



Universidade do Porto
Faculdade de Engenharia

FEUP

Cristina Falcão Sousa Calheiros Santos

**Metodologias de Desenvolvimento e
Normalização Gráfica do Human Capital Team
Tool na IBS Portugal**
Relatório do Estágio Curricular da LEIC 2003/2004

Faculdade de Engenharia da Universidade do Porto
Licenciatura em Engenharia Informática e Computação



**Metodologias de Desenvolvimento e Normalização Gráfica do
Human Capital Team Tool na
IBS Portugal**

Relatório do Estágio Curricular da LEIC 2003 / 2004

EIC 5202

Cristina Falcão Sousa Calheiros Santos

Orientador na FEUP: Prof. Ademar Aguiar
Orientador na IBS Portugal: Engenheiro Gonçalo Mata

Agosto de 2004

Universidade do Porto
Faculdade de Engenharia
Biblioteca
Nº
CDU
Data

004647.3) LEICEICS2022004/SAMC Vol. 1

Universidade do Porto Faculdade de Engenharia Biblioteca	1
Nº	81517
CDU	
Data	91/03/2006

Resumo

O estágio “Metodologias de Desenvolvimento e Normalização Gráfica do Human Capital Team Tool” realizou-se nas instalações da IBS Portugal, em Vila Nova de Gaia, e debruçou-se sobre a aplicação *Human Capital Team Tool* (HCTT), orientada para a gestão de Recursos Humanos, assente na tecnologia *Java 2 Platform Enterprise Edition*, a qual se encontra em desenvolvimento no Departamento de Java daquela instituição.

Foram três as fases que constituíram o estágio. Uma primeira fase foi dedicada à elaboração do *MethoDoc*, uma espécie de Manual de Qualidade contendo todo o *know-how* da equipa de desenvolvimento e as regras para o desenvolvimento do HCTT, uma tarefa que exigiu uma rápida integração na equipa e nos processos do Departamento de forma a ser levada a cabo com sucesso.

A segunda fase do estágio consistiu no desenvolvimento de uma nova funcionalidade do HCTT denominada Pesquisa por Palavras-Chave. Esta tarefa obrigou um grande envolvimento e exploração da arquitectura da aplicação e do código-fonte.

Por último, a terceira e mais longa fase de estágio consistiu na Normalização Gráfica do HCTT e implicou, basicamente, a implementação de processos XML e classes Java “genéricas” para a camada de *Adapters* com o intuito de proporcionar um constante reaproveitamento de código para as listagens de Responsáveis e de Colaboradores Activos do HCTT. Ainda nesta fase foi feita uma alteração a oito páginas críticas do HCTT de modo a estas apresentarem uma divisão em *tabs* (separadores) da informação originalmente lá contida, aumentando assim a usabilidade da aplicação.

O estágio foi concluído com sucesso e permitiu um primeiro contacto com a realidade do desenvolvimento de *software* a um nível industrial, bem diferente do nível académico.

Agradecimentos

Gostaria de agradecer ao meu orientador na IBS Portugal, Eng. Gonçalo Mata, bem como à Eng. Bárbara Costa, pela disponibilidade, envolvimento e atenção demonstrada durante o estágio.

Os meus agradecimentos vão também para o meu colega Carlos Rebelo, com quem trabalhei directamente em parte do estágio, bem como para os meus restantes colegas do Departamento de Java que sempre se mostraram disponíveis para ajudar e apoiar e proporcionaram um excelente ambiente de trabalho.

Agradeço também ao meu orientador na FEUP, Prof. Ademar Aguiar, pela constante disponibilidade e interesse no trabalho desenvolvido ao longo do estágio, tendo fornecido um excelente apoio.

Índice de Conteúdos

1	Introdução	1
1.1	Instituição de Estágio IBS Portugal	1
1.2	Departamento de Java	1
1.3	Apresentação do HCTT	1
1.4	Estrutura do Relatório	1
2	O Projecto HCTT	3
2.1	Apresentação Funcional	3
2.2	Apresentação Técnica – Arquitectura Lógica do HCTT	7
2.3	As Tecnologias Utilizadas no Desenvolvimento do HCTT	9
	Framework UnifFace	10
	Framework Persistence Builder	11
	Eclipse	12
	Tomcat	12
	IBM WebSphere	12
	CVS	13
	Reflection em Java	13
2.4	O Processo de Desenvolvimento do HCTT	14
2.5	O Trabalho de Estágio no Projecto HCTT	14
	Planeamento	16
3	O MethoDoc	19
3.1	Análise do Problema do MethoDoc	19
3.2	Descrição da Solução Proposta para o MethoDoc	20
3.3	Estrutura do MethoDoc	21
3.4	Conteúdo do MethoDoc	22
3.5	Principais Dificuldades Encontradas	22
3.6	Avaliação de Resultados e Conclusões	23
4	A Pesquisa por Palavras-Chave	24
4.1	Análise do Problema da Pesquisa por Palavras-Chave	24
4.2	Descrição da Solução para a Pesquisa por Palavras-Chave	25
4.3	Implementação da Pesquisa por Palavras-Chave	29
	Base de dados	29
	Camada de Persistência	31
	Controllers, Commands e Adapters	32
4.4	Principais Dificuldades Encontradas	34
4.5	Avaliação de Resultados e Conclusões	35
5	A Normalização Gráfica	37
5.1	Análise do Problema da Normalização Gráfica	37
5.2	Descrição da Solução para a Normalização Gráfica	38
5.3	Implementação da Normalização Gráfica	39
	Listagens de Responsáveis	39
	Listagens de Colaboradores Activos	41
	Páginas divididas em <i>tabs</i> (separadores) categorizados	43

5.4 Principais Dificuldades Encontradas.....	44
5.5 Avaliação de Resultados e Conclusões.....	44
6 Análise ao Planeamento de Estágio.....	46
6.1 Desvios Verificados ao Planeamento Inicial.....	46
6.2 Estatísticas.....	48
7 Conclusões e Perspectivas de Trabalho Futuro.....	50
Referências e Bibliografia.....	53
Glossário.....	54
ANEXO A: Resumo em Inglês.....	55
ANEXO B: Código-fonte do Processo HCTTOwners.xml.....	56
ANEXO C: Código-fonte da Classe do Adapter OwnersAdapter.java.....	62
ANEXO D: Código-fonte do Processo HCTTPersonsLists.xml.....	66
ANEXO E: Logbook de Estágio.....	71

Índice de Figuras

Figura 1 - Áreas Funcionais do HCTT	4
Figura 2 - Diagrama de Arquitectura Lógica do HCTT.....	8
Figura 3 – Fases do Estágio	15
Figura 4 - Cronograma do Planeamento Inicial do Estágio.....	18
Figura 5 - Ciclo de Evolução do MethoDoc.....	20
Figura 6 - Diagrama de Casos de Uso da Funcionalidade de Pesquisa por Palavras-Chave.....	25
Figura 7 - Página de Manutenção de Palavras-Chave Associadas a Colaboradores / Candidatos.....	26
Figura 8 - Página de Manutenção de Palavras-Chave Associadas a um Processo de Selecção.....	27
Figura 9 - Página do Formulário de Pesquisa num Processo de Selecção.....	28
Figura 10 - Página do Formulário de Pesquisa num Meio de Recrutamento	28
Figura 11 - Modelo relacional parcial da base de dados para a Pesquisa por Palavras-Chave	30
Figura 12 - Modelo de Classes do Domínio para a Pesquisa por Palavras-Chave.....	31
Figura 13 - Diagrama de Colaboração para a Pesquisa num Processo de Selecção.....	32
Figura 14 - Diagrama de Colaboração para a Pesquisa num Meio de Recrutamento	33
Figura 15 - Diagrama de Actividades do Algoritmo Utilizado na Pesquisa por Palavras-Chave.....	34
Figura 16 – Quadro-Resumo dos Aspectos Técnicos e Funcionais das Listagens de Responsáveis Antes das Alterações.....	40
Figura 17 – Quadro-Resumo dos Aspectos Técnicos e Funcionais das Listagens de Responsáveis Depois das Alterações.....	41
Figura 18 – Quadro-Resumo dos Aspectos Técnicos e Funcionais das Listagens de Colaboradores Activos Antes das Alterações	42
Figura 19 – Quadro-Resumo dos Aspectos Técnicos e Funcionais das Listagens de Colaboradores Activos depois das Alterações	43
Figura 20 – “Antes e Depois” da Página de Detalhes de uma Estrutura no Módulo de Base- Estrutura.....	44
Figura 21 – Cronograma do Planeamento Inicial de Estágio [repetição].....	46
Figura 22 – Cronograma dos Tempos Efectivamente Gastos Durante o Estágio.....	47

1 Introdução

O presente capítulo pretende apresentar de uma forma breve a instituição de estágio IBS Portugal, em particular o Departamento de Java, no qual o estágio decorreu, e a aplicação HCTT a qual foi objecto de estágio.

1.1 Instituição de Estágio IBS Portugal

A IBS Portugal é subsidiária da IBS AB, grupo com sede em Estocolmo (Suécia), um dos dez maiores fornecedores mundiais de soluções de gestão empresarial. A IBS AB foca a sua actividade no fornecimento de soluções aplicacionais que cobrem áreas desde a *Manufatura*, *Supply Chain Management*, *Finanças*, *e-Business* e *Customer Relationship Management*.

Na reorganização das Áreas de Negócio do Grupo IBS, realizada em Julho de 2002, foi criada uma nova área, designada Área Latina, que passou a reunir Portugal, Espanha, Brasil e Itália. [1]

A IBS Portugal possui dois escritórios em Portugal: um escritório em Vila Nova de Gaia, que conta com cerca de 113 colaboradores, e um escritório em Lisboa com cerca de 70 colaboradores.

O estágio aqui apresentado decorreu nas instalações de Vila Nova de Gaia, no Departamento de Java, e esteve integrado num projecto deste Departamento: a aplicação HCTT.

1.2 Departamento de Java

O Departamento de Java da IBS Portugal tem como objectivos a investigação e desenvolvimento em linguagem Java e é responsável pela coordenação e implementação de projectos nesta tecnologia, como é o caso do HCTT. A equipa deste Departamento é constituída, à data, por dez elementos nos quais estou incluída. Toda a equipa do Departamento de Java reúne-se semanalmente à segunda-feira naquilo que é denominada internamente como reunião *flash*.

1.3 Apresentação do HCTT

O HCTT é uma aplicação orientada para a Gestão de Recursos Humanos, que se encontra em desenvolvimento no Departamento de Java da IBS Portugal e sobre a qual se debruçou todo o trabalho de estágio. Devido à dimensão desta aplicação e à importância de uma apresentação detalhada para um bom entendimento do trabalho de estágio efectuado, foi-lhe dedicado o capítulo 2 deste relatório, intitulado “O Projecto HCTT”.

1.4 Estrutura do Relatório

O capítulo 2 “O Projecto HCTT” pretende apresentar em detalhe a aplicação HCTT e as tecnologias utilizadas no seu desenvolvimento, bem como o trabalho de estágio previsto.

O capítulo 3 “O MethoDoc”, o capítulo 4 “A Pesquisa por Palavras-Chave” e o capítulo 5 “A Normalização Gráfica” têm como objectivos a descrição detalhada do trabalho de estágio efectuado durante as várias fases que o constituíram.

O capítulo 6 “Análise ao Planeamento” descreve os desvios verificados em relação ao planeamento inicial e apresenta algumas estatísticas sobre os tempos de estágio.

O último capítulo resume as principais conclusões do estágio e refere as principais perspectivas de evolução do trabalho realizado durante o estágio.

Existem oito anexos a este relatório, nomeadamente o anexo A “Resumo em Inglês”, anexo B “Código-fonte do Processo HCTTOwners.xml”, anexo C “Código-fonte da Classe do Adapter OwnersAdapter.java”, anexo D “Código-fonte do Processo HCTTOwnersLists.xml”, anexo E “Logbook de Estágio”, anexo F “MethoDoc”, anexo G “Especificação da Pesquisa por Palavras-Chave” e anexo H “Análise Técnica e Funcional das Páginas com Listagens de Pessoas”.

Os anexos F, G e H encontram-se em volumes aparte do presente documento de Relatório de Estágio.

2 O Projecto HCTT

O HCTT é uma aplicação orientada para a gestão de Recursos Humanos. Tradicionalmente, as soluções existentes dirigiam-se para a área de Gestão de Pessoal e Vencimentos, centrando-se no processamento de entidades jurídicas, respondendo aos aspectos legais, numa dinâmica reactiva, funcionando apenas como ferramentas do Departamento de Pessoal.

As mutações constantes exigem uma nova dinâmica para a gestão dos Recursos Humanos – o HCTT implementa essa dinâmica de uma forma inovadora.

Baseado na tecnologia J2EE¹ Standard, são alcançadas características como multi-empresa, multi-plataforma e multi-idioma. [2]

Segue-se uma apresentação funcional e técnica do HCTT, bem como das tecnologias utilizadas no seu desenvolvimento.

2.1 Apresentação Funcional

Funcionalmente, o HCTT pode ser dividido em três grandes áreas (ilustradas na Figura 1):

- Informação do Núcleo
- Módulos Funcionais
- Ferramentas Principais

Informação do Núcleo

Esta área é constituída pelos módulos de Colaboradores e de Estrutura.

Colaboradores

O Colaborador é um dos principais elementos (núcleo) da aplicação. Nesta área está disponível um conjunto muito completo de informação pessoal e profissional sobre um Colaborador, bem como campos definidos pelo utilizador. A aplicação consegue associar a cada Colaborador uma lista de ficheiros externos (tal como um “curriculum_vitae.doc” ou um “foto.jpg”) que são carregados para o servidor.

Cada recurso humano tem um Perfil correspondente. Este Perfil é um conjunto de competências niveladas possuídas pelo Colaborador. [2]

Estrutura

Este elemento representa graficamente uma hierarquia organizacional contendo Estruturas, Nós, Funções e Posições, de uma forma muito simples e com uma interface amigável. As Posições podem estar livres ou preenchidas por um Colaborador.

Cada Função tem um Perfil correspondente. Este Perfil é um conjunto de competências niveladas que um Colaborador deve possuir para desempenhar aquela Função.

¹ J2EE – Java 2 Platform Enterprise Edition

Pode estar montada mais do que uma Estrutura, o que permite vistas paralelas e complementares da organização. [2]

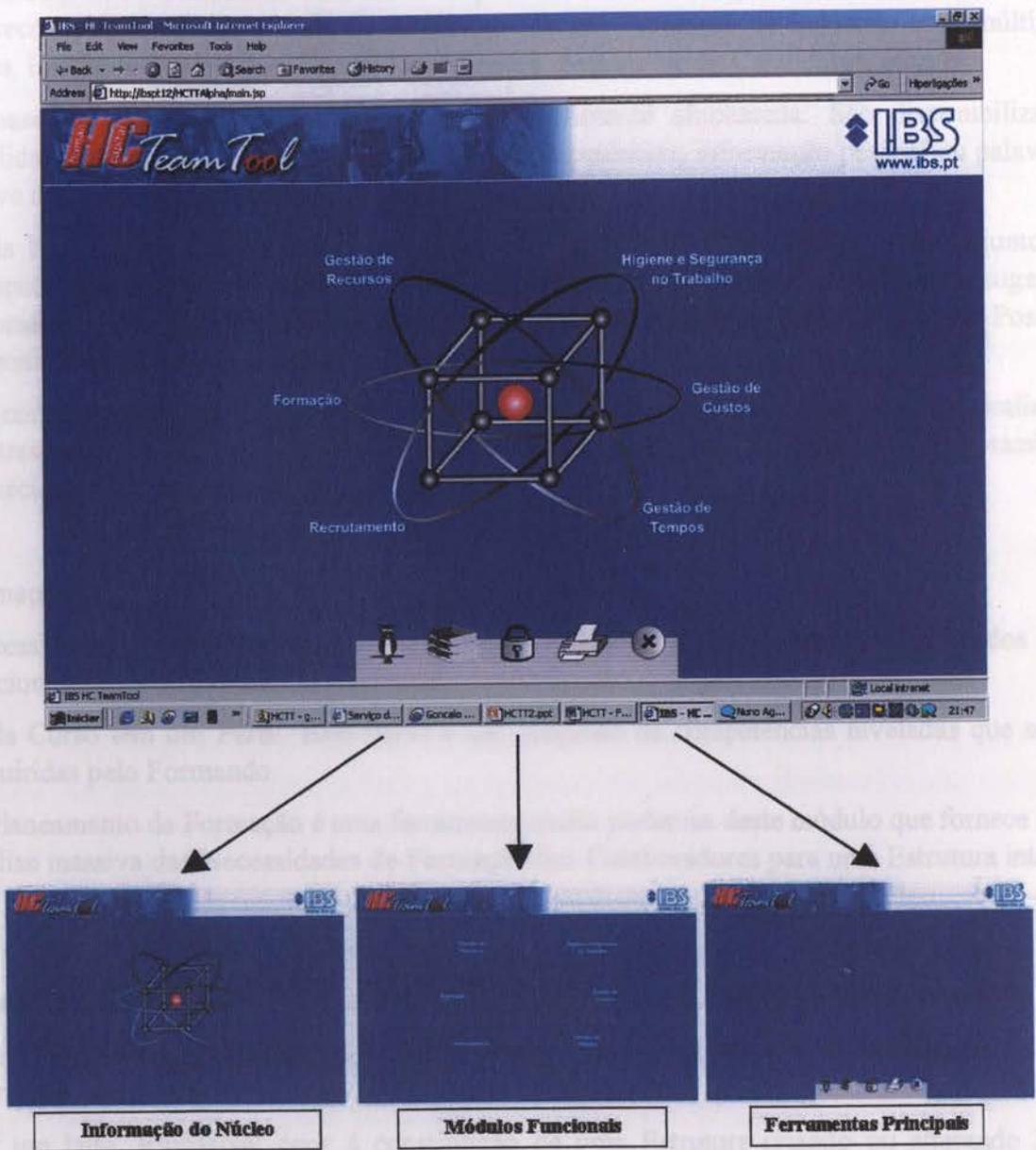


Figura 1 - Áreas Funcionais do HCTT

Módulos Funcionais

Deste conjunto de módulos fazem parte os módulos de Recrutamento, de Formação, de Gestão de Recursos, de Segurança e Higiene no Trabalho, de Gestão de Custos e de Gestão de Tempos.

Recrutamento

O módulo de Recrutamento trata todos os processos de informação envolvidos na actividade de recrutamento, desde Canais de Recrutamento até Processos de Selecção com múltiplas fases, incluindo Entrevistas, Testes, Documentos, Avaliações de Candidatos, etc.

A base de dados de Candidatos pode ser facilmente alimentada. São disponibilizadas facilidades de pesquisa baseadas em Perfis de competências, informação pessoal ou palavras-chave definidas pelo utilizador.

Cada Processo de Selecção tem um Perfil correspondente. Cada Perfil é um conjunto de competências niveladas desejáveis num Candidato a ser seleccionado. É feita uma sugestão automática para este Perfil pela associação de um Processo de Selecção a uma Posição disponível na Estrutura, baseada no Perfil da Função.

As configurações de um Processo de Selecção incluem o número de itens de avaliação (Entrevistas, Testes, Documentos) envolvidos em cada fase do processo. São também fornecidos *templates* para estas configurações. [2]

Formação

Necessidades de Formação, Cursos, Grupos, Sessões e Inscrições de Formandos são funcionalidades do módulo de Formação.

Cada Curso tem um Perfil. Este Perfil é um conjunto de competências niveladas que serão adquiridas pelo Formando.

O Planeamento da Formação é uma ferramenta muito poderosa deste módulo que fornece uma análise massiva das Necessidades de Formação dos Colaboradores para uma Estrutura inteira. [2]

Gestão de Recursos

Neste módulo cada utilizador pode operar sobre a sua área de responsabilidade dentro de uma Estrutura.

Por um lado, é possível gerir a constituição de uma Estrutura criando ou alterando Nós, Funções ou Posições, actualizando os seus respectivos atributos ou alterando as responsabilidades definidas para esses elementos.

Por outro lado, o utilizador pode gerir o Colaborador do qual é superior. Pode executar alocações simples a Posições envolvendo os Colaboradores disponíveis para tal ou executar processos mais complexos, eventualmente com impactos contratuais, tais como Processos de Admissão, Processos de Transferência ou Processos de Demissão.

As Sucessões e as Substituições são planeadas aqui e os Perfis dos Colaboradores são actualizados também neste módulo, através de Avaliações. [2]

Higiene e Segurança no Trabalho

O módulo de Higiene e Segurança no Trabalho fornece uma ferramenta eficiente para gerir a informação desta área.

A estrutura geográfica da organização é a base para a área de Segurança. Os Nós representam divisões físicas, com Locais de Trabalho alocados. Em cada Local de Trabalho pode trabalhar um ou mais Colaboradores.

Cada Nó, bem como cada Local de Trabalho, tem Regulações de Segurança hierarquicamente estruturadas.

Estão também contidos na mesma Estrutura os Perigos e Riscos e pode ser fornecido um Plano de Acção para cada elemento da árvore, com Acções de diferentes tipos agendadas, tais como prevenção, formação, fiscalização, manutenção, etc.

É também incluído neste módulo o reportar e tipificar de acidentes, bem como de danos e doenças.

Os dados clínicos, a vacinação, os exames médicos e a informação sobre consultas médicas fazem parte da área de Higiene. [2]

Gestão de Custos

Este módulo ainda não está disponível no HCTT. Os exemplos de tópicos para este módulo são orçamentos, simulações e análises de custos. [2]

Gestão de Tempos

Este módulo também ainda não está disponível no HCTT. Exemplos de tópicos para este módulo são planos de férias, horários, absentismo, planeamento de trabalho em equipa e agendamento de recursos. [2]

Ferramentas Principais

Como Ferramentas Principais existem as ferramentas de Perfil, de Templates, de Segurança, de Cesto e de Relatórios.

Perfil

O Perfil está presente em quase todas as áreas da aplicação. Fornece uma interoperabilidade transversal ao longo de diferentes áreas funcionais, cria dinamismo e facilita o fluxo de trabalho através de um objecto *standard* que adquire um significado particular, dependendo do contexto em que é utilizado, isto é, dependendo da entidade que o possui.

Os Perfis são constituídos por três diferentes listas de competências: Perfil Técnico, Perfil Psicológico e Perfil Salarial. [2]

Templates

Os *templates* são outro elemento-chave para a dinâmica da aplicação, concentrando todo o conhecimento tanto externo como interno. Esta ferramenta permite acelerar o processo de criação e manutenção de várias entidades.

Existem, então, templates para várias entidades: Perfil, Estrutura, configuração de Processos de Selecção, para nomear alguns. Qualquer uma destas entidades tem a possibilidade de

importar um *template*, ou seja, aplicar um *template* predefinido, e também de se exportarem a si mesmas, sendo guardadas como um *template*. [2]

Segurança

Além da autenticação, a aplicação fornece uma ferramenta eficiente para definir os privilégios de utilização para cada funcionalidade. O conceito de Responsável por uma entidade, para certas entidades, torna possível prevenir que os utilizadores tenham acesso à informação de outros. Isto garante que, mesmo que um certo utilizador tenha permissão para usufruir de uma certa funcionalidade, ele só é capaz de o fazer sobre as entidades pelas quais é responsável. [2]

Cesto

Cada Colaborador tem um Cesto. Um Cesto é uma ferramenta de fluxo de trabalho capaz de receber, enviar e executar Tarefas, com a aparência de uma “*to do list*”. Suporta um sistema de automação na execução de tarefas, lançado por várias funções na aplicação. Também apresenta ao utilizador uma interface simples e amigável. [2]

Relatórios

Cada módulo fornece vários relatórios das suas áreas funcionais. Estes relatórios podem ser gerados em formatos diferentes (PDF, Excel, HTML, XML, CSV,...) e impressos através da respectiva aplicação. [2]

2.2 Apresentação Técnica – Arquitectura Lógica do HCTT

O HCTT é uma aplicação assente na tecnologia J2EE.

Com uma arquitectura cliente-servidor, orientado para a *web*, é multi-plataforma e a sua lógica de negócio é implementada 100% em Java. A interface gráfica está a cargo da *framework* IBS *UniFace* e o acesso à base de dados é providenciado via tecnologia JDBC¹. [2]

¹ JDBC – *Java Database Connectivity*

A arquitectura lógica do HCTT é uma arquitectura típica em três camadas (interface, lógica de negócio e persistência) e está ilustrada na Figura 2:

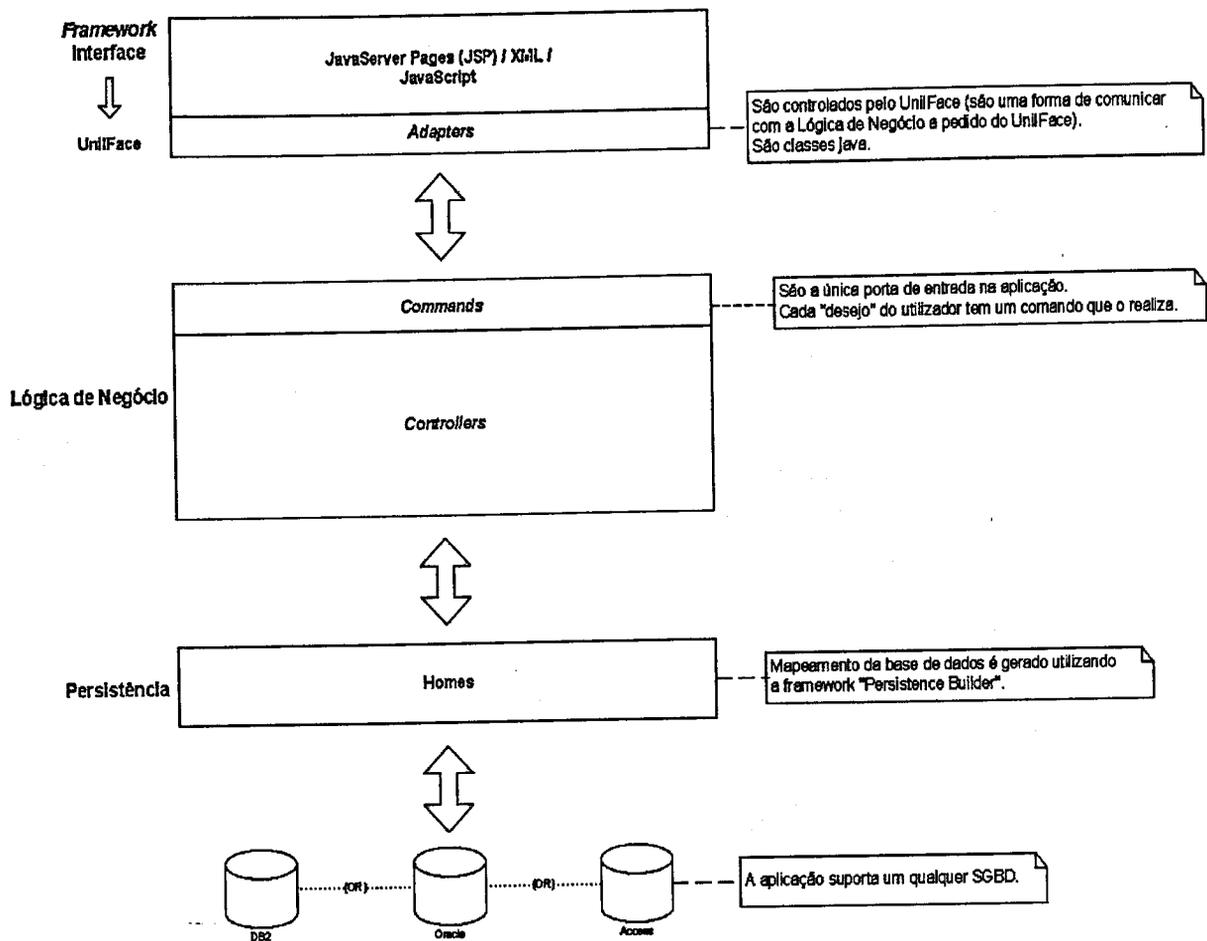


Figura 2 - Diagrama de Arquitectura Lógica do HCTT

Camada de Interface

Esta camada é gerada a partir de uma *framework* da IBS denominada *UniFace* que gera JSPs¹ a partir de processos XML². Um processo XML pode gerar uma ou mais páginas *web* e cada página e respectivas acções são suportadas pela sub-camada dos *Adapters*.

As classes de *Adapters* são implementadas pelos programadores e obedecem às interfaces impostas pelo *UniFace* quer a nível dos métodos que implementam, quer a nível dos objectos que recebem e devem devolver. Sendo esta sub-camada centralizada, um *Adapter* pode e deve servir várias páginas da camada de interface, comunicando com a sub-camada de *Commands* da lógica de negócio de modo a levar a cabo os pedidos da interface.

¹ JSP – *JavaServer Pages*

² XML – *Extended Markup Language*

Camada de Lógica de Negócio

Os *Commands* formam uma sub-camada descentralizada que capta não só as acções dos utilizadores, mas também os seus contextos. Por exemplo, a mesma acção executada em páginas diferentes irá corresponder a dois *Commands* e não a um só, pois a acção é a mesma mas os contextos não são. Uma das aplicações desta camada é a segurança, permitindo controlar em detalhe todas as interacções a que os utilizadores têm acesso.

A sub-camada de *Controllers* deve ser centralizada e orientada aos *Business Objects*, isto é, os *Controllers* que são implementados tendencialmente representam um *Business Object* e lidam com toda a lógica de negócio relativa. Para aceder à base de dados, os *Controllers* recorrem às classes da camada de persistência. *Business Objects* são tipicamente os objectos no domínio da aplicação que se querem persistentes num armazém de dados.

Camada de Persistência

Esta camada é constituída pelas classes geradas pela *framework Persistence Builder* do IDE¹ *Visual Age*. Encarrega-se de gerir todas as interacções com a base de dados. É uma camada complexa à qual pertencem os *Business Objects* que são utilizados nas restantes camadas.

Uma vez apresentada a arquitectura do HCTT, salienta-se de imediato que, de todas as camadas e respectivas sub-camadas apresentadas, aquelas onde o programador tem que tomar decisões com impacto sobre a arquitectura são as de *Adapters*, *Commands* e *Controllers*, pois é o programador que implementa as classes que constituem cada uma destas sub-camadas. A nível da interface, o programador só é responsável por construir os processos XML que irão gerar, via a *framework UniFace*, os ficheiros JSP. Ao nível da camada de persistência, todas as suas classes são geradas através da *framework Persistence Builder*.

2.3 As Tecnologias Utilizadas no Desenvolvimento do HCTT

Nesta secção pretende-se fazer uma breve descrição das tecnologias utilizadas no desenvolvimento do HCTT, nomeadamente:

- *Framework UniFace*
- *Framework Persistence Builder*
- *IDE Eclipse*
- *Tomcat*
- *IBM WebSphere*
- *CVS*
- *Reflection* em Java

¹ IDE – *Integrated Development Environment*

Framework UniFace



Esta *framework* permite a geração automática de JSPs a partir de processos XML.

O *UniFace* pode dividir-se em duas partes distintas: o *Client Framework* e a ferramenta geradora de JSPs *GuiGen*.

No que diz respeito à *Client Framework*, a IBS decidiu ter uma arquitectura para as interfaces do utilizador que pudesse ser aplicável a todo o seu portfólio de produtos. Essa arquitectura, a *Client Architecture*, é desenhada para construir aplicações clientes usando o protocolo HTTP¹.

A meta da *Client Framework* é disponibilizar uma infra-estrutura com as seguintes funcionalidades:

- Fornecer uma interface com o utilizador uniforme a todos os produtos de *software* da IBS.
- Ser capaz de tratar todos os tipos de fluxo de ecrã que os especialistas de domínio requerem.
- Ser eficiente em termos de quantidade de trabalho que um programador tem para criar uma interface com o utilizador para a sua funcionalidade. O trabalho que ele tem que fazer deve ser simples e rápido.
- Ser distribuído sob o Servidor Aplicacional *IBM WebSphere*, que traz os benefícios do J2EE (alta disponibilidade, segurança, escalabilidade, fiabilidade).
- Ser distribuído pela *web*, e conseqüentemente, as máquinas clientes não precisarem de nada mais além de um *browser* capaz de correr *JavaScript*. [3]

Além da *Client Framework*, o *UniFace* é ainda constituído pela ferramenta *GuiGen*. A IBS também decidiu construir uma biblioteca de *tags* por cima da *GUI² Framework* existente. Esta biblioteca de *tags* chama-se *Free Gui Format (FGF)* e foi desenvolvida de forma a ser compatível com *JavaServer Pages (JSP) Specification Version 1.1* e a *Servlet API Specification Version 2.2*. Para usar *tags* FGF, cada página definida num processo XML deve ter o seu próprio ficheiro JSP. Consumiria muito tempo e seria muito custoso os programadores desenvolverem esses ficheiros de raiz, porque o número de JSPs a implementar seria na ordem dos milhares. Devido a isso, a IBS construiu uma ferramenta chamada *GuiGen* para gerar o máximo possível de código-fonte de JSPs. Actualmente, devido às necessidades da IBS, o *GuiGen* é capaz de gerar 100% do código-fonte. [4]

A meta do *GuiGen* é disponibilizar uma infra-estrutura com as seguintes funcionalidades:

- Fornecer a possibilidade de gerar o máximo possível de código-fonte de ficheiros JSP.
- Fornecer a possibilidade de alterar ou criar facilmente *designs* gráficos de interfaces.
- Fornecer uma interface gráfica amigável na utilização do *GuiGen*.
- Fornecer ferramentas para gerir projectos, dar a possibilidade de fundir projectos e editar as configurações dos projectos. [4]

¹ HTTP – *HyperText Transfer Protocol*

² GUI – *Graphical User Interface*

Framework Persistence Builder



O *Persistence Builder* fornece uma *framework* para construir um suporte de persistência robusto e escalável para modelos de objectos. Modelos de objectos, representados por hierarquias de classes, dizem-se persistentes quando as instâncias criadas destas classes podem ser armazenadas num armazém de dados externo, tal como uma base de dados relacional. Além do suporte para objectos persistentes, a funcionalidade do *Persistence Builder* do *Visual Age* para Java (um IDE) inclui *frameworks* adicionais para suportar objectos em transacção e reforçar a integridade de objectos através de associações.

Embora os modelos de objectos para aplicações sejam reutilizados frequentemente, a tradução entre representações de modelos de negócio orientadas a objectos e não orientadas a objectos tem sido uma tarefa de desenvolvimento custoso. Dado que a maioria das bases de dados usadas hoje não são orientadas por objectos, a tarefa de mapear objectos em tabelas de bases de dados relacionais e dados de várias fontes tem sido a peça em falta nos *standards* da persistência de objectos. O *Persistence Builder* fornece essa peça em falta fornecendo os seguintes serviços: [5]

- Geração de código automática para as *frameworks* adjacentes.
- Facilidades de importação e exportação para trabalhar com *schemas* de bases de dados.
- Ferramentas de *debug* e monitorização para ajustes de *performance* onde forem desejados.

Estes serviços do *Persistence Builder* gerem os seguintes elementos:

- **Business Objects.** São tipicamente os objectos no domínio da aplicação que se querem persistentes num armazém de dados. Exemplo: os objectos Cliente e Conta numa aplicação bancária.
- **Associações.** No *Persistence Builder*, uma associação é uma relação que existe entre dois *business objects* persistentes. Existem três tipos principais de associações: um-para-um, um-para-muitos e muitos-para-muitos. Uma função primária do *Persistence Builder* é manter a integridade dos objectos em tempo de execução e do correspondente armazém de dados através de transacções atómicas. Exemplo: um objecto Cliente pode ter uma relação um-para-muitos com uma colecção de objectos Conta tal que cada Cliente conhece o seu conjunto de Contas e cada Conta conhece o seu Cliente.
- **Transacções.** Representam caminhos de execução de código. Exemplo: uma aplicação bancária pode ter uma transacção que actualize a conta bancária de um cliente. A actividade de código necessária para manipular os objectos persistentes ocorreriam na transacção. [5]

Eclipse



O Eclipse é uma plataforma de desenvolvimento integrada extensível “para tudo sem ser, contudo, para nada em particular”. Esta plataforma fornece vários blocos de construção e uma base para construir e correr ferramentas de desenvolvimento de *software* integradas. O Eclipse permite aos construtores de ferramentas o desenvolvimento independente destas que integram, posteriormente, com ferramentas desenvolvidas por outros e tudo isto de uma forma completamente transparente.[8] Isto traduz-se na existência de vários *plugins* desenvolvidos para o Eclipse que permitem, então, a integração de várias ferramentas diferentes nesta única plataforma de desenvolvimento.

Tomcat



No Departamento de Java é usado, para o desenvolvimento na máquina de cada programador, o IDE *Eclipse* com o *Sysdeo Eclipse Tomcat Launcher Plugin*, um *plugin* que permite integrar o *Tomcat* com o *Eclipse*.

O *Tomcat* é um *container* de *servlets* que é usado na implementação das tecnologias *Java Servlets* e *JavaServer Pages* (JSP). As especificações de *JSP / Servlets* são desenvolvidos e mantidos pela *Sun* sob o *Java Community Process* (JCP). O *Tomcat* é desenvolvido como um projecto aberto e participativo e é distribuído sob a *Apache Software License*. [6]

O *Tomcat*, em termos de utilidade, é parecido com o *Apache* (um servidor *web* para HTML, e expansível para PHP/ASP e outras linguagens de *script*) e também o IIS, que tem a mesma função do *Apache*, contudo em vez de interpretar essas linguagens de *script*, o *Tomcat* interpreta / compila páginas feitas em *Java* utilizando as tecnologias já referidas como *JSP / Servlets*, com a vantagem de se poder utilizar toda a infra-estrutura que o *Java* oferece.

IBM WebSphere



O *IBM WebSphere* é um servidor aplicacional que fornece uma solução para as implementações do modelo de programação do J2EE bem como para tecnologias alternativas tais como programas CGI¹ e *Active Server Pages*. O *WebSphere* fornece uma solução flexível e escalável apropriada para um vasto leque de ambientes, desde servidores de pequenos departamentos até grandes distribuições de *e-commerce*.

O *WebSphere Application Server*, versão 3.5 (utilizada actualmente no Departamento de Java), oferece as seguintes funcionalidades:

- Grande escalabilidade em grandes sistemas *Symmetric Multiprocessing* (SMP).
- Grande *performance* de sessões HTTP.
- *Cache* de fragmentos dinâmica, pois possibilita que partes de *servlets* e *JavaServer Pages* (JSPs) sejam colocadas em *cache*, melhorando bastante a *performance* em geral.
- *IBM Developer Kit* para a Plataforma *Java*, versão 1.3.

¹ CGI – *Common Gateway Interface*

- Servidor *web* integrado que pode ser utilizado em conjunto com um servidor *web* externo ou isoladamente.
- Melhor *performance* e flexibilidade porque os ficheiros JSP convergem para o mesmo nível de *performance* dos *servlets*.
- Melhoramentos ao nível de *Object Request Broker* (ORB) no tratamento de tipos de dados complexos tais como vectores e *hash tables*. [7]

CVS

CVS (*Concurrent Versions System*) é um sistema de controlo de versões. Usando-o, é possível armazenar o historial de ficheiros de código-fonte.

Por exemplo, pode acontecer aparecerem *bugs* quando o *software* é modificado e isso pode não ser detectado até muito tempo depois de ser feita a modificação. Com o uso de CVS pode facilmente recuperar-se versões antigas do código-fonte para ver exactamente quais as alterações que causaram o *bug*, o que pode ser uma grande ajuda.

Claro que há sempre a alternativa pouco prática de guardar cada versão de cada ficheiro, mas isto iria desperdiçar uma enorme quantidade de espaço em disco. O CVS guarda todas as versões de um ficheiro num único ficheiro de uma forma inteligente pois só são armazenadas as diferenças entre as versões.

O CVS é muito útil quando se faz parte de uma equipa de pessoas que trabalham no mesmo projecto. Neste caso, é muito fácil acontecerem casos em que as mudanças feitas por uns são gravadas por cima das mudanças feitas por outros a não ser que se seja extremamente cuidadoso. O CVS resolve este problema isolando os diferentes programadores uns dos outros. Cada programador trabalha no seu directório próprio e o CVS junta o trabalho feito quando cada programador termina a sua tarefa. [9]

Reflection em Java



Reflection é uma funcionalidade da linguagem de programação Java. Permite a um programa Java que esteja a ser executado examinar-se ou “introspeccionar-se” a si mesmo e manipular propriedades internas do programa. Por exemplo, é possível uma classe Java obter os nomes de todos os seus membros e imprimi-los no ecrã. A habilidade de examinar e manipular uma classe Java a partir de si mesma pode não parecer algo especial, mas em outras linguagens de programação este mecanismo simplesmente não existe. Por exemplo, não é possível obter informações sobre um programa em Pascal, C ou C++ a partir do próprio programa. [10]

Este mecanismo da linguagem Java pode ser utilizado de várias outras maneiras, e no caso da Normalização Gráfica, foi utilizado para invocar métodos de classes que não se sabia quais seriam até à altura de execução.

2.4 O Processo de Desenvolvimento do HCTT

Esta secção tem como objectivo retratar o processo de desenvolvimento percorrido por todos os programadores que integram a equipa de desenvolvimento do HCTT.

Análise e Especificação

Tipicamente, uma nova funcionalidade ou mesmo uma alteração a uma funcionalidade já existente inicia-se com uma análise e especificação que pode ser formal, resultando numa especificação segundo um modelo interno usado no Departamento de Java, elaborada pelos responsáveis pela Análise, ou informal, onde o documento de especificação é elaborado pelo programador após uma reunião com os responsáveis da Análise.

Implementação

De seguida é iniciada a fase de implementação. Todas as classes e ficheiros que constituem a aplicação HCTT estão disponíveis num repositório CVS, do qual cada programador deve fazer *checkout* (copiar), através do IDE Eclipse, dessas classes e ficheiros para a sua máquina local. O programador deve ter a preocupação de fazer uma sincronização frequente entre as versões que tem localmente na sua máquina e as que estão no CVS.

Testes

Após a fase de desenvolvimento segue-se uma fase de testes, no qual o programador deve testar localmente o que foi por si implementado. Para testar a aplicação localmente é utilizado, como servidor aplicacional, o *Tomcat*. Uma vez testado com sucesso, deve ser feito *commit* para o CVS de tudo o que foi alterado / criado / eliminado.

Existe um servidor, denominado *Alpha*, para o qual são exportadas as versões mais recentes e estáveis do HCTT e no qual se executam os testes finais à aplicação. Este servidor corre o *IBM WebSphere Application Server*. Cada programador é também responsável por exportar para esse servidor tudo o que desenvolveu e em executar, mais uma vez, os testes à aplicação.

Documentação

Se os testes confirmarem a estabilidade do que foi implementado, deve então ser preenchido um pedido técnico, que consiste num registo, em papel, de todas as classes e ficheiros que foram criados / alterados / eliminados durante a implementação que cada programador executou e que contém, também, informação sobre as datas em que essa implementação tomou lugar e o número de horas consumidas.

2.5 O Trabalho de Estágio no Projecto HCTT

Pretende-se fazer, de seguida, uma introdução ao trabalho realizado durante o estágio.

São elaborados, em primeiro lugar, um pequeno resumo e enquadramento das três fases em que consistiu o estágio (ilustradas na Figura 3), seguindo-se a apresentação do planeamento de estágio inicial. Também é dada a informação sobre o local onde se pode aceder ao *sítio web* de estágio e é fornecida uma *password* e um *username* de acesso.

Fase 1 – O MethoDoc

O *MethoDoc* é uma espécie de Manual de Qualidade da aplicação HCTT.

A equipa do Departamento de Java tinha já tentado elaborar este documento, mas até à data de início de estágio só tinha sido possível definir uma primeira estrutura / esqueleto do documento. Ficou a meu cargo e a cargo do meu colega de estágio Carlos Rebelo reunirmos toda a documentação existente sobre os processos de desenvolvimento do Departamento de Java e elaborarmos uma estrutura final para o *MethoDoc*, seguido da escrita efectiva de conteúdos nesse documento. A elaboração do *MethoDoc* teve também a intenção de servir como uma preparação para o restante do trabalho de estágio.

Este tema está desenvolvido em detalhe no capítulo 3 “O MethoDoc”.

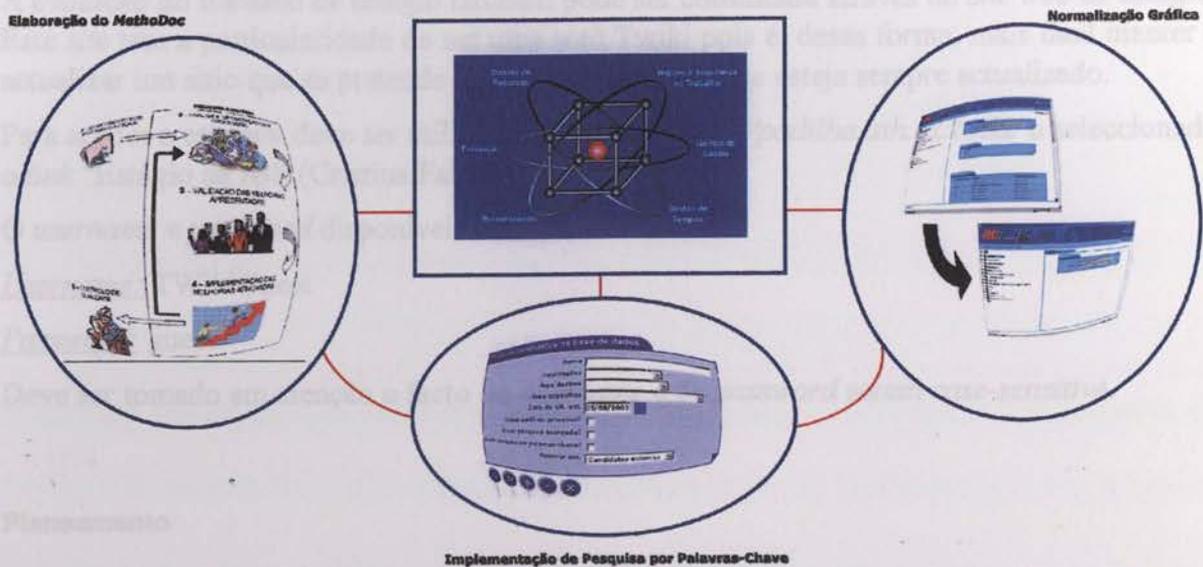


Figura 3 – Fases do Estágio

Fase 2 – A Pesquisa por Palavras-Chave

A Pesquisa por Palavras-Chave foi uma nova funcionalidade do HCTT cuja implementação foi da minha responsabilidade, e que incidiu sobre o Módulo de Recrutamento e sobre o Módulo Base – Colaboradores.

Já existiam dois tipos de pesquisa que se podiam efectuar sobre Candidatos e/ou Colaboradores: pesquisa por Perfil e pesquisa avançada, utilizando filtros definidos pelo utilizador. A nova Pesquisa por Palavras-Chave foi implementada com o intuito de poder ser utilizada conjuntamente com os dois tipos de pesquisa já existentes (ou seja, de modo a ser possível ao utilizador executar uma pesquisa em que se pretende a intersecção dos vários tipos de pesquisa) ou isoladamente. A implementação desta funcionalidade também pretendeu servir como base de preparação para a fase seguinte do estágio.

Este tema está desenvolvido em detalhe no capítulo 4 “A Pesquisa por Palavras-Chave”.

Fase 3 – A Normalização Gráfica

A Normalização Gráfica compreendeu, de uma forma geral, a normalização do *layout*, navegação e interacção com o utilizador em páginas complexas do HCTT bem como uma normalização técnica de forma a implementar processos XML e *Adapters* “genéricos” para as listagens de Responsáveis e Colaboradores Activos do HCTT que pudessem ser facilmente utilizados pelos programadores que necessitassem incluir este tipo de listagens num qualquer sítio da aplicação.

Esta necessidade foi surgindo ao longo do desenvolvimento da aplicação, pois não existiu uma preocupação, por parte dos programadores que já passaram pela equipa de desenvolvimento, em manter uma grande coerência em termos de informação apresentada ao utilizador, acções existentes, navegação e reaproveitamento de código para páginas do HCTT semelhantes e foram estes, então, os objectivos do trabalho de estágio efectuado nesta fase.

Este tema está desenvolvido em detalhe no capítulo 5 “A Normalização Gráfica”.

Site web de Estágio

A evolução do trabalho de estágio também pode ser consultada através do *site web* de estágio. Este *site* tem a particularidade de ser uma *web Twiki* pois é, dessa forma, mais fácil manter e actualizar um sítio que se pretende que seja dinâmico e que esteja sempre actualizado.

Para aceder a este *site* deve ser utilizado o endereço <http://padilha.ath.cx/twiki/> e seleccionado o *link* “Estágio na IBS (Cristina Falcão)”.

O *username* e *password* disponíveis são:

Username: TWikiGuest

Password: guest

Deve ser tomado em atenção o facto do *username* e da *password* serem *case-sensitive*.

Planeamento

No dia 3 de Março de 2004 (1º dia de estágio), foi definido, em reunião, o seguinte planeamento:

1ª Parte: Formação

- Duração: uma semana e meia
- Formação técnica
- Formação funcional

2ª Parte: 1ª versão do MethoDoc

- Duração: duas semanas e meia
- O *MethoDoc* é uma espécie de Manual de Qualidade da aplicação HCTT do Departamento de Java da IBS.
- Dividido originalmente em três secções:
 - Análise / Especificação
 - Desenvolvimento
 - Desenvolvimento em Equipa
- Contém, em geral, regras de desenvolvimento e metodologias.
- **Prazo:** até 31 de Março

3ª Parte: Implementação de uma funcionalidade do HCTT

- Duração: um mês
- A minha funcionalidade é a Pesquisa por Palavras-Chave entre os candidatos a colaboradores.
- **Prazo:** até 30 de Abril já testado

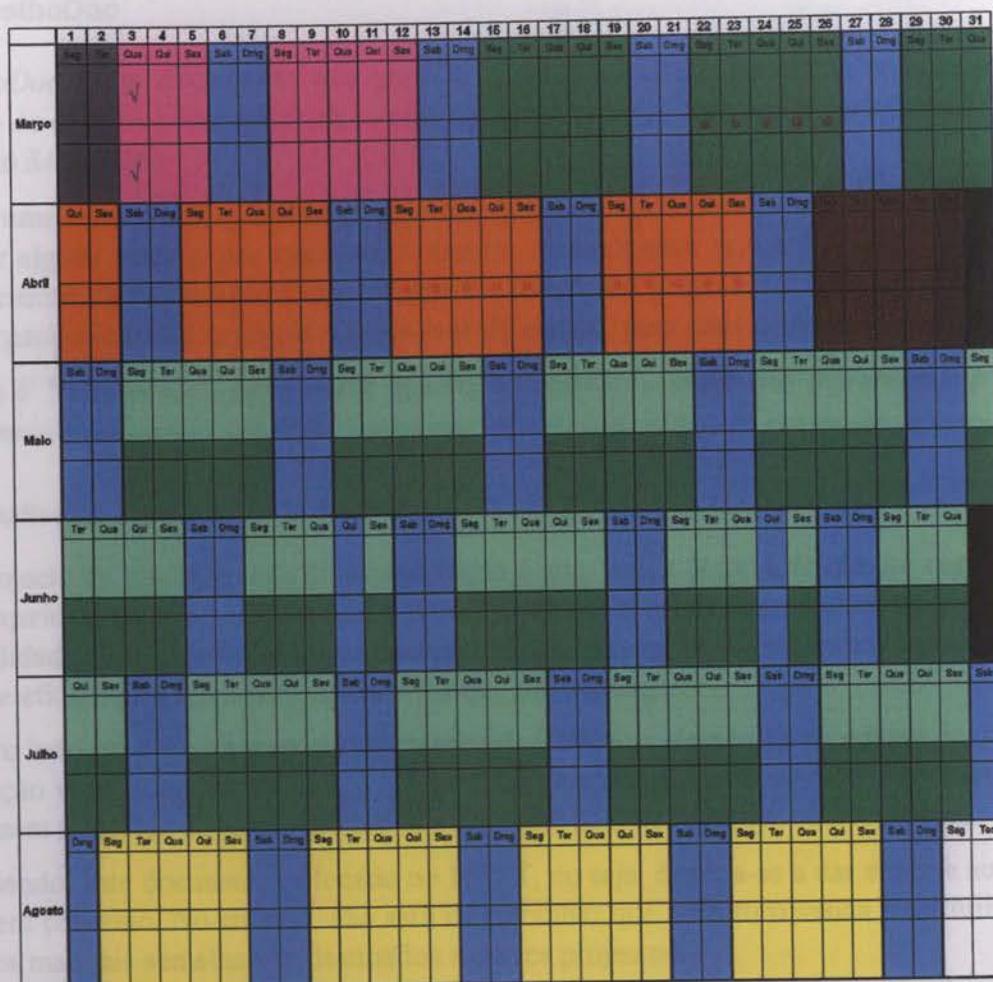
4ª Parte: Normalização gráfica da aplicação

- Duração: 3 meses
- Paralelamente, terminar a elaboração do *MethoDoc*
- **Prazo:** até fim de Julho

5ª Parte: Relatório de Estágio e finalizações

- Duração: 1 mês
- **Prazo:** até fim de Agosto.

Este planeamento está reflectido no cronograma da Figura 4. É executada uma análise mais profunda à evolução do planeamento de estágio no capítulo 6 “Análise ao Planeamento de Estágio”.



- Formação
- MethoDoc
- Funcionalidade do HCTT - Pesquisa por Palavras-Chave
- Normalização Gráfica do HCTT
- Relatório e Finalização
- Semana crítica para implementação da funcionalidade do HCTT - Pesquisa por Palavras-Chave
- Gonçalo Mata estará ausente

Figura 4 - Cronograma do Planeamento Inicial do Estágio

3 O MethoDoc

O *MethoDoc* é um documento que permite a um novo elemento aprender tudo acerca do processo de desenvolvimento praticado no Departamento de Java - esta foi a “visão” que deu origem ao *MethoDoc*.

Durante uma fase inicial, e depois de algumas reuniões, foi necessário reescrevê-la de modo a clarificar alguns pontos, por exemplo, o âmbito. Actualmente esta é a directriz fundamental do documento: “o *MethoDoc* é um documento, focado no projecto HCTT, que materializa de forma organizada todas as regras e *know-how* da equipa, para cada fase do desenvolvimento”.

O anexo F “MethoDoc”, em volume aparte, é constituído pelos três primeiros capítulos do *MethoDoc*.

3.1 Análise do Problema do MethoDoc

Num projecto de média/grande dimensão como é que, sem a certeza de que **há regras** e estas **são cumpridas**, se pode afirmar que o processo de desenvolvimento é eficiente e o resultado tem qualidade? Se o *MethoDoc* tiver sucesso na sua missão, permitirá aproximar a equipa dos níveis de eficiência e qualidade necessários que todos desejam.

Por outro lado irá permitir uma rápida integração de novos elementos na equipa, já que toda a informação vital estará sintetizada e organizada, dando todas as orientações necessárias para que possam trabalhar de forma produtiva e coerente com os restantes colaboradores.

Assim sendo, este documento é focado no HCTT, ou seja, destina-se a dar suporte ao projecto HCTT em concreto. No entanto, não será de estranhar que no futuro venha a ser útil à escrita de outros manuais semelhantes destinados a outros projectos.

Quanto à necessidade deste documento, esta deve-se essencialmente a dois factos:

- Os elementos que compõem a equipa de desenvolvimento podem mudar e conseqüentemente é necessário reter o conhecimento que cada indivíduo foi adquirindo.
- Ao longo das fases de desenvolvimento são estabelecidas regras das mais variadas espécies (interfaces, especificações, código, qualidade, etc.). É vital que estas regras sejam conhecidas, compreendidas, aceites e praticadas por todos os elementos da equipa.

Já tinha havido uma primeira tentativa, por parte da equipa de desenvolvimento do Departamento de Java, em elaborar o *MethoDoc*, embora até à data de início de estágio só tenham conseguido chegar a uma estrutura / esqueleto do documento.

Para sermos bem sucedidos neste desafio, eu e o outro responsável pela elaboração efectiva do *MethoDoc* (Carlos Rebelo) definimos a seguinte abordagem:

- Reunir toda a documentação existente sobre os processos de desenvolvimento do Departamento, a qual estava dispersa e apresentava-se sobre várias formas: rascunhos em papel, documentos *Word* impressos em papel, documentos electrónicos tais como *e-mails* ou registos numa base de dados *Lotus Notes*, etc.
- Definir um ciclo de evolução para o *MethoDoc*.
- Entrevistar alguns elementos do Departamento de Java.

- Redefinir uma estrutura / esqueleto do documento.
- Validar essa estrutura com os responsáveis do Departamento de Java e com os restantes membros da equipa.
- Colocar conteúdos no documento.

3.2 Descrição da Solução Proposta para o MethoDoc

O ciclo de evolução definido para o *MethoDoc* pode ser representado pela Figura 5, a seguir:

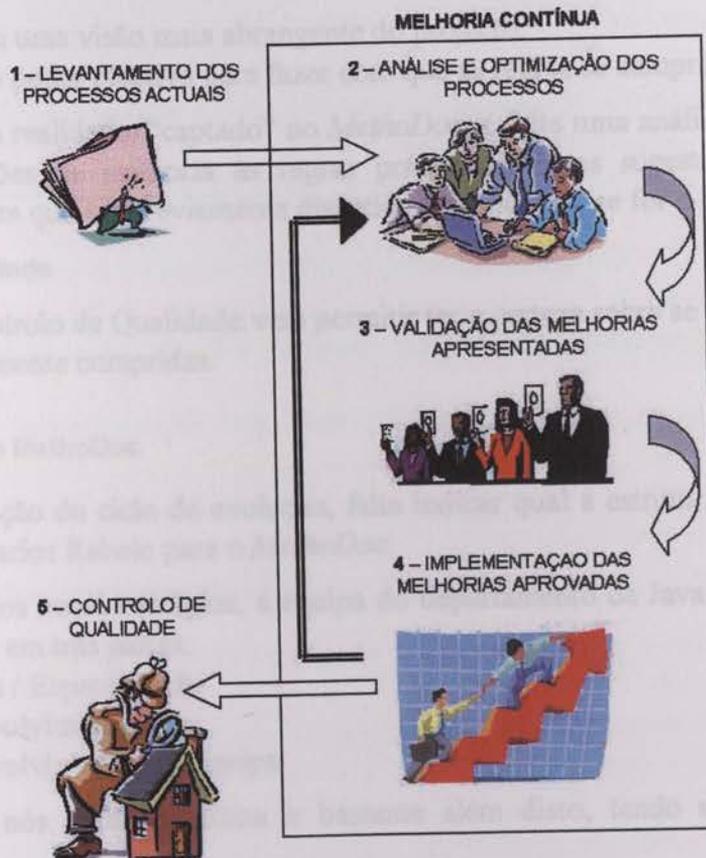


Figura 5 - Ciclo de Evolução do MethoDoc

O *MethoDoc* é um documento “vivo”. Para que atinja os seus objectivos, tem que estar sempre de acordo com a realidade praticada no Departamento. Isto só pode ser conseguido se todos os elementos contribuírem no Levantamento dos Processos Actuais.

Observando a figura, é possível dividir o ciclo de evolução do *MethoDoc* em três partes distintas:

- Levantamento dos processos
- Melhoria contínua
- Controlo de Qualidade

Levantamento dos Processos

O Levantamento dos Processos consiste em “alimentar” permanentemente o *MethoDoc* com todas as regras / *know-how* existentes (que ainda não tenham sido escritas) ou que tenham sido criadas / alteradas.

Numa fase inicial este levantamento poderá ser atribuído a 2 ou 3 pessoas devido ao facto de acarretar a sintetização de muita informação, tendo sido então atribuído a mim e ao meu colega Carlos Rebelo. Numa fase estável do documento é da responsabilidade de todos manter o documento de acordo com a realidade.

Melhoria Contínua

A Melhoria Contínua é igualmente uma tarefa que deve ser partilhada por todos, mas na qual os elementos associados à gestão do departamento e da equipa têm um papel principal a desempenhar:

- Possuem uma visão mais abrangente do projecto.
- Detêm o poder efectivo para fazer com que as regras se cumpram.

Sobre o reflexo da realidade, “captado” no *MethoDoc*, é feita uma análise crítica com vista a apresentar sugestões de melhoria às regras praticadas. Estas sugestões, antes de serem implementadas, têm que ser previamente discutidas e validadas, se for o caso, pela gestão.

Controlo de Qualidade

Finalmente, o Controlo de Qualidade vem permitir ter a certeza sobre se as alterações estão ou não a ser efectivamente cumpridas.

3.3 Estrutura do MethoDoc

Finda a apresentação do ciclo de evolução, falta indicar qual a estrutura / esqueleto definida por mim e pelo Carlos Rebelo para o *MethoDoc*.

Antes do início dos nossos estágios, a equipa do departamento de Java tinha chegado a uma estrutura definida em três partes:

- Análise / Especificação
- Desenvolvimento
- Desenvolvimento em Equipa

A estrutura por nós definida tentou ir bastante além disto, tendo sido dividida em seis capítulos:

- *Capítulo 1*: Introdução (onde se apresenta o *MethoDoc* e se explica como se aprova o desenvolvimento de novos projectos na IBS Portugal)
- *Capítulo 2*: Análise / Especificação
- *Capítulo 3*: Desenvolvimento
- *Capítulo 4*: Testes
- *Capítulo 5*: Trabalho em Equipa
- *Capítulo 6*: *Deployment* / Manutenção

Como é facilmente observável, optou-se por utilizar uma divisão em capítulos de acordo com os princípios da Engenharia de *Software* onde o ciclo de vida de um produto de *Software* é composto por várias fases e genericamente identificam-se as seguintes: Levantamento de Requisitos / Especificação (capítulo 2), Desenvolvimento (capítulos 3 e 5), Validação / Testes (capítulos 4 e 5) e *Deployment* / Manutenção (capítulo 6).

Assim sendo, o documento segue uma estrutura lógica mas que se debruça sobre todos os aspectos importantes do ciclo de desenvolvimento.

No que diz respeito à escrita de conteúdos, os capítulos 1 e 2 do *MethoDoc* encontram-se estáveis, estando o capítulo 3 (um dos capítulos mais extensos e mais críticos) ainda em

elaboração, considerando-se que esteja estável em cerca de metade da sua extensão. Não pode ser esquecido que, sendo este um documento “vivo” nunca poderá ser considerado como “concluído”. Os restantes capítulos ainda não possuem conteúdo porque se optou por focar a atenção, durante o estágio, nos primeiros três capítulos, dos quais os capítulos 2 e 3 são os mais críticos (a par com o capítulo 5) e aqueles em que existia mais urgência em haver regras documentadas.

3.4 Conteúdo do MethoDoc

Encontram-se como anexo a este relatório (anexo F "MethoDoc") os primeiros três capítulos do *MethoDoc*, tendo sido elaborado por mim:

- O capítulo 2 “Análise / Especificação”, que contém a descrição dos documentos que devem ser produzidos nesta fase e regras a ter em atenção.
- Do capítulo 3 “Desenvolvimento”, as secções
 - 3.2 "Frameworks", mini-tutoriais que explicam a utilização das ferramentas gráficas para as *frameworks* utilizadas no Departamento.
 - 3.6 "Implementação de Listagens de Responsáveis e Colaboradores Activos", um manual de utilização dos processos XML e das classes implementados por mim durante a fase de Normalização Gráfica.
 - 3.7 "Implementação de *tabs* (separadores)", contendo um manual de programação para a implementação de separadores em qualquer página do HCTT.

3.5 Principais Dificuldades Encontradas

A primeira sensação no início da execução da tarefa de elaborar o *MethoDoc* foi a de uma certa apreensão sobre se nós, dois estagiários recém-chegados ao Departamento, seríamos as pessoas mais indicadas para descrever os processos desse Departamento. Parecia ser mais lógico que essa tarefa fosse levada a cabo por quem já conhecesse a fundo esses processos. Esta ideia foi rapidamente dissipada quando a elaboração do *MethoDoc* revelou ser uma excelente forma de, em tempo recorde, nos inteirarmos de tudo o que era feito no Departamento e da forma como era feito.

Na primeira fase também existiram dificuldades relativamente a definição das fronteiras do documento, pois essas não foram explicitamente indicadas quando a tarefa nos foi atribuída. Deveria ser um “meta-manual” aplicável a todos os projectos da equipa de Java? Focar-se-ia apenas na fase de implementação e desenvolvimento? Há necessidade de um documento daquele tipo? Que vantagens objectivas traria?

Durante as reuniões individuais com elementos do Departamento de Java constatou-se que existia uma discrepância sobre o que cada pessoa pensava ser o *MethoDoc*, sendo possível identificar duas vertentes:

- As que achavam que o *MethoDoc* deveria ser um documento que descrevesse os procedimentos gerais do Departamento de Java para o desenvolvimento de *software*, sem se focar num projecto específico, ou seja, deveria conter, principalmente, as *best practices* de Engenharia de *Software*, para cada fase do processo de desenvolvimento de um novo *software* no Departamento. No que diz respeito a projectos específicos, deveria existir um documento para cada um aparte do *MethoDoc*.

- As que achavam que o *MethoDoc* deveria ser um documento específico do projecto HCTT e, como tal, deveria conter informação detalhada sobre este projecto no que diz respeito a arquitectura, nomenclaturas usadas para classes Java, boas práticas de programação, métodos de análise e especificação utilizados para este projecto, forma como a equipa de desenvolvimento deste projecto trabalha, como é feito o *deployment* e manutenção de versões instaladas em clientes, etc. Em suma, o *MethoDoc* deveria ser um documento detalhado sobre o projecto HCTT.

Em reunião com os responsáveis pelo Departamento de Java foi-nos transmitido que o *MethoDoc* seria, definitivamente, um documento específico do projecto HCTT.

3.6 Avaliação de Resultados e Conclusões

Após as primeiras dificuldades iniciais, que se prenderam com a definição do âmbito do *MethoDoc*, sobre se seria um documento sobre o Departamento de Java em geral ou específico para o HCTT, a tarefa de o elaborar não se revelou de elevada complexidade. A parte mais difícil e que exigiu mais esforço foi a definição da estrutura / esqueleto do documento, pois havia várias alternativas para essa estrutura e era necessário escolher a mais adequada e a que tornasse o *MethoDoc* um documento de consulta fácil e rápida.

A tarefa obrigou-me a mim e ao Carlos Rebelo a tomar rapidamente conhecimento dos processos do Departamento, através da análise de documentação já existente e através de entrevistas a alguns elementos da equipa, o que se revelou uma vantagem para a nossa integração na equipa e para a rápida evolução do nosso desempenho dentro dela. Foi, em suma, um trabalho a executar que também assumiu o papel de formação.

Existem, no entanto, algumas reflexões / preocupações a salientar e que se prendem com uma realidade muito simples e facilmente compreendida: a integração do *MethoDoc* na rotina da equipa de desenvolvimento, aproveitando todas as vantagens que um documento como este pode trazer, não será uma tarefa fácil e trivial. Se existe uma vontade efectiva de criar e manter este documento é então necessário que todos tenham a consciência que isso implicará executar tarefas que terão que ser efectivamente realizadas por todos e não só por alguns. Para que a consulta do *MethoDoc* e a escrita de conteúdos nele seja algo natural em qualquer um dos elementos da equipa do Departamento será então necessário envolvê-los nesta ambição para que o documento não seja encarado como uma ideia utópica ou teórica de outros.

4 A Pesquisa por Palavras-Chave

A Pesquisa por Palavras-Chave foi uma nova funcionalidade do HCTT cuja implementação foi da minha responsabilidade, e que incidiu sobre o Módulo de Recrutamento e sobre o Módulo Base – Colaboradores.

O grande objectivo desta funcionalidade é possibilitar a cada utilizador do HCTT pesquisar da maneira que for mais importante para o seu negócio. As palavras-chave permitem facilmente a cada utilizador associar aos seus Colaboradores e Candidatos tipos de informação não previstas no HCTT ou demasiado específicas da sua área de negócio, para depois efectuar pesquisas sobre elas. Por mais completa que seja a informação disponível sobre cada Colaborador no HCTT é também necessário fornecer ao utilizador um tipo de informação parametrizável e foi com base neste pressuposto que surgiu a necessidade de implementar as palavras-chave e a execução de pesquisas sobre elas, especificamente no domínio dos Processos de Selecção.

O anexo G "Especificação da Pesquisa por Palavras-Chave", em volume aparte, refere-se ao documento de especificação desta funcionalidade, segundo o modelo de documentos de especificação utilizado no Departamento.

4.1 Análise do Problema da Pesquisa por Palavras-Chave

Já existiam dois tipos de pesquisa que se podiam efectuar sobre Candidatos e/ou Colaboradores: pesquisa por Perfil e pesquisa avançada, utilizando filtros definidos pelo utilizador. Foi então necessário implementar também a Pesquisa por Palavras-Chave, inserida nos Processos de Selecção.

Um Processo de Selecção tem como objectivo a angariação de Candidatos e tratamento de Candidatos em fase de selecção.

Um Colaborador e Candidato têm associado a si vários tipos de informações (dados pessoais, formação profissional, dados clínicos, etc.) e pretende-se que também tenham associado a si um novo tipo de informação: **palavras-chave**, inseridas por um recrutador, onde cada palavra-chave dá uma informação adicional sobre esse Colaborador ou Candidato que não esteja já contemplada noutros tipos de informação sobre estes.

Por outro lado, cada Processo de Selecção deve ter também associado a si palavras-chave que se pretendem que sejam características possuídas pelos Colaboradores ou Candidatos, de modo a ser possível realizar uma pesquisa entre Colaboradores e / ou Candidatos angariados nesse Processo de Selecção que tenham associados a si essas palavras-chave. Ou seja, as palavras-chave associadas a Colaboradores ou Candidatos representam informação adicional possuídas por esses, e as palavras-chave associadas a Processos de Selecção são informação adicional que se pretende que seja possuída pelo universo de Pessoas (Colaboradores ou Candidatos) sobre as quais é efectuada a pesquisa.

A um Processo de Selecção estão associados um ou mais Meios de Recrutamento. Um Meio de Recrutamento é a forma como se angariam candidatos (por exemplo: anúncio de jornal, recrutamento interno, etc.). A Pesquisa por Palavras-Chave pode ser efectuada no domínio de um Processo de Selecção ou no domínio de cada um dos seus Meios de Recrutamento, que têm associado a si as mesmas palavras-chave do Processo de Selecção a que dizem respeito.

A diferença entre a pesquisa em cada domínio reside essencialmente no universo de Pessoas sobre as quais ela é efectuada. No caso da pesquisa no domínio de um Processo de Selecção é possível o utilizador escolher se pretende efectuar a pesquisa sobre todas as Pessoas da base de dados; sobre Candidatos externos ou então sobre Colaboradores internos.

No domínio do Meio de Recrutamento a pesquisa só é efectuada sobre os Candidatos angariados através desse Meio de Recrutamento.

Em anexo a este relatório (anexo G "Especificação da Pesquisa por Palavras-Chave") encontra-se uma especificação desta funcionalidade, segundo o modelo de especificação utilizado internamente no Departamento de Java.

4.2 Descrição da Solução para a Pesquisa por Palavras-Chave

Para ir de encontro aos objectivos descritos anteriormente, a nova funcionalidade de Pesquisa por Palavras-Chave implicou a execução de quatro grandes tarefas, descritas a seguir. A Figura 6 ilustra o diagrama de casos de uso da funcionalidade.

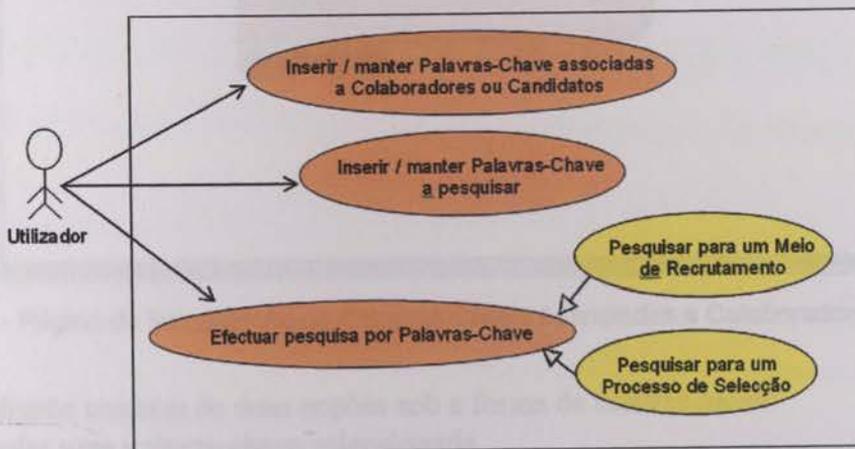


Figura 6 - Diagrama de Casos de Uso da Funcionalidade de Pesquisa por Palavras-Chave

1ª Tarefa: possibilitar ao utilizador a manutenção de palavras-chaves associadas a Pessoas (Colaboradores e Candidatos).

Esta sub-funcionalidade foi inserida na página de detalhes de uma Pessoa sob a categoria "Informação Adicional" (

).

Nesta página o utilizador tem à sua disposição quatro acções sob a forma de botão:

- Criar uma nova palavra-chave.
- Editar em modo livre as palavras-chave.
 - Neste caso, as palavras-chave são apresentadas numa caixa de texto, na forma de texto corrido, sendo as palavra-chave separadas umas das outras por um espaço em branco. O utilizador por então editar este texto livremente. Ao acabar a edição as palavras-chave são processadas (é feito um *parse*) de modo a ser detectado quais são para serem novamente dispostas na forma de uma listagem. Para o *parse* das palavras-chave após a edição em modo livre definiu-se que o elemento separador entre as palavras-chave era também o espaço em branco.

- Consultar um *template* de palavras-chave.
 - Entre as palavras-chave *template* apresentadas, o utilizador pode escolher entre adicionar à listagem de palavras-chave de uma Pessoa, ou remover desta, uma ou mais palavras *template* seleccionadas.
- Anular uma ou mais palavras-chave seleccionadas.

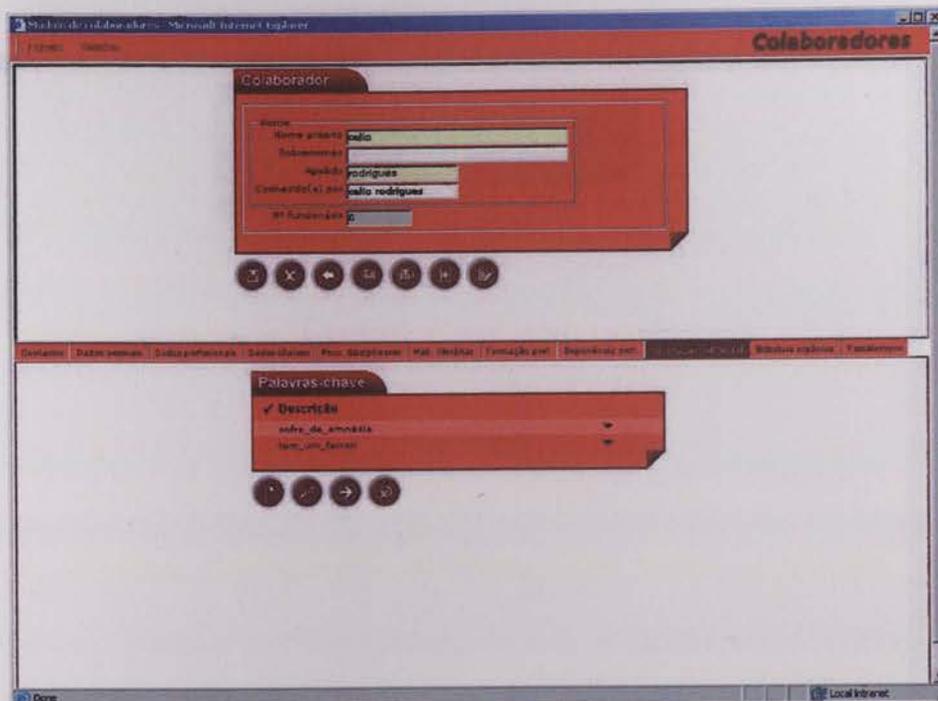


Figura 7 - Página de Manutenção de Palavras-Chave Associadas a Colaboradores / Candidatos

O utilizador dispõe também de duas acções sob a forma de *context-menu*:

- Anular uma palavra-chave seleccionada.
- Editar uma palavra-chave seleccionada

2ª Tarefa: possibilitar ao utilizador a manutenção de palavras-chave sobre as quais pretende efectuar uma pesquisa.

Esta sub-funcionalidade foi inserida na página de pesquisa num Processo de Selecção (Figura 8), bem como na página de pesquisa num Meio de Recrutamento, através de um botão com a etiqueta "Palavras-chave", que se encontra abaixo do formulário de pesquisa.

Nesta página o utilizador tem à sua disposição cinco acções sob a forma de botão:

- Criar uma nova palavra-chave.
- Editar em modo livre as palavras-chave.
 - Neste caso, as palavras-chave são apresentadas numa caixa de texto, na forma de texto corrido, sendo as palavras-chave separadas umas das outras por um espaço em branco. O utilizador por então editar este texto livremente. Ao acabar a edição as palavras-chave são processadas (é feito um *parse*) de modo a ser detectado quais são para serem novamente dispostas na forma de uma listagem. Para o *parse* das palavras-chave após a edição em modo livre definiu-se que o elemento separador entre as palavras-chave era o espaço em branco.
- Consultar um *template* de palavras-chave.

- Entre as palavras-chave *template* apresentadas, o utilizador pode escolher entre adicionar à listagem de palavras-chave de uma Pessoa ou remover desta uma ou mais palavras *template* seleccionadas.
- Anular uma ou mais palavras-chave seleccionadas.
- Voltar ao formulário de pesquisa.

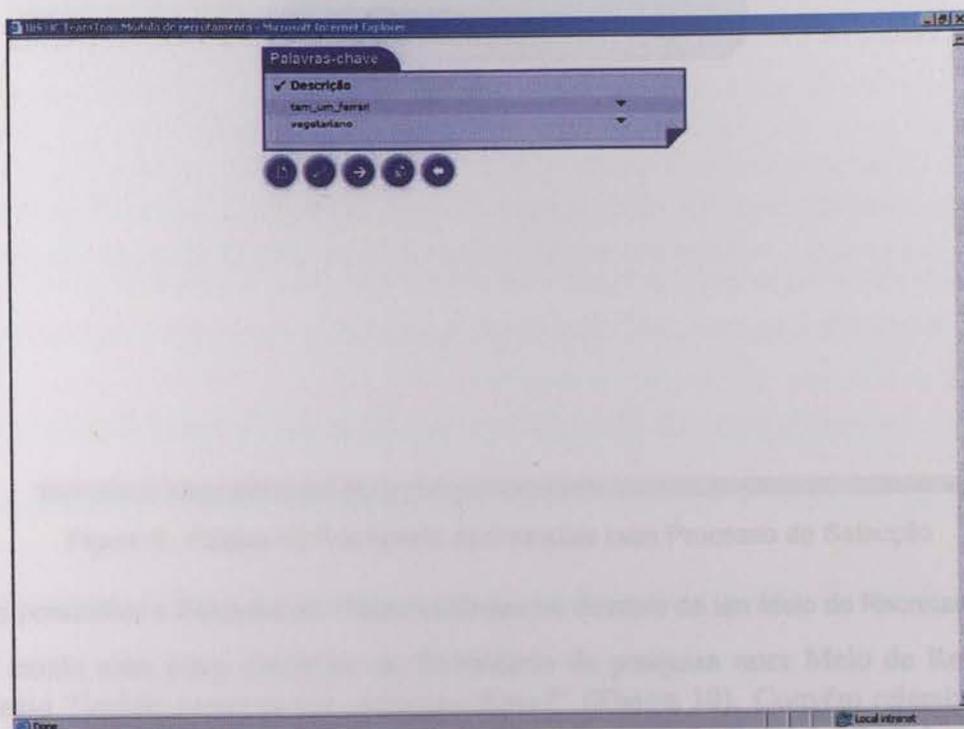


Figura 8 - Página de Manutenção de Palavras-Chave Associadas a um Processo de Selecção

O utilizador pode dispor também de duas acções sob a forma de *context-menu*:

- Anular uma palavra-chave seleccionada.
- Editar uma palavra-chave seleccionada

3ª Tarefa: possibilitar a Pesquisa por Palavras-Chave no domínio de um Processo de Selecção.

Para tal existe uma nova *checkbox* no formulário de pesquisa num Processo de Selecção, com o texto “Incluir pesquisa por palavras-chave?” (Figura 9). Convém lembrar que neste domínio o utilizador pode optar por efectuar a pesquisa sobre todas as Pessoas da base de dados, sobre Candidatos externos ou então sobre Colaboradores internos.

Ao ser seleccionada esta *checkbox* é então desencadeada a Pesquisa por Palavras-Chave e apresentado o resultado no ecrã. Os detalhes da implementação da pesquisa são descritos na secção seguinte intitulada “Implementação da Pesquisa por Palavras-Chave”.

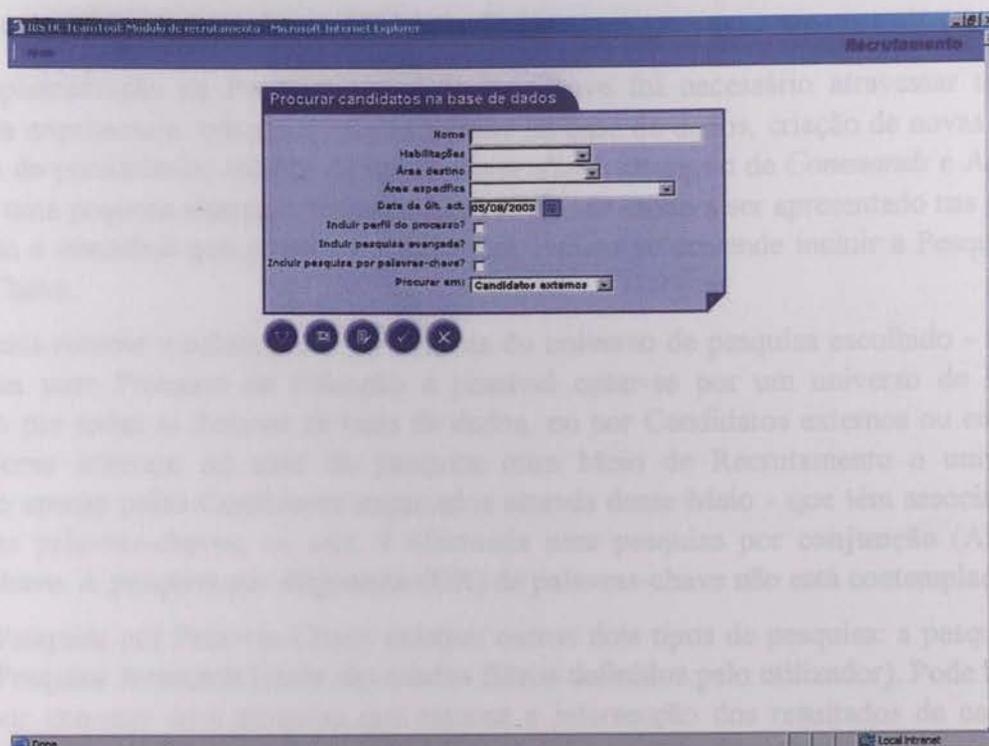


Figura 9 - Página do Formulário de Pesquisa num Processo de Selecção

4ª Tarefa: possibilitar a Pesquisa por Palavras-Chave no domínio de um Meio de Recrutamento.

Para tal existe uma nova *checkbox* no formulário de pesquisa num Meio de Recrutamento com o texto “Incluir pesquisa por palavras-chave?” (Figura 10). Convém lembrar que um Meio de Recrutamento tem associado a si as palavras-chave do Processo de Selecção ao qual diz respeito, sendo a pesquisa efectuada somente sobre os Candidatos angariados através desse Meio de Recrutamento.

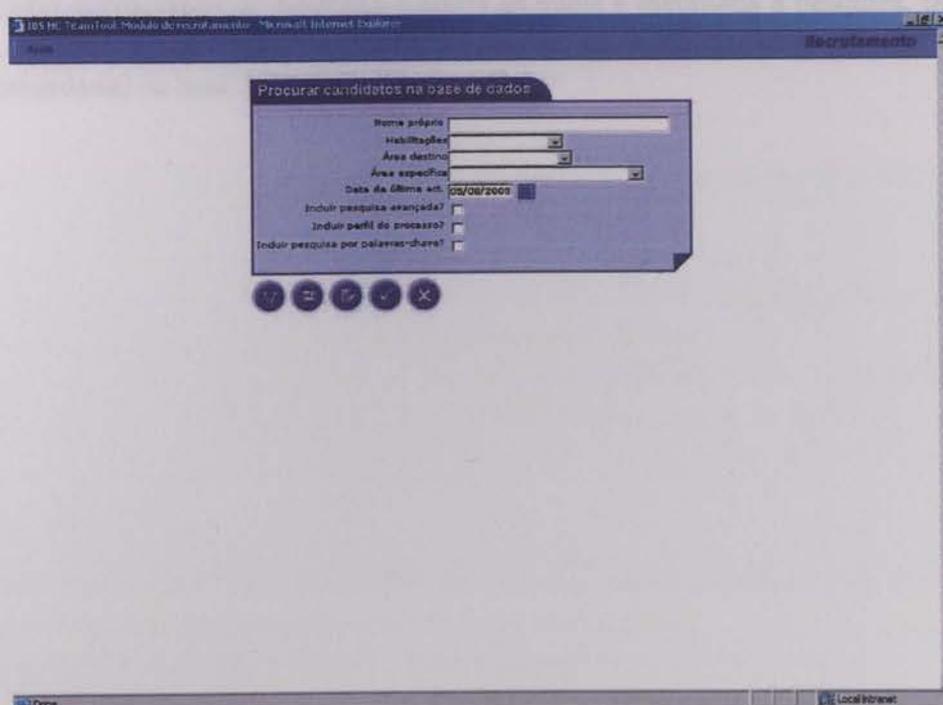


Figura 10 - Página do Formulário de Pesquisa num Meio de Recrutamento

4.3 Implementação da Pesquisa por Palavras-Chave

Para a implementação da Pesquisa por Palavras-Chave foi necessário atravessar todas as camadas da arquitectura: criação de novas tabelas na base de dados, criação de novas classes na camada de persistência, criação de novos *Controllers*, alteração de *Commands* e *Adapters* e somente uma pequena alteração num processo XML, de modo a ser apresentado nas páginas de pesquisa a *checkbox* que permite ao utilizador indicar se pretende incluir a Pesquisa por Palavras-Chave.

Esta pesquisa retorna o subconjunto de Pessoas do universo de pesquisa escolhido - no caso da pesquisa num Processo de Selecção é possível optar-se por um universo de Pessoas constituído por todas as Pessoas da base de dados, ou por Candidatos externos ou então por Colaboradores internos; no caso da pesquisa num Meio de Recrutamento o universo é constituído apenas pelos Candidatos angariados através desse Meio - que têm associado a si **todas** essas palavras-chaves, ou seja, é efectuada uma pesquisa por **conjunção (AND)** de palavras-chave. A pesquisa por **disjunção (OR)** de palavras-chave não está contemplada.

A par da Pesquisa por Palavras-Chave existem outros dois tipos de pesquisa: a pesquisa por Perfil e a Pesquisa Avançada (onde são usados filtros definidos pelo utilizador). Pode também optar-se por executar uma pesquisa que retorne a intersecção dos resultados de cada uma destas pesquisas, sendo somente necessário seleccionar nos formulários de pesquisa as *checkboxes* das pesquisas pretendidas.

Segue-se uma apresentação de todas as acções levadas a cabo em cada camada da arquitectura do *HCTT* para a implementação da Pesquisa por Palavras-Chave.

Base de dados

Foram criadas duas novas tabelas na base de dados: uma tabela para guardar os registos de palavras-chave associadas a Pessoas e uma outra tabela para guardar os registos de palavras-chave associadas a Processos de Selecção e sobre as quais é efectuada a pesquisa, tal como é indicado no seguinte modelo relacional parcial (pois só estão ilustradas as tabelas relevantes para a funcionalidade) da base de dados (Figura 11):

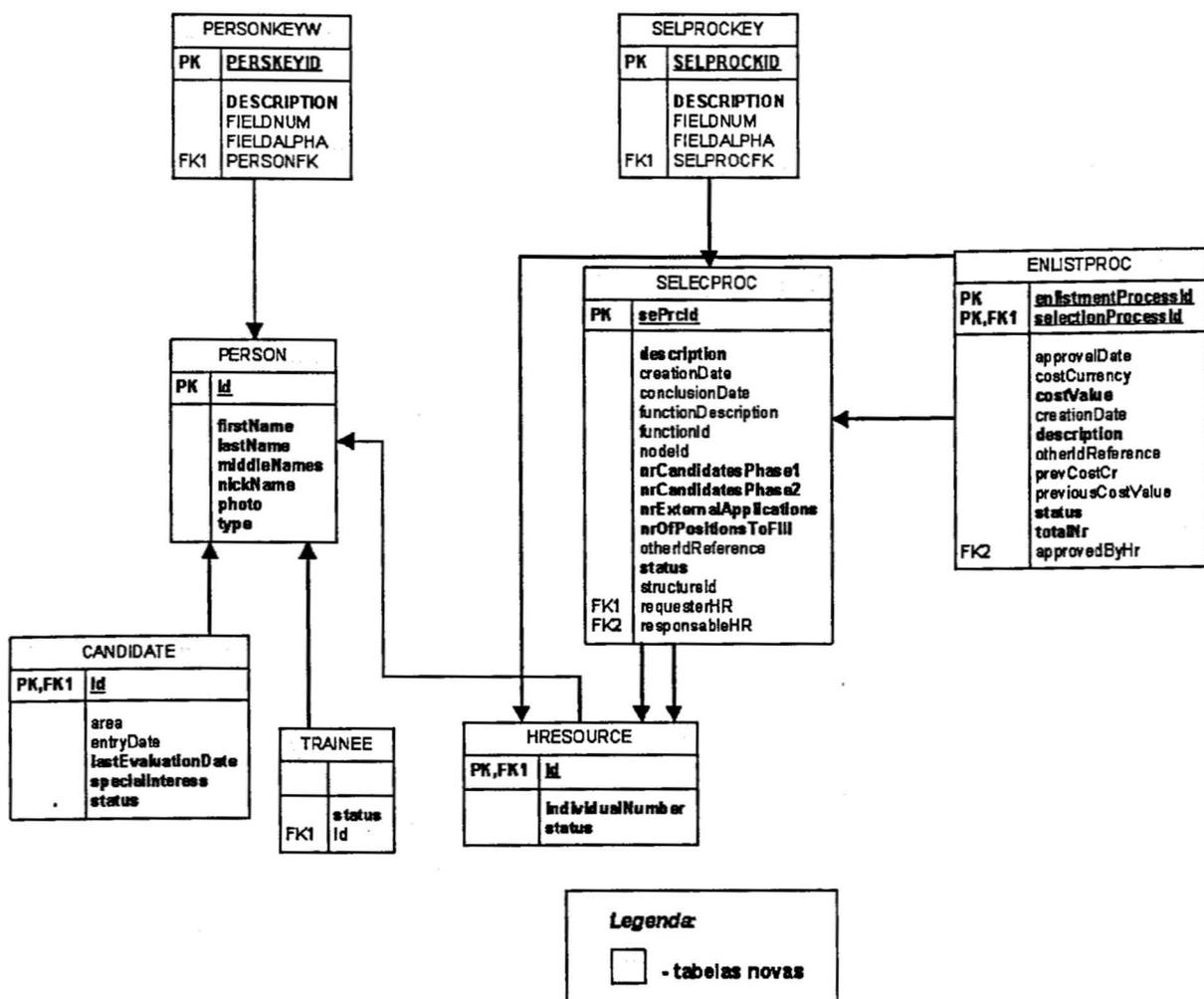


Figura 11 - Modelo relacional parcial da base de dados para a Pesquisa por Palavras-Chave

Tabela PERSONKEYW

Na tabela PERSONKEYW são guardadas as palavras-chaves associadas a Pessoas. Os campos são:

- PERSKEYID: Decimal (9,0) – **chave primária**
- PERSONFK: Varchar(10) – **chave-estrangeira** para a tabela PERSON (não faz parte da chave primária).
- DESCR: Varchar(50) – campo que irá conter a palavra-chave
- FIELDNUM: Decimal (10,0) – campo numérico extra caso venha a ser necessário futuramente (regra adoptada pelo Departamento de Java)
- FIELDALPHA: Varchar(50) – campo alfanumérico extra caso venha a ser necessário futuramente (regra adoptada pelo Departamento de Java).

Tabela SELPROCKEY

Na tabela SELPROCKEY são guardadas as palavras-chave associadas a Processos de Selecção e sobre as quais é elaborada a pesquisa. Os campos são:

- SELPROCKID: Decimal (9,0) – **chave primária**

- SELPROCFK: Varchar(10) – **chave-estrangeira** para a tabela SELECPROC, que é a tabela que contém toda a informação sobre os Processos de Selecção (não faz parte da chave primária)
- DESCR: Varchar(50) – campo que irá conter a palavra-chave
- FIELDNUM: Decimal (10,0) – campo numérico extra caso venha a ser necessário futuramente (regra adoptada pelo Departamento de Java)
- FIELDALPHA: Varchar(50) – campo alfanumérico extra caso venha a ser necessário futuramente (regra adoptada pelo Departamento de Java)

Camada de Persistência

Usando a *framework Persistence Builder* foram geradas as classes da camada de persistência que mapeiam a base de dados.

O modelo de classes do domínio a seguir (Figura 12) ilustra as relações entre as classes relevantes para a funcionalidade:

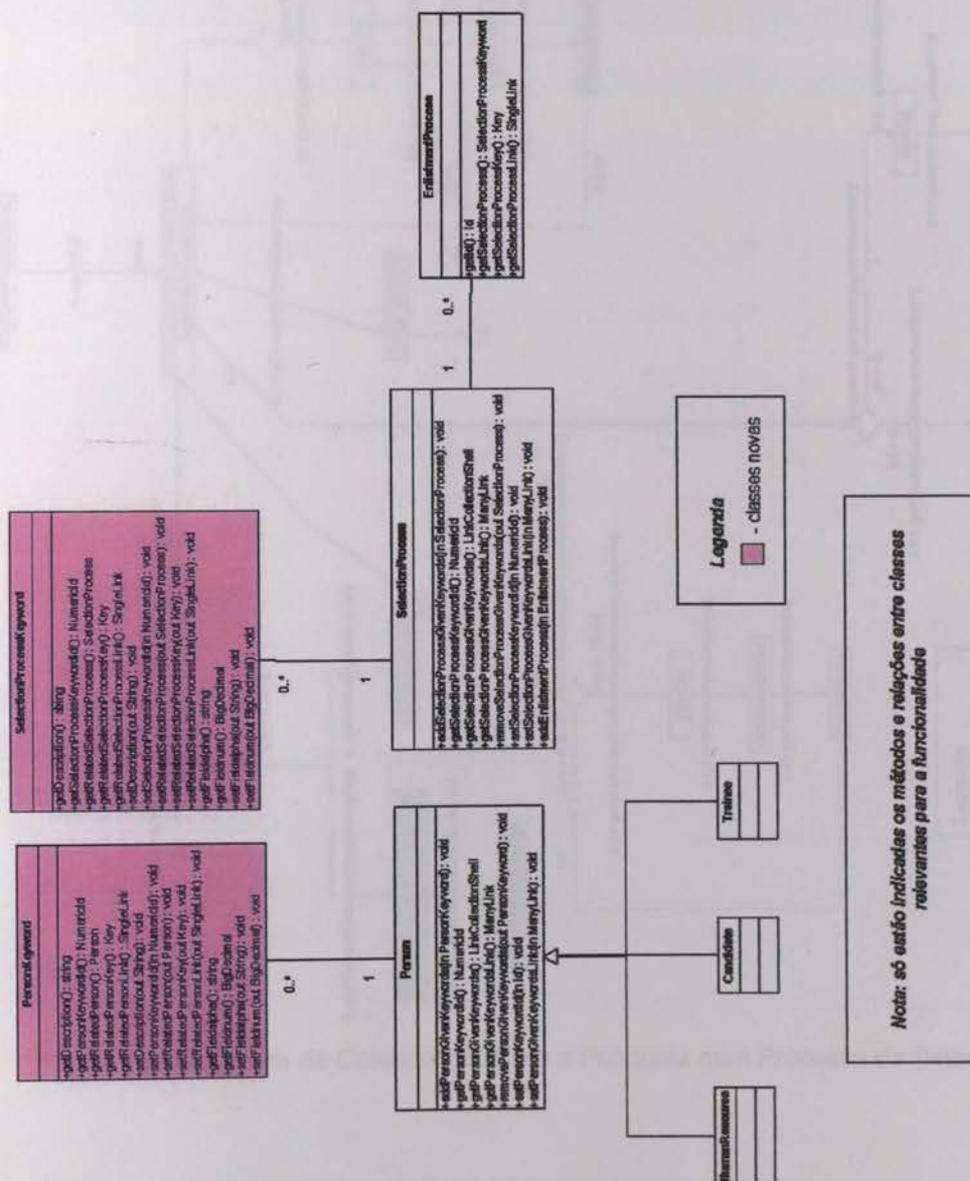


Figura 12 - Modelo de Classes do Domínio para a Pesquisa por Palavras-Chave

Controllers, Commands e Adapters

São de seguida apresentados dois diagramas de Colaboração: um diagrama para a pesquisa num Processo de Selecção (Figura 13) e outro para a pesquisa num Meio de Recrutamento (Figura 14) que retratam de forma clara onde e como foi integrada a Pesquisa por Palavras-Chave.

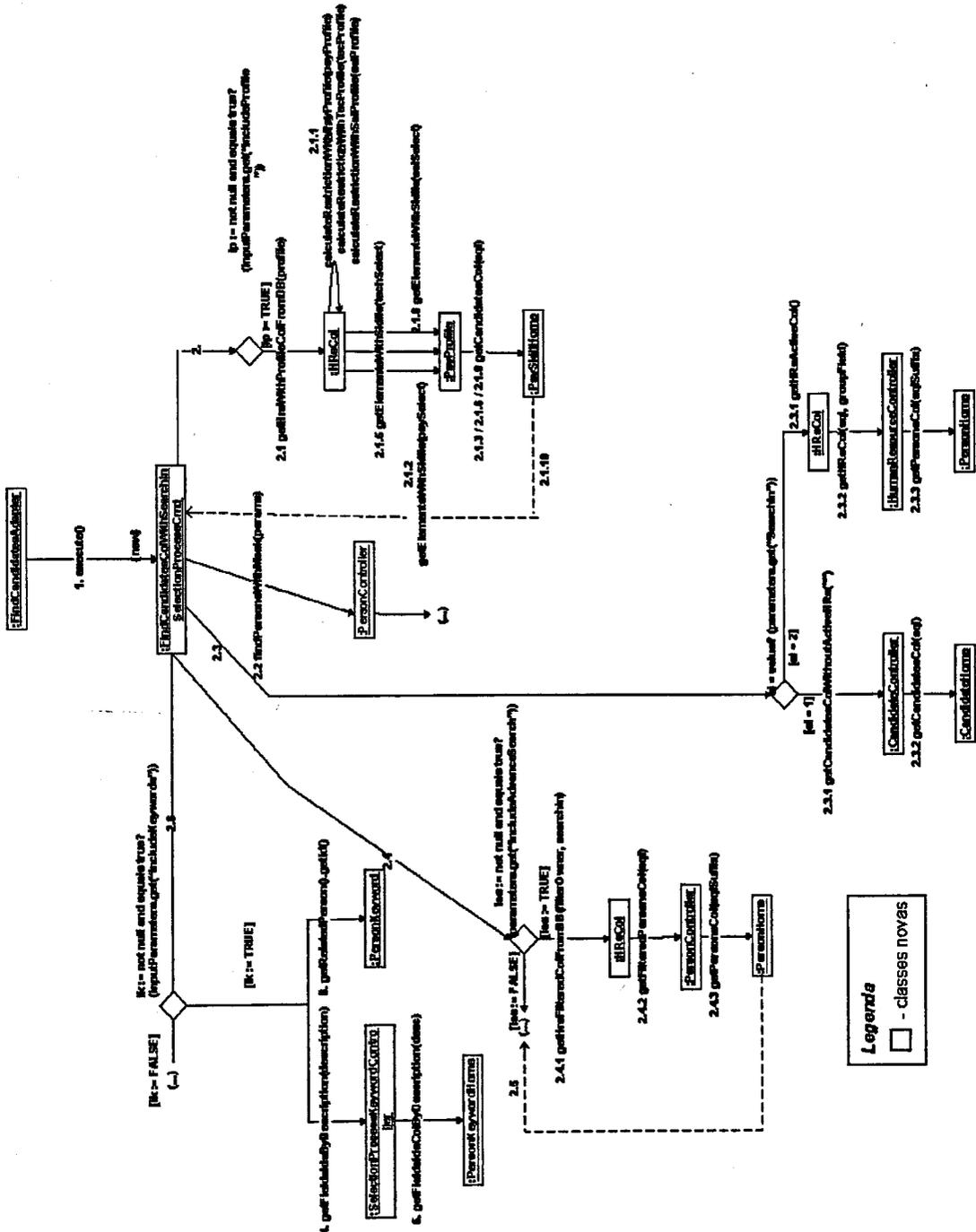


Figura 13 - Diagrama de Colaboração para a Pesquisa num Processo de Selecção

O algoritmo utilizado para a Pesquisa por Palavra-Chave, ou seja, para a pesquisa de Pessoas de um universo determinado, na base de dados, às quais estejam associadas pelo menos as mesmas palavras que estão associadas a um determinado Processo de Selecção, encontra-se ilustrado de uma forma simples no diagrama de actividades a seguir (Figura 15).

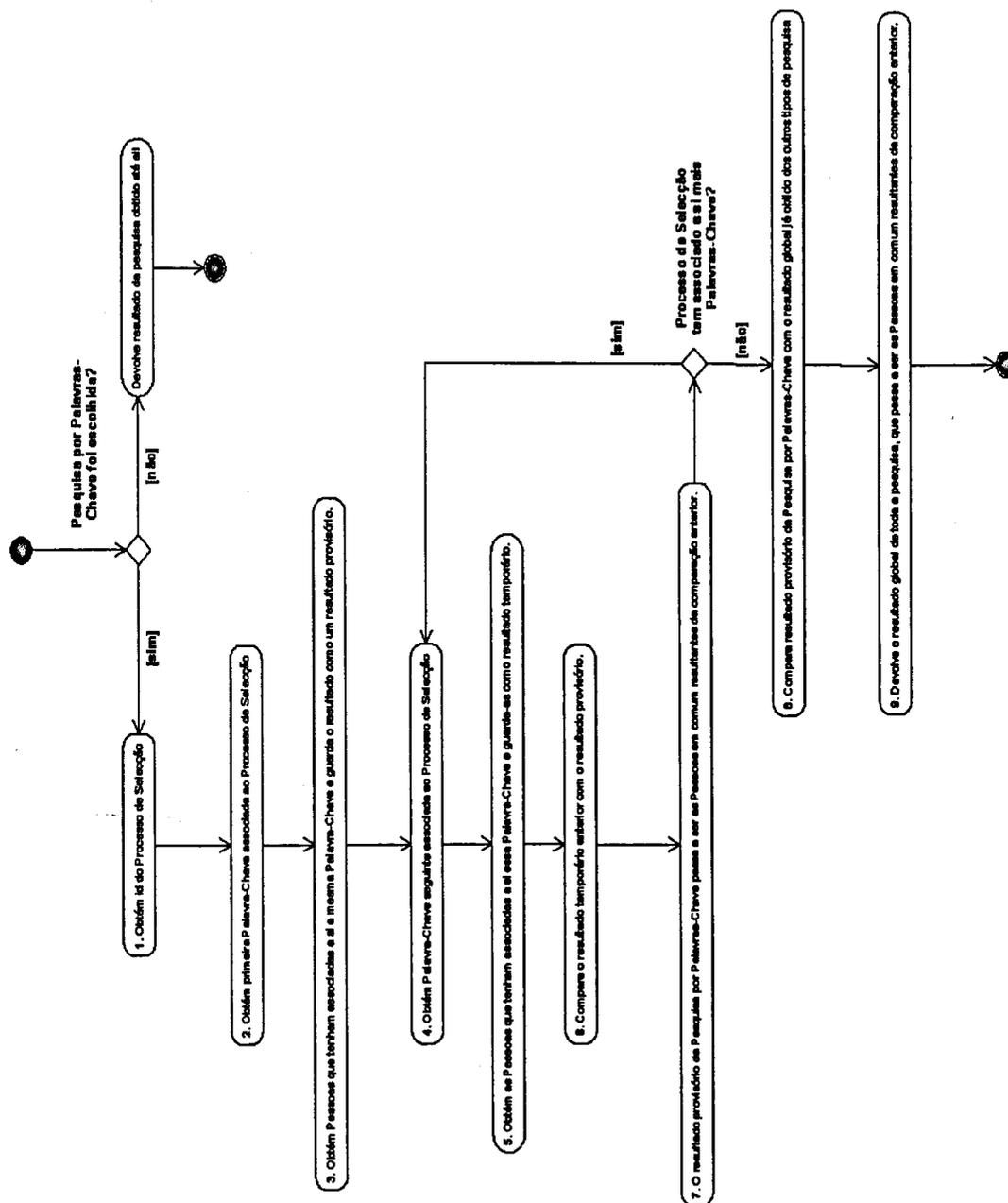


Figura 15 - Diagrama de Actividades do Algoritmo Utilizado na Pesquisa por Palavras-Chave

4.4 Principais Dificuldades Encontradas

A implementação desta funcionalidade foi o desafio mais exigente do estágio pois tinha um prazo muito bem definido, era uma funcionalidade crítica e urgente e foi o primeiro desenvolvimento que fiz dentro da aplicação ao mesmo tempo que explorava a arquitectura, as ferramentas e o código. Pode dizer-se que foi uma tarefa com o intuito de servir como formação, mas sendo também algo sério e concreto com toda a pressão que isso pode provocar.

Devido ao facto de ser difícil, para quem chega ao Departamento de Java, sentir-se logo à vontade a fazer desenvolvimento num projecto como o HCTT, que já atingiu uma grande dimensão (sendo a integração de uma pessoa no projecto demorada), existiu a necessidade de iniciar a implementação da funcionalidade de Pesquisa por Palavras-Chave antes de ter um documento de especificação concluído, pois a elaboração desse documento antes do início da implementação poderia comprometer o prazo de finalização da funcionalidade que tinha sido combinado e que era pouco flexível. Esse documento de especificação, tal como ele é elaborado no Departamento, deve conter já alguma informação "baixo-nível" sobre algumas classes que devem ser implementadas (como, por exemplo, o nome dos *Commands* a serem criados e as acções que executam).

Como a minha entrada no Departamento era muito recente e a funcionalidade de Pesquisa por Palavras-Chave foi o primeiro desenvolvimento que executei para o projecto HCTT, constatei que seria difícil conseguir "antecipar" essa informação detalhada num documento sem nunca ter feito algum desenvolvimento no projecto anteriormente e, por conseguinte, optei por efectuar a implementação em primeiro lugar, tendo que analisar detalhadamente o código já produzido na aplicação para me inteirar dos detalhes da aplicação e ir desenvolvendo o meu próprio código à medida que ia explorando o projecto e, no final, fazer engenharia inversa (elaborando a especificação depois da implementação).

Outra dificuldade prendeu-se com a geração de código automática, mais especificamente com a *framework Persistence Builder*, que gera as classes da camada de persistência. A geração de código automática pode ser interessante e útil se as ferramentas utilizadas para tal tiverem uma boa documentação para o utilizador e se este "tiver noção" do que está a fazer. No início da implementação da funcionalidade de Pesquisa por Palavras-Chave tive que dispensar muito tempo na geração de código automática para a camada de persistência pois surgiram alguns problemas com o código gerado automaticamente - problemas sobre os quais só se consegue perceber a origem depois de várias tentativas e várias experiências com a ferramenta, uma vez que a documentação desta ferramenta é pobre.

Por último, também existiram algumas dificuldades no controlo de navegação entre janelas para as páginas criadas para a funcionalidade. A navegação entre páginas no HCTT é controlado por funções *JavaScript*, responsáveis por executarem *reload* de páginas, abrir janelas, fechar janelas, etc. Um dos pontos mais demorados quando se faz desenvolvimento no HCTT é, precisamente, conseguir fazer um uso correcto das imensas funções de *JavaScript* já criadas para a aplicação, pois a documentação sobre estas funções é inexistente e o comportamento delas não é estável. Exemplo: uma função *JavaScript*, desenvolvida para o HCTT, responsável por fechar uma janela e fazer *reload* à página anterior, pode ter um comportamento perfeitamente estável num módulo do HCTT e não ter noutro, sendo difícil e demorado descobrir as razões para tal.

4.5 Avaliação de Resultados e Conclusões

Tendo sido este o maior desafio do estágio, pois foi a tarefa que exigiu mais responsabilidade e provocou maior pressão, uma vez que a funcionalidade era urgente e tinha um prazo específico, e tendo sido também a primeira experiência de desenvolvimento dentro do projecto HCTT, é reconfortante concluir que os objectivos foram atingidos com sucesso. No fim do prazo estipulado a funcionalidade estava implementada, embora faltassem alguns pormenores que não eram críticos, como, por exemplo, uma perfeita navegação entre páginas,

uma correcta apresentação, em termos gráficos, do resultado da pesquisa, e o documento de especificação. A tarefa, a par com o *MethoDoc*, demonstrou-se fundamental para a minha formação dentro do Departamento e para a minha experiência como elemento da equipa de desenvolvimento do HCTT.

5 A Normalização Gráfica

A Normalização Gráfica do HCTT constituiu a maior fase do estágio, em termos de tempo dispensado.

A necessidade de efectuar uma normalização não só gráfica, mas também técnica (ao nível do código da aplicação) foi surgindo ao longo do desenvolvimento do HCTT pois não existiu uma preocupação, por parte dos programadores que já passaram pela equipa de desenvolvimento, em manter uma grande coerência em termos de informação apresentada ao utilizador, acções existentes, navegação, bem como reaproveitamento de código para páginas do HCTT semelhantes (reaproveitamento ao nível dos *Adapters* utilizados).

O anexo H "Análise Técnica e Funcional das Páginas com Listagens de Pessoas", em volume aparte, apresenta um estudo elaborado no início desta fase de estágio, o qual consistiu numa análise técnica (rastreamento de processos XML, *Adapters*, *Commands* e *Controllers*) e funcional às páginas que apresentam listagens de Pessoas (Colaboradores, Formandos, Candidatos, Responsáveis, etc.) do HCTT.

5.1 Análise do Problema da Normalização Gráfica

Inicialmente foram definidas três etapas para esta fase embora tivesse sido também determinado que seria muito difícil estas três etapas serem completamente concluídas no tempo de estágio.

As três etapas definidas para esta fase são descritas de seguida.

Etapa 1: Normalização / *standardização* do *layout* e da navegação em páginas complexas.

Em páginas mais complexas do HCTT, tipicamente páginas de pesquisa, a lista dos resultados da pesquisa, bem como o formulário de pesquisa a preencher, estão muito desordenados no que diz respeito à sua posição na página. Pretendia-se normalizar os *layouts* de cada tipo de páginas complexas, identificando o que é comum para todos os módulos e o que é diferente de módulo para módulo, de modo a que o *layout* destas páginas fosse alterado para que a interacção com o utilizador fosse mais simples e intuitiva. A um nível mais técnico pretendia-se também alterar a forma como a pesquisa de Pessoas na base de dados estava implementada, caso se verificasse necessário. Esta fase comportava, então, os seguintes passos:

- Rastreamento de todas as páginas do HCTT onde existissem listagens de Pessoas (Candidatos, Colaboradores, Formandos, etc.) da base de dados.
- Estudo das páginas identificadas nesse rastreamento:
 - Estudo das funcionalidades disponíveis para o utilizador em cada página identificada no rastreamento (ordenação dos resultados das pesquisas, restrição dos universos de pesquisa, paginação dos resultados das pesquisas).
 - Estudo da forma como estão implementadas essas funcionalidades (quais os *Adapters*, *Commands*, *Controllers*, como está organizado / reaproveitado o código).
 - Apresentação de sugestões concretas de alteração.
- Implementação das alterações sugeridas no estudo.

Pretendia-se também nesta etapa implementar alterações a oito páginas críticas no HCTT onde a informação apresentada é muito extensa (listagens com informação muito diferente

entre si e, na maior parte das vezes, cada listagem continha um elevado número de registos) e, conseqüentemente, demoram muito tempo até serem carregadas e apresentadas no *browser*. A navegação nestas páginas não se efectuava da melhor forma, uma vez que se tinha optado por apresentar toda essa informação numa só página em vez de utilizar *tabs* categorizados (separadores), onde em cada *tab* estaria disponível um conjunto de informação restrita somente relacionada com a categoria apresentada. Com o uso de *tabs* categorizados a informação passaria a estar melhor organizada, a usabilidade destas páginas melhoraria e não haveria necessidade de carregar da base de dados grandes quantidades de registos para apresentação numa só página. Era necessário, então, implementar a navegação através de *tabs* categorizados nessas oito páginas críticas.

Etapa 2: Criação / estipulação de regras no *MethoDoc* para o *layout* de páginas e a informação presente em cada página.

Transcrição, no *MethoDoc*, das regras identificadas durante a elaboração do trabalho da etapa anterior e também das regras entretanto estipuladas pelo Departamento.

Etapa 3: Normalização / *standardização* das páginas de todos os módulos do HCTT de forma a implementarem as regras criadas / estipuladas na etapa anterior.

Esta fase da Normalização Gráfica incidia, principalmente, na normalização das páginas de todos os módulos do HCTT no que diz respeito a menus que devem estar presentes em cada página, posição de tabelas de informação na página, posição de botões na página, legenda de botões e nomes de menus, de forma a espelharem as regras criadas / estipuladas na etapa anterior.

5.2 Descrição da Solução para a Normalização Gráfica

Das três etapas apresentadas anteriormente foi executada a primeira etapa durante o tempo de estágio e existiu também uma alteração, após o estudo das páginas do HCTT com listagens de Pessoas, dos objectivos desta etapa. Inicialmente, pretendia-se que esta etapa se focasse essencialmente nas páginas de pesquisa do HCTT, mas verificou-se que as páginas de Responsáveis por Processos de Selecção, Estruturas e Nós eram mais críticas e necessitavam de uma intervenção mais urgente, bem como as páginas de listagem e pesquisa de Colaboradores Activos que pudessem ser seleccionados como Responsáveis.

Assim sendo, o trabalho efectivo realizado durante a fase de Normalização Gráfica foi:

- Levantamento de todas as páginas com listagens de Pessoas (quer fossem Colaboradores, Candidatos ou Formandos) do HCTT e estudo das alterações, quer a nível funcional e técnico, nas páginas que se revelassem mais críticas após o estudo. O documento contendo este estudo encontra-se como anexo a este relatório.
- Implementação das alterações sugeridas no estudo anterior, que se focaram nas páginas com listagens de Responsáveis de várias entidades (Processos de Selecção, Nós e Estruturas) em três módulos da aplicação (Recrutamento, Base – Estrutura e Gestão de Recursos) e nas páginas de listagem e pesquisa de Colaboradores Activos possíveis de serem seleccionados como Responsáveis.
- Colocação de oito páginas críticas do HCTT divididas em *tabs* (separadores) categorizados.

O estudo das páginas com listagens de Pessoas revelou que havia uma grande falta de reaproveitamento de código nas páginas de listagens de Colaboradores Activos, pois ao longo

do desenvolvimento da aplicação tinham sido criados JSPs diferentes, que chamavam *Adapters* diferentes e *Commands* diferentes mas que todos faziam exactamente o mesmo: seleccionavam, da base de dados, os Colaboradores Activos. Foi então aproveitada a última implementação diferente desenvolvida para efectuar esta listagem, que passou a ser a implementação utilizada nas páginas onde se podiam escolher novos Responsáveis por Processos de Selecção no módulo de Recrutamento, e por Nós e Estruturas no módulo de Gestão de Recursos.

O estudo também revelou que as próprias listagens de Responsáveis poderiam ser implementadas utilizando um só processo XML e um só *Adapter* ao qual seriam passados os *Commands* específicos de listagem de Responsáveis para cada entidade (Processo de Selecção, Estrutura ou Nó). Esses *Commands* poderiam, depois, ser invocados no *Adapter* utilizando o mecanismo de *reflection* que a linguagem Java oferece. Todas estas alterações foram efectuadas, tendo também incluído no *MethoDoc* uma secção que pretende ser um manual para os restantes programadores da equipa de desenvolvimento onde explico, passo a passo, e recorrendo a exemplos, como implementar listagens de Responsáveis e Colaboradores Activos em qualquer local no HCTT recorrendo às classes e processos XML por mim criados para esse efeito.

A forma como foram implementadas as alterações acima mencionadas é descrita em pormenor na secção seguinte deste relatório, intitulada “Implementação da Normalização Gráfica”.

A última parte da fase de Normalização Gráfica, que consistiu na colocação de *tabs* categorizados em oito páginas do HCTT revelou-se simples, pois esta forma de apresentação de informação já tinha sido utilizada nos módulos mais recentes da aplicação, não constituindo uma novidade e não exigindo uma investigação da forma como seria implementada. De qualquer forma, os passos a seguir na colocação de *tabs* categorizados em páginas do HCTT não estavam documentados em lado algum, tendo eu também incluído, no *MethoDoc*, uma secção onde os explico e exemplifico. Os resultados conseguidos com esta alteração são demonstrados também na secção seguinte, intitulada “Implementação da Normalização Gráfica”.

5.3 Implementação da Normalização Gráfica

O mecanismo de *reflection* em Java foi uma peça fundamental para o trabalho efectuado durante a fase de Normalização Gráfica. De seguida são descritos os pormenores da implementação das alterações às listagens de Responsáveis e, seguidamente, das listagens de Colaboradores Activos.

Por último, fornece-se um exemplo das alterações efectuadas às oito páginas críticas do HCTT que foram divididas em *tabs* (separadores) categorizados.

Listagens de Responsáveis

Foi criado novo código para as listagens de Responsáveis das entidades de Processos de Selecção (módulo de Recrutamento), Estruturas e Nós (módulo de Base-Estrutura e módulo de Gestão de Recursos) de forma a construir um processo XML e um *Adapter* “genéricos” que pudessem ser constantemente reutilizados não só para estas listagens mas também para qualquer outra listagem de Responsáveis de qualquer outra entidade do HCTT.

Antes da implementação destas alterações, os processos XML e os *Adapters* envolvidos nessas listagens eram vários, como se pode observar no quadro-resumo da Figura 16.

Método	Entidade	Informação Fornecida	Listagens de Responsáveis					
			Processo XML	Página	Adapter	Comandos		
						Seleção	Criação	Eliminação
Recrutamento	Processo de Selecção	Nome	Recrutment.xml	RecrutmentRSPOWNER_MLM	Select onProcessOwnersAdapter	GetSelectonProcessOwnersCmd	CrSelectonProcessOwnersCmd	DisSelectonProcessOwnersCmd
Bases - Estruturas	Estrutura	Por funcionário	ST.xml	STSTOWNERVIEW_MLM	base(,_)BSSTOwnerAdapter	base(,_)GetSTOwnerCmd	base(,_)CrSTOwnerCmd	base(,_)DisSTOwnerCmd
	Nó		STNode.xml	STNodeSTNODEOWNERVIEW_MLM	base(,_)BSSNodeOwnerAdapter	base(,_)GetSTNodeOwnerCmd	base(,_)CrSTNodeOwnerCmd	base(,_)DisSTNodeOwnerCmd
Gestão de Recursos	Estrutura	Nome	CRST.xml	CRSTOWNER_MLM	resourcecontrol(,_)BSSTOwnerAdapter	resourcecontrol(,_)GetSTOwnerCmd	resourcecontrol(,_)CrSTOwnerCmd	resourcecontrol(,_)DisSTOwnerCmd
	Nó	Caracterização por	CRSTNode.xml	CRSTNodeSTNODEOWNER_MLM	resourcecontrol(,_)BSSNodeOwnerAdapter	resourcecontrol(,_)GetSTNodeOwnerCmd	resourcecontrol(,_)CrSTNodeOwnerCmd	resourcecontrol(,_)DisSTNodeOwnerCmd
Legenda:								
áreas onde não incidir as alterações								

Figura 16 – Quadro-Resumo dos Aspectos Técnicos e Funcionais das Listagens de Responsáveis Antes das Alterações

A informação fornecida sobre cada Responsável de um Processo de Selecção residia somente no nome do Responsável, enquanto que para os Responsáveis de Estruturas e Nós era fornecido, além do nome, o número de funcionário e o seu *nickname*. Consequentemente, existiu aqui uma necessidade de normalizar a informação fornecida sobre cada Responsável de qualquer entidade, de forma a ser sempre exactamente a mesma.

A um nível técnico, era sempre gerada uma página nova, cada uma por um processo XML diferente, e implementado um *Adapter* novo de cada vez que fosse necessário listar Responsáveis. Existia aqui, então, uma grande falta de reaproveitamento de código.

Com o intuito de resolver os problemas acima apontados, implementei um processo XML denominado *HCTTOwners.xml* que passou a ser utilizado para a listagem de Responsáveis das entidades referidas no quadro-resumo da Figura 16. Este processo XML invoca os métodos de um só *Adapter*, denominado *OwnersAdapter*. Para resolver o problema de invocação dos *Commands* referentes a cada entidade, uma vez que estes não devem ser centralizados, foi então usado o mecanismo de *reflection*.

O quadro-resumo da Figura 17 ilustra as alterações efectuadas à forma como são, agora, listados Responsáveis.

Listagens de Responsáveis								
Módulo	Estado	Informação Fornecida	Processo XML	Páginas	Adapter	Comandos associados por Reflection		
						Seleção	Criação	Eliminação
Recrutamento	Processo de Seleção	NP Funcionário	HCTTOwners.xml	HCTTOwners/OWNERS_UI	OwnersAdapter	GetSelectorProcessOwnersCmd	CrSelectorProcessOwnerCmd	DISelectorProcessOwnerCmd
Base - Estruturas	Estados					base(.)GetSTOWnersCmd	base(.)CrSTOWnerCmd	base(.)DISTOWnerCmd
	Nó					base(.)GetSTNodeOwnersCmd	base(.)CrSTNodeOwnerCmd	base(.)DISTNodeOwnerCmd
Gestão de Recursos	Estados	Característica(s) por	HCTTOwners.xml	HCTTOwners/OWNERS_UI	OwnersAdapter	resourcescontrol(.)GetSTNodeOwnersCmd	resourcescontrol(.)CrSTNodeOwnerCmd	resourcescontrol(.)DISTNodeOwnerCmd
	Nó					resourcescontrol(.)GetSTNodeOwnersCmd	resourcescontrol(.)CrSTNodeOwnerCmd	resourcescontrol(.)DISTNodeOwnerCmd
Legenda:								
áreas alteradas								

Figura 17 – Quadro-Resumo dos Aspectos Técnicos e Funcionais das Listagens de Responsáveis Depois das Alterações

Nota-se de imediato o decréscimo de processos XML e de *Adapters* necessários para listar Responsáveis, pois a nova forma de implementação destas listagens teve como principal objectivo o reaproveitamento de código através da implementação de um processo XML e de um *Adapter* “genéricos” que pudessem ser utilizados para a listagem de Responsáveis de qualquer entidade.

No capítulo 3 do *MethoDoc*, contido no anexo F “MethoDoc”, pode ser encontrado um manual que contém os detalhes de implementação destas alterações às listagens de Responsáveis na secção 3.6.1 “Listagens de Responsáveis”. O código-fonte desenvolvido para o processo *HCTTOwners.xml* e para o *Adapter OwnersAdapter* também se encontra anexo a este relatório.

Listagens de Colaboradores Activos

Foi criado novo código para as listagens de Colaboradores Activos possíveis de serem seleccionados como Responsáveis das entidades de Processos de Selecção (módulo de Recrutamento), Estruturas e Nós (módulo de Base-Estrutura e módulo de Gestão de Recursos) de forma a construir um processo XML “genérico” que pudessem ser constantemente reutilizados não só para estas listagens mas também para qualquer outra listagem do mesmo tipo para qualquer outra entidade do HCTT. Neste caso, não foi necessária a criação de um *Adapter* também “genérico”, uma vez que para a listagem de Colaboradores Activos foi reutilizado um *Adapter* já implementado, denominado *ActiveHRsAdapter*. Para a acção de adição de um novo Responsável escolhido da listagem de Colaboradores Activos é utilizado o *OwnersAdapter* já referido anteriormente, que invoca por *reflection* o Comando de criação de um novo Responsável.

Antes da implementação das alterações, os processos XML e, conseqüentemente, as páginas geradas a partir desses processos, envolvidos nas listagens de Colaboradores Activos eram vários, como se pode observar no quadro-resumo da Figura 18.

Listagens de Colaboradores Activos possíveis de serem seleccionados como Responsáveis						
Módulo	Entidade	Informação Fornecida	Processo XML	Página	Adaptor	Comando de Seleção
Recrutamento	Processo de Seleção	Nº funcionário	Recruitment.xml	RecruitmentRESPONDER	IBSSTHumanResourceAdapter	GetHRColCmd
Base - Estrutura	Estrutura		ST.xml	STHUMANRESOURCE_LM		
	Nó	STNode.xml	STNodeHUMANRESOURCE_LM			
Gestão de Recursos	Estrutura	Nome	OPST.xml	OPSTHUMANRESOURCE_LM		
	Nó		OPSTNode.xml	OPSTNodeHUMANRESOURCE_LM		
Conhecido(s) por						
Legenda:						
áreas onde irão incidir as alterações						

Figura 18 – Quadro-Resumo dos Aspectos Técnicos e Funcionais das Listagens de Colaboradores Activos Antes das Alterações

Mais uma vez evidencia-se a falta de reaproveitamento de código ao nível dos processos XML e, conseqüentemente, das páginas geradas através desses processos.

Com o intuito de resolver este problema implementei um processo XML denominado *HCTTPersonsLists.xml* que passou a ser utilizado para a listagem de Colaboradores Activos possíveis de serem seleccionados como Responsáveis das entidades referidas no quadro-resumo da Figura 18.

O quadro-resumo seguinte (Figura 19) ilustra as alterações efectuadas à forma como são, agora, efectuadas essas listagens.

A colocação de uma página única de HCTT dividida em sete tabs teve como fim primordial o aumento da usabilidade desta página para o utilizador.

Esta são páginas únicas com:

- Módulo de Recrutamento

- Página de detalhes de um Processo de Seleção
- Página de detalhes de um Módulo de Recrutamento

- Módulo de Base - Recursos

- Página de detalhes de uma Estrutura
- Página de detalhes de um Nó
- Página de detalhes de uma Função

- Módulo de Gestão de Recursos

- Página de detalhes de uma Estrutura
- Página de detalhes de um Nó
- Página de detalhes de uma Função

Listagens de Colaboradores Activos possíveis de serem seleccionados como Responsáveis						
Módulo	Entidade	Informação Fornecida	Processo XML	Página	Adaptar	Comando de Selecção
Recrutamento	Processo de Selecção					
Base - Estrutura	Estrutura	Nº funcionário	HCTTPersonsLists.xml	HCTTPersonsLists ACTIVEHRS_LM	ActiveHRSAdapter	GetActiveHrsColCmd
	Nó	Nome				
Gestão de Recursos	Estrutura					
	Nó	Conhecido(a) por				
Legenda:						
áreas alteradas						

Figura 19 – Quadro-Resumo dos Aspectos Técnicos e Funcionais das Listagens de Colaboradores Activos depois das Alterações

Nota-se também de imediato o decréscimo de processos XML necessários para listar Colaboradores Activos, pois a nova forma de implementação destas listagens teve como principal objectivo o reaproveitamento de código através da implementação de um processo XML “genérico” que pudesse ser utilizados para a listagem de Colaboradores Activos possíveis de serem seleccionados como Responsáveis de qualquer entidade.

No capítulo 3 do *MethoDoc*, contido no anexo F "MethoDoc", pode ser encontrado um manual que contém os detalhes de implementação destas alterações às listagens de Colaboradores Activos na secção 3.6.2 “Listagens de Colaboradores Activos”. O código-fonte desenvolvido para o processo *HCTTPersonsLists.xml* também se encontra anexo a este relatório.

Páginas divididas em *tabs* (separadores) categorizados

A colocação de oito páginas críticas do HCTT dividida em *tabs* teve como fim primordial o aumento da usabilidade destas páginas para o utilizador.

Estas oito páginas críticas foram:

- Módulo de Recrutamento
 - Página de detalhes de um Processo de Selecção
 - Página de detalhes de um Meio de Recrutamento
- Módulo de Base- Estrutura
 - Página de detalhes de uma Estrutura
 - Página de detalhes de um Nó
 - Página de detalhes de uma Função
- Módulo de Gestão de Recursos
 - Página de detalhes de uma Estrutura
 - Página de detalhes de um Nó
 - Página de detalhes de uma Função

De forma a ilustrar o impacto das alterações efectuadas a estas páginas é demonstrado de seguida, a título de exemplo, uma espécie de “antes e depois” de uma das páginas. As alterações efectuadas para as restantes são em tudo semelhantes.

O exemplo escolhido é a página de detalhes de uma Estrutura do Módulo de Base- Estrutura (Figura 20):



Figura 20 – “Antes e Depois” da Página de Detalhes de uma Estrutura no Módulo de Base- Estrutura

Como se pode observar, antes do uso dos *tabs*, havia três tipos de informação diferentes numa só página: descrição da Estrutura, listagem da Missão e listagem de Responsáveis da Estrutura. Com o uso dos *tabs*, cada um destes tipos de informação ficou localizado num *tab*.

No capítulo 3 do *MethoDoc*, contido no anexo F “MethoDoc”, pode ser encontrado um manual que contém os detalhes de implementação de *tabs* em páginas do HCTT, na secção 3.7 “Implementação de *tabs* (separadores)”.

5.4 Principais Dificuldades Encontradas

Apesar de ser a fase mais longa do estágio foi aquela que menos dificuldades apresentou pois já tinha adquirido um bom domínio da aplicação com o desenvolvimento da funcionalidade de Pesquisa por Palavras-Chave.

A maior dificuldade a salientar prendeu-se, de novo, com a navegação entre páginas, também já referida como uma dificuldade aquando da implementação da Pesquisa por Palavras-Chave.

5.5 Avaliação de Resultados e Conclusões

A primeira etapa do trabalho efectuado na fase de Normalização Gráfica, que consistiu no levantamento e estudo de todas as páginas com listagens de Pessoas (quer fossem Colaboradores, Candidatos ou Formandos) do HCTT revelou-se trabalhosa, uma vez que foi necessária uma exploração exaustiva de toda a aplicação de modo a “descobrir” todas as páginas com listagens de Pessoas.

A par com o *MethoDoc*, esta fase de Normalização Gráfica revelou-se um dos desafios mais interessantes durante o estágio, no que diz respeito à normalização das listagens de Responsáveis e Colaboradores Activos, pois a forma como essas listagens passaram a estar implementadas é inovadora e exigiu, de certa forma, o uso do pensamento de “engenheiro”

(um pensamento inventivo, criativo e crítico com o objectivo de solucionar os problemas da forma mais simples possível) que tanto nos é transmitido ao longo da nossa formação universitária e, ao mesmo tempo, facilitará muito o trabalho dos restantes programadores quando houver a necessidade de implementar listagens de Responsáveis para outras entidades do HCTT. As alterações efectuadas a estas listagens permitiu um grande decréscimo do número de processos XML e de *Adapters* necessários para as executar, enfatizando o reaproveitamento de código e a escrita de código tendo em mente esse objectivo.

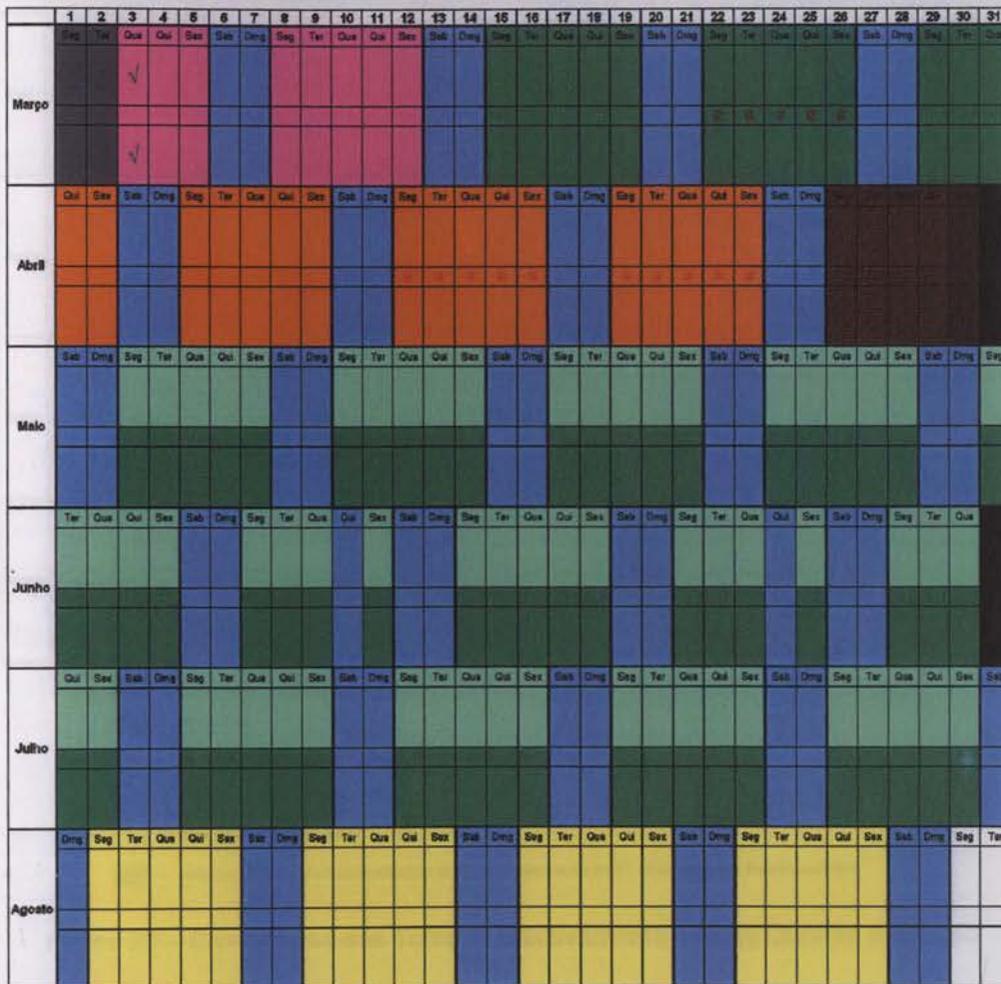
No que diz respeito às páginas alteradas de modo a ficarem divididas em *tabs*, isto já não constituiu uma novidade pois já tinha sido efectuado em módulos mais recentes da aplicação, no caso do módulo de Formação e no módulo de Segurança e Higiene no Trabalho, mas é, actualmente, uma boa forma de aumentar a usabilidade da aplicação.

6 Análise ao Planeamento de Estágio

Este capítulo pretende efectuar uma análise mais detalhada ao planeamento de estágio, demonstrando os atrasos e antecipações ocorridos e explicando as suas razões. No fim, são ilustradas estatísticas simples relativamente aos tempos de estágio previstos e efectivamente dedicados a cada fase do estágio.

6.1 Desvios Verificados ao Planeamento Inicial

O planeamento inicial do estágio está ilustrado no cronograma da Figura 21:



- Formação
- MethoDec
- Funcionalidade do HCTT - Pesquisa por Palavras-Chave
- Normalização Gráfica do HCTT
- Relatório e Finalização
- Semana crítica para implementação da funcionalidade do HCTT - Pesquisa por Palavras-Chave
- Gonçalo Mata estará ausente

Figura 21 – Cronograma do Planeamento Inicial de Estágio [repetição]

Os tempos efectivamente dedicados a cada tarefa durante o estágio estão ilustrados no cronograma da Figura 22.

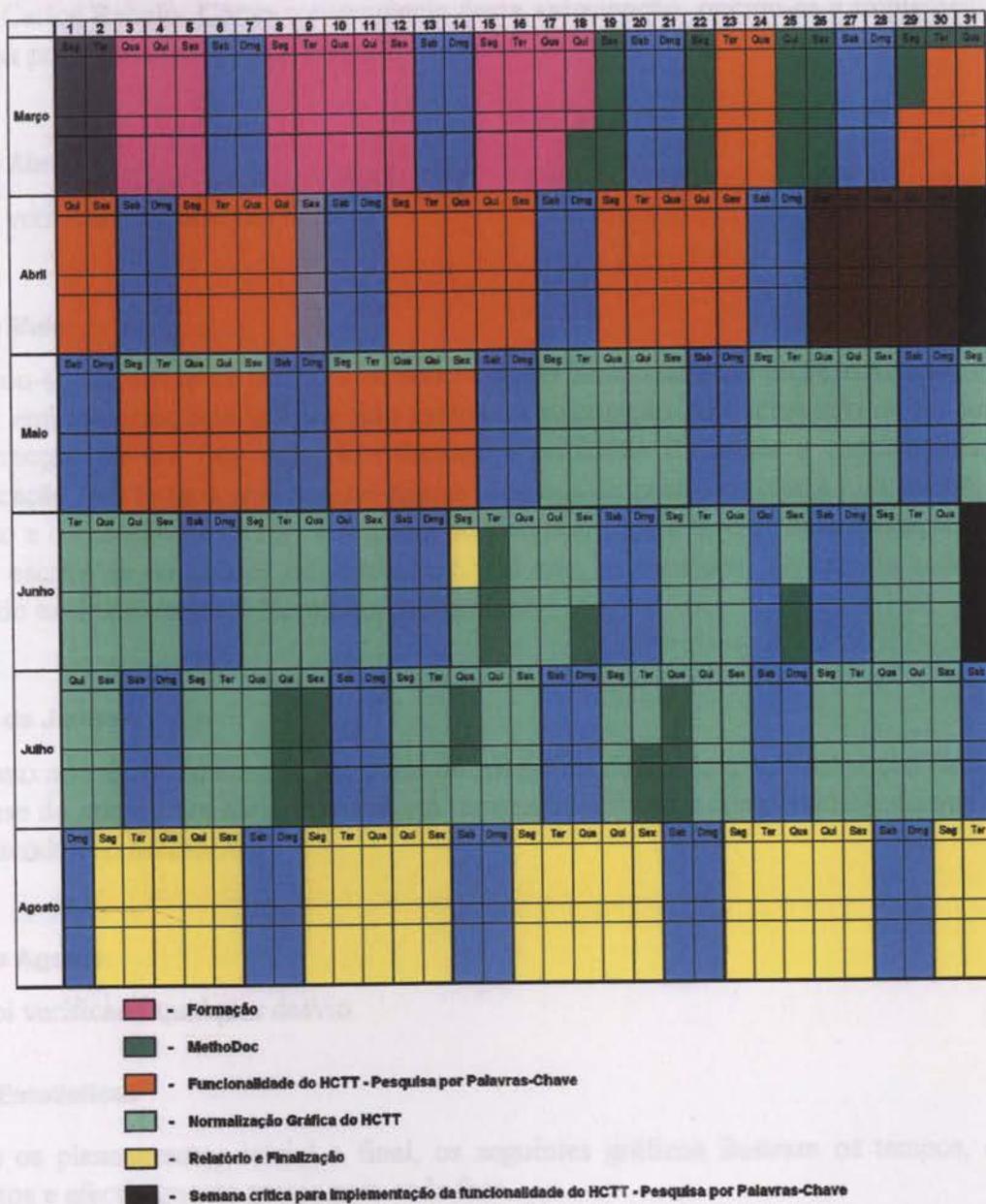


Figura 22 – Cronograma dos Tempos Efectivamente Gastos Durante o Estágio

Mês de Março

Verificou-se que o período de Formação se estendeu para além do previsto devido à complexidade do HCTT, ao tempo dispensado a explorar ferramentas de trabalho, à realização de manuais e guiões de instalação e configuração de *software*, à análise e estudo de documentação importante sobre algumas tecnologias utilizadas e também porque foi dispensado algum tempo na implementação de uma classe, em conjunto com o meu colega de estágio Carlos Rebelo, denominada *BundlesCollector* que tinha como objectivo centralizar os *bundles* de tradução, espalhados por diferentes directórios de um dos projectos do Eclipse, para um directório único (esta tarefa era executada manualmente por cada programador de cada vez que fosse necessário correr o *GuiGen* para a geração automática de JSPs).

Também se verificou que o período dispensado para a realização da estrutura do *MethoDoc* foi mais curto, pois já havia alguma documentação que foi aproveitada para essa tarefa e porque existiu uma boa conjugação de tarefas e um bom ritmo de trabalho entre mim e o meu colega Carlos Rebelo. Como consequência desta antecipação, iniciou-se a implementação da Pesquisa por Palavras-Chave mais cedo.

Mês de Abril

Não se verificaram quaisquer desvios.

Mês de Maio

Verificou-se um atraso de dez dias na conclusão da funcionalidade de Pesquisa por Palavras-Chave, embora estes dias tenham sido gastos na elaboração do documento de especificação, na correcção de um *bug* detectado durante a execução de testes à funcionalidade e na modificação, em termos gráficos, de alguns aspectos da pesquisa. Neste mês também estava previsto a dedicação de tempo, em iguais proporções, para a fase de Normalização Gráfica e para a escrita de conteúdos no *MethoDoc*. Tal não se verificou, tendo sido todo o tempo dedicado exclusivamente à Normalização Gráfica.

Meses de Junho e Julho

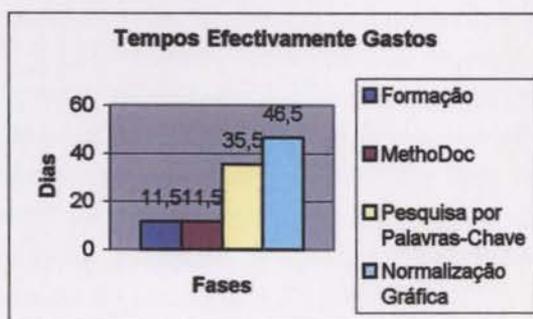
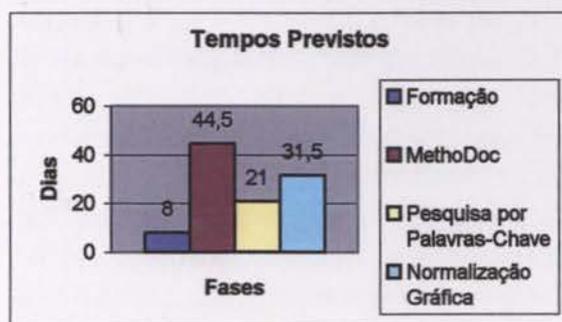
Tal como no mês de Maio, a maior parte do tempo foi dedicado à Normalização Gráfica, pois essa fase de estágio era mais exigente em termos de esforço e complexidade do que a escrita de conteúdos no *MethoDoc*.

Mês de Agosto

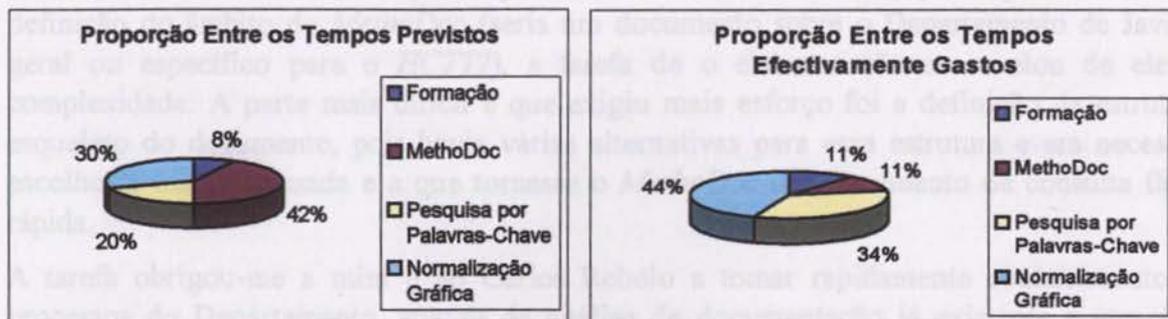
Não foi verificado qualquer desvio.

6.2 Estatísticas

Dados os planeamentos inicial e final, os seguintes gráficos ilustram os tempos, em dias, previstos e efectivamente gastos para cada fase.



A proporção entre os tempos previstos e efectivamente gastos para cada fase pode ser mais facilmente observada nos seguintes gráficos, que fornecem uma vista diferente dos gráficos anteriores.



Estava previsto uma maior dedicação de tempo ao *MethoDoc*, principalmente no que diz respeito à escrita de conteúdos. De facto, só os dois primeiros capítulos do *MethoDoc* ficaram completos durante o tempo de estágio, estando o terceiro capítulo a meio de se considerar também completo. A fase de Normalização Gráfica revelou-se bastante exigente e, conseqüentemente, parte do tempo previsto para o *MethoDoc* foi transferido para as tarefas exigidas pela Normalização Gráfica.

A Pesquisa por Palavras-Chave não sofreu um desvio muito relevante.

7 Conclusões e Perspectivas de Trabalho Futuro

Relativamente ao *MethoDoc*, após as primeiras dificuldades iniciais, que se prenderam com a definição do âmbito do *MethoDoc* (seria um documento sobre o Departamento de Java em geral ou específico para o HCTT?), a tarefa de o elaborar não se revelou de elevada complexidade. A parte mais difícil e que exigiu mais esforço foi a definição da estrutura / esqueleto do documento, pois havia várias alternativas para essa estrutura e era necessário escolher a mais adequada e a que tornasse o *MethoDoc* um documento de consulta fácil e rápida.

A tarefa obrigou-me a mim e ao Carlos Rebelo a tomar rapidamente conhecimento dos processos do Departamento, através da análise de documentação já existente e através de entrevistas a alguns elementos da equipa, o que se revelou uma vantagem para a nossa integração na equipa e para a rápida evolução do nosso desempenho dentro dela. Foi, em suma, um trabalho a executar que também assumiu o papel de formação.

Existem, no entanto, algumas reflexões / preocupações a salientar e que se prendem com uma realidade muito simples e facilmente compreendida: a integração do *MethoDoc* na rotina da equipa de desenvolvimento, aproveitando todas as vantagens que um documento como este pode trazer, não será uma tarefa fácil e trivial. Se existe uma vontade efectiva de criar e manter este documento é então necessário que todos tenham a consciência que isso implicará executar tarefas que terão que ser efectivamente realizadas por todos e não só por alguns. Para que a consulta do *MethoDoc* e a escrita de conteúdos nele seja algo natural em qualquer um dos elementos da equipa do Departamento será então necessário envolvê-los nesta ambição para que o documento não seja encarado como uma ideia utópica ou teórica de outros.

Futuramente, o *MethoDoc* deixará de ser um documento em formato *Word* e orientado às fases da Engenharia de *Software* e passará a ser um documento em formato *Wiki Web*, com todas as vantagens que isso pode trazer (maior facilidade em acrescentar, consultar e organizar conteúdos, maior facilidade em reestruturar o documento) e orientado aos processos do Departamento. Por exemplo, em vez de existir um tópico chamado “Análise / Especificação” passará a haver tópicos como “Elaborar um documento de especificação”, “Elaborar um documento com o *layout* das interfaces”, “Preencher um pedido técnico”, etc.

No que diz respeito à Pesquisa por Palavras-Chave, foi este o maior desafio do estágio, pois foi a tarefa que exigiu mais responsabilidade e provocou maior pressão (uma vez que a funcionalidade era urgente e tinha um prazo específico) e foi também a primeira experiência de desenvolvimento dentro do projecto HCTT. É reconfortante concluir que os objectivos foram atingidos com sucesso. No fim do prazo estipulado a funcionalidade estava implementada, embora faltassem alguns pormenores que não eram críticos, como, por exemplo, uma perfeita navegação entre páginas, uma correcta apresentação, em termos gráficos, do resultado da pesquisa, e o documento de especificação. A tarefa, a par com o *MethoDoc*, demonstrou-se fundamental para a minha formação dentro do Departamento e para a minha experiência como elemento da equipa de desenvolvimento do HCTT.

Ainda sobre o documento de especificação da Pesquisa por Palavras-Chave, devido ao facto de ser difícil, para quem chega ao Departamento de Java, sentir-se logo à vontade a fazer desenvolvimento num projecto como o HCTT, que já atingiu uma grande dimensão, sendo a integração de uma pessoa no projecto demorada, existiu a necessidade de iniciar a

implementação da funcionalidade de Pesquisa por Palavras-Chave antes de ter um documento de especificação concluído, pois a elaboração desse documento antes do início da implementação poderia comprometer o prazo de finalização da funcionalidade que tinha sido combinado e que era pouco flexível. Esse documento de especificação, tal como ele é elaborado no Departamento, deve conter já alguma informação "baixo-nível" sobre algumas classes que devem ser implementadas (como, por exemplo, o nome dos *Commands* a serem criados e as acções que executam).

Como a minha entrada no Departamento era muito recente e a funcionalidade de Pesquisa por Palavras-Chave foi o primeiro desenvolvimento que executei para o projecto HCTT, constatei que seria difícil conseguir "antecipar" essa informação "baixo-nível" num documento sem nunca ter feito algum desenvolvimento no projecto anteriormente e, por conseguinte, optei por efectuar a implementação em primeiro lugar (tendo que analisar detalhadamente o código já produzido na aplicação para me inteirar dos "meandros" da aplicação e ir desenvolvendo o meu próprio código à medida que ia explorando o projecto) e, no final, fazer *reverse engineering* (elaborando a especificação depois da implementação).

Outra dificuldade prendeu-se com a geração de código automática, mais especificamente com a *framework Persistence Builder* (que gera as classes da camada de persistência). A geração de código automática pode ser interessante e útil se as ferramentas utilizadas para tal tiverem uma boa documentação para o utilizador e se este "tiver noção" do que está a fazer. No início da implementação da funcionalidade de Pesquisa por Palavras-Chave tive que dispensar muito tempo na geração de código automática para a camada de persistência pois surgiram alguns problemas com o código gerado automaticamente - problemas sobre os quais só se consegue perceber a origem depois de várias tentativas e várias experiências com a ferramenta (uma vez que a documentação desta ferramenta é pobre).

Por último, também existiram algumas dificuldades no controlo de navegação entre janelas para as páginas criadas para a funcionalidade. A navegação entre páginas no HCTT é controlado por funções *JavaScript*, responsáveis por executarem *reload* de páginas, abrir janelas, fechar janelas, etc. Um dos pontos mais demorados quando se faz desenvolvimento no HCTT é, precisamente, conseguir fazer um uso correcta das imensas funções de *JavaScript* já criadas para a aplicação, pois a documentação sobre estas funções é inexistente e o comportamento delas não é estável (exemplo: uma função *JavaScript*, desenvolvida para o HCTT, responsável por fechar uma janela e fazer *reload* à página anterior pode ter um comportamento perfeitamente estável num módulo do HCTT e não ter noutro, sendo difícil e demorado descobrir as razões para tal).

Finalmente, resta a fase de Normalização Gráfica. A primeira etapa do trabalho efectuado na fase de Normalização Gráfica, que consistiu no levantamento e estudo de todas as páginas com listagens de Pessoas (quer fossem Colaboradores, Candidatos ou Formandos) do HCTT revelou-se trabalhosa, uma vez que foi necessária uma exploração exaustiva de toda a aplicação de modo a "descobrir" todas as páginas com listagens de Pessoas.

A par com o *MethoDoc*, esta fase de Normalização Gráfica revelou-se um dos desafios mais interessantes durante o estágio, no que diz respeito à normalização das listagens de Responsáveis e Colaboradores Activos, pois a forma como essas listagens passaram a estar implementadas é inovadora e exigiu, de certa forma, o uso do pensamento de "engenheiro" (um pensamento inventivo, criativo e crítico com o objectivo de "dar a volta" aos problemas da forma mais simples possível) que tanto nos é transmitido ao longo da nossa formação

universitária e, ao mesmo tempo, facilitará muito o trabalho dos restantes programadores quando houver a necessidade de implementar listagens de Responsáveis para outras entidades do HCTT. As alterações efectuadas a estas listagens permitiu um grande decréscimo do número de processos XML e de *Adapters* necessários para as executar, enfatizando o reaproveitamento de código e a escrita de código tendo em mente esse objectivo.

No que diz respeito às páginas alteradas de modo a ficarem divididas em *tabs*, isto já não constituiu uma novidade pois já tinha sido efectuado em módulos mais recentes da aplicação (no caso do módulo de Formação e no módulo de Segurança e Higiene no Trabalho) mas é, actualmente, uma boa forma de aumentar a usabilidade da aplicação.

Apesar de ser a fase mais longa do estágio foi aquela que menos dificuldades apresentou pois já tinha adquirido um bom domínio da aplicação com o desenvolvimento da funcionalidade de Pesquisa por Palavras-Chave.

A maior dificuldade a salientar prendeu-se, de novo, com a navegação entre páginas, também já referida como uma dificuldade aquando da implementação da Pesquisa por Palavras-Chave.

Futuramente, resta executar duas etapas na Normalização Gráfica: transcrição, no *MethoDoc*, das regras identificadas durante a elaboração do trabalho até aqui efectuado e também das regras entretanto estipuladas pelo Departamento e normalização / *standardização* das páginas de todos os módulos do HCTT de forma a implementarem as regras criadas / estipuladas.

Em suma, o estágio foi concluído com sucesso e a IBS Portugal forneceu e facilitou tudo o que fosse necessário para tal. Além disso, o estágio constituiu uma excelente oportunidade para conhecer de perto como se faz desenvolvimento de *software* a um nível industrial, bem diferente do nível académico vivido ao longo dos cinco anos de licenciatura.

<<http://java.sun.com/developer/technicalArticles/ALJ/Reflection/>>. Acesso em 19 Agosto 2004.

Referências e Bibliografia

- [1] IBS PORTUGAL. **Homepage**. Disponível em: <<http://www.ibs.pt>>. Acesso em: 2 Agosto 2004.
- [2] IBS PORTUGAL. **HCTT – Human Capital Team Tool: Functional and Technical Overview**. [S.l], 2004.
- [3] IBS. **Client Framework User Guide**. [S.l], Maio 2000. Revisto Janeiro 2001.
- [4] IBS. **GuiGen – Generator of JSP Files User Guide**. [S.l], Maio 2001.
- [5] IBM CORPORATION. **Building Applications With Persistence Support**. [S.l], 2000.
- [6] THE APACHE SOFTWARE FOUNDATION. **The Jakarta Site – Apache Jakarta Tomcat**. Disponível em < <http://jakarta.apache.org/tomcat/>>. Acesso em 3 Agosto 2004.
- [7] IBM CORPORATION. **Using a Web Application Server To Provide Flexible And Scalable E-Business Solutions**. [S.l], Abril 2002.
- [8] ECLIPSE FOUNDATION. **Eclipse.org**. Disponível em: <<http://www.eclipse.org>>. Acesso em 19 Agosto 2004.
- [9] CVS is a version control system. **CVS – Concurrent Versions System**. Disponível em < http://www.loria.fr/~molli/cvs/doc/cvs_1.html>. Acesso em 19 Agosto 2004.
- [10] MCCLUSKEY, Glen. **Using Java Reflection**. Disponível em <<http://java.sun.com/developer/technicalArticles/ALT/Reflection/>>. Acesso em 19 Agosto 2004.

Glossário

CGI – Common Gateway Interface.

GUI – Graphical User Interface.

HTTP – HyperText Transfer Protocol.

IDE – Integrated Development Environment.

JDBC – Java Database Connectivity.

J2EE – Java 2 Platform Enterprise Edition.

JSP – JavaServer Pages.

XML – Extended Markup Language.

At last, the third and longest phase of the internship consisted in the Graphical Normalization of HCTT and this implied, mainly, the implementation of "specific" XML processes and Java classes in the Adapter layer, intended to facilitate the issuing of code in the Drivers and Active Human Resources lists. Still during this phase it was performed some changes in eight critical pages of HCTT in a way that the information there related, visually, would be divided in tabs, for best usability.

The internship was successfully concluded and was a good opportunity to have a first contact with software development at an industrial level, which is quite different from the academic level.

ANEXO A: Resumo em Inglês

The internship “Development Methodologies and Graphical Normalization of the Human Capital Team Tool” took part in the IBS Portugal facilities, in Vila Nova de Gaia, and was based on the *Human Capital Team Tool* (HCTT) application, oriented to Human Resources management, developed under the *Java 2 Platform Enterprise Edition* technology in the Java Department of that institution.

The internship was divided in three phases. The first phase was dedicated to the execution of the *MethoDoc*, a kind of Quality Manual holding all the know-how of the development team and the rules for the development of HCTT and this demanded a quick integration in the team and in the Department processes so it could be a successful task.

The second phase consisted on the development of a new feature in HCTT called Search by Keywords. This task demanded a great involvement and exploration of the application architecture and code.

At last, the third and longest phase of the internship consisted in the Graphical Normalization of HCTT and this implied, mainly, the implementation of “generic” XML processes and Java classes in the *Adapters* layer, intended to facilitate the reusing of code in the Owners and Active Human Resources lists. Still during this phase it was performed some changes in eight critical pages of HCTT in a way that the information there holded originally would be divided in tabs, for best usability.

The internship was sucessfully concluded and was a good opportunity to have a first contact with software development at an industrial level, which is quite different from the academical level.

ANEXO B: Código-fonte do Processo HCTTOwners.xml

```

<?xml version="1.0"?>
<!DOCTYPE process SYSTEM "ibsprocess.dtd">
<process name="HCTTOwners">
  <page id="OWNERS_LM" url="/jsp/UserLocaleHere/HCTTOwners/OWNERS_LM.jsp"
default_bundle="IBSOPUI">
  <descriptive_data description="idSTOWNER_MLM"/>
  <title id="idSTOwnerListTitle" bundle="IBSOPUI"/>
  <image id="idImage" bundle="IBSRCUI"/>
  <view name="view_list_maintenance"/>
  <input>
    <input_parameter id="EntityId1" key="True" />
    <input_parameter id="EntityId2" key="True" />
    <input_parameter id="EntityId3" key="True" />
    <input_parameter id="EntityId4" key="True" />
    <input_parameter id="EntityId5" key="True" />
    <input_parameter id="PageReload"/>
    <input_parameter id="Commands"/>
  </input>
  <page_data element_count="-1">
    <parameter id="HRId" key="True"/>
    <parameter id="HRIndividualNumber" label="idHRIndividualNumber"
bundle="IBSHRUI" length_bundle="IBSHRUI"/>
    <parameter id="HRAllName" label="idHRName" bundle="IBSHRUI"
length_bundle="IBSHRUI"/>
    <parameter id="HRName" label="idHRNickName" bundle="IBSHRUI"
length_bundle="IBSHRUI"/>
  </page_data>
  <actions>
    <display view="view_list_maintenance_createOwnersOneAction">
      <action name="action_dbclick" value="idDbClick"
page_id="OWNER_DDV_DIFFWINDOW" bundle="IBSRCUI" selection="1">
        <hook event_name="onDbClick">
          <event_method name="setTargetOpenCenteredWindow" url="">
            <event_method_parameter name="document.forms[0]"/>
            <event_method_parameter name="'OWNERDELETE'"/>
            <event_method_parameter name="''"/>
            <event_method_parameter name="70"/>
            <event_method_parameter name="65"/>
            <event_method_parameter name="true"/>
          </event_method>
        </hook>
      </action>
      <output>
        <output_parameter id="EntityId1" />
        <output_parameter id="EntityId2" />
        <output_parameter id="EntityId3" />
        <output_parameter id="EntityId4" />
        <output_parameter id="EntityId5" />
        <output_parameter id="Commands"/>
        <output_parameter id="HRId"/>
      </output>
    </display>
    <action name="action_deletex" value="idDelete"
page_id="OWNER_DDV_DIFFWINDOW" bundle="IBSRCUI" selection="1">
      <hook event_name="onClick">
        <event_method name="setTargetOpenCenteredWindow" url="">
          <event_method_parameter name="document.forms[0]"/>
          <event_method_parameter name="'OWNERDELETE'"/>
          <event_method_parameter name="''"/>
          <event_method_parameter name="70"/>
        </event_method>
      </hook>
    </action>
  </actions>
</page>
</process>

```

```

    <event_method_parameter name="65"/>
  </event_method_parameter name="true"/>
</event_method>
</hook>
</output>
  <output_parameter id="EntityId1" />
  <output_parameter id="EntityId2" />
  <output_parameter id="EntityId3" />
  <output_parameter id="EntityId4" />
  <output_parameter id="EntityId5" />
  <output_parameter id="Commands"/>
  <output_parameter id="HRId"/>
</output>
</action>
<action name="action_create" value="idCreate"
page_id="HCTTPersonsLists|ACTIVEHRS_ONEACTION_MV" bundle="IBSRCUI"
selection="0">
  <hook event_name="onClick">
    <event_method name="hr_list" url="">
      <event_method_parameter name="this.form"/>
      <event_method_parameter name="'NEWSTOWNER'"/>
      <event_method_parameter name="'action_create'"/>
    </event_method>
  </hook>
  <output>
    <output_parameter id="EntityId1" />
    <output_parameter id="EntityId2" />
    <output_parameter id="EntityId3" />
    <output_parameter id="EntityId4" />
    <output_parameter id="EntityId5" />
    <output_parameter id="Commands"/>
  </output>
</action>
</display>
<display view="view_list maintenance_createOwnersTwoActions">
  <action name="action_dblick" value="idDblick"
page_id="OWNER_DDV_SAMEWINDOW" bundle="IBSRCUI" selection="1">
  <hook event_name="onClick">
    <event_method name="setWindowFlag" url=""/>
  </hook>
  <output>
    <output_parameter id="EntityId1" />
    <output_parameter id="EntityId2" />
    <output_parameter id="EntityId3" />
    <output_parameter id="EntityId4" />
    <output_parameter id="EntityId5" />
    <output_parameter id="Commands"/>
    <output_parameter id="HRId"/>
  </output>
</action>
  <action name="action_deletex" value="idDelete"
page_id="OWNER_DDV_SAMEWINDOW" bundle="IBSRCUI" selection="1">
  <hook event_name="onClick">
    <event_method name="setWindowFlag" url=""/>
  </hook>
  <output>
    <output_parameter id="EntityId1" />
    <output_parameter id="EntityId2" />
    <output_parameter id="EntityId3" />
    <output_parameter id="EntityId4" />
    <output_parameter id="EntityId5" />
    <output_parameter id="Commands"/>
    <output_parameter id="HRId"/>
  </output>
</action>

```

```

    <event_method_parameter name="65"/>
  </event_method_parameter name="true"/>
</event_method>
</hook>
</output>
<output_parameter id="EntityId1" />
<output_parameter id="EntityId2" />
<output_parameter id="EntityId3" />
<output_parameter id="EntityId4" />
<output_parameter id="EntityId5" />
<output_parameter id="Commands"/>
<output_parameter id="HRId"/>
</output>
</action>
<action name="action_create" value="idCreate"
page_id="HCTTPersonsLists|ACTIVEHRS_ONEACTION_MV" bundle="IBSRCUI"
selection="0">
  <hook event name="onClick">
    <event_method name="hr_list" url="">
      <event_method_parameter name="this.form"/>
      <event_method_parameter name="'NEWSTOWNER'"/>
      <event_method_parameter name="'action_create'"/>
    </event_method>
  </hook>
  <output>
    <output_parameter id="EntityId1" />
    <output_parameter id="EntityId2" />
    <output_parameter id="EntityId3" />
    <output_parameter id="EntityId4" />
    <output_parameter id="EntityId5" />
    <output_parameter id="Commands"/>
  </output>
</action>
</display>
<display view="view_list_maintenance_createOwnersTwoActions">
  <action name="action_dbclick" value="idDbClick"
page_id="OWNER_DDV_SAMEWINDOW" bundle="IBSRCUI" selection="1">
  <hook event name="onClick">
    <event_method name="setWindowFlag" url="">
  </hook>
  <output>
    <output_parameter id="EntityId1" />
    <output_parameter id="EntityId2" />
    <output_parameter id="EntityId3" />
    <output_parameter id="EntityId4" />
    <output_parameter id="EntityId5" />
    <output_parameter id="Commands"/>
    <output_parameter id="HRId"/>
  </output>
</action>
<action name="action_deletex" value="idDelete"
page_id="OWNER_DDV_SAMEWINDOW" bundle="IBSRCUI" selection="1">
  <hook event name="onClick">
    <event_method name="setWindowFlag" url="">
  </hook>
  <output>
    <output_parameter id="EntityId1" />
    <output_parameter id="EntityId2" />
    <output_parameter id="EntityId3" />
    <output_parameter id="EntityId4" />
    <output_parameter id="EntityId5" />
    <output_parameter id="Commands"/>
    <output_parameter id="HRId"/>
  </output>

```


ANEXO C: Código-fonte da Classe do Adapter OwnersAdapter.java

```

/*
 * Created on Jun 14, 2004
 *
 * To change the template for this generated file go to
 * Window>Preferences>Java>Code Generation>Code and Comments
 */
package pt.ibs.hctt.base.gui.adapters.hr;

import java.io.Serializable;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Locale;
import java.util.Map;
import java.util.StringTokenizer;
import java.util.Vector;

import pt.ibs.hctt.cf.cmds.CmdControl;
import pt.ibs.hctt.cf.control.GlobalExceptionInfo;
import pt.ibs.hctt.gui.utils.BaseHCTTAdapter;

import com.uniiface.util.ControlException;
import com.uniiface.util.UIProcessConstants;

/**
 * @author cfalcao
 *
 * To change the template for this generated type comment go to
 * Window>Preferences>Java>Code Generation>Code and Comments
 */
public class OwnersAdapter extends BaseHCTTAdapter {

    private final static boolean DEBUG = false;

    public Object create(Map environment, Map inputParameters, List createParameters)
    throws ControlException {
        String userId = (String) ((HashMap)
environment.get("UserParams")).get("Username");

        if ((DEBUG == true)) {
            System.out.println("+IBSRCSelectionProcessDefaultTestAdapter.create");
            System.out.println("inputParameters from create" + createParameters + "_");
        }
        if ((DEBUG == true))
            System.out.println("-IBSRCSelectionProcessDefaultTestAdapter.create");
        try {
            Map commands = getCommands(inputParameters);
            for (int i = 0; i < createParameters.size(); i++) {
                inputParameters.put("HRId", createParameters.get(i).toString());
                String cmdName = commands.get("CreateCommand").toString();
                CmdControl.invokeCmd(inputParameters, cmdName);
            }
            return null;
        }
        catch (Exception exc) {
            GlobalExceptionInfo res = new GlobalExceptionInfo();
            res.handleException(exc, (Locale)
environment.get(UIProcessConstants.ACTIVE_LOCALE));

```

```

    throw new ControlException(res.getExceptionName());
}
}

public Object create(Map environment, Map inputParameters, Map createParameters)
throws ControlException {
    String userId = (String) ((HashMap)
environment.get("UserParams")).get("Username");

    if ((DEBUG == true)) {
        System.out.println("+OwnersAdapter.create");
        System.out.println("inputParameters from create" + createParameters + "_");
    }
    if ((DEBUG == true))
        System.out.println("-OwnersAdapter.create");
    try {
        Map commands = getCommands(inputParameters);
        createParameters.putAll(inputParameters);
        String cmdName = commands.get("CreateCommand").toString();
        CmdControl.invokeCmd(createParameters, cmdName);
        return null;
    }
    catch (Exception exc) {
        GlobalExceptionInfo res = new GlobalExceptionInfo();
        res.handleException(exc, (Locale)
environment.get(UIProcessConstants.ACTIVE_LOCALE));
        throw new ControlException(res.getExceptionName());
    }
}

public Object update(Map environment, Map inputParameters, Map updateParameters)
throws ControlException {
    String userId = (String) ((HashMap)
environment.get("UserParams")).get("Username");
    if ((DEBUG == true)) {
        System.out.println("+OwnersAdapter.update");
        System.out.println("inputParameters from update_" + updateParameters + "_");
        System.out.println("-OwnersAdapter.update");
    }
    try {
        Map commands = getCommands(inputParameters);
        String cmdName = commands.get("UpdateCommand").toString();
        CmdControl.invokeCmd(updateParameters, cmdName);
        return null;
    }
    catch (Exception exc) {
        GlobalExceptionInfo res = new GlobalExceptionInfo();
        res.handleException(exc, (Locale)
environment.get(UIProcessConstants.ACTIVE_LOCALE));
        throw new ControlException(res.getExceptionName());
    }
}

public void delete(Map environment, Map inputParameters, Map deleteParameters)
throws ControlException {
    String userId = (String) ((HashMap)
environment.get("UserParams")).get("Username");
    if ((DEBUG == true)) {
        System.out.println("+OwnersAdapter.delete");
        System.out.println("inputParameters from delete" + deleteParameters + "_");
    }
    if ((DEBUG == true))

```

```

    System.out.println("-OwnersAdapter.delete");
    try {
        Map commands = getCommands(inputParameters);
        String cmdName = commands.get("DeleteCommand").toString();
        CmdControl.invokeCmd(inputParameters, cmdName);
    }
    catch (Exception exc) {
        GlobalExceptionInfo res = new GlobalExceptionInfo();
        res.handleException(exc, (Locale)
environment.get(UIProcessConstants.ACTIVE_LOCALE));
        throw new ControlException(res.getExceptionName());
    }
}

public List loadList(Map env, Map inputParameters, List attributesToLoad) throws
ControlException {
    String userId = (String) ((HashMap) env.get("UserParams")).get("Username");
    if ((DEBUG == true)) {
        System.out.println("OwnersAdapter.loadlist");
        System.out.println("inputParameters from loadlist" + inputParameters + "_");
        System.out.println("OwnersAdapter.loadlist");
    }
    try {
        Map commands = getCommands(inputParameters);
        String cmdName = commands.get("LoadListCommand").toString();
        List resultTemp = (List) CmdControl.invokeCmdWithResult(inputParameters,
cmdName);
        if (!resultTemp.isEmpty()) {
            ArrayList result = new ArrayList();
            result.addAll(resultTemp);
            return result;
        }
        else {
            return new ArrayList();
        }
    }
    catch (Exception exc) {
        GlobalExceptionInfo res = new GlobalExceptionInfo();
        res.handleException(exc, (Locale) env.get(UIProcessConstants.ACTIVE_LOCALE));
        throw new ControlException(res.getExceptionName());
    }
}

public Map loadObject(Map env, Map inputParameters, List attributesToLoad) throws
ControlException {
    String userId = (String) ((HashMap) env.get("UserParams")).get("Username");
    if ((DEBUG == true)) {
        System.out.println("+OwnersAdapter.loadobject");
        System.out.println("inputParameters from loadobject" + inputParameters + "_");
    }
    if ((DEBUG == true))
        System.out.println("-OwnersAdapter.loadobject");
    try {
        HashMap params = new HashMap();
        params.put("HRId", inputParameters.get("HRId"));
        pt.ibs.hctt.base.cmds.usr.hr.GetHRCmd cmd = new
pt.ibs.hctt.base.cmds.usr.hr.GetHRCmd(params, userId);
        cmd.execute();
        HashMap result = new HashMap();
        result.putAll((Map) cmd.getResult());
        inputParameters.put("HRName", ((pt.ibs.hctt.cf.types.Name)
result.get("HRName")).getShortName());
    }
}

```



```

<output_parameter id="EntityId3" />
<output_parameter id="EntityId4" />
<output_parameter id="EntityId5" />
<output_parameter id="Commands"/>
<output_parameter id="HRId" />
</output>
<method name="create" parameter_type="Map">
  <adapter class_name="pt.ibs.hctt.base.gui.adapters.hr.OwnersAdapter"
context=""/>
</method>
</action>
<action name="action_create_close" value="idOkBold"
page_id="ACTIVEHRS_ONEACTION_MV" bundle="IBSOPUI" selection="1">
  <hook event_name="onClick">
    <event_method name="setWindowFlag" url=""/>
  </hook>
<output>
  <output_parameter id="EntityId1" />
  <output_parameter id="EntityId2" />
  <output_parameter id="EntityId3" />
  <output_parameter id="EntityId4" />
  <output_parameter id="EntityId5" />
  <output_parameter id="Commands"/>
  <output_parameter id="HRId"/>
</output>
<method name="create" parameter_type="Map">
  <adapter class_name="pt.ibs.hctt.base.gui.adapters.hr.OwnersAdapter"
context=""/>
</method>
</action>
<action name="action_cancel_closeX" value="idCancel"
page_id="ACTIVEHRS_ONEACTION_MV" bundle="IBSOPUI" selection="0">
  <hook event_name="onClick">
    <event_method name="window.top.opener.top.dir_cima.location.reload" url=""/>
    <event_method name="window.close" url=""/>
  </hook>
</action>
</display>
<display view="view_list_maintenance_addOwnersTwoActions">
  <action name="action_dbclick" value="idDbClick"
page_id="ACTIVEHRS_TWOACTIONS_MV" bundle="IBSRCUI" selection="1">
  <hook event_name="onClick">
    <event_method name="setWindowFlag" url=""/>
  </hook>
<output>
  <output_parameter id="EntityId1" />
  <output_parameter id="EntityId2" />
  <output_parameter id="EntityId3" />
  <output_parameter id="EntityId4" />
  <output_parameter id="EntityId5" />
  <output_parameter id="Commands"/>
  <output_parameter id="HRId"/>
</output>
<method name="create" parameter_type="Map">
  <adapter class_name="pt.ibs.hctt.base.gui.adapters.hr.OwnersAdapter"
context=""/>
</method>
</action>
<action name="action_ok" value="idConfirm" page_id="ACTIVEHRS_TWOACTIONS_MV"
bundle="IBSRCUI" selection="1">
  <hook event_name="onClick">
    <event_method name="setWindowFlag" url=""/>
  </hook>

```

```

<output>
  <output_parameter id="EntityId1" />
  <output_parameter id="EntityId2" />
  <output_parameter id="EntityId3" />
  <output_parameter id="EntityId4" />
  <output_parameter id="EntityId5" />
  <output_parameter id="Commands"/>
  <output_parameter id="HRId"/>
</output>
<method name="create" parameter_type="Map">
  <adapter class_name="pt.ibs.hctt.base.gui.adapters.hr.OwnersAdapter"
context=""/>
  </method>
</action>
<action name="action_create_owner" value="idCreateOwner"
page_id="ACTIVEHRS_TWOACTIONS_MV" bundle="IBSRCUI" selection="AtLeastOne">
  <hook event_name="onClick">
    <event_method name="setWindowFlag" url=""/>
  </hook>
<output>
  <output_parameter id="EntityId1" />
  <output_parameter id="EntityId2" />
  <output_parameter id="EntityId3" />
  <output_parameter id="EntityId4" />
  <output_parameter id="EntityId5" />
  <output_parameter id="Commands"/>
</output>
<method name="create" parameter_type="List">
  <adapter class_name="pt.ibs.hctt.base.gui.adapters.hr.OwnersAdapter"
context=""/>
  </method>
</action>
<action name="action_back" value="idBack" page_id="" bundle="IBSRCUI"
selection="0">
  <hook event_name="onClick">
    <event_method name="setWindowFlag" url=""/>
  </hook>
</action>
</display>
<display view="view_list_maintenance_addOwnerByReference">
  <action name="action_dbclick" value="idDbClick" bundle="IBSRCUI"
selection="1">
    <hook event_name="onClick">
      <event_method name="setWindowFlag" url=""/>
    </hook>
    <output>
      <output_parameter id="EntityId1" />
      <output_parameter id="EntityId2" />
      <output_parameter id="EntityId3" />
      <output_parameter id="EntityId4" />
      <output_parameter id="EntityId5" />
      <output_parameter id="Commands"/>
      <output_parameter id="HRId" />
    </output>
  </action>
  <action name="action_ok" value="idConfirm" page_id="" bundle="IBSRCUI"
selection="1">
    <hook event_name="onClick">
      <event_method name="setWindowFlag" url=""/>
    </hook>
    <output>
      <output_parameter id="EntityId1" />
      <output_parameter id="EntityId2" />

```

```

<output>
  <output_parameter id="EntityId1" />
  <output_parameter id="EntityId2" />
  <output_parameter id="EntityId3" />
  <output_parameter id="EntityId4" />
  <output_parameter id="EntityId5" />
  <output_parameter id="Commands"/>
  <output_parameter id="HRId"/>
</output>
<method name="create" parameter_type="Map">
  <adapter class_name="pt.ibs.hctt.base.gui.adapters.hr.OwnersAdapter"
context=""/>
</method>
</action>
<action name="action_create_owner" value="idCreateOwner"
page_id="ACTIVEHRS_TWOACTIONS MV" bundle="IBSRCUI" selection="AtLeastOne">
  <hook event_name="onClick">
    <event_method name="setWindowFlag" url=""/>
  </hook>
<output>
  <output_parameter id="EntityId1" />
  <output_parameter id="EntityId2" />
  <output_parameter id="EntityId3" />
  <output_parameter id="EntityId4" />
  <output_parameter id="EntityId5" />
  <output_parameter id="Commands"/>
</output>
<method name="create" parameter_type="List">
  <adapter class_name="pt.ibs.hctt.base.gui.adapters.hr.OwnersAdapter"
context=""/>
</method>
</action>
<action name="action_back" value="idBack" page_id="" bundle="IBSRCUI"
selection="0">
  <hook event_name="onClick">
    <event_method name="setWindowFlag" url=""/>
  </hook>
</action>
</display>
<display view="view_list_maintenance_addOwnerByReference">
  <action name="action_dbclick" value="idDbClick" bundle="IBSRCUI"
selection="1">
    <hook event_name="onClick">
      <event_method name="setWindowFlag" url=""/>
    </hook>
    <output>
      <output_parameter id="EntityId1" />
      <output_parameter id="EntityId2" />
      <output_parameter id="EntityId3" />
      <output_parameter id="EntityId4" />
      <output_parameter id="EntityId5" />
      <output_parameter id="Commands"/>
      <output_parameter id="HRId" />
    </output>
  </action>
  <action name="action_ok" value="idConfirm" page_id="" bundle="IBSRCUI"
selection="1">
    <hook event_name="onClick">
      <event_method name="setWindowFlag" url=""/>
    </hook>
    <output>
      <output_parameter id="EntityId1" />
      <output_parameter id="EntityId2" />

```

```
<output_parameter id="EntityId3" />
<output_parameter id="EntityId4" />
<output_parameter id="EntityId5" />
<output_parameter id="Commands"/>
<output_parameter id="HRId" />
</output>
</action>
<action name="action_back" value="idBack" page_id="" bundle="IBSRCUI"
selection="0">
  <hook event_name="onClick">
    <event_method name="setWindowFlag" url=""/>
  </hook>
</action>
</display>
</actions>
<adapter class_name="pt.ibs.hctt.base.gui.adapters.hr.ActiveHRsAdapter"
context=""/>
</page>
</process>
```

ANEXO E: Logbook de Estágio

Semana 1

3 de Março de 2004 (4ª feira)

- Apresentação da empresa e suas normas.
- Visita às instalações.
- Reunião com Gonçalo Mata e Bárbara Costa onde foram definidos o trabalho a realizar durante o estágio, objectivos do estágio, planeamento do estágio e orientadores (o meu orientador é o Gonçalo Mata).
- Apresentação, por parte do Gonçalo Mata, do HCTT: arquitectura, funcionalidades, demonstração da aplicação, contextualização do trabalho de estágio a realizar dentro da aplicação.

4 de Março de 2004 (5ª feira)

- Estudo e análise de tecnologias: *UniIFace* (uma *framework* IBS para geração de interfaces gráficas a partir de XML).
- Exploração da rede interna da IBS (estrutura da rede, acesso ao *e-mail*, acesso à base de dados de endereços de *e-mail* de todos os colaboradores, acesso a calendários pessoais, acesso à *intranet*).
- Instalação e configuração do IDE *Eclipse* e do *Tomcat*.
- Exploração do repositório CVS.

5 de Março de 2004 (6ª feira)

- Exploração da aplicação HCTT (arquitectura, estrutura de directórios no *Eclipse*, módulos desenvolvidos e em desenvolvimento), da *framework UniIFace* e da integração do *UniIFace* no HCTT.
- Exploração do código-fonte do HCTT.

Semana 2

8 de Março de 2004 (2ª feira)

- Continuação da exploração da aplicação HCTT e sua arquitectura e da *framework UniIFace*.
- Exploração do *iSeries Access Navigator* (ferramenta para acesso / manipulação de base de dados) e do IDE *VisualAge*.
- Implementação de um exercício com a finalidade de gerar uma página básica do HCTT que obtém dados da base de dados e os lista no ecrã.

9 de Março de 2004 (3ª feira)

- Continuação da implementação do exemplo do dia anterior. Resolução de alguns problemas surgidos.
- Começo da instalação / configuração da minha máquina (que me foi entregue hoje).

10 de Março de 2004 (4ª feira)

- 1ª Reunião de Orientação de Estágio com os orientadores nas instalações da IBS.
- Conclusão da instalação / configuração da minha máquina.

11 de Março de 2004 (5ª feira)

- Análise de documentação sobre *Persistence Builder*
- *Checkout*, do CVS, do novo *branch* do HCTT e teste.
- Organização da documentação sobre o *MethoDoc* até agora reunida.
- Continuação da análise de documentação sobre *Persistence Builder*.
- Estudo e exploração da nova estrutura de directórios no *Eclipse* do HCTT.

12 de Março de 2004 (6ª feira)

- Continuação do estudo e exploração da nova estrutura de directórios no *Eclipse* do HCTT.
- Análise do código-fonte, em conjunto com o Carlos Rebelo, das classes do *package com.uniiface.tools.guiigen* do *UniFace* para perceber todo o funcionamento do *GuiGen* de modo a ser possível alterar a forma como são identificados os *Locales* e os *bundles* para esses *Locales*.
- Após conversa com Bárbara Costa, chegou-se à conclusão que o *UniFace* não deveria ser alterado, uma vez que está a ser recebida, de momento, uma nova versão que contempla já muitas mudanças no *package* referido acima. Assim sendo, optou-se por implementar uma classe nova em *IBS_Components*, chamada *BundlesCollector*, no projecto do HCTT, que copie os ficheiros dos *bundles* para um sítio centralizado em *HCTeamTool_Patterns*.
- Estudo da forma como se geram *javadocs*.
- Início da implementação da classe *BundlesCollector* acima referida, em conjunto com o Carlos Rebelo.

Semana 3

15 de Março de 2004 (2ª feira)

- Continuação da implementação da classe *BundlesCollector*, juntamente com o Carlos Rebelo.
- Reunião com todos os elementos do Departamento de Java cujo tema foi " *HCTT - Ponto da Situação / Circuito de Qualidade* ".
- Continuação da implementação da classe *BundlesCollector*.
- Reunião com o Gonçalo Mata de análise da funcionalidade do HCTT de Pesquisa por Palavras-Chave que terei que implementar.

16 de Março de 2004 (3ª feira)

- Começo da elaboração do documento de especificação da funcionalidade do HCTT de Pesquisa por Palavras-Chave (esta actividade foi mais tarde suspensa).

17 de Março de 2004 (4ª feira)

- Exploração da ferramenta *Rational Rose 2000 Enterprise Edition*.

- Conclusão com sucesso da implementação da classe *BundlesCollector*, juntamente com o Carlos Rebelo.

18 de Março de 2004 (5ª feira)

- Execução dos testes finais à classe *BundlesCollector*, juntamente com o Carlos Rebelo.
- Conclusão da realização da documentação (*Javadoc*) da classe implementada.
- Início da elaboração da 1ª versão do *MethoDoc* juntamente com o Carlos Rebelo: análise da documentação reunida.

19 de Março de 2004 (6ª feira)

- Continuação da elaboração da 1ª versão do *MethoDoc* juntamente com o Carlos Rebelo:
- Continuação da análise da documentação reunida.
- Elaboração de um *template* de documentos *Word*.
- Elaboração de uma primeira estrutura do *MethoDoc*.
- Definição da metodologia a adoptar na elaboração e evolução do *MethoDoc*.

Semana 4

22 de Março de 2004 (2ª feira)

- Elaboração de um guião, juntamente com o Carlos Rebelo, com a descrição dos procedimentos a seguir pelos elementos da equipa de desenvolvimento após fazerem *checkout* do CVS da nova classe *BundlesCollector*, de modo a que esta classe seja efectivamente utilizada por toda a equipa.
- Reunião semanal do Departamento de Java.
- Sincronização, no *Eclipse*, dos projectos do HCTT com o repositório CVS e resolução de problemas daí surgidos.
- Continuação da elaboração da 1ª versão do *MethoDoc*.

23 de Março de 2004 (3ª feira)

- Migração do exercício executado há 2 semanas (cuja finalidade era gerar uma página básica do HCTT que obtinha dados da base de dados e os listava no ecrã) para a nova estrutura de directórios no *Eclipse* do HCTT. Resolução de problemas surgidos.

24 de Março de 2004 (4ª feira)

- Continuação da elaboração da 1ª versão do *MethoDoc*.
- Conclusão da migração do exercício executado há 2 semanas (cuja finalidade era gerar uma página básica do HCTT que obtinha dados da base de dados e os listava no ecrã) para a nova estrutura de directórios no *Eclipse* do HCTT, em conjunto com o Carlos Rebelo.
- Exploração, através de experiências, do modo como os JSP, os processos XML, os *adapters* e os *bundles* influenciam a aplicação e a interface gráfica, juntamente com o Carlos Rebelo.

25 de Março de 2004 (5ª feira)

- Continuação da elaboração da 1ª versão do *MethoDoc* juntamente com o Carlos Rebelo, que incluiu duas reuniões com a Lígia Almeida (uma delas contou também com a presença do Sr. Ernesto Nogueira).

26 de Março de 2004 (6ª feira)

- Reunião com o Célio Rodrigues sobre o *MethoDoc*.
- Reunião com a Bárbara Costa sobre o *MethoDoc*.
- Conclusão da elaboração da 1ª versão do *MethoDoc* juntamente com o Carlos Rebelo.

Semana 5

29 de Março de 2004 (2ª feira)

- Pesquisa na Java API com o objectivo de descobrir uma classe que implemente a verificação de *strings* utilizando expressões regulares (necessário para a implementação da funcionalidade de Pesquisa por Palavras-Chave) - essa classe é a classe *Pattern*.
- Elaboração de um diagrama da arquitectura lógica do *HCTT*.
- Reunião *flash* semanal do Departamento de Java.
- Assisti à explicação que a Bárbara Costa deu ao Carlos Rebelo sobre os detalhes da funcionalidade que ele terá que implementar.
- Recebi explicações da Bárbara Costa sobre os detalhes da funcionalidade de Pesquisa por Palavras-Chave que terei que implementar.

30 de Março de 2004 (3ª feira)

Preparação da implementação da funcionalidade de Pesquisa por Palavras-Chave: análise de páginas (e respectivo código fonte) do *HCTT* cuja interacção com o utilizador e forma de implementação são semelhantes às pretendidas para a funcionalidade.

31 de Março de 2004 (4ª feira)

- Continuação da preparação da implementação da funcionalidade de Pesquisa por Palavras-Chave.
- *Checkout* do CVS do novo *branch* do *HCTT*.

1 de Abril de 2004 (5ª feira)

- Início da implementação da funcionalidade de Pesquisa por Palavras-Chave:
 - Geração das *domain classes* e interfaces e exportação para os respectivos projectos do *HCTT*.
 - Criação das tabelas necessárias da base de dados.
 - Geração das *data services classes* e exportação para os respectivos projectos do *HCTT*.
- Reunião com o Gonçalo Mata onde foi revista a análise da funcionalidade de Pesquisa por Palavras-Chave.

2 de Abril de 2004 (6ª feira)

- Eliminação das tabelas, *domain classes* e *data services classes* criadas no dia anterior (mostraram-se inadequadas após a revisão da análise da funcionalidade de Pesquisa por Palavras-Chave).
- Desenho, em papel, dos *layouts* das páginas a criar para a funcionalidade de Pesquisa por Palavras-Chave.

- Reunião com o Gonçalo Mata e com a Lígia Almeida para concluir a revisão da análise da funcionalidade de Pesquisa por Palavras-Chave.
- Definição, com o Gonçalo Mata, do trabalho a efectuar para a fase de Normalização Gráfica do HCTT durante o estágio.
- Elaboração do protocolo de estágio.

Semana 6

5 de Abril de 2004 (2ª feira)

- Continuação da implementação da funcionalidade de Pesquisa por Palavras-Chave:
 - Criação das *models classes*.
 - Criação das *schemas tables*.
 - Criação das *maps persistent classes*.
- Reunião *flash* semanal do Departamento de Java.
- Continuação da implementação da funcionalidade de Pesquisa por Palavras-Chave:
 - Geração das *domain classes* e interfaces e exportação para os respectivos projectos do HCTT.
 - Criação das tabelas necessárias da base de dados.
 - Geração das *data services classes* e exportação para os respectivos projectos do HCTT.

6 de Abril de 2004 (3ª feira)

- Resolução de um problema surgido com as *domain classes* e *data services classes* exportadas, no dia anterior, para o HCTT.

7 de Abril de 2004 (4ª feira)

- Conclusão da resolução do problema do dia anterior.

8 de Abril de 2004 (5ª feira)

- Actualização do Twiki.
- Tentativa de resolução de um novo problema.
- *De tarde: tolerância de ponto 5ª Feira Santa*

9 de Abril de 2004 (6ª feira)

- *Feriado 6ª Feira Santa*

Semana 7

12 de Abril de 2004 (2ª feira)

- Continuação da resolução do novo problema surgido na semana anterior.
- Reunião *flash* semanal do Departamento de Java.
- Conclusão da resolução do problema surgido.

- Continuação da implementação da funcionalidade de Pesquisa por Palavras-Chave:
 - Criação da acção de inserção de uma nova palavra-chave associada a uma *Person* (edição do processo XML responsável por apresentar as palavras-chave associadas a uma *Person* e respectivo *adapter*).

13 de Abril de 2004 (3ª feira)

- Continuação da implementação da funcionalidade de Pesquisa por Palavras-Chave:
 - Continuação da criação da acção de inserção de uma nova palavra-chave associada a uma *Person* (edição do processo XML e *adapter* respectivos; edição dos *commands* necessários).

14 de Abril de 2004 (4ª feira)

- Continuação da implementação da funcionalidade de Pesquisa por Palavras-Chave:
- Suspensão da criação da acção de inserção de uma nova palavra-chave associada a uma *Person*.
- Criação da acção de listagem das palavra-chave associadas a uma *Person* (edição do *command* necessário e criação do *controller*).

15 de Abril de 2004 (5ª feira)

- Continuação da implementação da funcionalidade de Pesquisa por Palavras-Chave:
 - Continuação da criação da acção de listagem das palavras-chaves associadas a uma *Person* (correções ao *command* necessário e ao *controller*, geração dos JSP. Realização de um teste: o resultado foram algumas excepções e o falhanço da listagem. Consequentemente, deu-se início à realização de um *debug*).

16 de Abril de 2004 (6ª feira)

- Continuação da implementação da funcionalidade de Pesquisa por Palavras-Chave:
 - Continuação da criação da acção de listagem das palavras-chaves associadas a uma *Person* (continuação do *debug* ao código, uma vez que a listagem não tinha sido bem sucedida).
 - O problema foi descoberto: quando se executa uma *query* a uma tabela, deve fazer-se *SELECT* de todos os campos da tabela, mesmo que nem todos sejam necessários, como era o caso (a forma como os *models* e os *services* são gerados no *Persistence Builder* obrigam a isso...). Se o *SELECT* não for feito a todos os campos da tabela ocorre uma *ArrayOutOfBoundsException*.

Semana 8

19 de Abril de 2004 (2ª feira)

- Continuação da implementação da funcionalidade de Pesquisa por Palavras-Chave:
 - Conclusão, com sucesso, da acção de listagem das palavra-chave associadas a uma *Person*.
 - Retoma da criação da acção de inserção de uma nova palavra-chave associada a uma *Person*.
- Reunião *flash* semanal do Departamento de Java.
- Continuação da implementação da funcionalidade de Pesquisa por Palavras-Chave:

- Continuação da criação da acção de inserção de uma nova palavra-chave associada a uma *Person* (edição do XML, edição do *adapter*, criação dos *commands*, edição do *controller*, geração dos JSP).
- Realização de um teste cujo resultado foi o falhanço da acção.

20 de Abril de 2004 (3ª feira)

- Continuação da implementação da funcionalidade de Pesquisa por Palavras-Chave:

- *Debug* ao código desenvolvido para a acção de inserção de uma nova palavra-chave associada a uma *Person*, devido a falhanço do teste realizado no dia anterior.
- Conclusão, com sucesso, da acção de inserção de uma nova palavra-chave associada a uma *Person*.
- Início da criação da acção de remoção de uma palavra-chave associada a uma *Person* (edição do XML, edição do *adapter*, criação do *command*, edição do *controller*, geração dos JSP).
- Realização de um teste cujo resultado foi o falhanço da acção.
- *Debug* ao código desenvolvido.
- Conclusão, com sucesso, da acção de remoção de uma palavra-chave associada a uma *Person*.
- Início da criação da acção de remoção de uma ou mais palavras-chave, ao mesmo tempo, associadas a uma *Person* (edição do XML, edição do *adapter*, criação do *command*, edição do *controller*, geração dos JSP).
- Realização de um teste cujo resultado foi o sucesso da acção. A acção de remoção de uma ou mais palavras-chaves, ao mesmo tempo, associadas a uma *Person*, foi concluída com êxito.
- Início da criação da acção de edição de uma palavra-chave associada a uma *Person* (edição do XML, edição do *adapter*, criação do *command*, edição do *controller*, geração dos JSP).
- Realização de um teste cujo resultado foi o falhanço da acção.

21 de Abril de 2004 (4ª feira)

- Continuação da implementação da funcionalidade de Pesquisa por Palavras-Chave:

- *Debug* ao código desenvolvido para a acção de edição de uma palavra-chave associada a uma *Person*, devido a falhanço do teste realizado no dia anterior.
- Conclusão, com sucesso, da acção de edição de uma palavra-chave associada a uma *Person*.
- Início da criação da acção de edição em modo livre das palavras-chave associadas a uma *Person* (edição do XML, criação do *adapter*, criação dos *commands*, edição do *controller*, geração dos JSP).
- Realização de alguns testes que revelaram alguns *bugs* na acção.
- *Debug* ao código desenvolvido.

22 de Abril de 2004 (5ª feira)

- Continuação da implementação da funcionalidade de Pesquisa por Palavras-Chave:

- Continuação de *debug* ao código desenvolvido para a acção de edição em modo livre.
- Conclusão, com sucesso, da acção de edição em modo livre das palavras-chave associadas a uma *Person*.
- Início da criação da acção de consulta do *template* simples de palavras-chave.

- Realização de alguns testes que revelaram alguns *bugs* na acção.
- *Debug* ao código desenvolvido.
- Conclusão, com sucesso, da acção de consulta do *template* simples de palavras-chave.
- Início da criação da acção de inserção de uma palavra-chave escolhida a partir do *template* simples, para uma *Person*.

23 de Abril de 2004 (6ª feira)

- Continuação da implementação da funcionalidade de Pesquisa por Palavras-Chave:

- Realização de alguns testes que revelaram alguns *bugs* na acção iniciada no dia anterior (inserção de uma palavra-chave escolhida a partir do *template* simples para uma *Person*).
- *Debug* do código desenvolvido.
- Conclusão, com sucesso, da acção de inserção de uma palavra-chave escolhida a partir do *template* simples para uma *Person*.
- Início da criação da acção de remoção das palavras-chaves coincidentes com uma palavra-chave escolhida a partir do *template* simples, para uma *Person*.
- Realização de um teste cujo resultado foi o sucesso da acção. A acção de remoção das palavras-chaves coincidentes com uma palavra-chave escolhida a partir do *template* simples, para uma *Person*, foi concluída com sucesso.
- Início da criação da acção de remoção de várias palavras-chave escolhidas a partir do *template* simples, para uma *Person*.
- Realização de um teste cujo resultado foi o sucesso da acção. A acção de remoção de várias palavras-chaves escolhidas a partir do *template* simples, para uma *Person*, foi concluída com sucesso.
- Início da criação da acção de inserção de várias palavras-chave escolhidas a partir do *template* simples, para uma *Person*.
- Realização de um teste cujo resultado foi o sucesso da acção. A acção de inserção de várias palavras-chaves escolhidas a partir do *template* simples, para uma *Person*, foi concluída com sucesso.
- Conclusão da funcionalidade de palavras-chaves associadas a uma *Person*.
- *Limpeza* de código.
- Realização de testes de integração.

Semana 9

26 de Abril de 2004 (2ª feira)

- Sincronização com o repositório CVS.
- Resolução de conflitos aparecidos.
- Continuação da implementação da funcionalidade de Pesquisa por Palavras-Chave:
 - Início da implementação da pesquisa por palavras-chave nos Processos de Selecção.
- Reunião *flash* do Departamento de Java.
- *Commit* para o CVS das classes criadas/ modificadas na semana passada.
- Cópia das classes criadas / modificadas a semana passada para o servidor *Alpha*.
- Testes no servidor.

27 de Abril de 2004 (3ª feira)

- Edição dos *scripts* da base de dados.
- Continuação da implementação da funcionalidade de Pesquisa por Palavras-Chave:
 - Continuação da implementação da pesquisa por palavras-chaves nos Processos de Selecção.

28 de Abril de 2004 (4ª feira)

- Continuação da implementação da funcionalidade de Pesquisa por Palavras-Chave:
 - Continuação da implementação da pesquisa por palavras-chaves nos Processos de Selecção.
 - Execução de testes que demonstraram alguns *bugs*.
 - *Debug* ao código desenvolvido.

29 de Abril de 2004 (5ª feira)

- Continuação da implementação da funcionalidade de Pesquisa por Palavras-Chave:
 - Continuação de *debug* ao código desenvolvido.
 - Realização de alterações ao código desenvolvido.
 - Realização de testes.
 - Conclusão, com sucesso, da implementação da pesquisa por palavras-chave nos Processos de Selecção.
 - Início da implementação da pesquisa por palavras-chave nos Meios de Recrutamento.
 - Realização de testes.
 - Conclusão, com sucesso, da implementação da pesquisa por palavras-chave nos Meios de Recrutamento.

30 de Abril de 2004 (6ª feira)

- Continuação da implementação da funcionalidade de Pesquisa por Palavras-Chave:
 - Início da implementação da funcionalidade de palavras-chaves associadas a um Processo de Selecção.

Semana 10

3 de Maio de 2004 (2ª feira)

- Continuação da implementação da funcionalidade de Pesquisa por Palavras-Chave:
 - Conclusão da implementação da camada de lógica de negócio da funcionalidade de palavras-chaves associadas a um Processo de Selecção.
- Reunião *flash* do Departamento de Java.
- Continuação da implementação da funcionalidade de Pesquisa por Palavras-Chave:
 - Geração dos JSP - camada de interface da funcionalidade de palavras-chave associadas a um Processo de Selecção.
 - Resolução de alguns problemas surgidos.

4 de Maio de 2004 (3ª feira)

- Continuação da implementação da funcionalidade de Pesquisa por Palavras-Chave:
 - Continuação da resolução de alguns problemas surgidos.

- Conclusão da implementação de toda a funcionalidade de palavras-chave associadas a um Processo de Selecção.
- *Commit* para o CVS.

- *De tarde: tolerância de ponto devido ao Cortejo da Queima das Fitas .*

5 de Maio de 2004 (4ª feira)

- *De manhã: tolerância de ponto devido à Queima das Fitas .*

- Continuação da implementação da funcionalidade de Pesquisa por Palavras-Chave:
 - Realização de testes.

6 de Maio de 2004 (5ª feira)

- Continuação da implementação da funcionalidade de Pesquisa por Palavras-chave:
 - Continuação da realização de testes.

7 de Maio de 2004 (6ª feira)

- Continuação da implementação da funcionalidade de Pesquisa por Palavras-Chave:
 - Resolução de alguns problemas surgidos no servidor *alpha* do HCTT.
 - Sincronização com o CVS.
 - Conclusão da implementação da funcionalidade de Pesquisa por Palavras-Chave.

- *De tarde: tolerância de ponto devido ao Baile de Finalistas.*

Semana 11

10 de Maio de 2004 (2ª feira)

- Resolução de alguns *bugs* detectados na Pesquisa por Palavras-Chave.
- Reunião *flash* semanal do Departamento de Java.
- Início da elaboração do documento de especificação da funcionalidade de Pesquisa por Palavras-Chave segundo o modelo de especificação utilizado pelo Departamento de Java.

11 de Maio de 2004 (3ª feira)

- Continuação da elaboração do documento de especificação da funcionalidade de Pesquisa por Palavras-Chave.

12 de Maio de 2004 (4ª feira)

- Conclusão da elaboração do documento de especificação da funcionalidade de Pesquisa por Palavras-Chave (o documento foi enviado para o Gonçalo Mata para validação).

13 de Maio de 2004 (5ª feira)

- *Checkout* do novo *branch* do CVS para o HCTT.
- Reunião com o Gonçalo Mata onde foram definidos os objectivos da próxima fase do estágio referente à Normalização Gráfica e onde foram também indicadas algumas correcções a efectuar na funcionalidade de Pesquisa por Palavras-Chave.

- Início da execução das correcções indicadas para a Pesquisa por Palavras-Chave.

14 de Maio de 2004 (6ª feira)

- Continuação das execução das correcções indicadas para a Pesquisa por Palavras-Chave.

- Elaboração de algumas experiências referentes ao código gerado pelo *Persistence Builder*.

Semana 12

17 de Maio de 2004 (2ª feira)

- Continuação da elaboração de algumas experiências referentes ao código gerado pelo *Persistence Builder*.
- Sessão de formação em *AnyWare*.
- Começo da Normalização Gráfica do *HCTT*:
 - Rastreo de todas as páginas do *HCTT* que contêm uma listagem de Pessoas e elaboração de um documento contendo o resultado deste rastreo.

18 de Maio de 2004 (3ª feira)

- Continuação da execução das correcções indicadas para a Pesquisa por Palavras-Chave.
- 2ª Reunião de Orientação de Estágio com os orientadores de estágio.
- Conclusão da execução das correcções indicadas para a Pesquisa por Palavras-Chave.
- Elaboração da versão final do documento de especificação da Pesquisa por Palavras-Chave (o documento foi mandado para o Gonçalo Mata para validação).
- *Commit* das correcções para o CVS e exportação para a *Alpha*.

19 de Maio de 2004 (4ª feira)

- Actualização dos *scripts* da base de dados para a criação das tabelas onde são guardados os registos referentes às palavras-chave.
- Escrita da acta da 2ª Reunião de Orientação de Estágio.
- Actualização dos pedidos técnicos.
- Continuação da Normalização Gráfica:
 - Continuação do rastreo de todas as páginas do *HCTT* que contêm uma listagem de Pessoas e elaboração de um documento contendo o resultado deste rastreo.

20 de Maio de 2004 (5ª feira)

- Continuação da Normalização Gráfica:
 - Continuação do rastreo de todas as páginas do *HCTT* que contêm uma listagem de Pessoas e elaboração de um documento contendo o resultado deste rastreo.
- Interrupção da tarefa anterior devido ao facto de estar a decorrer uma sessão de apresentação do *AnyWare* (que utiliza o mesmo servidor de base de dados que o *HCTT*) e ter sido pedido para que não fosse feito qualquer acesso a este servidor - isso implica parar com qualquer actividade do *HCTT*.

21 de Maio de 2004 (6ª feira)

- Continuação da Normalização Gráfica:
- Conclusão do rastreo de todas as páginas do *HCTT* que contêm uma listagem de Pessoas e elaboração de um documento contendo o resultado deste rastreo.

Semana 13

24 de Maio de 2004 (2ª feira)

- Continuação da Normalização Gráfica:
 - Início da análise / estudo das alterações a efectuar nas páginas identificadas no rastreio de páginas contendo listagens de Pessoas (num total de 40 páginas).
- Reunião *flash* semanal do Departamento de Java.
- Continuação da Normalização Gráfica:
 - Continuação da análise / estudo das alterações a efectuar nas páginas identificadas no rastreio de páginas contendo listagens de Pessoas (num total de 40 páginas).

25 de Maio de 2004 (3ª feira)

- Continuação da Normalização Gráfica:
 - Continuação da análise / estudo das alterações a efectuar nas páginas identificadas no rastreio de páginas contendo listagens de Pessoas (num total de 40 páginas).

26 de Maio de 2004 (4ª feira)

- Continuação da Normalização Gráfica:
 - Continuação da análise / estudo das alterações a efectuar nas páginas identificadas no rastreio de páginas contendo listagens de Pessoas (num total de 40 páginas).

27 de Maio de 2004 (5ª feira)

- Continuação da Normalização Gráfica:
 - Continuação da análise / estudo das alterações a efectuar nas páginas identificadas no rastreio de páginas contendo listagens de Pessoas (num total de 40 páginas).

28 de Maio de 2004 (6ª feira)

- Continuação da Normalização Gráfica:
 - Continuação da análise / estudo das alterações a efectuar nas páginas identificadas no rastreio de páginas contendo listagens de Pessoas (num total de 40 páginas).

Semana 14

31 de Maio de 2004 (2ª feira)

- Substituição dos projectos do Eclipse pelo novo *branch*.
- Reunião *flash* do Departamento de Java.
- Continuação da Normalização Gráfica:
 - Continuação da análise / estudo das alterações a efectuar nas páginas identificadas no rastreio de páginas contendo listagens de Pessoas (num total de 40 páginas).

1 de Junho de 2004 (3ª feira)

- Continuação da Normalização Gráfica:
 - Elaboração de um quadro-resumo da análise / estudo das alterações a efectuar nas páginas identificadas no rastreio de páginas contendo listagens de Pessoas (num total de 40 páginas).

- Reunião com o Gonçalo Mata e com a Bárbara Costa de discussão da análise efectuada e definição do trabalho a elaborar dada essa análise.

2 de Junho de 2004 (4ª feira)

- Continuação da Normalização Gráfica:

- Elaboração de diagramas de colaboração em *Visio* reflectindo a comunicação entre as classes e/ou objectos analisados no estudo elaborado.

3 de Junho de 2004 (5ª feira)

- Continuação da Normalização Gráfica:

- Inclusão dos diagramas de colaboração no documento que vem sendo realizado contendo o estudo.
- Inserção, no documento anteriormente referido, da análise funcional realizada.
- Começo da elaboração de um quadro-resumo do tipo de informação sobre Pessoas apresentada em cada listagem de Pessoas.

4 de Junho de 2004 (6ª feira)

- Continuação da Normalização Gráfica:

- Conclusão da elaboração de um quadro-resumo do tipo de informação sobre Pessoas apresentada em cada listagem de Pessoas.
- Elaboração de um quadro-resumo da análise técnica levada a cabo às listagens de Pessoas.
- Inclusão, no documento que vem sendo realizado sobre o estudo, dos quadros-resumo elaborados.

Semana 15

7 de Junho de 2004 (2ª feira)

- Continuação da Normalização Gráfica:

- Elaboração de um quadro-resumo da análise funcional levada a cabo às listagens de Pessoas.
- Inclusão, no documento que vem sendo realizado sobre o estudo, do quadro-resumo elaborado.

- Reunião *flash* semanal do Departamento de Java.

- Continuação da Normalização Gráfica:

- Exploração do uso de vistas em *UniFace* de forma a reduzir o número de páginas novas que têm que ser geradas quando se pretende que uma página apresente exactamente o mesmo conteúdo de outra página já existente anteriormente.

8 de Junho de 2004 (3ª feira)

- Continuação da Normalização Gráfica:

- Continuação da exploração do uso de vistas em *UniFace* de forma a reduzir o número de páginas novas que têm que ser geradas quando se pretende que uma página apresente exactamente o mesmo conteúdo que outra página já existente anteriormente.

- Chegou-se à conclusão que não é possível usar vistas que utilizem *adapters* diferentes (ou seja, para qualquer vista de uma mesma página, o *adapter* tem que ser o mesmo).
- Implementação de um novo processo XML *HCTTPersonsLists.xml* responsável por conter vistas para a listagem de Colaboradores Activos em várias páginas do HCTT.
- Modificação de algumas páginas dos processos *Recruitment.xml* e *OPST.xml* de modo a usarem o novo processo *HCTTPersonsLists.xml*.
- Resolução de alguns problemas surgidos.

9 de Junho de 2004 (4ª feira)

- Continuação da Normalização Gráfica:

- Resolução dos problemas surgidos no dia anterior.
- Modificação da página do processo *OPST.xml* responsável por listar Colaboradores Activos, de modo a usar um processo genérico (*HCTTListUtils.xml*) encarregue das páginas de pesquisa e navegação. A lógica de negócio que suporta esta página também foi alterada de modo a usar um comando genérico (*GetActiveHRsColCmd*) responsável por devolver os Colaboradores Activos da base de dados.
- Resolução de alguns problemas surgidos.

10 de Junho de 2004 (5ª feira)

- *Feridado Nacional*

11 de Junho de 2004 (6ª feira)

- Continuação da Normalização Gráfica:

- Conclusão da resolução dos problemas surgidos.
- Conclusão da modificação da página do processo *OPST.xml* responsável por listar Colaboradores Activos.
- Modificação da página do processo *Recruitment.xml* responsável por listar Colaboradores Activos, de modo a usar um processo genérico (*HCTTListUtils.xml*) encarregue das páginas de pesquisa e navegação. A lógica de negócio que suporta esta página também foi alterada de modo a usar um comando genérico (*GetActiveHRsColCmd*) responsável por devolver os Colaboradores activos da base de dados.
- Conclusão da modificação da página do processo *Recruitment.xml* responsável por listar Colaboradores Activos.

Semana 16

14 de Junho de 2004 (2ª feira)

- Início da definição da estrutura do Relatório de Estágio.

- Reunião *flash* semanal do Departamento de Java.

- Reunião com a Bárbara Costa para discussão da estratégia a adoptar na normalização das páginas de listagem de Responsáveis por Estruturas e Processos de Selecção.

- Continuação da Normalização Gráfica:

- Início da implementação das alterações às listagens de Responsáveis para um caso-exemplo: os Responsáveis por um Processo de Selecção.

15 de Junho de 2004 (3ª feira)

- Início da elaboração da versão 2.0 do *MethoDoc* - capítulo 2 "Análise e Especificação".

16 de Junho de 2004 (4ª feira)

- Continuação da Normalização Gráfica:

- Continuação da implementação das alterações às listagens de Responsáveis para um caso-exemplo: os Responsáveis por um Processo de Selecção.
- Resolução de alguns problemas surgidos.

17 de Junho de 2004 (5ª feira)

- Correção de um erro crítico na pesquisa do módulo de Recrutamento.

- Teste às alterações efectuadas.

- Exportação dos ficheiros alterados para a *alpha*.

18 de Junho de 2004 (6ª feira)

- Testes na *alpha* às correcções efectuadas.

- Correção de um erro no navegador da pesquisa no Meio de Recrutamento, no Módulo de Recrutamento.

- Resolução de problemas surgidos durante testes.

- Continuação da elaboração da versão 2.0 do *MethoDoc* - capítulo 2 "Análise e Especificação".

Semana 17

21 de Junho de 2004 (2ª feira)

- Sincronização com o CVS e resolução de conflitos.

- Reunião *flash* semanal do Departamento de Java.

- Actualização do HCTT com o novo *branch*.

- Continuação da Normalização Gráfica:

- Continuação da resolução de alguns problemas surgidos na implementação das alterações às listagens de Responsáveis para um caso-exemplo: os Responsáveis por um Processo de Selecção.

22 de Junho de 2004 (3ª feira)

23 Continuação da Normalização Gráfica:

- Conclusão da implementação das alterações às listagens de Responsáveis para um caso-exemplo: os Responsáveis por um Processo de Selecção.
- Execução de testes.
- Alargamento das alterações efectuadas a outro caso: os Responsáveis por uma Estrutura no módulo de Estrutura.

23 de Junho de 2004 (4ª feira)

- Continuação da Normalização Gráfica:

- Alargamento das alterações efectuadas a outro caso: os Responsáveis por um Nó no módulo de Estrutura.

- Alargamento das alterações efectuadas a outro caso: os Responsáveis por uma Estrutura no módulo de Gestão de Recursos.
- Alargamento das alterações efectuadas a outro caso: os Responsáveis por um Nó no módulo de Gestão de Recursos.
- Elaboração de um diagrama de colaboração esquematizando as alterações efectuadas.

- Reunião com Gonçalo Mata e Bárbara Costa para análise do trabalho efectuado e definição do trabalho seguinte.

24 de Junho de 2004 (5ª feira)

- *Feriado Local de S.João.*

25 de Junho de 2004 (6ª feira)

- Sincronização com o CVS.

- *Commit*, para o CVS, das alterações efectuadas esta semana.

- Continuação da elaboração da versão 2.0 do *MethoDoc* - conclusão do capítulo 2 "Análise e Especificação".

Semana 18

28 de Junho de 2004 (2ª feira)

- Continuação da Normalização Gráfica:

- Início da implementação das alterações à forma de listagem de Responsáveis e Colaboradores Activos, tal como foi delineado na reunião com o Gonçalo Mata e a Bárbara Costa.

- Reunião *flash* semanal do Departamento de Java.

29 de Junho de 2004 (3ª feira)

- Continuação da Normalização Gráfica:

- Continuação da implementação das alterações à forma de listagem de Responsáveis e Colaboradores Activos, tal como foi delineado na reunião com o Gonçalo Mata e a Bárbara Costa.

30 de Junho de 2004 (4ª feira)

- Continuação da Normalização Gráfica:

- Continuação da implementação das alterações à forma de listagem de Responsáveis e Colaboradores Activos, tal como foi delineado na reunião com o Gonçalo Mata e a Bárbara Costa.
- Execução de testes.

1 de Julho de 2004 (5ª feira)

- Continuação da Normalização Gráfica:

- Resolução de problemas surgidos com a navegação entre páginas.

2 de Julho de 2004 (6ª feira)

- Continuação da Normalização Gráfica:

- Conclusão da resolução de problemas surgidos com a navegação entre páginas.
- Resolução de outros problemas surgidos.

- Execução de alguns testes.

Semana 19

5 de Julho de 2004 (2ª feira)

- Sincronização com o CVS.
- Continuação da Normalização Gráfica:
 - Continuação da implementação das alterações à forma de listagem de Responsáveis e Colaboradores Activos para os módulos de Estrutura e Gestão de Recursos.
- Reunião *flash* semanal do Departamento de Java.

6 de Julho de 2004 (3ª feira)

- Continuação da Normalização Gráfica:
 - Continuação da implementação das alterações à forma de listagem de Responsáveis e Colaboradores Activos para os módulos de Estrutura e Gestão de Recursos.
 - Conclusão da implementação das alterações à forma de listagem de Responsáveis para o módulo de Estrutura.
 - Execução de testes.

7 de Julho de 2004 (4ª feira)

- Continuação da Normalização Gráfica:
 - Conclusão da implementação das alterações à forma de listagem de Responsáveis para o módulo de Gestão de Recursos.
 - Execução de testes.
 - Resolução de alguns problemas encontrados na navegação entre páginas.

8 de Julho de 2004 (5ª feira)

- Reunião do Departamento de Java sobre prazos e fiabilidade do *HCTT*.
- Reunião com o Carlos Rebelo e com o Gonçalo Mata para discussão de alguns pontos já desenvolvidos no *MethoDoc*.
- Reunião com o prof. Ademar Aguiar para demonstração da ferramenta *XSDoc*.

9 de Julho de 2004 (6ª feira)

- Continuação da elaboração da versão 2.0 do *MethoDoc* - capítulo 3 "Desenvolvimento" (conclusão da secção "Persistence Builder").
- Reunião com a Bárbara Costa, com o Gonçalo Mata e com a Cidália Neves para revisão da situação contratual.

Semana 20

12 de Julho de 2004 (2ª feira)

- Reunião *flash* semanal do Departamento de Java.
- Continuação da Normalização Gráfica:
 - Conclusão da execução de testes e resolução de problemas surgidos com a implementação das alterações à forma de listagem de Responsáveis e Colaboradores Activos para os módulos de Recrutamento, Estrutura e Gestão de Recursos (os problemas surgidos são relacionados com a navegação entre páginas).

13 de Julho de 2004 (3ª feira)

- Continuação da Normalização Gráfica:

- *Export* de todos os ficheiros novos / alterados para a *alpha*.
- Execução de testes na *alpha*.
- *Commit* para o CVS.

14 de Julho de 2004 (4ª feira)

- Continuação da elaboração da versão 2.0 do *MethoDoc* - capítulo 3 "Desenvolvimento" (elaboração da secção " *UniFace - GuiGen* ").

- Reunião com a Bárbara Costa para apresentar e analisar as alterações efectuadas às listagens de Responsáveis nos módulos de Recrutamento, Estrutura e Gestão de Recursos.

- Continuação da Normalização Gráfica:

- Execução de algumas experiências na implementação anteriormente efectuada para as listagens de Responsáveis.

- Reunião com a Bárbara Costa, com o Gonçalo Mata e com a Cidália Neves para conclusão da revisão da situação contratual.

15 de Julho de 2004 (5ª feira)

- Continuação da Normalização Gráfica:

- Execução de algumas alterações à implementação anteriormente efectuada para as listagens de Responsáveis.
- Execução de testes.

16 de Julho de 2004 (6ª feira)

- Continuação da Normalização Gráfica:

- Conclusão da execução de testes.
- Início da execução da alteração da forma como é criado um novo utilizador do HCTT com o intuito de passar a utilizar as páginas criadas para a listagem de Responsáveis.

Semana 21

19 de Julho de 2004 (2ª feira)

- Reunião *flash* semanal do Departamento de Java.

- Continuação da Normalização Gráfica:

- Conclusão das alterações à forma como é criado um novo utilizador do HCTT com o intuito de passar a utilizar as páginas criadas para a listagem de Responsáveis.
- Sincronização com o CVS e *commit* das alterações.
- Exportação para a *alpha* e execução de testes.

20 de Julho de 2004 (3ª feira)

- Conclusão da execução de testes na *alpha*.

- 3ª e última Reunião de Orientação de Estágio com os orientadores nas instalações da IBS.

- Formatação dos JSP gerados para as listas de Colaboradores Activos.

- Escrita da acta da última reunião de estágio com os orientadores.

- Continuação da elaboração da versão 2.0 do *MethoDoc* :

- Início da escrita de uma nova secção, do capítulo 3, intitulada "Implementação de Listagens de Responsáveis e de Colaboradores Activos" que consiste num manual para o programador onde é explicada e exemplificada a forma como as listagens de Responsáveis e de Colaboradores Activos devem ser implementadas a partir de agora, utilizando as novas classes e processos XML criados para o efeito até aqui.

21 de Julho de 2004 (4ª feira)

- Continuação da elaboração da versão 2.0 do *MethoDoc* :

- Conclusão da escrita da secção "Implementação de Listagens de Responsáveis e de Colaboradores Activos" do capítulo 3.

22 de Julho de 2004 (5ª feira)

- Continuação da Normalização Gráfica:

- Início da execução das alterações técnicas necessárias para que a página de detalhes de um Processo de Selecção fique dividida em *tabs* (separadores).

- Reunião com o Gonçalo Mata, a Bárbara Costa e o Carlos Rebelo sobre o ponto da situação do *MethoDoc* e definição de trabalho futuro.

- Continuação da Normalização Gráfica:

- Continuação da execução das alterações técnicas necessárias para que a página de detalhes de um Processo de Selecção fique dividida em *tabs* (separadores).

23 de Julho de 2004 (6ª feira)

- Continuação da Normalização Gráfica:

- Conclusão da execução das alterações técnicas necessárias para que a página de detalhes de um Processo de Selecção fique dividida em *tabs* (separadores).
- Execução das alterações técnicas necessárias para que a página de detalhes de um Meio de Recrutamento fique dividida em *tabs* (separadores).
- Execução de testes e início da implementação de algumas alterações aos botões de "Voltar" de algumas páginas de modo a que a navegação entre janelas funcione sem *bugs*.

Semana 22

26 de Julho de 2004 (2ª feira)

- Reunião *flash* semanal do Departamento de Java.

- Continuação da Normalização Gráfica:

- Continuação da execução das alterações técnicas necessárias para que a página de detalhes de um Meio de Recrutamento fique dividida em *tabs* (separadores) - continuação da implementação de algumas alterações aos botões de "Voltar" de algumas páginas de modo a que a navegação entre janelas funcione sem *bugs*.
- Início da execução das alterações técnicas necessárias para que a página de Estrutura do módulo de Estrutura fique dividida em *tabs* (separadores).

27 de Julho de 2004 (3ª feira)

- Continuação da Normalização Gráfica:

- Conclusão da execução das alterações técnicas necessárias para que a página de Estrutura do módulo de Estrutura fique dividida em *tabs* (separadores).
- Execução das alterações técnicas necessárias para que a página de Nó do módulo de Estrutura fique dividida em *tabs* (separadores).
- Execução das alterações técnicas necessárias para que a página de Função do módulo de Estrutura fique dividida em *tabs* (separadores).
- Execução das alterações técnicas necessárias para que a página de Estrutura do módulo de Gestão de Recursos fique dividida em *tabs* (separadores).
- Execução das alterações técnicas necessárias para que a página de Nó do módulo de Gestão de Recursos fique dividida em *tabs* (separadores).
- Início da execução das alterações técnicas necessárias para que a página de Função do módulo de Gestão de Recursos fique dividida em *tabs* (separadores).

- Reunião com o Gonçalo Mata, Bárbara Costa e Carlos Rebelo de discussão de um primeiro protótipo do *MethoDoc* em versão *Wiki*.

28 de Julho de 2004 (4ª feira)

- Continuação da Normalização Gráfica:

- Conclusão da execução das alterações técnicas necessárias para que a página de Função do módulo de Gestão de Recursos fique dividida em *tabs* (separadores).
- Execução de testes.
- Resolução de problemas surgidos com a navegação entre páginas.

29 de Julho de 2004 (5ª feira)

- Continuação da Normalização Gráfica:

- Conclusão da resolução dos problemas surgidos com a navegação entre páginas.
- Execução de testes.
- Sincronização com o repositório CVS e *commit*.

30 de Julho de 2004 (6ª feira)

- Conclusão da Normalização Gráfica:

- *Export* de todas as alterações efectuadas para a *alpha*.
- Execução de testes na *alpha*.

Semana 23

2 de Agosto de 2004 (2ª feira)

- Reunião *flash* semanal do Departamento de Java.

- Início da elaboração da versão inicial do Relatório de Estágio.

3 de Agosto de 2004 (3ª feira)

- Continuação da elaboração da versão inicial do Relatório de Estágio.

4 de Agosto de 2004 (4ª feira)

- Continuação da elaboração da versão inicial do Relatório de Estágio.

5 de Agosto de 2004 (5ª feira)

- Continuação da elaboração da versão inicial do Relatório de Estágio.

6 de Agosto de 2004 (6ª feira)

- Conclusão da elaboração da versão inicial do Relatório de Estágio.

- Envio da versão inicial do Relatório de Estágio para os orientadores (Gonçalo Mata e Prof. Ademar Aguiar).

Semana 24

9 de Agosto de 2004 (2ª feira)

- Reunião *flash* semanal do Departamento de Java.

- Finalizações:

- Escrita da secção "Implementação de tabs (separadores)" no capítulo 3 - "Desenvolvimento" do *MethoDoc* que pretende ser um manual para os programadores de como se implementam páginas com *tabs* no HCTT.
- Pequenas correcções aos capítulos 2 e 3 do *MethoDoc*.
- Resolução de um problema surgido no Eclipse (após um *synchronize* com o CVS não conseguiu compilar os projectos devido a erros de memória).
- Alteração da navegação entre páginas nos *tabs* das páginas de detalhes de um Processo de Selecção.

10 de Agosto de 2004 (3ª feira)

- Continuação das finalizações:

- Conclusão da alteração da navegação entre páginas nos *tabs* das páginas de detalhes de um Processo de Selecção.
- Eliminação das páginas de confirmação de "Apagar palavra-chave?" de Colaboradores e de Processos de Selecção.
- Execução de testes.
- *Commit* das alterações para o CVS.
- Exportação para a *alpha* e execução de testes na *alpha*.

11 de Agosto de 2004 (4ª feira)

- Conclusão das finalizações:

- Execução de alterações em algumas páginas de modo a respeitarem regras gráficas.

12 de Agosto de 2004 (5ª feira)

- Início da elaboração da versão final do Relatório de Estágio.

13 de Agosto de 2004 (6ª feira)

- Início da elaboração da versão final do Relatório de Estágio.

- Reunião com o Gonçalo Mata de discussão da versão inicial do Relatório de Estágio.

- Reunião com a Cidália Neves de Fim de Estágio.

Semana 25

16 de Agosto de 2004 (2ª feira)

- Início da elaboração do *poster* de estágio.

17 de Agosto de 2004 (3ª feira)

- Conclusão da elaboração do *poster* de estágio.

18 de Agosto de 2004 (4ª feira)

- Continuação da elaboração da versão final do Relatório de Estágio.

19 de Agosto de 2004 (5ª feira)

- Continuação da elaboração da versão final do Relatório de Estágio.

20 de Agosto de 2004 (6ª feira)

- Continuação da elaboração da versão final do Relatório de Estágio.

Semana 26

23 de Agosto de 2004 (2ª feira)

- Conclusão da elaboração da versão final do Relatório de Estágio.





FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

BIBLIOTECA



0000081517