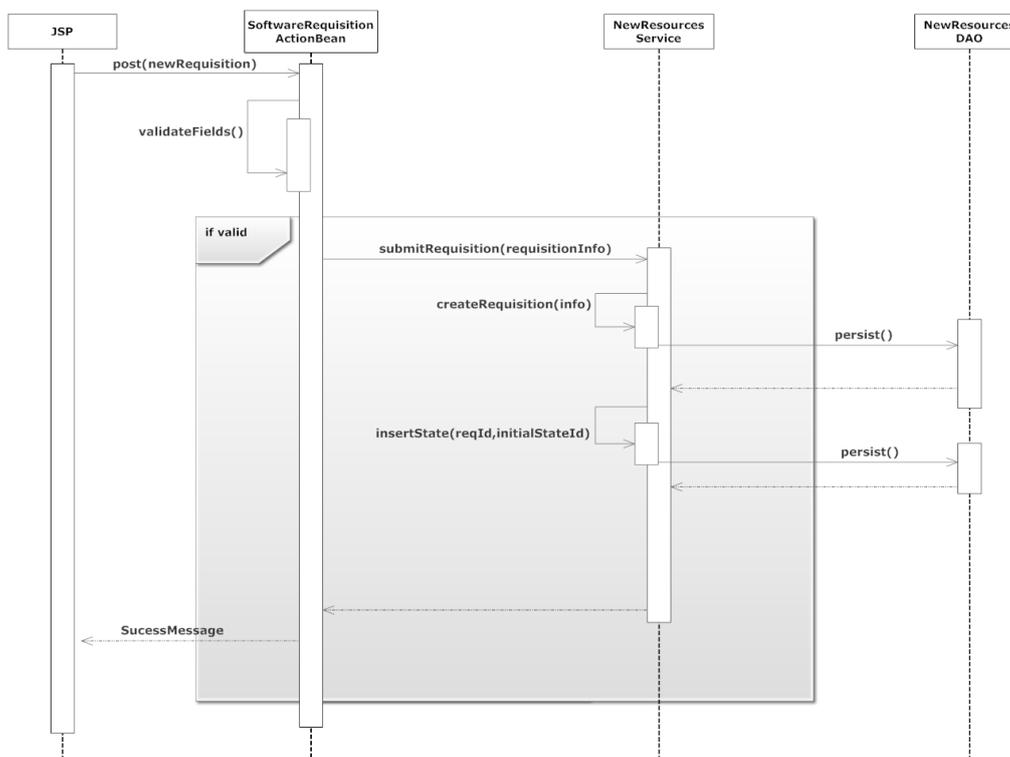


Figura 55 – Diagrama de sequência para requisição de Software

Uma requisição submetida é inicialmente validada no Action Bean de forma a verificar se todos os campos necessários estão preenchidos. No caso de a requisição ser válida, o pedido prossegue para o serviço do módulo onde é realizado o processamento necessário. O processo é concluído com a apresentação de uma mensagem de sucesso ao utilizador.

Ainda nesta página, o utilizador pode ter acesso ao seu histórico de requisições clicando em *O meu histórico*. A Figura 56 apresenta o histórico de requisições.

Software

O MEU SOFTWARE LISTA DE SOFTWARE RELATÓRIOS REQUISIÇÃO DE SOFTWARE

O meu Histórico

1 4 items encontrados.

Colaborador	Nome Software	Nome Versão	Tipo de Requisição	Área	Projecto	Aprovador
Diana Gomes	Enterprise Architect	Corporate	Geral			Pedro Murtinho
Diana Gomes	Windows Server	2012 Datacenter	Geral			Pedro Murtinho
Diana Gomes	Visual Studio	Professional with MSDN 2012	Geral			Pedro Murtinho
Diana Gomes	Windows 7	Enterprise	Projecto		Betfair Global Delivery Services (SoW6)	Silvia Rocha

1 4 items encontrados.

Opções de exportação: CSV | Excel | XML | PDF

Figura 56 – Histórico de requisições – Solução final

Todas as requisições realizadas pelo utilizador, canceladas, rejeitadas ou pendentes, são apresentadas através de uma lista na secção de requisições. A abordagem de *soft delete* utilizada nas licenças é também aplicada na informação de requisições, conseguindo desta forma gerar um histórico completo que inclua também as requisições canceladas. Clicando numa requisição da lista é carregada a página que permite visualizar os seus detalhes. A página de detalhes é apresentada na Figura 57.

Software

O MEU SOFTWARE LISTA DE SOFTWARE RELATÓRIOS REQUISIÇÃO DE SOFTWARE

Detalhes Requisição de Software

Detalhes de Software

Nome : Windows 7

Versão : Enterprise

Descrição : Licence ID: 66116011

Tipo :

Fabricante : Microsoft

Detalhes de Aprovação

Submetido
Diana Gomes
24/06/2014

Aprovador Inicial

Enviado a DSI

Atribuído

Utilização

Data do pedido : 24/06/2014

Colaborador : Diana Sofia Pereira Gomes

Empresa : Critical Software

Área : Delivery - Engineering

Utilização : Projecto : Betfair Global Delivery Services (SoW6)

Computador : Lenovo ThinkPad T530 (PK1DNNC)

[O meu Histórico](#)

Figura 57 – Detalhes de requisição – Solução final

A página de detalhes está dividida em três secções distintas com informação específica. O mapa de estados apresentado na secção de aprovação, permite monitorizar a requisição durante o seu ciclo de vida. No exemplo ilustrado, a requisição encontra-se atualmente no estado *Aprovador Inicial* à espera de uma decisão.

Dependendo dos privilégios associados ao utilizador no WISE, as listas de requisições pendentes para cada tipo de requisição poderão estar ou não visíveis.

Pending Requisitions

MY REQUISITIONS GENERAL AREA PROJECT

Requester	Requisition Date	Software Name	Software Version	Software Type
Diana Gomes	25/06/2014	Windows 7	Enterprise	

Export options: CSV | Excel | XML | PDF

Figura 58 – Lista de requisições pendentes para Projeto

Na Figura 58, o utilizador tem acesso a todas as listas de requisições pendentes, mostrando neste caso as requisições associadas ao(s) seu(s) projeto(s). A cada linha da tabela estão associadas duas funcionalidades: rejeitar a requisição através do botão colocado na última coluna e, visualizar os seus detalhes clicando em qualquer um dos seus campos. Caso o utilizador seja o responsável pela decisão da requisição, e apresente as permissões necessárias, terá acesso ao formulário que permite aceitar ou rejeitar a requisição na página de detalhes.

Software

MY SOFTWARE SOFTWARE LIST REPORTS SOFTWARE REQUISITION

Software Requisition Details

Software Details

Name : Windows 7
Version : Enterprise
Description : Licence ID: 66116011
Type :
Manufacturer : Microsoft

Approval Details

Submitted Diana Gomes 25/06/2014 → Initial Approver → Sent to DSI → Attributed

Initial Notes :

Utilization

Request Date : 25/06/2014
Worker : Diana Sofia Pereira Gomes
Company : Critical Software
Area : Delivery - Engineering
Utilization : Project : Betfair Global Delivery Services (SoW6)
Computer : Lenovo ThinkPad T530 (PK1DNNC)

My History

Confirm Reject

Figura 59 – Detalhes de uma requisição pendente - Aprovadores

O utilizador pode também introduzir notas que ache relevantes para a requisição. Aprovando ou rejeitando uma requisição, o seu processo transita automaticamente para o estado seguinte. O diagrama de sequência que retrata a aprovação de uma requisição está ilustrado na Figura 60.

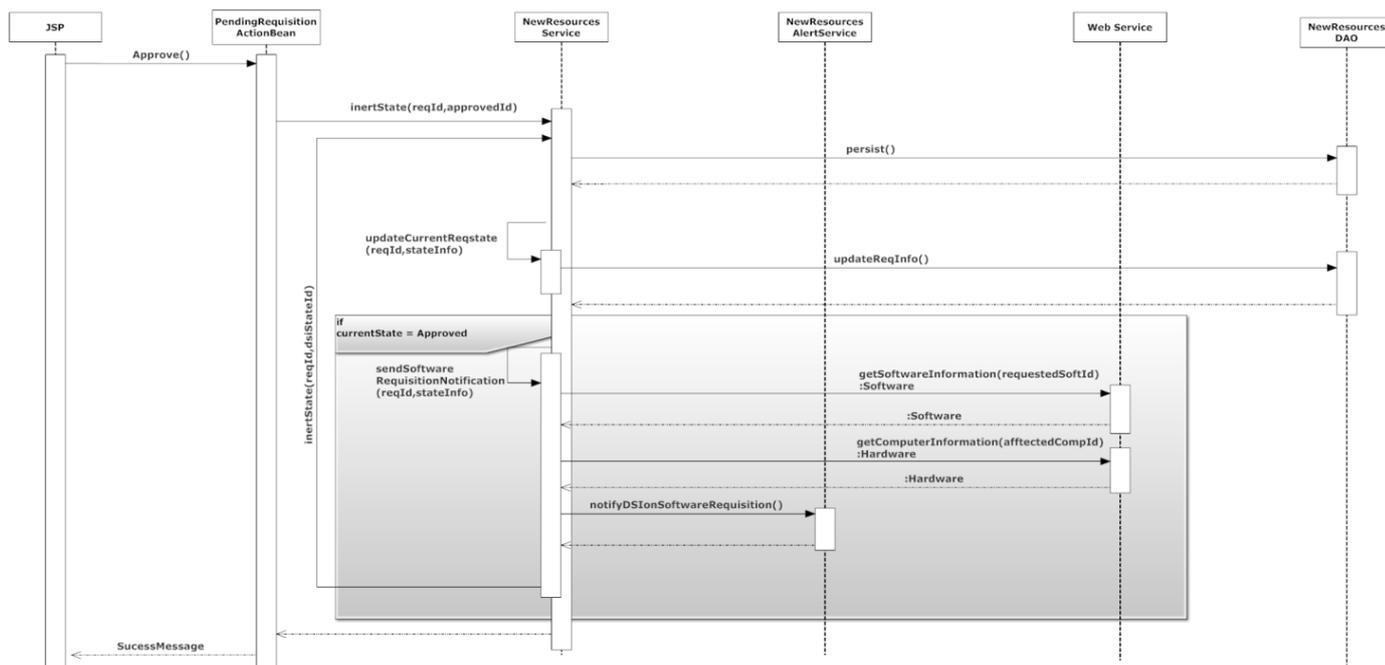


Figura 60 – Diagrama de sequência para aprovação de requisição

Quando uma requisição é aprovada ou rejeitada pelo responsável, introduz-se na base de dados o estado associado de forma a comprovar que a requisição já foi analisada. No caso de ser aprovada, é enviada por e-mail uma notificação a DSI informando que existe uma nova requisição. O conteúdo desta notificação segue as normas definidas durante a fase de requisitos. Se o e-mail for enviado com sucesso, é inserido na base de dados o respetivo estado comprovando que o DSI já foi notificado. É com base nestes estados armazenados que o mapa é construído, permitindo uma monitorização simples e eficaz para os intervenientes do processo.

No capítulo anterior referiu-se que a sincronização entre os dois sistemas é realizada através de duas operações executadas diariamente. A primeira, analisa as licenças atribuídas no GLPI e verifica se existe alguma combinação possível com as requisições pendentes no estado *Enviada a DSI*. Existem dois aspetos a considerar durante a execução desta tarefa: só são analisadas licenças atribuídas que tenham as mesmas propriedades da requisição (mesmo computador, software, versão e colaborador) e, o identificador numérico da atribuição não pode existir no WISE. Este último, serve para salvaguardar a integridade da informação em cenários que existam múltiplas requisições com as mesmas propriedades. Por exemplo, na presença de requisições com as mesmas propriedades (mesmo software, versão, computador e utilizador) e só uma licença atribuída para uma dessas requisições, é necessário garantir que não é atribuída a mesma licença nas restantes requisições. A segunda operação, tem como objetivo verificar se existem inconsistências entre os dois sistemas, isto é, se não existem licenças válidas num sistema e inválidas no outro e, se existem atribuições válidas no GLPI que não correspondam a nenhuma requisição pendente. Na Figura 61 encontra-se o diagrama de sequência para as tarefas descritas.

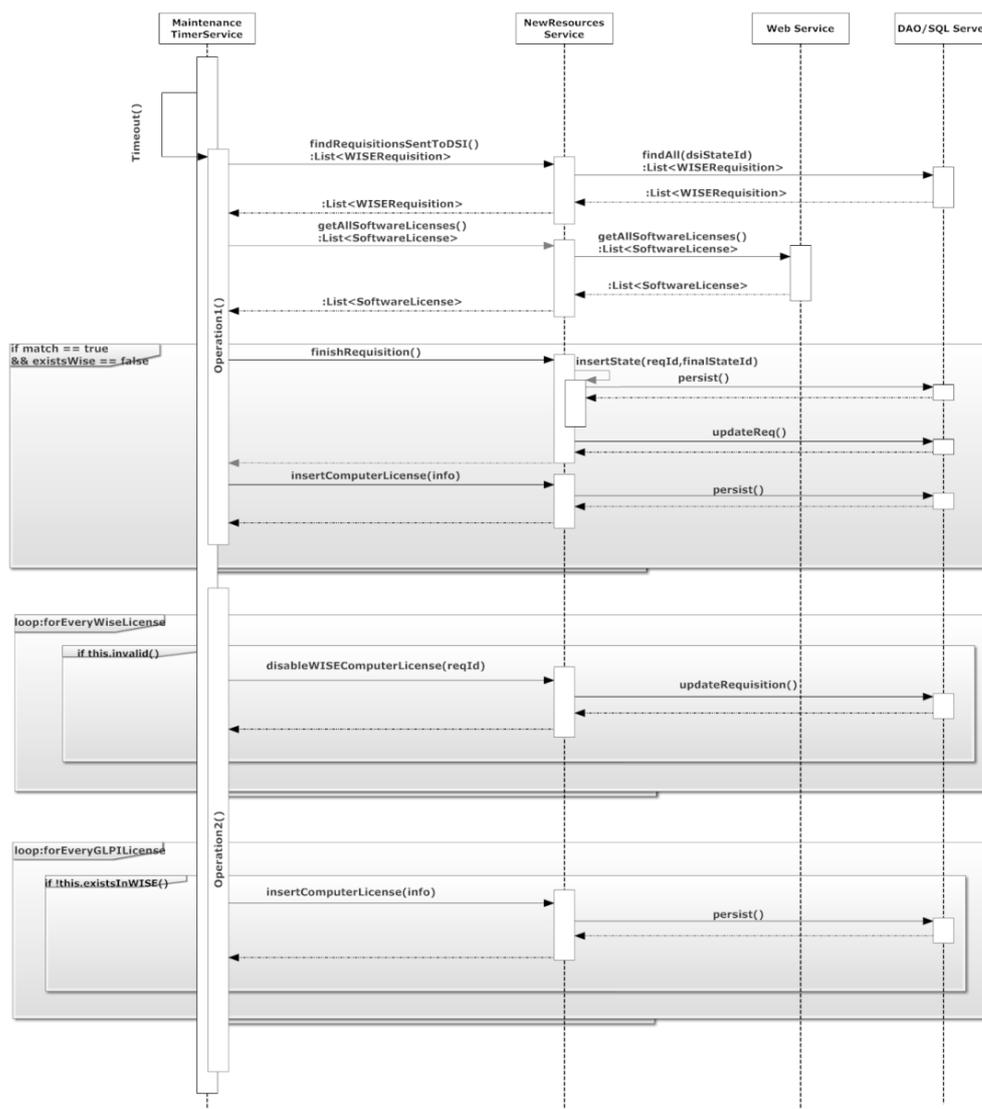


Figura 61 – Diagrama de sequência para processo de sincronização

Assim que o processo de sincronização finaliza, o ciclo de vida de uma requisição termina e o seu mapa de estados fica completo. A figura seguinte ilustra o mapa de estados completo para a requisição ilustrada na Figura 62.

Software

O MEU SOFTWARE LISTA DE SOFTWARE RELATÓRIOS **REQUISIÇÃO DE SOFTWARE**

Detalhes Requisição de Software

Detalhes de Software

Nome : Windows 7
 Versão : Enterprise
 Descrição : Licence ID: 66116011
 Tipo :
 Fabricante : Microsoft

Detalhes de Aprovação

Submetido Diana Gomes 26/06/2014 → Aprovador Inicial Silvia Rocha 26/06/2014 → Enviado a DSI → Atribuido

Utilização

Data do pedido : 26/06/2014
 Colaborador : Diana Sofia Pereira Gomes
 Empresa : Critical Software
 Área : Delivery - Engineering
 Utilização : Projecto : Betfair Global Delivery Services (SOW6)
 Computador : Lenovo ThinkPad T530 (PK1DNNC)

[O meu Histórico](#)

Figura 62 – Processo de Requisição completo

5.2. Módulo de Credenciações

A reestruturação do módulo permitiu superar as limitações impostas na versão anterior, resolvendo as lacunas identificadas que impossibilitavam a sua utilização.

Ao longo desta secção são apresentados os resultados obtidos na solução final destacando as alterações realizadas. A página inicial do módulo é ilustrada na Figura 63.

Clearances
Clearances List

Name / Login :
 Company :
 Area :
 Brand :
 Level :
 Start Date:
 End Date:
 Corporate :

Search

Worker	Company	Area	Brand	Level	Type	Request Date	Expiration Date
Luis Figueiredo	Critical Software	Subsidiaries	NATO	Confidencial	Individual	27/05/2010	-
Ana Cristina Guimarães	Critical Software	Biz Dev - ECS	NATO	Confidencial	Individual	27/05/2010	-
André Carvalho	Critical Software	Delivery - Engineering	NATO	Confidencial	Individual	27/05/2010	-
Pedro Miquel Cardoso	Critical Software	Delivery - Engineering	NATO	Cosmic Top Secret Atomal	Individual	27/05/2010	-
Tiago Gomes	Critical Software	Delivery - Engineering	NACIONAL	Secreto	Individual	27/05/2010	-
Rui Pereira Melo	Critical Software	Subsidiaries	NATO	Confidencial	Individual	27/05/2010	-
Mauro Gameiro	Critical Software	Delivery - Engineering	NATO	Confidencial	Individual	03/10/2012	-
Paulo Lourenço	Critical Software	Shared Services - DSI	NATO	Confidencial	Individual	03/10/2012	-
Luis Gargaté	Critical Software	Biz Dev - Safety Critical	NATO	Confidencial	Individual	11/10/2011	05/01/2017

Export options: CSV | Excel | XML | PDF

Figura 63 – Página inicial do módulo de Credenciações

Nas secções de credenciações ativas e pendentes, as alterações resumem-se na adição de novos campos informativos nas listas tornando-as mais completas e úteis ao utilizador e, a criação de novos filtros que permitem personalizar a informação pretendida. Na Figura 63 e 64, encontram-se realçadas as alterações efetuadas nas duas secções.

Clearances
Pending Clearances

Name / Login :
 Company :
 Area :
 Brand :
 Level :
 Current State :
 Request Date :
 Corporate :

Search

Worker	Type	Company	Area	Brand	Level	Current State	Request Date
Joaquim Tojal	Individual	EDUCED	Management	NATO	Confidencial	Request submitted	11/10/2011
Luis Afonso Ribeiro	Individual	Critical Software	Delivery - Engineering	NATO	Confidencial	Request submitted	11/10/2011
José Carqueira	Individual	Critical Software	Delivery - Engineering	NATO	Confidencial	Request submitted	11/10/2011
Tiago Neto Rodrigues	Individual	Critical Software	Delivery - Engineering	NATO	Confidencial	Request submitted	11/10/2011

Figura 64 – Página de credenciações pendentes alterada

Na lista de credenciações pendentes, clicando num item da lista o utilizador tem acesso aos detalhes do seu processo. A página que permite visualizar os detalhes pode ser encontrada na Figura 65.

The screenshot displays two main sections of the WISE interface:

Clearance Details: A table listing information for a worker named Luis Afonso Ribeiro. The 'Company' (Critical Software), 'Area' (Delivery - Engineering), and 'Type' (Individual) fields are highlighted with red boxes. Other details include Brand (NATO), Level (Confidential), Cost Centers (17598 - Credenciação Nato), and Submitted By (Paulo Lourenço).

State Map: A horizontal flow diagram showing the process stages: Request submitted (Paulo Lourenço, 11/10/2011), Approved by OS (highlighted with a blue circle), Forms delivered to the collaborator, Forms received from the collaborator, Forms validated, Request sent to GNS, and Clearance approved by GNS.

Clearance Process: A form for editing the process. The 'Cost Center' dropdown is open, showing a list of options including 'wise', '45206-BCG - WISE', '46211-MANBCG2014 - Manutenção WOW', '45229-Manutenção WISE e WOW 2010', '12416-TaskTracker4WISEv2', '13212-WISE', '12101-WISE', '12401-Wise - Expenses', '12402-WISE - Projects Phase 4', '12405-Wise Costcentre Weekly Validati', '12424-WISE Installation for CSWBR', and '12188-Wise Maintenance 1st level'. The 'Notes' field is empty. At the bottom right, there are 'Confirm', 'Reject', and 'Return' buttons.

Figura 65 – Página detalhes de credenciação pendente

Caso o utilizador apresente as permissões necessárias, terá acesso ao formulário para gerir o processo. Este formulário permite confirmar o estado atual e avançar para o estado seguinte ou, rejeitar o pedido. Realizaram-se ajustes na página para tornar a monitorização e controlo do processo mais eficaz. Foi adicionada informação na secção de detalhes, alterado o aspeto gráfico do mapa de estados conforme as necessidades do oficial de segurança, limitada a rejeição do processo até ao estado *Request Sent to GNS* e, por último, foi também adicionada uma caixa *autocomplete*, visível apenas ao oficial de segurança no estado *Approved by OS*, que permite alterar o centro de custos associado. Nesta caixa o utilizador pode pesquisar pelos centros de custo existentes introduzindo o respetivo nome ou, o identificador numérico associado.

A secção de relatórios do módulo era claramente o ponto central de falha, sendo necessária uma reestruturação total. Na Figura 66 é apresentado o resultado final.

Name / Login :

Company :

Area :

Brand :

Level :

Cost Center :

Start Date :

End Date :

State :

Corporate :

CLEARANCES REPORTS CONFIGURATION

25 items found.

Workers	Company	Area	Brand	Level	Type	Cost Center	Request Date	Expiration Date	State		
Mickael Marques	Critical Software	ASD - Engineering	NATO	Confidencial	Individual	-	27/05/2010	-	Valid		
Zeferino Raposo	Critical Software	ASD - Engineering	NATO	Confidencial	Individual	-	27/05/2010	-	Valid		
Wanderley Júnior	Critical Software	Security & Infrastructure	NATO	Confidencial	Individual	-	27/05/2010	-	Valid		
Jorge Alçada	Critical Software	Dependability & Embedded	NATO	Confidencial	Individual	-	27/05/2010	-	Valid		
David Alves	Critical Software	ECS - Engineering	NATO	Confidencial	Individual	-	27/05/2010	-	Valid		
António Abreu	Critical Software	-	NATO	Confidencial	Individual	-	27/05/2010	-	Valid		
Mário Alves	Critical Software	ASD - Engineering	NATO	Confidencial	Individual	-	27/05/2010	-	Valid		
David Gil	Critical Software	ASD - Engineering	NATO	Confidencial	Individual	-	27/05/2010	-	Valid		
Micael Pereira	Critical Software	Command & Control	NATO	Confidencial	Individual	-	27/05/2010	-	Valid		
Henrique Oliveira	Critical Software	Command & Control	NATO	Confidencial	Individual	-	27/05/2010	-	Valid		
Luis Figueiredo	Critical Software	Subsidiaries	NATO	Confidencial	Individual	-	27/05/2010	-	Valid		
António Abreu	Critical Software	-	NATO	Confidencial	Individual	-	27/05/2010	-	Valid		
Ana Cristina Guimarães	Critical Software	Biz Dev - ECS	NATO	Confidencial	Individual	-	27/05/2010	-	Valid		
André Carvalho	Critical Software	Delivery - Engineering	NATO	Confidencial	Individual	-	27/05/2010	-	Valid		
Luis Quedas	Critical Software	ECS - Engineering	NATO	Confidencial	Individual	-	27/05/2010	-	Valid		
Nuno Luis Almeida	Critical Software	BizDev	NATO	Confidencial	Individual	-	27/05/2010	-	Valid		
Pedro Miguel Cardoso	Critical Software	Delivery - Engineering	NATO	Cosmic Top Secret Atomal	Individual	-	27/05/2010	-	Valid		
Emanuel Teixeira	Critical Software	ASD - Engineering	NATO	Confidencial	Individual	-	27/05/2010	-	Valid		
Tiago Gomes	Critical Software	Delivery - Engineering	NACIONAL	Secreto	Individual	-	27/05/2010	-	Valid		
David Gil	Critical Software	ASD - Engineering	NATO	Confidencial	Individual	-	27/05/2010	-	Valid		
Rui Pereira Melo	Critical Software	Subsidiaries	NATO	Confidencial	Individual	-	27/05/2010	-	Valid		
Miguel David	Critical Software	ASD - Engineering	NATO	Cosmic Top Secret	Individual	-	27/05/2010	-	Valid		
Miguel David	Critical Software	ASD - Engineering	NATO	Confidencial	Individual	-	11/08/2010	-	Valid		
Mauro Gameiro	Critical Software	Delivery - Engineering	NATO	Confidencial	Individual	10008 - Security Office	03/10/2012	-	Valid		
Paulo Lourenco	Critical Software	Shared Services - DSI	NATO	Confidencial	Individual	10008 - Security Office	03/10/2012	-	Valid		

25 items found.

Export options: [CSV](#) | [Excel](#) | [XML](#) | [PDF](#)

Workers that have left the organization

Figura 66 – Secção de relatórios

Esta secção, limitada aos utilizadores responsáveis pela gestão, permite não só visualizar informação sobre credenciações válidas, expiradas, canceladas e rejeitadas, bem como, cancelar ou renovar credenciações. A operação de renovar é limitada ao oficial de segurança e aplicável exclusivamente a credenciações válidas, ou seja, que pertençam a colaboradores que ainda não saíram da empresa. A operação de cancelar só é aplicável a credenciações no estado *Válida*, ficando desativada nos restantes casos.

Como se pode verificar no relatório apresentado na Figura 66, existem credenciações válidas para colaboradores que já saíram da organização, pois quando as credenciações são atribuídas, têm duração mínima de um ano. Durante este tempo os colaboradores podem abandonar a organização mas a sua credenciação continua a ser válida. De modo a destacar estes casos e captar a atenção do utilizador, foi utilizada uma cor específica com a devida legenda no fundo da tabela. Relativamente aos filtros implementados, permitem condicionar a informação apresentada conforme os valores introduzidos ou selecionados. A sua utilização é intuitiva pois têm uma relação direta com os campos que constituem o relatório.

Clicando no campo *Workers*, o utilizador é reencaminhado para o histórico de credenciações do colaborador. Esta secção, ilustrada na Figura 67, foi reestruturada por completo de acordo com os requisitos definidos.

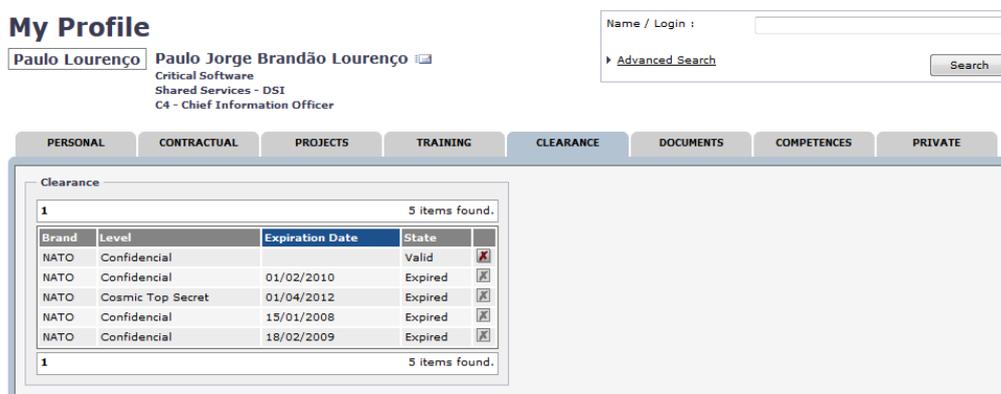


Figura 67 – Histórico de Credenciações

Se o utilizador apresentar as permissões necessárias, terá acesso à funcionalidade de cancelar credenciações válidas.

Por último, foi adicionada uma secção de administração, Figura 68, acessível unicamente pelo oficial de segurança. Aqui é possível configurar os valores utilizados na marca e nível das credenciações. Tipicamente, uma marca tem um conjunto de níveis associados que permitem identificar os diferentes graus de segurança.

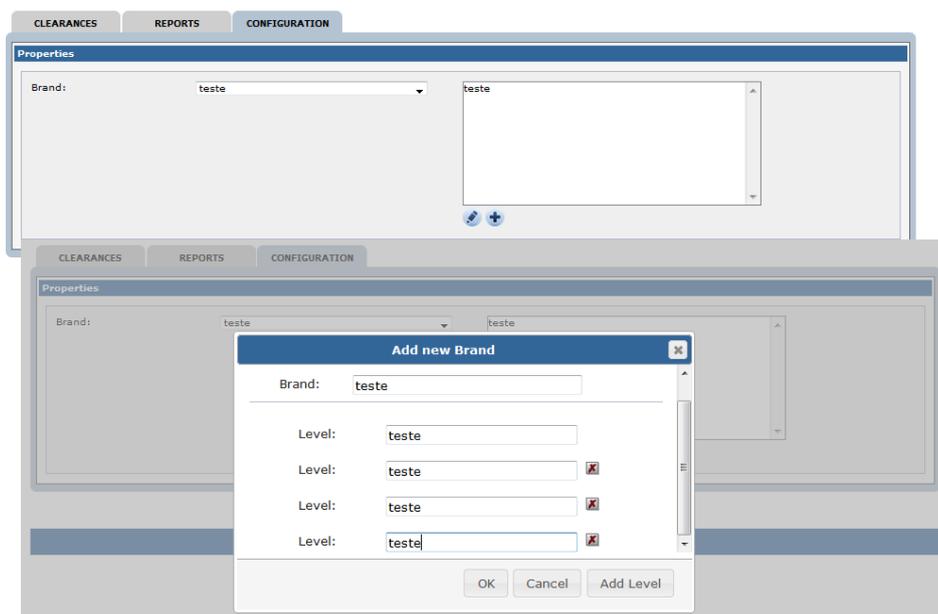


Figura 68 – Secção de administração – Solução final

Através do exemplo verificamos que a secção é relativamente simples. O utilizador tem a hipótese de seleccionar uma marca e editar a sua informação, ou adicionar uma marca com os níveis que desejar. Qualquer uma das funcionalidades é realizada através de um formulário, de preenchimento obrigatório, que aparece quando o utilizador clica na respetivo botão.

6 Conclusão

O estágio realizado na Critical Software pode ser dividido em duas partes distintas, relacionadas diretamente com cada semestre. A primeira pode-se resumir na aquisição de conhecimento essencial para realizar o projeto. Durante esta fase, interiorizou-se o contexto do projeto, estudados os processos de negócio de interesse e, analisada a documentação fornecida de forma a conhecer como se processa o desenvolvimento de software a nível profissional, recorrendo a um conjunto de tarefas, fases e normas estipuladas. Após analisar os processos de negócio e os problemas a resolver, foram especificados documentos técnicos que modelavam as soluções pretendidas através de requisitos, casos de uso e protótipos rápidos. Durante esta fase verificou-se a importância que um bom planeamento e, uma definição clara do problema com as respetivas soluções, têm no desenvolvimento do projeto.

O segundo semestre focou-se no desenvolvimento das soluções previamente definidas e respetiva validação. Verificou-se que a definição antecipada de documentos técnicos claros e detalhados, aceleram consideravelmente o processo de desenvolvimento. Durante esta fase foram ultrapassados vários desafios, no entanto, o mais valioso é sem dúvida a integração do GLPI com o sistema de informação. A oportunidade de estabelecer a comunicação entre dois sistemas distintos, cada um com objetivos específicos e independentes, aplicando uma camada de integração na arquitetura complexa do WISE, contribuiu para desenvolver e adquirir novas competências. À medida que o desenvolvimento se aproximava do fim, foram definidos os documentos de especificação de testes que validam as soluções implementadas.

Por fim, o processo de desenvolvimento foi sem dúvida o aspeto que mais contribuiu na aquisição de novos conhecimentos. O contacto com novas tecnologias, a oportunidade de trabalhar num sistema de informação complexo e de dimensão considerável e, seguir padrões e normas de qualidade estabelecidas numa empresa CMMI nível 5, são sem dúvida fatores positivos que contribuíram para manter a motivação ao longo do estágio, mas também, que marcam o início da carreira profissional.

6.1. Trabalho Futuro

O WISE é um projeto em constante evolução e manutenção, estando constantemente sujeito a alterações e atualizações. Durante a validação das soluções finais surgiram alguns aspetos que seriam úteis para melhorar a sua usabilidade e processos retratados.

Relativamente ao módulo de recursos, foram considerados alguns pontos que seriam sem dúvida melhorias interessantes. Um desses pontos contempla automatizar ainda mais o processo de requisição. Atualmente a comunicação com o GLPI é unilateral e só são realizadas consultas de informação. O objetivo seria disponibilizar métodos no Web Service que permitam injetar diretamente as requisições, e licenças libertadas, na base de dados do GLPI, evitando assim o uso de e-mails. Outro aspeto interessante seria criar novos eventos alertáveis que informem os intervenientes do processo, nomeadamente, quando uma licença é atribuída ou rejeitada. O último ponto é direcionado às configurações do Web Service e base de dados do GLPI. Na página de configuração do WISE encontram-se todos os parâmetros que configuram o seu ambiente. Seria útil

adicionar nesta página, campos configuráveis para o endereço do Web Service e dados de acesso à base de dados utilizada.

Por fim, existem dois aspetos finais a considerar relativamente à validação das soluções finais. No final da fase de validação, ficou estabelecido que seria necessário dedicar uma equipa ou entidade, responsável para realizar novamente os testes definidos e expandir ainda mais os cenários testados. Após esta equipa terminar e, obtida a aprovação final dos clientes, os módulos serão colocados no ambiente de produção.

7 Referências

- [1] Critical Software S.A. CSW-WISE4ALL-2011-MMO-04672-wise-overview.
- [2] Critical Software S.A, “CSW-QMS-2009-GBK-04120-wise-project-module-guidebook”.
- [3] Basics of Company Security Clearance - Defense Industry [Online]. Visto a Junho 02, 2014, disponível em: <http://defense.about.com/od/BusinessOps/a/FCL.html>
- [4] Obtaining a Security Clearance for Your Company and/or Employees | SBA.gov. [Online]. Visto a Junho 02, 2014, disponível em: <http://www.sba.gov/content/obtaining-security-clearance-your-8a-company-andor-employees>
- [5] Critical Software S.A, “CSW-WISE-2006-DOC-3607-vision-scope-credenciacoes”.
- [6] Chapter 1: What is Software Architecture? [Online]. Visto a Junho 02, 2014, disponível em: <http://msdn.microsoft.com/en-us/library/ee658098.aspx>
- [7] Liu, X., Heo, J., & Sha, L. “Modeling 3-Tiered Web Services” publicado em Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2005. 13th IEEE International Symposium on
- [8] What is the 3-Tier Architecture? [Online]. Visto a Junho 02, 2014, disponível em: <http://www.tonymarston.net/php-mysql/3-tier-architecture.html>
- [9] Cramon, J., Architect, C., & Goodin, B. “What readers are saying about Stripes . . . and Java Web Development Is Fun Again”. In Jacquelyn Carter (Ed.), (2009th–4th–20th ed.).
- [10] ActionBean (Stripes 1.5.7 API Documentation) [Online]. Visto a Junho 10, 2014, disponível em: <http://stripes.sourceforge.net/docs/current/javadoc/net/sourceforge/stripes/action/ActionBean.html>
- [11] J2EE Platform Overview (Sun Java System Application Server 9.1 Deployment Planning Guide) [Online]. Visto a Junho 10, 2014, disponível em: <http://docs.oracle.com/cd/E19159-01/819-3680/abfar/index.html>
- [12] ComponentSource: Commercial Component Creation (J2EE) - Technical White Paper. [Online]. Visto a Junho 10, 2014. Disponível <http://www.componentsource.com/services/whitepapers/j2eewhitepaper.asp#J2EE>
- [13] The Java Persistence API - A Simpler Programming Model for Entity Persistence [Online]. Visto a Junho 10, 2014, disponível em: <http://www.oracle.com/technetwork/articles/javaee/jpa-137156.html>
- [14] Introduction to the Java Persistence API - The Java EE 6 Tutorial [Online]. Visto a Junho 10, 2014, disponível em: <http://docs.oracle.com/javaee/6/tutorial/doc/bnbpz.html>

- [15] Project Management [Online]. Visto a Junho 10, 2014, disponível em:
http://en.wikipedia.org/wiki/Project_management
- [16] Waterfall model [Online]. Visto a Junho 04,2014, disponível em:
http://en.wikipedia.org/wiki/Waterfall_model
- [17] – Scrum (software development) [Online]. Visto a Junho 04,2014, disponível em:
[http://en.wikipedia.org/wiki/Scrum_\(software_development\)](http://en.wikipedia.org/wiki/Scrum_(software_development))
- [18] Critical Software S.A, “CSW-QMS-2002-SDP-0909-software-development”.
- [19] Critical Software S.A, “CSW-QMS-2003-PCS-2296-project-life-cycles”.
- [20] Christopher J. Alberts, Audrey J. Dorofee, Risk Management Framework, Software Engineering Institute, August 2010.
- [21] Chapter 9. Section 9.4. Erl, T. *Service-Oriented Architecture: A Field Guide to Integrating Xml and Web Services* (2004). Prentice Hall.
- [22] Java SE Technologies – Database [Online]. Visto a Junho 20, 2014, disponível em:
<http://www.oracle.com/technetwork/java/javase/jdbc/index.html>
- [23] Java Architecture for XML Binding (JAXB) [Online]. Visto a Junho 20, 2014, disponível em:
<http://www.oracle.com/technetwork/articles/javase/index-140168.html>
- [24] Building Web Services with JAX-WS [Online]. Visto a Junho 20, 2014, disponível em:
<http://docs.oracle.com/javasee/5/tutorial/doc/bnayl.html>
- [25] XML Schema (W3C) [Online]. Visto a Junho 20, 2014, disponível em :
http://en.wikipedia.org/wiki/XML_Schema_%28W3C%29

Anexos