



**David Emmanuel
Marques Campos**

**Distribuição de Serviço Docente: *back e front office*
Web**

“Para onde quer que o homem contribua com o seu trabalho deixa também
algo do seu coração”

Sienkiewicz, Henryk



**David Emmanuel
Marques Campos**

**Distribuição de Serviço Docente: *back e front office*
Web**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Professor Doutor António Luís Jesus Teixeira e Professor Doutor Tomás Oliveira e Silva, Professor Auxiliar e Professor Associado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

o júri / the jury

presidente / president

Prof. Dr. Aníbal Manuel de Oliveira Duarte
professor catedrático da Universidade de Aveiro

vogais / examiners committee

Prof. Dr. Rui Pedro Sanches de Castro Lopes
professor coordenador do Instituto Politécnico de Bragança

Prof. Dr. António Luís Jesus Teixeira
professor auxiliar da Universidade de Aveiro

agradecimentos / acknowledgements

Ao meu orientador, Professor António Luís Jesus Teixeira, o meu sincero agradecimento por toda a disponibilidade, apoio, amizade, conselhos e orientação ao longo deste árduo ano de trabalho. Também ao Professor Mário Lima, que sempre esteve presente nas horas mais complicadas de todo este processo. Ambos, pela sua forma de estar, tornaram este projecto num trabalho de equipa com um excelente ambiente.

Ao Pedro Silva, engenheiro de sistemas do IT, por toda a ajuda e conselhos disponibilizados em assuntos mais técnicos.

Ao Professor Rui Ribeiro, o meu agradecimento pela ajuda a esclarecer todas as questões e problemas mais críticos que foram surgindo.

Um agradecimento especial a toda a minha família. Pai (Luís) e mãe (Graça), obrigado não só por todo o apoio que me deram durante este período, mas também por toda a educação e regalias que me proporcionaram ao longo da vida, sem vocês, tudo isto não passaria de uma miragem. Irmãs (Jenny e Isis), também o meu obrigado por toda a ajuda e paciência que deram aqui ao puto, não só nesta fase, mas ao longo de toda a vida.

Raquel, obrigado por toda a força, carinho, amizade e paciência ilimitada. Obrigado pela pessoa que és.

Finalmente, Luís Ribeiro, Hugo Picado e Pedro Alves sempre prontos para uma boa directa de trabalho ou um bom “reset” ao cérebro. Obrigado pelos conselhos e amizade, aqui também está um pouco de vocês.

palavras-chave

Sistemas de Informação, Servidores Web, Linguagens de Programação Web, Distribuição de Serviço Docente

resumo

A evolução social conduziu a um forte desenvolvimento de novos conceitos e à necessidade de aquisição de conhecimento, reflectindo-se essencialmente no aumento da informação e na necessidade de formação individual. Assim, o número de alunos nas instituições universitárias aumentou consideravelmente, surgindo mais cursos e conseqüentemente a necessidade de mais docentes. Neste contexto, a complexidade e o número de tarefas necessárias para organizar as instituições universitárias também cresceu, tornando mais complicada a realização de todos os procedimentos internos sem o auxílio de ferramentas informáticas.

O DETI (Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro) não foi excepção e começou a desenvolver aplicações capazes de suportar a sua organização interna. No entanto, a alta descentralização e a incoerência dos dados são factores presentes, tornando-se essencial o desenvolvimento de uma ferramenta capaz de centralizar e lidar com grande parte das tarefas executadas no departamento, auxiliando a distribuição de serviço docente.

Este documento começa com o estudo detalhado do contexto onde a ferramenta desenvolvida será utilizada, definindo a base tecnológica mais apropriada. Posteriormente, é realizada uma análise cuidadosa de todas as tarefas que farão parte da aplicação, permitindo a definição do modelo do domínio, a base estrutural de toda a plataforma. Para permitir a interacção com os utilizadores, foram desenvolvidas interfaces Web inteligentes que tornam a execução dos procedimentos mais dinâmica e fácil. As soluções da plataforma foram testadas em serviço com utilizadores reais e processos reais realizados no DETI. Como consequência, a manutenção também foi um factor a tomar em conta.

Toda a plataforma foi desenvolvida a pensar em futuras evoluções. Através da centralização de informação e da suportada extensibilidade das suas funcionalidades, abre as portas para a elaboração de um conjunto de tarefas que prometem trazer valor acrescentado à organização interna dos departamentos universitários.

keywords

Information Systems, Web Servers, Web Programming Languages, Teaching Distribution Service

abstract

The social evolution conducted to a strong development of new concepts and knowledge requirements, which was mainly reflected in increased information and educational needs. So, the number of students on the educational institutions as grown, emerging more majors and consequently the need of more professors. In this context, the complexity and the number of tasks needed to organize the educational institutions increased, becoming very hard its arrangement without the support of informatic tools.

DETI (Department of Electronics, Telecommunications and Informatics of University of Aveiro) was not an exception and started to develop applications capable of supporting organization. However, the high decentralization and incoherence of data are still present factors. Thus, the need to develop a platform capable of centralizing and dealing with most of the running tasks inside a department of a university appeared, helping the teaching distribution service.

The work begins with a detailed study about the context where the developed tool will be used by defining the more appropriate technological base architecture. Furthermore, it is performed a careful analysis of all the tasks that will be part of the application in order to enable the definition of the domain model, the structural base of the whole platform. To allow the interactions with the users, intelligent Web interfaces were developed making the execution of procedures more dynamic and easy. The platform solutions were tested in-service with real users and timelines of processes running at DETI. As a consequence, the maintenance was also taken into account.

The platform was developed keeping future evolutions in mind. Due to the centralization and supported extensibility of its functionalities, it opens the doors to the elaboration of a set of tasks that promise added value to the internal organization of the departments in a university.

Índice

Índice	i
Lista de Figuras.....	v
Lista de Tabelas.....	ix
Lista de Acrónimos	xi
1 Introdução.....	1
1.1 Motivação.....	1
1.2 Objectivos.....	1
1.3 Estrutura da Dissertação	2
2 Âmbito e Requisitos	3
2.1 Distribuição de Serviço Docente	3
2.1.1 Aplicações existentes.....	3
2.1.2 Contexto.....	5
2.1.3 Definição do Problema	5
2.1.4 Requisitos e Objectivos.....	10
2.2 Tecnologias	12
2.2.1 Web 2.0.....	13
2.2.2 Servidor Web	15
2.2.3 Sistema de Gestão de Bases de Dados (SGBD).....	19
2.2.4 Páginas Web Dinâmicas	26
2.2.5 Interface.....	33
2.2.6 Dinamizar a Interface	34
2.3 Conclusões.....	42
2.3.1 Sistema de Gestão de Bases de Dados (SGBD).....	42
2.3.2 Páginas Web Dinâmicas	43
2.3.3 Servidor Web	44
2.3.4 Dinamizar Interface	44
2.3.5 Arquitectura Final.....	44

2.4	Sumário	45
3	Base Estrutural.....	47
3.1	Visão Geral.....	47
3.1.1	Descrição de Tarefas.....	47
3.1.2	Entidades e Actores.....	56
3.1.3	Fontes de Informação.....	57
3.2	Funcionalidades.....	57
3.2.1	<i>Front Office</i>	58
3.2.2	<i>Back Office</i>	64
3.3	Modelo do Domínio.....	67
3.3.1	Universidade.....	68
3.3.2	Docentes	68
3.3.3	Disciplinas.....	70
3.3.4	Alunos.....	72
3.3.5	Dissertações.....	73
3.3.6	Opções	74
3.3.7	<i>Wish List</i>	76
3.3.8	Administração.....	76
3.4	Sumário	78
4	Interacção	79
4.1	Visão Geral.....	79
4.1.1	Restrições.....	80
4.1.2	Área de trabalho	80
4.1.3	Características Gerais	85
4.1.4	Ajuda.....	86
4.2	Aplicação de Tarefas.....	87
4.2.1	Escolha de Dissertações.....	87
4.2.2	Escolha de disciplinas de Opção	89
4.2.3	<i>Wish List</i>	92
4.2.4	Gestão de Entidades e Conceitos.....	95
4.2.5	Horários.....	102
4.3	Manutenção.....	103
4.3.1	Preventiva	104
4.3.2	Correctiva.....	104

4.4	Sumário	106
5	Conclusão.....	109
5.1	Resultados	109
5.2	Conclusões.....	110
5.2.1	Objectivos.....	110
5.2.2	Aprendizagem.....	110
5.2.3	Balanço Geral.....	111
5.3	Trabalho Futuro.....	112
6	Anexos	115
	Anexo A	115
7	Bibliografia	121

Lista de Figuras

Figura 2.1: Imagem de marca da Distribuição de Serviço Docente do PACO.....	3
Figura 2.2: Logótipo do projecto Fénix do IST.....	4
Figura 2.3: Listagem extensiva de comandos disponíveis na ferramenta de construção de horários.	6
Figura 2.4: Interface da aplicação de construção de horários.....	7
Figura 2.5: Interface da página Web de Mestrado Integrado.....	8
Figura 2.6: Extracto de uma folha de cálculo para auxiliar a distribuição do serviço docente.....	10
Figura 2.7: Logótipo do projecto Apache HTTP Server.....	16
Figura 2.8: Imagem de marca da versão 7.0 do <i>Internet Information Services</i>	18
Figura 2.9: Abstracção fornecida pelo SGBD.....	19
Figura 2.10: Marca comercial do SGBD SQL Server da Microsoft.....	21
Figura 2.11: Logótipo do SGBD Oracle.....	23
Figura 2.12: Imagem de marca do SGBD MySQL da Sun Microsystems.....	24
Figura 2.13: Funcionamento das páginas Web de cliente (em cima) e das páginas Web de servidor (em baixo) [27].....	26
Figura 2.14: Marca comercial do ASP.net.....	28
Figura 2.15: Arquitectura para compilar e executar aplicações .NET[28].	29
Figura 2.16: Logótipo do PHP.	30
Figura 2.17: Organização do código em pedidos de processamento (<i>request processing</i>), lógica de funcionamento (<i>business logic</i>) e interface (<i>presentation</i>) [31].....	32
Figura 2.18: Arquitectura para compilar e executar aplicações JSP [31].....	33
Figura 2.19: Dificuldade de gerir o aspecto de páginas <i>Web</i> antes do CSS.....	34
Figura 2.20: Exemplo simples de utilização de CSS.....	34
Figura 2.21: Interacção tradicional e interacção <i>AJAX</i>	35
Figura 2.22: Funcionamento tradicional (á esquerda) e funcionamento do <i>AJAX</i> (á direita).....	36
Figura 2.23: Imagem de marca do Adobe Flash.....	38
Figura 2.24: Linha temporal do Adobe Flash CS3.	39
Figura 2.25: Logótipo do Silverlight.....	41
Figura 2.26: Arquitectura final da plataforma DSD.....	45
Figura 3.1: Processo de escolha de dissertações.....	49
Figura 3.2: Processo de escolha de disciplinas de opção.....	52
Figura 3.3: Processo de escolha de turmas das disciplinas de opção.....	54
Figura 3.4: Funcionalidades a que os actores tem acesso no pacote Sessão.	60

Figura 3.5: Funcionalidades a que os actores tem acesso no pacote Conta.....	60
Figura 3.6: Funcionalidades a que o actor docente tem acesso no pacote Dissertação.....	61
Figura 3.7: Funcionalidades a que o actor aluno tem acesso no pacote Dissertação.....	62
Figura 3.8: Funcionalidades a que os actores tem acesso no pacote Opção.....	63
Figura 3.9: Funcionalidades a que os actores tem acesso no pacote <i>Wish List</i>	64
Figura 3.10: Diagrama do domínio para suportar as entidades Departamento, Curso e Sala.....	68
Figura 3.11: Diagrama do domínio para suportar a entidade Pessoa e todos os seus dados associados.....	69
Figura 3.12: Diagrama do domínio para suportar a entidade Docente.....	69
Figura 3.13: Diagrama do domínio para suportar a entidade Categoria dos docentes.....	70
Figura 3.14: Diagrama do domínio para suportar os Cargos dos docentes.....	70
Figura 3.15: Diagrama do domínio para suportar o conceito disciplina.....	71
Figura 3.16: Diagrama do domínio para suportar as aulas e turmas de uma determina disciplina.....	72
Figura 3.17: Diagrama do domínio para suportar a entidade Aluno.....	72
Figura 3.18: Diagrama do domínio para suportar as inscrições de alunos em eventos específicos.....	73
Figura 3.19: Diagrama do domínio para suportar o conceito de Período de inscrições.....	73
Figura 3.20: Diagrama do domínio para suportar o conceito de dissertação.....	74
Figura 3.21: Diagrama do domínio para suportar as inscrições dos alunos em dissertações.....	74
Figura 3.22: Diagrama do domínio para suportar as disciplinas de opção.....	75
Figura 3.23: Diagrama do domínio para suportar a inscrição dos alunos em disciplinas de opção.....	75
Figura 3.24: Diagrama do domínio para suportar a inscrição de alunos em turmas.....	76
Figura 3.25: Diagrama do domínio para suportar as listas de desejos dos docentes.....	76
Figura 3.26: Diagrama do domínio para suportar o controlo de acessos.....	77
Figura 3.27: Diagrama do domínio para suportar a configuração da plataforma em tempo-real.....	77
Figura 3.28: Diagrama do domínio para suportar o questionário.....	78
Figura 4.1: Estatísticas da W3Schools acerca das resoluções de ecrã mais utilizadas na internet[47].....	80
Figura 4.2: Estatísticas da W3Schools acerca do número de cores mais utilizado na internet[47].....	80
Figura 4.3: Base interactiva da plataforma DSD.....	81
Figura 4.4: Cabeçalho da aplicação.....	81
Figura 4.5: Assistente de navegação da aplicação.....	81
Figura 4.6: Menu da aplicação.....	82
Figura 4.7: Aspecto de uma secção da aplicação.....	82
Figura 4.8: Rodapé da aplicação.....	83
Figura 4.9: Vista geral da interface do <i>Front Office</i> da plataforma DSD.....	84
Figura 4.10: Validação instantânea dos dados introduzidos.....	85
Figura 4.11: Indicação do processamento de informação.....	85
Figura 4.12: Utilização de <i>Modal Windows</i> na apresentação de informação.....	86
Figura 4.13: Utilização de <i>Modal Windows</i> nas mensagens de sucesso.....	86
Figura 4.14: Ajuda em tempo real disponibilizada na aplicação.....	86
Figura 4.15: Escolha dos cursos para a dissertação.....	87

Figura 4.16: Indicação de como o acordo será assinado com o aluno.....	87
Figura 4.17: Interface para demonstração de interesse por uma dissertação.....	88
Figura 4.18: Interface para assinar um acordo de uma dissertação.	88
Figura 4.19: Interface de inserção e alteração da inscrição do aluno nas disciplinas de opção.	89
Figura 4.20: Lógica de colocação do aluno na disciplina de opção.....	90
Figura 4.21: Feedback do estado da inscrição do aluno nas disciplinas de opção.	91
Figura 4.22: Lista de colocações provisória nas disciplinas de opção.	92
Figura 4.23: Estrutura em árvore de disciplinas organizada por área e subárea.	93
Figura 4.24: Escolha das disciplinas preferidas a leccionar.....	93
Figura 4.25: Visualização das <i>Wish Lists</i> dos vários docentes numa folha de Excel.	94
Figura 4.26: Processamento de informação das <i>Wish Lists</i>	95
Figura 4.27: Menu principal de gestão de conceitos do <i>Back Office</i>	96
Figura 4.28: Exemplo de representação de conceitos do ficheiro de dados da aplicação de construção de horários.	97
Figura 4.29: Interface para importação de dados através da análise sintáctica.	97
Figura 4.30: Exemplo de feedback retornado pela funcionalidade de importação de dados.	98
Figura 4.31: Exemplo de interface para execução de acções em grupo.	98
Figura 4.32: Exemplo de desactivação de objectos.	99
Figura 4.33: Interface de recuperação de objectos.	100
Figura 4.34: Vista de todos os dados associados a um aluno.....	101
Figura 4.35: Interface de configuração da plataforma.	102
Figura 4.36: Exemplo da visualização de um horário na folha de Excel gerada.	103
Figura 4.37: Mensagem do servidor a informar que estão demasiados utilizadores ligados.	105
Figura 5.1: Diagrama de análise SWOT.....	111

Lista de Tabelas

Tabela 2.1: Cotas de mercado dos vários servidores Web segundo quatro estudos distintos.	15
Tabela 2.2: Comportamento dos SGBD perante os requisitos da plataforma DSD.	43
Tabela 2.3: Comportamento das linguagens de páginas Web dinâmicas perante os Requisitos da plataforma DSD.....	43
Tabela 2.4: Comportamento dos servidores Web perante os Requisitos da plataforma DSD.	44
Tabela 2.5: Comportamento das ferramentas para dinamizar a interface perante os Requisitos da plataforma DSD.....	44
Tabela 3.1: Formato da inscrição nas turmas das disciplinas de opção.	53
Tabela 3.2: Funcionalidades do <i>Front Office</i>	59
Tabela 3.3: Funcionalidades do <i>Back Office</i>	67
Tabela 4.1: Comparativo entre os dois servidores utilizados.....	106

Lista de Acrónimos

AJAX	<i>Asynchronous JavaScript and XML</i>
ASP	<i>Active Server Pages</i>
CGI	<i>Common Gateway Interface</i>
CSS	<i>Cascading Style Sheets</i>
DETI	Departamento de Electrónica, Telecomunicações e Informática
DOM	<i>Document Object Model</i>
DSD	Distribuição de Serviço Docente
HTML	<i>HyperText Markup Language</i>
HTTP	<i>HyperText Transfer Protocol</i>
HTTDP	<i>HyperText Transfer Protocol Daemon</i>
IDE	<i>Integrated Development Environment</i>
IIS	<i>Internet Information Services</i>
ISAPI	<i>Internet Server Application Programming Interface</i>
JSON	<i>JavaScript Object Notation</i>
JSP	<i>Java Server Pages</i>
JVM	<i>Java Virtual Machine</i>
LDAP	<i>Lightweight Directory Access Protocol</i>
MXML	<i>Magic Extensible Markup Language</i>
PDF	<i>Portable Document Format</i>
PHP	<i>Hypertext Preprocessor</i>
PL/SQL	<i>Procedural Language/Structured Query Language</i>
SGBD	Sistema de Gestão de Bases de Dados
SMTP	<i>Simple Mail Transfer Protocol</i>
SQL	<i>Structured Query Language</i>
SWOT	<i>Strengths, Weaknesses, Opportunities and Threats</i>
TPC	<i>Transaction Processing Performance Council</i>
UA	Universidade de Aveiro
URL	<i>Uniform Resource Locator</i>
VB	<i>Visual Basic</i>
XAML	<i>eXtensible Application Markup Language</i>
XHTML	<i>Extensible Hypertext Markup Language</i>
XML	<i>Extensible Markup Language</i>
XSLT	<i>Extensible Stylesheet Language Transformations</i>

1 Introdução

Nesta dissertação pretende-se desenvolver uma plataforma capaz de auxiliar a Distribuição de Serviço Docente (DSD) de uma instituição universitária. Apesar de o trabalho apresentado se encontrar focado num departamento de uma universidade, a sua extensibilidade a outros departamentos e escolas é estruturalmente suportada.

1.1 Motivação

Com a evolução da sociedade, a detenção de um grau passou a ser uma das principais referências para qualquer elemento que dela faz parte. Desta forma, frequentar instituições de ensino há muito que deixou de ser um luxo, mas sim uma necessidade imposta pela própria sociedade em que vivemos. Assim, dado o crescente número de alunos, as instituições tiveram de evoluir de forma a responder às necessidades exigidas, crescendo em vários aspectos, como na sua dimensão e número de docentes.

Em toda e qualquer instituição, existe a necessidade de organizar as diferentes tarefas que nela se realizam, exigindo-se a gestão de um vasto conjunto de recursos. Com a crescente dimensão, as técnicas utilizadas anteriormente há muito que deixaram de ser usáveis, tomando-se extremamente penosas e complexas de realizar. Assim, acompanhando a evolução tecnológica, começaram a surgir soluções informáticas capazes de facilitar e dinamizar toda a organização e gestão das instituições.

O Departamento de Electrónica, Telecomunicações e Informática (DETI) da Universidade de Aveiro (UA) não foi excepção, tendo-se verificado um crescimento bastante acentuado – na altura da sua fundação oferecia apenas um curso, aumentando para três os cursos que são oferecidos presentemente. Desta forma, também as técnicas de gestão têm de evoluir, tendo-se já mostrado vontade de dar um passo mais além. Contudo, a organização de muitas tarefas ainda é feita de forma algo tradicional (com a utilização de papeis e folhas de cálculo), surgindo a expressa necessidade de uma plataforma informática capaz de automatizar, dinamizar e centralizar todo este complexo processo.

1.2 Objectivos

O principal objectivo deste trabalho passa pela construção de uma plataforma capaz de lidar com grande parte das tarefas realizadas num departamento universitário. Para que isto aconteça de forma eficiente, é necessário atingir cinco metas essenciais:

- Analisar o contexto e as necessidades da distribuição do serviço docente, bem como algumas soluções já existentes;
- Estudar as soluções tecnológicas existentes que melhor se adaptam à solução pretendida;
- Implementar a arquitectura base de suporte a toda a plataforma;
- Implementar as interacções com a plataforma base, através de uma interface usável e inteligente;
- Testar e efectuar a manutenção de toda a plataforma numa situação de vida real.

1.3 Estrutura da Dissertação

Este documento encontra-se dividido em cinco capítulos.

No capítulo 2 consta o estudo do contexto em que a ferramenta irá assentar, contribuindo para isso a análise de algumas ferramentas já existentes e a definição do problema encontrado, sendo feita a análise de requisitos e objectivos a atingir para tornar a sua resolução numa realidade. É também feito um exame cuidadoso das soluções tecnológicas existentes, encontrando a arquitectura que, no ponto de vista do autor, melhor se adapta aos requisitos encontrados.

No capítulo 3 é feito um estudo detalhado das tarefas passíveis de serem implementadas no âmbito deste trabalho, bem como a definição de funcionalidades capazes de responder às suas exigências. Também é apresentado o modelo do domínio capaz de responder às necessidades encontradas e que servirá de base estrutural para toda a plataforma.

No quarto capítulo é apresentada a forma de interacção com a base estrutural previamente definida, indicando também como é que as tarefas foram aplicadas ao modelo interactivo definido. Como consequência desta interacção num ambiente de vida real, são apresentados os cuidados de manutenção que foram tomados.

No quinto e último capítulo são apresentados alguns resultados medidos do contacto directo com utilizadores da aplicação, podendo-se daí tirar um balanço de todo o trabalho desenvolvido. Finalmente são apontados alguns caminhos para o trabalho futuro a ser realizado na plataforma.

2 Âmbito e Requisitos

2.1 Distribuição de Serviço Docente

De forma simplista, a Distribuição de Serviço Docente (DSD) corresponde à afectação de recursos humanos e físicos a um determinado conjunto de tarefas a realizar no âmbito de uma instituição de ensino num determinado período de tempo. Tipicamente quando se fala deste tema, apenas se pensa no problema de construção de horários, em que é necessário associar docentes a disciplinas, que por sua vez terão aulas que serão leccionadas em salas numa determinada hora, existindo uma panóplia de regras a respeitar. No entanto, a DSD não se resume a horários, estendendo-se também à organização das mais variadas tarefas, como a escolha de dissertações, escolha de disciplinas de opção, inscrições em turmas, entre muitas outras.

2.1.1 Aplicações existentes

Em algumas instituições de ensino existem ferramentas informáticas para auxiliar a distribuição de serviço docente, no entanto, dada a alta confidencialidade dos dados o acesso é restrito, não sendo possível qualquer análise mais profunda. Ainda assim são apresentados dois exemplos, da Universidade de Aveiro (análise mais profunda) e o do Instituto Superior Técnico (IST).

2.1.1.1 PACO – Distribuição de Serviço Docente

O sistema de Distribuição de Serviço Docente¹ (Figura 2.1) é parte integrante do PACO (Portal Académico Online), uma ferramenta de apoio aos alunos da Universidade de Aveiro, sendo uma das aplicações desenvolvidas pelo GaGI (Gabinete de Gestão de Informação), responsável por “obter, estruturar e disponibilizar informação identificada como necessária à gestão da Universidade de Aveiro” [1].



Figura 2.1: Imagem de marca da Distribuição de Serviço Docente do PACO.

¹ <https://paco.ua.pt/dsd/>

Esta ferramenta é essencialmente utilizada por funcionários das secretarias dos departamentos, não sendo possível o acesso através das contas de alunos ou docentes. Assim, para melhor conhecer as características desta ferramenta entrevistou-se um administrativo responsável pela manutenção dos dados no DETI.

Os funcionários em causa são responsáveis por manter toda a informação do departamento actualizada na ferramenta, o que se demonstrou ser uma tarefa bastante complexa e morosa, pois existem muito poucas funcionalidades que permitem a gestão/importação de dados em massa, exigindo-se a inserção dos dados um a um, o que com a existência de um grande conjunto de docentes, disciplinas e alunos se torna uma tarefa difícil e morosa. Como exemplo, na adição de turmas é necessário adicionar, um de cada vez, todos os dados que lhe estão associados (como docentes e cursos).

Após a inserção de dados, não existem funcionalidades que permitam obter uma vista geral dos dados, possibilitando o relacionamento da informação presente. Por exemplo, não é possível ver de forma instantânea quais as turmas leccionadas por um docente, sendo necessário auscultar cada registo até encontrar o docente que pretende. Nesta perspectiva, esta ferramenta ainda está numa fase de implementação inicial, servindo apenas de portal para supervisionar todos os processos do nível superior e serviços centrais.

Outro problema encontrado foi a falta de ligação entre conceitos, levando à inserção de dados que já se encontram de outra forma em alguma das plataformas, sendo por isso complexa a gestão de incoerências.

Para uma grande quantidade de dados, a ferramenta demonstrou um desempenho um pouco lento, levando alguns segundos a carregar páginas de iniciação às secções.

Infelizmente não foi possível retirar imagens desta ferramenta devido aos dados confidenciais por ela geridos.

2.1.1.2 IST – Projecto Fénix

O projecto Fénix²(Figura 2.2) do IST “tem como objectivo o desenvolvimento de um sistema integrado de informação académica” [2], estando acessível a docentes, alunos e funcionários do próprio estabelecimento de ensino.



Figura 2.2: Logótipo do projecto Fénix do IST.

² <https://fenix-ashes.ist.utl.pt/>

Neste projecto existe um sistema dedicado à distribuição de serviço docente, que tem “como principal objectivo o desenvolvimento de uma aplicação, acessível através do portal Fénix, que auxilie os departamentos do IST a realizar a tarefa de afectação de recursos humanos para o ensino de disciplinas” [3]. Esta ferramenta foi desenvolvida durante a realização de um trabalho de final de curso, surgindo trabalhos de investigação posteriores para aperfeiçoar algum do trabalho feito [3], tratando-se de uma ferramenta em constante desenvolvimento.

O acesso à ferramenta não é possível sem que se tenha uma conta no instituto, impossibilitando a sua análise mais profunda. No entanto, é um exemplo das muitas ferramentas inacessíveis a pessoas exteriores ao local de ensino, o que acontece devido à importância dos dados que estas aplicações gerem.

2.1.2 Contexto

É comum existir na estrutura organizativa de um departamento de uma instituição de ensino uma coordenação pedagógica, que é um grupo de docentes responsável por orientar e organizar as variadas tarefas de cariz pedagógico que lá se realizam. Com o crescente número de alunos e docentes do DETI e considerando as ferramentas de produtividade ditas comuns (Microsoft Office, OpenOffice.org, entre outros), começou a notar-se algumas limitações na realização destas tarefas de uma forma eficiente, dado o crescente de requisitos e evoluções (ex.: as dissertações passaram de projectos realizados em grupos para projectos de índole pessoal). Foi assente nessa necessidade que surgiu o projecto DSD, uma plataforma capaz de centralizar e facilitar a organização de grande parte das tarefas realizadas num departamento.

2.1.3 Definição do Problema

Com o aumentar das dificuldades em organizar as tarefas de índole de organização no DETI, foram surgindo algumas ferramentas informáticas para auxiliar de alguma forma todo este complexo processo.

2.1.3.1 Ferramenta de construção de horários

O primeiro grande problema que surgiu foi a construção de horários, que tinham de ser feitos em folhas de cálculo com dados inseridos e calculados à mão, não havendo qualquer sistema que relacionasse as informações de forma automática. Foi neste contexto que surgiu a primeira ferramenta informática, com o objectivo de auxiliar a construção de horários, simplificando-a drasticamente. Ainda que a construção dos horários continue a ser feita por “mão humana”, a alteração de docentes, salas e horários das turmas é muito mais simples, sendo feito através da utilização de atalhos do teclado. Apesar de a utilização das teclas do teclado facilitar e aumentar a velocidade de execução das tarefas, sem ajuda dentro da própria aplicação, a sua utilização é extremamente dificultada. Como podemos ver na Figura 2.3, a lista de comandos ainda é algo extensa, requerendo por isso uma especialização relativa para a sua correcta utilização.

```

usage: h -d[uplicate] data_file [out_file]
usage: h -e[dit] data_file
Useful keys:
  1,2,...      change the year
  F1,F2,...    in course/year view: change the course
               in all other views: change to the corresponding history
  h            add current item to the history (not valid in course/ye
  a,c         change to course/year view
  d           change to discipline view
  r           change to room view
  l           change to teacher data view
  s           toggle between the first and second semesters
  e           toggle the show all flag
  i           toggle the inactive flag of a class
  t           toggle between the display of teacher or room informati
  u           update (save)
  return      enter change mode
  escape      exit
  left,right  select the previous/next class (course/year view)
  up,down     select the previous/next discipline or room (discipline
  home,end    select the first/last discipline or room (discipline or
  tab,backspace select the previous/next discipline (course/year view)
Useful keys in change mode
  d           enter edit teacher mode
  left,right  choose an earlier/latter time for the class
  up,down     choose an earlier/latter day for the class
  space       choose the next room for the class
  return      accept the change
  backspace   ignore the change
Useful keys in edit teacher mode
  escape      return to previous edit mode
  top,down    select a class
  left,right  select the teacher slot
  c           copy teacher
  p           paste teacher
  d           set ??? teacher
  D           set --- teacher
  return      start and stop changing a teacher
  a-z A-Z     add character to pattern string and select next match
  backspace   delete last character from the pattern string
  home        clear the entire pattern string
  tab         select next match
Things that must be changed by editing the text file
  the fixed flag of a class (if set, the class cannot change)
  the list of courses of a class
  the number of classes, their type, and their duration, of a discipline

```

Figura 2.3: Listagem extensiva de comandos disponíveis na ferramenta de construção de horários.

Outra funcionalidade a realçar são as diferentes vistas disponíveis, possibilitando filtragem de horários por semestre, ano, curso, docente, disciplinas e salas. Na Figura 2.4 é possível ver um exemplo da interface desta ferramenta. Esta funcionalidade vem adicionar algo muito importante à tradicional aproximação, pois permite uma gestão fácil de incompatibilidades dos recursos mais sensíveis (docentes, alunos e salas).

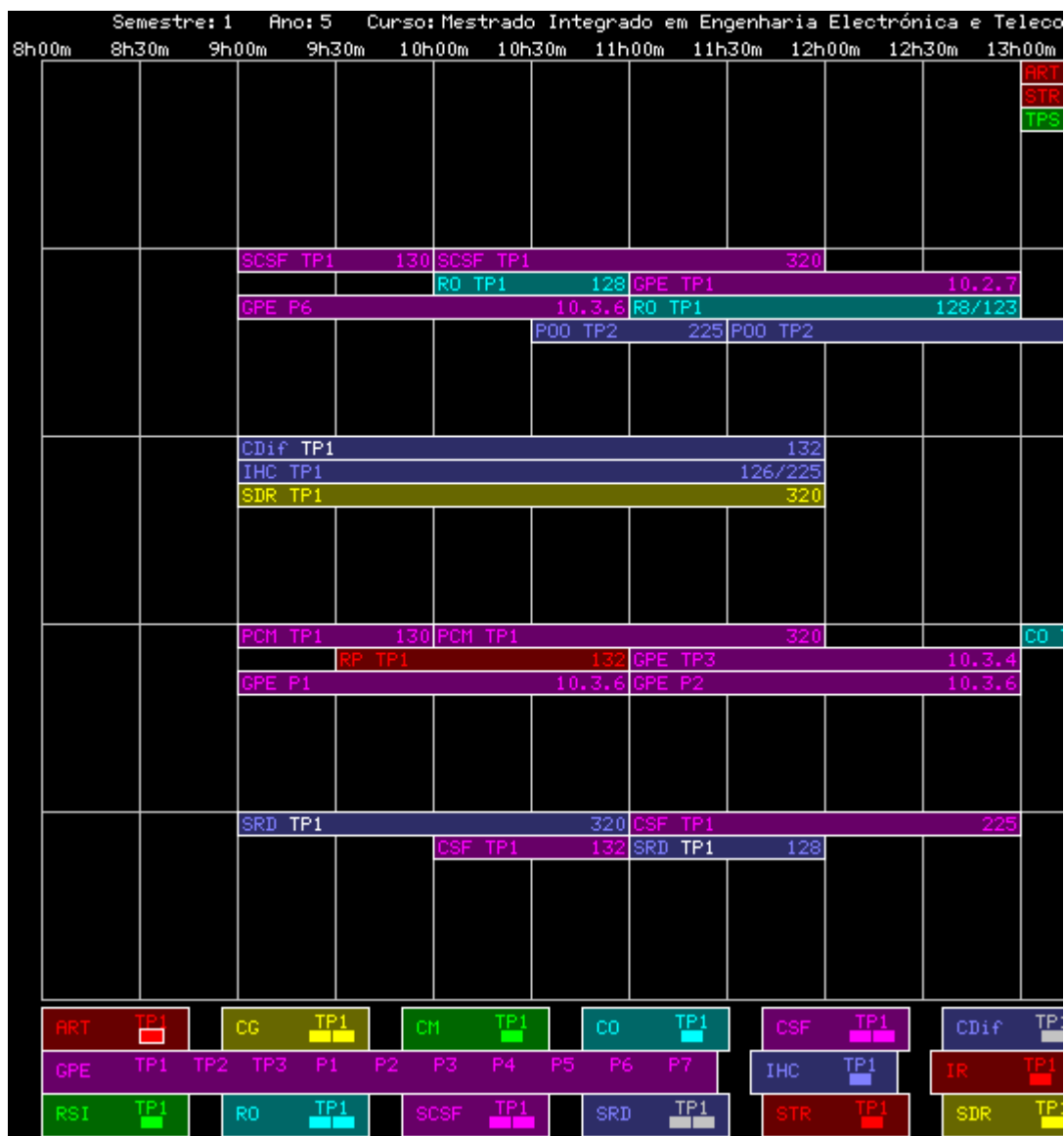


Figura 2.4: Interface da aplicação de construção de horários.

Também de salientar que a entrada e saída de dados é feita através de um ficheiro de texto simples com formatação pré-definida, permitindo a actualização dados de um ficheiro já existente.

Apesar das vantagens apresentadas, esta ferramenta ainda tem alguns problemas:

- **Não é multi-plataforma** – apenas disponível para Linux;
- **Descentralização de dados** – tratando-se de uma ferramenta de *Desktop*, os mesmos dados não estão disponíveis para todos os elementos organizadores, surgindo a necessidade de transportar o ficheiro de configuração entre as várias máquinas;
- **Pouca flexibilidade** – no caso de ser necessário adicionar ou remover informações, o processo de alteração requer o conhecimento profundo da sintaxe;

- **Interface** – a interface também é pouco adaptável, para utilizadores com ecrã de resolução menor, por vezes a informação pode não aparecer toda, dificultando a sua alteração e visualização.

2.1.3.2 Ferramenta de mestrado integrado

A escolha de disciplinas de opção por parte dos alunos processava-se da seguinte forma:

1. Docentes apresentavam propostas para leccionar as disciplinas de opção;
2. Lançamento da lista de disciplinas de opção disponíveis;
3. Aluno entregava na secretaria uma lista ordenada das suas preferências;
4. A Coordenação pedagógica tinha de ver as listas uma a uma, seriar os alunos e colocá-los nas opções;
5. Lançamento da lista de colocados em papel.

A escolha de projectos de final de curso não se processava de forma muito diferente, também através da utilização de folhas de papel. Como espectável, o trabalho desempenhado pela coordenação pedagógica era bastante penoso, tendo de analisar dezenas de listas em papel e no final nunca o resultado seria tão rigoroso como com a utilização de uma ferramenta informática. Foi neste contexto que surgiu a ferramenta de Mestrado Integrado, uma página Web (Figura 2.5) capaz de dinamizar os processos de escolhas das opções e das dissertações.

DETI - Departamento de Electrónica, Telecomunicações e Informática Universidade de Aveiro ESCOLHA DE TEMAS PARA DISSERTAÇÃO E DE OPÇÕES DE MESTRADO INTEGRADO ANO LECTIVO 2007/2008						
Escolha de temas para dissertação / Escolha de opções						
- Escolha de temas para dissertação -						
- Os alunos que chegarem a acordo com um orientador, relativamente ao tema de dissertação deverão formalizá-lo preenchendo uma declaração para o efeito, e entregando-a de seguida na Secretaria do DETI (falar com o orientador).						
- Data limite para acordos: 4 de Outubro 2007						
Lista de temas para dissertação (lista completa): (mostrar Todos ; mostrar não escolhidos ; mostrar escolhidos)						
Ref	Curso	Orientador (email)	Co-orientador	Local	Título	NMEC
1	MIEET	Nuno Borges Carvalho (nbcavvalho@ua.pt)	Dr. Pedro Miguel Cabral	Instituto de Telecomunicações	Construção do Andar de RF de um Payload de Satélite	26099
2	MIEET	Nuno Borges Carvalho (nbcavvalho@ua.pt)	João Nuno Matos	Instituto de Telecomunicações	Estudo de Rádios Epistémicos	30254
3	MIEET/MIECT	Paulo Pedreiras (pbrp@ua.pt)	Luis Almeida	DETI / IEETA	Construção de plataformas virtuais para aplicações de controlo tempo-real	17768
6	MIEET/MIECT	Ernesto Martins (erm@det.ua.pt)	Paulo Pedreiras	DETI/EETA	Controlo Distribuído de Plataformas para Experiências de Mecatrónica	31117
7	MIEET/MIECT	Arnaldo Oliveira (arnaldo.oliveira@ua.pt)	Paulo Pedreiras	DETI/EETA	Sincronização de Relógio em CAN Assistida por Hardware	30317
8	MIEET/MIECT	Arnaldo Oliveira (arnaldo.oliveira@ua.pt)	Paulo Pedreiras	DETI/EETA	Sniffer Ethernet para Redes Tempo-real Baseado em FPGA	30333
9	MIEET/MIECT	José Luis Oliveira (jlo@ua.pt)		DETI/EETA	Integração e extracção de conhecimento de bases de dados biológicas	31386
10	MIEET/MIECT	José Luis Oliveira (jlo@ua.pt)		DETI/EETA	Integração de ferramentas de análise de dados laboratoriais	27486

Figura 2.5: Interface da página Web de Mestrado Integrado.

No processo de escolhas das opções houve uma grande evolução, a lista de colocados nas opções era actualizada em tempo-real, ou seja, assim que um aluno adicionasse a sua inscrição, este podia logo ter uma ideia da sua colocação, podendo adaptar a inscrição para ficar colocado nas opções que pretendia de uma forma dinâmica e se possível construtiva. Dada a inexistência de contas pessoais, na inscrição os alunos tinham de mencionar o seu número mecanográfico sem que houvesse qualquer confirmação posterior, o que se revela uma grave brecha de segurança.

Com esta ferramenta a informatização não foi completa, dado que os professores continuaram a apresentar propostas para leccionar as opções em papel.

Na escolha das dissertações por parte dos alunos, o processo também foi parcialmente informatizado, visto que os alunos tinham de entrar em contacto com os docentes via e-mail ou pessoal, também o acordo tinha de ser assinado em papel. No entanto, a crescente lista das dissertações disponíveis com a sinalização das escolhidas foi de grande utilidade.

Esta ferramenta foi utilizada durante um ano, tendo sido obtidos resultados bastante satisfatórios, no entanto existem alguns problemas:

- **Incoerência de dados** – conceitos como docente e curso são tratados como campos de texto. Podendo acontecer um mesmo docente estar em dissertações distintas, com nomes diferentes.
- **Fraca ligação de conceitos** – como consequência da incoerência de dados, um mesmo conceito pode existir em diversos casos, mas apenas reconhecível ao olho humano;
- **Pouco extensível** – ausência de conceitos de ano e período de inscrições, o que comprometeria a sua utilização num novo ano mantendo os dados já existentes;
- **Baixo Desempenho** – utilização do Microsoft Access como SGBD compromete o desempenho, visto que este foi desenvolvido para pequenas ferramentas pessoais;
- **Baixa Segurança** – dados como palavras-chave não estão encriptados;
- **Interface** – demasiado simplista e em alguns casos pouco funcional.

2.1.3.3 Análise e relacionamento de dados

De forma a centralizar toda a informação que se encontra dispersa pelas várias ferramentas e folhas de cálculo existentes, elementos da coordenação pedagógica optaram por desenvolver uma folha de cálculo de dimensão considerável (7 MB) para ter todas as informações necessárias na distribuição do serviço docente, implementando já algum pós-processamento da informação. No entanto, como é possível ver na Figura 2.6, apesar da perspicácia e organização, a sua leitura é sempre algo complicada, agravando-se quando se tratam de centenas de linhas de informação.

	A	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	CONV	p	S2	p	S1	%	MsC PhD	ten d	final lim 4h para teses	Final Sem limite teses	T	P	T	TP	P	subj 1	T	P	T
2	C	12	10	12	14	100%	2	↓	117%	117%	0	4			8	_ACE	0	0	
3		12	11	12	8	79%	1	↑	88%	88%	1	3	2		6	_EP	0	0	
4		0	0	0	0	sabati	1	↑	0%	0,00%	0	0					0	0	
5		12	6	12	12	75%	2	↔	92%	92%	0	1			2	_ACE	1	0,7	3
6		0	0	0	1	sabati	2,5	↑	0%	0,00%	1	0	1			_MAP_I	0	0	
7		0	4,5	12	0	38%	11,5	↑	71%	133%	0	0					0	0	
8		12	7	12	5	50%	4,5	↑	83%	88%	1	0,7	3		2	_SC1	0	0	
9	C	12	11,9	12	10	91%	2	↔	108%	108%	1	1		2	2	_LAPS	0	0	
10		0	6	12	0	50%	1,5		63%	63%	0	0					0	0	
11	C	12	13	12	11	100%	0	↔	100%	100%	0	1			2	_ACE	0	4,5	
12		12	0		12	100%	5	↓	133%	142%	4	8	4		8	_CMU	0	0	
13	C	12	9	12	15	100%	2	↔	117%	117%	0	4,5			9	_P1	0	2	
14		12	7	12	13	83%	9	↔	117%	158%	2	1		4	2	_ACE	0	3	
15		12	8	12	8	67%		↑	67%	67%	0	2,7			8	_AC1	0	0	
16	CMU	12	7	12	6,9	58%	4,5	↔	91%	95%	0	0,7			2	_FR	0	0,6	
17		0	0	0	0	sabatice		↑	0%	0,00%	0	0					0	0	
18		12	8,1	12	8	67%		↑	67%	67%	0	4			8	_EP	0	0	
19	C	12	12	12	12	100%	1	↔	108%	108%	1	2		2	4	_ACE	0	6	
20	CMU	12	6	12	7	54%	0,5	↑	58%	58%	1	2	2		4	_SO	0	1	
21		12	10	12	10	83%	2	↔	100%	100%	0	0			8	_ITIC	0	1	
22		12	5,5	12	9,5	63%	1,5	↑	75%	75%	1	0,7	3		2	_EST	0,5	0	
23		12	7	12	8	63%	3	↑	88%	88%	0	1			2	_ACE	1	0	
24	C	12	11,5	12	12,8	101%	2	↔	118%	118%	2	0,7		6	2	_AC1	0	1,8	

Figura 2.6: Extracto de uma folha de cálculo para auxiliar a distribuição do serviço docente.

2.1.4 Requisitos e Objectivos

O objectivo desta plataforma passa por centralizar toda a informação que se encontra distribuída pelas várias ferramentas informáticas e folhas de cálculo. Para isso, não tem sentido utilizar o trabalho já feito, uma vez que a ferramenta de construção de horários já está um pouco desactualizada e na ferramenta de Mestrado Integrado foram encontrados vários problemas algo graves, incompatíveis com o tipo de plataforma que se pretende.

Para que a existência de uma plataforma deste tipo se justifique, há um conjunto de objectivos que têm de ser cumpridos:

- **Centralizar a informação** – os dados estarão todos no mesmo local, garantindo-se a coerência e não replicação dos dados;
- **Acesso Centralizado** – acesso à plataforma através de um único ponto, para que as actualizações sejam lançadas sem que os clientes tenham de instalar qualquer ferramenta;

- **Descentralizar o esforço** – as tarefas ficam distribuídas pelos elementos que nelas participam, repartindo-se o esforço;
- **Facilitar a análise de dados** – funcionalidades que facilitem a análise e associação de dados;
- **Transparência** – toda a informação ficará disponível para a comunidade, permitindo o acompanhamento dos processos por parte de qualquer elemento da academia.

De forma atingir estes objectivos, há um conjunto de requisitos essenciais:

- **Desempenho** – a plataforma tem de ter uma boa resposta, para que a sua utilização não seja prejudicada nem o tempo dos seus utilizadores seja desperdiçado em tarefas relativamente simples;
- **Usabilidade** – uma interface simples e eficaz será sempre um objectivo a atingir;
- **Escalabilidade** – a adição de funcionalidades terá de ser feita de forma acessível e o seu comportamento não se ressentir dessa evolução;
- **Manutenção** – a manutenção de toda a plataforma bem como das tecnologias associadas terá de ser feita da forma mais acessível possível;
- **Segurança** – manter os dados seguros também é importante, dado o conjunto de informações confidenciais que irão existir.

Assim de forma a atingir os objectivos pretendidos, nesta primeira versão a plataforma DSD irá suportar:

- **Escolha de Dissertações** – escolha de dissertações com processo totalmente informatizado;
- **Escolha de Disciplinas de Opção** – escolha das opções com processo totalmente informatizado;
- **Escolha de Turmas das Disciplinas de Opção** – suporte estrutural para a escolha de turmas em disciplinas de opção;
- **Lista de Desejos dos Docentes (*Wish List*)** – novo sistema de recolha das preferências de disciplinas e de horários dos docentes;
- **Horários** – importação e exportação de horários bem como o suporte estrutural para a sua geração automática;
- **Gestão de Entidades e Conceitos** – gestão de todas as entidades e conceitos associados às tarefas acima referidas.

Analisados alguns dos objectivos e requisitos de uma plataforma deste tipo, torna-se importante estudar as várias soluções informáticas mais utilizadas e adaptadas, apresentando-se por isso, as vantagens e desvantagens na utilização de cada uma das tecnologias. No domínio de conhecimento desta dissertação, torna-se um pouco complicado apresentar uma análise imparcial, visto que a maior parte das opiniões se baseiam muitas vezes em valores subjectivos. Contudo, tentou-se atingir alguma imparcialidade – através da análise de uma vasta gama de documentos.

2.2 Tecnologias

Nos dias que correm as aplicações de *Desktop* dominam o mercado por completo, no entanto as aplicações Web estão a ganhar cada vez mais o seu espaço, essencialmente por proporcionarem uma outra perspectiva ao nível de multi-utilizador. Desta forma, torna-se importante analisar as vantagens de utilização de cada uma das metodologias.

As aplicações de *Desktop* apresentam os seguintes pontos a seu favor [4]:

- **Aplicação como o centro das atenções** – os *browsers* bloqueiam muitas vezes e os utilizadores podem ter muitos separadores abertos, o que pode desviar as atenções da aplicação;
- **Utilização dos recursos do sistema** – utilizar a aceleração disponibilizada pelo hardware faz com que as aplicações sejam bastante mais avançadas, tornando-se mais úteis e comportando-se de forma mais eficaz.
- **Integração profunda com o *Desktop*** – arrastar objectos para o *Desktop*, salvar dados localmente e a utilização de atalhos, são algumas das muitas funcionalidades possíveis numa aplicação de *Desktop*;
- **Interface consistente com o utilizador** – nas aplicações Web, é necessária uma adaptação com a interface dos *browsers*. Nas aplicações de *Desktop*, é possível controlar todos os controlos visíveis, mantendo o utilizador focado no que se pretende.

Já as aplicações Web pelas suas características apresentam outro tipo de vantagens [4]:

- **Acesso simplificado** – o acesso às aplicações Web é muito simples, basta escrever um endereço no *browser*;
- **Execução centralizada da aplicação** – guardar a aplicação num só local, quando forem feitas actualizações, todos os utilizadores terão acesso a elas de forma automática e transparente;
- **Suporte para todas as plataformas** – aplicação disponível a todos os utilizadores, independentemente do sistema operativo utilizado;
- **Vantagem do conhecimento adquirido na utilização de um browser** – a maior parte dos utilizadores já sabe utilizar um *browser*, logo deve-se aproveitar este conhecimento em prol da aplicação;
- **Restrições de utilização** – em certos locais, como nas empresas, os utilizadores podem não ter permissões para instalar ou executar aplicações, restringindo a sua utilização.

Dadas as necessidades de acesso simplificado, execução centralizada e suporte multi-plataforma, facilmente se verifica que a plataforma DSD se identifica como uma aplicação Web. Desta forma, importa falar sobre um dos conceitos mais badalados da internet nos dias de hoje [5], a “Web 2.0”.

2.2.1 Web 2.0

O conceito de “Web 2.0” surgiu pela primeira vez pela *O'Reilly Media* e *MediaLive International*, sendo o nome de um conjunto de conferências sobre o tema [6], tornando-se com o tempo um rótulo definidor de uma série de páginas Web. No entanto, este conceito é criticado por muitos, que apenas o vêem como um mero e vazio truque de *marketing*.

Este conceito não deve ser entendido como uma nova tecnologia ou uma revolução da Web tradicional, mas sim como uma evolução natural dos paradigmas, conceitos e aplicações. Os utilizadores passam a interagir com as aplicações de forma participativa, a evolução do hardware permite que o processamento, antigamente feito exclusivamente no servidor passe a ser feito no cliente, libertando as aplicações das limitações de pedido/resposta e introduzindo a comunicação assíncrona entre cliente e servidor. Para uma melhor percepção deste conceito, foram definidas algumas regras [7]:

- **Web como plataforma** – a Web deve passar a ser vista como uma plataforma e não como um meio de comunicação entre cliente e servidor, as aplicações serão executadas nos computadores pessoais, estando disponíveis para qualquer pessoa no mundo que tenha um *browser*. A plataforma é a Web e não o sistema do utilizador;
- **Inteligência colectiva** – mais do que apenas consumir a informação disponibilizada, o utilizador agora pode fazer parte do processo de construção dos conteúdos;
- **Sabedoria das multidões** – quanto maior a utilização da aplicação, melhor será o comportamento desta, disponibilizando informações que melhor se adaptam ao perfil de cada utilizador;
- **Beta perpétuo** – as aplicações passam a ser disponibilizadas como um serviço e não em versões. O utilizador não precisa de instalar nada, tendo apenas de ir à internet para aceder ao serviço. Assim, as aplicações estão sempre em desenvolvimento, havendo melhorias com mais frequência, de acordo com o seu tempo de desenvolvimento. A resposta dos utilizadores às mudanças também é mais rápida. Se uma mudança não agrada, pode ser retirada e descartada;
- **Programação modular** – as aplicações devem ser de simples utilização e/ou aproveitadas em outras aplicações. A programação deve ser modular, pensando na reutilização e no acesso a esta aplicação por parte de outras;
- **Experiências ricas do utilizador** – tirando proveito da evolução tecnológica, parte do processamento pode ser feito no cliente, com a comunicação assíncrona com o servidor, é possível desenvolver uma interface muito mais rica. As interfaces agora ficam bem próximas das desenvolvidas para as aplicações de *Desktop*. A troca de informações passa a ser transparente para o utilizador, minimizando-se o tempo de espera;
- **Dados são o novo “Intel Inside”** – a referência ao “*Intel Inside*” é uma analogia ao facto de que, assim como os processadores são o coração dos computadores, os dados agora são o coração das aplicações. As aplicações são apenas ferramentas para manipular os dados.

Com o “Web 2.0”, houve uma grande explosão de aplicações, que revolucionaram de certa forma o mundo Web, incluindo a forma como as pessoas a utilizavam. De seguida apresentam-se algumas das aplicações “Web 2.0” de mais sucesso nas mais diversas áreas.

Social Networking (Rede Social)

As redes sociais focam-se na construção de comunidades online de pessoas que partilham os mesmos interesses e actividades, sendo disponibilizadas uma série de funcionalidades para aumentar a interacção entre os vários utilizadores, tais como partilha de ficheiros, construção de álbuns fotográficos, entre muitas outras.

Exemplos de serviços desta área são:

- Hi5 - <http://hi5.com>;
- Orkut - <http://www.orkut.com>;
- MySpace - <http://www.myspace.com>.

Wiki

É uma aplicação que permite aos utilizadores a criação e edição de conteúdos das páginas Web apenas com a utilização do *browser*. Associado a este conceito surgiram as enciclopédias online, nas quais os próprios utilizadores podem criar e editar grande parte das informações que nelas residem. No entanto, alguns utilizadores mal intencionados podem sempre adicionar informações falsas, havendo alguma dúvida relativa à fiabilidade da informação presente neste tipo de enciclopédias.

Uma das enciclopédias deste tipo de mais sucesso é a Wikipedia (<http://www.wikipedia.org>), existente em mais de 200 línguas [8], contém todo o tipo de informações. Para resolver a questão de informações falsas, existem dados que se encontram bloqueados, não sendo possível a sua edição por parte de utilizadores comuns.

Blog

É uma página Web que permite a adição e edição de conteúdos (textos, vídeos ou imagens), que por sua vez serão organizados de forma cronológica inversa (como um diário pessoal). Com a sua utilização em massa, os *blogs* tornaram-se um dos meios mais fáceis de fazer passar uma mensagem, seja por individuais ou por grupos.

Alguns exemplos são:

- Blogger - <http://www.blogger.com>;
- WordPress - <http://wordpress.org>.

Media Sharing

Nestas aplicações, o utilizador pode ver, adicionar e cotar vídeos ou imagens à plataforma, sendo possível uma pesquisa por todos os itens existentes.

Bons exemplos de aplicações nesta área são:

- YouTube - <http://www.youtube.com>;
- Google Video - <http://video.google.com>;
- Sapo Vídeos - <http://videos.sapo.pt>.

Social Bookmarking

É um método para os utilizadores de internet guardarem, organizarem e gerirem as suas páginas Web favoritas, podendo indicar cada uma delas como privada ou pública. Através de um

sistema de pesquisa e cotações, é possível fazer pesquisa pelas páginas Web favoritas de vários utilizadores.

Alguns exemplos deste tipo de aplicações:

- delicious - <http://delicious.com>;
- reddit - <http://www.reddit.com>.

Web Desktops

Nestas aplicações, pretende-se criar um ambiente *Desktop* numa página Web, disponibilizando um vasto conjunto de funcionalidades num único local. Jogos, calendários, processadores de texto e máquina de calcular são alguns dos exemplos.

Existem muitas aplicações deste género, desenvolvidas das mais diversas formas, de entre as mais conhecidas destacam-se:

- ajaxWindows - <http://www.ajaxwindows.com>;
- G.ho.st - <http://g.ho.st>;
- eyeOS - <http://eyeos.org>.

2.2.2 Servidor Web

As páginas Web são armazenadas numa máquina com um software capaz de fazer a gestão de pedidos dos clientes, retornando páginas HTML, imagens, vídeos, entre outros. No mercado existem várias ferramentas capazes de desempenhar este papel, tomando-se importante avaliar as que mais se evidenciam.

Para avaliar as ferramentas que mais marcam o mercado nesta área, analisaram-se vários estudos [9] [10] [11] [12]. No entanto, notou-se que os números apresentados variam de estudo para estudo, conforme os interesses envolvidos. Assim, foram considerados quatro estudos distintos, considerando os três servidores Web mais cotados em cada um.

	Apache	IIS	Netscape	Google	nginx	Zeus
Netcraft (Agosto 2008)[9]	49,82%	34,88%	-	5,94%	-	-
SecuritySpace (Agosto 2008)[10]	73,27%	17,91%	-	-	-	0,40%
Google (Junho 2007)[11]	66%	23%	-	-	4%	-
Port80 (Julho 2007)[12]	24,9%	55%	3,3%	-	-	-

Tabela 2.1: Cotas de mercado dos vários servidores Web segundo quatro estudos distintos.

Como se pode analisar na Tabela 2.1 existe uma grande variação dos dados obtidos, no entanto há um factor comum a todos os estudos, o Apache e o IIS são os servidores Web mais utilizados, merecendo uma atenção especial. Para a avaliação destas ferramentas foram considerados sete pontos essenciais, capazes de dar uma ideia geral das tecnologias [13] [14]:

- **Compatibilidade** – sistemas operativos suportados;
- **Componentes Dinâmicos** – linguagens de programação de componentes dinâmicos suportadas;
- **Ambiente de Execução** – características internas importantes de realçar;
- **Segurança e Autenticação** – características de segurança e autenticação a realçar;

- **Desempenho** – técnicas utilizadas para ajustar o desempenho;
- **Gestão** – facilidade/dificuldade na gestão das características do servidor Web.
- **Custos** – custos associados à sua utilização;

2.2.2.1 Apache HTTP Server

*Apache HTTP Server*³ (Figura 2.7) é um projecto integrante da *Apache Software Foundation*, sendo um servidor Web livre de grande sucesso. A sua primeira versão foi baseada no HTTPD (*Hyper Text Transfer Protocol Daemon*), revolucionando o mundo da internet. O seu historial já é longo, apesar de apenas se encontrar na versão 2.2, teve grandes evoluções ao longo dos tempos, sendo uma das ferramentas mais potentes nesta área.



Figura 2.7: Logótipo do projecto Apache HTTP Server.

Compatibilidade

Uma das suas principais características é a independência do sistema operativo, funcionando em Windows, Linux, Unix e MAC OS. No entanto, esta portabilidade também traz alguns custos, não havendo uma total integração com as funcionalidades dos sistemas operativos, surgindo a necessidade de utilizar módulos extra.

Ambiente de Execução

Esta independência deve-se em muito ao APR (*Apache Portable Runtime*) [13], que permite correr o Apache em qualquer sistema operativo com compilador⁴ de linguagem C, o que acontece em grande parte dos casos. Para a gestão de processos, o Apache utiliza o próprio gestor de processos do sistema operativo em causa [13], sendo a recepção e processamento de pedidos feita por processos distintos, permitindo a recepção de uma grande quantidade de pedidos.

Componentes Dinâmicos

Com a independência acima referida, obviamente que o Apache também foi feito para suportar uma vasta gama de linguagens de programação. PHP, CGI, Java, C/C++, Perl, Python e JSP [13] [14] são algumas das linguagens suportadas. Também é possível o suporte de ASP através dos módulos Apache::ASP⁵ ou Mod mono⁶ [13]. No entanto, o suporte completo da *Framework* .NET ainda não é possível.

Segurança e Autenticação

A segurança e administração do Apache não se encontram integradas com as funcionalidades do sistema operativo, no entanto existem módulos que ajudam a realizar estas

³ <http://httpd.apache.org/>

⁴ Programa (ou conjunto de programas) que permite transformar o código escrito numa linguagem de programação em linguagem de máquina, gerando código binário que pode ser executado pela máquina.

⁵ <http://www.apache-asp.org/>

⁶ http://www.mono-project.com/Mod_mono

tarefas, tais como LDAP⁷ e *Active Directory*⁸ [15]. No entanto, existem alguns problemas, por exemplo, pode acontecer um utilizador autenticar-se no Apache e ter acesso a ficheiros que não devia, uma vez que o Apache ignora as permissões definidas pelo sistema operativo.

A utilização de SSL⁹, IPsec¹⁰ e IPv6 é suportada.

Desempenho

O Apache tem um conjunto de ferramentas desenhadas para melhorar o seu desempenho. Por exemplo, o `mod_perl` e o `mod_php` permitem a execução de páginas dinâmicas de forma quase tão rápida como as páginas estáticas, isto acontece porque o processamento das páginas é feito logo na recepção [15]. Obviamente que a utilização do armazenamento automático de dados na cache também ajuda, permitindo a execução de pedidos sem que seja necessária a execução de uma aplicação externa [15].

Gestão

Toda a gestão do Apache é feita através de um ficheiro de texto, no entanto existem ferramentas de linhas de comando, *Web* e *Desktop* (ex.: *ApacheConf*) que facilitam esta configuração, sem ser necessário navegar num ficheiro com centenas de linhas. Embora possa parecer estranho, a utilização de um ficheiro de texto traz as suas vantagens, por exemplo, a partilha de configurações entre máquinas é muito facilitada, bastando apenas copiar e colar as configurações.

Custos

O Apache é completamente gratuito, bastando aceder à página do projecto, fazer o *download* da aplicação e instalar num dos sistemas operativos acima indicados.

É de notar que o código do Apache é aberto, querendo isto dizer que qualquer pessoa pode dar a sua contribuição no desenvolvimento do *software*.

2.2.2.2 Microsoft IIS

O *Internet Information Services*¹¹ (IIS - Figura 2.8) da Microsoft surgiu após o Apache, tendo a sua distribuição sido feita como um aplicativo opcional dos vários sistemas operativos que foram lançados (desde o Windows NT 4.0)[13]. Na realidade a primeira versão a ser realmente considerada para uso empresarial foi a versão 4. Com o lançamento da versão 6, o IIS sofreu grandes alterações, ficando ao nível dos melhores servidores Web. A versão actual é o IIS7 (Figura 2.8) que é distribuída com o Windows Server 2008 e o Windows Vista.

⁷ *Lightweight Directory Access Protocol*, protocolo para actualizar e pesquisar directórios.

⁸ Armazena informações da rede de computadores e disponibiliza-as a utilizadores e administradores.

⁹ *Secure Sockets Layer*, protocolo que permite comunicações seguras

¹⁰ Extensão ao protocolo IP que permite uma maior privacidade aos utilizadores

¹¹ <http://www.iis.net/>



Figura 2.8: Imagem de marca da versão 7.0 do *Internet Information Services*.

Compatibilidade

Uma das principais deficiências do IIS é facto de este só funcionar em sistemas operativos Windows, ainda assim, a mesma versão do IIS não é compatível com qualquer uma das versões do sistema operativo. Por exemplo, o IIS 7.0 não funciona no Windows Server 2003, sendo exigida a sua instalação no Windows Server 2008. Isto acontece porque esta tecnologia é profundamente integrada com o sistema operativo [16], o que traz grandes vantagens.

Ambiente de Execução

No IIS a gestão dos processos é feita de forma idêntica ao Apache, ou seja, existem processos separados para a recepção e o processamento dos pedidos, o que permite a aceitação de pedidos mesmo quando o IIS *Worker Process* está tecnicamente desligado [16].

Componentes Dinâmicos

Apesar das limitações no sistema operativo, o IIS suporta uma vasta gama de linguagens de programação. VBScript, Visual Basic, Java, C/C++, Perl, Python, JSP, PHP, CGI, ASP e a Framework .NET [16] são exemplos disso mesmo.

Segurança e Autenticação

Devido à forte relação com o sistema operativo, a segurança do servidor Web está integrada com a segurança do sistema operativo, assim, toda a gestão de permissões é bastante simplificada. Por exemplo, os utilizadores do servidor Web serão os utilizadores do sistema operativo, sendo as suas permissões as mesmas para ambos os sistemas.

Desempenho

O IIS também contém um conjunto de ferramentas para melhorar o desempenho, exemplos disso são as extensões ASP e ISAPI (*Internet Server Application Programming Interface*) [14], que permitem a execução de pedidos pelos próprios processos que os receberam. Para além disso, o armazenamento automático de dados na cache também é uma das funcionalidades suportadas, fazendo com que as respostas a pedidos sejam efectuadas de forma mais rápida.

Gestão

Para guardar as configurações, o IIS utiliza um ficheiro em XML [16] (pela mesma razão mencionada para o Apache), no entanto, também existem funcionalidades do sistema operativo que permitem facilitar a sua configuração.

Custos

O IIS é grátis, no entanto, apresenta-se incorporado no sistema operativo sendo por isso necessário obter licença do sistema operativo para que se possa utilizar o servidor Web.

2.2.3 Sistema de Gestão de Bases de Dados (SGBD)

Numa aplicação de grande complexidade, surge a necessidade de guardar dados de forma persistente, querendo isto dizer que os dados continuam a existir mesmo após:

- Terminar a aplicação que os gere;
- Terminar a sessão do utilizador;
- Desligar o computador.

Isto acontece dada a necessidade de adição de dados à página Web, ocorrendo uma troca de informações entre utilizadores. Neste contexto, surge o conceito de Base de Dados, que não é mais do que uma “coleção de dados estruturados, organizados e armazenados de forma persistente” [17].

Com a crescente dimensão das bases de dados, começaram a surgir ferramentas informáticas capazes de facilitar a gestão das mesmas, fornecendo uma “interface entre os dados que são armazenados fisicamente na base de dados e o utilizador (pessoa ou aplicação) desses dados. Assim, o utilizador deixa de ter de saber qual o formato dos dados, como e onde são armazenados, pesquisados, ordenados, etc.” [17] (Figura 2.9). A uma ferramenta com estas características é dado o nome de Sistema de Gestão de Bases de Dados (SGBD).

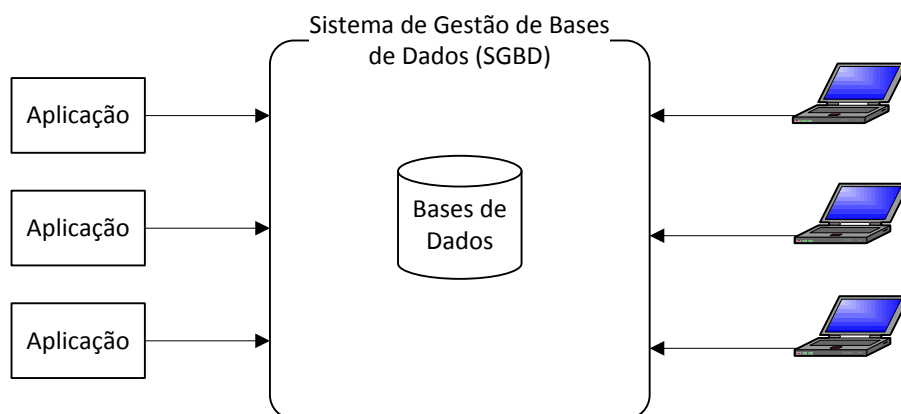


Figura 2.9: Abstracção fornecida pelo SGBD.

Para o acesso e gestão das bases de dados, uma das linguagens de maior sucesso é o SQL (*Structured Query Language*), que é uma linguagem declarativa que utiliza um método revolucionário. Tipicamente, com estas linguagens tinha de se indicar o caminho para chegar a um resultado, com o SQL, é indicada a forma do resultado em si. Com o evoluir dos SGBD e das aplicações, o SQL sofreu várias reformas, sempre com o objectivo de facilitar a interacção entre aplicações e bases de dados.

Com a utilização de um SGBD pretendem-se atingir quatro objectivos essenciais:

- **Eficiência** – ser capaz de aceder, processar e alterar grandes volumes de dados de forma eficiente;
- **Robustez** – manter os dados de forma consistente, mesmo após falhas de hardware ou erros de software;

- **Controlo de acessos** – controlar o acesso de múltiplos utilizadores aos dados de forma consistente e apenas a utilizadores autorizados;
- **Persistência** – manter os dados durante longos períodos, independentemente das aplicações que a eles acedem.

Uma vez atingidos estes objectivos fundamentais, a utilização de um SGBD traz uma série de vantagens [18]:

- **Partilha de informação** – a base de dados pertence a toda a empresa e não apenas a alguns elementos;
- **Controlo da redundância** – múltiplas cópias dos mesmos dados só são guardadas em casos de extrema necessidade;
- **Consistência dos dados** – um valor só é alterado uma vez, e todos os utilizadores terão acesso a esse novo valor;
- **Melhor controlo da concorrência** – execução de blocos de código como um só (transacções), não ocorrendo a perda de dados, aquando da alteração de um mesmo conjunto de dados por diferentes utilizadores em simultâneo;
- **Melhor segurança dos dados** – mecanismos para controlar acessos à base de dados, para que não existam acessos não autorizados que possam danificar os dados. Ainda assim, existem mecanismos de encriptação dos dados para os acessos não autorizados;
- **Melhoria da integridade dos dados** – possibilidade de definição de regras nos dados, por exemplo, um determinado valor não pode ultrapassar um máximo previamente definido. ACID (Atómico, Consistente, Isolado, Durável) são as regras de ouro para garantir a integridade dos dados;
- **Balanceamento dos conflitos de requisitos** – gestão das necessidades de cada utilizador, evitando ao máximo a existência de conflitos;
- **Economias de escala** – uma única plataforma para a manutenção dos dados relevantes da organização;
- **Backup & recovery** – os dados são salvaguardados periodicamente, permitindo o restauro de dados em caso de falha.

No entanto, associado à facilidade também existem várias desvantagens na sua utilização [18]:

- **Custos do SGBD** – aplicações muito complexas e conseqüentemente muito caras;
- **Custos do hardware** – necessidade de boas características ao nível do hardware para que o desempenho seja satisfatório;
- **Custos de programação** – como são aplicações muito complexas, é necessário algum conhecimento para que a sua utilização seja a melhor;
- **Vulnerabilidade acrescida** – dada a centralização dos recursos, haverá sempre riscos de segurança.

Ao longo dos tempos surgiram muitas ferramentas que se propunham a implementar a ideologia de um bom SGBD descrita acima, obviamente umas tiveram mais sucesso que outras. Dentro do grupo das soluções proprietárias as que mais se destacam são o SQL Server da

Microsoft e o Oracle, duas ferramentas extremamente potentes e de alguma incidência neste mercado [19]. Já no código livre, a que mais se destaca é o sem dúvida o MySQL da *Sun Microsystems*, também de grande incidência no mercado, que tem um ponto muito forte a seu favor, ser uma ferramenta “grátis”.

Para analisar estas ferramentas, foram considerados seis pontos demonstrativos das suas características:

- **Portabilidade** – capacidade do software se adaptar aos mais variados sistemas operativos;
- **Desenvolvimento** – facilidade ou dificuldade de desenvolver aplicações que utilizam esta ferramenta;
- **Desempenho** – analisar as características que proporcionam um bom ou mau desempenho ao SGBD. Como a quantidade de dados nas bases de dados varia com muita facilidade, também é necessário avaliar a sua escalabilidade¹².
- **Segurança** – características da segurança fornecida pelo SGBD;
- **Manutenção** – dificuldade de gerir a ferramenta ao longo da sua utilização;
- **Custos** – custos monetários associados à utilização do *software*;

Importa realçar que devido à grande complexidade destas ferramentas, para este documento, apenas interessa fazer uma análise superficial das funcionalidades e características de cada uma, não sendo dado grande realce a pormenores de funcionamento interno.

2.2.3.1 SQL Server

SQL Server¹³ (Figura 2.10) é a solução apresentada pela Microsoft para o armazenamento e gestão de dados. É um SGBD bastante complexo com o objectivo de cumprir grande parte dos objectivos definidos, no entanto, um pouco pesado a nível computacional.



Figura 2.10: Marca comercial do SGBD SQL Server da Microsoft.

Portabilidade

Como em todas as ferramentas da Microsoft, a portabilidade também é um problema no seu SGBD, visto que apenas funciona em sistemas operativos da própria empresa, Windows.

A sua utilização é possível através da maior parte das linguagens de programação, como Java e PHP, embora a integração com os produtos da própria empresa (Framework .NET) seja feita de forma mais profunda, possibilitando a utilização de funcionalidades complexas de forma mais simplificada.

¹² “Capacidade de manipular uma crescente quantidade de dados de forma uniforme, ou estar preparado para o crescimento dos mesmos” [56]

¹³ <http://www.microsoft.com/sqlserver/2008/en/us/default.aspx>

Desenvolvimento

A linguagem utilizada no SQL Server é o *Transact-SQL*, que estende as funcionalidades do SQL, melhorando-a em vários aspectos [20]:

- Utilização de variáveis locais;
- Utilização de funções de suporte ao processamento de strings, datas, entre outros;
- Melhorias em algumas funcionalidades (como no UPDATE e DELETE);
- Linguagem para controlo de fluxos (como BEGIN, END, BREAK, CONTINUE, entre outros).

Com estas melhorias, torna-se uma linguagem de utilização muito simples e ao mesmo tempo poderosa, sendo uma das vantagens da utilização deste SGBD.

Acompanhando o SQL Server existe um conjunto de ferramentas que facilitam o desenvolvimento de aplicações, automatizando várias tarefas de extrema complexidade. Para programadores mais avançados, é possível a abstracção destas funcionalidades, aumentando-se o nível de personalização.

Desempenho

Este SGBD exige boas características a nível de hardware, podendo comprometer o seu desempenho caso isto não aconteça.

Também é necessário tomar em consideração a forma de programar a aplicação desenvolvida, visto que diferentes formas de programar poderão ter diferentes tipos de desempenho, aconselhando-se a consulta prévia de documentação antes de se iniciar o desenvolvimento de uma aplicação.

Relativamente à escalabilidade, o SQL Server adapta-se facilmente ao crescimento das bases de dados, adaptando-se também ao crescente número de máquinas através de várias estratégias de replicação de dados, *Snapshot*, *Transactional* e *Merge* [21]. Na *Transactional* existe um servidor e clientes, o servidor gere a escrita de alterações e os clientes contactam-no para obter as actualizações. Na *Snapshot* é enviado uma visão geral da base de dados para os clientes. Na *Merge*, os servidores podem trocar actualizações entre si, tornando-se ideal para grandes distâncias geográficas entre máquinas.

Segurança

Gestão de utilizadores e encriptação de dados são algumas das funcionalidades do SQL Server para garantir a segurança dos dados que gere, garantindo ao máximo sua integridade.

Manutenção

O *SQL Server Management Studio* é a ferramenta que possibilita a gestão e configuração de todas as funcionalidades do SQL Server, simples de usar e extremamente útil. Com ele vem um conjunto de funcionalidades para facilitar a sua manutenção, apresentando relatórios de análise que alertam o programador para situações que possam influenciar o seu desempenho.

Para prevenção de problemas maiores aquando de certas situações indesejadas, é possível o *backup* dos dados de forma periódica, para que posteriormente se possa recuperar os dados após a resolução do problema.

Custos

Como referido anteriormente, este SGBD exige boas características a nível de hardware, requerendo-se um pequeno esforço na aquisição do servidor sobre o qual o SQL Server irá funcionar.

Como ferramenta proprietária que é, a aquisição deste produto não é grátis, sendo necessário o pagamento de uma licença. É de notar que, como apenas funciona em sistemas operativos da própria empresa, a aquisição de um sistema operativo Windows também é exigida.

Existe uma versão grátis direccionada para desenvolvimento caseiro, o SQL Server Express, no entanto contem algumas limitações, como o tamanho máximo da base de dados de 4GB [22].

2.2.3.2 Oracle

O Oracle¹⁴ (Figura 2.11) foi um dos primeiros SGBD que surgiram no mercado, tendo já largos anos de experiencia nesta área, sendo um dos que tem mais sucesso apesar dos grandes custos que lhe estão associados.



Figura 2.11: Logótipo do SGBD Oracle.

Portabilidade

A portabilidade é claramente uma vantagem do Oracle, visto que pode ser utilizado em praticamente todos os sistemas operativos, como o Windows, Linux, Unix e Solaris [23].

A sua utilização também é suportada por várias linguagens de programação, havendo uma integração especial com o Java, dado que este se encontra embebido no Oracle [23].

Desenvolvimento

O Oracle tem três linguagens embebidas [23]:

- **SQL** – a tradicional linguagem para aceder às bases de dados;
- **PL/SQL (Procedural Language/Structured Query Language)** – linguagem baseada em procedimentos que estende as funcionalidades do SQL, tornando-se extremamente poderosa. No entanto, peca por ser um pouco complicada, não sendo de fácil aprendizagem;
- **Java** – linguagem orientada a objectos muito conhecida e utilizada, integrada no Oracle suportando diferentes pontos de vista dos mesmos dados e pesquisas altamente complexas, tornando-se ideal para trabalhar do lado do servidor.

Esta relação com o Java só é possível devido à integração de uma JVM (*Java Virtual Machine*) com o Oracle, possibilitando a chamada de programas Java a partir do PL/SQL e vice-versa, o que se traduz numa grande vantagem a nível da integração. Assim, aconselha-se a utilização de um IDE Java, em especial destaque o Oracle JDeveloper, que é um IDE desenvolvido

¹⁴ <http://www.oracle.com>

pela própria empresa que contém ferramentas que facilitam ainda mais a integração entre as tecnologias.

Também existem ferramentas extra para integração com outros IDE, possibilitando o desenvolvimento de aplicações com .NET e PHP.

Desempenho

O Oracle também exige boas características de hardware, podendo colocar em causa o seu desempenho caso isso não aconteça.

Como o SQL Server, a forma como as aplicações são programadas influencia o desempenho do SGBD.

O suporte para sistemas distribuídos é garantido, comportando-se de forma notável mesmo quando o número de máquinas aumenta, sabendo lidar com a replicação das bases de dados.

Segurança

O Oracle também fornece várias funcionalidades para garantir a maior segurança dos dados, gestão dos utilizadores e encriptação de dados são algumas das suportadas.

Manutenção

O *Oracle Enterprise Manager* é a ferramenta para a configuração e gestão de todo o SGBD, facilitando configurações de grande complexidade, no entanto, é necessário um conhecimento um pouco profundo do SGBD para trabalhar com esta ferramenta. Também o *Backup & Restore* de dados é suportado.

Custos

O Oracle sempre se caracterizou por ser um dos SGBD mais caros, apesar de o custo associado à licença ter vindo a diminuir nos últimos anos. A nível de hardware, também é necessário algum esforço financeiro.

Para o Oracle também existe uma versão grátis, o Oracle Express, que tal como o SQL Server Express também tem várias limitações, tanto a nível de dimensão das bases de dados como do próprio desempenho [24].

2.2.3.3 MySQL

O MySQL¹⁵ (Figura 2.12) é um SGBD grátis de grande sucesso, caracterizando-se pelo código aberto. É mais direccionado para pequenas bases de dados, não suportando de forma profunda algumas funcionalidades essenciais para bases de dados de grande dimensão.



Figura 2.12: Imagem de marca do SGBD MySQL da Sun Microsystems.

¹⁵ <http://www.mysql.com>

Portabilidade

Tal como o Oracle, o *MySQL* também pode ser utilizado na maior parte dos sistemas operativos.

A sua utilização é suportada por uma vasta gama de linguagens, havendo uma ligação especial com o PHP.

Desenvolvimento

A linguagem utilizada é o dialecto *MySQL*, que também estende o *SQL*, contudo continua a ser muito simples, não sendo tão poderosa quanto as outras já aqui apresentadas. No entanto, existem esforços para levar esta linguagem mais além, prova disso é o suporte a variadas funcionalidades (*stored procedures*, *views* e *triggers*) na versão 5 [25].

Como já foi dito, existe uma melhor interacção com o PHP, todavia, o desenvolvimento é suportado para grande parte das linguagens, existindo pacotes extra que permitem a sua utilização.

Desempenho

Também este SGBD suporta a replicação das bases de dados, embora de forma mais limitada nas suas funcionalidades. A replicação apenas é suportada num sentido, um servidor comporta-se como *master* enquanto os outros se comportam como *slaves*. O *master* escreve as actualizações no registo e os *slaves* entram periodicamente em contacto com o *master* para obter os dados actualizados [26]. Para situações de maior distribuição geográfica esta não é a melhor solução, podendo haver perca de desempenho.

Segurança

Assim como os outros SGBD, o *MySQL* tem várias funcionalidades para garantir a maior segurança dos dados, sendo também bastante competente nesta área.

Manutenção

Para a manutenção e gestão, existem muitas ferramentas visuais, que facilitam todo e este processo. A destacar, o *MySQL Workbench*¹⁶ que permite a construção de bases de dados com interfaces gráficas de simples utilização.

Como um bom SGBD, o *Backup&Restore* das bases de dados também é devidamente suportado.

Custos

O *MySQL* é grátis, no entanto para versões com suporte e actualizações automáticas (*MySQL Enterprise*) é exigido o pagamento de uma licença. Obviamente que utilizadores especializados não necessitarão de pagar qualquer licença, visto que eles mesmos tratarão de actualizar e manter a aplicação a funcionar da melhor forma.

No que diz respeito a hardware, este SGBD é bastante leve, não requerendo qualquer tipo de atenção especial.

¹⁶ <http://dev.mysql.com/workbench/>

2.2.4 Páginas Web Dinâmicas

Na realidade, toda a base das páginas Web é o HTML (*HyperText Markup Language*), que não é mais do que uma linguagem descritiva, que nos permite introduzir hiperligações, definir tamanhos dos tipos de letras, intercalar texto com imagens, entre muitas outras funcionalidades. No entanto, tudo isto é feito de forma fixa e pouco inteligente, não sendo possível, por exemplo, realizar um simples cálculo matemático ou gerar uma página HTML do nada. A este tipo de páginas Web, dá-se a designação de páginas Web estáticas [27].

Com a evolução que houve nesta área, passaram-se a desenvolver grandes plataformas Web, surgindo a necessidade de funcionalidades que o HTML não suportava por si só, como por exemplo, a utilização de *scripts* para gerar páginas de forma automatizada. No fundo, as páginas Web passam a ter respostas diferentes perante contextos e/ou condições distintas, daí a designação de páginas Web dinâmicas [27].

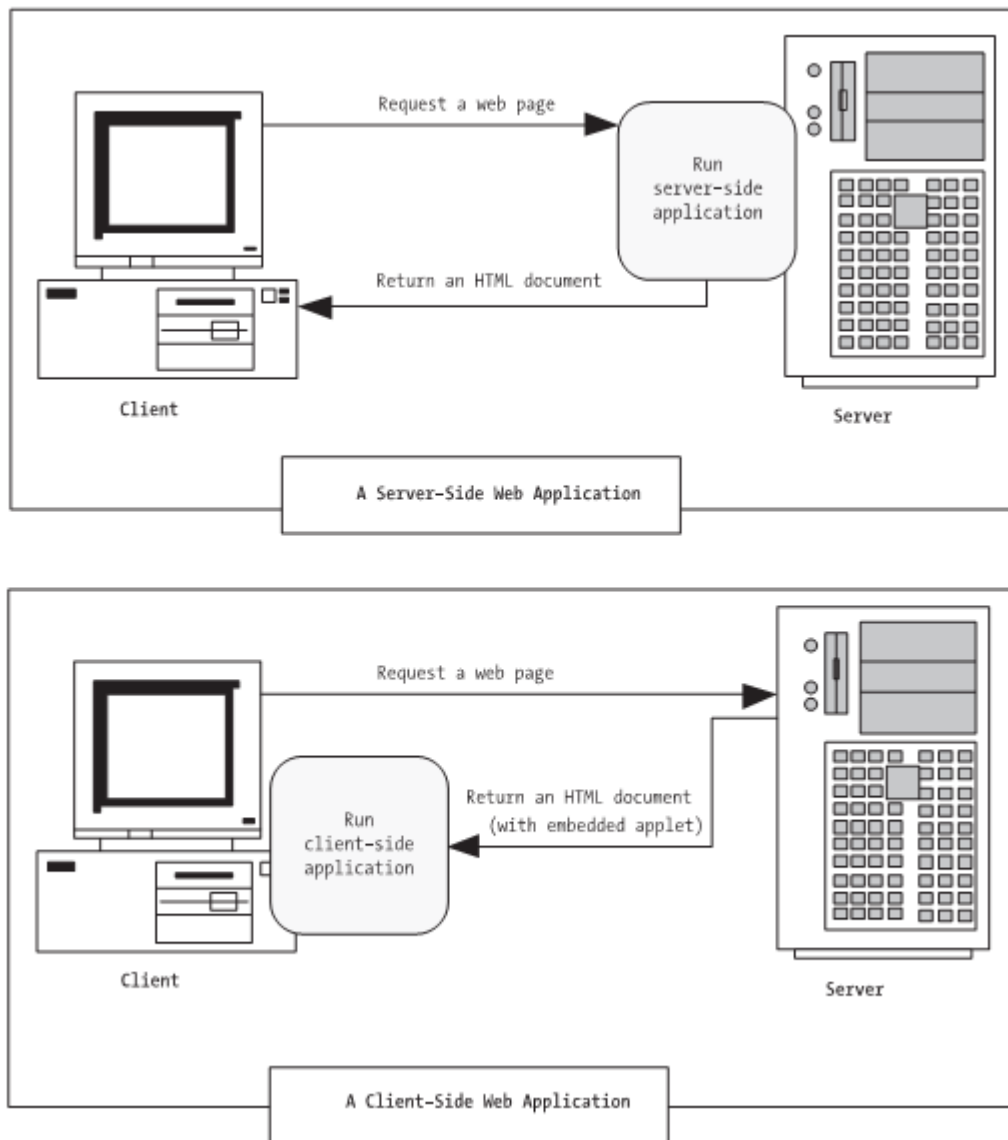


Figura 2.13: Funcionamento das páginas Web de cliente (em cima) e das páginas Web de servidor (em baixo) [27].

Para verificarmos as diferenças de desenvolvimento em ambas as tecnologias, nada melhor que um pequeno exemplo, imagine que queria fazer uma página para colocar notícias, atualizando-as diariamente. Com a utilização de páginas estáticas, teria de criar um ficheiro HTML para cada notícia, ou editar as já existentes, o que se torna bastante repetitivo. Já com páginas dinâmicas, podia criar uma base de dados e com apenas um script recolher as notícias que desejaria, o que se torna bastante mais simples, facilitando a manutenção das mesmas.

Como podemos ver na Figura 2.13, existem dois tipos de páginas Web dinâmicas: de cliente e de servidor. Nas páginas dinâmicas de cliente todo o processamento dos efeitos e das funcionalidades é feita pelo browser do cliente, enquanto nas de servidor essa tarefa fica a cargo do servidor da página Web. Usos típicos das páginas de cliente são os efeitos especiais como *rollovers*, controlo de janelas, controlo de formulários, cálculos, entre muitos outros. Já as páginas de servidor são especialmente utilizadas no acesso a informação centralizada (bases de dados no servidor) e na execução de cálculos que não podem ser realizados no cliente por razões de segurança. O facto de haver uma distinção deste tipo de páginas, não impede que estes não possam ser utilizados em simultâneo.

Com o surgir das páginas dinâmicas, a internet deixou de ter um cariz meramente informativo, passando a ser um meio de comércio e publicidade por parte das empresas, e como consequência, a complexidade das páginas aumentou de forma extraordinária. Nesta perspectiva, começaram a ser desenvolvidas plataformas para facilitar e dinamizar o desenvolvimento de páginas Web. De entre as mais conhecidas, destacam-se:

- ASP (*Active Server Pages*)¹⁷
- PHP (*PHP Hypertext Preprocessor*)¹⁸
- JSP (*JavaServer Pages*)¹⁹

Dada a variedade de tecnologias existentes para o mesmo fim, torna-se importante analisar cada uma delas, encontrando as suas vantagens e desvantagens. Com este objectivo, foram definidos cinco pontos essenciais demonstrativos das suas capacidades e características:

- **Portabilidade** – capacidade de se adaptar aos mais variados sistemas operativos e servidores Web;
- **Desenvolvimento** – dificuldade de aprendizagem, velocidade de desenvolvimento de aplicações e ferramentas de desenvolvimento existentes;
- **Desempenho** – performance obtida com utilização da tecnologia;
- **Manutenção** – dificuldades e/ou facilidades de fazer a manutenção das aplicações, flexibilidade e escalabilidade;
- **Custos** – custos monetários associados à sua utilização;

¹⁷ <http://www.asp.net/>

¹⁸ <http://www.php.net/>

¹⁹ <http://java.sun.com/products/jsp/>

2.2.4.1 ASP

ASP (*Active Server Pages* - Figura 2.14) é a solução apresentada pela Microsoft, tendo como versão mais recente o ASP.net 3.5, que tem uma serie de inovações com vista a facilitar e simplificar o desenvolvimento de páginas Web.



Figura 2.14: Marca comercial do ASP.net.

Portabilidade

A portabilidade é um dos principais problemas do ASP, visto que as suas funcionalidades só estão disponíveis na totalidade caso este seja utilizado sobre software criado pela própria empresa, ou seja, só funciona em sistemas operativos Windows e em servidores Web IIS. No entanto, tem surgido projectos bem interessantes que tomam possível a utilização desta tecnologia noutros softwares. Exemplo disso é o projecto Mono, que “disponibiliza o software necessário para desenvolver e executar aplicações .NET em Linux, Solaris, Mac OS X, Windows e Unix” [28], ainda que as funcionalidades não estejam disponíveis na sua totalidade. No entanto, já é demonstrada uma vontade de levar esta plataforma para além do Windows.

Desenvolvimento

Uma das principais características do ASP.net é a independência da linguagem utilizada, suportando uma série de linguagens distintas, como C#, VB.net, J#, C++, entre outras. Como podemos ver na Figura 2.15, isto só acontece porque independentemente da linguagem origem, o código será compilado transformado-se em *Intermediate Language Code* (ILC). A CLR (*Common Language Runtime*) apenas utiliza ILC, não reconhecendo qual a linguagem .NET que foi utilizada no desenvolvimento da aplicação. Finalmente, antes da execução da aplicação, a CLR irá transformar o ILC em código de máquina nativo, sendo este guardado na cache da máquina para poder ser reutilizado, aumentando o desempenho.

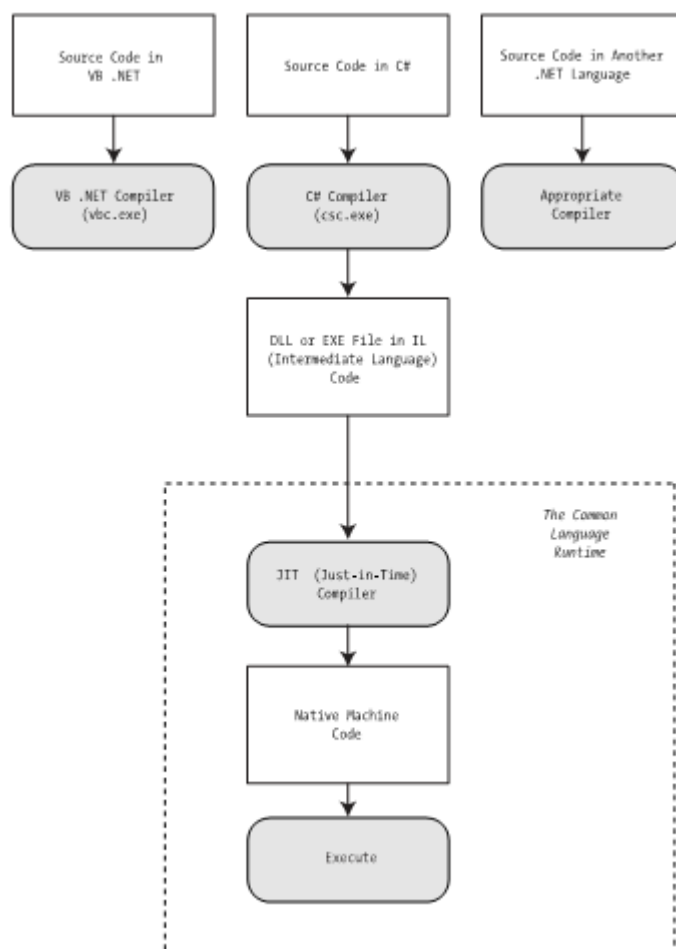


Figura 2.15: Arquitectura para compilar e executar aplicações .NET [28].

Esta independência de linguagem é um grande ponto a favor da plataforma .NET, visto não ser necessária a aprendizagem de novas linguagens para a utilizar (a não ser que nunca tenha programado em nenhuma das disponíveis).

De forma a mascarar toda a complexidade desta plataforma, o desenvolvimento de aplicações em ASP.net é extremamente facilitado pelo excelente IDE (*Integrated Development Environment*) criado pela própria empresa, o Visual Studio. Esta ferramenta é bastante versátil, sendo possível realizar uma mesma tarefa de formas totalmente distintas, adaptando-se aos vários tipos de exigências dos mais diferentes utilizadores. No entanto, isto não quer dizer que o desenvolvimento de aplicações ASP.net seja de extrema facilidade, podia ser bem pior caso não houvesse uma plataforma de desenvolvimento deste nível.

Também de realçar a excelente separação de código, havendo ficheiros distintos para a interface e para a lógica de funcionamento. Associando este facto à programação orientada a objectos, obtêm-se uma excelente organização, aumentando a sua escalabilidade.

Finalmente uma excelente gestão de erros e uma documentação bastante vasta também são factores muito importantes para facilitar o desenvolvimento de aplicações.

Desempenho

A independência de linguagem de programação traz grandes vantagens ao .NET, no entanto, o seu desempenho pode ser um pouco influenciado negativamente, consequência do seu funcionamento interno. No entanto, é possível colocar no servidor versões previamente compiladas, não sendo necessário compilar o código em tempo-real, aumentando o seu desempenho. Ainda assim, há a necessidade de servidores com boas características a nível de hardware para que o desempenho seja exemplar.

Manutenção

Devido às suas características que proporcionam uma excelente organização e a uma série de aplicações que proporcionam uma forte integração com o sistema operativo e o servidor Web, a manutenção destas aplicações é extremamente simples, não surgindo grandes dificuldades aquando da correcção de problemas ou no aumento de funcionalidades.

Custos

Associado a esta plataforma estão associados custos de licença relativamente elevados, visto que todas as ferramentas utilizadas terão de ser adquiridas, desde o sistema operativo ao IDE. Para além disso, grande parte das bibliotecas e *Frameworks* extras que lhe estão associadas, também são proprietárias.

2.2.4.2 PHP

PHP (*PHP Hypertext Preprocessor*, Figura 2.16) foi a resposta dada por um conjunto de programadores para corresponder às necessidades do mercado nesta área [29]. Como se trata de uma plataforma grátis de código aberto, passou por várias fases distintas de desenvolvimento, sempre com o objectivo de responder melhor às necessidades dos programadores. Esta acabou por ser uma das principais características do PHP, sendo até capaz de competir com plataformas desenvolvidas por grandes empresas de escala mundial.



Figura 2.16: Logótipo do PHP.

Portabilidade

A portabilidade favorece enormemente o PHP, pois é suportado em grande parte dos sistemas operativos, como o Windows, Linux e MAC OS [26]. Para além dos sistemas operativos, também nos servidores Web o suporte é grande, sendo até suportada a sua utilização no IIS.

Desenvolvimento

O desenvolvimento de aplicações é ao mesmo tempo simples e complicado. Para principiantes torna-se uma linguagem bastante simples de aprender, com um código bastante perceptível. No entanto, como o código da interface e da lógica de funcionamento se encontra todo no mesmo ficheiro, ainda que o suporte à programação orientada por objectos exista, para aplicações mais complexas, tudo se torna um tanto confuso e menos perceptível.

Também existem bons IDEs para o PHP, no entanto, regra geral precisam de aplicações extra para que estejam ao nível das melhores (ex.: Eclipse²⁰ e Netbeans²¹). Um dos mais completos é o Zend Studio²², que carece de licenciamento.

A documentação também é vasta, embora grande parte seja gerida pelos próprios programadores da plataforma e/ou pelos programadores que utilizam a plataforma, podendo por vezes a informação não ser de total confiança.

Desempenho

O PHP tem um excelente desempenho, isto acontece pois não há a necessidade compilar código em tempo real, sendo apenas interpretado, o que requer menos recursos, sendo consequentemente mais rápido [30].

Manutenção

Associada ao tipo de organização subjacente, a adição ou remoção de funcionalidades a aplicações de grande dimensão torna-se difícil para aplicações de grande dimensão.

Custos

Sendo grátis, e com alta portabilidade, é dada uma grande liberdade de escolha ao programador, podendo desenvolver aplicações em PHP sem que tenha algum custo associado. Também de realçar que grande parte das bibliotecas e *Frameworks* existentes também são grátis.

2.2.4.3 JavaServer Pages

JSP (JavaServer Pages) é uma tecnologia Java que permite o desenvolvimento de aplicações Web dinâmicas. Caracteriza-se essencialmente pela utilização de *servlets*, que de forma muito simplista, são trechos de código que adicionam funcionalidades ao servidor Web [31].

Portabilidade

Como Java que é, a sua portabilidade é garantida, sendo suportada em todos os sistemas operativos. Nos servidores Web, também existe suporte para grande parte deles, existindo alguns especialmente desenvolvidos para esta linguagem, como o Apache Tomcat²³ e o Glassfish²⁴, dois servidores Web grátis que são soluções bastante válidas para desempenhar este papel. De notar que a utilização de JSP também é possível com o IIS, através da utilização de módulos extra para estabelecer a ligação com um servidor Web que execute JSP de forma nativa [32].

Desenvolvimento

Como podemos ver na Figura 2.17 (lado esquerdo) gerar código HTML a partir do java seria muito trabalhoso e complexo, optando-se por dividir a apresentação da lógica funcional, o que traz grandes vantagens, como por exemplo a divisão de tarefas entre equipas. Criou-se também o conceito de *JavaBean* para a lógica funcional, que permite encapsular vários objectos num só, designado de *bean*. Isto permite transportar vários objectos de uma só vez, sem que se tenha de o fazer um de cada vez.

²⁰ <http://www.eclipse.org/>

²¹ <http://www.netbeans.org/>

²² <http://www.zend.com>

²³ <http://tomcat.apache.org>

²⁴ <https://glassfish.dev.java.net>

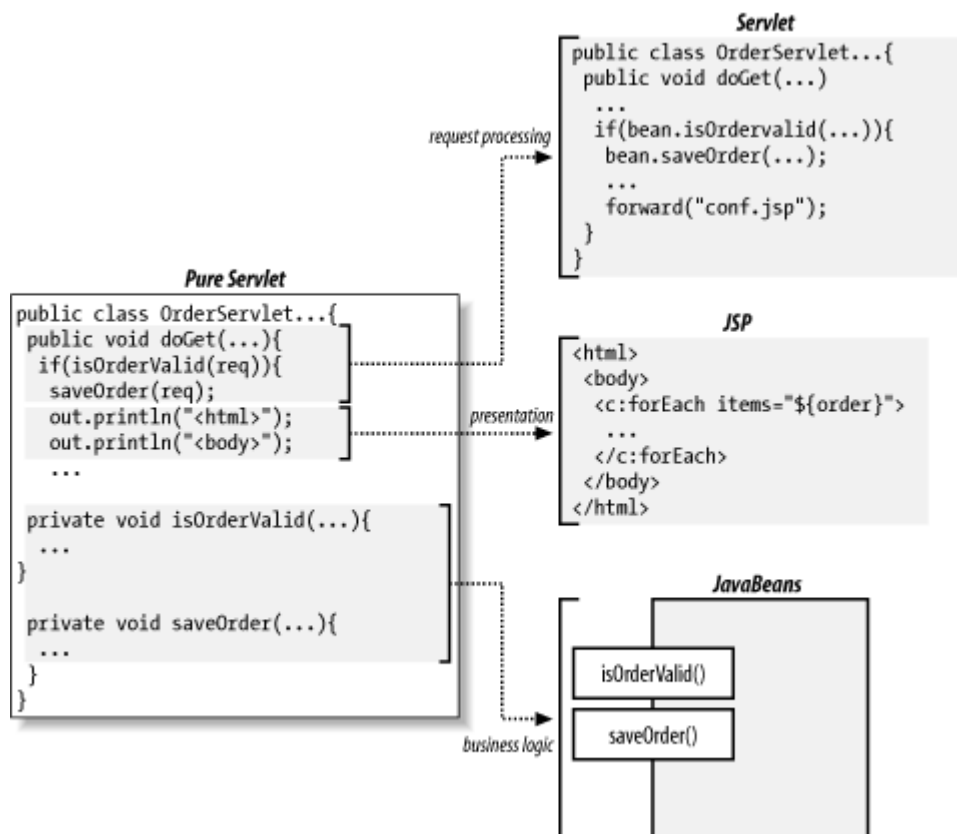


Figura 2.17: Organização do código em pedidos de processamento (*request processing*), lógica de funcionamento (*business logic*) e interface (*presentation*) [31].

Com esta organização tudo fica devidamente estruturado, embora possa ser um pouco complicado para principiantes, dados os conhecimentos algo avançados que lhe estão associados. Ainda assim, a própria linguagem JSP é um pouco complexa, pois utiliza o mesmo conceito de orientação por objectos que o Java, o que faz com que sejam tipicamente necessárias mais linhas de código para uma tarefa que nas outras linguagens Web, tornando o processo de desenvolvimento mais demorado.

Para desenvolver aplicações em JSP existe uma vasta gama de IDEs, na sua maioria grátis, destacando-se o Eclipse e o Netbeans, duas ferramentas extremamente potentes que facilitam em muito a elaboração de tarefas de grande complexidade, proporcionando um ambiente de desenvolvimento extremamente integrado com todas as funcionalidades necessárias (ex.: integração com os servidores Web).

Desempenho

Esta tecnologia acaba por ter um funcionamento idêntico ao ASP.net, como podemos ver na Figura 2.18, depois de o cliente fazer o pedido ao servidor, existem duas fases distintas: tradução e processamento do pedido. Na fase de tradução todos os elementos JSP são convertidos em código Java, que corresponde ao comportamento dinâmico pretendido. Após isso, o código java é compilado, gerando os ficheiros *class*, a partir dos quais é possível a execução na fase de processamento do pedido.

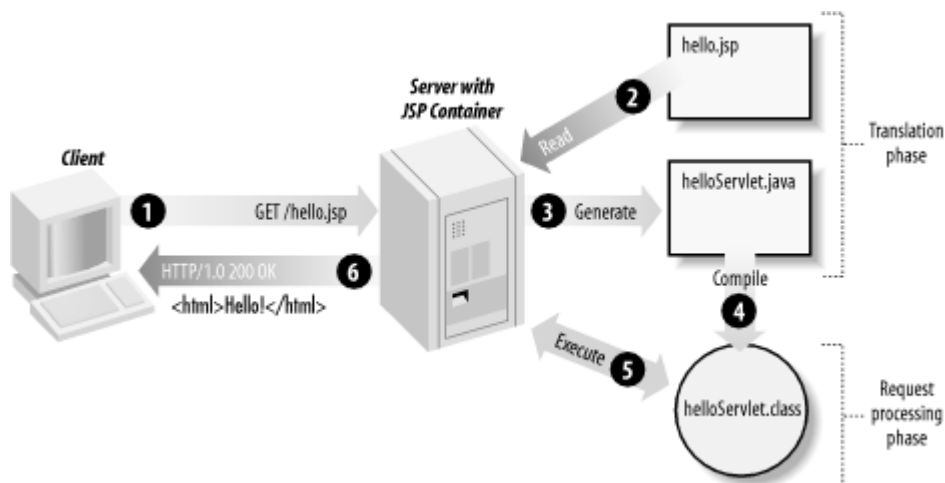


Figura 2.18: Arquitectura para compilar e executar aplicações JSP [31].

Como esperado, todo este processo é um pouco demorado, especialmente a fase de tradução. No entanto, assim como no ASP.net, é possível colocar ficheiros pré-compilados no servidor, o seu desempenho é melhorado visto que a fase de tradução deixa de ser necessária.

Manutenção

A organização do JSP já induz uma estrutura rigorosamente definida e exigente, contudo, sem alguma experiência por parte do programador pode-se tornar um pouco complexo adicionar, alterar ou remover funcionalidades.

Custos

Assim como o Java o JSP também é grátis, não havendo quaisquer custos associados ao desenvolvimento de aplicações, dado que os IDEs, servidores Web ou bibliotecas que lhe estão associados não requerem a aquisição de qualquer licença.

2.2.5 Interface

Para satisfazer as necessidades dos utilizadores, uma interface agradável passa sempre por uma das suas exigências. Para cumprir este requisito, a simples utilização de HTML torna-se complicada, surgindo a necessidade de centralizar os estilos utilizados. Desta forma, em 1997, surgiu uma das primeiras recomendações para a utilização de *HTML*, a separação da estrutura e apresentação de uma página Web [33]. Foi nessa expressa necessidade que surgiu o *CSS (Cascading Style Sheets)*, com a sua utilização, todo o código associado à apresentação estará separado do código restante. Se compararmos a Figura 2.19 com a Figura 2.20 (neste exemplo o *CSS* está no mesmo documento que o *HTML* para simplificar a visualização, tipicamente, o *CSS* deverá estar num ficheiro à parte), facilmente verificamos que antes da existência do *CSS* a manutenção das páginas era bastante mais complexa, tornando o código menos perceptível, surgindo até replicação.

```

<head>
</head>
<body>
<li><font FONT SIZE="3"
      FACE="Verdana, Arial, Helvetica, sans-serif"
      COLOR="#0000FF"> Olá Mundo 1! </font> </li>
<li><font FONT SIZE="3"
      FACE="Verdana, Arial, Helvetica, sans-serif"
      COLOR="#0000FF"> Olá Mundo 2! </font> </li>
<li><font FONT SIZE="3"
      FACE="Verdana, Arial, Helvetica, sans-serif"
      COLOR="#0000FF"> Olá Mundo 3! </font> </li>
</body>
</html>

```

Figura 2.19: Dificuldade de gerir o aspecto de páginas Web antes do CSS.

```

<head>
  <style type="text/css">
    li{
      font-size:16px;
      font:Verdana, Arial, Helvetica, sans-serif;
      color: #0000FF }
  </style>
</head>
<body>
  <li>Olá Mundo 1!</li>
  <li>Olá Mundo 2!</li>
  <li>Olá Mundo 3!</li>
</body>
</html>

```

Figura 2.20: Exemplo simples de utilização de CSS.

Com a utilização de CSS é possível personalizar uma série de características das páginas Web, tais como: tipos de letra, cores, espaçamentos, dimensões, margens, entre muitas outras.

A utilização desta metodologia traz grandes vantagens para os programadores:

- Páginas desenvolvidas em CSS são mais fáceis de alterar;
- Comunicação entre programadores mais fácil, pois todos seguem o mesmo padrão;
- Controlo facilitado sobre as características das páginas Web.

Contudo, existe um grande problema na utilização de CSS, os diferentes *browsers* podem não interpretar da mesma forma um conjunto de instruções CSS, o que faz com que o resultado obtido possa ser diferente de browser para browser. Por esta razão, aconselha-se o teste das aplicações que utilizam CSS no maior número de browsers possível.

2.2.6 Dinamizar a Interface

No desenvolvimento de toda e qualquer plataforma Web, há sempre um termo que paira nas cabeças dos Web *designers*, usabilidade. “Simplificar, rentabilizar, otimizar, facilitar, melhorar, acelerar são verbos que gravitam em torno do conceito de usabilidade. Mais usabilidade é sinónima de maior flexibilidade e de maior interacção” [34].

Com a simples utilização de HTML por vezes a navegação nas páginas Web torna-se cansativa e frustrante, dado o constante carregamento total das páginas a cada acção. Com a utilização de outras tecnologias é possível ultrapassar esse problema, afectando-se apenas as zonas com as quais o utilizador acabou de interagir, não tendo de esperar que toda a página volte

a ser carregada. Assim, foram consideradas três soluções bastante válidas para cumprir esta tarefa: AJAX, Flash e Silverlight. Existem várias semelhanças e diferenças gerais que serão cuidadosamente analisadas.

2.2.6.1 AJAX

AJAX (*Asynchronous JavaScript and XML*) é um método de construção de páginas Web que processa os pedidos do utilizador de forma imediata. O AJAX não é uma tecnologia, mas sim a combinação de várias tecnologias, as quais já suportadas pelos vários *browsers* existentes. Cada uma delas terá uma função específica:

- **XHTML** (*Extensible Hypertext Markup Language*), **CSS** – apresentação de dados;
- **DOM** (*Document Object Model*) – interacção e apresentação de dados dinâmica;
- **XML** (*Extensible Markup Language*), **XSLT** (*Extensible Stylesheet Language Transformations*), **JSON** (*JavaScript Object Notation*) – manipulação e troca de dados;
- **XMLHttpRequest**, **IFrames** – comunicações assíncronas;
- **JavaScript** – interacção com o cliente, fazendo com que tudo funcione como um só.

Nas páginas Web tradicionais, quando o utilizador executa uma qualquer acção sobre a página Web, tem de esperar que toda a página seja novamente carregada para obter o resultado da sua acção. Já com AJAX isso não acontece, quando o utilizador executa uma acção, obtêm o resultado de imediato, sem que tenha de esperar uns segundos para que a página seja totalmente carregada (Figura 2.21). Um bom exemplo da utilização deste conjunto de tecnologias é o *Google Maps*²⁵, onde os utilizadores podem manipular o mapa em tempo-real.

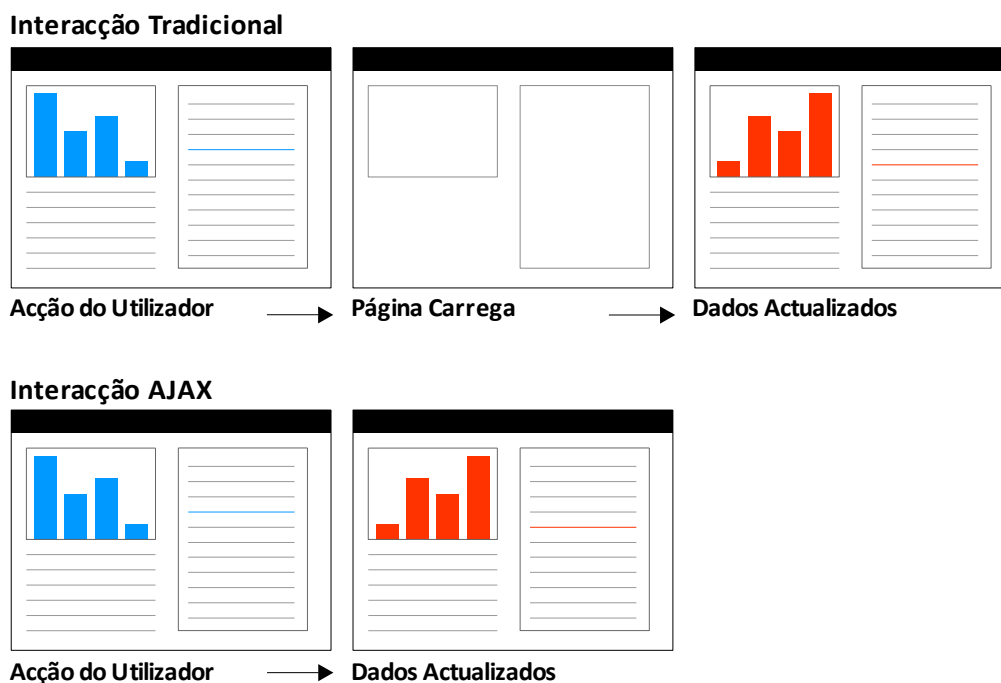


Figura 2.21: Interação tradicional e interação AJAX.

²⁵ <http://maps.google.com/>

Para obter este resultado, o funcionamento das páginas Web com AJAX é um pouco diferente. Como podemos ver na Figura 2.22, nas páginas Web tradicionais, quando o utilizador executa uma acção sobre a interface, é enviado um *HTTP Request* para o servidor. Após o servidor receber os dados, processa-os devolvendo uma página *HTML* para o cliente. A este tipo de comunicação dá-se a designação de Comunicação Síncrona, visto que o utilizador tem de esperar que o servidor envie uma resposta para poder continuar a utilizar a página Web.

Com o *AJAX* é adicionada uma camada intermédia entre o utilizador e o servidor, responsável pela apresentação de dados ao utilizador e pela comunicação com o servidor. Desta forma, as acções do utilizador deixam de gerar um *HTTP Request* mas sim uma chamada em forma de *JavaScript* para a camada *AJAX*. A camada *AJAX* irá processar os dados e só em caso de necessidade irá comunicar com o servidor (por exemplo, na verificação de formulários não é necessário comunicar com o servidor). Caso exista essa necessidade, a camada *AJAX* enviará um *HTTP Request* para o servidor (utilizando o objecto *XMLHttpRequest*), que por sua vez responderá com uma mensagem em *XML*, *HTML*, texto simples ou um objecto *JSON*, que será devidamente interpretada pela camada *AJAX*. A este tipo de comunicação dá-se o nome de Comunicação Assíncrona, uma vez que mesmo que o utilizador efectue uma acção, não terá de esperar pelo resultado podendo continuar a interagir com a página Web.

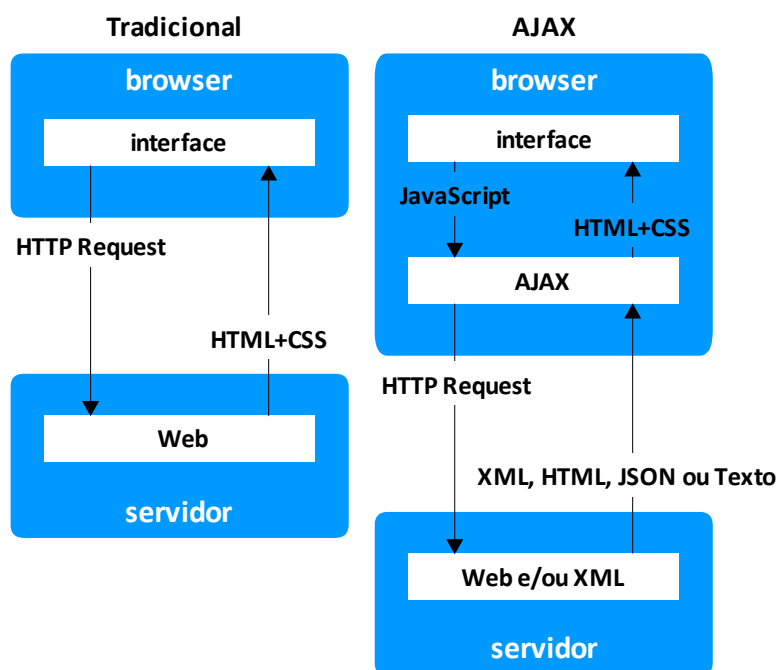


Figura 2.22: Funcionamento tradicional (à esquerda) e funcionamento do *AJAX* (à direita).

Como título de curiosidade, actualmente o *JavaScript* pode ser substituído por outras linguagens de programação para desempenhar o mesmo papel, como por exemplo o *VBScript* [35].

Com este novo tipo de funcionamento surgiram uma série de novas funcionalidades complexas com vista a facilitar e dinamizar a utilização das páginas Web, apesar de o utilizador comum não se aperceber. De entre as várias funcionalidades, são de destacar as seguintes:

- **Validação de formulários em tempo-real:** dados complexos (códigos postais, datas, entre outros) podem ser validados pelo servidor antes de o utilizador submeter o formulário;
- **Formulários auto-completáveis:** os dados são completados de forma automática enquanto o utilizador vai escrevendo;
- **Carregamento parcial de dados:** as páginas podem carregar secções de dados em processos secundários, fazendo com que a página carregue mais rapidamente;
- **Efeitos visuais sofisticados:** controlos com aspecto visual mais rico;
- **Submissão parcial de dados:** submeter dados para o servidor sem recarregar toda a página;
- **Actualização de dados:** actualização automática de dados;
- **Mashups:** as páginas podem ter componentes provenientes de outras páginas na internet;
- **Páginas como Aplicações:** com AJAX podem-se criar páginas parecidas com aplicações de *Desktop*.

A utilização deste tipo de funcionalidades traduz-se numa série de vantagens para os utilizadores:

- **Redução no consumo da Largura de Banda** – como é possível carregar apenas algumas partes da página Web, isto diminui significativamente a quantidade de dados trocados entre cliente e servidor;
- **Redução de conexões ao servidor** – uma vez que os scripts e as páginas de estilos só terão de ser carregadas uma única vez;
- **Maior interactividade** – a utilização de comunicações assíncronas torna a utilização das páginas mais simples, rápida e interactiva;
- **Acessibilidade** – o *JavaScript* fornece um conjunto de funcionalidades extra que permite uma maior interactividade com as páginas, para que estas também estejam acessíveis a pessoas portadoras de deficiência;
- **Plugin** – não é necessário qualquer tipo de *plug-in* para navegar em páginas que utilizam AJAX.

No entanto, também existem uma série de desvantagens que nos levam a tolerar a sua utilização:

- **Integração com o Browser** – as páginas criadas dinamicamente não serão registadas no historial do *browser*, o que faz com que o botão voltar não funcione da forma esperada. Foram desenvolvidas algumas soluções que passam pela utilização de *IFrames* (componente que permite incorporar outros ficheiros HTML na mesma página) invisíveis. Os favoritos também deixam de funcionar para certas partes da página. Também já foi encontrada uma solução que passa pela utilização de identificadores de fragmentos (a parte de um endereço HTML após o carácter '#');
- **Tempo de Resposta** – deverá ser dado *feedback* ao utilizador quando são pedidos dados ao servidor, caso contrário, o utilizador não saberá que os dados estão a ser processados. Isto acontece pois a tradicional barra de carregamento da página existente no *browser* não disponibiliza qualquer *feedback* sobre estes pedidos;

- **Motores de Pesquisa** – como os motores de busca não executam código JavaScript necessário para aceder a dados da página, a informação poderá não ser indexada pelos mesmos;
- **Implementações do JavaScript** – os *browsers* implementam JavaScript de formas distintas, o que poderá traduzir-se em problemas de compatibilidade. Por esta razão, as aplicações desenvolvidas em *AJAX* deverão ser testadas no maior número de *browsers* possível;
- **Controlos Incompatíveis** – dependendo da *Framework* *AJAX* utilizada, alguns controlos HTML – como upload de ficheiros e árvores – não são compatíveis com a utilização de *AJAX*. Já foram encontradas algumas soluções, no entanto, a maior parte delas são proprietárias.

Com o surgir desta tecnologia houve uma explosão de *Frameworks* e Bibliotecas para desenvolver aplicações em *AJAX*. Obviamente que foram desenvolvidas em várias áreas, para se adaptarem aos mais variados ambientes de programação. De entre a vasta lista [36], destacam-se as seguintes:

- *jQuery*²⁶ (*JavaScript*) – *framework* em *JavaScript* de código aberto;
- *ASP.net AJAX*²⁷ (*.NET*) – *framework* *.Net* grátis;
- *Google Web Toolkit*²⁸ (*Java*) – biblioteca em *Java* de código aberto;
- *XAJAX*²⁹ (*PHP*) – biblioteca em *PHP* de código aberto.

2.2.6.2 Flash

O Adobe Flash³⁰ (Figura 2.23) é uma ferramenta maioritariamente vectorial da Adobe (suportando também imagens e vídeos) que permite a criação de animações e de qualquer tipo de funcionalidades, não se restringindo apenas a aplicações Web. A sua aplicabilidade é bastante vasta, permitindo a construção de aplicações de alguma complexidade, como jogos interactivos (ex. <http://miniclip.com>), páginas Web (ex. <http://2advanced.com>), animações (ex. <http://happytreefriends.atomfilms.com>), entre outros.



Figura 2.23: Imagem de marca do Adobe Flash.

²⁶ <http://jquery.com>

²⁷ <http://www.asp.net/ajax/>

²⁸ <http://code.google.com/webtoolkit/>

²⁹ <http://xajaxproject.org/>

³⁰ <http://www.adobe.com/products/flash/>

Para correr aplicações desenvolvidas em Flash é necessária a instalação de uma ferramenta capaz de ler os ficheiros gerados (extensão *.swf), o Flash Player. Na Web, esta ferramenta funciona como um *Plug-in* (ferramenta que estende as funcionalidades de uma outra de maior dimensão) dos *browsers*, estando disponível para todos os sistemas operativos.

Inicialmente o Flash era mais direccionado para *designers*, tendo uma interface de desenvolvimento orientada à linha temporal (Figura 2.24), com um cariz de desenho vectorial bastante acentuado. No entanto, com a grande utilização desta tecnologia, a necessidade de uma vertente mais orientada à programação foi-se acentuando, surgindo o Adobe Flex³¹, que “oferece um modelo moderno de linguagem e programação baseado em padrões que suporta modelos comuns de design. MXML, linguagem declarativa baseada em XML, é usada para descrever os comportamentos e o aspecto da interface, e ActionScript 3, linguagem de programação avançada, usada para criar a lógica do cliente” [37]. Mais tarde, também surgiu a necessidade de desenvolver aplicações de *Desktop*, sendo criado o Adobe AIR³², que torna possível a criação de aplicações de *Desktop* utilizando Flash, Flex ou AJAX.

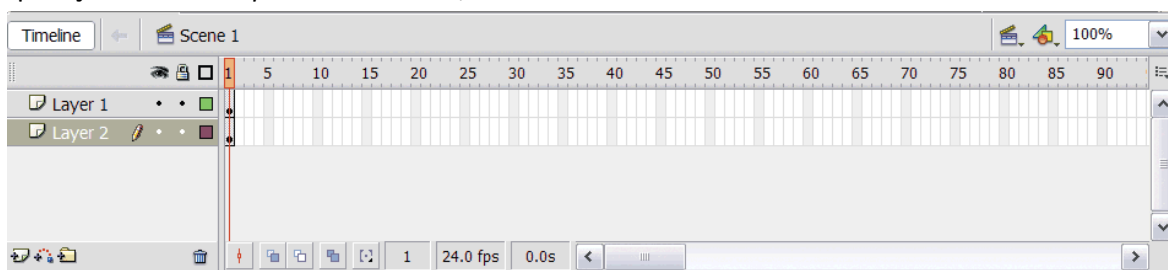


Figura 2.24: Linha temporal do Adobe Flash CS3.

Com a utilização do Flash é possível o desenvolvimento de todo o tipo de funcionalidades, existindo uma grande liberdade criativa proporcionada pela alta interactividade que esta tecnologia proporciona. A sua utilização não requer que a página seja inteiramente desenvolvida nesta tecnologia, podem uma página em HTML ter apenas alguns blocos em *Flash*.

Desta forma, através da sua utilização foram encontradas as seguintes vantagens:

- **Interactividade** – suporte integrado de imagens, vídeo, áudio, microfone, câmara de vídeo e o desenvolvimento de animações nunca antes vistas tornaram-se nas principais características do Flash;
- **Interfaces atractivas e usáveis** – com a vertente vectorial, torna-se bastante simples o desenvolvimento de interfaces ricas a nível gráfico, estando sempre presente a vertente de animação, que traz uma “espectacularidade” nunca antes vista em páginas Web. Para além disso, o facto de não ser necessário de recarregar toda a página a cada acção também é um ponto a seu favor;
- **Compatibilidade** – independentemente do *browser* ou do sistema operativo, o Flash tem sempre o mesmo comportamento, apresentando sempre as interfaces, funcionalidades e animações como o projectado;

³¹ <http://www.adobe.com/products/flex/>

³² <http://www.adobe.com/products/air/>

- **Acessibilidade** – contém um conjunto de funcionalidade que torna possível a interacção de todo o tipo de utilizadores com as páginas.

No entanto, tanta interactividade e atractividade trás os seus pontos fracos:

- **Pesado** – os ficheiros exportados são um pouco grandes, demorando a ser carregados;
- **Plug-in** – a necessidade da instalação de uma ferramenta extra acaba por ser um dos principais impeditivos da utilização desta tecnologia, não havendo por vezes vontade por parte dos utilizadores de o fazer (esta faceta verifica-se mais a nível empresarial);
- **Integração com o Browser** – ao navegarmos numa página inteiramente desenvolvida com esta tecnologia, existe apenas um URL (*Uniform Resource Locator*) para todos os conteúdos que a página possa conter. Desta forma, apenas ficará guardado um único URL no historial do browser, o que faz com que o botão voltar não funcione como o normal. No entanto, já foram desenvolvidas soluções para este problema utilizando identificadores de fragmentos.
- **Motores de Pesquisa** – alguns já conseguem fazer a indexação dos textos contidos em ficheiros flash, no entanto, esta tecnologia ainda não se encontra muito avançada. Também de realçar, que as páginas desenvolvidas inteiramente com a esta tecnologia não terão endereços para conteúdos específicos, não sendo possível o redireccionamento directo para eles através de apontadores devolvidos por uma pesquisa.

Para desenvolver aplicações Flash podem ser utilizados dois IDEs distintos:

- **Adobe Flash CS3** – orientado à linha temporal, sendo mais indicado para *designers*. Só existem versões para Windows e Mac, embora possa ser executado em sistemas UNIX através da utilização do WINE (projecto para sistemas operativos UNIX que permite executar software desenvolvido para Windows);
- **Adobe Flex 3** - orientado à programação, sendo mais indicado para programadores avançados. Existem versões finais para Windows e Mac, estando a ser desenvolvida uma versão para sistemas UNIX. Este IDE é baseado no Eclipse.

Como título de curiosidade, recentemente foi lançada a versão CS4 do Adobe Flash, que traz grandes progressos na animação, trazendo um novíssimo sistema de Cinemática Inversa (“determinar os parâmetros de um corpo articulado dada a localização de um extremo do corpo” [38]) denominado de “*Bones Tool*” e uma ferramenta para animação 3D.

2.2.6.3 Silverlight

*Silverlight*³³ (Figura 2.25) é a solução apresentada pela Microsoft para competir com o Flash, que na altura não tinha um competidor directo ao seu nível, dominando por completo o mercado nesta área. Assim, como ambas as tecnologias foram desenvolvidas com o mesmo objectivo, é normal serem encontradas muitas semelhanças. Desta forma, através da utilização do

³³ <http://silverlight.net/>

Silverlight podemos produzir vídeo e áudio, utilizar imagens, construir interfaces interactivas e animações.



Figura 2.25: Logótipo do Silverlight.

Para correr aplicações desenvolvidas com *Silverlight* também é necessária a instalação de um plug-in nos browsers, sendo de notar que a execução destas aplicações fora do browser é impossível. Ao contrário do Flash, o modelo de publicação não é feito para um ficheiro único, encontrando-se dividido em vários ficheiros XAML, imagens, vídeos, áudio, JavaScript e binários.

Uma das suas principais características é o XAML (*Extensible Application Markup Language*) [39], linguagem utilizada para descrever a interface. Como é uma *Markup Language* (como o HTML) é possível ser gerado código dinamicamente do lado do servidor, independentemente da linguagem de programação neste utilizada. Esta capacidade fornece uma grande flexibilidade para os programadores, dando a possibilidade de gerar componentes *Silverlight* a partir da linguagem de servidor.

Também é de realçar que as linguagens de programação utilizadas não são propriamente novas: XAML, *JavaScript*, *C#* e/ou *VB.NET* [39]. Querendo isto dizer que para desenvolver aplicações *Silverlight* não é necessária a aprendizagem de novas linguagens de programação, exigindo-se apenas a aprendizagem do encadeamento das mesmas.

Apesar de o *Silverlight* ter vindo para competir com o Flash, este não colmatou as principais deficiências que já existiam. Integração com o *browser*, motores de pesquisa, uso de *plug-in* e ficheiros pesados são problemas que também residem nesta tecnologia. No entanto, ainda foi adicionado um problema algo grave, a compatibilidade. Os sistemas operativos suportados são o Microsoft Windows e o Mac OS. Apesar de o Linux ter sido deixado de parte, o Moonlight (parte integrante do projecto Mono) promete “correr aplicações *Silverlight* em Linux e providenciar todas as ferramentas necessárias para desenvolver aplicações *Silverlight* em Linux” [40]. Nos *browsers* a compatibilidade também não é completa, não existindo suporte para o Opera e Safari em Windows [41].

Desde o início o *Silverlight* utilizou alguma da experiência adquirida pelo Flash para se adaptar rapidamente às necessidades do mercado, disponibilizando de imediato várias formas distintas para desenvolver aplicações. O Microsoft Expression Studio é um conjunto de ferramentas utilizado essencialmente para desenhar as interfaces interactivas e animações e o Microsoft Visual Studio para desenvolver e depurar as aplicações. No entanto, o Microsoft




Expression Studio pode ser descartado, pois o Visual Studio também é capaz de desempenhar este papel, sendo esta solução a mais indicada para programadores avançados.

Num futuro próximo será lançada a versão 2 do Silverlight, trazendo grandes progressos a vários níveis, desde o melhor encadeamento com funcionalidades de desenvolvimento avançado, mais controlos de interface, suporte para mais tipos de vídeo e áudio, suporte para novas linguagens de programação (ex.: IronRuby – *ruby* da plataforma .NET), suporte para *Web Services*, *sockets*, entre outros [42].

2.3 Conclusões

Para toda e qualquer plataforma, as escolhas das tecnologias terá de ser feita para que haja a melhor adaptação possível ao contexto e local em que esta vai ser utilizada. Como vimos acima, todas as tecnologias tem as suas vantagens e desvantagens, restando agora analisar aquelas que melhor se adaptam aos requisitos da plataforma DSD.

Ao longo desta análise final serão utilizados três símbolos importantes:

-  - O software cumpre devidamente o requisito;
-  - O software não cumpre o requisito;
-  - O software cumpre o requisito mas não de forma tão eficaz, eficiente ou fácil como outras;

2.3.1 Sistema de Gestão de Bases de Dados (SGBD)

Na escolha do SGBD é importante frisar que se deu grande valor a um possível futuro, ou seja, é crucial termos uma ferramenta escalável e de bom desempenho, para que caso a sua utilização se estenda a mais que um departamento, não haja qualquer problema em adaptar o sistema para uma utilização mais intensa.

De forma a comparar o desempenho dos vários SGBD tomou-se como referência os testes realizados pela TPC (*Transaction Processing Performance Council*) [19], que é uma organização independente que define um conjunto padrão de transacções, medindo então o número de transacções que um SGBD em condições muito específicas (consideradas as características dos servidores) processa por unidade de tempo. Assim, foram considerados os seguintes testes:

- **Desempenho** – o Oracle tem um melhor desempenho que o SQL Server, existindo referências aos dois no top 10;
- **Relação desempenho/custos** – neste teste a TPC considerou que não seria justo uma comparação entre moedas diferentes. No dólar americano o Oracle encontra-se à frente, embora existam mais referências ao SQL Server no top 10. Nas restantes moedas presentes no teste o SQL Server domina por completo, não surgindo qualquer referência ao Oracle.

Desta forma, não se pode dizer que o Oracle tem um melhor desempenho que o SQL Server ou vice-versa, visto que ambos são amplamente utilizados nos mais diversos contextos. De notar que o MySQL não surge nestes testes, isto acontece pois a empresa realiza os seus

próprios³⁴. No entanto, como foi analisado na secção 2.2.3.3 devido às características de replicação de bases de dados do MySQL, a escalabilidade e desempenho para grandes plataformas pode vir a ser influenciada, adaptando-se melhor a bases de dados de menor dimensão.

A nível funcional o Oracle apenas apresenta um pequeno problema, a sua manutenção é um pouco complexa, exigindo conhecimentos técnicos mais profundos para tirar o melhor partido possível da ferramenta. No entanto, há um factor de peso que leva a descartar esta solução, o DETI não tem licenças de utilização e os custos são bastante elevados. Contudo, com esta análise ficou-se com a plena consciência das capacidades deste SGBD, que seria uma solução bastante válida para os requisitos apresentados.

Na Tabela 2.2 é possível ver o comportamento dos vários SGBD analisados face aos requisitos da plataforma DSD.

SISTEMA DE GESTÃO DE BASES DE DADOS					
Softwares/Requisitos	Desempenho	Escalabilidade	Segurança	Manutenção	Licença
SQL Server					
Oracle					
MySQL					

Tabela 2.2: Comportamento dos SGBD perante os requisitos da plataforma DSD.

2.3.2 Páginas Web Dinâmicas

Na escolha da ferramenta para gerir as páginas Web dinâmicas pesou essencialmente o seu desempenho, a dificuldade de desenvolvimento de aplicações, a manutenção e a integração com o SGBD já escolhido. Como podemos ver na Tabela 2.3, todas as tecnologias analisadas seriam capazes de responder às necessidades do DSD, embora existam vantagens e desvantagens em cada uma delas. Embora o ASP.net perca um pouco no seu desempenho, com uma boa máquina e com a pré-compilação da aplicação, o seu desempenho torna-se bastante competente, podendo assim tirar-se proveito das suas excelentes características de manutenção, da sua rapidez e facilidade de desenvolvimento e da sua especial integração com o *SQL Server*.

PÁGINAS WEB DINÂMICAS					
Softwares/Requisitos	Desempenho	Desenvolvimento	Manutenção	Integração SQL Server	Licença
ASP.net					
PHP					
JSP					

Tabela 2.3: Comportamento das linguagens de páginas Web dinâmicas perante os Requisitos da plataforma DSD.

³⁴ <http://www.mysql.com/why-mysql/benchmarks/>

2.3.3 Servidor Web

Nos servidores Web analisou-se o desempenho, segurança e manutenção. Como podemos ver na Tabela 2.4, verificou-se que tanto o IIS como o Apache são extremamente competentes, tendo-se frisado a necessidade de módulos extra para o apache funcionar com as mais variadas linguagens e sistemas. No entanto, a Framework .NET aqui utilizada não é suportada na sua plenitude por este servidor Web, tendo-se optado então pela utilização do IIS.

SERVIDOR WEB					
Softwares/Requisitos	Desempenho	Segurança	Manutenção	Suporte ASP.net	Licença
IIS	✓	✓	✓	✓	✓
Apache	✓	✓	✓	✗	✓

Tabela 2.4: Comportamento dos servidores Web perante os Requisitos da plataforma DSD.

2.3.4 Dinamizar Interface

Nesta categoria importa analisar o público-alvo da plataforma, visto que a interface será directamente utilizada por ele, podendo esta escolha influenciar a utilização da plataforma. Num ambiente universitário, mais do que noutros, o suporte multi-plataforma é essencial, visto a utilização dos mais variados sistemas operativos por alunos e docentes. Também associado a este factor, existe a compatibilidade com os *browsers*. Caso a plataforma não seja compatível com o sistema operativo e *browser* de eleição do utilizador, isto poderá ser suficiente para a sua não utilização.

Também de realçar a necessidade de *Plug-in*, em todo o lado existem utilizadores que pretendem ter toda a informação disponível da forma mais rápida possível, sem terem de instalar qualquer tipo de ferramenta extra nem de esperar alguns segundos para que a informação seja carregada.

Como podemos ver na Tabela 2.5, o AJAX é a tecnologia que melhor se adapta às exigências da plataforma, tendo sido a escolhida para dinamizar a interface.

DINAMIZAR INTERFACE					
Softwares/Requisitos	Utilização Multi-plataforma	Compatibilidade Browser	Não requer Plug-in	Leve	Licença
AJAX	✓	✓	✓	✓	✓
Flash	✓	✓	✗	✗	✗
Silverlight	✗	✗	✗	✗	✓

Tabela 2.5: Comportamento das ferramentas para dinamizar a interface perante os Requisitos da plataforma DSD.

2.3.5 Arquitectura Final

Com todas as escolhas das tecnologias devidamente justificadas, na Figura 2.26 podemos ver a arquitectura tecnológica final que irá suportar toda a plataforma DSD.

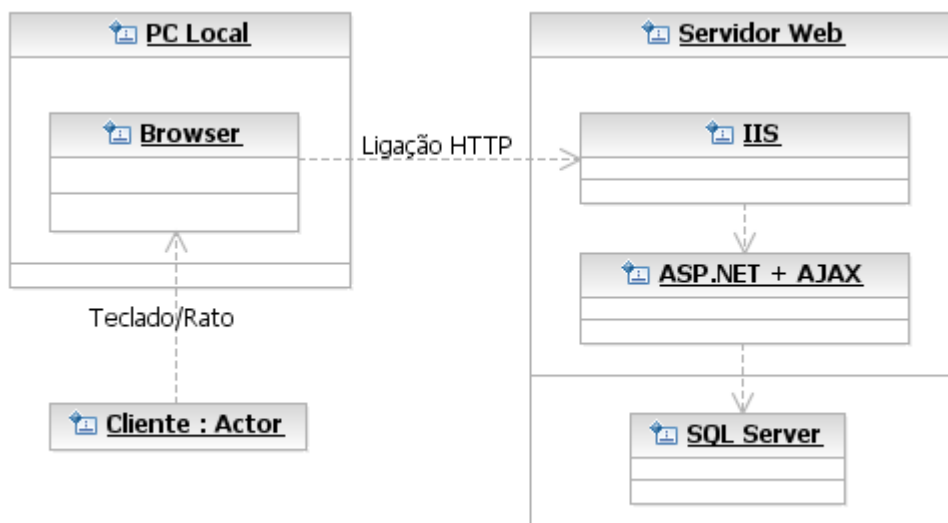


Figura 2.26: Arquitectura final da plataforma DSD.

2.4 Sumário

Este capítulo apresenta o contexto do problema que se pretende resolver, realçando as ferramentas já existentes e os problemas presentes. Após isso, é feita uma análise dos objectivos e requisitos da plataforma, para que seja possível um estudo cuidadoso das opções tecnológicas disponíveis para desempenhar os mais variados papeis que uma ferramenta deste nível exige. Neste estudo, apresentam-se as vantagens e desvantagens de cada uma das soluções, de forma a encontrar aquelas que melhor se adaptam aos requisitos e objectivos exigidos/preteridos.

3 Base Estrutural

Analisados os requisitos e os objectivos desta plataforma e encontrada a solução tecnológica que melhor se adapta ao conceito, neste capítulo pretende-se estudar de forma mais detalhada as tarefas que farão parte da DSD, para que posteriormente se possam definir em mais detalhe todas as suas funcionalidades. Finalmente, será apresentada a arquitectura base desenvolvida para suportar toda a persistência de dados que estas funcionalidades e tarefas exigem.

3.1 Visão Geral

Antes de avançar no desenvolvimento de qualquer plataforma, é essencial conhecer em mais detalhe todas as tarefas que serão suportadas, bem como os seus utilizadores e as diversas fontes de informação que fornecerão os dados que darão alma a toda a plataforma.

3.1.1 Descrição de Tarefas

Ao nível das tarefas pretende-se dar continuidade às funcionalidades disponíveis nas ferramentas já existentes (ver secção 2.1.3) e adicionar novas funcionalidades de forma lógica e funcional. Para analisar as várias tarefas, foram definidos quatro pontos fundamentais:

- **Contexto** – algumas das funcionalidades já eram de alguma forma informatizadas antes desta plataforma, interessando analisar o seu funcionamento e de que forma este pode ser melhorado;
- **Melhorias** – em tarefas já existentes, interessa demonstrar de que forma os processos serão melhorados;
- **Intervenientes** – analisar de que forma as várias pessoas intervêm em cada processo;
- **Funcionamento ou Objectivos** – analisar o funcionamento do novo processo a ser implementado na DSD ou os objectivos a atingir com a tarefa.

3.1.1.1 Escolha de Dissertações

Contexto

Como analisado na secção 2.1.3.2, a ferramenta de Mestrado Integrado auxilia a inscrição nas dissertações por parte dos alunos, tornando-se essencial analisar o seu funcionamento e de que forma este pode ser melhorado.

No antigo processo, as dissertações iam sendo adicionadas à ferramenta por parte dos docentes, disponibilizando-se aos alunos uma lista actualizada em tempo-real, para que estes

pudessem entrar em contacto com os docentes via e-mail na eventualidade de existir interesse. Caso o docente e o aluno chegassem a um acordo, teriam de assinar uma folha em papel que demonstrasse o entendimento das partes, terminando assim o processo.

Melhorias

Um dos principais problemas encontrados neste processo é a ausência de contas pessoais, sem que exista qualquer controlo sobre as dissertações que cada docente adiciona, o que não permite a sua edição directa, exigindo-se o contacto com a administração para realizar esta simples tarefa. A utilização de contas pessoais também tornará possível a criação de um sistema de demonstração de interesse nas dissertações por parte dos alunos, abolindo a necessidade de troca de *e-mails* entre alunos e docentes nesta fase.

Assim, com o novo sistema de escolha de dissertações, surgem dois termos essenciais:

- **Acordo em Papel** – documento em papel que comprova o acordo a que docente e aluno chegaram, exigindo-se a assinatura de ambos;
- **Acordo Digital** – acordo a que ambas as partes chegaram após o terem assinado de forma digital, utilizando as palavras-chave associadas às suas contas pessoais.

Um dos grandes objectivos passa por abolir a utilização de folhas em papel, incentivando a utilização de acordos digitais, no entanto, o suporte a acordos em papel tem de ser mantido, de forma a não tornar a transição algo repentina para alguns utilizadores menos receptivos às novas tecnologias. Neste contexto, o sistema teve de se adaptar aos utilizadores, sendo flexível quanto ao tipo de acordo.

Intervenientes

Os intervenientes do processo de escolha de dissertações são:

- **Docente** – adiciona dissertações e escolhe o aluno para a realizar;
- **Aluno** – mostra interesse pelas dissertações;
- **Administrador** – pode intervir de várias formas em todo o processo.

Funcionamento

Todo o processo é despoletado quando a data de início do período de inscrições nas dissertações é atingida, a partir desse momento, os docentes já podem adicionar dissertações à plataforma. Na sua inserção, para além de todos os dados que estão associados a uma dissertação, o docente também pode indicar se já existe um acordo prévio com um aluno, caso exista, este pode ser assinado de duas formas distintas, em papel ou digital. No caso do acordo em papel, uma vez que ambas as partes já o assinaram, apenas restará que o administrador confirme a sua veracidade. Já no caso de acordo digital, ambas as partes terão de assiná-lo digitalmente, através da utilização da palavra-chave da sua conta. Na situação em que o docente indica que ainda não existe qualquer acordo, será dada a oportunidade aos alunos de mostrarem interesse pela dissertação, para que posteriormente o orientador escolha o aluno que melhor se adapta aos critérios definidos, após isso, ambas as partes assinam o acordo digitalmente, finalizando este processo. Apesar de poder parecer um pouco complexo, através da análise da Figura 3.1 tudo se torna mais fluído e simples.

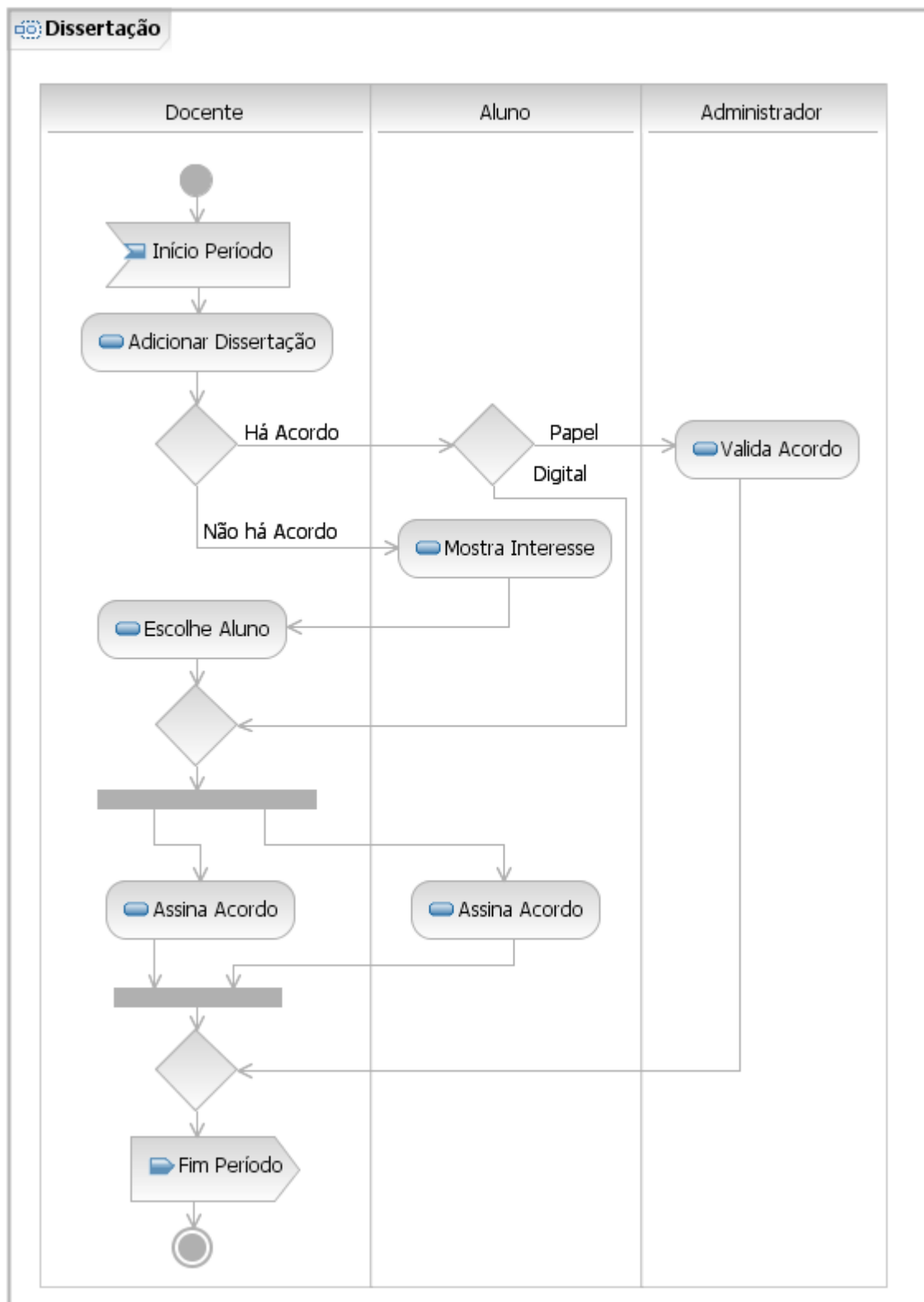


Figura 3.1: Processo de escolha de dissertações.

Ao longo deste processo há duas situações que raramente acontecem, não sendo adicionadas ao diagrama. Após o orientador adicionar a dissertação, já tendo indicado que a

assinatura seria feita através de acordo digital, ainda pode alterar esta situação, indicando que já tem um acordo em papel com um aluno, exigindo-se de seguida a confirmação deste acordo por parte da administração. Também de notar que o administrador tem poder total sobre todo o processo, podendo adicionar dissertações para um determinado docente (sendo este o orientador), alterar as dissertações e alterar os tipos de acordos, indusivamente escolher o aluno que ira realizar uma dissertação.

3.1.1.2 Escolha de Disciplinas de Opção

Contexto

Também a inscrição nas disciplinas de opção já era um processo de alguma forma suportado pela ferramenta de Mestrado Integrado (ver secção 2.1.3.2), tornando-se essencial o seu suporte nesta nova plataforma. No antigo processo, as disciplinas de opção eram disponibilizadas na ferramenta, sendo então possível a inscrição por parte dos alunos. Nesta inscrição os alunos definiam o número de disciplinas em que se queriam inscrever, indicando então a lista ordenada com as suas opções preferidas. Após isso, poderiam ver de imediato uma lista provisória de colocados, que era actualizada de forma automática a cada inscrição ou alteração de inscrição. Quando a data do final do período de inscrições era atingida, a administração gerava a lista final de colocados nas disciplinas de opção.

Melhorias

Através da utilização das contas pessoais irão surgir várias melhorias ao nível da disponibilização e gestão de informação. De realçar a informatização completa do processo de negociação de disciplinas de opção a leccionar.

Intervenientes

Os intervenientes na escolha de disciplinas de opção são:

- **Docente** – apresenta propostas de disciplinas de opção;
- **Aluno** – inscreve-se nas disciplinas de opção;
- **Administrador** – gere as disciplinas de opção que irão ser leccionadas bem como todo o processo de inscrição por parte dos alunos;

Funcionamento

A Figura 3.2 dá uma ideia geral da logística de todo o processo de inscrição nas disciplinas de opção, através da qual podemos afirmar que existem duas fases distintas:

1ª – Apresentação de Propostas para as disciplinas de opção;

2ª – Inscrição dos alunos nas disciplinas de opção.

Para a primeira fase não existe uma data que marque o seu início, apenas é exigido que aquando do início da segunda fase, as disciplinas de opção estejam todas organizadas convenientemente.

Inicialmente, a coordenação pedagógica e entidades da universidade têm de chegar a um consenso sobre quais as disciplinas de opção que poderão vir a ser leccionadas no departamento, após isso, a administração adiciona-as à plataforma, inserindo dados gerais relativamente à disciplina. Posteriormente, os docentes poderão adicionar propostas para a docência das

disciplinas, apresentando dados mais específicos da disciplina segundo o ponto de vista da equipa que a propõe.

À medida que as propostas forem surgindo, o administrador pode aceitá-las quando entender que é a melhor altura para o fazer, visto que aceitar uma proposta é o mesmo que colocar a disciplina disponível para a inscrição dos alunos. Apesar de a fase de inscrição ainda não estar disponível, a lista de opções disponíveis vai crescendo em tempo real, para que os alunos possam pensar nas disciplinas em que se querem inscrever. No entanto, o administrador pode sempre voltar atrás na escolha que fez, escolhendo outra proposta para a disciplina.

Com o aproximar do início da 2ª fase, grande parte das disciplinas de opção já deverão ter uma proposta aceite, para que o lote de escolhas seja satisfatório.

A 2ª fase terá início quando a data indicada para o início do período de opções for atingida, permitindo que os alunos se inscrevam nas disciplinas de opção. Na inscrição os alunos indicam o número de disciplinas que irão ter e a lista de preferências, depois de terminar a inscrição existirá um *feedback* imediato acerca do estado da sua inscrição, podendo-a adaptar às colocações que vão sendo disponibilizadas em tempo real, tornado o processo extremamente interactivo. Desta forma, um aluno que esteja interessado numa disciplina e tenha sido um dos primeiros a não ficar colocado, pode interagir com colegas que tenham ficado colocados para que a sua colocação se possa tornar uma realidade.

Ao longo do período de inscrições, a administração poderá fechar disciplinas de opção, isto acontece porque o número de alunos inscritos não é suficiente para preencher os lugares exigidos para a leccionar. Quando uma opção é fechada, os alunos que lá estavam colocados deixam de o estar automaticamente, analisando-se as próximas opções da inscrição de cada aluno. Este processo é reversível, não surgindo qualquer problema uma vez que todo o processo de colocações funciona de forma automatizada.

Após o período de inscrições terminar, surgirá um pequeno período para forçar colocações, tentando-se resolver as situações de alunos que não foram colocados no número de opções que pretendiam. Passado este período, será lançada a lista definitiva de colocações nas opções, tornando-as numa realidade.

Também ao longo da 2ª fase, o administrador tem poder sobre todas as tarefas, podendo alterar inscrições dos alunos e forçar colocações e não colocações de alunos.

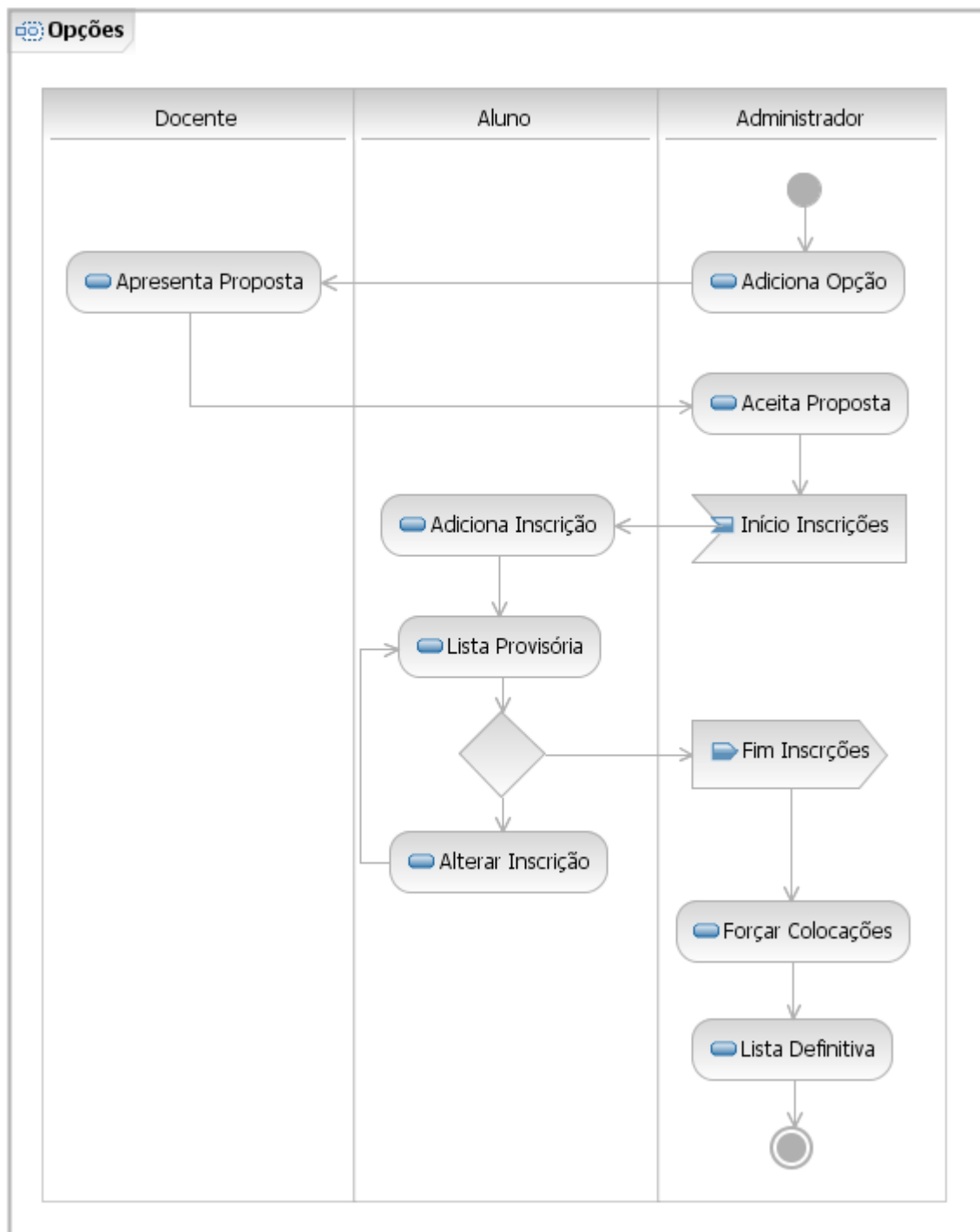


Figura 3.2: Processo de escolha de disciplinas de opção.

3.1.1.3 Escolha de Turmas de Disciplinas de Opção

Contexto

Na inscrição nas disciplinas de opção poderá surgir a necessidade de inscrição nas turmas da opção em que se ficou colocado, que só acontecerá em disciplinas que exigirão mais do que

uma turma para suportar todos os alunos que foram colocados. Nas restantes disciplinas os alunos são automaticamente colocados na turma existente.

Esta funcionalidade foi projectada a pensar no futuro, visto que no DETI todas as disciplinas de opção apenas exigem uma turma para que esta seja leccionada, contudo, o mesmo não acontece em outros departamentos. Desta forma, não se justificava pelos menos para já, a implementação das interacções que permitissem o pleno funcionamento desta tarefa. Assim, o suporte existirá apenas ao nível da estrutura base que sustentará toda a persistência de dados exigida.

Intervenientes

Os intervenientes da escolha de turmas são:

- **Aluno** – inscreve-se nas turmas das disciplinas de opção;
- **Administrador** – gestão de todos os dados associados às turmas bem como a resolução de problemas associados à colocação de alunos em turmas.

Funcionamento

Ainda que a sua implementação não exista, o seu funcionamento foi pensado de forma detalhada.

Antes do início do período de inscrições nas turmas, a administração tem que disponibilizar as turmas existentes para cada disciplina de opção. Através da análise da Figura 3.3, verificamos que o processo de escolha de turmas é muito semelhante com a 2ª fase da escolha das disciplinas de opção, não sendo necessário repetir a sua explicação. No entanto o conceito de inscrição muda. Apesar de continuar a ser uma lista ordenada de itens, cada item corresponderá às escolhas das turmas para cada opção, como é possível ver na Tabela 3.1. Assim, caso um aluno não consiga ficar colocado na turma de uma opção, a escolha será considerada não colocada, passando para a próxima escolha que o aluno submeteu.

Escolha / Opção	Opção 1	Opção 2	Opção 3
1ª	Turma 1	Turma 1	Turma 1
2ª	Turma 2	Turma 1	Turma 1
3ª	Turma 1	Turma 2	Turma 1

Tabela 3.1: Formato da inscrição nas turmas das disciplinas de opção.

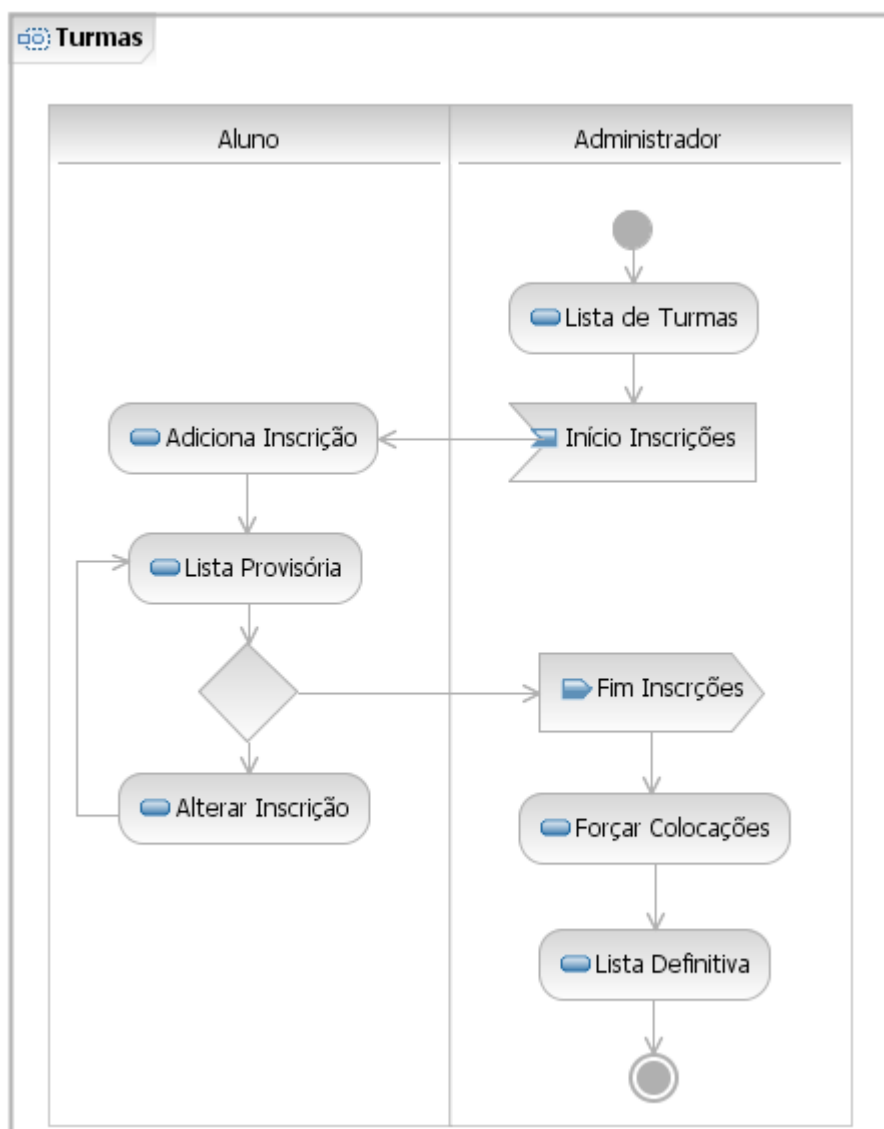


Figura 3.3: Processo de escolha de turmas das disciplinas de opção.

3.1.1.4 Lista de Desejos dos Docentes (*Wish List*)

Contexto

Durante a construção de horários, seja de forma manual ou automatizada, há sempre um requisito que deve ser respeitado ao máximo, os desejos e requisitos dos docentes. Obter esta informação é um pouco complexo, visto que sem qualquer ferramenta apenas um contacto directo da coordenação pedagógica com os docentes poderá resolver esta questão. No entanto, quando o número de docentes excede a centena, esta torna-se uma tarefa praticamente impossível. Assim, criou-se um novo sistema para recolha das preferências dos docentes relativamente às disciplinas que deseja ou não leccionar e às indisponibilidades de horário.

A esta nova funcionalidade integrante da DSD deu-se o nome de *Wish List* (Lista de Desejos).

Intervenientes

Os actores intervenientes são:

- **Docente** – submissão das suas preferências de disciplinas e horários;
- **Administrador** – visualização e análise das preferências dos docentes.

Objectivo

O principal objectivo desta nova ferramenta passa pela recolha das seguintes informações:

- Disciplinas que o docente deseja leccionar;
- Disciplinas que o docente não deseja leccionar;
- Dentro das disciplinas que o docente deseja leccionar, as suas preferidas de forma ordenada;
- Dentro das preferidas, os tipos de aulas que desejam leccionar (Teórica, Prática, Teórico-prática e Orientação Tutorial);
- Limitações ao nível de horário.

Com estas informações, será possível pelo menos tentar satisfazer todos os desejos dos docentes, desrespeitando-os apenas em última instância.

3.1.1.5 Horários

Contexto

Dada a existência de uma ferramenta para construção de horários, o suporte deste conceito é essencial. Como analisado na secção 2.1.3.1, esta ferramenta já permitiu grandes progressos, no entanto ainda tem alguns problemas que podem vir a ser corrigidos.

Intervenientes

Para a construção e visualização de horários apenas existe um interveniente, o administrador, responsável pela distribuição de todo o serviço docente.

Objectivo

A construção de horários implica uma grande complexidade, logo, para já o objectivo é suportar todas as informações que lhe estão associadas, permitindo a importação dos horários construídos na ferramenta que já existe. A visualização dos horários importados também será permitida, disponibilizando várias vistas para a mesma informação.

De notar que também nesta fase se pensou no futuro, abrindo as portas à implementação de uma ferramenta que permita a construção automática de horários.

3.1.1.6 Gestão de Entidades e Conceitos

Contexto

Como consequência do suporte das tarefas acima referidas, existe uma panóplia de entidades e conceitos associados, surgindo a necessidade de suportar toda a sua gestão, por exemplo, relacionado com dissertações há cursos, docentes, alunos, áreas, subáreas e períodos. Todos estes conceitos e entidades terão de ser geridos, suportando a sua adição, edição, remoção, entre outras funcionalidades específicas de cada.

Esta tarefa ocorre como consequência da independência e da relação entre conceitos que foi introduzida na plataforma, permitindo a manutenção dos dados de forma independente mas estruturada e organizada, tornando uma alteração visível para toda a plataforma. Como exemplo, um docente está relacionado com várias tarefas (dissertações, disciplinas de opção, turmas, entre outras), caso um qualquer dado em específico do docente seja alterado, esta nova informação ficará de imediato disponível para toda a plataforma, independentemente do ponto a que se acede aos dados do docente.

Intervenientes

Os administradores são os únicos intervenientes na gestão de todas as entidades e conceitos, cabendo-lhe a eles estabelecer a relação entre a vida real e os dados da plataforma.

Objectivo

O principal objectivo desta tarefa é facilitar a gestão de todos os dados da plataforma através de interfaces gráficas simples e de funcionalidades que agilizem o processo de importação e exportação de grandes volumes de dados.

Em toda a manutenção há um ponto de grande relevância, guardar historial de todas as tarefas realizadas. Portanto, é fundamental fazer uma boa gestão dos vários anos e semestres que vão passando, mantendo um historial de todas as informações relativas a docentes, disciplinas e alunos.

Na plataforma DSD os principais conceitos e entidades nunca são de facto removidos, são apenas tornados inactivos, precavendo situações de perda de dados. Assim, é preciso todo um sistema para fazer a gestão dos itens inactivos, permitindo a sua reactivação quando requerido.

3.1.2 Entidades e Actores

Analisadas as várias tarefas que serão suportadas nesta plataforma, importa resumir as entidades envolvidas na DSD:

- **Docentes** – elementos participativos na submissão de dados;
- **Alunos** – elementos participativos na inscrição em eventos;
- **Coordenação Pedagógica** – entidade responsável por toda a organização da distribuição do serviço docente do departamento;
- **Secretaria** – entidade que auxilia a coordenação pedagógica na execução e organização de algumas tarefas.

Visto que tanto a coordenação pedagógica como a secretaria tem um papel de administração do sistema, a melhor solução passa pela definição de vários administradores com níveis de acesso distintos, que poderão ser criados e geridos em tempo real. Também de realçar a existência do factor tempo, que intervém em todos os processos inicializando e finalizando os períodos de inscrições, terminando as sessões por *time out*, entre outras intervenções.

Assim, podemos concluir que os actores da plataforma DSD são: Docente, Aluno, Administrador e Tempo.

3.1.3 Fontes de Informação

Uma plataforma deste nível exige a gestão de uma grande quantidade de dados, o que torna impraticável a sua introdução manual. Assim, pretendeu-se minimizar ao máximo a introdução manual de dados, contribuindo para isso a existência de ferramentas informáticas que já tinham informação necessária em formato digital.

3.1.3.1 Ferramenta de construção de horários

Esta foi a principal fonte de informação da DSD, a partir da qual se obteve toda a informação associada a docentes, disciplinas, salas e cursos.

Como a informação estava contida num simples ficheiro de texto, tiveram de ser desenvolvidas funcionalidades para fazer a análise sintáctica dos dados e a consequente importação dos conceitos para a base de dados.

3.1.3.2 PACO (Portal Académico Online)

Com os dados obtidos através da ferramenta de construção de horários, apenas resta adquirir a informação relativa aos alunos, sendo o PACO a fonte de informação mais confiável para este objectivo, onde se encontram os rankings oficiais dos alunos na universidade.

Infelizmente não foi possível a integração informática com esta ferramenta por razões de tempo e de definição oficial de uma estratégia, e por tal recorreu-se mais uma vez à construção de ferramentas de análise sintáctica (desta vez a ficheiros Excel).

3.2 Funcionalidades

Após a análise de todas as tarefas a realizar na DSD e de todos os seus actores, é altura de entrar em maior detalhe na descrição das funcionalidades que efectivamente farão parte desta plataforma.

Fazendo jus à velha técnica “Dividir para Conquistar”³⁵, optou-se por repartir a plataforma DSD em dois blocos lógicos, permitindo uma melhor percepção, organização e até funcionamento:

- **Front Office** – parte da plataforma que trata da interacção directa com os utilizadores (alunos e docentes);
- **Back Office** – parte da plataforma responsável pela sua própria gestão e manutenção, sendo maioritariamente utilizada pelos administradores.

Para além desta divisão global, dentro de cada bloco (“Office”) também foram criados Pacotes de Gestão, permitindo dividir cada bloco lógico em pequenos pacotes mais fáceis de identificar e trabalhar. Tendo esta estrutura por base, optou-se por apresentar as funcionalidades de forma repartida, enquadrando-as em cada bloco funcional e pacote de gestão.

³⁵ por Karatsuba Anatolii Alexeevich, permite dividir um problema maior em problemas menores, até que este possa ser resolvido directamente.

3.2.1 Front Office

No *Front Office* encontramos os seguintes pacotes de gestão:

- **Sessão** – gestão da sessão do utilizador;
- **Conta** – gestão da conta do utilizador;
- **Dissertação** – gestão dos dados das dissertações associados a cada tipo de utilizador;
- **Opção** – gestão dos dados das disciplinas de opção associados a cada tipo de utilizador;
- **Wish List** – gestão dos dados das listas de desejos de cada um dos docentes.

Na Tabela 3.2 encontra-se uma breve descrição de todas as funcionalidades de cada pacote pertencente a este bloco lógico.

Pacote	Funcionalidade	Descrição
Sessão	Iniciar Sessão	Autenticar-se na plataforma, tendo acesso às funcionalidades que lhe dizem respeito.
	Terminar Sessão	Sair da plataforma.
	Terminar Sessão por time out	Passados 30 minutos de o utilizador ter iniciado a sessão, esta será automaticamente terminada, saindo da plataforma.
Conta	Definir Palavra-chave	Registrar-se no sistema.
	Activar Conta	Activar a conta. Só para alunos com registo prévio.
	Novo código de activação	Pedir um novo código para activar a conta.
	Recuperar Palavra-chave	Pedir uma nova palavra-chave.
	Editar Conta	Editar os dados relativos à conta pessoal.
Dissertação	Pesquisar Dissertação	Pesquisar por uma dissertação, tendo ao seu dispor dois tipos de pesquisa: básica e avançada.
	Adicionar Dissertação	Adicionar uma dissertação.
	Listar Dissertações Adicionadas	Ver todas as dissertações que adicionou.
	Ver Dissertação	Ver todos os dados relativos a uma dissertação.
	Editar Dissertação	Editar os dados de uma dissertação adicionada pelo próprio utilizador.
	Listar Interessados	Ver os alunos interessados numa dissertação adicionada pelo próprio utilizador.
	Ver Aluno	Ver os dados relativos a um aluno que mostrou interesse por uma das dissertações que o utilizador adicionou.
	Escolher Aluno para Acordo	Escolher um aluno para assinar acordo numa dissertação.
	Remover aluno	Remover a escolha do aluno previamente feita.

	escolhido para Acordo	
	Listar Acordos	Listar os acordos de dissertação em que está envolvido.
	Ver Acordo	Ver todos os dados associados a um acordo.
	Assinar Acordo	Assinar o acordo de uma dissertação.
	Remover Assinatura	Remover a assinatura de um acordo.
	Mostrar Interesse	Mostrar interesse por uma dissertação.
	Listar Dissertações Interessado	Listar todas as dissertações em que mostrou interesse.
	Ver Interesse	Ver todos os dados associados a um interesse mostrado por uma Dissertação.
	Remover Interesse	Remover o interesse mostrado por uma Dissertação.
	Editar Interesse	Editar o interesse mostrado por uma Dissertação.
	Opção	Pesquisar Opção
Listar Colocações		Ver a lista de colocações nas opções (provisória e definitiva).
Adicionar Proposta		Adicionar uma proposta para leccionar uma opção.
Listar Propostas Adicionadas		Ver a lista de propostas de opções que adicionou.
Editar Proposta		Editar uma proposta adicionada.
Adicionar Inscrição		Adicionar inscrição nas Opções.
Ver Inscrição		Ver a inscrição previamente adicionada.
Editar Inscrição		Editar a inscrição previamente adicionada.
Wish List	Adicionar <i>Wish List</i>	Adicionar uma <i>Wish List</i> . Só para docentes.
	Ver <i>Wish List</i>	Ver os dados da <i>Wish List</i> previamente adicionada.
	Editar <i>Wish List</i>	Editar os dados da <i>Wish List</i> previamente adicionada.

Tabela 3.2: Funcionalidades do *Front Office*.

Como neste bloco lógico existem vários actores, para uma melhor percepção dos papéis desempenhados por cada um dos actores em cada pacote, a apresentação de uma descrição ilustrada é a melhor solução, permitindo dar uma ideia de como funcionará a interacção entre as várias funcionalidades.

Relativamente ao pacote Sessão (Figura 3.4), os actores docente e aluno tem acesso às mesmas funcionalidades, podendo iniciar e terminar a sessão. Também o actor Tempo tem aqui o seu papel, terminando a sessão do utilizador activo após a passagem de um período de tempo definido.

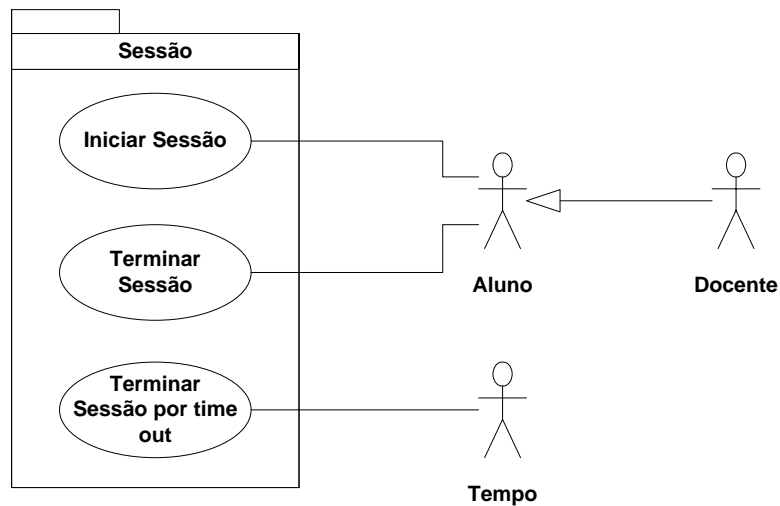


Figura 3.4: Funcionalidades a que os actores tem acesso no pacote Sessão.

No pacote Conta (Figura 3.5), dado que o docente já tem a sua palavra-chave pré-definida, este não precisa de a criar, podendo apenas editar os dados da sua conta e recuperar a palavra-chave em caso de esquecimento. Já o aluno tem acesso a todas a funcionalidades deste pacote, necessitando de efectuar o registo e conseqüente activação da conta.

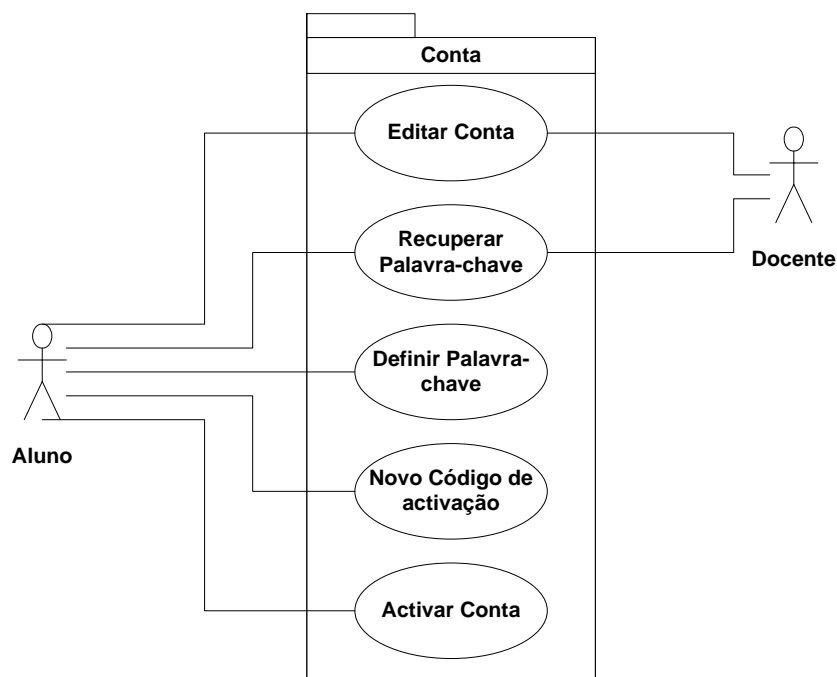


Figura 3.5: Funcionalidades a que os actores tem acesso no pacote Conta.

Para facilitar a percepção da interacção dos actores docente e aluno com o pacote dissertação, optou-se pela divisão em dois diagramas.

Na Figura 3.6 podemos ver a interacção do actor docente com este pacote, tendo acesso a toda a gestão das dissertações por ele adicionadas, onde se inclui a visualização dos alunos interessados em cada uma delas e a escolha de alunos para a realização das dissertações.

Finalmente, a gestão de acordos de dissertação também são funcionalidades acessíveis ao docente.

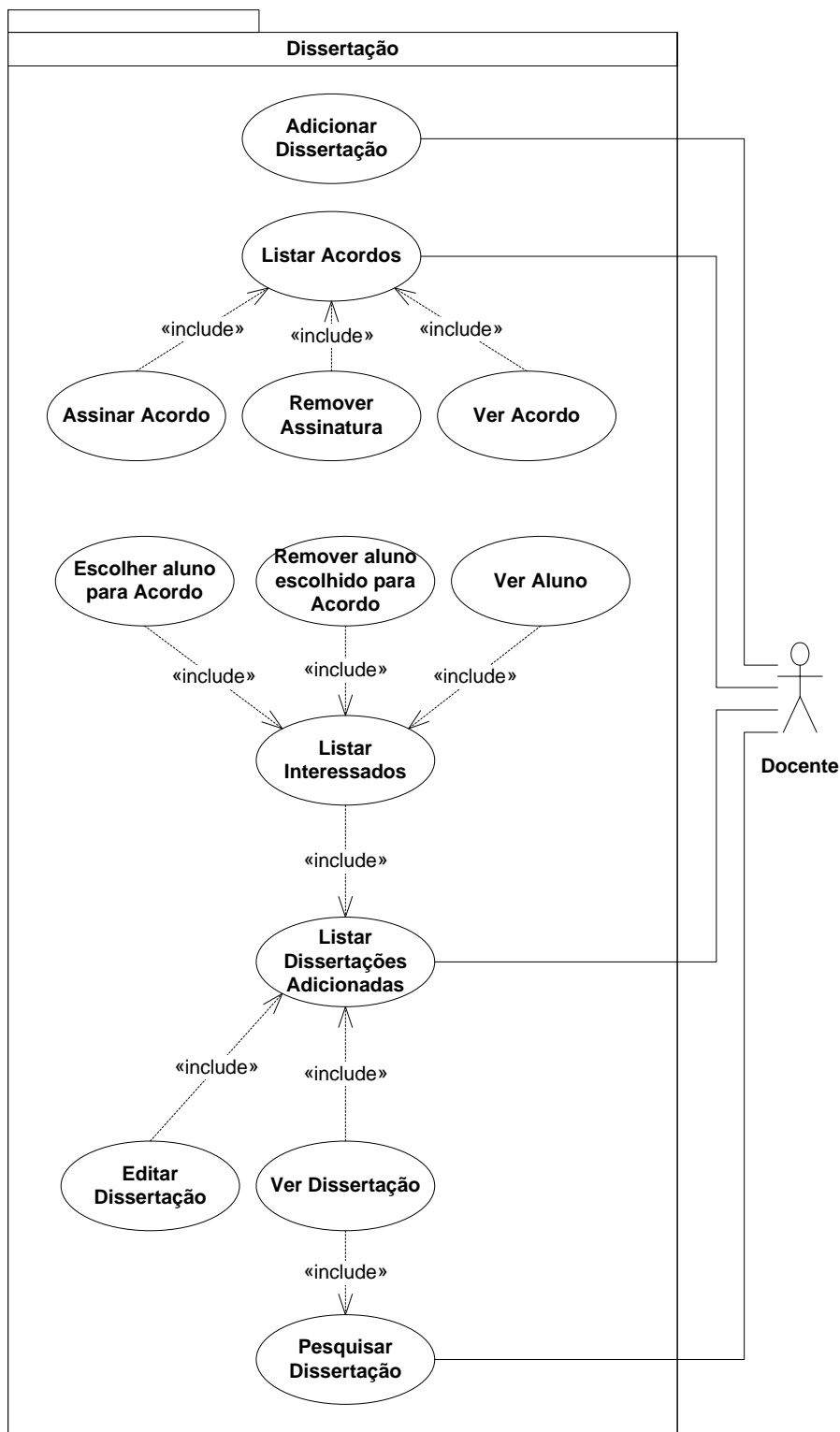


Figura 3.6: Funcionalidades a que o actor docente tem acesso no pacote Dissertação.

De realçar a utilização do conceito «include», que de forma simplista, indica a inclusão de uma funcionalidade numa outra. Por exemplo, quando o docente acede à lista de acordos, poderá assinar um dos acordos presentes na lista.

A interacção do aluno com o pacote Dissertação pode ser vista na Figura 3.7, sendo-lhe possível fazer a gestão dos acordos e também dos interesses demonstrados nas dissertações.

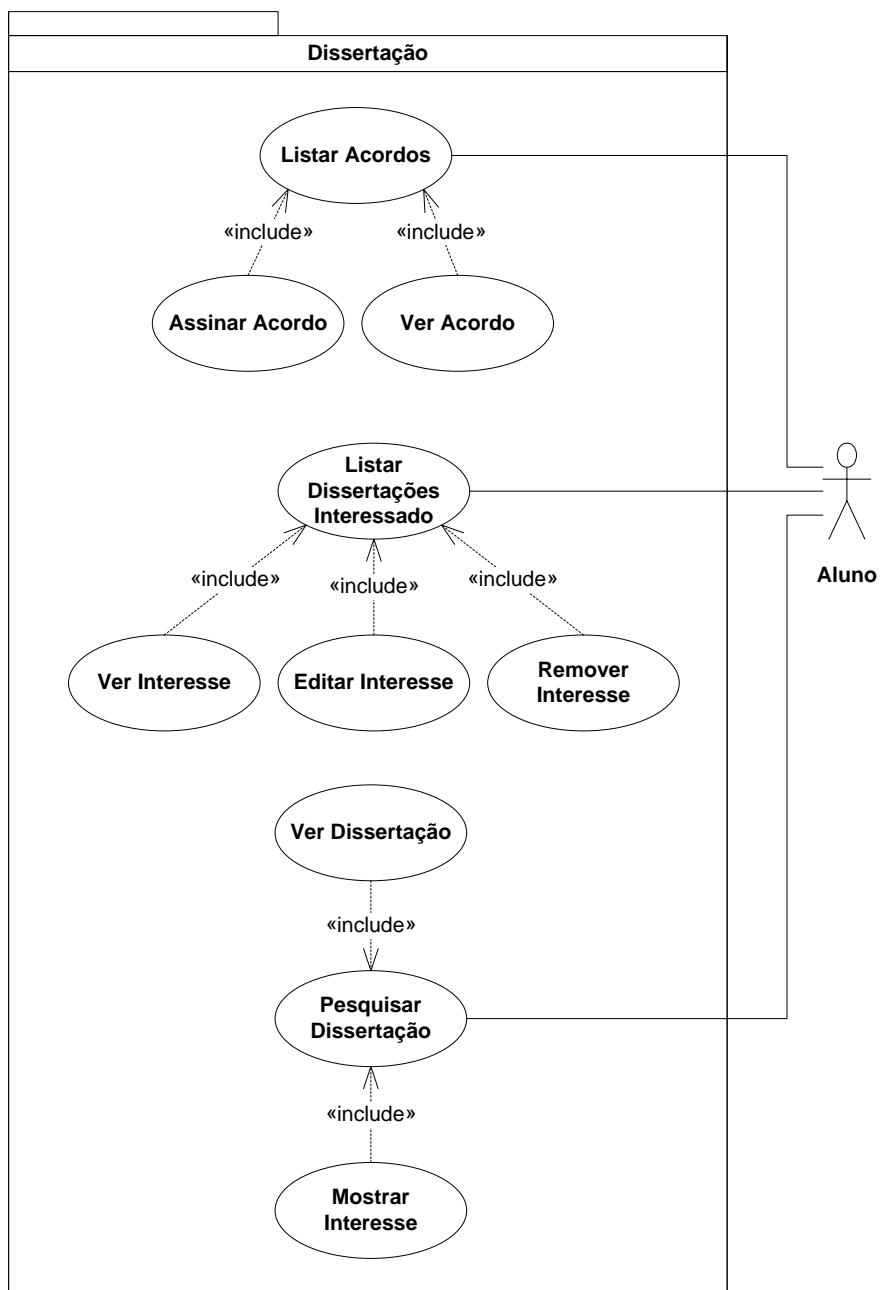


Figura 3.7: Funcionalidades a que o actor aluno tem acesso no pacote Dissertação.

Relativamente ao pacote Opção, o docente tem acesso a todas as funcionalidades relativas à apresentação de propostas para leccionar as disciplinas de opção, podendo também ver as disciplinas de opção existentes e as respectivas colocações. Já o aluno, tem acesso às

funcionalidades de inscrição nas disciplinas de opção, tendo também acesso à lista de opções existentes para fazer um estudo prévio da inscrição que irá submeter.

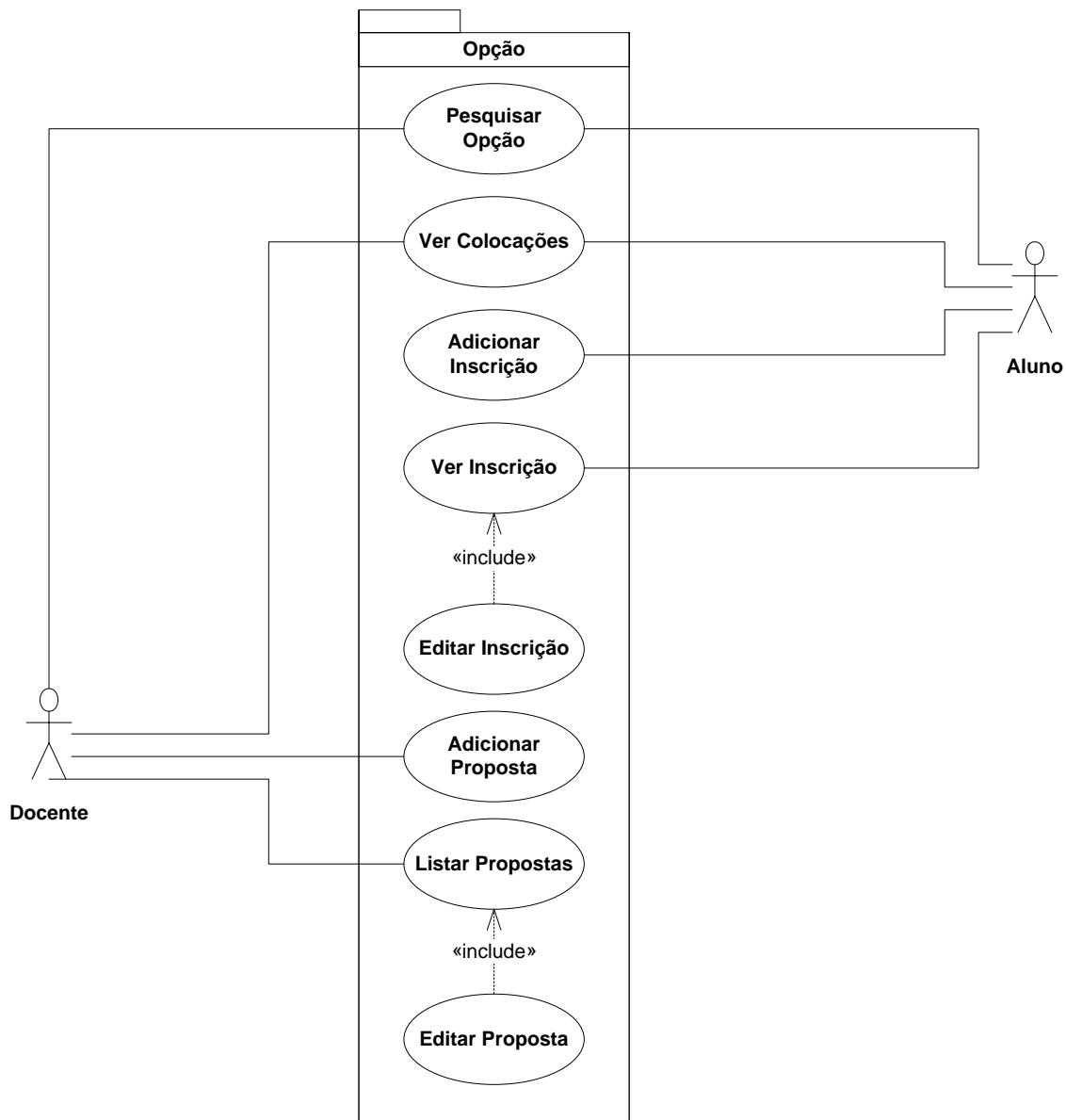


Figura 3.8: Funcionalidades a que os actores tem acesso no pacote Opção.

Finalmente, apenas o docente interage com o pacote *Wish List* (Figura 3.9), podendo adicionar, ver e alterar a sua lista de desejos.

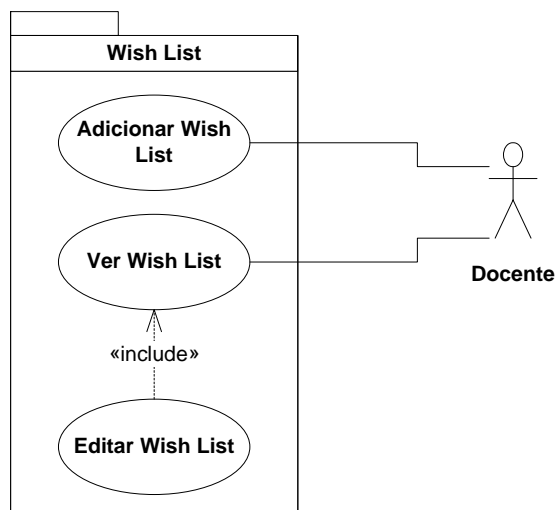


Figura 3.9: Funcionalidades a que os actores tem acesso no pacote *Wish List*.

3.2.2 Back Office

Um local onde se reflecte a alta complexidade desta plataforma é exactamente no *Back Office*, onde são necessárias muitas funcionalidades para suportar a manutenção de todas as entidades e conceitos.

Assim, os pacotes que fazem parte deste bloco são:

- **Sessão** – controlar a sessão do utilizador;
- **Departamentos** – gestão dos departamentos existentes na plataforma;
- **Cursos** – gestão dos cursos;
- **Disciplinas** – gestão das disciplinas, não incluindo turmas e aulas;
- **Opções** – gestão das disciplinas de opção;
- **Docentes** – gestão dos docentes de cada departamento;
- **Alunos** – gestão dos alunos de cada curso;
- **Cargos** – gestão dos cargos dos docentes;
- **Categorias** – gestão das categorias dos docentes;
- **Salas** – gestão das salas de cada departamento;
- **Dissertações** – gestão de todos os dados associados às dissertações;
- **Períodos** – gestão dos períodos de inscrições;
- **Acessos** – gestão dos tipos de acessos ao *Back Office*;
- **Administradores** – gestão dos utilizadores com acesso privilegiado ao *Back Office*;
- **Semestres** – análise dos vários semestre existentes na plataforma;
- **Dados Externos** – importação e exportação de dados em massa;
- **Wish List** – análise das listas de desejos adicionadas pelos docentes;
- **Horários** – análise dos horários existentes na plataforma;
- **Configuração** – gestão de configurações gerais da plataforma;
- **Inactivos** – gestão dos objectos inactivos que existem na plataforma;

Na Tabela 3.3 encontra-se uma breve descrição de todas as funcionalidades de cada pacote que faz parte do *Back Office*.

Pacote	Funcionalidade	Descrição
Sessão	Iniciar Sessão	Autenticar-se na plataforma, tendo acesso às funcionalidades que lhe dizem respeito.
	Terminar Sessão	Sair da plataforma.
	Terminar Sessão por time-out	Passados 30 minutos de o utilizador ter iniciado a sessão, esta será automaticamente terminada, saindo da plataforma.
Departamentos	Adicionar	Adicionar um departamento.
	Pesquisar	Explorar os departamentos que já existem.
	Editar	Editar os dados de um departamento.
	Desactivar	Desactivar um departamento.
Cursos	Adicionar	Adicionar um curso.
	Pesquisar	Explorar os cursos que já existem.
	Editar	Editar os dados de um curso.
	Desactivar	Desactivar um curso.
	Visível/Invisível	Tornar um curso visível ou invisível ao <i>Front Office</i> .
Disciplinas	Adicionar	Adicionar uma disciplina.
	Pesquisar	Explorar as disciplinas que já existem.
	Editar	Editar os dados de uma disciplina.
	Desactivar	Desactivar uma disciplina.
Opções	Adicionar	Adicionar uma opção.
	Pesquisar	Explorar as opções que já existem.
	Editar	Editar os dados de uma opção.
	Ver Propostas	Ver a lista de todas as propostas que foram feitas para uma opção.
	Adicionar Proposta	Adicionar uma proposta para uma opção.
	Editar Proposta	Editar os dados de uma proposta.
	Escolher Proposta	Escolher uma das propostas.
	Remover Proposta Escolhida	Remover a escolha feita anteriormente.
	Ver Colocações	Ver a lista de colocações, sendo possível a filtragem de dados.
	Ver/Adicionar/Alterar Inscrições	Ver, alterar ou adicionar uma inscrição de um aluno nas opções.
Docentes	Desactivar	Desactivar uma opção.
	Adicionar	Adicionar um docente.
	Pesquisar	Explorar os docentes que já existem.
	Editar	Editar os dados de um docente.

	Ver	Ver todos os dados relacionados com um docente (dados pessoais, dados profissionais, opções e dissertações)
	Desactivar	Desactivar um docente.
Alunos	Adicionar	Adicionar um aluno.
	Pesquisar	Explorar os alunos que já existem.
	Editar	Editar os dados de um aluno.
	Ver	Ver todos os dados relacionados com um aluno (dados pessoais, dados escolares, opções e dissertações)
	Desactivar	Desactivar um aluno.
Cargos	Adicionar	Adicionar um cargo.
	Pesquisar	Explorar os cargos que já existem.
	Editar	Editar os dados de um cargo.
	Desactivar	Desactivar um cargo.
Categorias	Adicionar	Adicionar uma categoria.
	Pesquisar	Explorar as categorias que já existem.
	Editar	Editar os dados de uma categoria.
	Desactivar	Desactivar uma categoria.
Salas	Adicionar	Adicionar uma sala.
	Pesquisar	Explorar as salas que já existem.
	Editar	Editar os dados de uma sala.
	Desactivar	Desactivar uma sala.
Dissertações	Adicionar	Adicionar uma dissertação.
	Pesquisar	Explorar as dissertações que já existem.
	Editar	Editar os dados de uma dissertação.
	Ver	Ver todos os dados relativos a uma dissertação.
	Desactivar	Desactivar uma dissertação.
Períodos	Adicionar	Adicionar um período de inscrições.
	Pesquisar	Explorar os períodos de inscrições que já existem.
	Editar	Editar os dados de um período.
Acessos	Adicionar	Adicionar um tipo de acesso.
	Pesquisar	Explorar os tipos de acesso que já existem.
	Editar	Editar os dados de um tipo de acesso.
	Ver	Ver todos os dados relativos a um acesso.
	Desactivar	Desactivar um acesso.
Administradores	Adicionar	Adicionar um novo administrador.
	Pesquisar	Explorar os administradores que já existem.
	Editar	Editar os dados de um administrador.

	Desactivar	Desactivar um administrador.
Semestres	Adicionar	Adicionar um semestre.
	Pesquisar	Ver os semestres que existem.
Dados Externos	Importar Dados	Importar dados para a base de dados a partir do ficheiro de texto proveniente da ferramenta de construção de horários.
	Importar Alunos	Importar alunos para a base de dados a partir do ficheiro Excel fornecido pelo PACO.
	Importar Inscrições	Importar as inscrições nas opções realizadas no PACO, a partir de um ficheiro Excel.
	Exportar Colocações	Exportar as colocações nas opções para um ficheiro Excel.
Wish List	Ver	Ver as listas de desejos dos docentes num ficheiro Excel, com vários modos de visualização.
Horários	Ver	Ver os horários num ficheiro Excel, com vários modos de visualização.
Configuração	Ver/Editar	Ver/Editar as várias configurações da plataforma.
Inactivos	Recuperar	Recuperar objectos inactivos, sendo possível recuperar Departamentos, Cursos, Salas, Docentes, Alunos, Cargos, Categorias, Disciplinas, Opções, Dissertações, Acessos e Administradores.

Tabela 3.3: Funcionalidades do *Back Office*.

Como o único actor que interage com este bloco lógico é o Administrador, não há necessidade de uma descrição ilustrada das interacções dos actores com os vários pacotes.

3.3 Modelo do Domínio

Analizadas todas as tarefas a realizar na plataforma e todas as funcionalidades necessárias para responder às suas necessidades, importa definir um modelo do domínio cuidado capaz de suportar todos os requisitos especificados.

O modelo de domínio de uma aplicação é onde são retratados os vários conceitos e as relações entre si, obtendo-se uma perspectiva estrutural de toda a ferramenta a ser desenvolvida [43]. Tipicamente o modelo do domínio pode ser representado através da utilização de diagramas de classes [44], onde cada classe representa uma tabela na base de dados onde os dados serão armazenados de forma persistente.

Com a complexidade na conjugação de todas as tarefas numa só plataforma, chegar à solução final aqui apresentada foi um processo bastante moroso, tentando-se respeitar todas as regras de boa modelação e tendo sempre em vista a ligação coesa de todos os conceitos. Neste contexto, o modelo de domínio será apresentado de forma faseada, facilitando a percepção de todos os conceitos e as ligações entre si.

No Anexo A encontra-se a descrição detalhada de cada atributo das várias classes pertencentes ao modelo do domínio.

3.3.1 Universidade

Para que exista um suporte a vários departamentos, há a necessidade de uma classe que represente cada um deles. Num departamento existem sempre alunos de vários cursos, tendo de se guardar os dados de cada curso existente em cada departamento. Como as aulas tem de ser realizadas em espaços físicos do departamento, interessa também ter uma classe que represente as várias salas existentes. Na Figura 3.10 podemos ver o diagrama de relação destes conceitos.

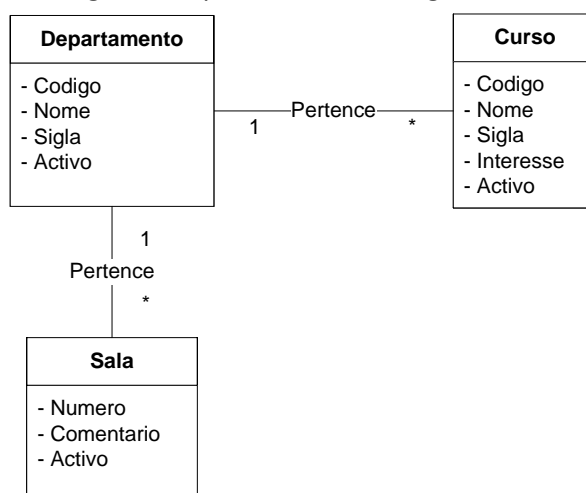


Figura 3.10: Diagrama do domínio para suportar as entidades Departamento, Curso e Sala.

3.3.2 Docentes

Visto que os docentes e os alunos têm vários dados em comum, a melhor solução passa pela utilização de uma classe Pessoa, de forma a evitar a replicação de dados. Quando da inserção poderão não existir dados como *e-mail*, telefone e conta de acesso, pelo que terão de ser utilizadas classes independentes para evitar a ocorrência de dados nulos, que muitos problemas causam na gestão e utilização de bases de dados (ex.: a ocorrência de valores nulos em duas colunas corresponde a ter valores duplicados; conflitos em operações escalares). Também a nacionalidade dos utilizadores é factor de interesse na obtenção de dados estatísticos. Na Figura 3.11 podemos ver os conceitos relacionados com a entidade pessoa.

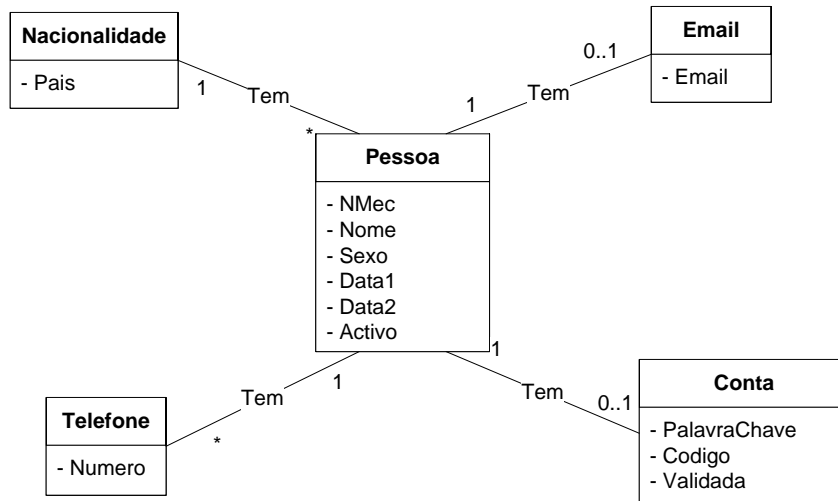


Figura 3.11: Diagrama do domínio para suportar a entidade Pessoa e todos os seus dados associados.

Assim, como podemos ver na Figura 3.12, um Docente será uma Pessoa, que irá pertencer a um determinado Departamento. Nesta fase é de realçar que existirão dados que apenas farão respeito a um semestre específico, assim surge o conceito de DocenteAno, que terá os dados específicos do docente para aquele semestre.

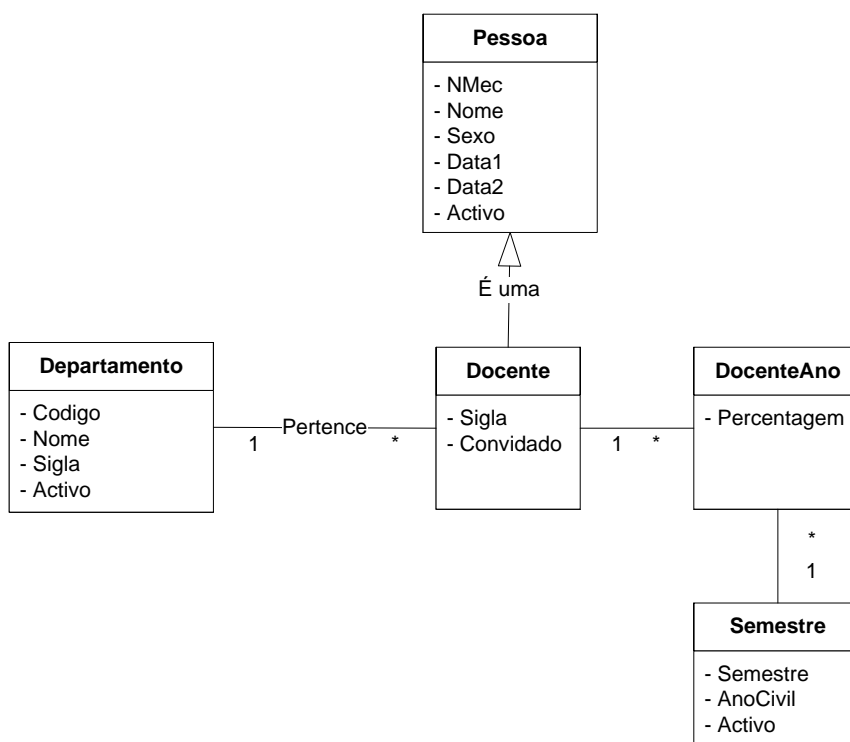


Figura 3.12: Diagrama do domínio para suportar a entidade Docente.

Todo e qualquer docente terá uma categoria associada, exemplos de categorias são “Professor Catedrático” e “Professor Auxiliar”. Através da classe Docente_Categoria é possível guardar os períodos em que um docente fez parte de uma determinada categoria, como é possível ver na Figura 3.13.

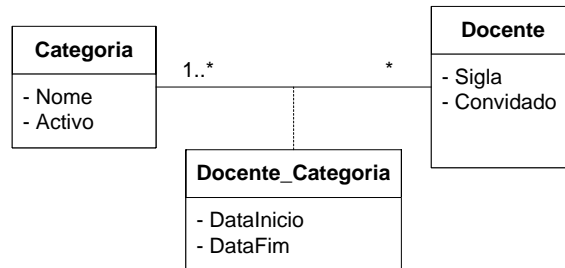


Figura 3.13: Diagrama do domínio para suportar a entidade Categoria dos docentes.

Nas instituições universitárias os docentes podem desempenhar cargos de responsabilidade na organização de departamentos ou cursos. Assim, foram definidos dois tipos de cargos: associados a departamentos e associados a cursos. Exemplo de um cargo associado a um departamento é o de Coordenador Pedagógico, operando sobre a organização das tarefas do departamento de que faz parte. Director de Curso é um exemplo de cargo associado a cursos, influenciando apenas tarefas realizadas num curso em específico.

Esta metodologia foi utilizada para minimizar ao máximo a replicação de dados, assim, um cargo só existirá uma vez, relacionando-se a ele todos os cursos ou departamentos que lhe estão associados. Como nas categorias, também é possível guardar os períodos em que um determinado docente foi responsável por um cargo.

Com a análise da Figura 3.14, é possível ter uma melhor percepção desta abstracção.

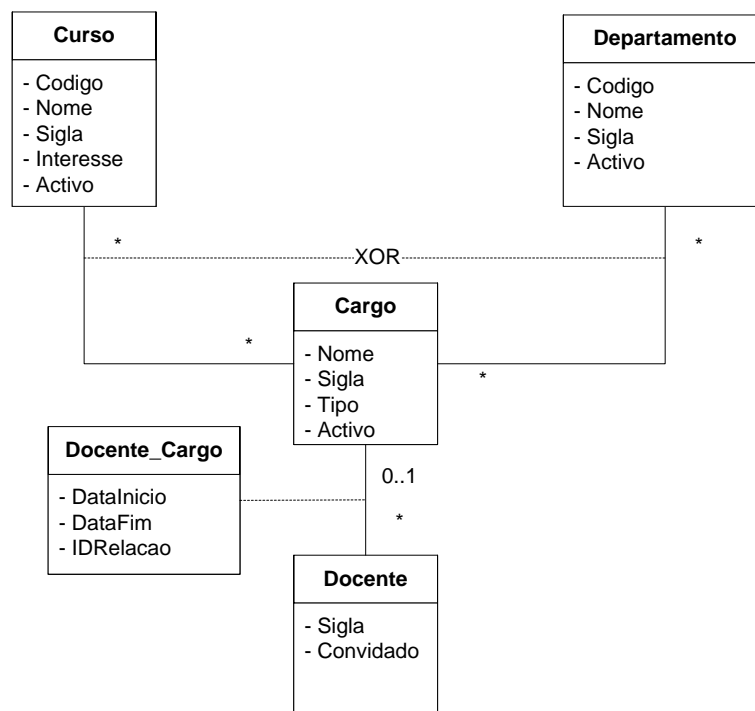


Figura 3.14: Diagrama do domínio para suportar os Cargos dos docentes.

3.3.3 Disciplinas

Todas as disciplinas se encontram relacionadas com subáreas e por sua vez com áreas. Exemplo de uma subárea é Sistemas de Informação que faz parte da área de Informática. Uma

disciplina também está associada a vários cursos, sendo importante o estabelecimento destas relações para a posterior filtragem de dados que serão disponibilizados aos alunos.

Por fim, também as disciplinas têm dados únicos de cada semestre. Exemplo disso são os docentes que estão relacionados com cada disciplina em cada semestre, factor de grande importância na construção de horários. Na Figura 3.15 podemos ver a base do conceito disciplina.

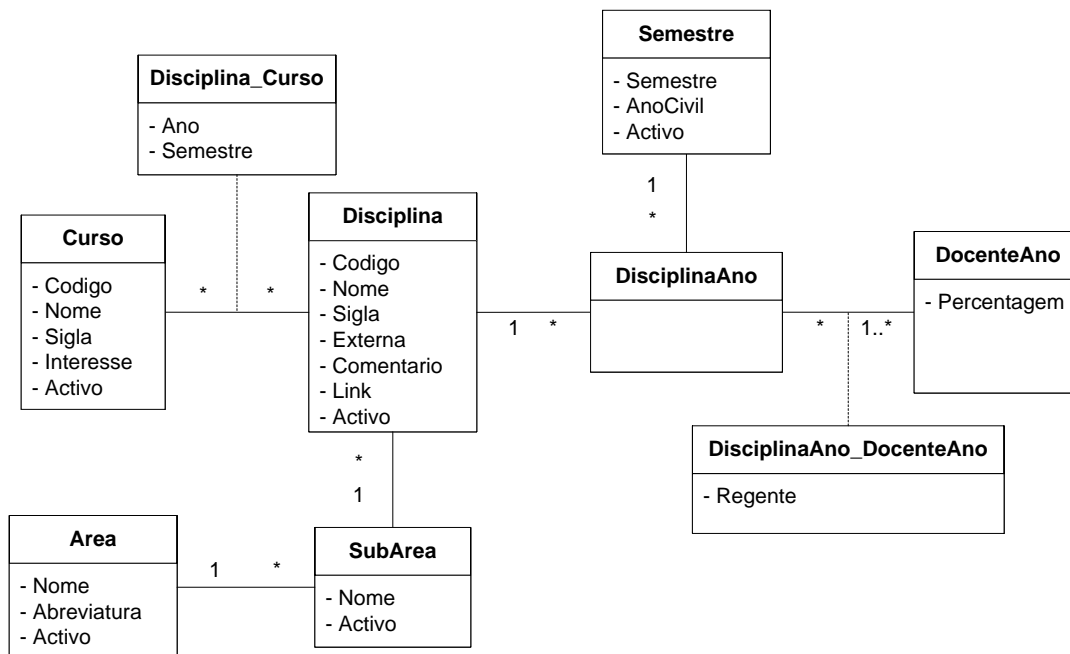


Figura 3.15: Diagrama do domínio para suportar o conceito disciplina.

Também para a construção de horários, surge o conceito de Aula, que não é mais do que o tipo de aulas leccionadas em cada disciplina. Ou seja, caso uma disciplina tenha aulas Teóricas, Práticas e Teórico-práticas, existirão três tipos de aulas relacionados com essa disciplina.

Diferentes tipos de aulas exigem diferentes tipos de salas, por exemplo, uma aula teórica poderá ser leccionada num anfiteatro, enquanto uma aula prática exigirá salas com computadores. Assim, é necessário estabelecer uma relação entre os conceitos aula e sala, guardando as possíveis salas em que cada aula pode ser leccionada.

Cada aula, pode ter várias turmas, que serão realizadas num determinado dia num determinado período de tempo numa determinada sala (dentro do grupo das salas possíveis definidas para a aula com que a turma está relacionada).

Também de realçar que quando uma disciplina é leccionada a vários cursos, pode surgir a necessidade de distinguir as turmas, identificando o curso a que os alunos pertencem.

Todas estas relações podem ser vistas de forma organizada na Figura 3.16.

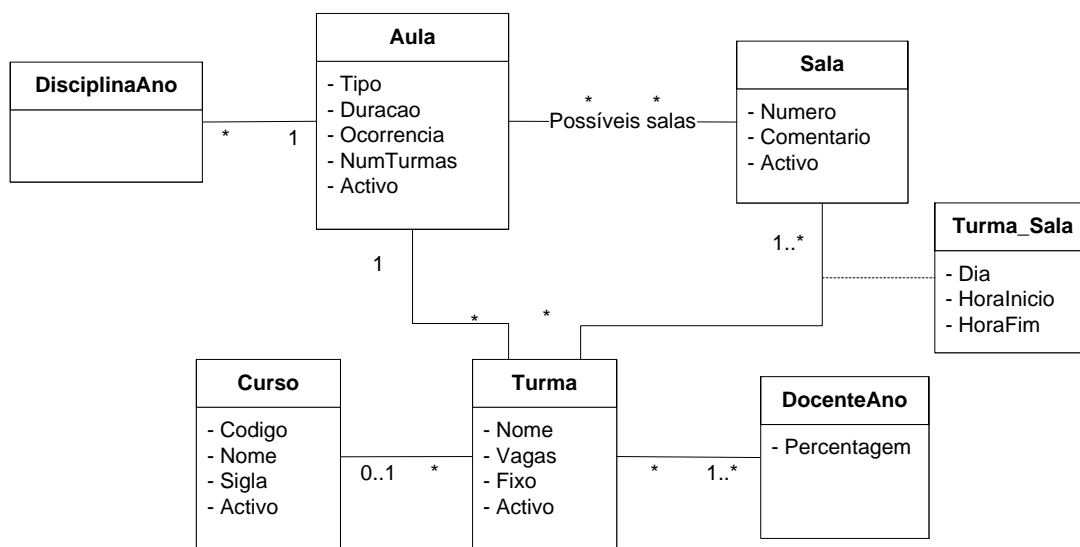


Figura 3.16: Diagrama do domínio para suportar as aulas e turmas de uma determina disciplina.

3.3.4 Alunos

Assim como o Docente, o Aluno também é uma Pessoa (Figura 3.17), mas neste caso fica relacionado com um curso. Aquando da inserção do aluno na base de dados, dados como a média, ECTS³⁶ (*European Credit Transfer and Accumulation System*) e ranking poderão ainda não estar disponíveis, utilizando-se uma classe dedicada para estes dados (evitar dados nulos).

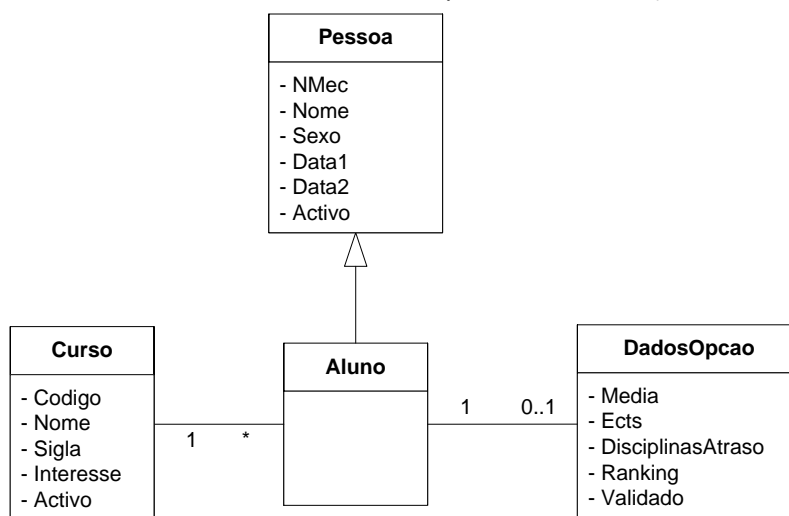


Figura 3.17: Diagrama do domínio para suportar a entidade Aluno.

A entidade aluno surge nesta plataforma essencialmente como elemento participativo e m tarefas que exigem a sua inscrição. Assim, surge o conceito de Inscrição (Figura 3.18), havendo dados específicos para inscrições em diferentes tipos de tarefas. Uma inscrição apenas poderá ser de um dos três tipos indicados.

³⁶ Sistema Europeu de Acumulação e Transferência de Créditos, pretende medir as horas que o estudante tem que trabalhar para alcançar os objectivos do programa de estudos da disciplina.

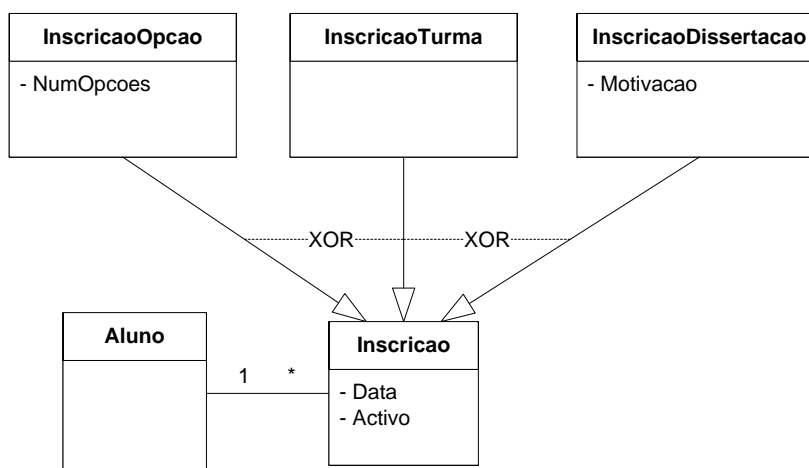


Figura 3.18: Diagrama do domínio para suportar as inscrições de alunos em eventos específicos.

3.3.5 Dissertações

De forma a suportar todo este contexto das inscrições, criou-se o conceito de período, através do qual os processos de inscrições serão iniciados e terminados de forma automática, sem que exista a necessidade de intervenção humana. Para cada tipo de período apenas haverá um activo, sendo que os tipos de períodos identificam as várias tarefas realizadas, como inscrições em Dissertações e Opções.

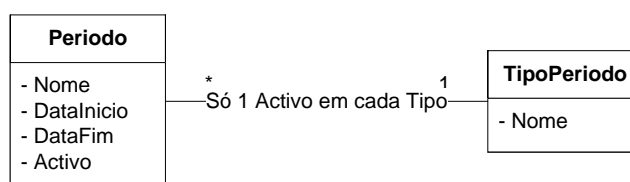


Figura 3.19: Diagrama do domínio para suportar o conceito de Período de inscrições.

Finalmente podemos abordar o conceito de dissertação (Figura 3.20), que estará relacionado com um período de inscrições para permitir a inscrição dos alunos nas mesmas. Uma dissertação poderá não se adaptar a todos os cursos, estabelecendo-se uma relação com os vários cursos em que faz sentido que ela seja realizada.

Também como forma de filtragem, cada dissertação encaixa-se numa determinada área de ensino, surgindo a relação com a classe SubArea. Assim, os alunos poderão pesquisar pelas dissertações que melhor se adaptam aos seus interesses.

Uma dissertação poderá ter vários docentes relacionados, sendo importante conhecer os dados do docente relativos ao semestre em causa, relacionando-se com DocenteAno e não com Docente.

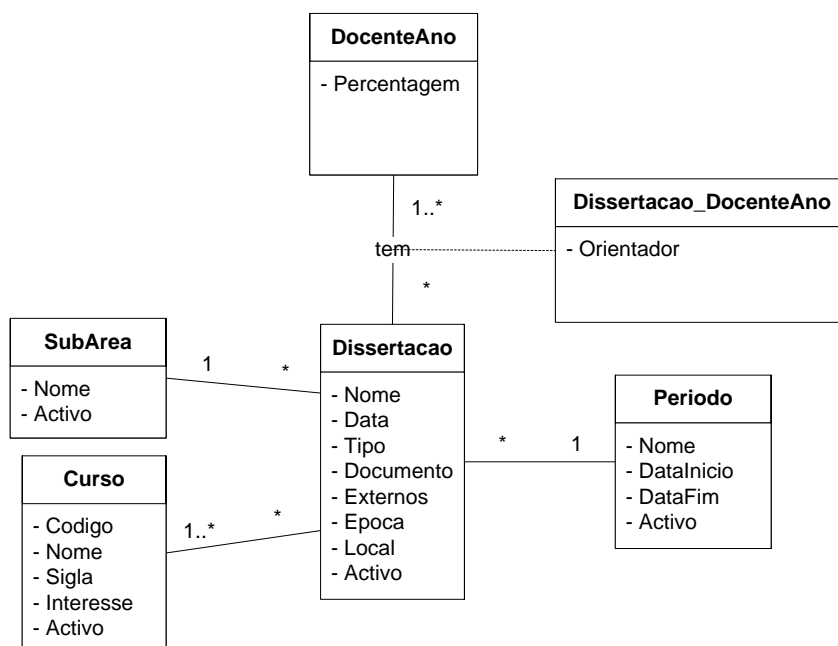


Figura 3.20: Diagrama do domínio para suportar o conceito de dissertação.

Para um aluno se inscrever numa dissertação, tem de apresentar um texto demonstrativo do seu interesse (Figura 3.21). Caso o docente escolha esse aluno para realizar a dissertação, será criado um registo na classe *DissertacaoAcordo* relativo à inscrição aceite, sendo aqui que serão guardados os dados relativos ao acordo.

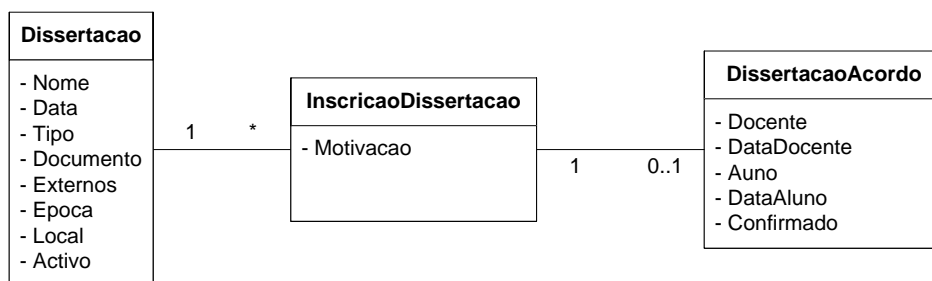


Figura 3.21: Diagrama do domínio para suportar as inscrições dos alunos em dissertações.

3.3.6 Opções

Através da análise da Figura 3.22, verifica-se que uma opção acaba por ser uma disciplina normal, no entanto, tem alguns com factores distintivos. Pode estar aberta ou fechada, o que define o seu funcionamento no semestre em causa (relação com *DisciplinaAno*). Também de realçar a relação com o *Período*, que identifica para que período de inscrições a disciplina de opção foi lançada.

A escolha das disciplinas de opção é um processo um pouco mais complicado de gerir, dada a necessidade de um “concurso” para a escolha de quem lecciona a disciplina, nesse contexto foi criado o conceito de *OpcaoProposta*, onde um docente apresenta dados mais específicos da disciplina e uma equipa de docentes para a leccionar.

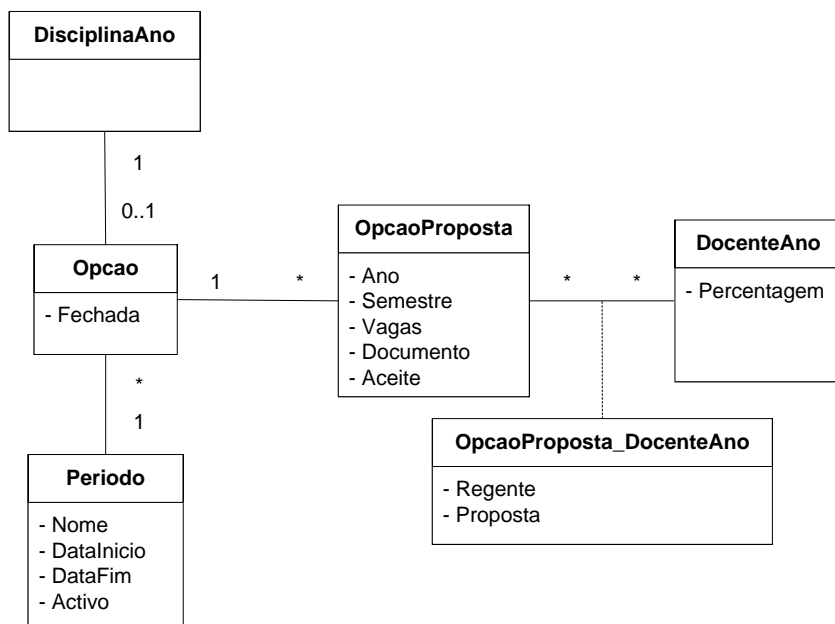


Figura 3.22: Diagrama do domínio para suportar as disciplinas de opção.

De forma a suportar a inscrição dos alunos nas opções, como se pode ver na Figura 3.23, foi criado o conceito de InscricaoOpcao, que é uma inscrição em que se define o número de opções em que se deseja inscrever, bem como as opções ordenadas conforme a sua preferência.

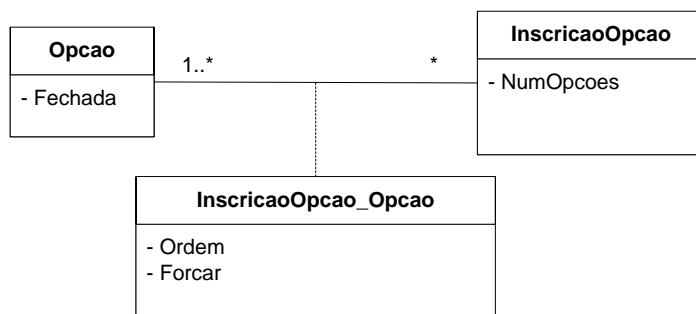


Figura 3.23: Diagrama do domínio para suportar a inscrição dos alunos em disciplinas de opção.

A inscrição em turmas também é suportada (Figura 3.24), o conceito Escolha permite colocar as várias turmas preferidas do aluno. Sendo uma inscrição um conjunto de várias escolhas.

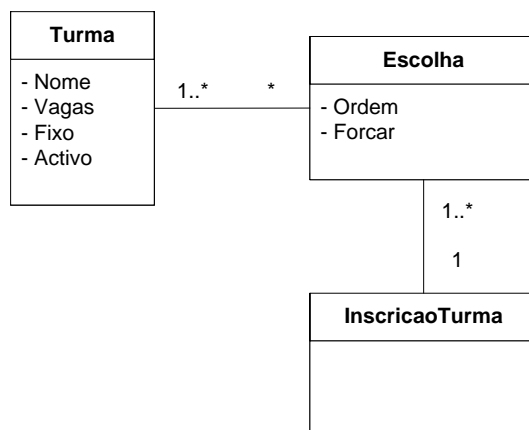


Figura 3.24: Diagrama do domínio para suportar a inscrição de alunos em turmas.

3.3.7 Wish List

Como analisado anteriormente (ver secção 3.1.1.4), cada docente terá uma lista de desejos para cada um dos semestres (relação com DocenteAno em vez de Docente). Possibilitando-lhe a oportunidade de indicar quais as disciplinas que melhor se enquadram com as suas áreas de interesse e investigação, fisicamente suportado através da relação de *Wish List* com DisciplinaAno. Também os dados relativos às disciplinas e tipos de aulas preferidas são suportados por esta relação. As incompatibilidades de horário são guardadas na própria classe *Wish List*.

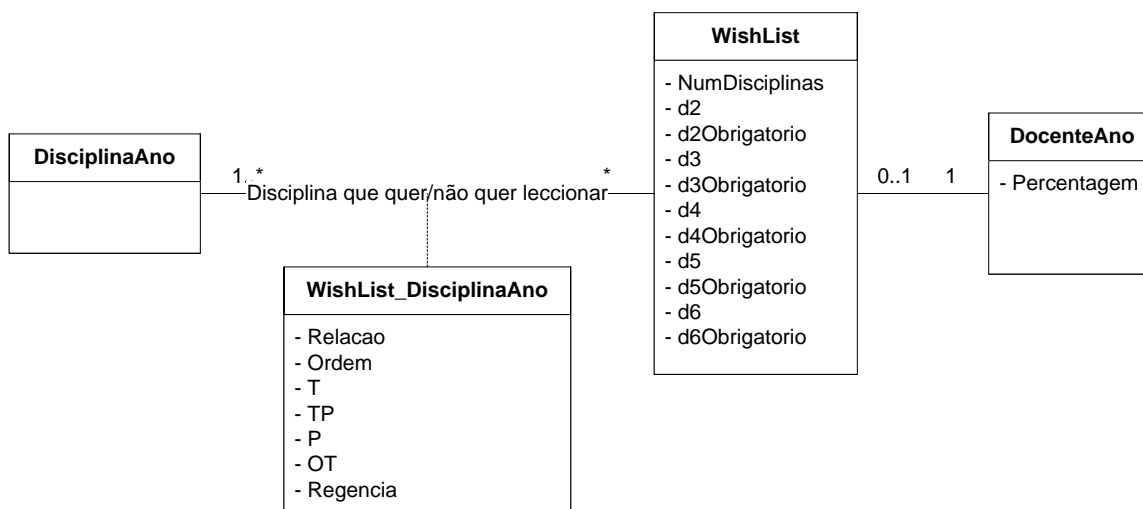


Figura 3.25: Diagrama do domínio para suportar as listas de desejos dos docentes.

3.3.8 Administração

Dado que um Administrador não necessita dos dados associados a uma Pessoa, optou-se pela sua criação de forma independente.

Ao nível de acesso ao *Back Office* a plataforma tem de ser muito flexível, permitindo o acesso de administradores e docentes (a pensar num futura contribuição dos docentes no auxílio da distribuição de serviço docente). Desta forma, a melhor solução passou pela criação do

conceito Acesso, onde é possível definir as regras de cada tipo de acesso, definindo-se uma espécie de grupos de utilizadores, onde os pertencentes a um tipo de acesso terão direito às permissões nele definidas. Como em situações anteriores, torna-se importante manter a informação dos tipos de acesso que cada administrador ou docente já teve e em que períodos.

Torna-se mais fácil perceber as relações acima esclarecidas através da análise da Figura 3.26.

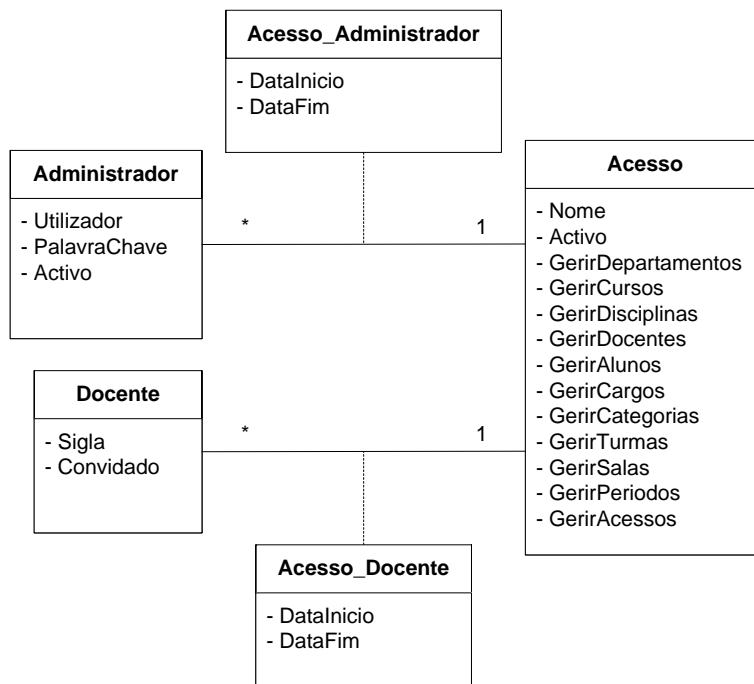


Figura 3.26: Diagrama do domínio para suportar o controlo de acessos.

De forma a permitir a adaptabilidade da plataforma aos vários ambientes e aos vários anos, criou-se uma classe configuração (Figura 3.27), que irá conter dados relativos ao endereço da página Web, e-mail da administração, períodos de inscrições activos e semestres relativos às wish lists.

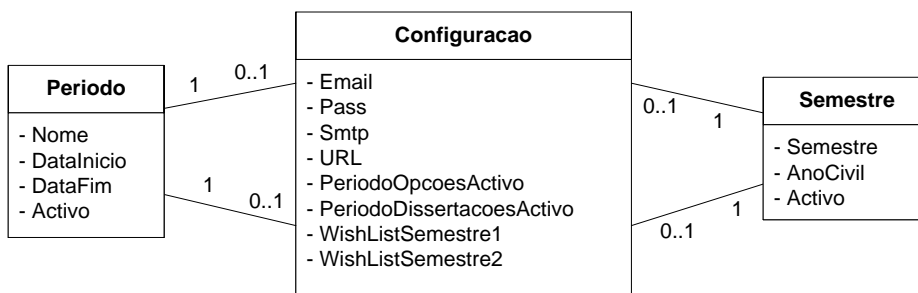


Figura 3.27: Diagrama do domínio para suportar a configuração da plataforma em tempo-real.

Para obter algum feedback por parte dos utilizadores foi realizado um questionário. Uma das soluções poderia passar por receber o questionário por e-mail sem que fosse necessário criar uma infra-estrutura para o suportar, no entanto, o poder de processamento das informações

obtidas seria muito inferior. Assim, cada utilizador apenas pode responder ao questionário uma vez (Figura 3.28), apresentando a sua opinião relativamente à plataforma.

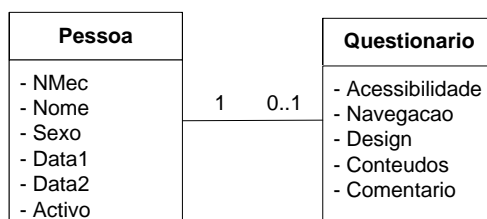


Figura 3.28: Diagrama do domínio para suportar o questionário.

3.4 Sumário

Neste capítulo foi feita uma análise cuidada das várias tarefas que serão realizadas nesta plataforma e de todas aquelas que é importante ter um suporte estrutural. Ao longo desta análise foi feito metodicamente um estudo do papel dos vários intervenientes e dos tipos de informação exigidos, para que no final fosse possível definir os actores da plataforma e as fontes de informação mais ajustadas.

Como consequência da análise acima referida, surgem as várias funcionalidades que cada uma das tarefas exige, sendo feita uma divisão pensada de toda a plataforma em blocos menores e portanto de melhor percepção.

Finalmente apresentou-se o modelo do domínio, onde são apresentadas todas as relações entre os vários conceitos que foram surgindo ao longo de toda a análise, chegando-se assim também a uma estrutura base capaz de atingir os objectivos definidos.

4 Interacção

Analisadas todas as tarefas que serão suportadas pela plataforma e definida toda a base estrutural que as suporta, apenas resta definir a forma de interacção dos actores com os dados, desenvolvendo-se uma base interactiva dinâmica e usável, capaz de suportar tanto quanto possível as funcionalidades que as tarefas exigem. Como consequência da utilização destas interacções num ambiente de teste “*in service*”, serão apresentados alguns cuidados tomados ao nível da manutenção de toda a plataforma.

4.1 Visão Geral

A interface da aplicação é o principal ponto de contacto entre o utilizador e a plataforma, tornando-se num dos principais elementos desta complexa arquitectura. Dada a grande variedade de ferramentas disponíveis e as diferenças entre os utilizadores, é fundamental tirar proveito de algum trabalho de investigação já feito nesta área. Neste contexto, as dez heurísticas de Jakob Nielsen [45] são uma base de trabalho importante para desenvolver uma interface usável. Assim, apresenta-se de forma resumida a síntese de um conhecimento maior.

1ª – Feedback: o sistema deve de informar o utilizador daquilo que ele está a fazer;

2ª – Falar a linguagem do utilizador: os termos utilizados devem-se adaptar ao contexto da ferramenta e do utilizador, evitando o uso de termos técnicos;

3ª – Saídas claramente demarcadas: o utilizador deve controlar o sistema, podendo abortar qualquer tarefa quando desejar;

4ª – Consistência: um mesmo comando deve estar disponibilizado sempre no mesmo local e o seu efeito ser sempre o mesmo;

5ª – Prevenir erros: evitar situações de erro;

6ª – Minimizar a sobrecarga de memória do utilizador: o utilizador não se deve de ter de se lembrar de nada para executar uma acção, sendo apresentados os dados necessários;

7ª – Atalhos: para utilizadores mais experientes executarem as operações mais rapidamente;

8ª – Diálogos simples e naturais: deve apresentar-se exactamente a informação que o utilizador precisa no momento;

9ª – Boas mensagens de erro: linguagem clara e sem códigos;

10ª – Ajuda e documentação: o ideal seria um software tão simples de utilizar quem nem precisa de ajuda, mas no caso de precisar, esta deve estar facilmente disponível.

A implementação de uma interface altamente usável que respeite grande parte das heurísticas acima referidas e que honre todos os requisitos analisados para as tarefas a realizar é um processo bastante moroso, requerendo muitas vezes a implementação e teste de várias soluções para que no final o resultado seja usável.

4.1.1 Restrições

Uma vez que uma página Web é acedida e funciona consoante as características das máquinas dos utilizadores, há estatísticas relativas à resolução de ecrã e ao número de cores suportadas que devem ser consideradas.

Segundo a W3Schools [46], actualmente, 86% dos utilizadores da internet utilizam uma resolução igual ou superior a 1024x768 (Figura 4.1). Dado o daro crescente das resoluções utilizadas, não tem sentido limitar as dimensões de toda a estrutura interactiva para a resolução de 800x600, podendo assim tirar proveito de uma maior resolução e conseqüente maior área de trabalho.

Date	Higher	1024x768	800x600	640x480	Unknown
January 2008	38%	48%	8%	0%	6%
January 2007	26%	54%	14%	0%	6%
January 2006	17%	57%	20%	0%	6%
January 2005	12%	53%	30%	0%	5%
January 2004	10%	47%	37%	1%	5%
January 2003	6%	40%	47%	2%	5%
January 2002	6%	34%	52%	3%	5%
January 2001	5%	29%	55%	6%	5%
January 2000	4%	25%	56%	11%	4%

Figura 4.1: Estatísticas da W3Schools acerca das resoluções de ecrã mais utilizadas na internet[47].

Relativamente ao número de cores, a W3Schools indica que actualmente 90% dos utilizadores utilizam dispositivos de 24 ou 32 bits [46], permitindo a utilização de 16,777,216 cores. Esta característica dá uma grande liberdade de utilização de cores na plataforma DSD.

Date	16,777,216	65,536	256
January 2008	90%	8%	2%
January 2007	86%	11%	2%
January 2006	81%	16%	3%
January 2005	72%	25%	3%
January 2004	65%	31%	4%
January 2003	51%	44%	5%
January 2002	43%	50%	7%
January 2001	37%	55%	8%
January 2000	34%	54%	12%

Figura 4.2: Estatísticas da W3Schools acerca do número de cores mais utilizado na internet[47].

4.1.2 Área de trabalho

Tratando-se de uma plataforma bastante complexa, é fundamental dar alguma atenção a toda a base de interacção, a organização dos conteúdos. Após alguma análise e estudo, no contexto geral a área de trabalho foi dividida em cinco blocos essenciais (Figura 4.3).



Figura 4.3: Base interactiva da plataforma DSD.

Cabeçalho

O Cabeçalho (Figura 4.4) contém o logótipo da plataforma com a respectiva versão identificada e a informação acerca da conta activa, permitindo editar a sua conta e sair da plataforma.



Figura 4.4: Cabeçalho da aplicação.

Assistente de Navegação

O assistente de navegação (Figura 4.5) ajuda o utilizador a situar-se na árvore de navegação da ferramenta, permitindo a deslocação para pontos anteriores. Servirá de auxílio ao menu, proporcionando atalhos para acesso mais rápido.



Figura 4.5: Assistente de navegação da aplicação.

Menu

O Menu (Figura 4.6) é a base de toda a navegação, disponibilizando de forma organizada grande parte das funcionalidades a que o utilizador tem acesso, adaptando-se às suas permissões através da não disponibilização das funcionalidades a que este não tem acesso. Assim, a utilização do menu permite uma rápida aprendizagem da utilização da plataforma, conhecendo rapidamente as capacidades da ferramenta.

De forma a dar importância aos dados do utilizador, optou-se por dividir cada uma das tarefas apresentadas no menu em duas secções:

- **Geral** – funcionalidades de cariz público, que contem informações disponíveis a todos;
- **Pessoal** – funcionalidades de cariz particular, contendo informações privadas.

No menu é sempre indicada a secção em que o utilizador está, como se pode ver na Figura 4.6 é dado algum realce ao item “Listagem” pois é a secção em que o utilizador se encontra.



Figura 4.6: Menu da aplicação.

Secção

A secção (Figura 4.7) é o local onde serão colocados todos os dados, sendo dado algum realce ao seu título, com o objectivo de manter o utilizador consciente da secção em que se encontra.

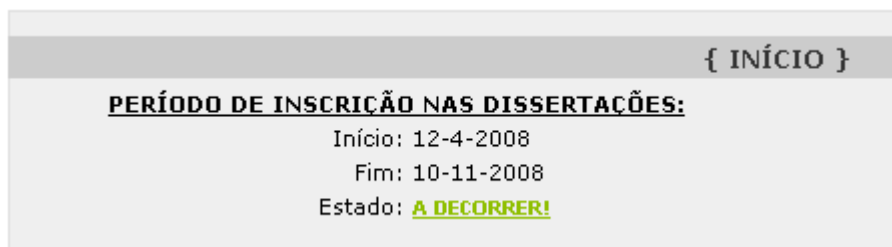


Figura 4.7: Aspecto de uma secção da aplicação.

Rodapé

O rodapé (Figura 4.8) tem como objectivo fornecer alguma informação acerca do local em que a aplicação está a ser desenvolvida. Também a informação acerca do seu criador e o contacto do administrador se encontra aqui disponível.



Figura 4.8: Rodapé da aplicação.

Juntando todos estes blocos obtemos uma interface bastante simples e fácil de utilizar (Figura 4.9), sendo o grande ponto de partida para respeitar grande parte das heurísticas caracterizadoras de uma interface usável. Também se investiu bastante na construção de uma interface atractiva e rica a nível visual, obtendo-se um desenho bastante limpo promovendo a fácil detecção de elementos chave.

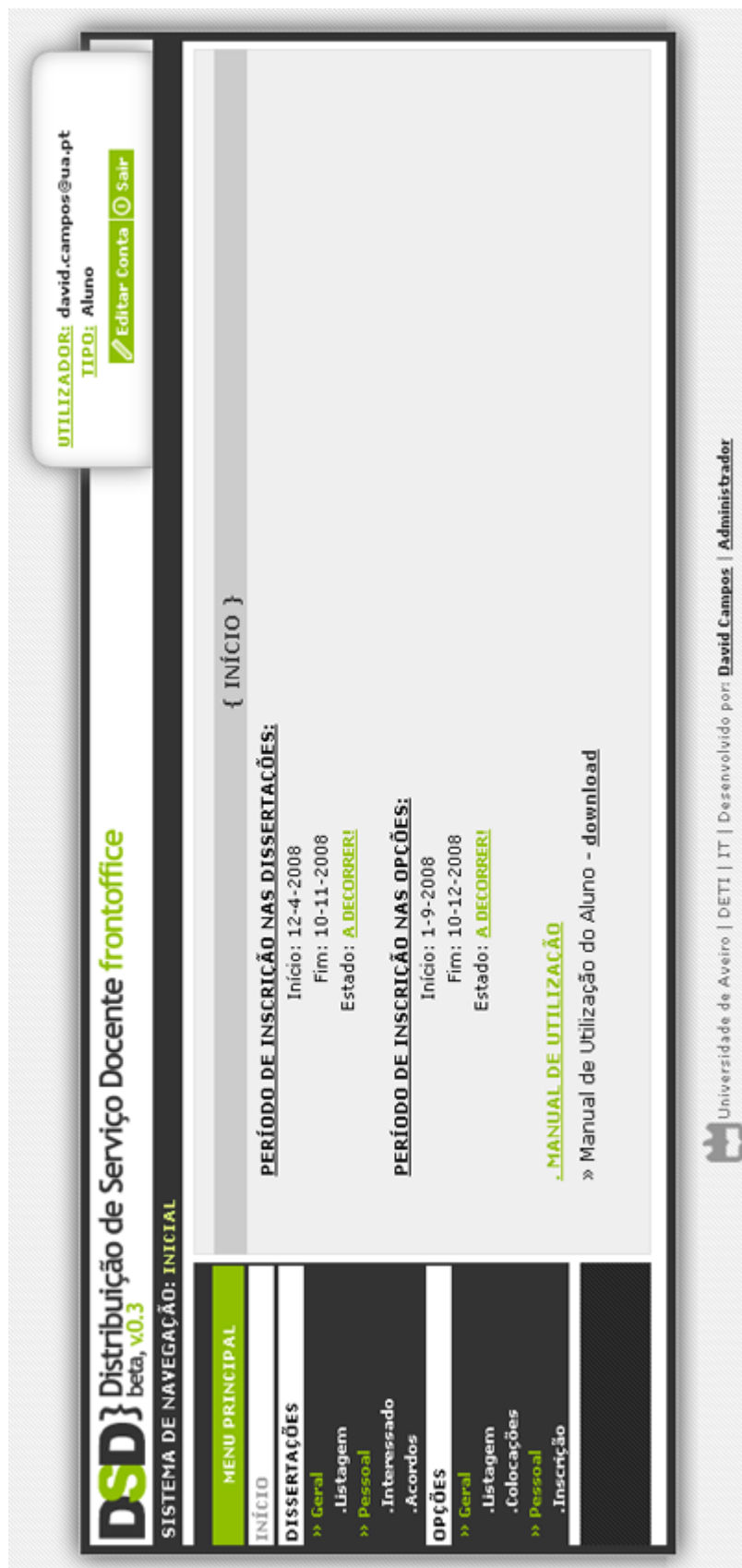


Figura 4.9: Vista geral da interface do *Front Office* da plataforma DSD.

4.1.3 Características Gerais

Com utilização de AJAX, ao longo de toda aplicação foram utilizadas técnicas para dinamizar a interação com o utilizador, poupando tempo e tomando algumas tarefas mais simples.

Validação de Dados

Nas funcionalidades em que o utilizador adiciona dados ao sistema, antes que estes entrem de facto na base de dados, precisam de ser validados, para que não ocorram problemas de incompatibilidade de tipos de dados ou inconsistência dos mesmos.

Tipicamente, esta validação apenas é feita após o utilizador fazer o pedido de execução da acção, tendo de carregar toda a página para disponibilizar os problemas encontrados. Nesta aplicação a validação será feita em tempo real, ou seja, logo após o utilizador digitar os dados, em caso de erro será dado realce à caixa de texto mostrando informação relativa ao erro cometido. Na Figura 4.10 podemos ver um exemplo da utilização desta técnica.



Figura 4.10: Validação instantânea dos dados introduzidos.

Processamento de dados

Com a utilização de AJAX, a execução de grande parte das acções deixa de dar *feedback* no *browser*, assim, para que o utilizador tenha o total controlo daquilo que está a fazer, é mostrada uma animação indicando o processamento de dados. Na Figura 4.11, após a execução da acção “Ver Todas”, é demonstrado o indicador de processamento de dados para que o utilizador tenha a noção do que está a ser feito e não cancele a operação.



Figura 4.11: Indicação do processamento de informação.

Utilização de *Modal Windows*

Com a utilização da plataforma DSD, em alguns casos é necessário disponibilizar muita informação ao mesmo tempo, sem que se perca o contexto actual da aplicação. Assim, estudou-se a utilização de *Modal Windows*, que são janelas que surgem de forma embutida na própria janela do *browser* para chamar a atenção para uma determinada informação, exigindo a interacção do utilizador para que esta seja fechada. Como podemos ver na Figura 4.12, após o utilizador escolher a acção de ver os dados da dissertação, estes surgem numa *Modal Window* despertando o interesse do utilizador. Ao redor da janela é utilizada uma cor preta com transparência, para que o utilizador continue a saber onde se encontra na aplicação.

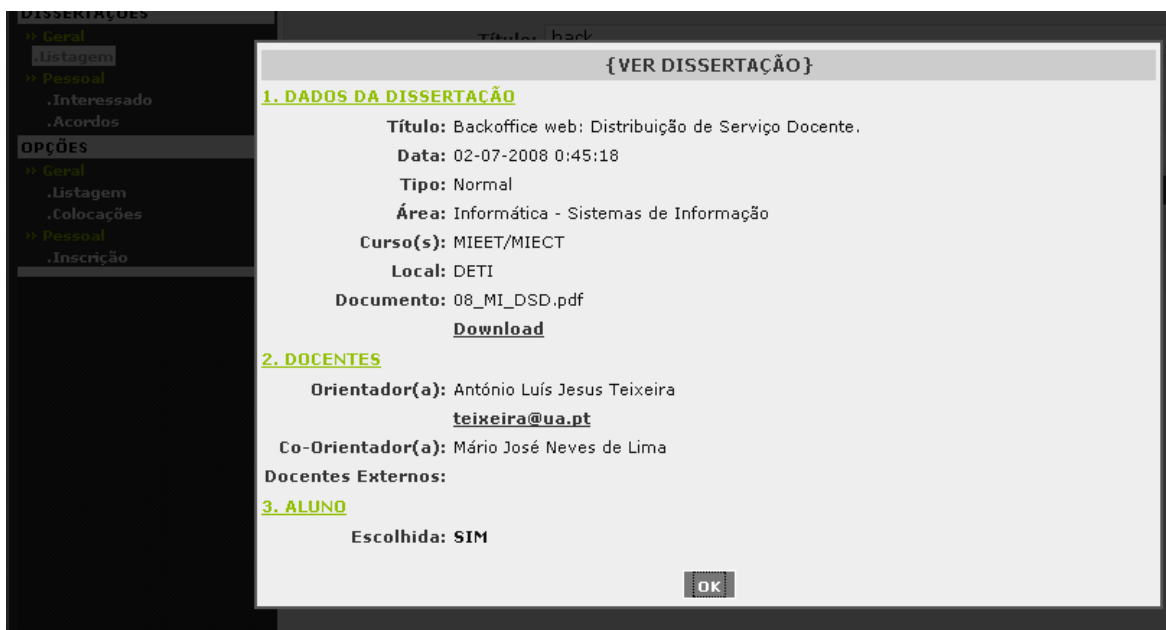


Figura 4.12: Utilização de *Modal Windows* na apresentação de informação.

Também nas mensagens de sucesso se optou pela utilização de *Modal Windows* (Figura 4.13), com a utilização de AJAX quando eram executadas duas acções iguais seguidas, era perdido o *feedback*, surgindo a dúvida da execução com sucesso da segunda acção. Desta forma, o utilizador é obrigado a premir o botão “OK” para confirmar o sucesso de execução da acção pedida.

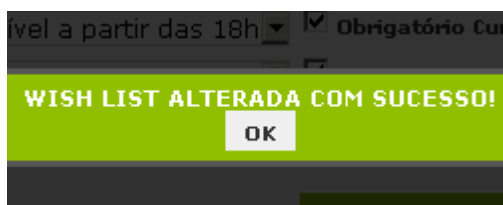


Figura 4.13: Utilização de *Modal Windows* nas mensagens de sucesso.

4.1.4 Ajuda

Ao longo de toda aplicação foi tomado em conta o desenvolvimento das funcionalidades para que estas fossem utilizadas de forma lógica, sem a necessidade qualquer tipo de auxílio. No entanto, em algumas situações isso não foi possível, recorrendo-se à utilização de mensagens simples e claras acerca das precauções que o utilizador deverá ter (Figura 4.14).

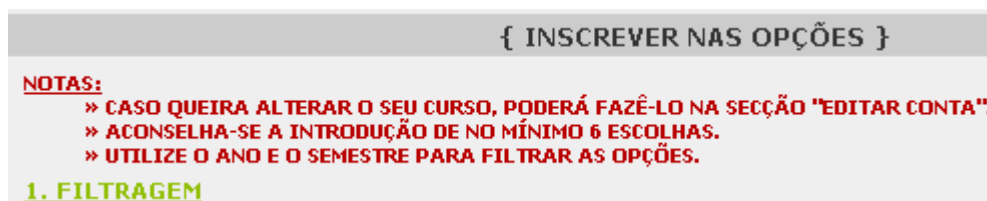


Figura 4.14: Ajuda em tempo real disponibilizada na aplicação.

Contudo, também foram criados manuais em PDF (*Portable Document Format*) para auxiliar a utilização da aplicação, existindo versões diferentes direccionadas a cada tipo de utilizador.

Com o passar do tempo, a aplicação foi sofrendo alterações, sendo corrigidos problemas encontrados e adicionadas novas funcionalidades, de forma a informar o utilizador destas alterações foi criado um *ChangeLog* (Registo de alterações), onde são registadas todas as alterações realizadas no *Front Office*.

4.2 Aplicação de Tarefas

É fundamental demonstrar de que forma as várias tarefas previamente descritas (secção 3.1.1) se encaixaram na base interactiva definida, apresentando-se algumas situações de maior interesse, seja pela sua complexidade ou perspicácia.

Ao longo desta secção poderão surgir imagens com zonas desfocadas, isto acontece devido à alta confidencialidade dos dados, não podendo ser disponibilizados publicamente.

4.2.1 Escolha de Dissertações

Na adição de uma dissertação (como noutras funcionalidades), o docente ou o administrador tem de indicar os cursos a que esta se aplica. Para realizar esta tarefa foram consideradas várias soluções, porém, acabou por se considerar que cada curso seria uma caixa de verificação (Figura 4.15), sendo considerados os cursos seleccionados. Alterando o departamento na lista, os cursos disponíveis serão alterados.



Curso(s):

Departamento: DETI

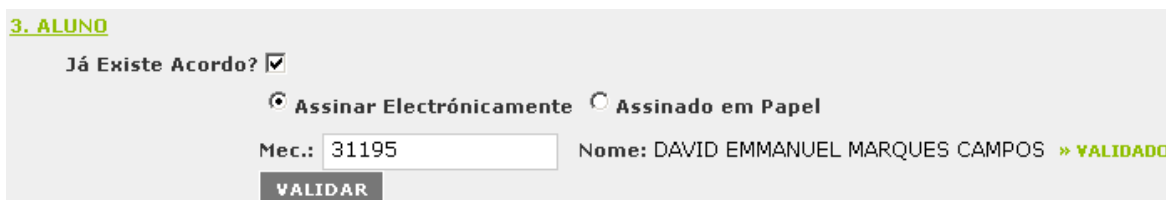
LTIC LTSI MIECT MIEET

Figura 4.15: Escolha dos cursos para a dissertação.

Na indicação do aluno (Figura 4.16), caso a caixa de verificação seja deixada em branco, indica que ainda não existe nenhum aluno escolhido, permitindo que os alunos mostrem interesse. Quando a caixa de verificação tem um visto, tem duas opções:

1ª – Assinar Electronicamente (Acordo Digital): assinará o acordo através da aplicação;

2ª – Assinado em Papel (Acordo em Papel): tem de apresentar um acordo em papel assinado por ambas as partes ao administrador, para que este verifique a veracidade do mesmo.



3. ALUNO

Já Existe Acordo?

Assinar Electrónicamente Assinado em Papel

Mec.: 31195 Nome: DAVID EMMANUEL MARQUES CAMPOS » VALIDADO

VALIDAR

Figura 4.16: Indicação de como o acordo será assinado com o aluno.

No caso de um aluno poder mostrar interesse por uma dissertação, terá de apresentar um texto demonstrativo da sua motivação para a realização da tarefa em questão, fazendo-o através da interface que se encontra na Figura 4.17, sendo enviado um e-mail para o orientador a indicar que um aluno demonstrou interesse por uma dissertação sua.

{ MOSTRAR INTERESSE }

1. DISSERTAÇÃO
Título: Laboratório virtual de sistemas de controlo

2. ALUNO
Motivação:

FECHAR MOSTRAR INTERESSE

Figura 4.17: Interface para demonstração de interesse por uma dissertação.

De seguida, o docente pode aceder à lista de alunos interessados numa das suas dissertações e escolher um aluno para a realizar, sendo enviado um e-mail para ambas as partes a informar que têm de assinar o acordo, disponibilizando uma funcionalidade para assinatura de acordo através do simples acesso a um endereço electrónico. Caso não queiram assinar desta forma, podem aceder à aplicação e na secção de acordos, escolher a opção “Assinar Acordo”, surgindo uma *Modal Window* com toda a informação relativa (Figura 4.18).

{ ASSINAR ACORDO }

1. DISSERTAÇÃO
Título: Backoffice web: Distribuição de Serviço Docente.

2. DOCENTE
Nome: António Luís Jesus Teixeira

3. ALUNO
Nome: JOAO FRANCISCO CRUZEIRO TABORDA
Número Mecanográfico: 32666
Curso: MIECT

4. ASSINATURA
Palavra-Chave:

FECHAR ASSINAR

Figura 4.18: Interface para assinar um acordo de uma dissertação.

Após ambas as partes assinarem o acordo, este terá de ficar disponível em forma de documento. A solução passou por gerar um documento PDF de forma dinâmica, visto este ser um dos formatos mais utilizados a nível global [47]. Contudo, o ASP.net por si só não suporta esta

funcionalidade de forma muito prática, requerendo-se a utilização de uma biblioteca extra. Nesta escolha pesou sobretudo o factor “grátis”, limitando logo à partida o espectro de soluções. A biblioteca encontrada foi o iTextSharp³⁷, permitindo “criar e ler ficheiros PDF sem quaisquer custos adicionais” [47].

Assim, acedendo à lista de acordos, alunos e professores podem fazer download do acordo, sendo este gerado em tempo real com todos os dados do acordo seleccionado.

4.2.2 Escolha de disciplinas de Opção

Como visto na secção 3.1.1.2, o processo de escolha de disciplinas de opção é iniciado com a negociação entre a administração e os docentes das disciplinas de opção que serão apresentadas aos alunos como possíveis escolhas.

Passada essa fase, inicia-se o período de inscrições, onde o aluno pode apresentar uma lista ordenada de opções como inscrição. Podemos ver na Figura 4.19 a interface simplista elaborada para esta tarefa, onde as disciplinas serão filtradas pelo curso do aluno e posteriormente por ano e semestre. Adicionando as disciplinas de opção à lista, com a utilização dos botões de ordenação (à direita da lista) poderá reordenar as suas preferências.

Figura 4.19: Interface de inserção e alteração da inscrição do aluno nas disciplinas de opção.

Após a inscrição do aluno será dado *feedback* imediato do estado da sua colocação, para cumprir este objectivo teve de ser criado um método de colocação de alunos em tempo real.

³⁷ <http://itextsharp.sourceforge.net>

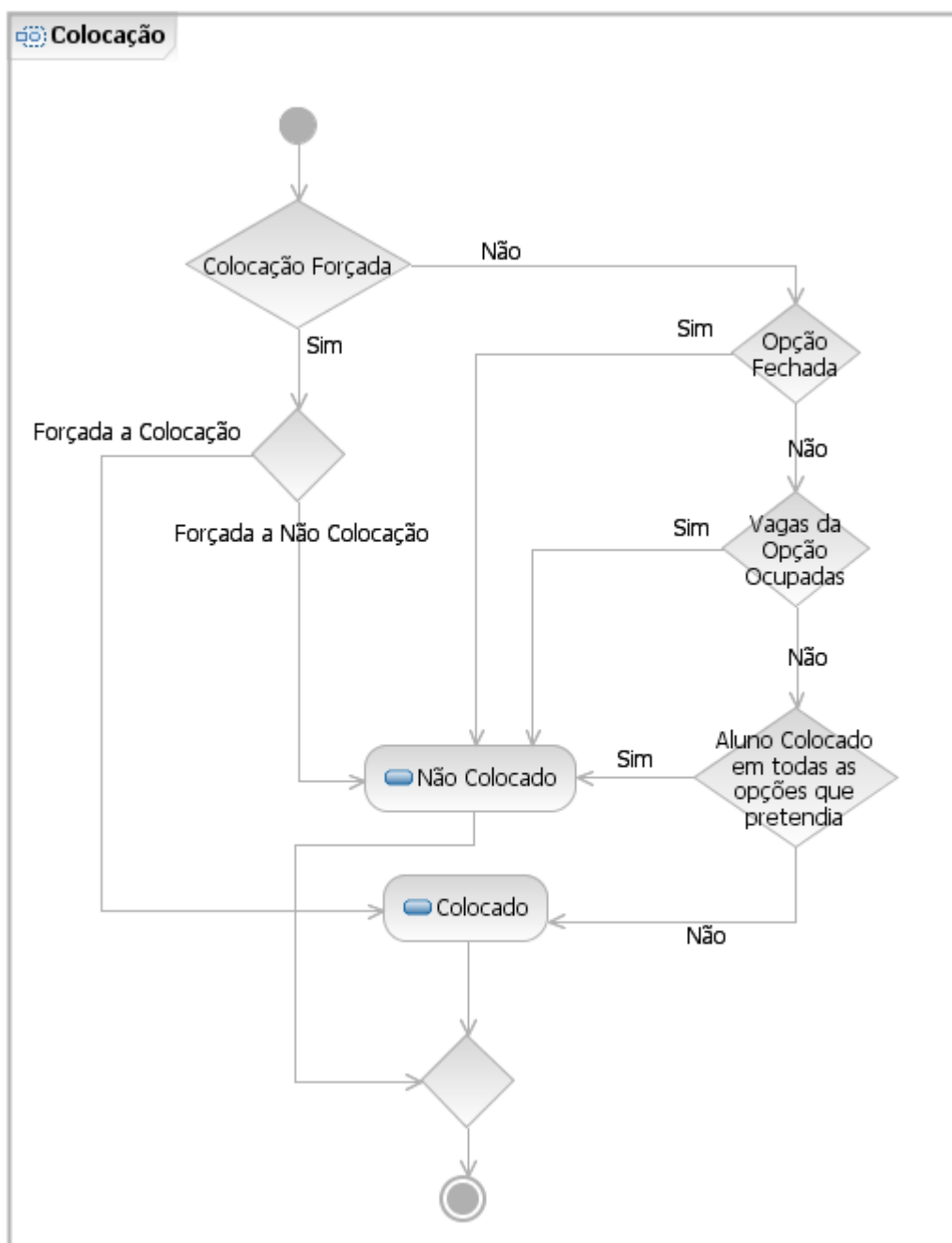


Figura 4.20: Lógica de colocação do aluno na disciplina de opção.

Este método divide-se em três fases essenciais:

1ª Fase - obter lista de alunos inscritos nas opções ordenada por ordem descendente de ranking;

2ª Fase - obter lista das disciplinas de opção e respectivas informações;

3ª Fase - percorrendo a lista de alunos, para cada aluno será analisada a sua lista ordenada de disciplinas de opção, onde para cada escolha será percorrida a lógica de colocação definida na Figura 4.20. Inicialmente verifica-se se a administração forçou de alguma forma a sua

colocação, caso isso não tenha acontecido, há três condições essenciais que tem de ser respeitadas, ou seja, caso de uma delas se confirme, o aluno não será colocado na opção:

- **Opção Fechada** – neste caso a disciplina de opção não será leccionada, pelo que não poderá aceitar colocações;
- **Vagas da Opção Ocupadas** – todas as vagas existentes da disciplina já foram ocupadas;
- **Aluno colocado em todas as opções que pretendia** – o aluno já foi colocado no número de disciplinas de opção que pretendia.

Usando este algoritmo será possível ao aluno ter de imediato *feedback* do estado da sua inscrição, indicando no momento se ele se encontra colocado com sucesso e as disciplinas em que conseguiu obter a colocação. Podemos ver na Figura 4.21 um exemplo desse *feedback*, onde inicialmente são apresentadas as escolhas do aluno e posteriormente os resultados obtidos para cada uma delas.

1. LISTA DE ESCOLHAS	
1ª - Electrónica e Sistemas Médicos	
2ª - Sistemas de Rádio Dedicados	
3ª - Redes de Comunicação em Ambientes Industriais	
4ª - Planeamento em Comunicações Móveis	
5ª - Especificação, Modelação e Projecto de Sistemas Embutidos	
2. RESULTADOS	
1ª - Electrónica e Sistemas Médicos	COLOCADO
2ª - Sistemas de Rádio Dedicados	COLOCADO
3ª - Redes de Comunicação em Ambientes Industriais	COLOCADO
4ª - Planeamento em Comunicações Móveis	NÃO COLOCADO
5ª - Especificação, Modelação e Projecto de Sistemas Embutidos	NÃO COLOCADO
FOI COLOCADO COM SUCESSO! OPÇÕES A COLOCAR: 5 OPÇÕES COLOCADO: 3	
Legenda:	
 - Dentro do Número de Opções definido	
 - Fora do Número de Opções definido	

Figura 4.21: Feedback do estado da inscrição do aluno nas disciplinas de opção.

Para ter uma ideia geral das colocações em todas as disciplinas foi criada uma lista ordenada de colocações, onde se apresentam todos os alunos inscritos na opção, diferenciando os que ficaram colocados dos que não ficaram colocados (Figura 4.22). A ordem da disciplina na inscrição do aluno também é apresentada, demonstrando a sua preferência pela disciplina, o que favorece a comunicação entre os alunos (como na situação descrita na secção 3.1.1.3).

OPÇÃO: FILTRAGEM ADAPTATIVA PARA PROCESSAMENTO DE SINAL E TELECOMUNICAÇÕES			
DOCENTES: TOMÁS ANTÓNIO MENDES OLIVEIRA E SILVA (R)			
MIEET(5º ANO - 1º SEMESTRE)			
OPÇÃO FECHADA!			
	SANDRA CATARINA OTERO DA SILVA E COSTA	6ª	17686
	RUI MANUEL DA CUNHA SACADURA BOTTE	7ª	33407
	PEDRO REGALADO DE MELO ROSA	6ª	33402
	ANDRE PEDRO MENDES	8ª	30852
	TIAGO ALEXANDRE DA COSTA GONCALVES	22ª	33082
	HUGO GOMES LIMA	5ª	31338

OPÇÃO: INTEGRAÇÃO DE REDES			
DOCENTES: FRANCISCO MANUEL MARQUES FONTES (R)			
MIEET(5º ANO - 1º SEMESTRE)			
#	NOME	ESCOLHA	MEC.
	CARLOS MIGUEL DOS SANTOS MIRANDA	2ª	19790
	NUNO MIGUEL TRIGO MOUTINHO NUNES DOS SANTOS	2ª	23010
1	RUI PEDRO DE SA VALBOM	3ª	30066
2	JOAO MONTEIRO SOARES	2ª	32660
3	SAUL MANUEL SEQUEIRA PARADA	1ª	33687
4	JOAO RODRIGUES SINTRA	2ª	30789
5	PEDRO MIGUEL ROCHA MATOS	4ª	33698

Figura 4.22: Lista de colocações provisória nas disciplinas de opção.

4.2.3 Wish List

O grande problema que surgiu nas *Wish Lists* foi a forma de apresentar os dados aos docentes. Após várias tentativas, o preenchimento da lista de desejos foi dividido em quatro etapas:

- 1ª – Disciplinas que se encontram alinhadas com as competências do docente;
- 2ª – Disciplinas que não se encontram alinhadas com as competências do docente;
- 3ª – Preferência das disciplinas que deseja leccionar;
- 4ª – Indisponibilidades de horário.

Após esta divisão, o problema residia na forma de disponibilizar as disciplinas ao docente, optando-se por as apresentar numa estrutura em árvore organizada por área e subárea, onde cada folha será uma disciplina (Figura 4.23). A grande vantagem desta técnica é a selecção das disciplinas em bloco, por exemplo, caso um docente não se sinta de todo capaz de leccionar as disciplinas de uma subárea, pode de imediato seleccionar a subárea, seleccionando todas as disciplinas que lhe dizem respeito.

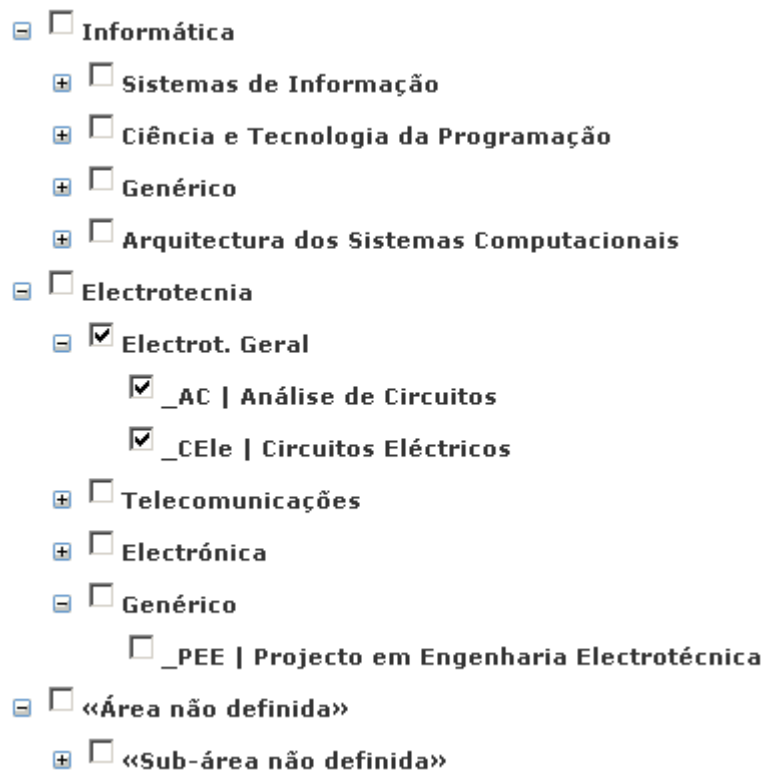


Figura 4.23: Estrutura em árvore de disciplinas organizada por área e subárea.

De notar que as disciplinas escolhidas na 1ª etapa não irão constar na árvore da 2ª etapa, de forma a facilitar a tarefa do docente.

Após o cumprimento das duas etapas iniciais, o docente terá de apresentar uma lista ordenada das disciplinas que prefere leccionar, tendo de indicar para cada uma delas os tipos de aula que pretende. Para realizar a escolha dos tipos de aulas, optou-se por um sistema bastante simples mas ao mesmo tempo extremamente funcional. Como podemos ver na Figura 4.24, o docente selecciona a disciplina que pretende e escolhe os tipos de aulas preferidos, pressionando o botão alterar, de imediato surgirão os tipos indicados em frente ao nome da disciplina.

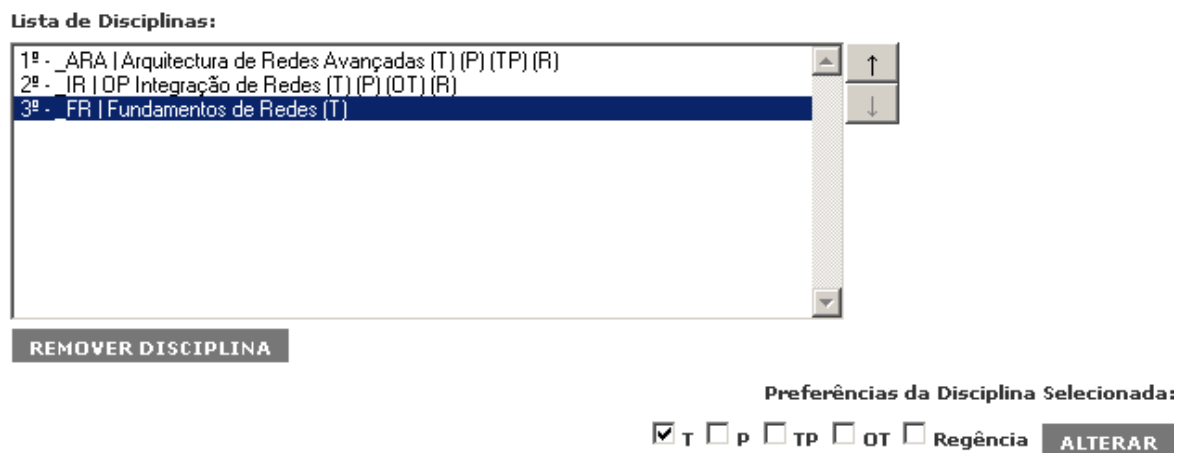


Figura 4.24: Escolha das disciplinas preferidas a leccionar.

Assim como os alunos nas inscrições nas disciplinas de opção, os docentes também terão disponível o resumo da sua *Wish List*, onde serão apresentados todos os dados introduzidos de forma simples e de fácil percepção.

Para ver as *Wish Lists* adicionadas por todos os docentes, o administrador precisa de uma funcionalidade que agrupe e resuma todos os dados adicionados. Como se pretende fazer algum estudo estatístico sobre estes dados, pelo menos para já, a solução passa por exportar os dados para uma folha de Excel. Contudo, o ASP.NET não suporta esta funcionalidade de forma nativa, exigindo-se mais uma vez a utilização de uma biblioteca externa. Também neste caso, a existência de custos adicionais está fora de questão. Assim, a solução encontrada foi a CarlosAg Excel Xml Writer Library³⁸ [47], que permite “criar e ler ficheiros Excel sem quaisquer custos adicionais e sem a instalação do Excel no servidor” [48].

Assim, colocando os docentes à esquerda e as várias disciplinas organizadas por áreas e subáreas no topo, com a utilização de cores representativas das intenções dos docentes, facilmente se dá uma visão geral das preferências de todos os docentes (Figura 4.25).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	WISH LIST - DISCIPLINAS [1º SEMESTRE DE 2008/2009]																					
2					E-APS								E-C			E-E						
3		_CN	_SAB	_SIS	_CAV	_CSS	_LAPS	_MPE	_MPEI	_PDS	_PDV	_RP	_TPS	_CDif	_CDig	_SCI	_E	_E1	_E3	_EBP	_EMPSE	_EP
4																						
5																						
6																						
7																						
8																						
9																						

Figura 4.25: Visualização das *Wish Lists* dos vários docentes numa folha de Excel.

Após a lista de intenções dos docentes (Figura 4.26) apresenta-se a legenda que identifica o que cada uma das cores representa, tornando o documento auto-descritivo. Abaixo, surge uma tabela que resume as intenções dos docentes por disciplina, ou seja, apresenta o número de docentes que colocaram a disciplina na ordem apresentada à esquerda. Esta tabela é importantíssima para a distribuição de serviço docente, permitindo identificar para cada disciplina o número de docentes que estão interessados a leccioná-la.

³⁸ <http://www.carlosag.net/Tools/ExcelXmlWriter>

88		- Disciplina Escolhida para Leccionar
89		- Disciplina Escolhida para Leccionar com Preferência (Ordem Indicada na Célula)
90		- Disciplina Escolhida para Não Leccionar
91		
92	RESUMO [NÚMERO DE PREFERÊNCIAS PARA CADA DISCIPLINA]	
93	1ª	0 1 1 1 0 1 0 2 2 1 1 0 0 0 3 0 3 3 0 1 2 1 0 0 0 1
94	2ª	0 0 0 0 0 0 3 2 0 0 0 1 0 1 0 2 2 2 0 0 1 0 1 1 1 0
95	3ª	1 0 1 0 0 3 3 1 0 0 0 0 0 0 2 6 0 0 0 3 1 0 0 0 0
96	4ª	0 0 0 0 0 0 0 3 0 0 0 0 1 0 2 0 1 0 0 0 0 0 0 0 0
97	5ª	0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1
98	6ª	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 2 0 0

Figura 4.26: Processamento de informação das *Wish Lists*.

Além da vista acima descrita, ainda existem mais três de funcionamento equivalente:

- **Vista de Regência** – permite identificar os docentes que pretendem ser regentes das disciplinas;
- **Vista de Aulas** – permite identificar os tipos de aulas que os docentes pretendem leccionar para cada uma das disciplinas preferidas;
- **Vista de Horários** – permite identificar as indisponibilidades de horários dos docentes.

4.2.4 Gestão de Entidades e Conceitos

A gestão de todos os conceitos da plataforma é uma tarefa de grande complexidade, exigindo-se uma organização cuidada e acima de tudo a apresentação de funcionalidades de forma simples e ao mesmo tempo funcional. É possível verificar a sua complexidade através do menu do *Back Office* (Figura 4.27), que disponibiliza uma vasta gama de funcionalidades para gestão de todos os conceitos da plataforma (um item com seta à direita contém um submenu com funcionalidades relativas à entidade especificada).

MENU PRINCIPAL	
Início	
Departamentos	▶
Cursos	▶
Disciplinas	▶
Opções	▶
Docentes	▶
Alunos	▶
Cargos	▶
Categorias	▶
Salas	▶
Dissertações	▶
Períodos de Inscrições	▶
Administradores	▶
Acessos	▶
Semestres	▶
Dados Externos	▶
Wish List	
Horários	
Configuração	
Inactivos	

Figura 4.27: Menu principal de gestão de conceitos do *Back Office*.

4.2.4.1 Importação dos dados

A importação de dados é uma tarefa fundamental desta versão da plataforma, visto a falta de dados que dessem corpo a toda a estrutura definida.

A primeira fonte de informação considerada (secção 3.1.3) foi a ferramenta de construção de horários, onde os dados se encontram disponíveis num ficheiro de texto simples. Para os importar para a base de dados teve de se construir um analisador sintáctico com o objectivo de detectar as entidades chave e os respectivos dados associados. A Figura 4.28 exemplifica a representação das entidades no ficheiro de texto, verificando-se alguma dificuldade na sua interpretação de forma directa.


```

DEPARTAMENTOS
Departamento: 5 DETI "Electrónica, Telecomunicações e Informática"

CURSOS
Curso: 8240 MIECT "Mestrado Integrado em Engenharia de Computadores e Telemática"

SALAS
Departamento: DETI
Sala: ANF_IV para aulas T (144)

CATEGORIAS
Categoria: Professor Catedrático

CARGOS
Cargo: Coordenador da Comissão Pedagógica ALT ALT

DOCENTES
Docente: M 1616 ALT "António Luís Jesus Teixeira" DETI "Professor Auxiliar"
        "teixeira@ua.pt" 100 100

DISCIPLINAS
Disciplina: 50071 _ART "Análise de Redes de Telecomunicações"
Curso: 8204 5 0 0
Aulas: TP 1x(3h00m) 104 106 119 123 225 320
Horário: AN SS TP1 20 104 2 13h00m-16h00m

```

Figura 4.28: Exemplo de representação de conceitos do ficheiro de dados da aplicação de construção de horários.

Como a análise sintáctica e a respectiva importação são um processo computacionalmente pesado, tornou-se possível o administrador escolher as entidades que deseja importar, respeitando as dependências que existem entre elas, por exemplo, não é possível adicionar um docente de um departamento sem que o departamento exista. Na Figura 4.29 podemos ver a interface utilizada para importação de dados, escolhendo as entidades através da colocação de vistos nas caixas de verificação respectivas.

Figura 4.29: Interface para importação de dados através da análise sintáctica.

Caso um objecto já exista na base de dados, verifica-se se existem alterações, em caso afirmativo, este será actualizado. Esta capacidade é extremamente importante nesta fase, visto ainda ser utilizada a ferramenta de construção de horários. Assim, é possível manter a plataforma actualizada através da importação dos dados provenientes do ficheiro de texto.

Quando executada a acção de importação dos dados, será retornada informação acerca de cada objecto, indicando se foi adicionado, actualizado ou se já estava actualizado na base de dados (Figura 4.30).

» **DISCIPLINA HORAS EM 00 - O DOCENTE ESTÁ ACTUALIZADO!**
 » **DISCIPLINA HORAS EM 00 - O DOCENTE ESTÁ ACTUALIZADO!**

- DISCIPLINAS
ANO: 2009, SEMESTRE: 1º
 » **ART - OS DADOS DA DISCIPLINA FORAM ACTUALIZADOS COM SUCESSO!**
 » **AMS - OS DADOS DA DISCIPLINA FORAM ACTUALIZADOS COM SUCESSO!**

Figura 4.30: Exemplo de feedback retornado pela funcionalidade de importação de dados.

Na importação dos dados do PACO foi utilizada uma metodologia equivalente à acima descrita, mas neste caso foi feita a análise sintáctica de ficheiros Excel.

4.2.4.2 Acções sobre grupos de objectos

Ao longo de toda a aplicação há funcionalidades que podem ser realizadas para um conjunto de objectos ao mesmo tempo, poupando tempo e tornando a execução das tarefas mais simples.

Para implementar esta técnica bastou colocar uma caixa de verificação à esquerda de cada item da lista, como é possível ver na Figura 4.31. Por exemplo, caso o administrador quisesse fechar várias disciplinas de opção, bastava-lhe seleccionar as opções com um visto e executar a acção, sendo retornada uma lista a indicar o sucesso ou insucesso da operação para cada um dos objectos. Em caso de insucesso, será apresentada a descrição detalhada do problema.



Figura 4.31: Exemplo de interface para execução de acções em grupo.

4.2.4.3 Gestão de Inactivos

Muitas vezes na gestão de grandes quantidades de informação, podem ser removidos dados por engano, causando grandes transtornos caso esta acção seja irreversível. Para evitar este tipo situações, na plataforma DSD os dados nunca são removidos, são apenas desactivados, permitindo assim a recuperação dos dados. Porém, esta funcionalidade exigiu a implementação de todo um sistema de controlo de dependências entre conceitos, tanto na desactivação como na recuperação de dados.

Optou-se por informar o administrador das dependências que impedem a desactivação ou recuperação do objecto (Figura 4.32), podendo este resolvê-las através das várias funcionalidades da plataforma.

Principalmente na desactivação, a resolução das dependências pode ser feita de várias formas, por exemplo, ao tentar desactivar um docente que está relacionado com dissertações activas (Figura 4.32), há duas formas para resolver esta situação:

- Desactivar as dissertações relacionadas com o docente;
- Remover a relação do docente com as dissertações.

A melhor estratégia passou por deixar o administrador decidir qual a solução que melhor se adapta ao problema real tendo sido esta a principal razão da não implementação de um sistema para a resolução automática de dependências.

<input type="checkbox"/>	90016	PML	Patrício Miguel Ribeiro Lameirão
<input type="checkbox"/>	90017	SSS	Camel de Sousa Silva

1 2 3 4 5

OPERAÇÕES SOBRE OS ITENS SELECIONADOS:

X Desactivar

DESACTIVAR

DOCENTE: PML | PATRÍCIO MIGUEL RIBEIRO LAMEIRÃO

O DOCENTE NÃO PODE SER DESACTIVADO, EXISTEM AS SEGUINTE DEPENDÊNCIAS:

» HÁ DISSERTAÇÕES ACTIVAS ASSOCIADAS AO DOCENTE

DOCENTE: Y1 | Y1PML |

O DOCENTE FOI DESACTIVADO COM SUCESSO!

Figura 4.32: Exemplo de desactivação de objectos.

Para reactivar os objectos, em vez de existir uma funcionalidade para cada entidade que os permitisse reactivar, optou-se pela criação de uma funcionalidade para gestão dos inactivos de todas as entidades, como podemos ver na Figura 4.33. Assim, basta escolher o tipo de objecto que deseja recuperar e escolher a opção pesquisar, listando todos os objectos inactivos daquela entidade com a opção de recuperação disponível.



Figura 4.33: Interface de recuperação de objectos.

4.2.4.4 Vista geral de docentes e alunos

Na gestão de entidades como docentes e alunos, com o passar do tempo eles vão-se responsabilizando pela execução e cumprimento de diversas tarefas. Muitas vezes é importante ter uma ideia geral de todos os dados que lhe estão associados, sendo impraticável procurar em todas as tarefas a presença do docente ou aluno pretendido.

Neste contexto, são disponibilizadas funcionalidades para ver todos os dados que estão associados a docentes ou alunos.

Na Figura 4.34 pode ver a vista geral de um aluno, apresentado os seguintes dados:

- Dados pessoais;
- Dados do aluno na instituição de ensino;
- Para cada período de inscrição nas dissertações, as dissertações em que o aluno mostrou interesse, indicando o interesse demonstrado e se foi o escolhido para a realizar;
- Para cada período de inscrição das disciplinas de opção, a lista ordenada de opções que constaram na sua inscrição. Para saber o estado da sua colocação, há uma funcionalidade no *Back Office* para ver e alterar inscrições nas disciplinas de opção que apresenta esse tipo de informação.

Relativamente ao docente são apresentados os seguintes dados:

- Dados pessoais;
- Dados do docente relativos à instituição de ensino;
- Para cada período de inscrição nas dissertações, as dissertações em que se encontra envolvido seja como orientador ou co-orientador, apresentando o aluno caso esta tenha sido atribuída;
- Para cada período de inscrição nas disciplinas de opção, as propostas apresentadas para leccionar disciplinas de opção, apresentado informação sobre se foi aceite ou se foi de facto leccionada.

{ VER ALUNO }

1. DADOS PESSOAIS

Nome: [REDAZIDO]
Sexo: M
Nacionalidade: Portugal
E-mail: [REDAZIDO]
Telefone: «Não Tem»

2. DADOS DO ALUNO

Número Mecanográfico: [REDAZIDO]
Curso: MIECT | Mestrado Integrado em Engenharia de Computadores e Telemática
Ranking: 51295153008454
Média: 153
ECTS: 234
Disciplinas em Atraso: Engenharia de Software

3. DISSERTAÇÕES

DISSERTAÇÕES 2008/2009

ID: 65
Nome: [REDAZIDO]
Curso(s): MIEET/MIECT
Orientador: [REDAZIDO]
Co-orientador: -
Data do Interesse: 26-06-2008 15:36:56
Interesse:
Inscrição Aceite: SIM

4. OPÇÕES

OPÇÕES 2008/2009

Número de Opções: 3
1ª Escolha - Redes e Serviços em Imagiologia
2ª Escolha - Sistemas de Tempo Real
3ª Escolha - Modelação e Visualização 3D
4ª Escolha - Especificação, Modelação e Projecto de Sistemas Embutidos
5ª Escolha - Computação Móvel
6ª Escolha - Optimização em Redes de Telecomunicações

OK

Figura 4.34: Vista de todos os dados associados a um aluno.

4.2.4.5 Configuração

Com o passar dos anos há configurações que tem de se adaptar ao ano presente, exigindo-se a renovação de um conjunto de dados que lhe estão associados (mantendo os antigos seguros). Os períodos de inscrições são exemplo disso, sendo necessária a alteração dos períodos de inscrição de ano para ano. Neste caso, cada período de inscrição em dissertações tem as suas dissertações específicas, ou seja, com a alteração do período activo todas as dissertações e demonstrações de interesse são alteradas para as existentes no período escolhido. O mesmo acontece com os períodos de inscrição nas disciplinas de opção.

Os semestres para o preenchimento das *Wish Lists* por parte dos docentes também mudam com o passar dos anos, exigindo-se a adaptação dos mesmos à realidade.

Finalmente, por toda a plataforma são utilizados dados relativos ao servidor actual e a configurações actuais, como o endereço electrónico da página Web e dados do e-mail do administrador. Caso surja a necessidade de alterar estes dados, sem qualquer centralização, seria necessário procurar por toda a ferramenta a existência destes dados e alterá-los. Para precaver este tipo de situações, este tipo de configurações foram guardadas na base de dados da DSD,

desta forma, a alteração dos dados que estão na base de dados afectará todos os locais em que estes são utilizados, sendo feita apenas uma única alteração de dados.

Para cumprir os requisitos acima descritos foi criada uma funcionalidade de configuração da plataforma de simples utilização (Figura 4.35), mascarando toda a complexidade de gestão associada a estes conceitos.

The image shows a configuration interface with four sections, each with a title in green and a form below it:

- 1. DADOS DE CONFIGURAÇÃO NO SERVIDOR**
 - E-mail:
 - Palavra-Chave:
 - SMTP:
 - URL:
- 2. PERÍODO DE INSCRIÇÕES NAS DISSERTAÇÕES ACTIVO**
 - Período de Dissertações:
- 3. PERÍODO DE INSCRIÇÕES NAS OPÇÕES ACTIVO**
 - Período de Opções:
- 4. WISH LISTS ACTIVAS**
 - Wish List [1º Semestre]:
 - Wish List [2º Semestre]:

Figura 4.35: Interface de configuração da plataforma.

4.2.5 Horários

Como referido anteriormente (ver secção 3.1.1.5), nesta versão apenas será possível importar e exportar horários. Assim, para a exportação foi escolhida a utilização de folhas Excel, que oferecem uma base eficaz para a visualização de dados desta natureza.

Na Figura 4.36 podemos ver um trecho de um horário, identificando cada uma das disciplinas com uma cor única. Em cada turma, encontra-se o seu identificador único, a sala em que esta é realizada e o docente que a lecciona.

Além da vista por cursos abaixo representada, a vista por salas também está disponível, permitindo avaliar a sobreposição de turmas.

175	Mestrado					
176						
177	Dia/Hora	8h	9h	10h	11h	12h
178	2ª Feira		SO-P01(104) [ARB]		SO-P04(104) [ACP]	
179			SO-P02(106) [ARB]		SO-P05(106) [ACP]	
180			SO-P03(123) [PLSM]		SO-P06(123) [PLSM]	
181			IIA-T1(ANF_Y) [LSL]		AMS-P5(225) [ICD,JPC]	
182	3ª Feira		SO-P03(123) [RFC]		AC1-TP3(ANF_IV) [MBC]	
183			SO-P10(225) [RFC]		AC1-TP4(ANF_Y) [MBC]	
184				AC1-P3a(104) [JLA]		AC1-P1a(104) [IOU]
185				AC1-P3b(106) [IOU]		AC1-P1b(106) [MBC]
186				AC1-P3c(119) [IOU]		AC1-P1c(119) [AAR]
187				AC1-TP1(ANF_IV) [JLA]		
188			AC1-TP2(ANF_Y) [JLA]			
189	4ª Feira				IIA-OT1(130) [???	
190	5ª Feira		AC1-P4a(104) [AAR]		SO-P14(225) [JMR]	
191			AC1-P4b(106) [AO]		IIA-P2(123) [PLSM]	
192			AC1-P4c(119) [JAF]		FR-P2(220) [RVY]	
193			FR-P1(220) [AM]		AC1-P2a(104) [PL]	
194			IIA-P1(123) [LSL]		AC1-P2b(106) [PL]	
195		SO-P13(225) [RFC]		AC1-P2c(119) [PL]		
196	6ª Feira		FR-P3(220) [AM]		FR-OT1(130) [???	
197						
198						
199		SO	- Sistemas de Operação			
200		IIA	- Introdução à Inteligência Artificial			
201		AMS	- Análise e Modelação de Sistemas			
202		AC1	- Arquitectura de Computadores I			
203		FR	- Fundamentos de Redes			

Figura 4.36: Exemplo da visualização de um horário na folha de Excel gerada.

Como a importação de dados permite a sua actualização, será possível manter sempre todos os horários da plataforma DSD iguais aos existentes na ferramenta de construção.

4.3 Manutenção

A manutenção de software é a “disciplina que visa lidar com as alterações de uma aplicação depois de tornada pública” [49], tendo como principais objectivos [49]:

- **Providenciar continuidade de serviço** – o sistema tem de estar disponível 24h por dia, 7 dias por semana;
- **Suportar melhorias** – faz parte da própria evolução da aplicação;
- **Suportar pedidos de melhorias** – quanto melhor o sistema mais ele será utilizado e mais serão os utilizadores a pedir por melhorias.

Dada a activa interacção dos utilizadores com a aplicação e analisados as razões e objectivos da manutenção, torna-se óbvia a necessidade de manutenção da plataforma DSD.

Alguns procedimentos de manutenção podem levar à inutilização da plataforma, tornando-se importante estudar os períodos de menos utilização da ferramenta. Durante o dia os docentes trabalham e os alunos têm aulas, como a plataforma é parte integrante de tarefas realizadas por ambas as partes, é neste período que a sua utilização é mais acentuada. Para não interromper as tarefas que estes podem estar a elaborar e/ou para não sobrecarregar o sistema, grande parte da manutenção será feita fora de horas, ou seja, durante as primeiras horas do dia.

Foram definidos dois tipos de manutenção:

- **Preventiva** – todas as acções tomadas com vista a evitar a ocorrência de problemas, ou seja, que serão feitas ao longo de toda a existência da aplicação;
- **Correctiva** – todas as acções que tem como objectivo resolver problemas encontrados.

Para cada um deles, serão apresentadas as acções realizadas, bem como alguns exemplos de situações críticas que ocorreram.

4.3.1 Preventiva

Desde o lançamento da ferramenta que foram tomadas opções no sentido de evitar problemas maiores, realçando-se as cópias de segurança e o contacto constante com os utilizadores.

4.3.1.1 Cópias de segurança

Como disse Alex Smolen³⁹, “a única maneira de ter o seu computador seguro, é desligá-lo da tomada” [50]. Ainda que a utilização de um sistema operativo com boas características de segurança, firewall e o SGBD também tenha as suas funcionalidades de segurança (encriptação de dados entre outras), nenhum sistema informático pode ser considerado totalmente seguro, tornando-se fundamental fazer cópias de segurança de todos os dados da aplicação. Assim, através de um sistema automatizado, todos os dias às 5h é feita uma cópia de segurança de toda a base de dados. Além disso, todas as semanas é feita uma cópia das cópias de segurança para outro dispositivo que não fosse o próprio servidor, precavendo situações de falha no hardware de armazenamento.

4.3.1.2 Contacto activo com utilizadores

Desde o lançamento da primeira versão pública da plataforma DSD, houve sempre contacto directo com os seus utilizadores, quer pessoalmente ou através da troca de e-mails. Este contacto foi fundamental para a manutenção da aplicação, auxiliando na detecção de problemas, na melhoria de muitas das funcionalidades já existentes ou até mesmo na implementação de novas funcionalidades propostas.

Também a assistência prestada aos utilizadores foi fundamental, ajudando-os na resolução de pequenos problemas que lhe eram alheios e precavendo a má utilização da aplicação. Como exemplo, quando um aluno não tinha o seu número mecanográfico na base de dados, não se podia registar, exigindo-se a sua adição por parte da administração.

4.3.2 Correctiva

Ao longo da utilização da aplicação DSD surgiram algumas situações críticas, exigindo-se algum esforço para a resolução dos problemas que se apresentaram. Segue-se a análise de algumas dessas situações.

³⁹ Consultor de segurança com conhecimentos em desenvolvimento de software e segurança Web. Página Web pessoal em <http://alexsmolen.com/resume.html>.

4.3.2.1 *Wish Lists*

Logo após o lançamento do novíssimo sistema de recolher os desejos dos docentes (*Wish List*), estes foram informados de que poderiam começar a submeter os seus dados. Pouco tempo depois, começaram a chegar *e-mails* a indicar a existência de problemas, nomeadamente que a lista por eles submetida era totalmente diferente da lista apresentada pelo sistema.

Após a análise de alguns dados fornecidos e o contacto directo com os utilizadores, verificou-se que se tratava de um problema provocado pelo ambiente de multi-utilizador, impossível de detectar com os testes feitos antes do seu lançamento.

Encontrada a origem do problema, em poucas horas foi encontrada a sua resolução, procedendo-se então ao teste num ambiente multi-utilizador para verificar se de facto o problema tinha sido resolvido. Colocaram-se então seis utilizadores externos a efectuar a submissão de *Wish Lists* ao mesmo tempo, ocorrendo tudo sem qualquer percalço. O problema tinha sido resolvido, podendo-se colocar a nova versão à disposição dos utilizadores.

4.3.2.2 **Servidores**

O primeiro servidor utilizado para executar a plataforma DSD era bastante modesto, optando-se pela instalação do Windows XP SP2 com o IIS 5.0 (Tabela 4.1). Inicialmente o seu comportamento foi bastante aceitável, respondendo com sucesso a todos os pedidos dos clientes. No entanto, na altura de inscrição nas disciplinas de opção, o número de utilizadores aumentou consideravelmente, surgindo constantemente a mensagem a indicar que tinha sido atingido o número máximo de utilizadores (Figura 4.37).

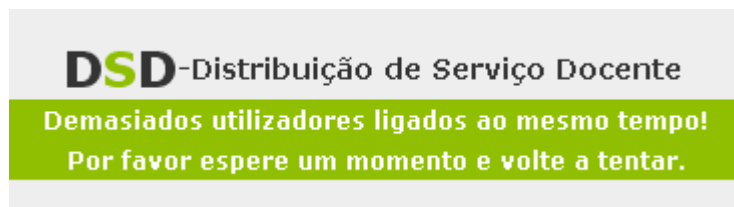


Figura 4.37: Mensagem do servidor a informar que estão demasiados utilizadores ligados.

No entanto, após alguma análise do número de utilizadores ligados e do momento em que a mensagem surgia, concluindo-se que esta era lançada quando estavam apenas dez utilizadores ligados. Com alguma pesquisa, verificou-se que se tratava de uma actualização do Windows XP que limitava o número de clientes que se podiam ligar à máquina [51]. Com a resolução do problema, o número de ligações deixou de estar limitado, no entanto, começaram a vir ao de cima as fragilidades físicas do servidor, deixando de ser capaz de responder a todos os pedidos dos clientes, impossibilitando a utilização da aplicação.

Desta forma, optou-se pela aquisição de um novo servidor com características de hardware ao nível actual (Tabela 4.1). Como toda a instalação e configuração teria de ser feita de novo, optou-se pela instalação de uma versão do Windows própria para servidores, tirando proveito das suas funcionalidades próprias e da nova versão do IIS.

Características/Servidor	1º	2º
Processador	Intel P4 2GHz	Intel Core 2 Duo 3.16 GHz
Memória	756 MB	2 GB
Sistema Operativo	Windows XP SP2	Windows Server 2003 SP2
Versão IIS	5.0	6.0

Tabela 4.1: Comparativo entre os dois servidores utilizados.

Após toda a nova configuração, faltava apenas testar o comportamento do novo servidor com mais de 10 utilizadores. Desta forma, colocaram-se aproximadamente 20 utilizadores a utilizar a plataforma ao mesmo tempo, ocorrendo tudo como o esperado, um desempenho mais eficaz da aplicação e sem qualquer tipo de erros inesperados.

De notar que com o novo servidor chegaram a estar 50 utilizadores ligados ao mesmo tempo sem que houvesse qualquer tipo de problema.

4.3.2.3 Opções

Durante o processo de escolha das disciplinas de opção, para que a justiça das colocações fosse a máxima possível, tentou-se manter os rankings dos alunos o mais actualizados quanto fosse possível. Neste contexto, a certa altura foi necessário importar a lista actualizada de alunos. No entanto, ocorreram problemas na sua importação, consequência da mudança de estratégia de cálculo dos rankings e a não totalidade dos alunos na lista.

Assim, as colocações disponíveis não tinham qualquer sentido, optando-se por cortar o acesso à plataforma, informando os utilizadores do problema ocorrido. Contudo, o acesso local continuava a ser possível, permitindo aos elementos da coordenação pedagógica resolver várias questões:

- Forçar colocações para resolver o problema de alunos não colocados na totalidade das opções que pretendiam;
- Importação da lista completa de alunos com rankings actualizados;
- Verificação manual de alguns rankings.

Após todas as questões resolvidas, a plataforma ficou de novo acessível, garantindo-se ao máximo a justiça das colocações disponibilizadas.

4.4 Sumário

Neste capítulo foi feito um estudo dos cuidados a ter no desenvolvimento de interfaces usáveis, tirando proveito do conhecimento maior sintetizado nas heurísticas de Jakob Nielsen. Posteriormente, é apresentada a estrutura de toda a base interactiva, bem como as características gerais da sua utilização. Após isso, é analisado de que forma as variadas tarefas suportadas são aplicadas à base interactiva definida.

Como consequência da utilização da plataforma num ambiente de teste *"in service"*, são apresentados os cuidados tomados na sua manutenção, estudando alguns problemas críticos que surgiram e a sua resolução.

O *Front Office* da plataforma encontra-se disponível em <http://dsd.av.it.pt> e o *Back Office* em <http://dsd.av.it.pt/backoffice/>.

5 Conclusão

5.1 Resultados

Para avaliar os resultados obtidos com esta plataforma, nada melhor do que passar a palavra aos seus utilizadores [45]. Assim, para obter algum *feedback* acerca da satisfação na utilização da plataforma, foi elaborado um pequeno questionário. A sua construção focou-se nas heurísticas de Jakob Nielsen [45], criando-se um pequeno conjunto de perguntas capaz de resumir os dez grandes princípios de usabilidade, dando uma ideia geral do comportamento da ferramenta segundo o ponto de vista do utilizador.

Para todas as questões obrigatórias foi considerada uma escala de um a cinco, onde um indica Muito Mau e cinco Muito Bom. Foram consideradas quatro questões obrigatórias:

1ª – Acessibilidade: Como avalia a estrutura da página Web?

2ª – Navegação: Com que facilidade navega na página Web?

3ª – Design: Como avalia a interface?

4ª – Conteúdo: Como avalia a disposição e tipo de conteúdos?

Também foi considerada uma questão livre, onde os utilizadores poderiam fazer comentários ou dar sugestões, tendo sido uma das formas de saber quais os problemas com que os utilizadores se enfrentavam.

Como a percentagem de respostas aos questionários enviados por e-mail ronda os 30% [52], a apresentação do questionário foi um pouco diferente, tentando-se obter o número máximo de respostas possível. Assim, todas as vezes que um utilizador entra com a sua conta na plataforma, é proposto o preenchimento do questionário, assim que ele faça, o questionário nunca mais volta a surgir.

O questionário foi apresentado aos utilizadores durante o período “*in service*” da ferramenta, das 299 contas registadas tanto de docentes como de alunos, foram obtidas 213 respostas, o que equivale a 71,23% de respostas.

Em média, os resultados às questões apresentadas foram os seguintes:

- **Acessibilidade:** 3,80
- **Navegação:** 3,61
- **Design:** 3,85
- **Conteúdos:** 3,77

Em valores arredondados dá uma média de 4 valores, o que mostra ser um resultado muito reconfortante para a avaliação de uma ferramenta desta complexidade e com utilizadores tão diferentes:

- **Alunos** – pela sua curiosidade constante, em geral mais atentos aos novos conceitos da internet;
- **Docentes** – pelo grande conjunto de tarefas diárias que têm a cumprir, acabam por disponibilizar menos tempo para estar a par das novidades do mundo Web.

Também de realçar que se trata a primeira versão da plataforma, com estes resultados, mostra-se a sua validade e a sua margem para o progresso, sendo facilmente aceite pelos utilizadores satisfeitos com a actual utilização.

5.2 Conclusões

Nesta secção serão analisados os objectivos atingidos com a plataforma DSD, os conhecimentos aprofundados com a sua elaboração e será feito um balanço geral do estado da plataforma.

5.2.1 Objectivos

Os principais objectivos propostos para este documento foram claramente atingidos:

- **Base Tecnológica** – definição de uma arquitectura tecnológica baseada no alto desempenho e segurança, não esquecendo as suas excelentes características de escalabilidade, que fornecem um forte suporte para a sua extensibilidade;
- **Base Estrutural** – através da análise profunda de todas as tarefas e de todo o contexto envolvente, definiu-se uma base estrutural forte, baseada nas boas regras de modulação e estabelecendo uma forte ligação entre conceitos, abrindo as portas para futuras evoluções;
- **Base Interactiva** – conhecendo os seus utilizadores e baseado em conhecimentos maiores da usabilidade, definiu-se uma base interactiva extremamente organizada e de fácil utilização, dinamizando a realização de todas as tarefas antes complexas;
- **Teste e Manutenção** – com a utilização de toda a plataforma num ambiente “*in service*”, provou-se o seu verdadeiro funcionamento, exigindo apenas algumas tarefas simples de manutenção.

Desta forma, centralizou-se a organização de grande parte das tarefas realizadas no departamento, tendo havido ainda espaço para o desenvolvimento de funcionalidades inovadoras, como o sistema de recolha dos desejos dos docentes (*Wish List*). Assim, com toda a complexa arquitectura definida e com os entusiasmantes resultados obtidos com a sua utilização, esta será certamente uma ferramenta de futuro.

5.2.2 Aprendizagem

Ao longo do desenvolvimento desta complexa plataforma, foi necessário obter conhecimentos profundos acerca de todo o processo de desenvolvimento e publicação de aplicações Web. A nível de linguagens de programação foram obtidos ou aprofundados conhecimentos em HTML, CSS, XML, ASP.NET, ASP.NET AJAX, C# e SQL. Na utilização de ferramentas de desenvolvimento constaram o Microsoft *SQL Server Management Studio* e o

Microsoft Visual Studio. Para o SGBD teve de se explorar profundamente o Microsoft SQL Server. Na publicação da plataforma desenvolvida, exigiu-se a exploração e configuração do sistema operativo Microsoft Windows Server e da aplicação IIS (*Internet Information Services*).

Para além de toda esta panóplia de conhecimentos tecnológicos aprofundados ou adquiridos, houve também uma aprendizagem de interacção social impossível de ignorar, resultado do constante contacto com docentes e alunos.

5.2.3 Balanço Geral

Para fazer um balanço geral de todo o cenário que envolve a plataforma DSD, foi elaborada uma análise SWOT (*Strengths, Weaknesses, Opportunities, Threats*), proporcionando um resumo estruturado das características que favorecem ou não a ferramenta a nível interno e externo [53].



Figura 5.1: Diagrama de análise SWOT.

5.3 Trabalho Futuro

Com o arranque desta nova plataforma, existe um grande conjunto de tarefas que podem ser exploradas e elaboradas, tanto a nível de novas funcionalidades como na utilização de novas técnicas para dinamizar o funcionamento interno da ferramenta.

Nas funcionalidades destacam-se as seguintes oportunidades de trabalho futuro:

- **Controlo automático de quotas dos docentes nas Dissertações** – recentemente foi tomado conhecimento que um docente tem um limite máximo de orientações de dissertações, havendo uma forma de controlar o número de dissertações em que cada docente está envolvido. Assim, nada melhor que um controlo automático para limitar as dissertações que cada docente pode orientar ou co-orientar de forma a não ultrapassar a sua quota;
- **Construção automática e gestão de Horários** – a plataforma desenvolvida já disponibiliza os dados necessários para que se torne possível o desenvolvimento de uma ferramenta para criação automática de horários, havendo pouca intervenção da mão humana. Obviamente que após a geração de horários, esta terá de suportar a alteração manual dos vários dados dos horários.
- **Salas com tipo associado e fotografias** – para uma melhor identificação das salas, nada melhor que a utilização de fotografias. Também de salientar a necessidade de associar cada uma das salas a um tipo, para aulas teóricas, práticas ou teórico-práticas;
- **Definição das possíveis salas para as aulas** – para a construção automática dos horários, é importante saber em que salas cada aula pode ser leccionada. Apesar de já haver alguma informação nesta matéria, com o passar do tempo esta vai-se desactualizando. Assim, seria importante permitir aos docentes que definissem quais as salas que melhor se adaptam às necessidades de cada tipo de aula de uma disciplina;
- **Registo das operações de cada utilizador** – para uma melhor gestão da plataforma, saber as operações realizadas por cada utilizador é um dos melhores pontos de partida, simplificando a resolução de problemas;
- **Mais Informação estatística** – construção de funcionalidades para avaliar a satisfação dos docentes e fornecer mais informações estatísticas, tendo em vista facilitar a distribuição de serviço docente e dar uma ideia geral da distribuição realizada;
- **Notícias dinâmicas e respectiva gestão** – de forma a facilitar o anúncio de notícias na própria plataforma, a inclusão de um sistema para fazer a gestão das notícias é a melhor solução;
- **Transporte de dados inseridos em anos anteriores** – muitas vezes os dados são parcialmente repetidos de ano para ano. Assim, a construção de funcionalidades capazes de copiar a informação relativa a semestres já existentes para um novo semestre parece fundamental. Como exemplo, na inserção de uma *Wish List*, o

docente pode querer copiar a *Wish List* do ano anterior e apenas fazer pequenas alterações;

- **Geração automática de datas de exames** – apesar de ainda não haver suporte, a construção de uma ferramenta para gerar datas de exames de forma automática também se mostra um excelente desafio, levando esta plataforma para territórios nunca antes pensados.

A nível de funcionamento interno, destacam-se as seguintes soluções a ser exploradas:

- **Abstracção do SGBD** – a utilização de uma camada de abstracção do SGBD é sempre uma boa técnica a utilizar, não ficando dependente do gestor de base de dados utilizado.
- **Estudar o modelo de programação de persistência de entidades** – a persistência consiste em reunir as vantagens de utilizar um modelo orientado a objectos na construção de uma aplicação, com o desempenho e confidencialidade das bases de dados actuais. Desta forma, será feito um mapeamento das tabelas da base de dados para a aplicação, deixando de haver acesso directo ao SGBD, utilizando-se apenas os objectos mapeados [20].

Esta metodologia simplifica e dinamiza bastante a construção de funcionalidades, no entanto, há questões de desempenho que precisam de ser estudados.

- **URL Rewriting** – a utilização desta técnica promove a simplificação dos endereços das várias páginas da plataforma. De forma muito simplista, considerando o endereço http://dsd.av.it.pt/lista_pesquisar.aspx?var1=1&var2=2, com a utilização de URL *Rewriting* poderá ficar um endereço bastante mais simples de recordar, <http://dsd.av.it.pt/lista/1/2>. Esta metodologia traz grandes vantagens ao nível da segurança, percepção por parte dos utilizadores e indexação dos motores de busca [54].

Neste momento já está ser feito algum trabalho para dar continuidade à plataforma aqui apresentada, estando em curso a elaboração das seguintes dissertações de mestrado integrado:

- **“Back Office Web: Distribuição de Serviço Docente”⁴⁰** – na qual serão exploradas várias funcionalidades do *Back Office* sendo dada especial atenção à construção da ferramenta de geração automática de horários;
- **“DSD Front Office: Interfaces e interacções”⁴¹** – investigação e desenvolvimento de novas funcionalidades para o *Front e Back Office*.

⁴⁰ http://dsd.av.it.pt/App_Upload/Dissertacoes/2008/08_MI_DSD.pdf

⁴¹ http://dsd.av.it.pt/App_Upload/Dissertacoes/2008/08_MI_DSD_fo.pdf

6 Anexos

Anexo A

Neste anexo pode encontrar a descrição detalhada de todos os atributos de todas as classes que fazem parte do modelo do domínio.

Pessoa

NMec: número Mecanográfico, número único de identificação de uma pessoa na universidade;

Nome: nome da Pessoa;

Sexo: sexo da Pessoa;

Data1: data utilizada para realçar as dissertações novas desde a última visita da pessoa;

Data2: data utilizada para realçar as dissertações novas desde a última visita da pessoa;

Activo: para indicar se a pessoa está activa.

Nacionalidade

Pais: nome do País.

Email

Email: endereço de correio electrónico.

Telefone

Numero: número de Telefone.

Conta

PalavraChave: palavra-chave encriptada;

Codigo: código para activação da conta;

Validada: indica se a conta já foi validade (tornada activa).

Departamento

Codigo: código, valor numérico único que identifica um departamento na universidade.

Nome: nome do Departamento;

Sigla: sigla única que identifica o departamento na universidade;

Activo: para indicar se o Departamento está activo.

Curso

Codigo: código, valor numérico único que identifica o curso na universidade.

Nome: nome do Curso;

Sigla: sigla única que identifica o curso na universidade;

Interesse: são necessários muitos cursos para a construção dos horários, no entanto, não há interesse em que sejam todos visíveis para os utilizadores do *front office*. Desta forma é possível controlar os cursos que têm interesse em ser mostrados;

Activo: para indicar se o curso está activo.

Docente

Sigla: sigla única identificadora do docente na universidade;

Convidado: para indicar se o docente está com o estatuto de convidado.

DocenteAno

Porcentagem: percentagem de horas que o docente irá realizar no semestre associado.

Semestre

Semestre: semestre;

AnoCivil: ano civil do semestre;

Activo: para indicar se o semestre está activo.

DadosOpcao

Media: média de curso do aluno;

Ects: número total de ECTS obtidos;

DisciplinasAtraso: disciplinas que o aluno tem em atraso no momento;

Ranking: ranking oficial do aluno;

Validado: para indicar se os dados adicionados pelo alunos (média e ECTS) dizem respeito à realidade.

Cargo

Nome: nome do Cargo;

Sigla: sigla única identificadora do cargo;

Tipo: para indicar se o cargo está associado a Departamentos ou Cursos;

Activo: para indicar se o cargo está activo.

Docente_Cargo

DataInicio: data em que o docente foi associado ao cargo;

DataFim: data em que o docente deixou de estar associado com o cargo;

IDRelacao: para identificar qual é o Departamento ou Curso a que o docente está associado.

Categoria

Nome: nome da categoria;

Activo: para indicar de a categoria está activa.

Docente_Categoria

DataInicio: data em que o docente foi associado à categoria;

DataFim: data em que o docente deixou de estar associado com a categoria;

Area

Nome: nome da área;

Abreviatura: abreviatura da área;

Activo: para indicar se a área esta activa.

SubArea

Nome: nome da subárea;

Activo: para indicar se a subárea está activa.

Disciplina

Codigo: código, valor numérico único que identifica a disciplina na universidade;

Nome: nome da disciplina;

Sigla: sigla única identificadora da disciplina na universidade;

Externa: para indicar se a disciplina é externa ao departamento;

Comentario: comentário associado à disciplina;

Link: endereço electrónico para a página Web da universidade de Aveiro com informações relacionadas à disciplina;

Activo: para indicar de a disciplina está activa.

Disciplina_Curso

Ano: ano em que a disciplina é leccionada;

Semestre: semestre em que a disciplina é leccionada.

DisciplinaAno_DocenteAno

Regente: para indicar se o docente é regente da disciplina.

Sala

Numero: numero ou sigla que identifica a sala;

Comentario: comentário associado à sala;

Activo: para indicar se a sala está activa.

Aula

Tipo: tipo da aula (T -Teórica, TP -Teórico-prática, P -Prática ou OT -Orientação Tutorial);

Duracao: duracao da aula;

Ocorrencia: número de vezes que a aula ocorre numa semana;

NumTurmas: número de turmas associadas à aula;

Activo: para indicar se a aula está activa.

Turma

Nome: nome da turma;

Vagas: número máximo de alunos permitidos para a turma;

Fixo: para indicar se a turma pode ser alterada;

Activo: para indicar se a turma está activa.

Turma_Sala

Dia: dia da semana em que a turma está associada com a sala;

Horainicio: hora de início da associação;

HoraFim: hora do fim da associação.

Periodo

Nome: nome do período de inscrições;

DataInicio: data em que o período de inscrições se inicia;

DataFim: data em que o período de inscrições termina;

Activo: para indicar se o período de inscrições está activo.

TipoPeriodo

Nome: nome do tipo de período de inscrições.

Wish List

NumDisciplinas: número de disciplinas que o docente pretende leccionar;

d2: disponibilidade (disponível, não disponível o dia todo, não disponível de manhã, não disponível de tarde, não disponível a partir das 18h) do docente no 2º dia da semana;

d2Obrigatorio: para indicar se a disponibilidade indicada para o 2º dia da semana, é obrigatória cumprir;

d3: disponibilidade do docente no 3º dia da semana;

d3Obrigatorio: para indicar se a disponibilidade indicada para o 3º dia da semana, é obrigatória cumprir;

d4: disponibilidade do docente no 4º dia da semana;

d4Obrigatorio: para indicar se a disponibilidade indicada para o 4º dia da semana, é obrigatória cumprir;

d5: disponibilidade do docente no 5º dia da semana;

d5Obrigatorio: para indicar se a disponibilidade indicada para o 5º dia da semana, é obrigatória cumprir;

d6: disponibilidade do docente no 6º dia da semana;

d6Obrigatorio: para indicar se a disponibilidade indicada para o 6º dia da semana, é obrigatória cumprir;

Wish List_DisciplinaAno

Relacao: indica se o docente quer ou não quer leccionar a disciplina;

Ordem: ordem de preferência da disciplina;

T: indica se o docente quer leccionar turmas teóricas relacionadas com a disciplina;

TP: indica se o docente quer leccionar turmas teórico-práticas relacionadas com a disciplina;

P: indica se o docente quer leccionar turmas práticas relacionadas com a disciplina;

OT: indica se o docente quer leccionar turmas de orientação tutorial relacionadas com a disciplina;

Regencia: indica se o docente deseja ser regente da disciplina.

Opcao

Fechada: indica se a opção está fechada para aceitar inscrições.

OpcaoProposta

Ano: ano que o docente propõe que a disciplina seja leccionada;

Semestre: semestre que o docente propõe que a disciplina seja leccionada;

Vagas: número máximo de alunos permitidos que o docente propõe;

Documento: documento PDF com informações detalhadas sobre a opção;

Aceite: indica se a proposta foi aceite pela administração.

OpcaoProposta_DocenteAno

Regente: indica se o docente é regente da opção;

Proposta: indica se o docente fez a proposta.

Dissertacao

Nome: nome da dissertação;

Data: data em que a dissertação foi adicionada na plataforma;

Tipo: tipo da dissertação (normal, evolução ou licenciatura)

Documento: documento PDF com informações detalhadas sobre a opção;

Externos: docentes externos relacionados com a dissertação;

Epoca: época em que é previsto que a dissertação seja entregue (Junho ou Dezembro);

Local: local em que a dissertação será realizada;

Activo: indica se a dissertação está activa.

Dissertacao_DocenteAno

Orientador: indica se o docente é orientador da dissertação.

Inscricao

Data: data em que o aluno efectuou a inscrição;

Activo: para indicar se a inscrição está activa.

InscricaoOpcao

NumOpcoes: número de opções em que o aluno deseja ser colocado;

InscricaoDissertacao

Motivacao: motivação do aluno para querer fazer a dissertação.

InscricaoOpcao_Opcao

Ordem: ordem de preferência da opção na inscrição do aluno;

Forcar: indica se a colocação ou não colocação na opção deverá ser forçada.

Escolha

Ordem: ordem de preferência da escolha de turmas;

Forcar: indica se a colocação ou não colocação na escolha de turmas deverá ser forçada.

DissertacaoAcordo

Docente: indica se o docente já assinou o acordo;

DataDocente: data em que o docente assinou o acordo;

Aluno: indica se o aluno já assinou o acordo;

DataAluno: data em que o aluno assinou o acordo;

Confirmado: indica se o acordo foi confirmado pela administração (para situações em que o acordo é entregue em papel).

Administrador

Utilizador: nome de utilizador do administrador;

PalavraChave: palavra-chave encriptada;

Activo: indica se o administrador está activo.

Acesso_Administrador

DataInicio: data em que o administrador foi associado ao tipo de acesso;

DataFim: data em que o administrador deixou de estar associado ao tipo de acesso.

Acesso_Docente

DataInicio: data em que o docente foi associado ao tipo de acesso;

DataFim: data em que o docente deixou de estar associado ao tipo de acesso.

Acesso

Nome: nome do tipo acesso;

Activo: indica se o acesso está activo;

Gerir*: indica se pode gerir o pacote de conceitos em questão.

Configuracao

Email: endereço de correio electrónico do Administrador, utilizado para enviar e-mails de forma automática;

Pass: palavra-chave associada ao endereço de correio electrónico;

Smtip: endereço SMTP do servidor de e-mail;

URL: endereço electrónico da plataforma;

PeriodoOpcoesActivo: período activo de inscrição nas opções;

PeriodoDissertacoesActivo: período activo de inscrição nas dissertações;

WishListSemestre1: semestre utilizado para o preenchimento da *wish list* do 1º semestre;

WishListSemestre2: semestre utilizado para o preenchimento da *wish list* do 2º semestre.

Questionario

Acessibilidade: resposta à pergunta sobre acessibilidade;

Navegacao: resposta à pergunta sobre navegação;

Design: resposta à pergunta sobre design;

Conteudos: resposta à pergunta sobre os conteúdos;

7 Bibliografia

- [1] GaGI. *Gabinete de Gestão de Informação*. 2008. Disponível em: <http://www2.adm.ua.pt/gagi/>. Acedido em Setembro 21, 2008.
- [2] Instituto Superior Técnico. *Centro de Informática do IST - Projecto Fénix*. 2008. Disponível em: <https://ciist.ist.utl.pt/projectos/fenix.php>. Acedido em Setembro 25, 2008.
- [3] Sitefane JPM. *Desenvolvimento de Software Centrado no Domínio - O caso da Distribuição do Serviço Docente*: Instituto Superior Técnico - Universidade Técnica de Lisboa; 2007. Dissertação.
- [4] Stewart R. *Desktop vs. Browser - when to deploy applications for each*. 2007. Disponível em: <http://blogs.zdnet.com/Stewart/?p=329>. Acedido em Setembro 28, 2008.
- [5] Casarez V, Cripe, B, Sini, J, Weckerle, P. *Reshaping Your Business with Web 2.0*: McGraw-Hill Osborne Media; 2008.
- [6] O'Reilly T. *What Is Web 2.0*. 2005. Disponível em: <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html?page=1>. Acedido em Setembro 8, 2008.
- [7] O'Reilly T. *Web 2.0 Compact Definition: Trying Again*. 2006. Disponível em: <http://radar.oreilly.com/archives/2006/12/web-20-compact.html>. Acedido em Setembro 8, 2008.
- [8] Wikipedia. *Wikipédia em outras línguas*. 2008. Disponível em: http://pt.wikipedia.org/wiki/Wikipedia:Wikip%C3%A9dia_em_outras_l%C3%ADnguas. Acedido em Setembro 5, 2008.
- [9] Netcraft. *August 2008 Web Server Survey*. 2008. Disponível em: http://news.netcraft.com/archives/2008/08/29/august_2008_web_server_survey.html. Acedido em Setembro 25, 2008.
- [10] SecuritySpace. *Web Server Survey*. 2008. Disponível em: http://www.securityspace.com/s_survey/data/200808/index.html. Acedido em Setembro 21, 2008.
- [11] Modadugu N. Google Online Security Blog. *Web Server Software and Malware*. 2007. Disponível em: <http://googleonlinesecurity.blogspot.com/2007/06/web-server-software-and-malware.html>. Acedido em Setembro 25, 2008.
- [12] port80 Software. *Top 1000 Web Servers Survey*. 2008. Disponível em:

- <http://www.port80software.com/surveys/top1000webservers/>. Acedido em Setembro 25, 2008.
- [13] Brown M. *IIS vs. Apache, Looking Beyond the Rhetoric*. 2008. Disponível em: http://www.serverwatch.com/tutorials/article.php/10825_3074841_1. Acedido em Setembro 25, 2008.
- [14] Hiner J. *The Web server software choice: iPlanet vs. Apache vs. IIS*. 2000. Disponível em: http://articles.techrepublic.com.com/5100-10878_11-1034514.html. Acedido em Setembro 25, 2008.
- [15] Bowen R, Coar, K. *Apache Cookbook*: O'Reilly; 2007.
- [16] Volodarsky M, Londer, O, Hill, B, et al. *Internet Information Services (IIS) 7.0*: Microsoft Press; 2008.
- [17] Damas L. *SQL*: FCA; 2005.
- [18] Ricardo CM. *Databases Illuminated*: Jones & Bartlett Publishers; 2004.
- [19] Transaction Processing Performance Council. *TPC-C*. 2008. Disponível em: <http://www.tpc.org/tpcc/default.asp>. Acedido em Setembro 25, 2008.
- [20] Beauchemin B, Berglund, N, Sullivan, D. *A First Look at SQL Server 2005 for Developers*: Addison-Wesley; 2004.
- [21] Otey M. SQL Server 2005 Replication. *SQL Server Magazine*. Fevereiro 2007.
- [22] Microsoft Corporation. *Sql Server Express Datasheet*. 2006. Disponível em: <http://download.microsoft.com/download/4/2/c/42c6af76-a28c-4560-8269-e8fe906ea675/SQLExpressdatasheet.pdf>. Acedido em Outubro 4, 2008.
- [23] Oracle. Oracle® Database Concepts. *SQL, PL/SQL, and Java*. 2005. Disponível em: http://download.oracle.com/docs/cd/B19306_01/server.102/b14220/sqlplsql.htm. Acedido em Outubro 4, 2008.
- [24] Oracle. *Oracle Database 10g Express Edition Datasheet*. 2008. Disponível em: http://www.oracle.com/technology/products/database/xe/pdf/dbxe_datasheet.pdf. Acedido em Outubro 2, 2008.
- [25] MySQL AB. *MySQL 5.0 Now Available for Production Use*. 2005. Disponível em: <http://www.mysql.com/news-and-events/generate-article.php?id=976>. Acedido em Outubro 2, 2008.
- [26] Converse T, Park, J, Morgan, C. *PHP5 and MySQL Bible*: Wiley Publishing, Inc.; 2004.
- [27] MacDonald M. *Beginning ASP.NET 2.0 in C# 2005: From Novice to Professional*: Apress; 2006.
- [28] *Mono*. 2008. Disponível em: http://www.mono-project.com/Main_Page. Acedido em Setembro 1, 2008.
- [29] PHP. *History of PHP and related projects*. 2008. Disponível em: <http://pt.php.net/history>. Acedido em Agosto 27, 2008.
- [30] Hull S. *PHP vs. ASP.NET Redux*. 2008. Disponível em: http://www.oracle.com/technology/pub/columns/hull_php2.html. Acedido em Setembro 21,

2008.

- [31] Bergsten H. *JavaServer Pages*. 3 ed: O'Reilly; 2003.
- [32] Wang D. *How does JSP work on IIS?* 2005. Disponível em: <http://blogs.msdn.com/david.wang/archive/2005/10/11/How-does-JSP-work-on-IIS.aspx>. Acedido em Setembro 7, 2008.
- [33] HTML XaCB3E. *Brian Pfaffenberger, Steven M. Schafer, Charles White, Bill Karow*: Wiley Publishing, Inc.; 2004.
- [34] Seara.com. *Estudos de Usabilidade*. 2007. Disponível em: <http://seara.com/gca/?id=24>. Acedido em Setembro 12, 2008.
- [35] Chris Ullman LD. *Beginning Ajax*: Wrox; 2007.
- [36] Mahemoff M. *Ajax Design Patterns*: O'Reilly; 2006.
- [37] Adobe. *Flex 3*. 2008. Disponível em: <http://www.adobe.com/br/products/flex/overview/>. Acedido em Setembro 12, 2008.
- [38] Elias H. *Inverse Kinematics*. 1998. Disponível em: http://freespace.virgin.net/hugo.elias/models/m_ik.htm. Acedido em Outubro 2, 2008.
- [39] Wenz C. *Essential Silverlight*: O'Reilly Media, Inc; 2007.
- [40] Mono Project. *Moonlight*. 2008. Disponível em: <http://www.mono-project.com/Moonlight>. Acedido em Setembro 25, 2008.
- [41] Microsoft Corporation. *Silverlight Developer Center: Supported Operating Systems and Browsers*. 2008. Disponível em: [http://msdn.microsoft.com/en-us/library/dd229102\(VS.95\).aspx](http://msdn.microsoft.com/en-us/library/dd229102(VS.95).aspx). Acedido em Outubro 4, 2008.
- [42] Microsoft Corporation. *Microsoft Releases Silverlight 2*. 2008. Disponível em: <http://www.microsoft.com/presspass/press/2008/oct08/10-13Silverlight2PR.mspx>. Acedido em Outubro 2, 2008.
- [43] Fowler M, Rice, D, Foemmel, M, Hieatt, E, Mee, R, Stafford, R. *Patterns of Enterprise Application Architecture*: Addison Wesley; 2002.
- [44] Fowler M. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*: Addison Wesley; 2003.
- [45] Nielsen J. *Usability Engineering*: Morgan Kaufmann; 1993.
- [46] W3Schools. *Browser Display Statistics*. 2008. Disponível em: http://www.w3schools.com/browsers/browsers_display.asp. Acedido em Setembro 28, 2008.
- [47] Avery J, Holmes, J. *Windows Developer Power Tools*. 1 ed: O'Reilly Media, Inc.; 2007.
- [48] Mares CA. *CarlosAg.ExcelXmlWriter Help*. 2008. Disponível em: <http://www.carlosag.net/downloads.aspx?file=CarlosAg.ExcelXmlWriter.Help.zip>. Acedido em Outubro 5, 2008.
- [49] Grubb P, Takang, AA. *Software Maintenance: Concepts and Practice*. 2 ed: World Scientific

Publishing Co. Pte. Ltd.; 2003.

- [50] Lino M. PC's vão acabar? *Expresso*. Novembro 2008.
- [51] Microsoft. *Microsoft Help and Support*. 2008. Disponível em:
<http://support.microsoft.com/kb/314882>. Acedido em Outubro 5, 2008.
- [52] Sekaran U. *Research Methods for Business - A Skill-building Approach*. 3rd Edition ed: Jon Wiley & Sons, Inc.; 2000.
- [53] Wehrich H. The TOWS Matrix: A Tool for Situational Analysis. *Journal of Long*. 1982; 15(2).
- [54] Yack D, Mayo, J, Hanselman, S, et al. *ASP.NET 2.0 MVP Hacks and Tips*: Wiley Publishing, Inc.; 2006.
- [55] Mitchell S, Mack, D, Walther, S, et al. *ASP.NET: Tips, Tutorials and Code*: SAMS; 2001.
- [56] Bondi AB. Characteristics of Scalability and Their Impact on Performance. Paper presented at: Proceedings of the 2nd international workshop on Software and performance, 2000; Ottawa, Ontario, Canada.