

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática



**DESENVOLVIMENTO EM TECNOLOGIA
OUTSYSTEMS DE APLICAÇÃO PARA GESTÃO
DE PATRIMÓNIO**

Dinis Premji

PROJECTO

Mestrado em Engenharia Informática
Arquitectura, Sistemas e Redes de Computadores

2010

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática



**DESENVOLVIMENTO EM TECNOLOGIA
OUTSYSTEMS DE APLICAÇÃO PARA GESTÃO
DE PATRIMÓNIO**

Dinis Premji

PROJECTO

Trabalho orientado pela Prof. Doutora Maria Isabel Batalha Reis Gama Nunes
E co-orientado por Eng. Gonçalo Maria Granja Cardoso Menezes de Matos

Mestrado em Engenharia Informática
Arquitectura, Sistemas e Redes de Computadores

2010

Agradecimentos

Dedico este trabalho primeiramente a Deus e à memória do meu Pai.

Este trabalho é o fruto de um longo percurso de aprendizagem que começou na escola primária até à faculdade. Este percurso foi marcado pela intervenção de diversas pessoas que contribuíram de alguma forma por este projecto e das quais gostaria de mencionar algumas.

No âmbito do PEI, quero agradecer à Truewind pela forma como me receberam e integraram nos projectos, bem como pelo todo o apoio que me deram e por todos os recursos que me disponibilizaram. Gostaria de destacar o contributo de duas pessoas, o Gonçalo Matos e o Hélder Faria, pelo tempo que dedicaram a orientar-me durante todo projecto, desde o seu desenvolvimento até à elaboração do relatório.

Gostaria ainda de agradecer à professora Isabel Nunes por me ter orientado durante este projecto e pela sua revisão desta tese.

A todos os meus amigos por estarem sempre presentes e por simplesmente por serem meus amigos aposto que alguns ainda vão ler "por alto" esta tese.

Finalmente agradeço à minha família, à minha mãe, ao meu pai, ao meu mano e à minha *bhabhi* (cunhada) por todo o apoio incondicional durante todo o percurso.

Para a minha “mummy” e “papa”.

Resumo

Com a crescente competitividade do mercado, observa-se um maior investimento por parte das organizações na aquisição e implementação de sistemas aplicacionais empresariais – *ERP* –, que são actualmente peças vitais na operação e planeamento de uma organização. Este tipo de sistemas de informação é altamente personalizável e proporciona uma integração absoluta entre as diversas áreas de negócio de uma organização. Muitas empresas optam pelo sistema *Oracle E-Business Suite*, que por ser um sistema genérico, requer configurações complexas, reservadas a recursos humanos especializados para o efeito. Estas configurações, em casos específicos, podem revelar-se ineficientes, sendo por vezes benéfico o desenvolvimento de aplicações periféricas em redor do núcleo do sistema.

O objectivo do projecto aqui descrito foi optimizar e simplificar as tarefas realizadas no *ERP Oracle E-Business Suite*, através da implementação de uma aplicação nessa camada aplicacional periférica. Estas aplicações, desenvolvidas para processos de negócio específicos, oferecem um ambiente *Web* mais amigável e, tirando proveito das potencialidades disponibilizadas pelo *ERP*, podem resolver de uma forma simples, problemas mais complexos associados ao processo de negócio em questão.

O trabalho descrito neste relatório consiste no desenvolvimento de uma aplicação, em tecnologia *OutSystems*, que implementa as acções mais importantes do módulo de gestão de património (*Fixed Assets*) do *Oracle E-Business Suite*. Para a sua realização foi utilizada a metodologia ágil *SCRUM*, que privilegia uma entrega mais rápida das funcionalidades mais importantes e tem a capacidade de alinhar as entregas com as alterações de prioridades, naturais à evolução de um projecto.

Palavras-chave: *OutSystems*, *Oracle E-Business*, *SCRUM*, Gestão de Activos, Desenvolvimento Ágil.

Abstract

In the increasingly competitive market, there is a greater investment by organizations in the acquisition and implementation of business application systems – ERP –, which are now vital parts in the operation and planning of an organization. This type of information systems is highly customizable and provides a complete integration between the various business areas of an organization. Many companies choose the system Oracle E-Business Suite, which is a generic system that requires complex configuration, reserved for specialized human resources for this purpose. These settings, in specific cases, may prove ineffective, and sometimes is beneficial to the development of peripheral applications around the core of the system.

This project aimed to optimize and simplify the tasks performed in ERP Oracle E-Business Suite, by implementing an additional layer applied in this peripheral. These applications, developed for specific business processes, offer a more user-friendly *Web* environment and taking advantage of capabilities provided by ERP can solve in a simple form, more complex problems associated with the business process in question.

The work described in this report is to develop an application in OutSystems technology, which implements the most important activities of the asset management module (Fixed Assets) of the Oracle E-Business Suite. For its implementation we used the SCRUM agile methodology, which favors a quicker delivery of the more important and has the ability to align supply with changes in priorities, the natural evolution of a project.

Keywords: OutSystems, Oracle E-Business, SCRUM, Asset Management, Agile Development.

Conteúdo

Capítulo 1	Introdução	1
1.1	Motivação	1
1.2	Objectivos	2
1.3	Organização do documento	3
1.4	Empresa de acolhimento.....	4
1.5	Integração profissional	4
Capítulo 2	Contexto e enquadramento tecnológico	7
2.1	<i>ERP (Enterprise Resource Planning)</i>	7
2.1.1	<i>Oracle E-Business Suite</i>	7
2.2	<i>OutSystems</i>	9
2.2.1	Plataforma de desenvolvimento	9
2.2.2	<i>Agile Network</i>	11
2.3	<i>SQL Server 2005</i>	14
2.4	<i>C#</i>	14
2.5	<i>SQL</i>	15
2.6	<i>PL/SQL</i>	15
Capítulo 3	Metodologia e planeamento.....	17
3.1	Metodologia de desenvolvimento	17
3.2	Planeamento inicial do projecto	20
3.3	Planeamento final do projecto	22
3.3.1	<i>Product backlog</i>	22
3.3.2	<i>Sprint backlog</i>	24
3.3.3	Mapa de <i>Gantt</i>	26
Capítulo 4	Trabalho realizado.....	29
4.1	Considerações gerais	29

4.2	Análise de requisitos	30
4.3	Arquitectura do sistema	33
4.4	Carregamento de dados	34
4.4.1	Tratamento de dados em <i>Excel</i>	34
4.4.2	Migração de dados de <i>Excel</i> para o módulo <i>FA</i> da <i>Oracle</i>	35
4.5	Conector <i>Oracle</i>	37
4.6	Funcionalidade implementadas.....	38
4.6.1	Listagem de activos	38
4.6.2	Pesquisa de activos	44
4.6.3	Consulta de activos	45
4.6.4	Campos flexíveis	46
4.6.5	<i>Package PL/SQL</i> para adição do activo.....	51
4.6.6	Adição simples do activo	55
4.6.7	Extensões criadas em <i>OutSystems</i>	59
4.7	Análise de outras funcionalidades.....	61
4.7.1	Adição de activo (simples e detalhada)	62
4.7.2	Adição em bloco e classificação de activos	64
4.7.3	Abate de activos	69
4.8	Transição da aplicação <i>OutSystems</i> 4.2 para 5.0	70
4.9	Testes.....	71
4.9.1	Qualidade dos dados	72
4.9.2	Testes unitários	73
4.9.3	Testes funcionais	74
Capítulo 5	Conclusões e trabalho futuro	79
5.1	Conclusões	79
5.2	Trabalho futuro	81
Bibliografia e Referências		83

Glossário	85
Anexos	87
A. <i>Sprints Backlogs</i>	87
<i>Sprint Backlog 1</i>	87
<i>Sprint Backlog 2</i>	87
<i>Sprint Backlog 3</i>	87
<i>Sprint Backlog 4</i>	88
<i>Sprint Backlog 5</i>	88
<i>Sprint Backlog 6</i>	88
<i>Sprint Backlog 7</i>	89
<i>Sprint Backlog 8</i>	89
<i>Sprint Backlog 9</i>	89
<i>Sprint Backlog 10</i>	90
B. <i>Product Backlog</i>	91
C. Tabela CPRX_FA_MASSADD_TDP.....	95
D. Tabela das funções e <i>queries</i> dos testes unitários	98

Lista de Figuras

Ilustração 1 - Plataforma <i>OutSystems</i> Ágil	10
Ilustração 2 - <i>OutSystems Agile Network</i>	12
Ilustração 3 – Trilhos de certificação.....	14
Ilustração 4 - Metodologia <i>scrum</i>	18
Ilustração 5 - Mapa de <i>Gantt</i>	21
Ilustração 6 - Excerto do <i>product backlog</i>	23
Ilustração 7 - <i>Sprint backlog</i> 6 (1/2)	24
Ilustração 8 - <i>Sprint backlog</i> 6 (2/2)	25
Ilustração 9 - Mapa de <i>Gantt</i> final	26
Ilustração 10 - Arquitectura do sistema	33
Ilustração 11 - Fluxo de criação do Conector <i>Oracle</i>	38
Ilustração 12 - Listagem de activos	39
Ilustração 13 - Exemplo da lista de activos do ficheiro <i>Excel</i>	40
Ilustração 14 - Fluxo do processo do ecrã listagem de activos	41
Ilustração 15 - <i>Query</i> da lista de activos	43
Ilustração 16 - Detalhe do filtro.....	44
Ilustração 17 - Exemplo de pesquisa	45
Ilustração 18 - Consulta do activo	46
Ilustração 19 - Conceitos básicos do campo flexível.....	47
Ilustração 20 - Ecrã editar	49
Ilustração 21 - Classificação da localização	50
Ilustração 22 - Esquema de um <i>package</i>	52
Ilustração 23 - Modelo de dados das tabelas auxiliares	53
Ilustração 24 - Adição simples	55
Ilustração 25 - Exemplo de falha na adição do activo	56
Ilustração 26 - Fluxo de adição simples.....	58
Ilustração 27 - Adição simples com sucesso	59
Ilustração 28 - Extensão <i>Assets_Entities</i>	59
Ilustração 29 - Extensão <i>Assets_Actions</i>	61

Ilustração 30 - Modelo de dados adição do activo	64
Ilustração 31 - Fluxo de adição em bloco até classificação de património	66
Ilustração 32 - Modelo de dados de adição em bloco.....	69
Ilustração 33 - Ecrã de testes.....	74
Ilustração 34 - Teste funcional para a pesquisa.....	77
Ilustração 35 - <i>Sprint backlog</i> 1 (1/2)	87
Ilustração 36 - <i>Sprint backlog</i> 1 (2/2)	87
Ilustração 37 - <i>Sprint backlog</i> 2 (1/2)	87
Ilustração 38 - <i>Sprint backlog</i> 2 (2/2)	87
Ilustração 39 - <i>Sprint backlog</i> 3 (1/2)	87
Ilustração 40 - <i>Sprint backlog</i> 3 (2/2)	88
Ilustração 41 - <i>Sprint backlog</i> 4 (1/2)	88
Ilustração 42 - <i>Sprint backlog</i> 4 (2/2)	88
Ilustração 43 - <i>Sprint backlog</i> 5 (1/2)	88
Ilustração 44 - <i>Sprint backlog</i> 5 (2/2)	88
Ilustração 45 - <i>Sprint backlog</i> 6 (1/2)	88
Ilustração 46 - <i>Sprint backlog</i> 6 (2/2)	88
Ilustração 47 - <i>Sprint backlog</i> 7 (1/2)	89
Ilustração 48 - <i>Sprint backlog</i> 7 (2/2)	89
Ilustração 49 - <i>Sprint backlog</i> 8 (1/2)	89
Ilustração 50 - <i>Sprint backlog</i> 8 (2/2)	89
Ilustração 51 - <i>Sprint backlog</i> 9 (1/2)	89
Ilustração 52 - <i>Sprint backlog</i> 9 (2/2)	89
Ilustração 53 - <i>Sprint backlog</i> 10 (1/2)	90
Ilustração 54 - <i>Sprint backlog</i> 10 (2/2)	90
Ilustração 55 - <i>Product backlog</i> (1/5)	91
Ilustração 56 - <i>Product backlog</i> (2/5)	92
Ilustração 57 - <i>Product backlog</i> (3/5)	93
Ilustração 58 - <i>Product backlog</i> (4/5)	94
Ilustração 59 - <i>Product backlog</i> (5/5)	95

Lista de Tabelas

Tabela 1 - Plano de trabalho	20
Tabela 2 - Tarefas realizadas até à entrega do relatório preliminar	21
Tabela 3 - Requisitos funcionais	32
Tabela 4 - Excerto da tabela CPRX_FA_MASSADD_TDP	36
Tabela 5 - Descrição de classificação da localização	48
Tabela 6 - Tabela com dados necessário para adição simples do activo	62
Tabela 7 - Lista de teste funcional	76
Tabela 8 - Tabela CPRX_FA_MASSADD_TDP	98
Tabela 9 - Tabela das funções e <i>queries</i> dos testes unitários	99

Capítulo 1

Introdução

Este relatório apresenta e descreve o projecto realizado no âmbito da disciplina do Projecto de Engenharia Informática, leccionada na Faculdade de Ciências da Universidade de Lisboa no ano lectivo de 2009/2010. Este projecto foi realizado na empresa Truewind - Tecnologias de Informação, S.A. - [1] e focou-se no desenvolvimento de uma aplicação para gestão de património, em tecnologia *OutSystems*.

1.1 Motivação

Actualmente, existem diversas plataformas de *software* desenvolvidas para integrar os vários departamentos de uma empresa. Estas plataformas, conhecidas como *ERP* (*Enterprise Resource Planning*), são actualmente peças vitais na operação e planeamento de uma organização de média e grande dimensão, pois permitem um controlo e gestão eficaz das suas operações, onde se inclui naturalmente a gestão de património. Temos como exemplos de *ERPs* o *SAP* (Sistemas, Aplicativos e Produtos para Processamento de Dados), o *Oracle E-Business Suite* e o *Navision*.

Estas aplicações empresariais têm como principal vantagem o facto de proporcionarem a integração de múltiplas áreas de operação, tais como a contabilidade, gestão de património, facturação, entre outros serviços, optimizando os processos de decisão e eliminando a redundância de actividades. Entre as desvantagens deste tipo de *Business Suites*¹, encontram-se o seu elevado grau de parametrização e configuração, o seu

¹ *Business Suites* - Conjunto de aplicações de negócio integradas que permitem executar estratégias empresariais e de tecnologias de informação.

elevado custo de implementação e o facto de requererem uma formação intensa dos seus utilizadores finais. Esta formação intensa é necessária devido ao elevado grau de complexidade envolvido na realização de certas operações, nas quais pode haver a necessidade de navegar por mais que um ecrã ou módulo. Esta realidade mostra que existem lacunas nos processos e que é possível efectuar a sua simplificação e optimização através da reunião de funcionalidades e identificação de padrões.

Tendo em conta a especificidade própria de cada organização cliente, pretendeu-se com este trabalho o desenvolvimento de novas interfaces de acesso às funções disponibilizadas pelo *ERP*. Estas interfaces devem ter baixo custo, ser rápidas e optimizadas para a organização em causa. A solução residiu na adopção de metodologias ágeis² e *frameworks* que promovem a rápida codificação e *deployment*³ de aplicações que pudessem suprimir esta lacuna. A *framework* de desenvolvimento *OutSystems*, com geração automática de código em *Microsoft .Net* ou *Java* apresenta-se hoje como uma plataforma de desenvolvimento rápido de aplicações periféricas. Estas aplicações externas devem reutilizar a lógica de negócio já implementada no sistema nuclear da organização, permitindo assim um acesso ao repositório de informação das organizações.

1.2 Objectivos

Pretendeu-se com este projecto, além de uma formação base forte no desenvolvimento de aplicações com a tecnologia *OutSystems*, a construção de uma aplicação de gestão de património que, interagindo com o modelo de dados da estrutura centralizada de um *ERP*, permite optimizar a gestão dos processos financeiros ligados ao controlo do património de uma empresa. Objectivo não foi o de substituir integralmente o módulo de gestão de património do *ERP*, mas sim complementá-lo, dotando-o de processos mais optimizados e simplificados para o utilizador final.

² Metodologias Ágeis - Conjunto de métodos de desenvolvimento de *software* que proporcionam uma estrutura para coordenar projectos de engenharia de *software*.

³ Deployment - Conjunto de actividades que fazem com que uma aplicação, ou componentes de *software*, fique pronta para uso.

Principais objectivos do projecto:

- Análise do módulo aplicacional do *ERP Oracle E-Business Suite Fixed Assets*⁴ para a extracção do modelo de dados e métodos que permite a execução de funcionalidades do *ERP*;
- Definição de estruturas de dados complementares e codificação de módulos desenhados em tecnologia *OutSystems* para a invocação das funcionalidades do *ERP*, privilegiando a prototipagem rápida indispensável à garantia da viabilidade da solução;
- Validação e avaliação do funcionamento das aplicações desenvolvidas por oposição às funcionalidades standard do *ERP*.

1.3 Organização do documento

Este documento encontra-se organizado da seguinte forma:

- Capítulo 2 – Contexto e Enquadramento Tecnológico
Neste capítulo são descritas as tecnologias necessárias para a realização do projecto.
- Capítulo 3 – Metodologia e Planeamento
Neste capítulo é apresentado um planeamento do projecto bem como a metodologia utilizada para a sua realização.
- Capítulo 4 – Trabalho realizado
Neste capítulo é descrita a pesquisa efectuada com o intuito de identificar e analisar os requisitos principais do projecto e são descritas as funcionalidades implementadas.
- Capítulo 5 – Conclusões e trabalho futuro
Neste capítulo é feita uma breve conclusão acerca do trabalho realizado, onde é exposta uma visão crítica do projecto e apresentada alguma motivação para trabalho futuro de extensão ou melhoramento do produto realizado.
- Bibliografia e Referências

⁴ *Oracle E-Business Suite Fixed Assets* – É uma solução completa de gestão de activos apresentada pela *Oracle*.

Referência a documentos, livros e páginas Web que serviram de fonte de consulta para a realização do projecto.

1.4 Empresa de acolhimento

A Truewind - Tecnologias de Informação, S.A. foi criada em 2008. É uma empresa de consultoria focada na rápida criação de valor para os clientes através da utilização de soluções ágeis. Tem como objectivos prestar serviços ao mercado das tecnologias de informação (TI) contando para isso com especialistas certificados no desenvolvimento e suporte de soluções baseadas em tecnologias *Java*, *Microsoft*, *Oracle* e *OutSystems*.

1.5 Integração profissional

No dia 14 de Setembro de 2009 dei início ao estágio nas instalações da Truewind. O primeiro dia na empresa teve como objectivo dar a conhecer a estrutura da empresa, as competências de cada unidade de negócio, a sua organização e métodos de trabalho. No próprio dia foi-me atribuído um portátil com o objectivo de nele instalar as ferramentas adequadas ao desenvolvimento do projecto proposto.

Após este período de ambientação, iniciei a formação *online* sobre a tecnologia *OutSystems* tendo adquirido uma certificação com o grau de *Trainee Developer*⁵. Mais contexto sobre esta formação será dado no Capítulo 2.

Depois desta etapa fui integrado numa equipa de *OutSystems* que se encontrava nas instalações de um cliente da Truewind. Aqui foram executadas tarefas relacionadas com a plataforma *OutSystems* e realizada mais formação *online* e *on-job* de forma a solidificar e expandir os conhecimentos obtidos anteriormente. A Truewind organizou também um conjunto de *workshops*⁶ sobre *Oracle E-Business-Suite* e *ADO.NET*⁷ que

⁵ *Trainee Developer* - Um dos níveis de certificação da *OutSystems*.

⁶ *Workshop* - Uma reunião de grupos de trabalho interessados num determinado tema e onde é realizada uma discussão e/ou apresentação do referido tema.

⁷ *ADO.NET* - Conjunto de classes definidas pela .NET framework que pode ser utilizado para aceder aos dados armazenados numa base de dados remota.

contribuíram para a familiarização dos vários componentes relacionados com o projecto.

Capítulo 2

Contexto e enquadramento tecnológico

Este capítulo enquadra o trabalho realizado e descreve as tecnologias utilizadas na sua realização, introduzindo também um conjunto de conceitos importantes que servirão de base para o desenvolvimento de uma aplicação de gestão de património.

2.1 *ERP (Enterprise Resource Planning)*

Os *ERPs* [2] são sistemas de informação que centralizam e integram os dados e processos de diversos departamentos de uma organização num único sistema. Em termos gerais, são compostos por um conjunto de actividades e processos de negócio, executados numa plataforma de *software*, com o objectivo de automatizar e armazenar todos os processos de gestão de uma empresa nas mais importantes fases do seu negócio.

A integração dos diversos departamentos ou entidades pode ser vista, a título de exemplo, na colaboração no desenvolvimento de um produto interno, na interacção de fornecedores e clientes, na gestão contabilística e financeira, ou na gestão de projectos.

2.1.1 *Oracle E-Business Suite*

A *Oracle Corporation*⁸ desenvolve e comercializa uma linha integrada de produtos de *software* para gestão de dados, gestão de negócio, apoio à decisão e automação de processos de organizações.

⁸ *Oracle Corporation* - É uma companhia que desenvolve *softwares* empresariais. O seu principal produto é o sistema de gestão de base de dados relacionais chamado *Oracle*.

As aplicações *Oracle* fornecem o *E-Business Suite* [3], que é composto por um conjunto integrado de mais de 70 módulos de *software* para gestão financeira, gestão de cadeia de abastecimento, produção, projecto de sistemas, recursos humanos, vendas e gestão de serviços.

O *Oracle E-Business Suite (OEBS)* é o conjunto mais abrangente de integração de aplicações de negócio global, que dispõe de:

- O mais completo portfólio de *business intelligence*⁹ integrada;
- A plataforma de negócios mais globalmente adaptável;
- A estratégia focada nas aplicações do cliente.

Do *Oracle E-Business Suite* destacam-se as seguintes aplicações: *Supply Chain Management, Financials, Projects, Human Resources* e *Front Office Applications*. Destas aplicações, vamos focar-nos no *Financials* que permite facilitar a gestão de activos, propriedades e recebimentos.

A aplicação *Oracle Financial* [4] é composta por um conjunto de módulos. Estes são o *General Ledger (GL)*, *Accounts Receivable (AR)*, *Accounts Payable (AP)*, *Fixed Assets (FA)*, *Financial Intelligence (FI)*, *Inventory Management* e *Daily Business Intelligence (DBI)*. O módulo mais relevante para este projecto, que permite efectuar a gestão do património, é o *Fixed Assets*.

O Oracle Fixed Assets (FA) [5] automatiza a gestão de activos e simplifica as tarefas de contabilidade. Usando uma fonte única de activos de uma organização (com dados adquiridos através de aplicações Oracle, bem como de outros sistemas externos), o FA do Oracle permite uma visão sobre o património total da organização.

⁹ *Business Intelligence* - Descreve a habilidade das empresas em organizar, agrupar, analisar e monitorizar informação que oferece suporte à gestão de negócios.

Um outro módulo com ligação ao *Fixed Assets*, e cuja análise é relevante para o projecto, é o *Accounts Payable*. Sendo este módulo responsável pela gestão dos fornecedores e facturas, é possível criar activos no *FA* através da introdução de uma factura, ficando estes em fila de espera para mais tarde serem classificados.

As tarefas de gestão de património tais como transferência de activos, alienações, classificações, ajustes financeiros e a conversão de dados podem ser rationalizados com o fluxo de negócios automatizados. Para tal, estão disponíveis relatórios de contabilidade, operações e registo, que facilitam a reconciliação e análise de dados.

2.2 *OutSystems*

A *OutSystems* [6] é uma empresa Portuguesa que actua no mercado de desenvolvimento ágil de software e que fornece uma plataforma que permite às empresas desenvolver, alterar e manter aplicações empresariais.

A missão da empresa é oferecer uma tecnologia que permite alterar as aplicações em qualquer fase do seu ciclo, reduzindo os seus custos de entrega e agilizando a gestão de aplicações de negócio *Web*. Para isso, são utilizadas metodologias ágeis, nomeadamente a metodologia *OutSystems*, criada com base na metodologia *Scrum*¹⁰.

2.2.1 Plataforma de desenvolvimento

O poder da plataforma ágil começa com um conjunto de ferramentas e serviços que automatizam totalmente o processo de entrega de aplicações de negócio. Estas integram-se com os sistemas existentes e alcançam os utilizadores através da *Web*, *e-mail* e interfaces de equipamentos móveis.

¹⁰ *Scrum* - Metodologia ágil e flexível para o desenvolvimento de qualquer produto ou gestão de qualquer trabalho.

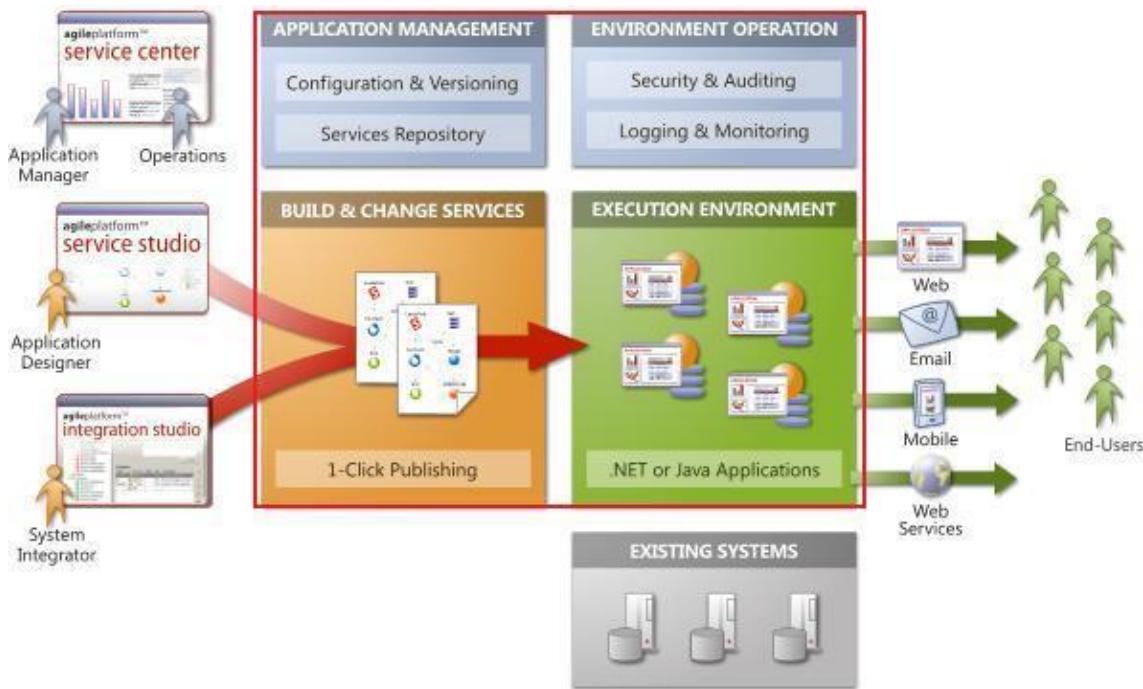


Ilustração 1 - Plataforma *OutSystems* Ágil

A plataforma *OutSystems* é constituída pelos seguintes módulos mostrados na Ilustração 1:

- **Service Studio** – É um ambiente de desenvolvimento visual integrado, orientado para o desenvolvimento de aplicações de negócio Web sujeitas a rápidas mudanças nos processos de negócio. O *Service Studio* reúne todos os componentes necessários para a sua construção, sem haver a necessidade de recorrer a qualquer código escrito. Esta ferramenta permite criar, modificar e publicar automaticamente as aplicações.
- **Service Center** – É uma consola centralizada Web que permite a gestão e monitorização de toda a plataforma ágil. Através desta consola é possível configurar as políticas de controlo de acesso, realizar um controlo de qualidade sobre as equipas de desenvolvimento e sobre as aplicações desenvolvidas, obter um histórico de todas as versões das aplicações realizadas e monitorizar e auditar a execução de aplicações para detectar e isolar problemas de desempenho e qualidade.
- **Integration Studio** - É um ambiente de trabalho para os programadores criarem componentes personalizados para integrar aplicações, e acessos a bases de dados, externas. Este ambiente inclui assistentes para identificar, mapear e integrar bases de dados, bibliotecas de APIs, componentes SAP e

funciona com *Microsoft Visual Studio*. O *Integration Studio* permite também criar conectores para integrar com sistemas já existentes que podem ser reutilizados em qualquer aplicação *OutSystems* através do *Service Studio*.

- **Platform Server** – É o ambiente onde as aplicações são armazenadas, publicadas e executadas. Este pode ter como base a plataforma *.Net* ou *Java* e disponibiliza um conjunto de serviços para compilar e monitorizar as aplicações. O *Platform Server* encontra-se assinalado a vermelho na Ilustração 1.

2.2.2 Agile Network

O *Agile Network* [7] é um portal que fornece à comunidade de profissionais da *OutSystems* um conjunto completo de serviços relacionados com a utilização da plataforma. Estes serviços incluem ferramentas de gestão de projectos ágeis, material de ajuda a vendas, cursos de formação *online*, fórum técnico, área de *downloads*, gestão de parcerias e requisição de licenças de utilização da plataforma.

Uma visão da página inicial de um utilizador com conta na rede da *OutSystems* é apresentada na Ilustração 2.

Welcome to the OutSystems Agile Network!

OutSystems Agile Network offers customers and partners a comprehensive set of on-demand services supporting the full application life cycle management processes. With easy to use and complete programs available online, customers and partners can easily learn how to improve their processes leveraging the Agile Platform.

Download Sales Kit
This complete and easy to use Sales Kit helps you to successfully promote and sell OutSystems Web Business Solutions.

Scope and Size Agile Projects
Access your Project Sizings history or create new sizings from scratch easily and accurately. Take advantage of Agile pre-sized projects.

Deliver and Manage Agile Projects
Deliver Agile Projects in fast changing environments with full control driving agility to its full benefits. Manage your backlog, control your risks, track project evolution with constant customer visibility.

Online Training and Certification
Whether you want to improve your skills, projects or resume, our training and certification programs help you achieve your goals.

Access technical resources on the TechCenter
Visit our TechCenter to download samples, get new technical notes, and search our online Forums.

Take the agilenetwork Tour
Watch this video tour to learn how to use the **agilenetwork** to scope, size and manage your Agile Projects.
(5 min)

[Watch the agilenetwork Video Tour »](#)

Ilustração 2 - OutSystems Agile Network

Na vertente de gestão de projectos ágeis, a *Agile Network* facilita a estimativa de um projecto e apoia as equipas de projecto na gestão da entrega das aplicações através dos seguintes serviços:

- **Scope & Size:** os gestores de projecto capturam os requisitos e criam a estimativa de projecto, baseada na combinação do historial do utilizador de alto nível e com padrões pré-definidos. O resultado é um plano completo, que define claramente o esforço e o tempo necessário para entregar o projecto;
- **Deliver & Control:** permite aos gestores de projecto coordenarem todos os aspectos da entrega do projecto ágil: definir *sprints*¹¹, especificar as futuras alterações a implementar e atribuir tarefas de trabalho para a equipa. Todos

¹¹ *Sprints* - Período inferior a 30 dias no qual é realizado um conjunto de tarefas.

os membros da equipa, incluindo o próprio cliente, terão acesso aos últimos estados do projecto;

- **Change & Evolve:** durante todo o projecto, os utilizadores chave são capazes de apresentar os seus *feedbacks*¹² directamente na aplicação. Os gestores de projecto podem incluir esses *feedbacks* no plano de tarefas e negociar as tarefas com os clientes.

A *Agile Network* disponibiliza um conjunto de benefícios como a alta previsibilidade, a visibilidade do projecto nas suas várias fases, relatórios sobre a sua evolução e gestão integrada de múltiplos projectos, assegurando assim um pleno controlo durante a sua execução.

A formação *online* proporcionada pela *Agile Network* permite obter vários níveis de certificação para cada um dos papéis previstos na metodologia *OutSystems*:

- **Developer:** Responsável por codificar e testar uma tarefa do projecto;
- **Delivery Manager:** Responsável pela entrega das funcionalidades do projecto, tanto ao nível de cumprimento de prazos, como ao nível da qualidade;
- **Engagement Manager:** Responsável pela definição do âmbito do projecto e alinhamento de expectativas com o cliente.

Do conjunto de certificações representadas na Ilustração 3, vamos focar-nos no *Developer* que consiste em obter formação para desenvolver aplicações de negócio *Web*. Esta certificação tem um conjunto de cinco níveis, dos quais o *Trainee Developer* (um principiante em *OutSystems* que entende os conceitos básicos da plataforma e dos métodos ágeis) até ao nível mais alto o *Expert Developer* (programador sénior de confiança que pode resolver tudo, com experiência, com capacidade de projectar soluções de qualidade de forma autónoma. Pode resolver todas as questões relacionadas com plataforma técnica e tem um conhecimento significativo sobre metodologia ágil).

¹² *Feedbacks* – Processos que consistem em fornecer informação a uma pessoa ou grupo sobre o desempenho, de modo a reorientar e/ou estimular uma ou mais acções no sentido de atingir objectivos propostos.

Até ao momento o nível de certificação adquirido em *OutSystems* foi o de *Trainee Developer*.



Ilustração 3 – Trilhos de certificação

2.3 SQL Server 2005

O *SQL Server 2005* [8] criado pela *Microsoft* é uma plataforma de gestão de bases de dados abrangente que oferece ferramentas de gestão de dados. A principal linguagem usada é o *Transact-SQL (T-SQL)*, implementação do standard *SQL* usada pela *Microsoft*. O motor de base de dados *SQL Server 2005* proporciona um armazenamento mais seguro e fiável tanto para dados relacionais como estruturados, permitindo-lhe criar e gerir aplicações de dados de elevada disponibilidade e desempenho.

2.4 C#

C# é uma linguagem de programação orientada a objectos desenvolvida pela *Microsoft* como parte da plataforma *.NET*, sendo uma linguagem muito influenciada pelo *Java* e *C++*.

Esta linguagem pretende preencher algumas lacunas de seguranças existentes em outras linguagens tais como, por exemplo, a protecção da memória devido à não utilização de apontadores em modo seguro. É a linguagem mais utilizada no mundo para aplicações na *internet*.

A plataforma de desenvolvimento *OutSystems* gera automaticamente código em *C#* e permite também adicionar métodos desenvolvidos em *C#* para a plataforma.

2.5 SQL

SQL [9], acrónimo de *Structured Query Language*, ou Linguagem de Consulta Estruturada, é uma linguagem de pesquisa declarativa para bases de dados, que permite a recolha e gestão da informação presente num sistema de base de dados.

2.6 PL/SQL

PL/SQL [10] é um acrónimo para *Procedural Language extension of SQL*, linguagem de procedimentos para *SQL*. Esta linguagem é uma combinação de *SQL* com as características processuais das linguagens de programação. Foi desenvolvido pela *Oracle Corporation* no início dos anos 90 para melhorar as capacidades do *SQL*.

Capítulo 3

Metodologia e planeamento

3.1 Metodologia de desenvolvimento

A metodologia utilizada para o desenvolvimento da aplicação foi o *scrum* [11]. Esta opção foi tomada tendo em conta as características do projecto, a experiência adquirida em pequenas tarefas sobre a sua importância e aplicabilidade, e pelo sucesso no histórico da empresa da sua aplicação no desenvolvimento de outros projectos.

O *scrum* é uma metodologia ágil e flexível para o desenvolvimento de um produto, ou mais genericamente, para a gestão e controlo de um trabalho. É uma metodologia de desenvolvimento de *software* que utiliza uma prática iterativa e incremental, a qual proporciona uma percepção diária acerca do estado do desenvolvimento do produto.

O *scrum* é uma metodologia de desenvolvimento de projectos em equipa que se foca em entregar rapidamente ao cliente as funcionalidades de negócio com maior valor, adaptando-se perfeitamente a projectos em que os requisitos mudam rapidamente ou com vários ritmos de progressão.

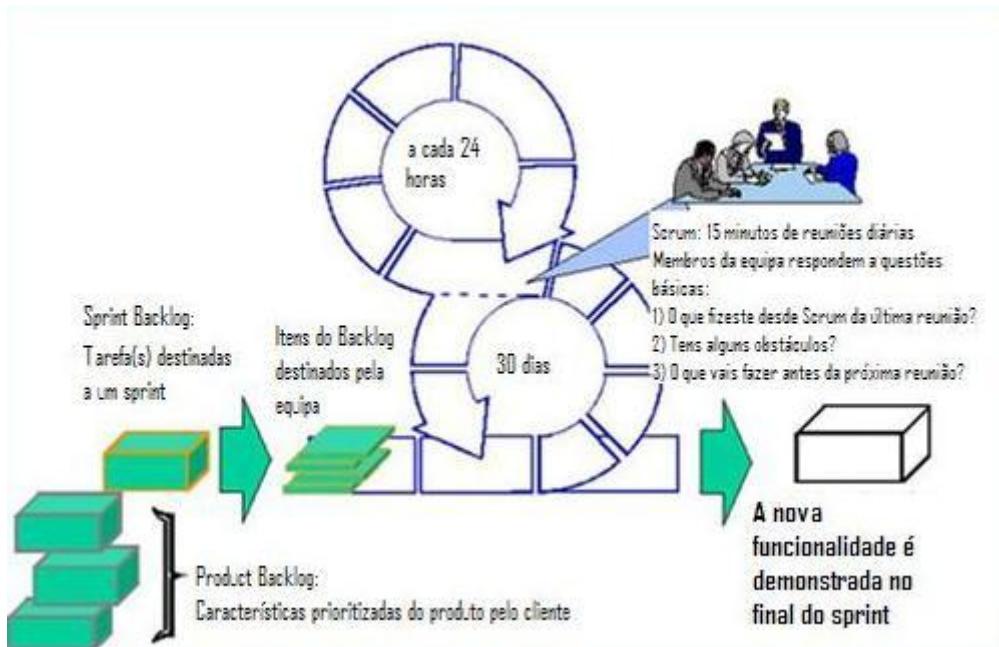


Ilustração 4 - Metodologia scrum

A Ilustração 4 demonstra o funcionamento da metodologia *scrum* e os seus principais componentes. Os seus conceitos mais importantes são:

- **Product backlog:** Lista de todas as tarefas a serem realizados no âmbito do projecto, ordenadas por prioridade.
- **Sprint:** Período inferior a 30 dias (tipicamente de 2 a 3 semanas) no qual é realizado um conjunto de tarefas. No final, obtém-se uma entrega tangível e incremental.
- **Sprint backlog:** Conjunto de tarefas a realizar num *sprint*, ordenadas por prioridade.
- **Scrum meeting:** Reunião diária em que o progresso e os obstáculos do trabalho são revistos.
- **Scrum meeting rules:** Protocolo para realização de reuniões *scrum* diárias.

Tal como é demonstrado na Ilustração 4 esta metodologia propõe que um projecto seja iniciado com uma lista de funcionalidades prioritzadas do produto desejado pelo cliente - é uma lista dinâmica constituída por um conjunto de frases simples que especificam as funcionalidades desejadas por parte do utilizador do produto final - esta lista pode ser

modificada pelo cliente a qualquer momento. A lista é dividida em conjuntos de tarefas que são desenvolvidas ao longo de ciclos de 30 dias denominados por *sprint*.

Diariamente são realizadas reuniões de 15 minutos entre os membros da equipa para discutir questões relacionadas com o progresso e os obstáculos do projecto.

No final de cada ciclo é implementado um conjunto de funcionalidades que compõem uma parte funcional do produto final e que pode ser testado pelo cliente. Desta forma é possível obter um *feedback* do cliente que permite ajustar o planeamento do projecto. Este ciclo repete-se até à entrega do produto final desejado pelo cliente.

A metodologia *scrum* define os três papéis chave seguintes:

- **Product Owner:** É a pessoa responsável pelo *product backlog* – lista de tarefas definidas através do levantamento de requisitos – utilizada para expressar quais as funcionalidades com maior prioridade e comunicar as tarefas aos restantes elementos da equipa. É igualmente responsável por definir os *sprints* e realizar demonstrações das diversas fases do produto até ao desenvolvimento do produto final.
- **Scrum Master:** É o gestor da equipa que garante que as tarefas sejam realizadas pelos diversos membros sem qualquer tipo de inconveniente, que é responsável por que a equipa siga as boas práticas da metodologia *scrum*. É o representante da equipa perante o *product owner* e protege a equipa de interferências externas.
- **Team Member:** É o elemento da equipa de desenvolvimento que tem como principais objectivos o desenvolvimento do produto e o de tornar as tarefas do *sprint backlog* em funcionalidades potencialmente concluídas no fim de cada *sprint*.

A equipa de trabalho era constituída por apenas um elemento – o autor do relatório. O trabalho foi mediado por outros dois elementos – o co-orientador da empresa com formação em *OutSystems* e um sénior da empresa perito em *Oracle E-Business Suite*. O papel desempenhado pelos dois séniores consistia em acompanhar o trabalho realizado pelo autor, através da elucidação de questões relacionadas com determinados

componentes do projecto, do esclarecimento sobre o uso de cada uma das ferramentas envolvidas e da ajuda na tomada de decisão em algumas vertentes do projecto.

A metodologia usada adaptou-se perfeitamente à realidade do projecto e à equipa de trabalho. As reuniões diárias revelaram-se importantes pois foram uma forma de controlar o trabalho realizado. Em algumas, foi possível reunir simultaneamente o co-orientador e o sénior experiente em *OEBS*, com o objectivo de ajudar o autor a suprir os obstáculos encontrados e dar orientação ao longo do projecto.

3.2 Planeamento inicial do projecto

Esta secção descreve o plano de trabalho inicial, bem como as tarefas que foram realizadas até à entrega do relatório preliminar.

Na Tabela 1 é apresentado o plano de trabalho geral, onde se podem ver as macro-tarefas necessárias para realizar o trabalho proposto.

Tarefa	Período	Duração
Integração na empresa	14-Set-2009 / 14-Set-2009	1 dia
Formação e certificação na tecnologia <i>OutSystems</i> e complementares em <i>.Net</i>	15-Set-2009 / 13-Nov-2009	2 meses
Desenvolvimento de aplicação usando metodologia ágil <i>OutSystems</i>	16-Nov-2009 / 14-Mai-2010	6 meses
Recolha dos resultados e redacção do relatório	17-Mai-2010 / 17-Jun-2010	1 mês

Tabela 1 - Plano de trabalho

Com base na Tabela 1 foi criado o mapa de *Gantt* representado na Ilustração 5. Este mapa mostra num eixo temporal as tarefas da tabela anterior, bem como as suas dependências.

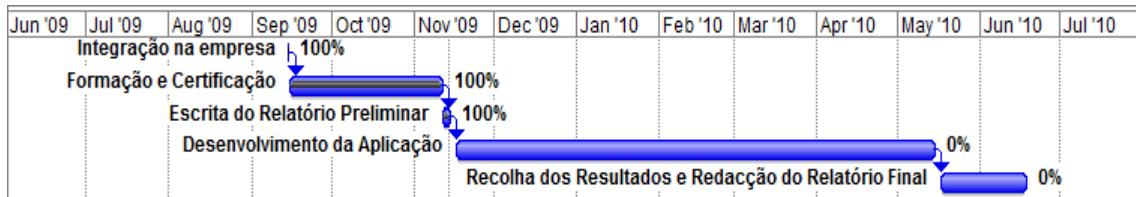


Ilustração 5 - Mapa de Gantt

Na Tabela 2 é descrito o plano detalhado das tarefas realizadas até à entrega do relatório preliminar.

Tarefa	Inicio	Fim
Integração na empresa	14-Set-2009	14-Set-2009
Formação em <i>OutSystems - Developer Course 1</i>	15-Set-2009	22-Set-2009
Recolha de informação sobre <i>ERP</i>	23-Set-2009	25-Set-2009
Informação prática sobre o módulo <i>Fixed Asset</i>	28-Set-2009	30-Set-2009
Recolha de Informação sobre os módulos de <i>Oracle E-Business Suite</i>	1-Out-2009	2-Out-2009
Realização de tarefas em projectos <i>OutSystems</i>	5-Out-2009	8-Out-2009
<i>Workshop Oracle E-Business Suite</i>	9-Out-2009	9-Out-2009
Realização de tarefas em <i>OutSystems</i>	12-Out-2009	15-Out-2009
Formação em <i>OutSystems - Developer Course 2</i>	16-Out-2009	19-Out-2009
<i>Workshop Oracle E-Business Suite</i>	20-Out-2009	20-Out-2009
Formação em <i>OutSystems - Developer Course 2</i>	21-Out-2009	21-Out-2009
Realização de tarefas em <i>OutSystems</i>	22-Out-2009	29-Out-2009
Formação em <i>OutSystems - Developer Course 2</i>	30-Out-2009	2-Nov-2009
Realização de tarefas em <i>OutSystems</i>	3-Nov-2009	3-Nov-2009
<i>Wokshop sobre ADO.NET</i>	4-Nov-2009	4-Nov-2009
Início do relatório preliminar	5-Nov-2009	5-Nov-2009
Realização de tarefas em <i>OutSystems</i>	6-Nov-2009	10-Nov-2009
Relatório preliminar	11-Nov-2009	23-Nov-2009

Tabela 2 - Tarefas realizadas até à entrega do relatório preliminar

3.3 Planeamento final do projecto

Neste capítulo é reflectido todo o trabalho realizado até à entrega do relatório final. Tendo em conta que a metodologia aplicada no projecto foi o *scrum*, de seguida são apresentados o *product backlog* do projecto e um exemplo de um *sprint backlog*.

3.3.1 Product backlog

O *product backlog* é a lista mestre de todas as funcionalidades desejadas para o projecto. No momento em que o projecto de engenharia foi iniciado, não houve uma necessidade de estabelecer detalhadamente todas as tarefas pretendidas para concretização do projecto, pois segundo a metodologia utilizada este detalhe é efectuado com o decorrer do projecto. Inicialmente foram anotadas as tarefas mais evidentes e essenciais, e à medida que o tempo foi decorrendo, houve um amadurecimento sobre as reais necessidades do cliente e do produto. As tarefas resultantes foram adicionadas ao *product backlog* que, tratando-se de uma lista dinâmica, pode mudar e crescer ao longo do tempo até ao final do projecto.

A seguinte imagem (Ilustração 6) representa um excerto do *product backlog* final, onde são descritas todas as tarefas realizadas no Projecto de Engenharia Informática.

Sprint	Componente	Tarefa	Observação	Prioridade	Estimativa (horas)	Estado
	Carregamento de dados no módulo FA		Recepção da lista de activos em formato Excel e carregamento do módulo FA com base nessa lista	1		
2		Elaboração de um documentos único de Excel	Com base em vários ficheiros Excel e inputs do cliente, corrigir e inserir dados de forma a criar um único documento de Excel	1	40	Completa
2		Migração dos dados da folha Excel para a tabela de interface de carregamento em massa de activos (CPRX_FA_MASSADD_ICE_P)	Através de um script SQL (comando SELECT seguido de INSERT)	1	32	Completa
2		Execução do carregamento de dados	Usando o package cprx_fa_load	1	8	Completa
2		Pequeno teste de adição de activos (rápida/detalhada)	Existem dois modos de adição de activo: Rápida: Criação do activo com um nível mínimo de detalhe Detalhada: Criação do activo com um elevado nível de detalhe	1	40	Completa

Ilustração 6 - Excerto do *product backlog*

Como podemos observar na Ilustração 6 no excerto do *product backlog* estão presentes as seguintes sete colunas:

- **Sprint:** Número de *sprint* (períodos de 3 semanas) no final do qual é entregue ao cliente uma versão do produto funcional.
- **Componente:** Nome da componente de *software* a ser desenvolvida.
- **Tarefa:** O conjunto de tarefas que compõem uma componente a ser desenvolvida.
- **Observação:** Anotação que permite perceber e interpretar os componentes.
- **Prioridade:** Nível de preferência dada pelo cliente e pelo programador para cada uma das tarefas. Neste projecto vai de 0 a 4, respectivamente 0 com a maior prioridade e 4 de menor prioridade.
- **Estimativa:** Previsão em horas para o desenvolvimento de cada uma das tarefas.
- **Estado:** A situação em que se encontra a tarefa no actual momento.

3.3.2 Sprint backlog

O *sprint backlog* contém as tarefas a realizar num determinado *sprint*. Este é construído no início de cada *sprint*, a partir do *product backlog*, de onde são identificadas as tarefas mais importantes a ser realizadas nas próximas semanas. Neste projecto, os *sprints* tiveram uma duração de 3 semanas e a maioria das tarefas agendadas para cada *sprint* foram cumpridas com sucesso.

De seguida é feita a descrição de um dos exemplares do *sprint backlog*, sendo que os restantes *sprint backlog*s se encontram em anexo. O *sprint backlog* escolhido foi o *sprint* 6 é constituído pela junção das duas seguintes ilustrações (Ilustração 7 e Ilustração 8)

Product Backlog								
Worktype	Deliverable Area	Item or Group	ID	Sprint Work Item Description	Pri	Driver	Status	Initial
Feature	Campos flexíveis	Campo flexível localização	1	Desenvolvimento	1	Dinis Premji	Completed	32 88
Feature	Campos flexíveis	Campo flexível conta de custos	2	Desenvolvimento	1	Dinis Premji	Completed	8 80
Feature	Campos flexíveis	Campo flexível categoria	3	Desenvolvimento	1	Dinis Premji	Completed	8 72
Feature	Campos flexíveis	Campo flexível chave	4	Desenvolvimento	1	Dinis Premji	Completed	8 64
Feature	Campos flexíveis	Pop-up de lista de segmentos	5	Desenvolvimento	1	Dinis Premji	Completed	32 32
Feature	Campos flexíveis	Pop-up de lista valores do segmento	6	Desenvolvimento	1	Dinis Premji	Completed	32 0

Ilustração 7 - Sprint backlog 6 (1/2)

A Ilustração 7 demonstra uma parte de um sprint que é constituído por um conjunto de nove colunas:

- **Worktype** – O tipo de trabalho que é realizado, neste caso é uma *feature* que se refere a uma característica distintiva de um item de software.
- **Deliverable area** – O item de *software* que será entregue.
- **Product backlog item or group** – A descrição da tarefa a que pertence no *product backlog*.
- **ID** – Identificação numérica única do *sprint*.
- **Sprint work item description** – A descrição do item de *software* a concretizar.
- **Pri** – A prioridade do item do *software*.
- **Driver** – A pessoa que realiza a tarefa.
- **Status** – Indica o estado do item até à data.
- **Initial** – O número de horas inicial reservado para a tarefa.

9-Mar-2010	10-Mar-2010	11-Mar-2010	12-Mar-2010	15-Mar-2010	16-Mar-2010	17-Mar-2010	18-Mar-2010	19-Mar-2010	22-Mar-2010	23-Mar-2010	24-Mar-2010	25-Mar-2010	26-Mar-2010	29-Mar-2010	Total Effort
Spe Left	Day 1 Spe Left	Day 2 Spe Left	Day 3 Spe Left	Day 4 Spe Left	Day 5 Spe Left	Day 6 Spe Left	Day 7 Spe Left	Day 8 Spe Left	Day 9 Spe Left	Day 10 Spe Left	Day 11 Spe Left	Day 12 Spe Left	Day 13 Spe Left	Day 14 Spe Left	Day 15 Spe Left
8 24 InPr	8 16 InPr	8 8 InPr	8 0 Corr	0 0 Corr	0 0 Corr	0 0 Corr	0 0 Corr	0 0 Corr	32						
0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	8
0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	8
0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	0 8 Peni	8
0 32 Peni	0 32 Peni	0 32 Peni	0 32 Peni	0 32 Peni	0 32 Peni	0 32 Peni	0 32 Peni	0 32 Peni	0 32 Peni	0 32 Peni	0 32 Peni	0 32 Peni	0 32 Peni	0 32 Peni	32
0 32 Peni	0 32 Peni	0 32 Peni	0 32 Peni	0 32 Peni	0 32 Peni	0 32 Peni	0 32 Peni	0 32 Peni	0 32 Peni	0 32 Peni	0 32 Peni	0 32 Peni	0 32 Peni	0 32 Peni	32
0 0 Corr	0 0 Corr	0 0 Corr	0 0 Corr	0 0 Corr	0 0 Corr	0 0 Corr	0 0 Corr	0 0 Corr	0 0 Corr	0 0 Corr	0 0 Corr	0 0 Corr	0 0 Corr	0 0 Corr	0

Ilustração 8 - Sprint backlog 6 (2/2)

A Ilustração 8 mostra as restantes colunas do *sprint*, organizado em 15 dias. Cada dia é constituído por 3 colunas:

- **Spent** – O número de horas gasto até ao momento para terminar a tarefa.
- **Left** – O número de horas que falta para terminar a tarefa.
- **Status** (coluna em branco) – O estado em que a tarefa se encontra no momento.

Finalmente, a última coluna do *sprint backlog* é:

- **Total effort** – O esforço total no final do prazo de 3 semanas.

3.3.3 Mapa de Gantt

O mapa de *Gantt* final da Ilustração 9 mostra a forma como as tarefas foram distribuídas pelo tempo. Nesta figura estão descritas todas as actividades e a duração de cada uma delas envolvidas na implementação do Projecto de Engenharia Informática.

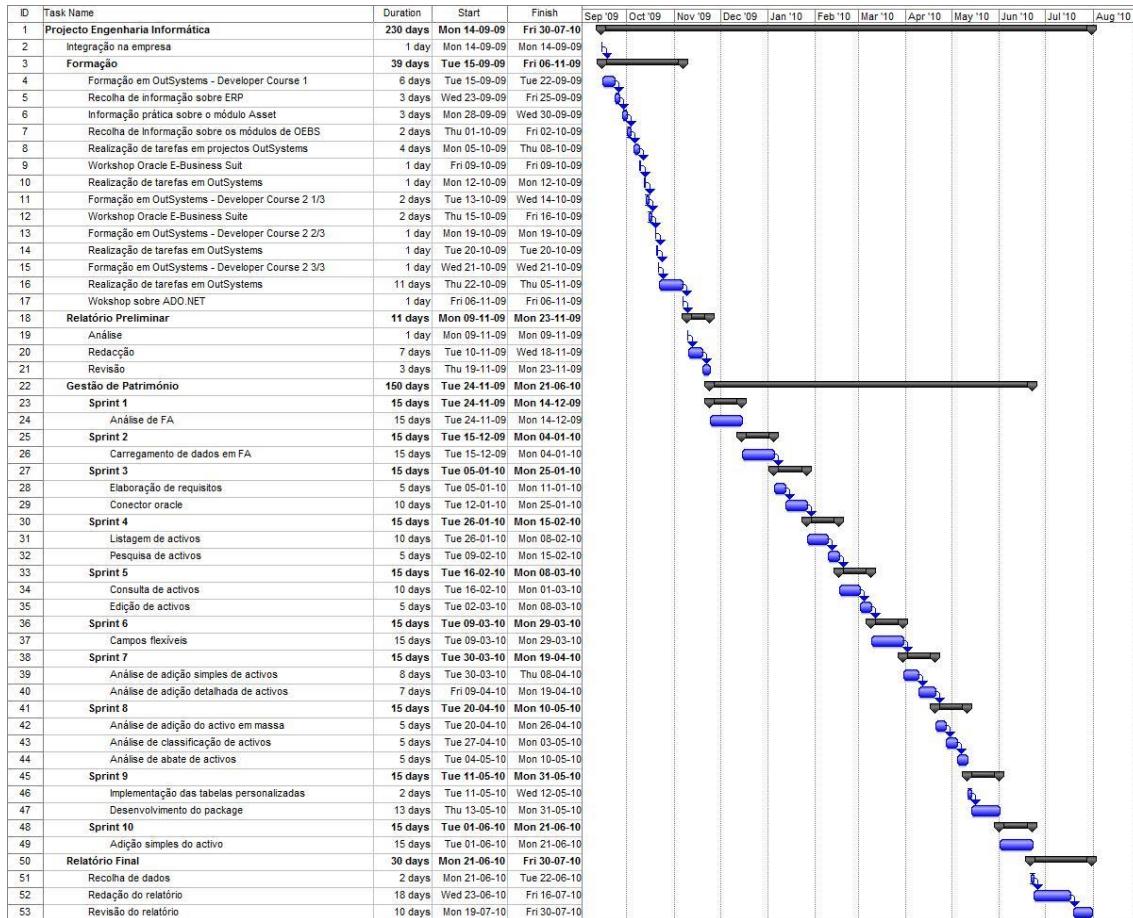


Ilustração 9 - Mapa de Gantt final

Podemos aqui observar que o Projecto de Engenharia Informática é constituído por quatro das seguintes áreas:

- **Formação** – onde são descritas todas as tarefas que tiveram como foco a instrução nas diversas áreas necessárias para a realização do projecto de engenharia informática.
- **Relatório Preliminar** – constituído pelo conjunto de tarefas responsáveis pela elaboração do relatório preliminar.

- **Gestão de Património** – nesta área são descritas todas as actividades necessárias para desenvolver a aplicação de gestão do património. Esta área é dividida em todos os *sprints* realizados até ao momento no desenvolvimento da aplicação. Esta área é constituída pela seguinte colecção de *sprints*:
 - **Sprint 1** - Neste *sprint* foi realizada uma análise ao módulo *FA* do *OEBS*, para obter conhecimento de gestão dos formulários do *Oracle*.
 - **Sprint 2** - Neste *sprint* ocorreu o carregamento e tratamentos dos dados fornecidos através de ficheiros *Excel* relativos a activos da empresa para *OEBS*.
 - **Sprint 3** - Neste *sprint* foi elaborada a lista de requisitos do cliente quanto à aplicação pretendida. Também foi efectuada uma análise ao Conector *Oracle* desenvolvido pela Truewind para a sua utilização na aplicação a desenvolver.
 - **Sprint 4** - Neste *sprint* foi implementado em *OutSystems* o ecrã de listagem de activos e função de pesquisa do activo. Esta é baseada nas diversas características do activo.
 - **Sprint 5** - Neste *sprint* foi implementado em *OutSystems* o ecrã de consulta das características específicas de cada activo.
 - **Sprint 6** – Neste *sprint* foi implementado em *OutSystems* o ecrã dos campos flexíveis.
 - **Sprint 7** – Neste *sprint* foi realizada uma análise às duas formas de adição de um activo proporcionada pelo *OEBS*. Foram efectuados testes nos formulários proporcionados pela aplicação do *FA* do *OEBS*.
 - **Sprint 8** - Neste *sprint* foi realizada uma investigação sobre adição em massa: como se desenrola o fluxo de adição desde o módulo *Accounts Payable* até ao módulo *FA* do *OEBS*. Foi feita uma análise ao processo de classificação dos activos quando chegam ao módulo *FA* do *OEBS*.

- **Sprint 9** - Neste *sprint* foram criadas as tabelas personalizadas e métodos do *package*¹³ na base dados que permitem adicionar as linhas de dados referentes a cada activo, que posteriormente serão utilizados para adicionar um activo usando a *API* da *Oracle*.
 - **Sprint 10** - Neste *sprint* foi implementado em *OutSystems* o ecrã de adição de um activo de forma rápida, empregando os métodos desenvolvidos no *package*.
-
- **Relatório Final** – esta área tem três fases: a de recolha de dados que consistiu numa pesquisa sobre a estrutura e constituição do relatório, a redacção do relatório e, no final, as revisões necessárias até à versão final do relatório.

¹³ *PL/SQL Package* - um *package* é uma colecção de elementos *PL/SQL* onde são agrupadas as funções, cursores e *procedures* dentro de uma sintaxe BEGIN END.

Capítulo 4

Trabalho realizado

Neste capítulo é apresentado o trabalho realizado no projecto. Aqui são descritas as análises efectuadas, o modo como foi implementado o projecto realizado, as escolhas efectuadas para a obtenção dos melhores resultados e as técnicas utilizadas. Também neste capítulo é realizada uma avaliação sobre o custo de desenvolvimento e suporte do projecto, bem como os testes efectuados sobre o mesmo.

4.1 Considerações gerais

As *suites* horizontais de aplicações empresariais – *ERP* –, como o *Oracle E-Business Suite*, são actualmente peças vitais na operação e planeamento de uma organização, permitindo o controlo e funcionamento de uma organização onde se destaca a gestão de património.

O Projecto de Engenharia Informática consistiu numa formação base forte no desenvolvimento de aplicações com a plataforma *OutSystems* e na construção de uma aplicação de gestão de património em tecnologia *OutSystems* cujo modelo de dados é baseado na estrutura centralizada no *Oracle E-Business Suite*, permitindo de forma optimizada gerir os processos financeiros ligados ao controlo do património de uma empresa.

No projecto foi adoptada a metodologia de trabalho *scrum* que melhor se adequava ao projecto. Efectuou-se o estudo de todas as ferramentas necessárias para a execução do projecto (como *PL/SQL* e Conector *Oracle*), foi realizado um levantamento da lista de requisitos, foram desenvolvidas tabelas, funções e métodos na base de dados *Oracle*, foi

construída uma componente generalizada de classificação de campos flexíveis que se adapta facilmente a uma outra aplicação *OutSystems* e criados os respectivos ecrãs. Foram também criados ecrãs de listagem dos activos, de consulta, edição e de adição simples de um activo. Foram ainda implementados na aplicação *Web* a função de pesquisa, diversos filtros de pesquisa, classificação de campos flexíveis, consulta de um conjunto de especificações de um determinado activo e a adição de forma simples de activo no sistema.

4.2 Análise de requisitos

Inicialmente todo o processo de adição, edição, procura e abate de activos era realizado através de diferentes funcionalidades fornecidas pelos formulários do módulo *FA* do *OEBS*. Este tipo de processo era demasiado árduo para um conjunto de utilizadores sem formação em *OEBS*, pois para executar este tipo de tarefas é necessário um indivíduo com um conhecimento aprofundado no módulo *FA*.

O tempo consumido para a formação dos elementos da empresa para este tipo de acções é elevado porque as funcionalidades apresentadas pelas interfaces do *ERP* são complexas e apresentam demasiada informação (que por vezes é desnecessária). Para apenas realizar uma única tarefa é necessário passar por vários formulários, preencher campos desnecessários e validar diversas informações até realizar a tarefa desejada, o que aumenta significativamente o tempo consumido.

Para diminuir o desperdício de tempo, uma das opções seria contratar pessoas com formação no módulo *FA* do *OEBS* para apenas realizar essas tarefas. Esta opção teria um custo extremamente elevado devido à necessidade de ter pessoas em cada uma das instalações espalhadas pelos vários pontos do país.

Para solucionar estes problemas foi concebida uma aplicação em *OutSystems* que permite adicionar, editar, pesquisar e abater os activos. A realização destas tarefas utilizando esta aplicação é feita de uma forma autónoma, rápida e simples, libertando assim tempo na sua execução, e consequentemente, dos recursos humanos da empresa. A aplicação mantém também o registo de todas as operações realizadas em *OEBS*, permitindo assim um maior controlo sobre as acções realizadas.

Os requisitos funcionais (*user stories*) foram recolhidos seguindo a metodologia *scrum*. Uma *user story* [12] é uma descrição de alto nível de um requisito funcional, sendo a sua recolha uma forma rápida e simples de reunir os requisitos do cliente, sem elaborar uma vasta documentação formal. Cada *user story* é constituída por 3 componentes: *Card*, *Conversation* e *Confirmation*:

- ***Card*** – A secção de cabeçalho que deve incluir o nome/descrição da história do utilizador e um identificador.
- ***Conversation*** – A secção onde deve ser indicado o que a aplicação deve fazer, algo que ajude a explicar sucintamente o requisito. Esta descrição é feita do ponto de vista do utilizador.
- ***Confirmation*** – A secção onde são declarados os teste de validação do requisito. A descrição dos testes na fase inicial ajuda a identificar fluxos de utilização que os utilizadores, programadores e analistas apenas podiam detectatar numa fase mais tardia do processo.

Com base nos requisitos apresentados pelo cliente foi elaborada a Tabela 3 que descreve as *user stories* do projecto:

<i>Card</i>	<i>Conversation</i>	<i>Confirmation</i>
1. Listagem	Como utilizador, preciso de visualizar a listagem de todos os activos, para ter a noção da totalidade de activos existentes.	A aplicação tem um <i>link</i> que permite visualizar todos os activos.
2. Pesquisa	Como utilizador, preciso de pesquisar por uma determinada característica de um activo, bem como filtrar um conjunto de activos por uma determinada característica, para ter um acesso às especificações de um activo de uma forma rápida.	A aplicação tem um campo de pesquisa.

<i>Card</i>	<i>Conversation</i>	<i>Confirmation</i>
3. Consulta	Como utilizador, preciso de ver informações mais detalhadas de um activo.	A aplicação tem diferentes separadores com diversas informações sobre um activo.
4. Edição	Como utilizador, preciso de editar certas propriedades dos activos, para alterar especificações de um determinado activo.	A aplicação permite alterar os dados de um activo.
5. Adição Simples	Como utilizador, preciso apenas preencher os dados obrigatórios, para adicionar um activo de uma forma rápida.	A aplicação tem um <i>link</i> adição rápida de um activo.
6. Adição Detalhada	Como utilizador, preciso especificar todas as características disponíveis de um activo, para adicionar um activo de uma forma detalhada.	A aplicação tem um <i>link</i> adição detalhada de um activo.
7. Adição em Massa	Como utilizador, preciso de preencher conjunto de características em comum, para realizar uma adição em bloco.	A aplicação tem um <i>link</i> de adição em bloco.
8. Classificação	Como utilizador, preciso classificar um conjunto de activos, para eles serem lançados e cuja classificação patrimonial se encontra finalizada.	A aplicação permite a classificação de um conjunto de activos.
9. Abate	Como utilizador, preciso de abater um activo, de modo a retirar da lista de activos da empresa.	A aplicação permite abater um determinado activo.
10. Campos Flexíveis	Como utilizador, preciso de escolher os segmentos de um determinado campo flexível, para permitir uma combinação válida.	A aplicação suporta um método que permite a escolha de combinações válidas num formato simples.

Tabela 3 - Requisitos funcionais

4.3 Arquitectura do sistema

A Ilustração 10 representa a arquitectura global do sistema desenvolvido. Neste diagrama podemos observar três componentes distintos, que representam as diferentes camadas responsáveis pela interacção da aplicação *OutSystems* com *Oracle E-Business Suite*.

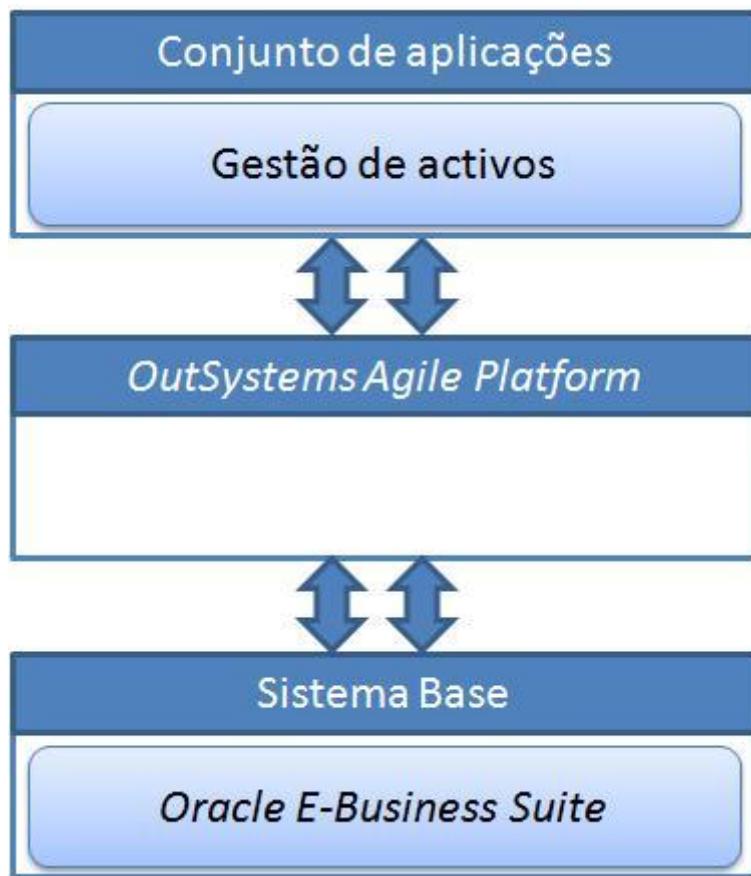


Ilustração 10 - Arquitectura do sistema

De seguida são descritas com maior detalhe as diferentes camadas da arquitectura envolvidas no desenvolvimento da aplicação de gestão de activos.

- **Sistema Base** – A primeira camada da arquitectura do sistema contém o sistema nuclear que interage com a instância da *OutSystems*. Neste projecto o sistema base é *Oracle E-Business Suite* onde se encontra a lógica de negócio. Esta lógica de negócio é constituída por tabelas do *OEBS*, pela *API* composta por uma serie de funções e procedimentos em *PL/SQL* que permite interagir com as tabelas.

- ***OutSystems Agile Platform*** – A segunda camada da arquitectura do sistema é a plataforma da *OutSystems*. Esta, através do conjunto de ferramentas que disponibiliza, permite integrar, desenvolver e gerir aplicações *Web*.
- **Conjunto de Aplicações** – A terceira camada da arquitectura do sistema é composta por aplicações periféricas desenvolvidas através da plataforma e disponibilizadas aos utilizadores finais da aplicação. Esta camada contém a aplicação de gestão de activos desenvolvida no âmbito do projecto.

4.4 Carregamento de dados

No âmbito do projecto da gestão de património, foi feita uma migração dos activos que se encontravam em documentos do tipo *Excel* para o módulo *FA* do *OEBS*. Pretende-se com esta secção descrever os principais processos implementados e as tarefas necessárias para concluir a transição dos dados.

4.4.1 Tratamento de dados em *Excel*

Os ficheiros *Excel* com a listagem de activos continham linhas incompletas, colunas mal definidas, artigos repetidos e campos vazios. Para corrigir estas falhas foi necessário realizar um trabalho conjunto com o cliente de forma a obter uma listagem coerente sobre a qual pudessem ser exercidas configurações extras de forma a adapta-las com facilidade às tabelas do módulo *FA* do *OEBS*.

Um dos processos mais importantes foi a atribuição de siglas a uma das características mais importantes de um activo, a localização. Em *OEBS* a localização é composta por quatro campos: o local, o sub-local, o edifício/piso e ala/sala/departamento. Foi necessário atribuir siglas a cada uma das opções desses campos; o somatório de cada um dos campos resulta numa localização com um identificador único, permitindo definir com rigor a localização de um determinado activo.

O processo de atribuição de siglas à chave do activo foi feito depois de uma consulta ao módulo *FA*, com o objectivo de compreender a composição desse campo. O campo chave do activo é composto por dois valores: o método de compra, que representa a forma

como o artigo foi comprado, e o estado, que indica as condições físicas do activo. O resultado final é um valor composto por estas duas siglas e representa a chave do activo.

Todos os dados relacionados com os activos foram transcritos para um único ficheiro *Excel*, com o formato desejado para a transição dos dados para a tabela de *buffer* da base de dados *Oracle*.

4.4.2 Migração de dados de *Excel* para o módulo FA da *Oracle*

Esta fase consistiu na transferência dos dados presentes no documento *Excel* para uma tabela *buffer* da base de dados *Oracle*. Foi realizada através da adição de uma coluna com uma instrução de “*INSERT INTO*” que permite inserir um registo numa tabela, neste caso a *CPRX_FA_MASSADD_TDP*. Foi realizado um *script* com a instrução anterior que permitiu popular a tabela do *Oracle*. A Tabela 4 enumera algumas colunas da tabela *CPRX_FA_MASSADD_TDP* com o respectivo tipo de dados e uma sucinta descrição.

Coluna	Tipo de valor	Descrição
ASSET_NUMBER	VARCHAR2 (15 Byte)	Número do activo
TAG_NUMBER	VARCHAR2 (15 Byte)	Número da etiqueta
DATE_PLACED_IN_SE RVICE	DATE	Data de início de utilização
BOOK_TYPE_CODE	VARCHAR2 (15 Byte)	Livro do activo
CAT_MOD_LEGAL	VARCHAR2 (15 Byte)	Modelo legal da categoria do activo
AKEY_METAQUI	VARCHAR2 (15 Byte)	Método da chave do activo
DESCRIPTION	VARCHAR2 (200 Byte)	Descrição do activo
FIXED_ASSETS_UNITS	NUMBER	Número de unidades

Coluna	Tipo de valor	Descrição
PAYABLES_COST	NUMBER	Custo do activo
FIXED_ASSETS_COST	NUMBER	Custo fixo do activo
UNREVALUED_COST	NUMBER	Valor da reavaliação
DEPRN_RESERVE	NUMBER	Valor depreciação
VIDA_UTIL	NUMBER	Vida útil do activo
INFO_ABATE	VARCHAR2 (1000 Byte)	Informação sobre abate do activo

Tabela 4 - Excerto da tabela CPRX_FA_MASSADD_TDP

Após a transferência de dados para a tabela *buffer* da *Oracle* foi executado um *procedure*¹⁴ (*pre_reqs_val* do *package* CPRX_FA_LOAD), que permite de uma forma rápida e segura preencher a tabela FA_MASS_ADDITIONS tendo como fonte a tabela intermédia CPRX_FA_MASSADD_TDP. O método de trabalho utilizado para fazer a transição de dados da tabela intermédia para a tabela do *OEBS* é um método muito frequente e recomendado pela *Oracle*.

O passo de transferência directa dos dados do documento *Excel* para a tabela FA_MASS_ADDITIONS, não foi considerado porque é um processo impraticável pois existe um conjunto de dados com um determinado nível de detalhe (como códigos de identificação do local e o tipo de livro) que só é possível definir no ambiente do *OEBS* e não é praticável num documento *Excel*.

Posteriormente à execução do *procedure*, foi executada a rotina *Mass Additions Posting*, que é um programa concorrente cujo objectivo é popular as restantes tabelas relacionadas com o modelo de dados do módulo FA do *OEBS*. Este é o método padrão para carregamento em massa de activos no *ERP* da *Oracle*.

¹⁴ *Procedure* - é um bloco em *PL/SQL* que pode ser compilado e armazenado numa base de dados como um objecto.

4.5 Conector *Oracle*

O Conector *Oracle* [13] é uma aplicação interna desenvolvida pela Truewind que permite a integração das funcionalidades da *Oracle* com a plataforma *OutSystems*, em particular os procedimentos e funções *PL/SQL* disponíveis no *OEBS* que realizam funcionalidades importantes e que respeitam todas as regras de negócio.

Um ponto fraco da plataforma *OutSystems* é a incapacidade de converter procedimentos e funções *PL/SQL* do *OEBS* em métodos de *OutSystems*. Esta apenas tem implementada a integração com os procedimentos do *ERP* da *SAP*, e sempre que se pretende invocar funções do *Oracle* é necessário desenvolver interfaces próprias para o efeito. O desenvolvimento destas interfaces para invocação das funções e procedimentos é um trabalho moroso e susceptível à inserção de erros, consumindo uma quantidade de tempo considerável aos programadores e não acrescentando valor às soluções desenvolvidas.

O Conector *Oracle* permitiu ultrapassar essa lacuna através da geração dos pontos de conexão para as funções e procedimentos da base de dados *Oracle*. Cada interface de ligação consiste numa chamada à base de dados relativa a um procedimento ou função *PL/SQL* com os seus respectivos parâmetros de entrada, saída e entrada/saída. Estes procedimentos e funções possibilitam a utilização da lógica de negócio do *OEBS* directamente na plataforma de desenvolvimento *OutSystems*, permitindo aos programadores, que não são especialistas no *OEBS*, adicionar de forma ágil novas funcionalidades.

A Ilustração 11 apresenta o fluxo que ilustra o processo de geração das bibliotecas para estarem disponíveis na aplicação *OutSystems*.

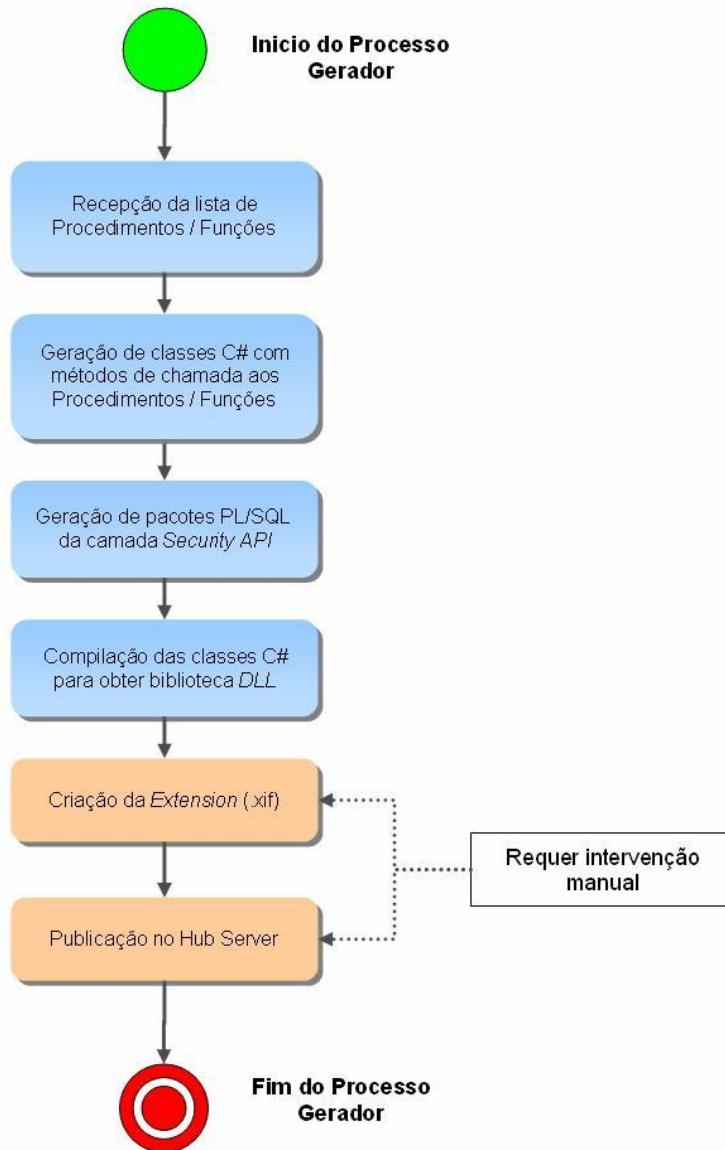


Ilustração 11 - Fluxo de criação do Conector Oracle

4.6 Funcionalidade implementadas

Esta secção relata o trabalho desenvolvido no projecto até ao momento e nela se descrevem as funcionalidades implementadas.

4.6.1 Listagem de activos

A listagem de activos corresponde ao ecrã principal da aplicação, sendo este o seu ponto de entrada. Neste ecrã encontra-se a listagem do património da empresa. Aqui podem-se

visualizar todos os activos da empresa incluindo as suas características mais evidentes, de forma a reconhecer o activo com uma maior facilidade.

O ecrã de listagem de activos tem as seguintes funcionalidades:

- Filtro de pesquisa de activos
- *Links* de navegação para percorrer toda a listagem de activos
- *Link* de acesso aos detalhes do activo
- Número total de activos da empresa
- *Link* de acesso à criação de um activo de forma rápida
- *Link* de acesso à criação de um activo de forma detalhada
- *Link* de acesso à adição em massa de activos
- Possibilidade de exportar a lista de activos para um documento em *Excel*
- *Link* de acesso à ajuda do utilizador

The screenshot shows a list of assets in a table format. A modal dialog box titled "Transferência de ficheiros" (File Transfer) is open, asking if the user wants to "Prende abrir ou guardar este ficheiro?" (Open or Save this file?). The file name is "ListaDeActivos.xls", the type is "Microsoft Office Excel 97-2003 Worksheet", and the location is "outsystems-d.turismodeportugal.pt". Buttons for "Abrir" (Open), "Guardar" (Save), and "Cancelar" (Cancel) are visible. The main table lists assets like "ARVÁRIO - MÓVEIS", "CAIXA INDIVIDUAL - METAL PRETO", and "COLCHÃO INDIVIDUAL". The bottom right corner of the table shows navigation links: "previous" (with page numbers 1, 2, 3, 4, 5), "next", and "1 to 30 of 609872 records".

Ilustração 12 - Listagem de activos

Na Ilustração 12 podemos observar o ecrã de listagem de activos, bem como o acesso às funcionalidades descritas anteriormente. Como plano de frente pode-se visualizar a opção que permite exportar uma lista de activos para um documento em *Excel*. Um exemplo da listagem é demonstrado na Ilustração 13. Também é possível observar no canto inferior direito o menu de navegação pelas restantes páginas da listagem e no topo

da tabela do lado direito o número total de activos da empresa. As restantes funcionalidades são facilmente visíveis para um utilizador comum.

	A	B	C	D	E	F	G	H
1	Nº Activo	Nº Serie	Descrição	Categoria	Tipo	Livro	Inicio Utilização	Nº Factura
2	024970		ARMÁRIO - S/POR	CIBE 103.01 01	CAPITALIZED	TDP	2006-9-29 00:00	FAC20060356
3	024971		ARMÁRIO - S/POR	CIBE 103.01 01	CAPITALIZED	TDP	2006-9-29 00:00	FAC20060356
4	020891		CADEIRA FIXA - MADEIRA BEGE	CIBE 107.01 07	CAPITALIZED	TDP	2009-1-1 00:00	ND
5	020897		CAMA INDIVIDUAL - METAL PRETO	CIBE 107.01 08	CAPITALIZED	TDP	2006-9-29 00:00	FAC20060356
6	041081		COLCHÃO INDIVIDUAL	CIBE 107.01 10	CAPITALIZED	TDP	2006-9-30 00:00	FAC20060356
7	020895		MESA APOIO - MADEIRA BEGE 50X40X50	CIBE 107.01 20	CAPITALIZED	TDP	2006-9-29 00:00	FAC20060356
8	020892		CADEIRA FIXA - MADEIRA BEGE	CIBE 107.01 07	CAPITALIZED	TDP	2009-1-1 00:00	ND
9	020915		MESA APOIO - MADEIRA BEGE 50X40X50	CIBE 107.01 20	CAPITALIZED	TDP	2006-9-29 00:00	FAC20060356
10	020917		CAMA INDIVIDUAL - METAL PRETO	CIBE 107.01 08	CAPITALIZED	TDP	2006-9-29 00:00	FAC20060356
11	020926		MESA APOIO - MADEIRA BEGE 50X40X50	CIBE 107.01 20	CAPITALIZED	TDP	2006-9-29 00:00	FAC20060356
12	041088		COLCHÃO INDIVIDUAL	CIBE 107.01 10	CAPITALIZED	TDP	2006-9-30 00:00	FAC20060356
13	020932		CADEIRA FIXA - MADEIRA BEGE	CIBE 107.01 07	CAPITALIZED	TDP	2009-1-1 00:00	ND
14	020947		CAMA INDIVIDUAL - METAL PRETO	CIBE 107.01 08	CAPITALIZED	TDP	2006-9-29 00:00	FAC20060356
15	041102		COLCHÃO INDIVIDUAL	CIBE 107.01 10	CAPITALIZED	TDP	2006-9-30 00:00	FAC20060356
16	021245		CADEIRA FIXA - MADEIRA BEGE	CIBE 107.01 07	CAPITALIZED	TDP	2009-1-1 00:00	ND

Ilustração 13 - Exemplo da lista de activos do ficheiro Excel

A Ilustração 14 mostra o código *OutSystems* desenvolvido para alimentar o ecrã de listagem de activos. Podemos verificar que no enquadramento 1 refresca o ecrã sempre que as funções de pesquisa e de navegação são usadas, a lista é populada pela query assinalada no enquadramento 2 da imagem, e sempre que ocorre algo inesperado é lançada uma excepção juntamente com uma mensagem de erro para o ecrã enquadramento 3.

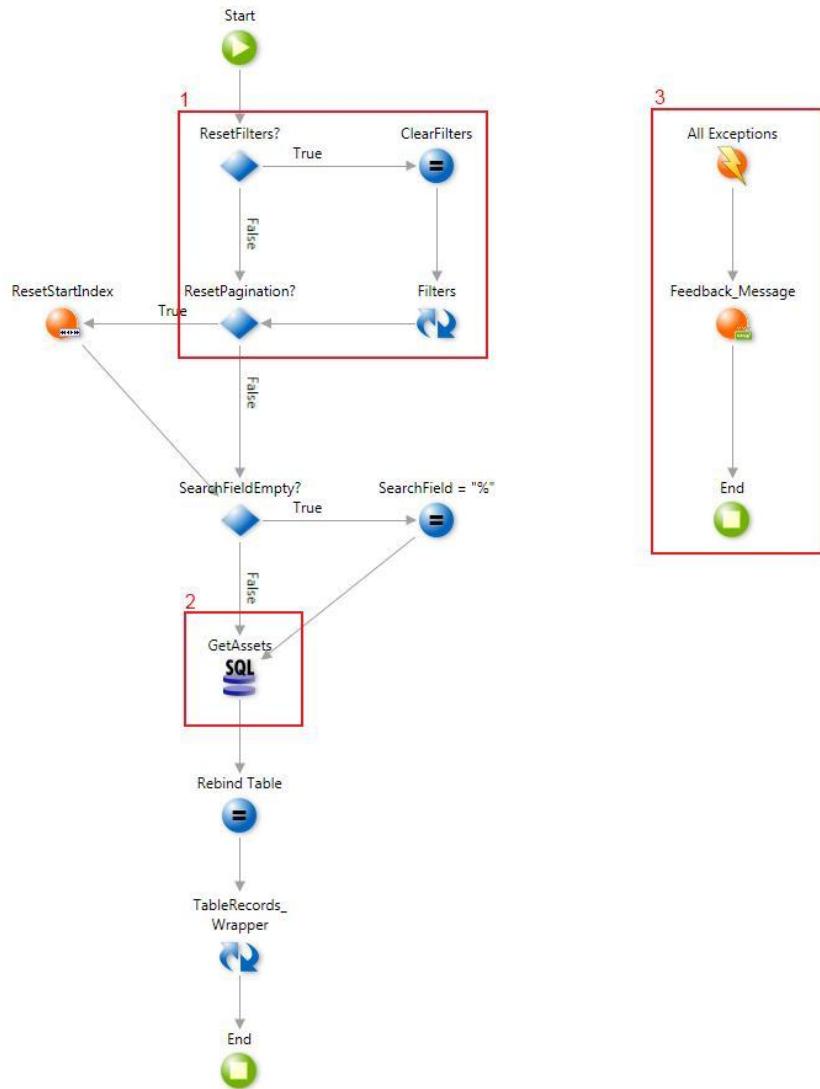


Ilustração 14 - Fluxo do processo do ecrã listagem de activos

Um dos pontos fulcrais deste fluxo é a *query* que permite alimentar a lista de activos apresentada ao utilizador. Na Ilustração 15 é demonstrada a *query* de uma forma mais detalhada, onde podemos observar que existem onze parâmetros de entrada: a “*SearchKeyword*”, o valor dado pelo utilizador no campo de procura, os quatro segmentos que compõem o campo flexível localização (“*Local*”, “*SubLocal*”, “*BuildingFloor*” e “*Department*”), todos eles disponíveis nas respectivas lista de valores disponíveis no campo de pesquisa, os 3 segmentos que compõem o campo flexível categoria do activo (“*LegalModel*”, “*ClassType*” e “*AssetImob*”), e dois parâmetros (“*BeginDate*” e “*EndDate*”) para definir um intervalo de tempo para o início de utilização do activo e uma variável de verificação a “*NullText*”. Tem como parâmetro de saída uma estrutura personalizada que tem o nome de “*Documents*” que guarda o

resultado devolvido pela *query*. A *query* é composta por um conjunto de junções, a *view* FA_ADDITIONS_V com a tabela FA_BOOKS, com a *view* FA_INVOICE_DETAILS_V, com a tabela FA_DISTRIBUTION_HISTORY através da coluna “*asset_id*” que está presente em todas as tabelas mencionadas anteriormente, e por último a junção da tabela FA_DISTRIBUTION_HISTORY com a *view* FA_LOCALIZACOES_V através da coluna “*location_id*”. Existe também um conjunto de restrições a nível da palavra de procura e do valor escolhido através da lista de valores apresentada ao utilizador que permite assim retornar os dados desejados de cada uma das tabelas.

GetAssetsList

Parameters

- SearchKeyword
- Local
- SubLocal
- BuildingFloor
- Department
- LegalModel
- ClassType
- AssetImob
- BeginDate
- EndDate
- NullText

No object selected

SQL **Test Inputs** **Test Output**

```

SELECT DISTINCT
FAAv.ASSET_NUMBER,
FAAv.SERIAL_NUMBER,
FAAv.DESCRIPTION,
FAAv.CATEGORY,
FAAv.ASSET_TYPE,
FAB.BOOK_TYPE_CODE,
FAB.DATE_PLACED_IN_SERVICE,
FAIDv.INVOICE_NUMBER,
FAIDv.VENDOR_NAME,
FAIDv.VENDOR_NUMBER,
FADH.LOCATION_CONCATENATED,
FAAv.MANUFACTURER_NAME,
FAAv.MODEL_NUMBER,
FAAv.CURRENT_UNITS,
FAAv.ASSET_KEY,
FADH.ACCTING_CONCATENATED

FROM {FA_ADDITIONS_WEB_V} FAAv
JOIN {FA_BOOKS} FAB ON (FAAv.asset_id = FAB.asset_id)
JOIN {FA_INVOICE_DETAILS_V} FAIDv ON (FAAv.asset_id = FAIDv.asset_id)
JOIN {FA_DISTRIBUTION_HIST_WEB_V} FADH ON (FAAv.asset_id = FADH.asset_id)
JOIN {FA_LOCATIONS} FAL ON (FAL.LOCATION_ID = FADH.LOCATION_ID)
JOIN {FA_CATEGORIES_B} FAC ON (FAC.CATEGORY_ID = FAAv.ASSET_CATEGORY_ID)

WHERE (
    1 = 1
    AND (FAB.DATE_INEFFECTIVE is null)
    AND (FAB.TRANSACTION_HEADER_ID_OUT is null)
    AND (FAB.DATE_PLACED_IN_SERVICE >= @BeginDate AND FAB.DATE_PLACED_IN_SERVICE <=@EndDate)
    AND (@Local=@NullText or FAL.SEGMENT1 =@Local)
    AND (@SubLocal=@NullText or FAL.SEGMENT2 =@SubLocal)
    AND (@BuildingFloor=@NullText or FAL.SEGMENT3 =@BuildingFloor)
    AND (@Department=@NullText or FAL.SEGMENT4 =@Department)
    AND (@LegalModel=@NullText or FAC.SEGMENT1 =@LegalModel)
    AND (@ClassType=@NullText or FAC.SEGMENT2 =@ClassType)
    AND (@AssetImob=@NullText or FAC.SEGMENT3 =@AssetImob)
    AND ((LOWER (FAAv.description) LIKE ('%' || LOWER (@SearchKeyword) || '%'))
        OR (LOWER (FAAv.serial_number) LIKE ('%' || LOWER (@SearchKeyword) || '%'))
        OR (LOWER (FAAv.asset_number) LIKE ('%' || LOWER (@SearchKeyword) || '%'))
        OR (LOWER (FAIDv.vendor_name) LIKE ('%' || LOWER (@SearchKeyword) || '%'))
        OR @SearchKeyword = ''))

)

```

Verify **Test**

Ilustração 15 - Query da lista de activos

4.6.2 Pesquisa de activos

Esta aplicação desenvolvida em *OutSystems*, oferece uma interface intuitiva e de fácil utilização, que permite efectuar uma pesquisa ou filtrar um conjunto de activos através de um conjunto de características definidas pelo cliente, tais como a descrição, número de série, etiqueta, activo, nome e número do fornecedor. Foi decidido durante a implementação desta funcionalidade, que o campo de pesquisa mantém o valor até o utilizador clicar no botão "Limpar". De forma a obter este comportamento, foram utilizadas variáveis de sessão para guardar os valores do filtro.

Lista de Activos										
<input style="width: 20px; height: 20px; border: 1px solid red; vertical-align: middle; margin-right: 10px;" type="text" value="1"/> <input style="width: 200px; border: 1px solid #ccc; vertical-align: middle; margin-right: 10px;" type="text" value="Descrição do activo, Nº de série, Nº de actv."/> <input style="border: 1px solid #ccc; padding: 2px 10px; vertical-align: middle; margin-right: 10px;" type="button" value="Pesquisar"/> <input style="border: 1px solid #ccc; padding: 2px 10px; vertical-align: middle;" type="button" value="Limpar"/>										
Localização:	<input type="text" value="LISBOA"/>	Sublocais:	<input type="text" value="Av. Ant. Aug. Aguiar, 23 - 3º"/>	Edifício:	<input type="text" value="PISO 3"/>	TODOS	NAO ESPECIFICADO	Bem:	PISO 3	
Categoría:	<input type="text" value="Modelo Legal: TODOS"/>	Classe / Tipo:	<input type="text" value="TODOS"/>							
Intervalo de inicio de utilização:	<input type="text" value="Data de inicio: YYYY-MM-DD"/>	Data de fim:	<input type="text" value="YYYY-MM-DD"/>							
Adição Simples Adição Detalhada Adições em Bloco Exportar para Excel										
Nº Activo	Nº Série	Descrição	Categoría	Tipo Activo	Livro	Inicio Utilização	Nº Factura	Nome Fornecedor	Nº Fornecedor	Loca
024970	ARMÁRIO - S/PORTAS	CIBE 103.01 01	CAPITALIZED	TDP	2006-09-29	FAC20060356				LAMGI/RVAJPIR
024971	ARMÁRIO - S/PORTAS	CIBE 103.01 01	CAPITALIZED	TDP	2006-09-29	FAC20060356				LAMGI/RVAJPIR
020891	CADEIRA FIXA - MADEIRA BEGE	CIBE 107.01 07	CAPITALIZED	TDP	2009-01-01	ND				LAMGI/RVAJPIR
020897	CAMA INDIVIDUAL - METAL PRETO	CIBE 107.01 08	CAPITALIZED	TDP	2006-09-29	FAC20060356				LAMGI/RVAJPIR
041081	COLCHÃO INDIVIDUAL	CIBE 107.01 10	CAPITALIZED	TDP	2006-09-30	FAC20060356				LAMGI/RVAJPIR
020895	MESA APOIO - MADEIRA BEGE 50X40X50	CIBE 107.01 20	CAPITALIZED	TDP	2006-09-29	FAC20060356				LAMGI/RVAJPIR
020892	CADEIRA FIXA - MADEIRA BEGE	CIBE 107.01 07	CAPITALIZED	TDP	2009-01-01	ND				LAMGI/RVAJPIR
020915	MESA APOIO - MADEIRA BEGE 50X40X50	CIBE 107.01 20	CAPITALIZED	TDP	2006-09-29	FAC20060356				LAMGI/RVAJPIR
020917	CAMA INDIVIDUAL - METAL PRETO	CIBE 107.01 08	CAPITALIZED	TDP	2006-09-29	FAC20060356				LAMGI/RVAJPIR
020926	MESA APOIO - MADEIRA BEGE 50X40X50	CIBE 107.01 20	CAPITALIZED	TDP	2006-09-29	FAC20060356				LAMGI/RVAJPIR
041088	COLCHÃO INDIVIDUAL	CIBE 107.01 10	CAPITALIZED	TDP	2006-09-30	FAC20060356				LAMGI/RVAJPIR
020932	CADEIRA FIXA - MADEIRA BEGE	CIBE 107.01 07	CAPITALIZED	TDP	2009-01-01	ND				LAMGI/RVAJPIR
020947	CAMA INDIVIDUAL - METAL PRETO	CIBE 107.01 08	CAPITALIZED	TDP	2006-09-29	FAC20060356				LAMGI/RVAJPIR
041102	COLCHÃO INDIVIDUAL	CIBE 107.01 10	CAPITALIZED	TDP	2006-09-30	FAC20060356				LAMGI/RVAJPIR
021245	CADEIRA FIXA - MADEIRA BEGE	CIBE 107.01 07	CAPITALIZED	TDP	2009-01-01	ND				LAMGI/RVAJPIR

Ilustração 16 - Detalhe do filtro

Como vem descrito na Ilustração 16, a pesquisa, além de permitir a procura de dados pelo parâmetro colocado no campo de procura ponto 1 da imagem, ainda permite filtrar os dados através da lista de valores que constitui a localização e a categoria do activo e também é possível definir o intervalo de início de utilização do activo. Os filtros de pesquisa encontram-se assinalados na imagem com ponto 2 que estão embutidos no campo extensível na opção de procura da aplicação.

A Ilustração 17 demonstra um exemplo de pesquisa realizada com a palavra “cadeira”. O valor de pesquisa é *case insensitive*, ou seja, as letras maiúsculas e minúsculas são interpretadas de igual forma. No topo da tabela pode-se ver o número total de registos encontrados na região de Lisboa, parâmetro escolhido a partir da lista de valores disponível no filtro e cujo activo contém na descrição a palavra “cadeira”.

Lista de Activos															
Filtrar resultados															
Detalhes / Edifícios / Edifícios em Bloco / Inventário / Excluir															
Nº Activo	Nº Serie	Descrição	Categoria	Tipo Activo	Livro	Válida Utilização	Nº Factura	Nome Fornecedor	Nº Fornecedor	Localização	Marca	Modelo	Unidades	Chave	Conta Custos
B15048		CADERIRA FIXA - INÁPA PRETO	CIBE103_01095	Capitalizado	TDP	1984-12-30	ND	LISVAD/PI01/SALAB/HF						02/00	01/02/05/01/02/0000/0623/00
B15232		CADERIRA FIXA - TECIDO CASTANHO	CIBE103_01095	Capitalizado	TDP	1990-08-02	ND	LISVAD/PI01/SALATOR						02/00	01/02/05/01/02/0000/0623/00
B15042		CADERIRA FIXA - MADEIRA CASTANHO	CIBE103_01095	Capitalizado	TDP	1979-03-30	ND	LISVAD/PI01/SALAB						02/15	01/02/05/01/02/0000/0623/00
B15045		CADERIRA FIXA - MADEIRA CASTANHO	CIBE103_01095	Capitalizado	TDP	1990-08-02	ND	LISVAD/PI01/SALAB						02/00	01/02/05/01/02/0000/0623/00
B14784		CADERIRA FIXA	CIBE103_01095	Capitalizado	TDP	2004-12-16		DATAMÓVEL - SISTEMAS DE ESCRITÓRIO, LDA	036115	LISVAAE/PI01/SESCD				02/00	01/02/45/01/02/0000/0623/00
B14800		CADERIRA FIXA - FÓRMICA CASTANHO	CIBE107_01097	Capitalizado	TDP	2001-12-31		PAPELARIA E LIVRARIA SAIGES	037993	LISVAAE/PI01/SESCD				02/00	01/02/05/01/02/0000/0623/00
B14801		CADERIRA FIXA - FÓRMICA CASTANHO	CIBE107_01097	Capitalizado	TDP	2000-08-22	ND	LISVAAE/PI01/SESCD						02/00	01/02/45/01/02/0000/0623/00
B14878		CADERIRA RODADA - PVC AZUL COSTAS ALTAS C/BRACOS	CIBE103_01095	Capitalizado	TDP	2000-12-13	ND	LISVAAE/PI01/SESCD						02/00	01/02/45/01/02/0000/0623/00
B15156		CADERIRA FIXA - MADEIRA ROSA C/BRACOS	CIBE107_01097	Capitalizado	TDP	2004-12-16		DATAMÓVEL - SISTEMAS DE ESCRITÓRIO, LDA	036115	LISVAAE/PI01/REEST				02/00	01/02/45/01/02/0000/0623/00
B15157		CADERIRA FIXA - MADEIRA ROSA C/BRACOS	CIBE107_01097	Capitalizado	TDP	2004-12-16		DATAMÓVEL - SISTEMAS DE ESCRITÓRIO, LDA	036110	LISVAAE/PI01/REEST				02/00	01/02/45/01/02/0000/0623/00
B15128		CADERIRA FIXA - MADEIRA ROSA C/BRACOS	CIBE107_01097	Capitalizado	TDP	2004-12-16		DATAMÓVEL - SISTEMAS DE ESCRITÓRIO, LDA	036110	LISVAAE/PI01/REEST				02/00	01/02/45/01/02/0000/0623/00
B15158		CADERIRA FIXA - MADEIRA ROSA C/BRACOS	CIBE107_01097	Capitalizado	TDP	2004-12-16		DATAMÓVEL - SISTEMAS DE ESCRITÓRIO, LDA	036110	LISVAAE/PI01/REEST				02/00	01/02/45/01/02/0000/0623/00
B15161		CADERIRA FIXA - MADEIRA ROSA C/BRACOS	CIBE107_01097	Capitalizado	TDP	2004-12-16		DATAMÓVEL - SISTEMAS DE ESCRITÓRIO, LDA	036110	LISVAAE/PI01/REEST				02/00	01/02/45/01/02/0000/0623/00
B15079		CADERIRA FIXA - MADEIRA ROSA C/BRACOS	CIBE107_01097	Capitalizado	TDP	2004-12-16		DATAMÓVEL - SISTEMAS DE ESCRITÓRIO, LDA	036115	LISVAAE/PI01/ROCA				02/00	01/02/45/01/02/0000/0623/00

Ilustração 17 - Exemplo de pesquisa

4.6.3 Consulta de activos

O acesso aos detalhes do activo é realizado através do *link* existente no número do activo. Estes detalhes estão disponíveis num novo ecrã, que se encontra dividido em seis separadores. Cada um dos separadores da Ilustração 18 fornece dados sobre um determinado tipo de informação do activo:

- **Info linha** – Este separador apresenta a informação de carácter geral do activo. Contém essencialmente a mesma informação que se pode ver na listagem de activos.
- **Livro** – Este separador apresenta a informação financeira do activo, como por exemplo o nome do livro de depreciação e o método de amortização.
- **Atribuições** – Este separador apresenta a informação relativa à atribuição do activo, como por exemplo a localização do activo, conta de custos e a quem pertence.
- **Informação patrimonial** – Este separador apresenta a informação sobre identificação do património do activo, como por exemplo a descrição do activo e a categoria ao qual pertence.
- **Linha de origem** – Este separador tem a informação relativa à fonte de origem do activo, como por exemplo o número da factura e o nome do fornecedor.
- **Consulta financeira** – Este separador tem a informação sobre valor monetário do activo, como por exemplo o valor de aquisição do activo e o valor líquido do activo.

Lista de Activos

Info Linha	Livros	Atribuições	Informação Patrimonial	Linha De Origem	Consulta Financeira
N Activo	011649		N Referência		
Livro	TDP		Comentários		
<u>Informações Financeiras</u>					
Valor Aquisição Actual	571,20		Limite Reavaliação		
Valor Aquisição Original	571,20		Reserva Reavaliação	0	
Tipo Valor Sucata			Método	STL	
Valor Residual	0		Vida Útil Anos	4	
Valores Reposição	571,20		Vida Útil Meses	0	
Valor Líquido Activo	0		Inicio Utilização	2004-04-26 00:00:00	
Amortização Acumulada	571,20		Conv. Acertos Datas	FINAL-MÊS	
Percentagem Valor Sucata					
Amortizacao AE	0				

Ilustração 18 - Consulta do activo

Este ecrã permite consultar todas as informações relativas a um activo e também se encontra preparado para funcionar em modo de edição. Todas as características do activo, à excepção do seu número, podem ser alteradas pelo utilizador de negócio. O número do activo não é editável, pois este é automaticamente atribuído pelo módulo *FA* na fase inicial do processo de adição do activo ao património da empresa.

4.6.4 Campos flexíveis

Entre as funcionalidades mais importantes do *Oracle E-Business Suite* encontra-se a classificação os campos flexíveis [14]. Estes são utilizados como meio de classificação e podem ser encontrados em vários módulos da *Oracle*.

Um campo flexível em *Oracle E-Business Suite* é constituído por segmentos. Existem dois tipos de campos flexíveis (Ilustração 19): “campo flexível chave” e “campo flexível descrição”. O “campo flexível chave” aparece no formulário do *OEBS* como campo de texto normal com um aviso adequado. As empresas criam siglas ou códigos que representam segmentos para identificar contas da Contabilidade e outras entidades comerciais. Cada segmento pode representar uma característica da entidade. O “campo flexível descrição” aparece como um campo de texto com dois caracteres (parêntesis recto “[“]) a assinalar o tipo de campo flexível. Este proporciona um espaço

personalizável nos formulários do *OEBS* e permite anexar informação adicional. O “campo flexível descrição” pode ser sensível ao contexto, armazenando informação contextual do utilizador em diversas partes do formulário. Ao abrir os dois tipos de campo flexível aparecem com uma janela *pop-up* que contém um campo separado para cada segmento. Cada segmento tem um nome e um conjunto de valores válidos e estes podem também ter uma descrição do valor.

A função de classificação do módulo *FA* do *OEBS* permite atribuir vários tipos de segmentos em diversas funções de negócio desde classificação do local, da conta de custos, da chave do activo, categoria do activo no módulo *FA* e têm outro tipo de classificação em outros módulos.

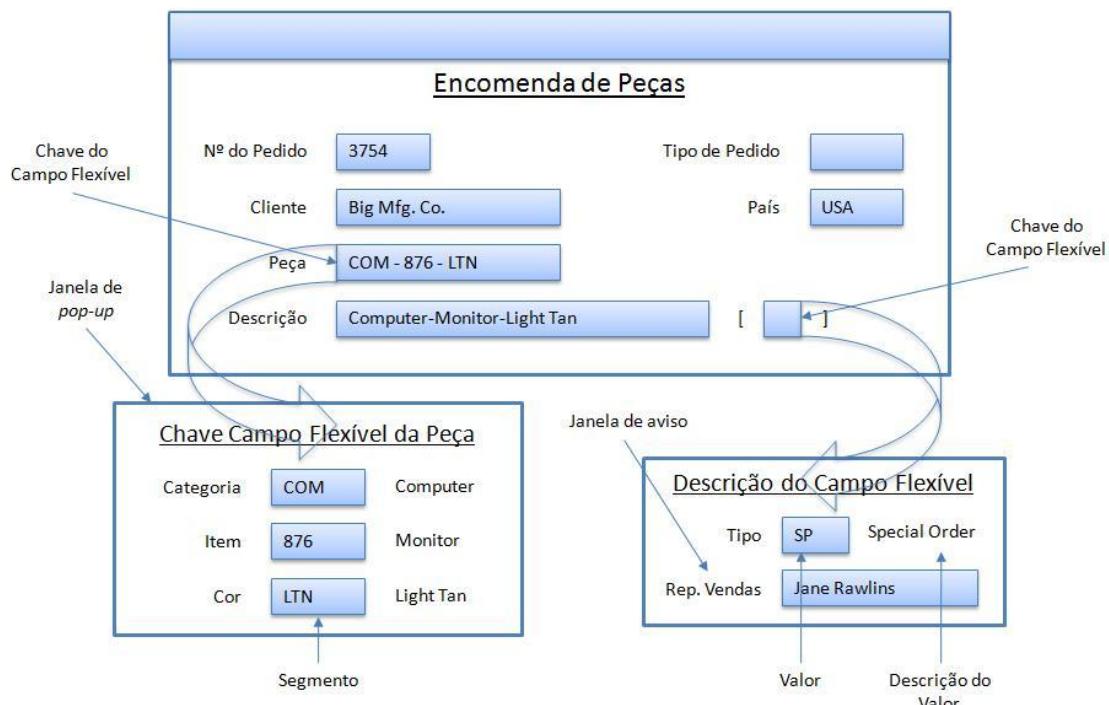


Ilustração 19 - Conceitos básicos do campo flexível

Um dos requisitos principais deste projecto era universalizar esta função nuclear do *OEBS* através de uma aplicação em *OutSystems*, de modo a ser reutilizado em futuras aplicações da empresa, pois existe uma forte aposta na ligação da plataforma *OutSystems* com o *OEBS*, e este componente de classificação vai permitir aos programadores da plataforma *OutSystems* adicionar às suas aplicações a classificação nos segmentos de uma forma simples e transparente.

Esta aplicação de classificação foi concebida com dois ecrãs: um onde podemos visualizar o número e o tipo de campos necessários para a classificação de um determinado valor que pretendemos atribuir e no outro a descrição e sigla correspondente de cada um dos segmentos válidos.

A Tabela 5 representa a descrição formal da classificação de um dos valores do módulo *FA* a localização, que é composta por um conjunto de quatro segmentos descrito na tabela abaixo.

Componentes da classificação de uma localização	Número de segmentos e a posição na classificação	Valores válidos de um segmento
Local	SEGMENT1	Representa a cidade onde o activo se encontra
Sublocal	SEGMENT2	Representa a instituição no local onde se encontra o activo
EdificioPiso	SEGMENT3	Representa o piso do edifício onde o activo se encontra considerando o local e sub-local
AlaSalaDep	SEGMENT4	Representa a sala onde o activo se encontra considerando o local, sub-local e edifício/piso

Tabela 5 - Descrição de classificação da localização

A componente dos campos flexíveis foi desenvolvida da forma mais generalizada possível, para que esta componente possa ser utilizada em futuras aplicações desenvolvidas em *OutSystems*. Foi verificado que existe sempre a necessidade de atribuir valores por omissão para cada tipo de campo (localização, categoria, conta de

custos e chave do activo), valores que podem ser alterados consoante o tipo de campo de uma forma simples e prática.

O ecrã editar é constituído por vários separadores, cada um com um tipo específico de informação relativa ao activo. No separador “Atribuições” está implementado o bloco de campos flexíveis em dois valores: a Conta de Custos e Localização, como se pode observar na Ilustração 20.

[Lista de Activos](#)

Info Linha	Livros	Atribuições	Informação Patrimonial	Linha De Origem	Consulta Financeira
N Activo	013366		Alteração Unidade		
Livro	TDP		Unidades	1	
Comentários			Empregado Nome		
Total Unidades	1		Empregado Número		
N Referência	Número de Referência		Conta Custos	01 02 310 020000 6626 000 510	editar
DataTransferência	2007-08-01 00:00:00		Localização	LISV RISARC PISO0 ATRIAPO	editar
Unidade Medida			Save	Cancel	
Unidades Atribuir					

Ilustração 20 - Ecrã editar

A Ilustração 21 representa um exemplo prático de utilização desta componente no projecto pois podemos verificar que o valor da localização é constituído por quatro segmentos. Ao editar o campo localização temos acesso a um primeiro *pop-up* que contém a informação relativa à constituição do campo, e um segundo *pop-up* com a lista de segmentos válidos para cada um dos campos seguindo a regra de campos flexíveis estabelecida pela *OEBS*.

O primeiro *pop-up* é todo gerado dinamicamente consoante os parâmetros de entrada fornecidos à acção “editar”. Os parâmetros de entrada existentes são o *In_id_flex_code* e o *In_application_id*, respectivamente, código de identificação do campo flexível e o identificador do tipo do módulo do *OEBS*. No exemplo de edição de localização o valor do parâmetro *In_id_flex_code* é “LOC#” e o valor do parâmetro *In_application_id* é “140”.

O segundo *pop-up*, a lista de valores, também é gerado dinamicamente consoante os valores seleccionados nos segmentos anteriores. Para ajudar o utilizador a encontrar um determinado valor com maior celeridade, foi implementada uma função de pesquisa que permite ao utilizador procurar o valor desejado através da descrição deste.

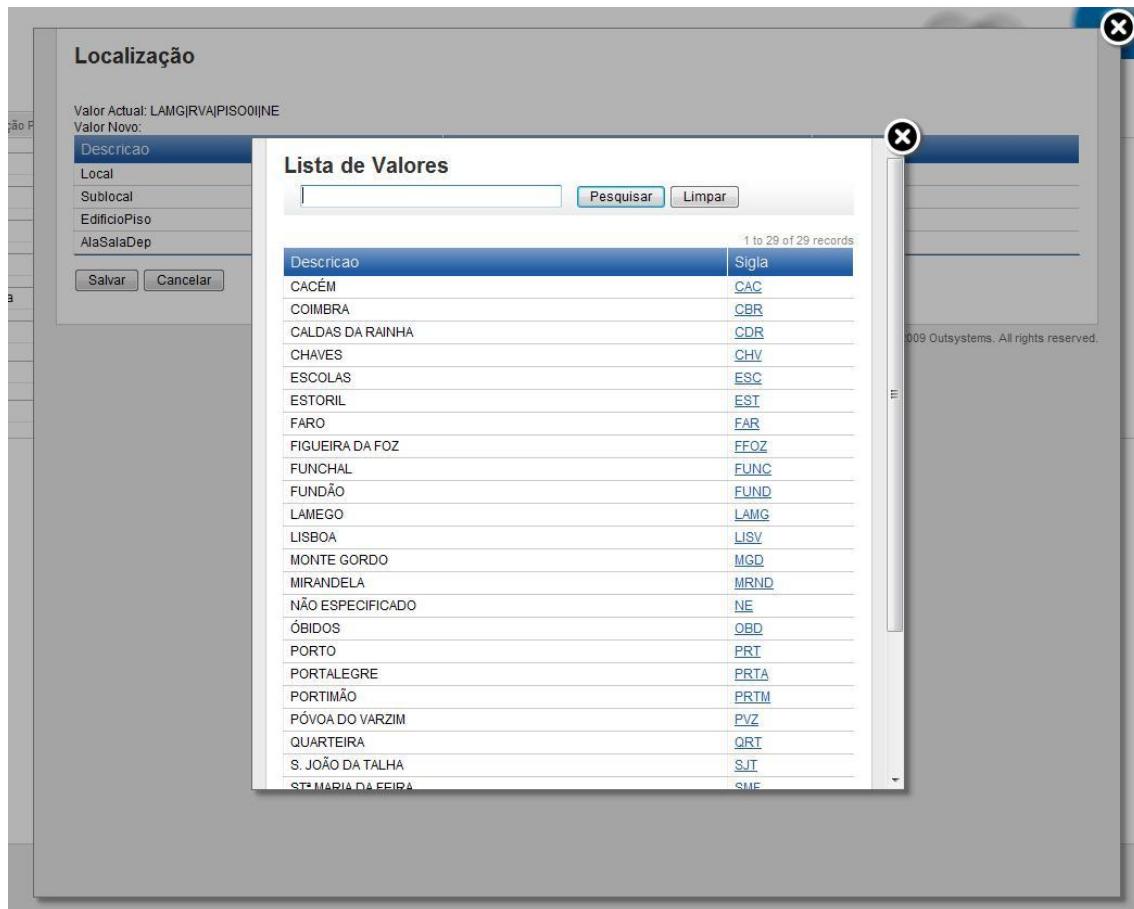


Ilustração 21 - Classificação da localização

No desenvolvimento deste componente foram encontradas várias dificuldades. Um dos principais desafios foi a implementação das interrogações à base de dados tendo em conta os vários parâmetros seleccionados anteriormente. A interrogação para a lista de valores do segmento deverá ser condicionada com os valores dos segmentos anteriores, pois assim o resultado da interrogação será sempre um conjunto de valores válidos e não todos os possíveis.

4.6.5 Package PL/SQL para adição do activo

Na biblioteca do *OEBS* (no módulo *Fixed Assets*) existe um procedimento *PL/SQL* que permite adicionar activos de forma simples ou detalhada, de acordo com os dados fornecidos ao formulário do *OEBS*. Em casos simples o Conector *Oracle* desenvolvido pela Truewind converte o método de adição do módulo *Fixed Assets* para o método em *OutSystems*, mas neste caso essa transição directa não é possível.

Esta inviabilidade na transição do método *Oracle* para *OutSystems* através do conector falha pela elevada complexidade do método de adição de activos. Este método tem parâmetros de entrada cujo tipo de dados não é primitivo (por exemplo *number*, *varchar*, *date* ou *boolean*), mas sim complexo (estruturas ou listas) e que não é suportado pelo conector.

Uma das lacunas do conector é a incapacidade de traduzir variáveis que não sejam do tipo primitivo para *OutSystems*. Para ultrapassar esta dificuldade foi desenvolvido um *package* em *PL/SQL* com um conjunto de funções.

Um *package* é um pacote que agrupa um conjunto de objectos do tipo *PL/SQL*, relacionando logicamente itens e subprogramas. Como evidenciado na Ilustração 22, os pacotes têm geralmente duas partes: uma especificação e um corpo. A especificação é a interface das suas aplicações e onde são declarados os tipos, variáveis, constantes, excepções, cursores e subprogramas disponíveis para serem usados. O corpo define totalmente os cursores e os subprogramas – são blocos de *PL/SQL* que podem aceitar parâmetros e ser invocados, podendo ser declarados como procedimentos ou como funções. Geralmente um procedimento é usado para executar uma acção e uma função para efectuar cálculos. A função é semelhante a um procedimento, excepto que a função tem de retornar um valor.

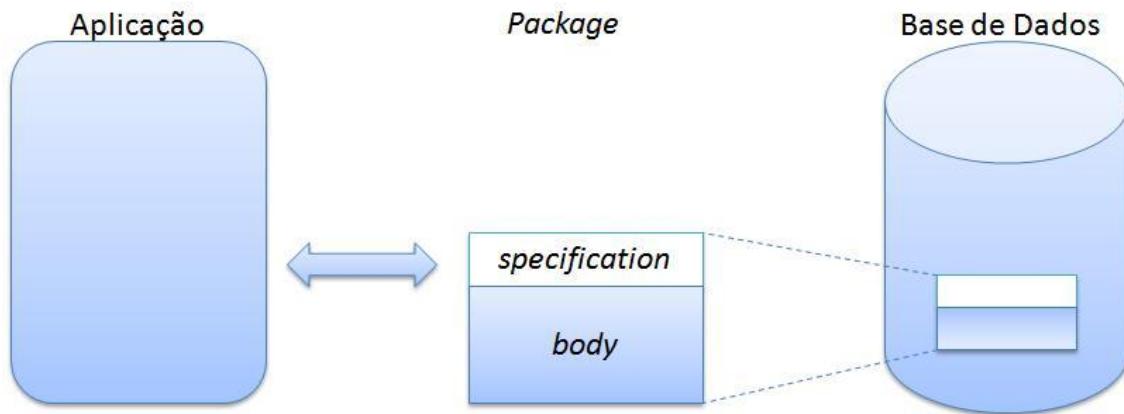


Ilustração 22 - Esquema de um package

O desenvolvimento deste pacote em *PL/SQL* envolveu uma profunda análise da execução do método e se todas as tabelas envolvidas de forma a não entrar em conflito com o modelo de dados do *OEBS*. Como o método de adição *fa_addition_pub.do_addition* do *OEBS* recebe como parâmetro um conjunto de valores de tipos não primitivos, para cada um desses parâmetros foram criadas tabelas auxiliares na base dados e os respectivos métodos de adição de dados para as preencher.

O modelo de dados desenvolvido (Ilustração 23) revela a implementação das tabelas criadas de modo a inserir e extraír os dados necessários para execução perfeita do método de adição do activo em *OEBS* e a sua posterior transição para *OutSystems*.

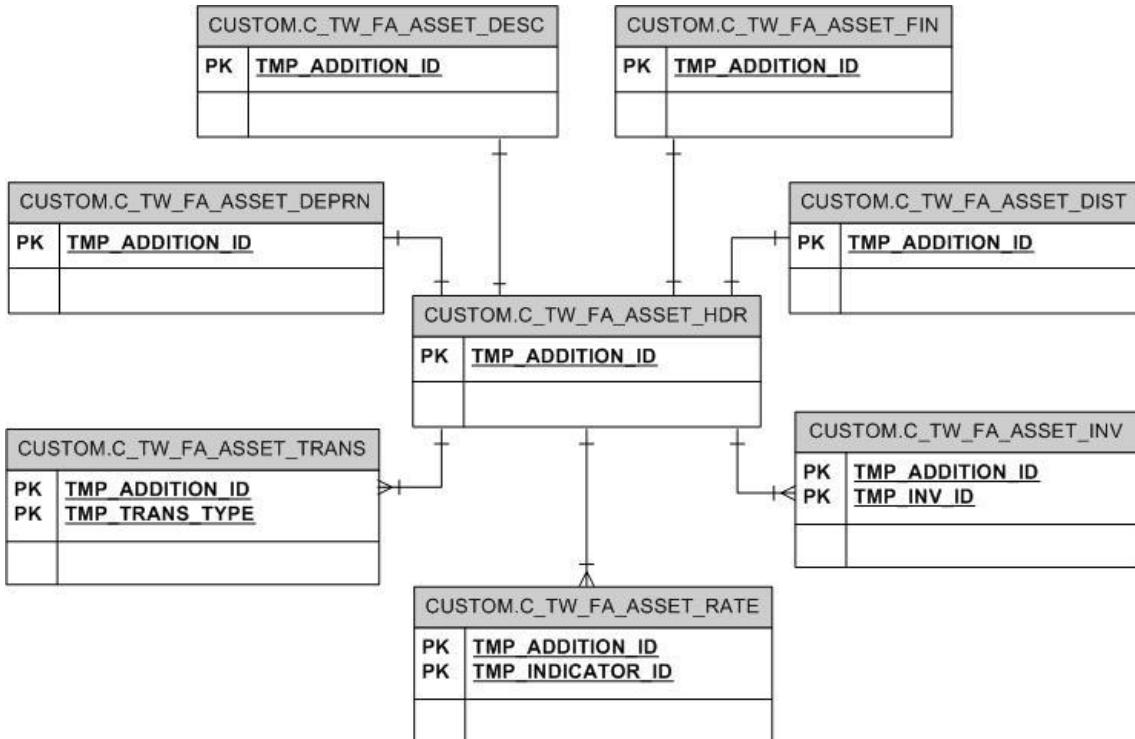


Ilustração 23 - Modelo de dados das tabelas auxiliares

Para o funcionamento correcto de todas as funções implementadas no *package* e tendo em conta uma análise detalhada ao método *fa_addition_pub.do_addition* do *OEBS*, foram criadas tabelas personalizadas que permitem suportar os dados necessários para a execução correcta do método. As tabelas criadas foram as seguintes (todas com o prefixo “C_TW_FA_ASSET_”):

- **HDR**: Contém informações de identificação única para um determinado activo, como a identificação de activos e código do tipo de livro.
- **DESC**: Contém informações relativas à descrição de um activo, tais como número de etiqueta, número de série e a descrição do activo.
- **TRANS**: Contém informações sobre a transacção.
- **FIN**: Contém informações financeiras sobre um determinado activo.
- **DEPRN**: Contém informações de depreciação de um determinado activo.
- **DIST**: Contém informações que representam uma única fonte ou linha de distribuição de destino.
- **INV**: Contém informações de uma factura única associada a um activo.
- **RATE**: Contém informações das taxas de uma factura única associada a um activo.

Para cada uma destas tabelas foram criadas funções de adição que permitem adicionar linhas às respectivas tabelas. Para a maioria das tabelas existem duas versões de adição: uma que apenas insere uma linha na tabela respectiva com os dados necessários e outra com todos os valores possíveis. A versão curta da inserção dos valores na tabela é utilizada na função de adição rápida enquanto que a segunda versão (com todos os valores possíveis) é empregada na função de adição detalhada. As excepções são as tabelas C_TW_FA_ASSET_DIST e C_TW_FA_ASSET_RATE, que têm apenas uma função de inserção de valores. Para estas tabelas não foi necessário implementar duas versões de inserção de dados porque as tabelas não têm um número considerável de colunas. Todas estas funções encontram-se no *package c_tw_fa_api*.

O pacote desenvolvido (*package c_tw_fa_api*) é constituído por um conjunto de funções de onde se destacam algumas:

- ***insert_asset_hdr*** (duas versões) – Insere uma linha na tabela C_TW_FA_ASSET_HDR e tem como variável de output o P_TMP_ADDITION_ID. É obrigatório que esta função seja invocada antes de todas as outras porque as restantes funções estão dependentes da variável de output retornada por esta função. Essa variável permite identificar todos os dados referentes a um determinado activo.
- ***insert_asset_desc*** (duas versões) – Insere uma linha na tabela C_TW_FA_ASSET_DESC..
- ***insert_asset_dist*** – Insere uma linha na tabela C_TW_FA_ASSET_DIST.
- ***do_addition*** – Insere um activo em *OEBS*, através dos dados recolhidos nas tabelas auxiliares e da posterior invocação do método *fa_addition_pub.do_addition* do *OEBS*. Ao contrário *insert_asset_hdr* esta função é executada no final de todas as funções do *package* desenvolvido.

4.6.6 Adição simples do activo

Uma das formas de adicionar um bem no módulo *FA* do *OEBS* é a adição simples. Esta é uma forma rápida e simples de adicionar um activo, tendo esta funcionalidade sido projectada para a aplicação desenvolvida.

Adição Simples do Activo

Número de Activo	068677	Unidades	1
Número de Etiqueta		Número de Fornecedor	0
Número de Série		Número da OC	
Tipo de Activo	CAPITALIZED	Valor de Aquisição	0
Nome Fornecedor		Início de Utilização	2010-08-19 August, 2010 YYYY-MM-DD
Número de Factura	0	Amortizar	<input checked="" type="checkbox"/>
Livro	TDP	Conveção Acertos Datas	
Método	STL	Data de Rateio	
Descrição	ROUTER	Nome Empregado	
Categoría	101.01 99	Número de Empregado	
Chave do Activo	02 12	Unidades Atribuídas	0
Activo Pai	0	Conta de Custos	01 02 311 020000 6628 000 510 editar *
Vida Útil	Anos: 2 * Meses: 12 *	Localização	LISV RIS6 PISO0 SALA6 editar *

Save Cancel

Ilustração 24 - Adição simples

O ecrã de adição simples foi implementado como está representado na Ilustração 24. Este ecrã permite uma interpretação evidente do formulário apresentado, com uma usabilidade prática e elementar. Neste ecrã, além da existência de simples campos de texto, existe também um campo constituído por uma lista de valores (o campo “Meses” na vida útil do bem), vários *links* de acesso à janela de campos flexíveis (já mencionada anteriormente no relatório), onde o utilizador pode escolher os valores necessários para preencher o campo flexível (os campos categoria, chave do activo, conta de custos e localização) e um campo de data, onde é utilizado um componente de calendário da *OutSystems* (campo de início de utilização).

Todos os campos obrigatórios para a função de adição funcionar correctamente estão devidamente assinalados para ajudar o utilizador a não cometer erros na validação dos dados introduzidos. Em caso de falha por parte do utilizador na introdução de dados, este é informado por parte da aplicação através de uma mensagem de erro (ver Ilustração 25).

Adição Simples do Activo

Número de Activo	068677	*	Unidades	1	*
Número de Etiqueta			Número de Fornecedor	0	
Número de Série			Número da OC		
Tipo de Activo	CAPITALIZED		Valor de Aquisição	0	*
Nome Fornecedor			Início de Utilização	2010-08-19	*
Número de Factura	0		Amortizar	<input checked="" type="checkbox"/>	
Livro	TDP		Convenção Acertos Datas		
Método	STL		Data de Rateio		
Descrição	ROUTER	*	Nome Empregado		
Categoría		editar *	Número de Empregado		
Chave do Activo	02 12	editar *	Unidades Atribuídas	0	
Activo Pai	0		Conta de Custos	01 02 311 020000 6628 000 510	editar *
Vida Útil	Anos: 2	*	Localização	LISV RIS6 PISO0 SALA6	editar *
	Meses: 12	<input type="button" value="Save"/> <input type="button" value="Cancel"/>			

Ilustração 25 - Exemplo de falha na adição do activo

A Ilustração 26 mostra o código fonte *OutSystems* responsável pelo processo de adição simples. Este fluxo começa com uma simples atribuição de valores a determinados parâmetros que são da responsabilidade do utilizador assinalados no ponto 1, tal como o número de identificação do utilizador que está a adicionar o activo. Durante o fluxo existe sempre uma verificação de dados introduzidos pelo utilizador; caso sejam inválidos é devolvida uma mensagem de erro como, por exemplo, no ponto 2 da imagem.

Os métodos do pacote desenvolvido em *PL/SQL* são invocados numa determinada ordem, ordem essa que tem de ser respeitada para o funcionamento correcto do processo. Este fluxo começa com a função *insert_asset_hdr*, que além de inserir dados na tabela *C_TW_FA_ASSET_HDR*, gera também um identificador único (o *P_TMP_ADDITION_ID*) e devolve-o como parâmetro de saída. Todas as restantes funções estão dependentes deste parâmetro para inserção dos dados nas respectivas tabelas. Este parâmetro serve como chave primária e todos os dados registados nas diferentes tabelas, relacionados com determinado activo, encontram-se identificados por este parâmetro. Após a inserção dos dados referentes a um determinado activo é invocada a acção *do_addition*, que insere o activo no *OEBS* através da invocação da função da *API* do módulo *FA* com os dados registados nas tabelas anteriores. Esta é a

última acção a ser invocada no processo de adição simples de um activo. Toda esta invocação de lógica do *OEBS* está enquadrada no ponto 3.

No final do fluxo, denotado pelo ponto 4 da imagem, é apresentada uma mensagem a informar o utilizador de que o activo foi adicionado com sucesso e o utilizador é direcionado para o ecrã da lista de activos.

No caso de algum parâmetro de entrada dado ao conjunto de métodos desenvolvidos em *PL/SQL* ser inválido, ou no caso de ocorrer algum erro durante a adição do activo, é lançada uma excepção. Qualquer erro é tratado pelo fluxo de excepção enquadrado no ponto 5 da imagem. Este tem a particularidade de devolver o erro que ocorreu e cancelar todas as acções envolvidas na transacção, respeitado assim a propriedade da transacção, a atomicidade, que consiste em garantir que uma transacção ocorra na sua plenitude, isto é, todas as unidades da acção tem de ter sucesso para que a acção seja efectuada com sucesso; em caso de falha a transacção é desfeita.

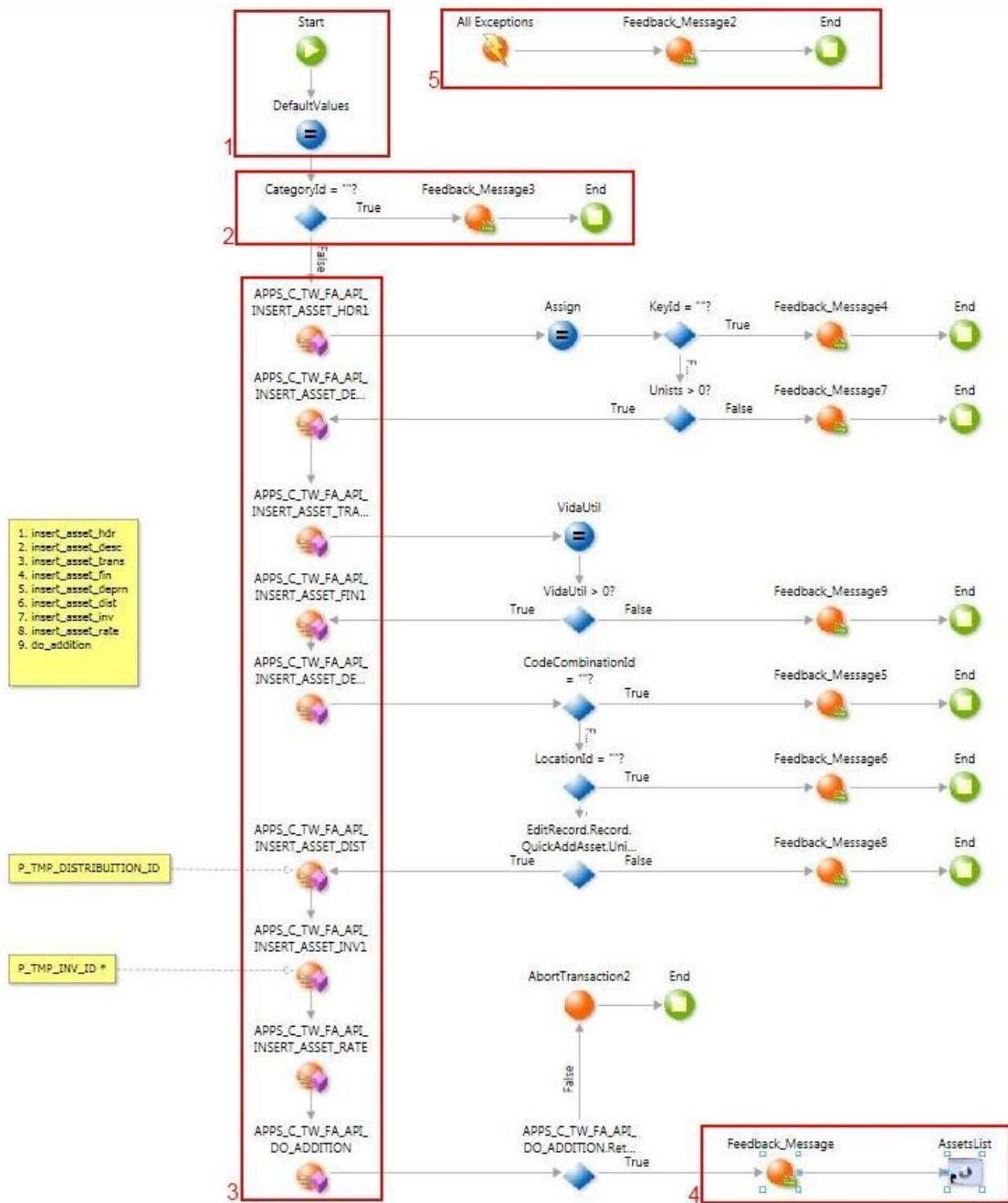


Ilustração 26 - Fluxo de adição simples

No final, se a acção de adição tiver sido realizada com êxito, é apresentada uma mensagem de sucesso com número do activo e o utilizador é direcionado para o ecrã com a lista de activos como se pode ver na Ilustração 27. No *OEBS*, o activo é adicionado na tabela de FA_ADDITIONS_B, sobre a qual será executado o programa concorrente do *OEBS* cujo objectivo é popular as restantes tabelas relacionadas com o modelo de dados do módulo FA do *OEBS*.

Turismo de Portugal
Gestão de Activos

✓ Activo 068677 adicionado com sucesso

Lista de Activos

Descrição do activo, Nº de serie, Nº de activo

[Adição Simples](#) | [Adição Detalhada](#) | [Adições em Bloco](#) | [Exportar para Excel](#)

Nº Activo	Nº Série	Descrição	Categoria	Tipo Activo	Livro	Início Utilização	Nº Factura	Nome Fornecedor	Nº Fornecedor
024970		ARMÁRIO - S/PORTAS	CIBE 103.01 01	CAPITALIZED	TDP	2006-09-29	FAC20060356		
024971		ARMÁRIO - S/PORTAS	CIBE 103.01 01	CAPITALIZED	TDP	2006-09-29	FAC20060356		
020891		CADEIRA FIXA - MADEIRA BEGE	CIBE 107.01 07	CAPITALIZED	TDP	2009-01-01	ND		
020897		CAMA INDIVIDUAL - METAL PRETO	CIBE 107.01 08	CAPITALIZED	TDP	2006-09-29	FAC20060356		
041081		COLCHÃO INDIVIDUAL	CIBE 107.01 10	CAPITALIZED	TDP	2006-09-30	FAC20060356		
020895		MESA APOIO - MADEIRA BEGE 50X40X50	CIBE 107.01 20	CAPITALIZED	TDP	2006-09-29	FAC20060356		
020892		CADEIRA FIXA - MADEIRA BEGE	CIBE 107.01 07	CAPITALIZED	TDP	2009-01-01	ND		
020915		MESA APOIO - MADEIRA BEGE 50X40X50	CIBE 107.01 20	CAPITALIZED	TDP	2006-09-29	FAC20060356		
020917		CAMA INDIVIDUAL - METAL PRETO	CIBE 107.01 08	CAPITALIZED	TDP	2006-09-29	FAC20060356		
020926		MESA APOIO - MADEIRA BEGE 50X40X50	CIBE 107.01 20	CAPITALIZED	TDP	2006-09-29	FAC20060356		
041088		COLCHÃO INDIVIDUAL	CIBE 107.01 10	CAPITALIZED	TDP	2006-09-30	FAC20060356		

Ilustração 27 - Adição simples com sucesso

4.6.7 Extensões criadas em OutSystems

A aplicação desenvolvida em *OutSystems* está intrinsecamente ligada à base de dados da *Oracle*. Para ter acesso às tabelas da *Oracle* e assim realizar as funcionalidades desenvolvidas, foi criada a extensão “*Assets_Entities*” que contém o conjunto de tabelas do *OEBS* necessário para a operação da aplicação.

Assets_Entities - Integration Studio

File Edit View Help

Extension Resources

Assets_Entities

1. Entities

- CONTABILIDADE_TP
- FA_ADD_WARRANTIES
- FA_ADDITIONS_V
- FA_ADDITIONS_WEB_V
- FA_ASSET_KEYWORDS
- FA_BOOKS
- FA_CATEGORIAS_V
- FA_CATEGORIES_B
- FA_CHAVE_ACTIVOS_V
- FA_DEPRN_DETAIL
- FA_DISTRIBUTION_HIST_WEB_V
- FA_DISTRIBUTION_HISTORY
- FA_INVOICE_DETAILS_V
- FA_LOCALIZACOES_V
- FA_LOCATIONS
- FA_MASS_ADDITIONS
- FND_FLEX_VALUE_SETS
- FND_FLEX_VALUES_VL
- FND_ID_FLEX_SEGMENTS_VL
- GL_CODE_COMBINATIONS_KFV
- PO_VENDORS

2. PO_VENDORS Entity Configuration

Name:	PO_VENDORS
Physical Table Name:	PO.PO_VENDORS
Logical Database:	ORACLE
Identifier:	VENDOR_ID
Description:	
Default Value behavior:	No conversion to/from Database

3. Attributes

Name	Mandat...	Auto N...	Delete Rule	Data Type
VENDOR_ID	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Integer
LAST_UPDATE_DATE	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Date Time
LAST_UPDATED_BY	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Decimal
VENDOR_NAME	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Text
VENDOR_NAME_ALT	<input type="checkbox"/>	<input type="checkbox"/>		Text
SEGMENT1	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Text

Ilustração 28 - Extensão Assets_Entities

A extensão é criada através do *Integration Studio* da *OutSystems* e está representada na Ilustração 28. No enquadramento 1 da imagem estão alocadas as tabelas do *OEBS* necessárias para aplicação da gestão dos activos. No ponto 2 é apresentado nome da tabela e logo abaixo no ponto 3 o conjunto de atributos da tabela seleccionada (nome, tipo, tamanho e descrição). O enquadramento 4 apresenta três botões importantes para a extensão: o “*Verify*”, que permite verificar se existe algum erro na extensão, o “*Verify and Save*” que verifica se existe algum erro e salva a extensão no disco e por último o botão “*Publish*” que permite publicar a extensão no servidor da plataforma para que possa ser utilizada pelas aplicações desenvolvidas na plataforma *OutSystems*.

Após a publicação da extensão, é possível utilizar esta na aplicação desenvolvida em *Service Studio* da *OutSystems*, permitindo assim o acesso às tabelas do *OEBS* por parte da aplicação. As tabelas foram sendo adicionadas à extensão à medida da necessidade das funcionalidades implementadas na aplicação. A adição de novas tabelas é muito trivial e de fácil gestão, bastando adicionar as novas tabelas à extensão e publicar a nova versão da extensão na plataforma.

Uma outra extensão criada foi a “*Assets Actions*”. Esta extensão agrupa todos os métodos desenvolvidos em *PL/SQL*, permitindo assim a sua utilização na aplicação *OutSystems*. A criação desta extensão não foi tão trivial como a criação da “*Assets Entities*”, pois esta envolve um processo mais complexo. Visto que *OutSystems* ainda não tem nenhuma função que permita importar acções de um *package*, foi necessário utilizar o conector desenvolvido pela Truewind que cria um ficheiro *DLL* com as funções do *package* e através da opção “*Import Actions from .Net Assembly*” do *Integration Studio* da *OutSystems* é possível criar a extensão com todas as acções criadas em *PL/SQL* transitadas para *C#* e no final para a plataforma da *OutSystems*. Após a adição da extensão “*Assets Actions*” no *Service Studio* da *OutSystems* por parte do programador, como podemos observar pela Ilustração 29, os métodos ficam disponíveis como qualquer outra ferramenta do *Service Studio*.

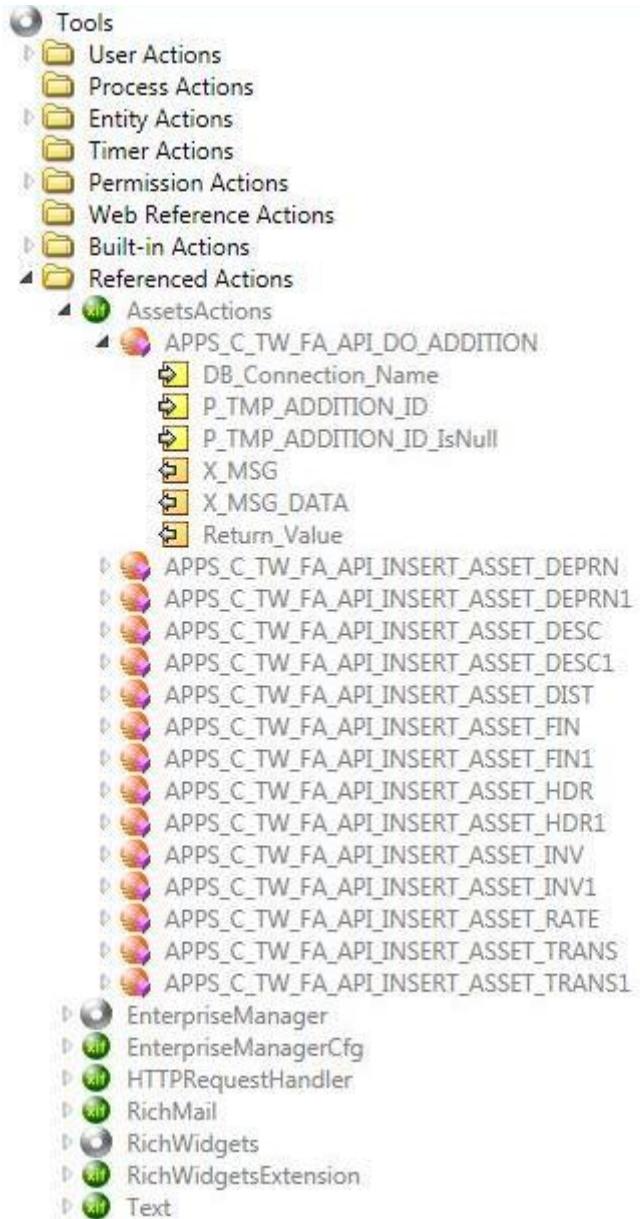


Ilustração 29 - Extensão Assets Actions

4.7 Análise de outras funcionalidades

Devido a várias dificuldades encontradas na concepção do projecto, principalmente a nível da análise do módulo *FA* do *OEBS* e disponibilização tardia de um ambiente de testes por parte do cliente, não foi possível implementar na janela temporal disponível para o Projecto de Engenharia Informática todas as funcionalidades previamente acordadas com o cliente. No entanto, estas funcionalidades foram analisadas e são apresentadas nesta secção.

4.7.1 Adição de activo (simples e detalhada)

A aplicação de gestão de activos da *Oracle E-Business Suite* possui duas formas de adição de activos: uma de forma simples já implementada (ver 4.6.6), e outra de forma detalhada.

O processo de adição simples permite adicionar os activos que apenas necessitam de um conjunto reduzido de especificações. Neste tipo de adição apenas as características obrigatórias do activo são preenchidas, sendo que os restantes dados são inseridos por omissão através da categoria, do livro e data em que o activo foi colocado em serviço.

Esta operação de adição tem campos obrigatórios e opcionais. A Tabela 6 indica campos estipulados para adição simples de um activo.

Campos obrigatórios	Campos opcionais
Número do Activo	Número da Etiqueta
Descrição	Número de Série
Tipo de Activo	Nome do Fornecedor
Categoria	Número da Factura
Unidades	Chave do Activo
Livro	Número do Fornecedor
Valor de Aquisição	Empregado Unidades
Início de Utilização	Empregado Nome
Localização	Empregado Número
Método	Activo do Grupo
Vida Útil: Anos	
Vida Útil: Meses	

Tabela 6 - Tabela com dados necessário para adição simples do activo

O processo de adição detalhada permite adicionar activos mais complexos, onde o processo de adição simples se revela insuficiente. O processo de adição simples assume determinados parâmetros por omissão, sendo apenas possível especificar esses parâmetros no processo de adição detalhada. A título de exemplo, esses parâmetros podem ser um valor de resgate ou a sua atribuição a um empregado.

Ambos os tipos de adição envolvem o seguinte modelo de dados constituído pelas seguintes tabelas *Oracle*:

- **FA_DISTRIBUTION_HISTORY** – Tabela que contém a informação relativa ao empregado, localização e valores de contabilidade atribuídos a cada activo. Esta tabela contém um historial de transferência do activo.
- **GL_CODE_COMBINATIONS** – Tabela que contém as combinações válidas para contabilidade dentro da estrutura da *Oracle*.
- **FA_EMPLOYEES** – Tabela que contém informação dos empregados da empresa. O *Oracle* usa-a para atribuir responsabilidades a cada empregado por determinado activo.
- **FA_LOCATIONS** – Tabela que contém os segmentos de localização válida de cada um dos activos. O *Oracle* usa a tabela para evitar a transferência de activos para locais inválidos.
- **FA_BOOKS** – Tabela que contém as informações necessárias para o *Oracle* calcular a depreciação de um activo.
- **FA_METHODS** – Tabela que contém a informação sobre os métodos de depreciação. O programa de depreciação usa a tabela para identificar as taxas de depreciação e método adequado a aplicar.
- **FA_BOOK_CONTROLS** – Tabela que contém informações de controlo que afectam todos os activos no livro de depreciação da empresa.
- **FA_ASSET_HISTORY** – Tabela que contém informação sobre as reclassificações de activos, capitalizações, ajustes de unidade, dados físicos e históricos do activo.
- **FA_CATEGORIES_B** – Tabela que contém informação sobre as categorias dos activos.
- **FA_ADDITIONS_B** – Tabela que contém informações descritivas que permitem identificar um determinado activo.
- **FA_ASSET_INVOICES** – Tabela que contém a informação associada à factura do activo.

A Ilustração 30 demonstra o modelo relacional que apresenta as ligações entre as tabelas mencionadas anteriormente e as respectivas chaves primárias.

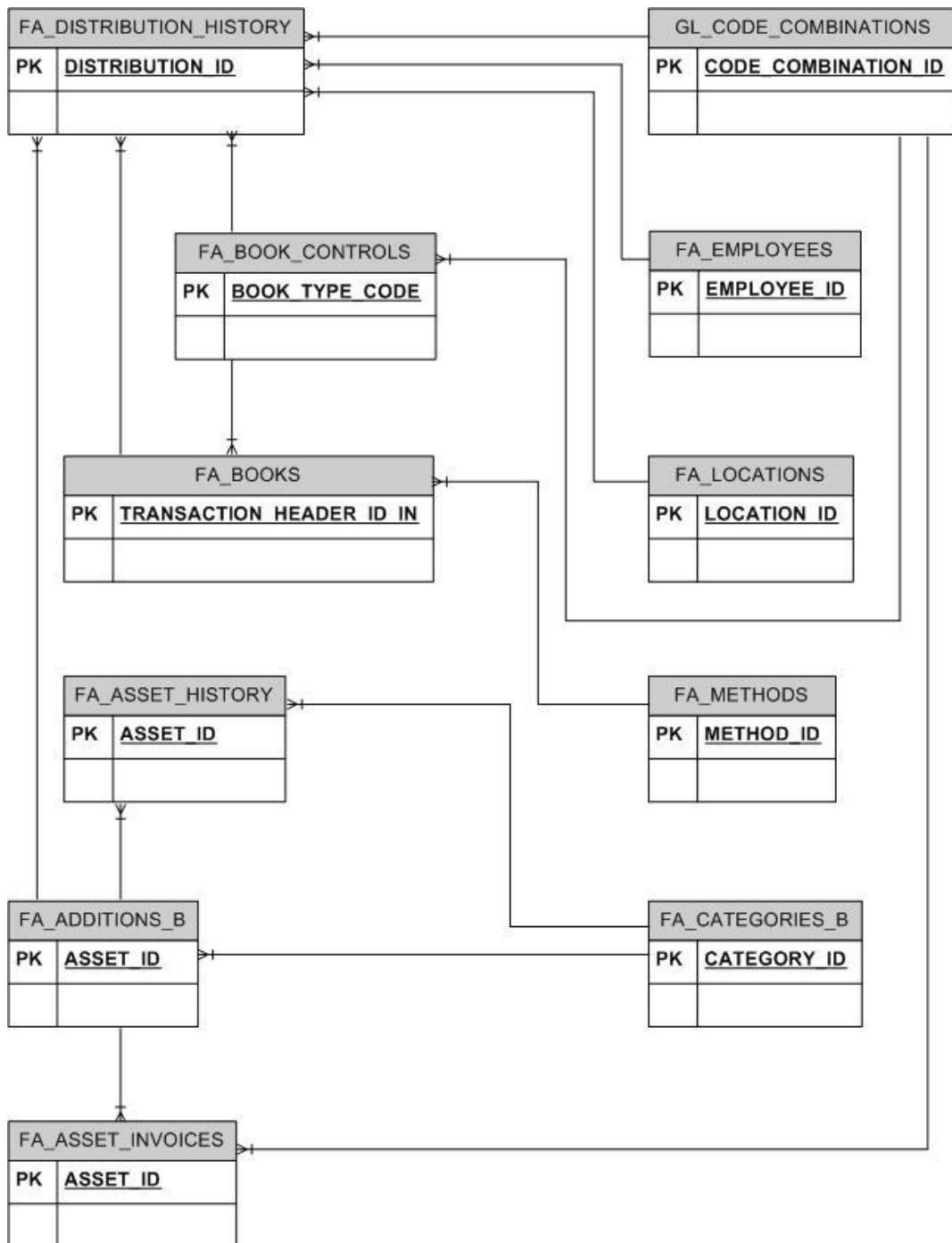


Ilustração 30 - Modelo de dados adição do activo

4.7.2 Adição em bloco e classificação de activos

O processo de adição em bloco é usado para adicionar activos de forma automática a partir de uma fonte externa. Cria um activo a partir de uma ou várias linhas de

distribuição da factura do *Accounts Payable* (módulo do *Oracle*), informações sobre activos de um outro sistema de activos, ou informação de qualquer outro sistema de alimentação.

É possível criar novos activos recorrendo ao *link* “Adições em Bloco” que se encontra no ecrã de listagem de activos, onde serão incluídos todos os activos que estão ainda em fase de configuração. Este ecrã funcionará como um repositório de trabalho onde se poderão efectuar diversas alterações aos activos antes de estes serem carregados na janela de activos e cujas alterações estão circunscritas à janela de trabalho não tendo, nessa fase, qualquer reflexão financeira ou patrimonial.

Os novos activos, criados a partir do módulo de gestão de fornecedores (*AP*), são carregados para a janela “Adições em Bloco”, e ficam em espera até serem classificados.

Na janela “Adições em bloco” podemos ter diversos activos em diversas filas de espera. O módulo *FA* tem diversas filas de espera configuradas, sendo no entanto possível ao utilizador responsável pela gestão patrimonial, adicionar novas filas de espera para facilitar a gestão e organização dos activos na área de trabalho. Neste projecto foram assumidos de acordo com o modelo do *OEBS* os seguintes tipos de fila de espera:

- **Novo** – são colocados os activos importados do módulo *AP* e cuja classificação patrimonial não foi iniciada.
- **Bloqueado** – podemos encontrar os activos cuja classificação patrimonial está a decorrer e ainda não se encontra concluída.
- **Lançar** – podemos encontrar os activos cuja classificação patrimonial se encontra finalizada. São os activos desta fila que ficam disponíveis para serem lançados.
- **Lançados** – podemos encontrar todos os activos lançados com sucesso para a janela “Activos”.
- **Apagar** – podemos encontrar os activos a remover, sendo que estes nunca irão ser lançados e por esse facto, será perdida a sua contabilização.

Na janela “Adições em Bloco” estarão disponíveis as linhas de investimento das facturas transferidas do módulo *AP*. A informação transferida para o módulo *AP* está circunscrita ao foro financeiro, pelo que será necessário proceder ao que se pode designar por “classificação patrimonial” dos activos recém-chegados. Os activos, no momento em que são exportados para o módulo *FA*, são colocados na fila “Novo”, e poderão ficar na fila “Bloqueado” durante todo o processo de classificação patrimonial.

A classificação patrimonial dos activos é de extrema importância e o seu rigor é fundamental para que os activos sejam correctamente criados e classificados. O utilizador com a responsabilidade da gestão patrimonial deve indicar, entre outras informações, a informação da chave do activo, a informação da descrição do activo, localização, chave contabilística de custos, empregado associado, descrição, marca, modelo, número de série, data de inicio de utilização, tipo de activo, etc.

A Ilustração 31 apresenta todo o fluxo da adição em bloco até à fase de classificação do património.



Ilustração 31 - Fluxo de adição em bloco até classificação de património

De seguida é descrito, cada uma das fases do fluxo representado na Ilustração 31:

- 1. Introduzir Factura de Fornecedor** – o processo é iniciado no momento em que o utilizador introduz uma factura no módulo *AP Oracle E-Business Suite - Gestão de fornecedores*.
- 2. Informação de Artigo** – ao introduzir a informação da factura de fornecedor, o utilizador tem de introduzir as linhas respectivas onde terá de seleccionar o artigo comprado.
- 3. Exportar para GL** – considera-se a factura concluída, quando esta estiver aprovada e contabilizada. Só depois dos movimentos contabilísticos respeitantes à factura terem sido exportados para o módulo de contabilidade (*GL*), é que é possível proceder ao seu tratamento patrimonial.
- 4. Exportar para FA** – para transferir as linhas de investimento para o módulo *FA*, e assim proceder ao seu tratamento patrimonial, tem de ser executado o pedido “Adições em Bloco Criadas”, disponível nos pedidos das responsabilidades do módulo *AP*; até ao momento este pedido é realizado na aplicação do *OEBS*.
- 5. Classificação Patrimonial** – nesta fase estão disponíveis na janela “Adições em Bloco” as linhas de investimento das facturas transferidas do módulo *AP*. A informação transferida do módulo *AP* ao *FA* abrange apenas o núcleo financeiro do activo. É necessário proceder à classificação patrimonial dos activos adicionados através do módulo *AP*. Os activos, no momento em que são exportados para o módulo *FA*, são colocados na fila “Novo”, e poderão ficar na fila “Bloqueado” até ao final do processo de classificação patrimonial.

A adição em bloco envolve o seguinte modelo de dados constituído pelas seguintes tabelas *Oracle*:

- **FA_MASSADD_DISTRIBUTIONS** – Tabela que contém a informação relativa ao empregado, localização e valores de contabilidade atribuídos a cada activo. Esta tabela contém um historial de transferência do activo.
- **GL_CODE_COMBINATIONS** – Tabela que contém as combinações de Contabilidade válidas dentro do ambiente do *OEBS*.

- **FA_EMPLOYEES** – Tabela que contém informação dos empregados da empresa. *Oracle* usa para atribuir responsabilidades a cada empregado por determinado activo.
- **FA_LOCATIONS** – Tabela que contém os segmentos de localização válida de cada um dos activos. *Oracle* usa a tabela para evitar a transferência de activos para locais inválidos.
- **FA_BOOK_CONTROLS** – Tabela que contém informações de controlo que afectam todos os activos no livro de depreciação da empresa.
- **FA_ADDITIONS_B** – Tabela que contém informações descritivas que permitem identificar um determinado activo.
- **FA_CATEGORIES_B** – Tabela que contém informação sobre as categorias dos activos.
- **AP_INVOICE_DISTRIBUTIONS_ALL** – Tabela que mantém a linha de informação da distribuição na entrada das facturas. Há uma linha para cada factura de distribuição. A linha de distribuição deve ser associada a uma factura e uma factura pode ter várias linhas de distribuição.
- **FA_MASS_ADDITIONS** – Tabela que armazena informações sobre recursos que pretendemos que sejam adicionados automaticamente em *Oracle Assets* de um outro sistema. Pode usar a tabela para carregar activos para conversão do sistema.
- **FA_DELETED_MASS_ADDITIONS** – Tabela armazena informações sobre adição em bloco que não pretendemos publicar em *Oracle Assets*.
- **FA_ASSET_KEYWORDS** – Tabela armazena combinações de valores válidos para a chave de activo.

A Ilustração 32 representa o modelo relacional que demonstra as ligações entre as tabelas mencionadas anteriormente e as respectivas chaves primárias.

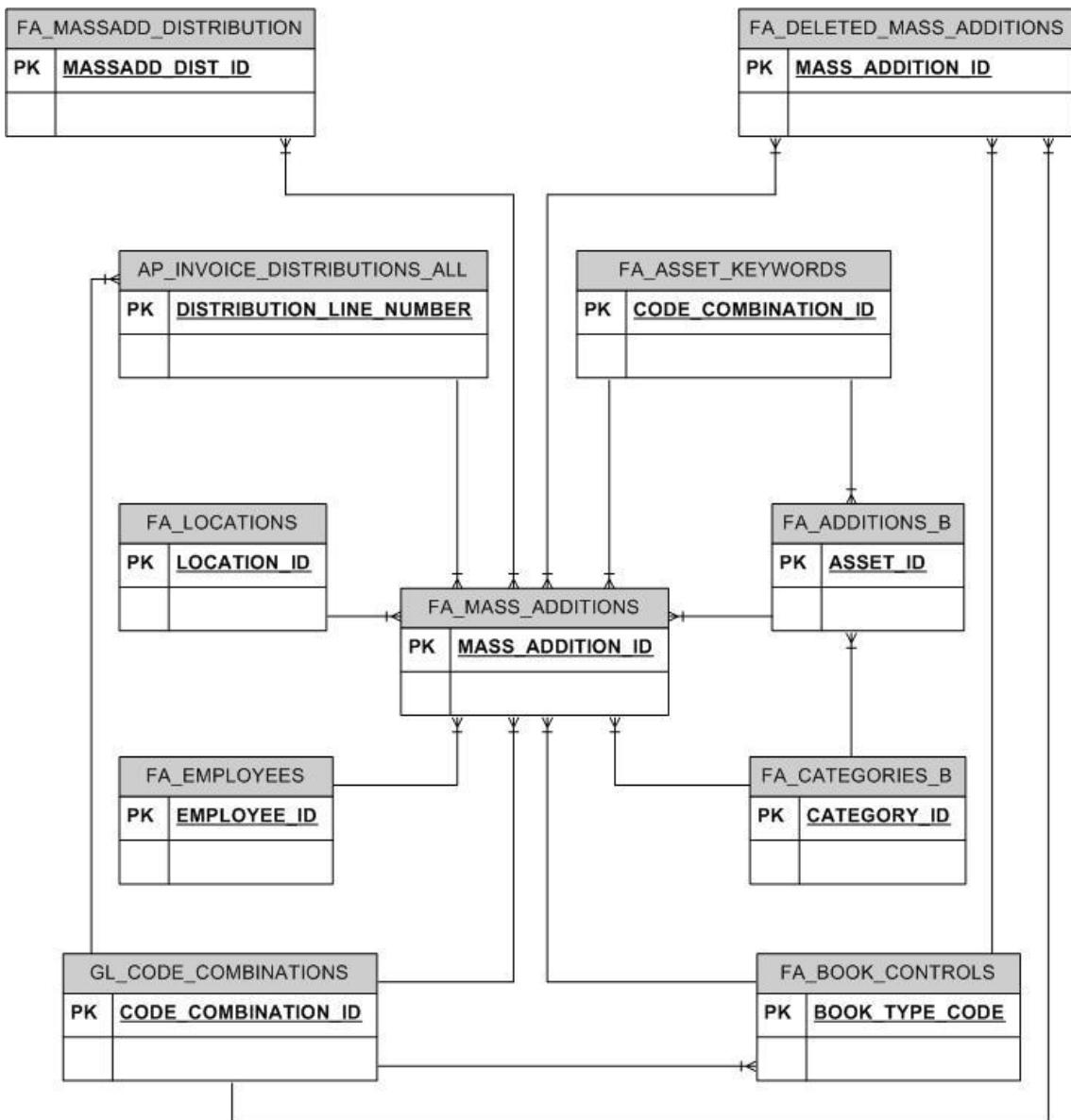


Ilustração 32 - Modelo de dados de adição em bloco

4.7.3 Abate de activos

O processo de abate consiste em apresentar um activo como perdido, furtado, danificado, vendido, devolvido, ou qualquer outro motivo que levou a empresa a parar de utilizar o activo.

Este processo de abate de activos pode ser feito em duas formas em *OEBS*: um abate individual ou abate em bloco. Considerando os requisitos iniciais do cliente, a análise foi apenas focada no abate individual.

O abate individual pode ser realizado por valor ou por unidades, e para tal, deve-se introduzir a informação financeira respeitante ao abate, nomeadamente o número de unidades a abater ou o valor a abater. O módulo *FA* possibilita estas duas formas de abate: por unidades ou por valor, sendo que ao abater por valor, o valor restante será distribuído equitativamente por todas as unidades.

Esta funcionalidade ainda não chegou a ser implementada no projecto e existem várias questões pendentes relativamente à implementação do abate. Uma das questões é se será criado um ecrã para realizar esta operação ou adicionada esta funcionalidade no ecrã de edição do activo ou numa coluna adicional do ecrã de listagem, com um botão que permite realizar o respectivo abate.

Será importante que, antes de abater um activo, o utilizador tenha a possibilidade de verificar se o abate está a ser realizado para o activo correcto. Para tal, é importante apresentar um conjunto de características importantes do activo antes de executar a operação de abate.

Na aplicação do *OEBS* o processo de abate passa por proceder ao preenchimento da informação patrimonial do abate, como sendo o tipo de abate, a data do abate, a autorização de abate, a proposta de abate e a data de autorização de abate.

Existe também a possibilidade de desfazer um abate mas esta funcionalidade não é um requisito do cliente. Caso tenha sido efectuado um abate dum activo por erro de utilizador, é sempre possível desfazer o abate do activo directamente na aplicação do *OEBS*, desde que o período do módulo *FA* ainda não tenha sido encerrado.

4.8 Transição da aplicação *OutSystems 4.2* para *5.0*

Devido a uma actualização da plataforma *OutSystems* da versão 4.2 para a versão 5.0 nos servidores do cliente, todas as aplicações desenvolvidas na versão 4.2 tiveram que ser transitadas para 5.0. Esta actualização não é um processo totalmente automático por

haver a necessidade de garantir a estabilidade da aplicação após a transição, devido à existência de *breaking changes*¹⁵ relevantes tais como:

- Eliminação na versão 5.0 das *tags paragraph, bold, italic, e underline*
- A conversão para decimal na versão 4.2, que era efectuada utilizando vírgula (”,”) na língua portuguesa e ponto (”.”) na língua inglesa, na versão 5.0 passou a ser utilizado em todas as línguas o ponto (”.“)
- Na versão 4.2, as expressões condicionais (*if*) retornavam sempre o tipo de dados texto, enquanto na versão 5.0 retornam agora o tipo de dados colocado como parâmetro de entrada da expressão condicional.

Para a migração para a nova versão da aplicação foi necessário deixar um clone da aplicação em 4.2, para rever e comparar todas as funcionalidades desenvolvidas até ao momento, de modo a verificar todos os fluxos das acções e realizar os testes necessários para garantir o funcionamento correcto da aplicação em 5.0 tal como estava em 4.2 antes da finalização da migração.

4.9 Testes

Em qualquer aplicação desenvolvida, o processo dos testes [15] é fulcral para o sucesso da aplicação. O processo dos testes tem como objectivos focar na prevenção de erros, descobrir os sintomas, diagnosticar e corrigir os erros da aplicação.

Os testes realizados neste projecto tiveram como objectivo validar a qualidade dos dados na base de dados do cliente, bem como a qualidade do código desenvolvido através de realização de testes unitários e funcionais. A abordagem mais tradicional seria alocar uma pessoa (com maior experiência e conhecimento do produto) no projecto, que para cada funcionalidade implementada realizava os testes necessários para validar a funcionalidade implementada. Neste caso, como o projecto incluía um único elemento, foi seguida a metodologia *scrum* e no final de cada funcionalidade

¹⁵ *Breaking Changes* - Uma alteração numa parte de um sistema de *software* que faz com que as outras partes falhem. Ocorre mais frequentemente em bibliotecas compartilhadas de código utilizado por várias aplicações.

implementada o programador realizou os testes necessários para validar a funcionalidade desenvolvida.

4.9.1 Qualidade dos dados

A qualidade dos dados de uma base de dados é um requisito essencial para implementação da solução desejada pelo cliente, pois vai facultar uma interacção íntegra da aplicação desenvolvida com os dados do cliente. A qualidade dos dados está intrinsecamente ligada aos processos de armazenamento, recolha, organização, monitorização e análise de dados.

Relativamente à aplicação desenvolvida, a lógica seguida para a componente “pesquisa” foi que, é necessário identificar um activo por conjunto de características e não apenas pelo seu identificador único. Para que a funcionalidade de pesquisa funcione devidamente é necessário que o activo seja identificado por uma outra característica além do identificador único, como por exemplo o número do activo, o número de série e/ou o número de etiqueta.

A qualidade dos dados depende da forma como estes são introduzidos no sistema, logo o processo da migração dos dados das folhas de *Excel* até à base de dados do *Oracle* foi essencial para verificar a qualidade dos dados recolhidos e posteriormente inseridos. O papel da tabela *buffer CPRX_FA_MASSADD_TDP* foi muito valorizado pois permitiu concretizar um rastreio sobre os dados recolhidos, corrigir e introduzir informação necessária de forma a obter uma tabela com dados coerentes. Um outro factor que pode levar à falta de qualidade dos dados é a introdução de dados incoerentes por parte dos utilizadores, por falta de conhecimento ou por distração. Uma das formas de conservar a qualidade dos dados do sistema é concentrar no processo de introdução dos dados. Na aplicação desenvolvida, o método que é responsável pela introdução de novos activos na base dados apresenta um conjunto de mecanismos que impedem a introdução de dados erróneos através de validações da informação durante o fluxo da introdução do activo e igualmente proporciona uma resolução de possíveis problemas no momento mais oportuno. Este tipo de estratégia preventiva assegura uma base de dados coerente e estável.

4.9.2 Testes unitários

É uma fase de testes que serve para validar blocos de código desenvolvidos; os testes são preparados pelo próprio programador. Os alvos dos testes são métodos e pequenos trechos de código; o objectivo é encontrar falhas de funcionamento em pequenas partes do sistema sem interferência de outros componentes do sistema. O teste consiste em validar os dados através dos parâmetros de entrada e saída.

Os testes unitários permitem ao programador verificar e assegurar que cada unidade de código é executada conforme o esperado. Este tipo de teste é particularmente útil para revelar falhas da componente de software relacionadas com funções incorrectas, suporte dos dados e de acesso à base de dados. À medida que cada método, função e *query* era desenvolvida iam sendo realizados testes unitários de forma a validar as funcionalidades implementadas através da introdução de parâmetros de entrada válidos e inválidos sobre o método, função e *query* de forma a testar a lógica do código implementado e verificar se o fluxo do código desenvolvido se comportava correctamente com os dados inseridos. Durante a execução deste tipo de testes é realizada uma revisão do código desenvolvido, definido um tratamento de erros, apresentadas as mensagens de sucesso ou insucesso ao utilizador.

No projecto os testes unitários foram realizados através de um ecrã de teste (Ilustração 33) muito simples constituído por um campo de entrada, onde são introduzidos vários *inputs* (valores desejados e valores não desejados) de cada vez, um botão que executa a função a ser testada e um campo de saída que retorna o valor da função. Simultaneamente à realização do teste é examinado o fluxo da função, as variáveis, os parâmetros de entrada e saída, as mensagens de erro, as excepções e a forma correcta de as tratar.

Na Ilustração 33 é retratado o teste sobre a função GetLocationId onde dado o conjunto de segmentos que define um local é devolvido o respectivo identificador único. No campo “*input*” é dado como valor do parâmetro de entrada o conjunto de segmentos. Ao clicar no botão “Testar” é executada a função teste e devolvido o resultado no campo de “*output*”. Para as outras funções de teste com a mesma lógica bastava apenas substituir a lógica implementada através do botão “Testar” pela nova função de teste.

Ecrã de teste de funções

Input FUNC AIC PISO0 SALACCTV	Função de teste <input type="button" value="Testar"/>	Output 3902
---	---	-----------------------

Ilustração 33 - Ecrã de testes

Os testes também foram efectuados para todas as *queries* que estão incluídas na aplicação. Estes eram mais triviais de ser realizados porque a ferramenta da *query* em *Service Studio* da *OutSystems* permite inserir os valores de *input* e recolher o resultado através dos separadores que estão presentes na ferramenta (Ilustração 15). Uma outra forma de teste aplicada foi o modo de *debug* proporcionado pelo *Service Studio* da *OutSystems*, um método que permite acompanhar todo o fluxo da acção de forma faseada e onde podemos observar qual o valor de cada uma das variáveis intervenientes, o resultado das *queries*, os diferentes caminhos do fluxo e o resultado final do método.

Entre muitas outras, as funções e *queries* sobre as quais foram realizados os teste unitários, foram a função que devolve o número de identificação da chave de um determinado activo dado as siglas de constituem o campo flexível da chave (GetCategoryId), a *query* que devolve todos os locais disponíveis no módulo FA do OEBS (GetLocal) e a *query* que devolve todos os activos de acordo com os valores de filtro inseridos (GetAssetsList).

4.9.3 Testes funcionais

Testes funcionais são um tipo de teste que garante que todos os requisitos funcionais do cliente são satisfeitos.

Nesta fase foi usado o teste de caixa preta que consiste em testar o *software* a partir da sua funcionalidade, através dos parâmetros de entrada e saída, não tendo em consideração os detalhes do código.

Este tipo de teste é particularmente útil para revelar falhas no programa relacionadas com funções incorrectas, erros de interface, erros de estrutura, de acesso à base de dados e de determinadas funcionalidades.

Numa situação ideal todos os parâmetros de entrada seriam testados, mas isto teria encargos muito altos para um projecto com apenas um indivíduo. Devido a essa desvantagem apenas foram efectuados os testes sobre a lista de requisitos fornecida pelo cliente e estruturada pelo programador.

Como a metodologia adoptada para este projecto foi o *scrum*, que consiste no desenvolvimento de software utilizando uma prática iterativa e incremental, foi adoptado, uma integração contínua dos testes unitários em todo o processo de desenvolvimento do *software*. Como esta metodologia está dividida em ciclos e o produto de cada ciclo é uma versão funcional do programa final, são efectuados testes unitários no final de cada ciclo. Este processo permitiu aumentar a qualidade do produto através das validações efectuadas no final de cada ciclo. Esta fase de teste induz numa redução do esforço de desenvolvimento porque permite circunscrever e resolver os erros numa fase inicial de desenvolvimento e identificar que componentes recentemente integradas são responsáveis pelo estado incoerente do produto. Dentro de cada um dos ciclos, as funcionalidades foram sempre testadas e acrescentadas de forma sucessiva e contínua, validando sempre a integridade do produto no final de cada integração através dos testes.

A fase de teste funcional sobre a aplicação desenvolvida teve como base o teste de caixa preta e lista de requisitos inicial. A Tabela 7 apresenta os componentes sobre os quais foram realizados os testes funcionais.

Funcionalidade	Teste	Validação
Listagem	Clicar no <i>link</i> de “Lista de activos”	É apresentada uma lista de todos os activos da empresa
Pesquisa	Clicar no botão “Pesquisar”.	É apresentada uma lista de activos de acordo com os filtros seleccionados

Funcionalidade	Teste	Validação
Consulta	Clicar no <i>link</i> de um determinado activo.	É apresentada a informação de um determinado activo.
Adição Simples	Após clicar no <i>link</i> “Adição Simples” e preencher os campos obrigatórios.	É adicionado um activo na lista de activos da empresa
Campos Flexíveis	Após clicar no <i>link</i> “editar” no campo da localização.	É possível preencher o campo localização com um conjunto de segmentos que formam a localização.

Tabela 7 - Lista de teste funcional

Em certas funcionalidades desenvolvidas foi efectuado um teste funcional mais específico devido à complexidade da funcionalidade. Este teste consistiu na inserção de um conjunto de valores possíveis para operação de pesquisa para depois analisar o resultado obtido. Um exemplo de valores inseridos e o resultado obtido vem representado na Ilustração 34.

Campo de Pesquisa	Localização					Categoria			Intervalo de início de utilização		Resultado
	Local	Sublocal	Edifício	Piso	Ala/Sala Dep	Modelo Legal	Classe / Tipo Bem	Bem	Data de início	Data de fim	
Null	TODOS	TODOS	TODOS	TODOS	TODOS	TODOS	TODOS	TODOS	Null	Null	Lista de todos os activos
router	TODOS	TODOS	TODOS	TODOS	TODOS	TODOS	TODOS	TODOS	Null	Null	Lista de activos com características desejadas
router	Lisboa	Rua D.Luís	TODOS	TODOS	TODOS	TODOS	TODOS	TODOS	Null	Null	Lista de activos com características desejadas
router	Lisboa	Rua D.Luís	TODOS	TODOS	TODOS	TODOS	TODOS	TODOS	Null	Null	Lista de activos com características desejadas
router	Lisboa	Rua D.Luís	PISO2	TODOS	TODOS	TODOS	TODOS	TODOS	Null	Null	Lista de activos com características desejadas
router	Lisboa	Rua D.Luís	PISO2	Geral	TODOS	TODOS	TODOS	TODOS	Null	Null	Lista de activos com características desejadas
router	Lisboa	Rua D.Luís	PISO2	Geral	CIBE	TODOS	TODOS	TODOS	Null	Null	Lista de activos com características desejadas
router	Lisboa	Rua D.Luís	PISO2	Geral	CIBE	Equip. infor:Hardware	TODOS	Null	Null	Null	Lista de activos com características desejadas
router	Lisboa	Rua D.Luís	PISO2	Geral	CIBE	Equip. infor:Hardware	Router	Null	Null	Null	Lista de activos com características desejadas
router	Lisboa	Rua D.Luís	PISO2	Geral	CIBE	Equip. infor:Hardware	Router	01-08-2009	Null	Null	Lista de activos com características desejadas
router	Lisboa	Rua D.Luís	PISO2	Geral	CIBE	Equip. infor:Hardware	Router	01-08-2009	01-05-2010	Null	Lista de activos com características desejadas

CIBE - Cadastro e inventario dos bens do estado

Ilustração 34 - Teste funcional para a pesquisa

Este tipo de teste não garante que o programa funcione na sua totalidade sem qualquer tipo de erro; permite somente fornecer bons indícios sobre a qualidade do programa e detectar erros que podem ser facilmente resolvidos.

Teste de *software* é uma das actividades mais custosas do processo de desenvolvimento de *software* porque consome uma quantidade significativa de recursos de um projecto. A perspectiva crítica do programador em relação à aplicação desenvolvida é responsável pelo custo associado a esta actividade.

A fase de teste de software é uma actividade muito importante porque, além de contribuir directamente para a qualidade do resultado final, reflecte-se também numa

redução do esforço de desenvolvimento, através da detecção de erros na fase inicial do projecto. A detecção de erros na fase mais tardia do projecto podia envolver custos mais elevados a nível dos recursos disponíveis para o projecto.

Capítulo 5

Conclusões e trabalho futuro

5.1 Conclusões

O principal objectivo deste projecto foi o desenvolvimento de uma aplicação de gestão de património em *OutSystems* integrada com o sistema *Oracle E-Business Suite*.

A aplicação desenvolvida tira proveito dos processos de negócio existentes no *ERP* da *Oracle*, reutilizando-os e compondo-os de uma forma mais simples e mais adequada às necessidades particulares do cliente em questão. Esta aplicação não pretende de alguma forma substituir a aplicação existente em *OEBS*, mas sim providenciar ao cliente uma forma mais eficiente de trabalhar. Neste aspecto, a aplicação cumpre os seus objectivos, pois nas funcionalidades implementadas reduz a sua complexidade e minimiza a sua curva de aprendizagem¹⁶.

Utilizando a tecnologia *OutSystems*, foi possível reutilizar a lógica de negócio usada em *Oracle E-Business Suite*, de modo a que a listagem de activos fosse apresentada com a informação mais relevante para o cliente e com a possibilidade de facilmente navegar e pesquisar um activo com determinadas características. Através do próprio ecrã de listagem de activos é possível aceder de uma forma elementar para o utilizador aos detalhes específicos de cada activo.

Foi também desenvolvido em *OutSystems* uma solução para uma função essencial do *Oracle E-Business Suite*: os campos flexíveis. Estes são campos compostos por vários segmentos e são empregados em diversos formulários da interface gráfica do *OEBS*. A

¹⁶ Uma curva de aprendizagem descreve o grau de sucesso obtido durante a aprendizagem em decorrência do tempo.

funcionalidade desenvolvida permite facilmente definir o valor destes campos compostos, independentemente do seu número de segmentos, e pode ser reutilizada em outras aplicações desenvolvidas em *OutSystems* de uma forma prática e simples.

A função de adição simples do activo no *ERP* da *Oracle* foi implementada e testada na solução desenvolvida. Comparativamente à função disponibilizada pelo *OEBS* é mais *user friendly*. O utilizador pode aceder facilmente ao *link* de adição do activo que está disponível no ecrã da lista dos activos e o próprio ecrã de adição do activo tem a indicação se os campos a preencher são obrigatórios ou opcionais, simplificando a inserção e validação em oposição ao actualmente implementado no *OEBS*.

A realização do projecto teve vários obstáculos tais como o conhecimento limitado relativamente ao universo de negócio que envolve o módulo *FA* e *AP* do *Oracle E-Business Suite*, a análise das funcionalidades do módulo que demorou mais tempo que o inicialmente estimado devido à sua elevada complexidade, a disponibilização tardia de um ambiente de testes por parte do cliente e a complexidade inerente à construção deste tipo de aplicações integradas com *Oracle E-Business Suite*. Estes factores, aliados à alteração de prioridades das tarefas prevista na metodologia *scrum*, fazem com que nem todos os requisitos presentes no *product backlog* inicial do projecto tenham sido concluídos até ao momento.

Perante as adversidades encontradas ao longo do projecto como no caso da análise do fluxo de negócio do *ERP* (*Oracle E-Business Suite*) que teve uma forte influência na alteração do planeamento inicial, concluiu-se que a experiência académica se mostrou insuficiente para ultrapassar esta barreira no tempo inicialmente estipulado. Ainda assim, a aprendizagem das novas ferramentas, algumas das quais nunca foram utilizadas na FCUL, para o desenvolvimento da aplicação de gestão de património foi ultrapassado com relativa tranquilidade porque a faculdade foca na aprendizagem de diversas tecnologias de programação o que facilitou na adaptação das novas ferramentas de programação utilizadas no projecto. Um outro obstáculo foi a tardia disponibilização do ambiente de teste, que foi da total responsabilidade do cliente que não cumpriu com o planeamento estabelecido inicialmente. O período de tempo em que o ambiente de teste estava indisponível foi aproveitado para participar em outros projectos dos clientes da Truewind. Estes projectos permitiram ganhar competências técnicas em *OutSystems*,

compreender o mundo de negócio relacionado com o património de uma empresa em outras vertentes, ganhar contacto com outras tecnologias e desenvolver capacidades de gestão.

O percurso desde o início da licenciatura até à conclusão do PEI permitiu adquirir uma experiência académica que passou por: projectos realizados, aulas assistidas e formação em diversas linguagens, o que permitiu uma fácil adaptação ao mundo profissional. Igualmente foi adquirida uma experiência humana através da interacção com diversos grupos de trabalho ao longo do percurso académico e com diferentes professores, o que permitiu uma integração mais espontânea na empresa. Estas experiências foram úteis na aquisição de experiência profissional uma vez que durante o desempenho do PEI houve interacção: com o cliente em diversas alturas da implementação da aplicação, com a entidade patronal sobre algumas divergências como o atraso do ambiente de teste, com equipas de trabalho através da participação em alguns projectos da Truewind.

O PEI permitiu obter uma certificação em *OutSystems*, adquirir competências no desenvolvimento em aplicações *Web*, aprender a utilizar metodologias adequadas aos diferentes tipos de projecto, representar diferentes papéis no desenvolvimento de projectos participados (como por exemplo *product owner* e *team member* de acordo com a metodologia *scrum*), adquirir competências em diversos módulos do *Oracle E-Business Suite* e fortalecer saberes em outras linguagens de programação como *C#* e *PL/SQL*.

5.2 Trabalho futuro

Como já foi referido anteriormente, existem ainda requisitos do cliente no *product backlog* por implementar. Para estes requisitos, foram efectuadas análises e recolha de dados para dar início à sua implementação.

Os requisitos que se encontram por implementar são:

- A função de adição de activo detalhada
- A função de adição em bloco e classificação de activos
- A função do abate do activo

Após o desenvolvimento de cada uma das funcionalidades referidas anteriormente, deverão ser realizados os testes de *software* (unitários e funcionais) necessários para garantir a qualidade do produto desenvolvido.

Tal como acontece com qualquer produto realizado para um cliente, o fornecimento de suporte aplicacional sobre a aplicação desenvolvida também será efectuado, de forma a garantir a legitimidade do produto.

Bibliografia e Referências

[1]. Truewind. [Online] [Citação: Setembro de 2009][2]

<http://www.truewind.pt/>

[2]. ERP (Enterprise Resource Planning). [Online] [Citação: Setembro de 2009]

http://www.cio.com/article/40323/ERP_Definition_and_Solutions

[3]. Oracle E-Business Suite. [Online] [Citação: Setembro de 2009]

<http://www.oracle.com/us/products/applications/ebusiness/018714.htm>

[4]. Oracle E-Business Suite Financials. [Online] [Citação: Outubro de 2009]

<http://www.oracle.com/applications/financials/intro.html>

[5]. Oracle Assets. [Online] [Citação: Outubro de 2009]

<http://www.oracle.com/applications/financials/assets.html>

[6]. OutSystems. [Online] [Citação: Setembro de 2009]

<http://www.outsystems.com/>

[7]. OutSystems Agile Network. [Online] [Citação: Setembro de 2009]

<https://secure.outsystems.com/network/>

[8]. Sql Server 2005. [Online] [Citação: Janeiro de 2010]

<http://www.microsoft.com/sqlserver/2005/en/us/default.aspx>

[9]. R. Ramakrishnan, J. Gehrke, *Database Management Systems*, McGraw Hill, 3^a edição, 2002. ISBN 0072465638

[10]. **Steven Feuerstein**, *Oracle PL/SQL Programming*, O'Reilly, 3^a edição, 2002.
ISBN 0596003811

[11]. **Scrum**. [Online] [Citação: Outubro de 2009]

<http://www.controlchaos.com/about/>

[12]. **User Stories**. [Online] [Citação: Janeiro de 2010]

<http://www.agile-software-development.com/2008/01/user-stories-answers-on-postcard-please.html>

[13]. **Ivan David de Almeida**, Relatório de Mestrado *Conektor OutSystems para Oracle E-Business Suite*, 2008.

[14]. **Flexfields**. [Online] [Citação: Março de 2010]

http://download.oracle.com/docs/cd/B25516_17/current/acrobat/115flexug.pdf

[15]. **Roger S. Pressman**, *Software Engineering: A Practitioner's Approach*, McGraw-Hill, 6^a edição, 2005. ISBN 0071238409

Glossário

1. ADO.NET - Active Data Objects .NET
2. API - Application Program Interface
3. ERP - Enterprise Resource Planning
4. FA - Fixed Assets
5. OEBS - Oracle E-Business Suite
6. PEI - Projecto de Engenharia Informática
7. SAP - Sistemas, Aplicativos e Produtos para Processamento de Dados
8. SQL - Structured Query Language
9. TI - Tecnologias de Informação
10. T-SQL - Transact Structured Query Language

Anexos

A. Sprints Backlogs

Sprint Backlog 1

Product Backlog															
Worktype	Deliverable Area	Item or Group			ID	Sprint Work Item Description			Pri	Driver	Status	Initial			
Feature	Análise de FA	Recolha de informação sobre o funcionamento do módulo FA			1	Análise e pesquisa			1	Dinis Premji	Completed	40	80		
Feature	Análise de FA	Modelo de dados (Identificar tabelas e suas relações)			2	Modelo de dados			1	Dinis Premji	Completed	40	40		
Feature	Análise de FA	Identificação de APIs (métodos ou procedimento) que permitem a execução de funcionalidades do ERP			3	Análise e pesquisa			1	Dinis Premji	Completed	40	0		

Ilustração 35 - Sprint backlog 1 (1/2)

24-Nov-2009	26-Nov-2009	27-Nov-2009	30-Nov-2009	1-Dec-2009	2-Dec-2009	3-Dec-2009	4-Dec-2009	7-Dec-2009	8-Dec-2009	9-Dec-2009	10-Dec-2009	11-Dec-2009	14-Dec-2009	15-Dec-2009	Total Effort
Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14	Day 15	
Spe Left	Spe Left	Spe Left	Spe Left	Spe Left	Spe Left	Spe Left	Spe Left	Spe Left	Spe Left	Spe Left	Spe Left	Spe Left	Spe Left	Spe Left	
8 32 InPr	8 24 InPr	8 16 InPr	8 0 Corr	0 0 Corr	0 0 Corr	0 0 Corr	0 0 Corr	0 0 Corr	0 0 Corr	0 0 Corr	0 0 Corr	0 0 Corr	0 0 Corr	0 0 Corr	40
0 40 Pen	0 40 Pen	0 40 Pen	0 40 Pen	0 40 Pen	0 40 Pen	0 40 Pen	0 40 Pen	0 40 Pen	0 40 Pen	0 40 Pen	0 40 Pen	0 40 Pen	0 40 Pen	0 40 Pen	40
0 40 Pen	0 40 Pen	0 40 Pen	0 40 Pen	0 40 Pen	0 40 Pen	0 40 Pen	0 40 Pen	0 40 Pen	0 40 Pen	0 40 Pen	0 40 Pen	0 40 Pen	0 40 Pen	0 40 Pen	40

Ilustração 36 - Sprint backlog 1 (2/2)

Sprint Backlog 2

Product Backlog															
Worktype	Deliverable Area	Item or Group			ID	Sprint Work Item Description			Pri	Driver	Status	Initial			
Feature	Carregamento de dados no módulo FA	Elaboração de um documentos único de Excel			1	Composição de dados			1	Dinis Premji	Completed	40	80		
Feature	Carregamento de dados no módulo FA	Migração dos dados da folha Excel para a tabela de interface de carregamento em massa de activos (CPRX_FA_MASSADD_JCEP)			2	Migração de dados			1	Dinis Premji	Completed	32	48		
Feature	Carregamento de dados no módulo FA	Execução do carregamento de dados			3	Actualização da base de dados Oracle			1	Dinis Premji	Completed	8	40		
Feature	Carregamento de dados no módulo FA	Pequeno teste de adição de activos (rápida/detalhada)			4	Análise			1	Dinis Premji	Completed	40	0		

Ilustração 37 - Sprint backlog 2 (1/2)

15-Dec-2009	16-Dec-2009	17-Dec-2009	18-Dec-2009	21-Dec-2009	22-Dec-2009	23-Dec-2009	24-Dec-2009	28-Dec-2009	29-Dec-2009	30-Dec-2009	31-Dec-2009	2-Jan-2010	3-Jan-2010	4-Jan-2010	Total Effort
Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14	Day 15	
Spe Left	Spe Left	Spe Left	Spe Left												
8 32 InPr	8 24 InPr	8 16 InPr	8 0 Corr	0 0 Corr	0 0 Corr	0 0 Corr	40								
0 32 Pen	0 24 InPr	8 16 InPr	8 0 Corr	0 0 Corr	0 0 Corr	0 0 Corr	32								
0 8 Pen	0 8 Pen	0 8 Pen	0 8 Pen	8											
0 40 Pen	0 40 Pen	0 40 Pen	0 40 Pen	40											

Ilustração 38 - Sprint backlog 2 (2/2)

Sprint Backlog 3

Product Backlog															
Worktype	Deliverable Area	Item or Group			ID	Sprint Work Item Description			Pri	Driver	Status	Initial			
Feature	Elaboração de Requisitos	Colecção de requisitos com os utilizadores de negócio			1	Elaboração da lista de requisitos			0	Dinis Premji	Completed	40	80		
Feature	Conector Oracle	Análise			2	Análise			3	Dinis Premji	Completed	8	(2)		
Feature	Conector Oracle	Utilização e funcionamento			3	Aplicação prática do Conector Oracle			3	Dinis Premji	Completed	8	64		
Feature	Conector Oracle	Correção de eventuais bugs do conector			4	Aplicação prática do Conector Oracle			3	Dinis Premji	Pending	16	48		
Feature	Conector Oracle	Extração de métodos que permitem a execução de funcionalidades do ERP			5	Análise e pesquisa			3	Dinis Premji	Completed	8	40		

Ilustração 39 - Sprint backlog 3 (1/2)

Ilustração 40 - Sprint backlog 3 (2/2)

Sprint Backlog 4

Product Backlog								
Worktype	Deliverable Area	Item or Group	ID	Sprint Work Item Description	Pri	Driver	Status	Initial
Feature	Listagem de activos	Erçá com a listagem de activos	1	Desenvolvimento	1	Dinis Premji	Completed	20
Feature	Listagem de activos	Construção da query	2	Desenvolvimento	1	Dinis Premji	Completed	20
Feature	Pesquisa de activos	Campo de pesquisa	3	Desenvolvimento	2	Dinis Premji	Completed	20
Feature	Pesquisa de activos	Construção da query	4	Desenvolvimento	2	Dinis Premji	Completed	40

Ilustração 41 - Sprint backlog 4 (1/2)

Ilustração 42 - Sprint backlog 4 (2/2)

Sprint Backlog 5

Product Backlog								
Worktype	Deliverable Area	Item or Group	ID	Sprint Work Item Description	Pri	Driver	Status	Initial
Feature	Consulta de activos	Ecrã de consulta de activos	1	Desenvolvimento	1	Dinis Premji	Completed	16
		Implementação de separadores para cada tipo de informação do activo	2	Desenvolvimento	1	Dinis Premji	Completed	32
Feature	Consulta de activos	Ecrã de edição de activos	3	Desenvolvimento	4	Dinis Premji	Completed	40
Feature	Edição de activos	Reutilização do ecrã de consulta de activos	4	Desenvolvimento	4	Dinis Premji	Completed	8

Ilustração 43 - Sprint backlog 5 (1/2)

Ilustração 44 - Sprint backlog 5 (2/2)

Sprint Backlog 6

Worktype	Deliverable Area	Product Backlog		ID	Sprint Work Item Description	Pri	Driver	Status	Initial
		Item or Group							
Feature	Campos flexíveis	Campo flexível localização		1	Desenvolvimento	1	Dinis Premji	Completed	32 88
Feature	Campos flexíveis	Campo flexível conta de custos		2	Desenvolvimento	1	Dinis Premji	Completed	8 80
Feature	Campos flexíveis	Campo flexível categoria		3	Desenvolvimento	1	Dinis Premji	Completed	8 72
Feature	Campos flexíveis	Campo flexível chave		4	Desenvolvimento	1	Dinis Premji	Completed	8 64
Feature	Campos flexíveis	Pop-up de lista de segmentos		5	Desenvolvimento	1	Dinis Premji	Completed	32 32
Feature	Campos flexíveis	Pop-up de lista de segmentos		6	Desenvolvimento	1	Dinis Premji	Completed	32 32

Ilustração 45 Sprint backlog 6 (1/2)

Ilustração 46 Sprint backlog 6 (2/2)

Sprint Backlog 7

Product Backlog Item or Group								
Worktype	Deliverable Area	ID	Sprint Work Item Description	Pri	Driver	Status	Initial	
Feature	Adição simples de activos	1	Modelo de dados e APIs	1	Dinis Premji	Completed	20	100
Feature	Adição detalhada de activos	2	Modelo de dados e APIs	1	Dinis Premji	Completed	20	80

Ilustração 47 - Sprint backlog 7 (1/2)



Ilustração 48 - Sprint backlog 7 (2/2)

Sprint Backlog 8

Worktype	Deliverable Area	Product Backlog		ID	Sprint Work Item Description	Pri	Driver	Status	Initial
		Item or Group							
Feature	Adição de activos em massa	Pesquisa e análise		1	Modelo de dados e APIs	1	Dinis Premji	Completed	20
Feature	Classificação de activos	Pesquisa e análise		2	Modelo de dados e APIs	1	Dinis Premji	Completed	20
Feature	Abate de activos	Pesquisa e análise		3	Modelo de dados e APIs	1	Dinis Premji	Completed	20

Ilustração 49 - Sprint backlog 8 (1/2)



Ilustração 50 - Sprint backlog 8 (2/2)

Sprint Backlog 9

		Product Backlog Item or Group	ID	Sprint Work Item Description	Pri	Driver	Status	Initial
Worktype	Deliverable Area							
Feature	Implementação das tabelas personalizadas	Análise discussão do modelo de dados	1	Análise	1	Dinis Premji	Completed	8
Feature	Implementação das tabelas personalizadas	Criação das tabelas buffer na base de dados	2	Desenvolvimento	1	Dinis Premji	Completed	8
Feature	Desenvolvimento do pacote	Desenvolvimento de métodos em PL/SQL	3	Desenvolvimento	1	Dinis Premji	Completed	104

Ilustração 51 - Sprint backlog 9 (1/2)

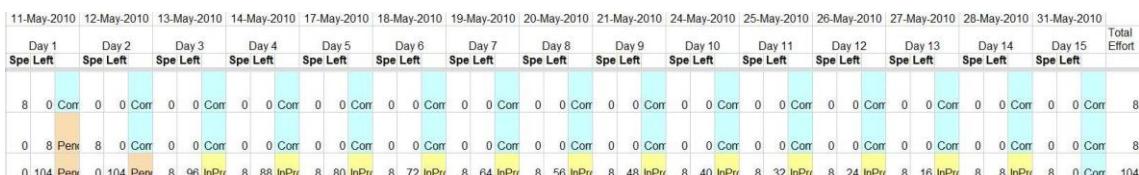


Ilustração 52 - Sprint backlog 9 (2/2)

Sprint Backlog 10

Worktype	Deliverable Area	Product Backlog												Pri	Driver	Status	Initial		
		Item or Group																	
Feature	Ecrã de adição detalhada do activo	Ecrã de adição detalhada do activo												1	Desenvolvimento	2	Dinis Premji	Completed	16 104
Feature	Ecrã de adição detalhada do activo	Implementação fluxo de adição de activo consoante os métodos do package desenvolvido												2	Desenvolvimento	2	Dinis Premji	Completed	104 0

Ilustração 53 - Sprint backlog 10 (1/2)

1-Jun-2010	2-Jun-2010	3-Jun-2010	4-Jun-2010	5-Jun-2010	6-Jun-2010	7-Jun-2010	8-Jun-2010	9-Jun-2010	10-Jun-2010	11-Jun-2010	12-Jun-2010	13-Jun-2010	14-Jun-2010	15-Jun-2010	16-Jun-2010	17-Jun-2010	18-Jun-2010	19-Jun-2010	Total Effort
Day 1 Spe Left	Day 2 Spe Left	Day 3 Spe Left	Day 4 Spe Left	Day 5 Spe Left	Day 6 Spe Left	Day 7 Spe Left	Day 8 Spe Left	Day 9 Spe Left	Day 10 Spe Left	Day 11 Spe Left	Day 12 Spe Left	Day 13 Spe Left	Day 14 Spe Left	Day 15 Spe Left	Day 16 Spe Left	Day 17 Spe Left	Day 18 Spe Left	Day 19 Spe Left	
8 8 InPr	8 0 Cor	0 0 Cor	0 0 Cor	0 0 Cor	0 0 Cor	0 0 Cor	0 0 Cor	0 0 Cor	0 0 Cor	0 0 Cor	0 0 Cor	16							
0 104 Penk	0 104 Penk	8 96 InPr	8 88 InPr	8 80 InPr	8 72 InPr	8 64 InPr	8 56 InPr	8 48 InPr	8 40 InPr	8 32 InPr	8 24 InPr	8 16 InPr	8 8 InPr	8 0 Cor	8 0 Cor	8 0 Cor	8 0 Cor	104	

Ilustração 54 - Sprint backlog 10 (2/2)

B. Product Backlog

Sprint	Componente	Tarefa	Observação	Prioridade	Estimativa (horas)	Estado
	Integração na empresa					
0		Integração na empresa		0	8	Completa
	Formação em OutSystems					
0		Formação em OutSystems Developer Course 1	Formação online constituído por webinars e testes	0	40	Completa
0		Recolha de informação sobre ERP		1	8	Completa
0		Informação prática sobre o módulo FA		1	8	Completa
0		Recolha de Informação sobre os módulos de Oracle E-Business Suite		1	8	Completa
0		Realização de tarefas em projectos OutSystems	Participação em projectos na área de OutSystems	2	40	Completa
0		Workshop Oracle E-Business Suit		1	8	Completa
0		Realização de tarefas em OutSystems	Participação em projectos na área de OutSystems	2	8	Completa
0		Formação em OutSystems Developer Course 2 1/3	Formação online constituído por webinars e testes	0	16	Completa
0		Workshop Oracle E-Business Suite		3	8	Completa
0		Formação em OutSystems Developer Course 2 2/3	Formação online constituído por webinars e testes	0	16	Completa
0		Realização de tarefas em OutSystems	Participação em projectos na área de OutSystems	2	24	Completa
0		Formação em OutSystems Developer Course 2 3/3	Formação online constituído por webinars e testes	0	8	Completa
0		Realização de tarefas em OutSystems	Participação em projectos na área de OutSystems	2	32	Completa
0		Wokshop sobre ADO.NET		3	8	Completa
	Relatório preliminar					
0		Análise		0	8	Completa
0		Redacção	Escrita do relatório preliminar sobre o projecto desenvolvido até ao momento	0	48	Completa
0		Revisão		0	24	Completa
	Análise do módulo aplicacional FA		Fixed Assets (Património) do ERP Oracle E-Business Suite			
1		Recolha de informação sobre o funcionamento do módulo FA		1	16	Completa

Ilustração 55 - Product backlog (1/5)

Sprint	Componente	Tarefa	Observação	Prioridade	Estimativa (horas)	Estado
1		Modelo de dados (Identificar tabelas e suas relações)		1	24	Completa
1		Identificação de AP/s (métodos ou procedimento) que permitam a execução de funcionalidades do ERP	Identificação das API's do EBS para serem incluídas no connector Oracle	1	40	Completa
	Carregamento de dados no módulo FA		Recepção da lista de activos em formato Excel e carregamento do módulo FA com base nessa lista	1		
2		Elaboração de um documentos único de Excel	Com base em vários ficheiros Excel e inputs do cliente, corrigir e inserir dados de forma a criar um único documento de Excel	1	40	Completa
2		Migração dos dados da folha Excel para a tabela de interface de carregamento em massa de activos (CPRX_FA_MASSADD_JCE P)	Através de um script SQL (comando SELECT seguido de INSERT)	1	32	Completa
2		Execução do carregamento de dados	Usando o package cprx_fa_load	1	8	Completa
2		Pequeno teste de adição de activos (rápida/detalhada)	Existem dois modos de adição de activo: Rápida: Criação do activo com um nível mínimo de detalhe Detalhada: Criação do activo com um elevado nível de detalhe	1	40	Completa
	Elaboração de Requisitos					
3		Colecção de requisitos com os utilizadores de negócio	Elaboração de user stories para cada requisito	0	40	Completa
3	Conektor Oracle					
3		Análise		3	8	Completa
3		Utilização e funcionamento		3	8	Completa
3		Correcção de eventuais bugs do conector		3	16	Cancelada
3		Extracção de métodos que permitem a execução de funcionalidades do ERP	Gerar através do Conektor os métodos identificados anteriormente	3	8	Completa

Ilustração 56 - Product backlog (2/5)

Sprint	Componente	Tarefa	Observação	Prioridade	Estimativa (horas)	Estado
	Listagem de activos					
4		Ecrã com a listagem de activos	Devolve uma lista de todos os activos da empresa com menu de navegação	1	40	Completa
	Pesquisa de activos					
4		Campo de pesquisa	Permite realizar a procura de um determinado activo	2	40	Completa
	Consulta de activos					
5		Ecrã de consulta de activos	Permite consultar várias características de um activo	1	16	Completa
5		Implementação de separadores para cada tipo de informação do activo		1	32	Completa
	Edição de activos					
5		Ecrã de edição de activos	Permite editar as diferentes características de um activo	4	40	Em progresso
5		Reutilização do ecrã de consulta de activos		4	8	Completa
	Campos flexíveis					
6		Campo flexível localização	Obter lista de segmentos e respectivas listas de valores	1	32	Completa
6		Campo flexível conta de custos	Obter lista de segmentos e respectivas listas de valores	1	8	Completa
6		Campo flexível categoria	Obter lista de segmentos e respectivas listas de valores	1	8	Completa
6		Campo flexível chave	Obter lista de segmentos e respectivas listas de valores	1	8	Completa
6		Pop-up de lista de segmentos	Lista dinâmica de segmentos do campo flexível	1	32	Completa
6		Pop-up de lista valores do segmento	Lista dinâmica de valores do segmento	1	32	Completa
	Adição simples de activos					
7		Pesquisa e análise	Modelo de dados e APIs	1	20	Completa
	Adição detalhada de activos					
7		Pesquisa e análise	Modelo de dados e APIs	1	20	Completa

Ilustração 57 - Product backlog (3/5)

Sprint	Componente	Tarefa	Observação	Prioridade	Estimativa (horas)	Estado
	Adição de activos em massa					
8		Pesquisa e análise	Modelo de dados e APIs	1	20	Completa
	Classificação de activos					
8		Pesquisa e análise	Modelo de dados e APIs	1	20	Completa
	Abate de activos					
8		Pesquisa e análise	Modelo de dados e APIs	1	20	Completa
	Implementação das tabelas personalizadas					
9		Análise discussão do modelo de dados	Modelo de dados da tabelas buffer	1	8	Completa
9		Criação das tabelas buffer na base de dados	Permite guardar todos os dados antes da invocação do método de adição do activo da API do OEBS	1	8	Completa
	Desenvolvimento do package					
9		Desenvolvimento de métodos em PL/SQL	Implementação dos métodos para popular as tabelas buffer e invocar o método de adição do activo da API do OEBS	1	104	Completa
	Adição simples do activo					
10		Ecrã de adição simples do activo	Permite recolher um conjunto de dados de um determinado activo	1	16	Completa
10		Implementação fluxo de adição de activo consoante os métodos do package desenvolvido	Permite adicionar um activo na base dados Oracle através dos dados fornecidos do ecrã de adição	1	104	Completa
	Relatório final			0		
		Recolha de dados		0	16	Completa
		Redacção	Escrita do relatório final sobre o projecto desenvolvido	0	144	Completa
		Revisão		0	80	Completa
	Adição detalhada do activo					
11		Ecrã de adição detalhada do activo	Permite recolher um conjunto de dados de um determinado activo	2	16	Em progresso

Ilustração 58 - Product backlog (4/5)

Sprint	Componente	Tarefa	Observação	Prioridade	Estimativa (horas)	Estado
11		Implementação fluxo de adição de activo consoante os métodos do package desenvolvido	Permite adicionar um activo na base de dados Oracle através dos dados fornecidos do ecrã de adição	2	104	Pendente
	Adição em Massa					
12		Ecrã de adição em massa dos activos		3	16	Pendente
12		Implementação fluxo de adição em massa dos activos		3	104	Pendente
	Classificação dos activos					
13		Ecrã de classificação dos activos		3	16	Pendente
13		Implementar a lógica de classificação		3	104	Pendente
	Abate do activo					
14		Ecrã de abate do activo		2	16	Pendente
14		Implementação fluxo do abate de um activo		2	104	Pendente

Ilustração 59 - Product backlog (5/5)

C. Tabela CPRX_FA_MASSADD_TDP

Coluna	Tipo de valor	Descrição
NUMERO_LINHA	NUMBER	Número de identificação única da linha.
	VARCHAR2	Número do activo
ASSET_NUMBER	(15 Byte)	
	VARCHAR2	Número da etiqueta
TAG_NUMBER	(15 Byte)	
DATE_PLACED_IN_SE		Data de início de utilização
RVICE	DATE	
	VARCHAR2	Livro do activo
BOOK_TYPE_CODE	(15 Byte)	
	VARCHAR2	Modelo legal da categoria do activo
CAT_MOD_LEGAL	(15 Byte)	
	VARCHAR2	Classe do imobilizado da categoria do activo
CAT_CATEG	(15 Byte)	
	VARCHAR2	Bem imobiliario da categoria do activo
CAT_SUBCATEG	(15 Byte)	

Coluna	Tipo de valor	Descrição
AKEY_METAQUI	VARCHAR2 (15 Byte)	Método da chave do activo
AKEY_ESTADO	VARCHAR2 (15 Byte)	Estado da chave do activo
AKEY_LEI_REAV	VARCHAR2 (15 Byte)	Lei da reavaliação da chave do activo
DESCRIPTION	VARCHAR2 (200 Byte)	Descrição do activo
FIXED_ASSETS_UNITS	NUMBER	Número de unidades
DEPRECiate_FLAG	VARCHAR2 (3 Byte)	<i>Flag</i> da depreciação
PAYABLES_COST	NUMBER	Custo do activo
FIXED_ASSETS_COST	NUMBER	Custo fixo do activo
UNREVALUED_COST	NUMBER	Valor da reavaliação
DEPRN_RESERVE	NUMBER	Valor depreciação
VIDA_UTIL	NUMBER	Vida útil do activo
CTAQUI_ORG	VARCHAR2 (15 Byte)	Organização do activo na aquisição
CTAQUI_GO	VARCHAR2 (15 Byte)	Grupo orçamental do activo na aquisição
CTAQUI_CR	VARCHAR2 (15 Byte)	Centro de Responsabilidade do activo na aquisição
CTAQUI_ACCAO	VARCHAR2 (15 Byte)	Acção do activo na aquisição
CTAQUI_POCP	VARCHAR2 (15 Byte)	Plano Oficial de Contabilidade Pública do activo na aquisição
CTAQUI_ORC	VARCHAR2 (15 Byte)	Mercado do activo na aquisição
CTAQUI_FF	VARCHAR2 (15 Byte)	Fontes de Financiamento do activo na aquisição
LOC_PAIS	VARCHAR2	Local do activo

Coluna	Tipo de valor	Descrição
	(15 Byte)	
LOC_LOCAL	VARCHAR2 (15 Byte)	Sub local do activo
LOC_ANDAR	VARCHAR2 (15 Byte)	Piso/edificio do activo
LOC_SALA	VARCHAR2 (15 Byte)	Ala/sala/departamento do activo
NEW_USED	VARCHAR2 (4 Byte)	Activo Novo ou Usado.
PROPERTY_TYPE_COD	VARCHAR2 E (10 Byte)	Propriedade do activo
ASSIGNED_TO	NUMBER	Activo atribuído
INVOICE_NUMBER	VARCHAR2 (50 Byte)	Número da factura
MANUFACTURER_NA	VARCHAR2 ME (30 Byte)	Nome do fornecedor
MODEL_NUMBER	VARCHAR2 (40 Byte)	Número do modelo
SERIAL_NUMBER	VARCHAR2 (35 Byte)	Número de série
NUM_FORN	VARCHAR2 (15 Byte)	Número de fornecedor
REVIEWER_COMMEN	VARCHAR2 TS (60 Byte)	Comentários
ATTRIBUTE11	VARCHAR2 (30 Byte)	Comprimento do activo
ATTRIBUTE12	VARCHAR2 (30 Byte)	Altura do activo
ATTRIBUTE13	VARCHAR2 (30 Byte)	Largura do activo
ATTRIBUTE14	VARCHAR2 (30 Byte)	Outra dimensão de medida

Coluna	Tipo de valor	Descrição
	VARCHAR2	Informação sobre abate do activo
INFO_ABATE	(1000 Byte)	

Tabela 8 - Tabela CPRX_FA_MASSADD_TDP

D. Tabela das funções e *queries* dos testes unitários

Funções e <i>queries</i>	Descrição
<i>FlexField_GetTitle</i>	Devolve o nome do campo flexível
<i>GetCategoryId</i>	Devolve o número de identificação de uma determinada categoria de um activo.
<i>GetCodeCombinationId</i>	Devolve o número de identificação de uma determinada conta de custos de um activo
<i>GetKeyId</i>	Devolve o número de identificação de uma determinada chave de um activo
<i>GetLocationId</i>	Devolve o número de identificação de uma determinada localização de um activo
<i>GetAssetEdit</i>	Devolve um conjunto de informação essencial de um determinado activo
<i>GetBooksEdit</i>	Devolve informação relativamente ao livro de um determinado activo
<i>GetAttributionsEdit</i>	Devolve informação relativamente aos encargos de um determinado activo
<i>GetFinancialEdit</i>	Devolve informação financeira de um determinado activo
<i>GetOriginLineEdi</i>	Devolve informação relativamente à origem de um determinado activo
<i>GetLocal</i>	Devolve o nome de todos os locais disponíveis no módulo FA do OEBS
<i>GetLegalModel</i>	Devolve o nome de todos os modelos legais disponíveis no módulo FA do OEBS
<i>GetSubLocation</i>	Devolve todos os sublocais dado o local
<i>GetBuilding</i>	Devolve todos os edifícios dado local e o

Funções e <i>queries</i>	Descrição
	sublocal
<i>GetDepartment</i>	Devolve todos os departamentos dado local, sublocal e o edifício
<i>GetClassTypeAsset</i>	Devolve todas as classes de um activo dado o modelo legal
<i>GetAssetImob</i>	Devolve todos os tipos de imobilizado de um activo dado o modelo legal e a classe do activo
<i>GetAssetsList</i>	Devolve todos os activos de acordo com os valores de filtro inseridos.
<i>GetFlexFields</i>	Devolve os campos flexíveis
<i>GetSegments</i>	Devolve os valores de um determinado campo flexível

Tabela 9 - Tabela das funções e *queries* dos testes unitários

