

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

Análise e desenvolvimento de módulos de gestão de informação para plataforma de gestão da inovação

Nuno André Trindade Marinho

Mestrado Integrado em Engenharia Informática e Computação

Orientador: José António Faria (Professor Doutor)

Co-orientador: Gil Gonçalves (Professor Doutor)

20 de Julho de 2011

Análise e desenvolvimento de módulos de gestão de informação para plataforma de gestão da inovação

Nuno André Trindade Marinho

Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo júri:

Presidente: Nome do Presidente (Título)

Vogal Externo: Nome do Arguente (Título)

Orientador: José António Faria (Professor Doutor)

31 de Julho de 2010

Resumo

Todos os dias as pessoas observam inúmeros dados sendo uns mais relevantes que outros. A partir desses dados o nosso cérebro efectua uma filtragem dos mais relevantes e produz informação sobre os mesmos, essa informação pode constituir conhecimento, dada a sua maior ou menor utilização. Esta é uma definição muito sucinta da relação existente entre dados, informação e conhecimento. Neste contexto surgem os trabalhadores que baseiam o seu trabalho no conhecimento e experiência, e a necessidade de sistemas de suporte ao trabalho dos mesmos.

O trabalho baseado no conhecimento está a assumir uma importância muito grande nas organizações, tanto em valor como em número, pois os trabalhadores que baseiam o seu trabalho na sua experiência criam muito mais valor para as organizações que qualquer outro tipo de trabalhadores e o seu número nas organizações está a aumentar. Este tipo de pessoas têm características muito específicas, sendo por isso muito difícil criar uma plataforma que se adapte às reais necessidades destes.

A plataforma deve ter como principais características, a usabilidade, pois se esta não for facilitada esse tipo de pessoas não utilizará a plataforma e permanecerá nos seus métodos; e o suporte a diferentes tipos de processos de trabalho, pois o trabalho baseado no conhecimento tem como característica a possibilidade de utilização de diferentes tipos de processos, estruturados, semi-estruturados e não estruturados.

O sistema deverá suportar uma componente de gestão de conteúdos de informação, onde o principal objectivo é a unificação de toda a informação. A unificação dos recursos de informação consiste em tratar todos os recursos associados a um caso de igual forma, *e-mails*, dados em diferentes bases de dados e documentos vistos apenas como recursos de informação. A dissertação aqui exposta consiste em desenvolver os módulos de suporte a gestão de conteúdos de informação.

O projecto consistiu no desenvolvimento de dois módulos de gestão de informação. O módulo Explorer consiste numa espécie de explorador do *Windows*, sobre uma pasta numa biblioteca em *SharePoint*. O módulo Viewer consiste num visualizador de relações entre instâncias de entidades relacionadas.

Os dois módulos de gestão de informação foram desenvolvidos em *C#* sobre a plataforma *SharePoint*.

As plataformas de desenvolvimento utilizadas no desenvolvimento da dissertação foram o *Microsoft Office SharePoint Designer 2007* e o *Visual Studio 2010*.

Abstract

Every day people watch a lot of data is more relevant than others. From these data our brain performs a filtering and produces more relevant information about them, this information may constitute knowledge, given their greater or lesser use. This is a very succinct definition of the relationship between data, information and knowledge. In this context, emerged workers who base their work on knowledge and experience, and the need to support systems work the same.

The knowledge-based work is assuming great importance in organizations, both in value and number, because workers who base their work on their experience to create value more organizations than any other type of workers and their number in organizations is increasing. Such people have very specific characteristics, so it is very difficult to create platform that suits the real needs of these.

The platform must have its main features, usability, because if this is not facilitated such people will not use the platform and remain in their methods; and support for different types of work processes, because knowledge work is characterized the possibility of using different types of processes, structured, semi-structured and unstructured.

The system should support a management component of information content, where the main objective is the unification of all information. The unification of information resources is to treat all resources associated with the event of a similarly, email, data in different databases and documents viewed solely as information resources. The thesis presented here is to develop modules to support content management information.

The project consist in the development of two modules of information management. The module Explorer is a kind of *Windows* explorer on a folder in a library in *SharePoint*. The module Viewer is a viewer of relations between instances of related entities.

The two modules of information management have been developed in *C#* on the platform *SharePoint*.

Development platforms used in the development of the dissertation were the *Microsoft Office SharePoint Designer 2007* and *Visual Studio 2010*.

Agradecimentos

Apesar da natureza individual deste trabalho, vários foram aqueles que contribuíram, para a sua concretização e sem o contributo dos quais não seriam possíveis os resultados alcançados.

Cabe-me agradecer a todos a preciosa colaboração, não podendo no entanto, deixar de expressar a minha gratidão particular àqueles cuja orientação, apoio e motivação, assumiu uma importância fundamental no seu desenvolvimento.

Muito particularmente, ao Professor Doutor José António Faria, meu orientador, por todo o apoio, orientação e incentivo constantes ao longo deste trabalho. Aos engenheiros Hélder Marques e João Silva, pelo permanente interesse, disponibilidade e apoio demonstrado ao longo de todo o período de dissertação.

À Faculdade de Engenharia da Universidade do Porto por me ter acolhido.

A toda a minha família, particularmente à minha mãe, que me apoiou e estimulou durante todos estes anos de estudo e dedicação.

À minha namorada, Raquel, por toda a compreensão, apoio e estímulo demonstrados durante a realização deste projecto.

Aos meus amigos, por todo o apoio e paciência apresentados durante todo o curso.

A todos o meu muito obrigado!

Nuno Marinho

“New industries differ from the traditional industry in that they will employ predominantly knowledge workers rather than manual workers.”

Peter Drucker

Conteúdo

1	Introdução	1
1.1	Contexto/Enquadramento	1
1.2	Plataforma de gestão de inovação	2
1.2.1	Gestor de Processos	3
1.2.2	Gestor de informação	4
1.2.3	Gestor de Colaboração	6
1.3	Objectivos	6
1.4	Metodologia	7
1.5	Estrutura do documento	8
2	Sistemas de trabalho baseados em conhecimento intensivo	9
2.1	Trabalho do conhecimento	9
2.1.1	Conhecimento	9
2.1.2	Gestão do conhecimento	10
2.1.3	Trabalho baseado no conhecimento	11
2.2	Processos de trabalho	13
2.2.1	Processos de trabalho baseado no conhecimento	13
2.2.2	Sistema de suporte ao trabalho baseado no conhecimento	14
2.2.3	Síntese	15
2.3	Comparação de soluções existentes	16
2.3.1	Resumo	25
3	Módulo Explorer+	29
3.1	Exemplo de aplicação módulo: Página projectos	30
3.1.1	Arquitectura página projectos	34
3.1.2	Modelo de dados página projectos	34
3.1.3	Estrutura do <i>dossier</i> projectos	35
3.2	Análise requisitos	36
3.2.1	Requisitos não funcionais	37
3.2.2	Requisitos funcionais	38
3.2.3	Casos de utilização	40
3.3	Desenho do módulo	41
3.3.1	Arquitectura	41
3.3.2	Modelo de dados	43
3.4	Implementação	43

CONTEÚDO

4	Módulo Viewer	49
4.1	Exemplo de aplicação módulo: Página projectos	50
4.2	Análise requisitos	54
4.2.1	Requisitos não funcionais	54
4.2.2	Requisitos funcionais	54
4.2.3	Casos de utilização	56
4.3	Modelo de dados e arquitectura	57
4.3.1	Modelo de dados	57
4.3.2	Arquitectura	58
4.4	Tratamento de eventos	61
4.5	Implementação	68
5	Conclusões e trabalho futuro	81
5.1	Revisão	81
5.2	Avaliação final do trabalho	82
5.3	Trabalho Futuro	83
	Referências	85
A	Controlo ExplorerMais	87
A.1	Página Projectos	87
A.1.1	Diagrama casos de uso Página Projectos	87
A.1.2	Estrutura pastas projectos	88
A.1.3	Estrutura pastas projectos e atributos <i>Managed Folder</i>	89
A.1.4	Tabelas descrição dos casos de uso	90
B	Controlo Viewer	95
B.1	Base de dados completa Viewer	95
B.2	Configuração <i>WebPart</i> com controlos Viewer	96
B.3	Evento <i>Load</i>	97
B.4	Evento <i>Click</i>	98
B.5	Evento <i>Drag and Drop</i>	99

Lista de Figuras

1.1	Arquitectura plataforma de gestão de inovação.	2
1.2	Áreas de um projecto IDI.	3
1.3	Mapa de processos de gestão de inovação	4
1.4	Modelo de domínio.	5
1.5	Esquema entidades, instâncias e dossiers associados.	6
1.6	Modelo de desenvolvimento em espiral.	7
2.1	Pirâmide do conhecimento.	10
2.2	Sharepoint.	16
2.3	Gráfico de respostas a "Aplicações do SharePoint?"	17
2.4	Gráfico de respostas a "Quais dos seguintes sistemas deseja integrar com o SharePoint?"	18
2.5	FileDoc.	18
2.6	Módulos do File Doc.	19
2.7	KnowledgeTree.	20
2.8	iPortalDoc.	21
2.9	Document Locator.	22
2.10	eDocLink.	23
2.11	Alfresco.	24
2.12	DropBox.	25
3.1	Módulo Explorer+.	29
3.2	Página projecto com o módulo Explorer+	30
3.3	Página projecto <i>Aventura</i>	31
3.4	Adicionar assunto a projecto	31
3.5	Adicionar reunião a projecto	32
3.6	Adicionar relatório a projecto	32
3.7	Gestão de conteúdos de <i>dossier</i>	33
3.8	Arquitectura página projectos	34
3.9	Modelo de dados projectos.	35
3.10	Diagrama casos do módulo Explorer+.	40
3.11	Menu de contexto do módulo Explorer+ sorbe uma pasta	41
3.12	Arquitectura do módulo Explorer+	42
3.13	Modelo de dados do módulo Explorer+.	43
3.14	Estrutura do nome dos ficheiros em pasta versionada.	44
3.15	Exemplo de uma nota em XML.	45
3.16	Estrutura do nome dos ficheiros nota.	45

LISTA DE FIGURAS

3.17	Exemplo de uma ligação em XML.	45
3.18	Estrutura do nome dos ficheiros ligação	46
3.19	Exemplo de acção explorar sobre pasta no módulo Explorer+	47
4.1	Conteúdos de instâncias relacionadas com a página	49
4.2	Entidades relacionadas com projectos	50
4.3	Instâncias relacionadas	51
4.4	<i>Dossier</i> associado à instância	51
4.5	Diferentes repositórios de informação	52
4.6	Relacionamento entre configuração, dados e gestores de entidade	53
4.7	Diagrama casos do módulo Viewer.	56
4.8	Modelo de dados do módulo Viewer	57
4.9	Arquitectura em camadas utilizada pelo módulo Viewer.	59
4.10	Modelo de dados <i>ViewerNodes</i>	60
4.11	Diagrama de sequência evento <i>Load</i>	62
4.12	Esquema representativo do tratamento do evento <i>load</i>	63
4.13	Diagrama de sequência evento <i>Click</i>	64
4.14	Esquema representativo do tratamento do evento <i>Click</i>	65
4.15	Diagrama de sequência evento <i>Drag and Drop</i>	66
4.16	Esquema representativo do tratamento do evento <i>Drag and Drop</i>	67
4.17	Menu de contexto disponibilizado na configuração <i>in place</i> do Viewer	69
4.18	Tipo de nós passíveis de adicionar ao Viewer	70
4.19	Adicionar nó virtual ao Viewer	71
4.20	Adicionar nó entidade ao Viewer	72
4.21	Estrutura <i>queries</i>	73
4.22	Protótipo de interface para representar um configurador de <i>queries</i>	74
4.23	Modelo de abstracção na especificação das <i>queries</i> por parte do utilizador	75
4.24	Relacionamento entre tabelas sites, bibliotecas e pastas de topo	77
4.25	Adicionar nó pasta ao Viewer	78
4.26	Funcionalidade de pesquisa e filtragem de conteúdos	79
A.1	Diagrama casos de uso página projecto	87
A.2	Estrutura de pastas para projectos	88
A.3	Estrutura de pastas para projectos e atributos <i>Managed Folder</i>	89
B.1	Base de dados completa do controlo Viewer.	95
B.2	Controlo Viewer	96
B.3	Controlo Viewer - botão modo edição	96
B.4	Controlo Viewer em modo edição	96
B.5	Diagrama de actividade evento <i>Load</i>	97
B.6	Diagrama de actividade evento <i>Click</i>	98
B.7	Diagrama de actividade evento <i>Drag and Drop</i>	99

Lista de Tabelas

3.1	Campos adicionais <i>Managed Folder</i>	36
3.2	Valores campos <i>Managed Folder</i> quando pasta do tipo <i>Folder</i>	36
4.1	Exemplo nó virtual na tabela em variável de sessão (<i>RuntimeNodes</i>) . . .	71
4.2	Exemplo nó entidade na tabela em variável de sessão (<i>RuntimeNodes</i>) . .	72
4.3	Exemplo nó <i>Query</i> na tabela em variável de sessão (<i>RuntimeNodes</i>) . . .	75
4.4	Exemplo nó instância na tabela em variável de sessão (<i>RuntimeNodes</i>) . .	76
4.5	Exemplo nó ficheiro na tabela em variável de sessão (<i>RuntimeNodes</i>) . .	76
4.6	Exemplo nó pasta na tabela em variável de sessão (<i>RuntimeNodes</i>)	77
A.1	Caso de uso <i>Drag and Drop</i> de ficheiros e pastas	90
A.2	Caso de uso adicionar nota	90
A.3	Caso de uso editar nota	91
A.4	Caso de uso eliminar nota	91
A.5	Caso de uso adicionar ligação	91
A.6	Caso de uso editar ligação	92
A.7	Caso de uso eliminar ligação	92
A.8	Caso de uso adicionar pasta	92
A.9	Caso de uso eliminar pasta	93
A.10	Caso de uso adicionar pasta versionada	93
A.11	Caso de uso eliminar pasta versionada	94
B.1	Caso de uso abrir ficheiro	100
B.2	Caso de uso expandir nó	100
B.3	Caso de uso abrir explorador	100
B.4	Caso de uso navegar para instância	101
B.5	Caso de uso adicionar nó	101
B.6	Caso de uso eliminar nó	101
B.7	Caso de uso editar nó	102
B.8	Caso de uso <i>Drag and Drop</i> entre ficheiros e pastas	102

LISTA DE TABELAS

Abreviaturas e Símbolos

BPM	Business Process Management
ACM	Adaptative Case Management
KWs	Knowlegde Workers
ECM	Enterprise Content Management
AIIM	Association for Information and Image Management
SP	SharePoint
C#	C Sharp
SP++	Plataforma SharePoint++
SP++1.0	Plataforma SharePoint++ Versão 1.0
MS	Microsoft
TI	Tecnologia de Informação
IDI	Sistemas de Gestão de Investigação, Desenvolvimento e Inovação
SQL	Structured Query Language

ABREVIATURAS E SÍMBOLOS

Capítulo 1

Introdução

Neste capítulo são apresentados o contexto em que se desenvolveu esta dissertação, os objectivos e as motivações da mesma. Também é apresentado o método de desenvolvimento utilizado para a sua realização e a estrutura do documento.

1.1 Contexto/Enquadramento

O trabalho apresentado neste documento foi desenvolvido no âmbito do projecto de investigação e desenvolvimento, designado por *I9Source*. Este projecto *I9Source* tem por objectivos fundamentais o desenvolvimento de:

- Uma metodologia de análise e síntese de sistemas de gestão da inovação;
- Uma plataforma de *software* para suporte às actividades de gestão da inovação.

As actividades de inovação, necessitam de sistemas de trabalho baseados em conhecimento intensivo, uma vez que inovar é criar novo conhecimento e tipicamente este novo conhecimento é criado utilizando conhecimento existente. Os processos de criação de conhecimento são processos pouco estruturados, tanto ao nível de fluxo de trabalho como ao nível da informação que lhes está associada. Por isso, neste contexto a gestão de informação associada a estes processos é particularmente complexa. A informação, associada a estes processos, pode ser muito diversa, tanto na origem como no tipo de informação.

A informação é tão importante que a gestão moderna exige o máximo de informação possível a cada tomada de decisão, tal como ilustrado na afirmação seguinte.

"A informação tornou-se uma necessidade crescente para qualquer sector da actividade humana e é-lhe indispensável mesmo que a sua procura não seja ordenada ou sistemática, mas resultante apenas de decisões casuísticas e/ou

intuitivas. Uma empresa em actividade é, por natureza, um sistema aberto e interactivo suportado por uma rede de processos articulados, onde os canais de comunicação existentes dentro da empresa e entre esta e o seu meio envolvente são irrigados por informação." [Bra99, cap. Informação]

Também a utilização do conhecimento intrínseco de cada trabalhador a cada tomada de decisão, alheado da informação que o rodeia deixou de ser suficiente, como refere o mesmo autor.

"O conhecimento adquirido *know how* deixa de ser suficiente, uma vez que o meio ambiente empresarial onde as empresas operam apresenta características diferentes daquelas a que estavam habituados e é bastante turbulento. Se em ambientes mais estáveis a informação assumia o papel de redutora de incerteza, cada vez mais a actualização se apresenta como um factor crítico de sucesso." [Bra99, cap. Informação]

Analisando estas afirmações, percebe-se a necessidade de existir um sistema de gestão de informação incorporado para ajudar o trabalhador do conhecimento a cada tomada de decisão. Neste contexto, podemos concluir que uma plataforma para gerir o trabalho baseado no conhecimento necessita de um gestor de informação.

1.2 Plataforma de gestão de inovação

Conforme já referido, o projecto *I9Source* tem por objectivos desenvolver uma metodologia de análise e síntese de sistemas de gestão da inovação e uma plataforma de suporte às actividades de gestão da inovação.

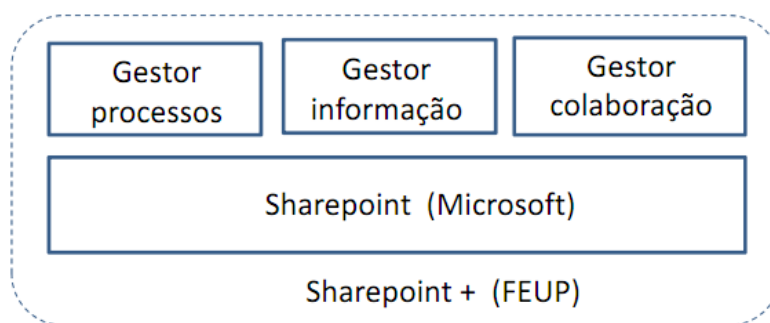


Figura 1.1: Arquitectura plataforma de gestão de inovação.

Esta plataforma oferece três grandes conjuntos de funcionalidades (como apresenta a figura 1.1) :

- Gestão de processos;

- Gestão de informação;
- Gestão de comunicação ou colaboração.

Os módulos a desenvolver inserem-se na gestão da informação.

1.2.1 Gestor de Processos

A plataforma suporta um conjunto de módulos para gerir processos de trabalho associados a um sistema de gestão de inovação.

A plataforma *I9Source* engloba diversas áreas de acção (apresentadas na figura 1.2), mas numa primeira fase, foi dada prioridade à área de suporte aos processos de vigilância, gestão de projectos IDI e à base de conhecimento.

A figura 1.2 representa as principais áreas de acção de um sistema de gestão de inovação.

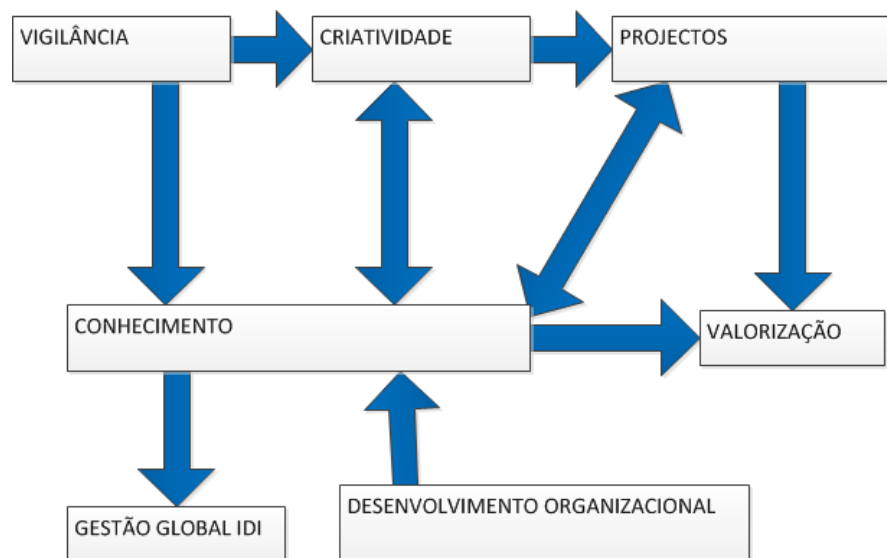


Figura 1.2: Áreas de um projecto IDI.

Introdução

Os principais processos associados a cada uma das áreas de actividade, de um sistema de gestão de inovação, é apresentado na figura 1.3.

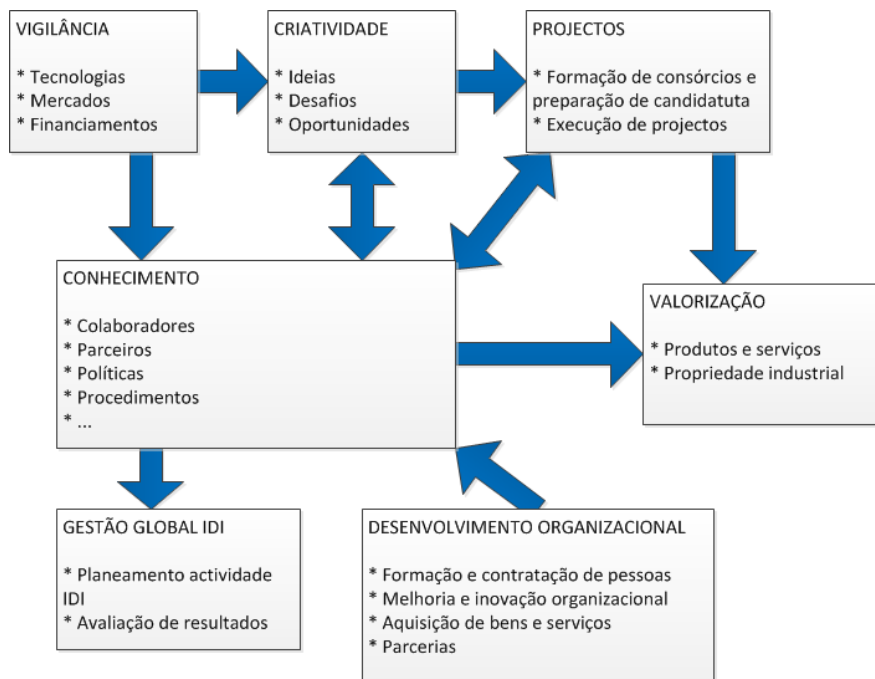


Figura 1.3: Mapa de processos de gestão de inovação

1.2.2 Gestor de informação

A figura 1.4 representa o modelo das principais entidades do domínio de aplicação gerido na plataforma *I9Source*, segundo as quais está organizada a informação.

Para cada instância pertencente as principais entidades do domínio de aplicação, existe um *dossier* no qual são arquivados os diversos tipos de conteúdos relativos a essa instância.

Um *dossier* associado a uma instância contém vários tipos de conteúdos, como por exemplo:

- ficheiros e pastas;
- *e-mails*;
- notas;
- ligações.

Por exemplo, no *dossier* de um projecto serão arquivados os documentos de gestão (contrato, acordo de consórcio, relatórios de execução financeira), os documentos técnicos, as actas das reuniões, as mensagens de correio electrónico trocadas dentro da equipa

Introdução

de projecto e com entidades externas, as actas de reuniões, anotações diversas e hiperligações úteis.

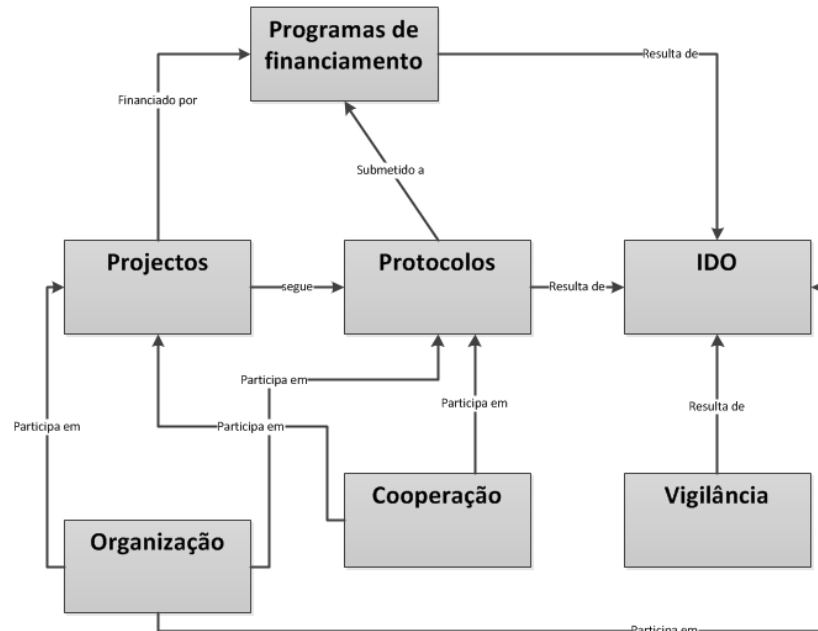


Figura 1.4: Modelo de domínio.

Pretende-se que a página de cada instância, além das funcionalidades ao nível da gestão dos processos, ofereça um acesso expedito a toda a informação relevante contida não apenas no *dossier* da própria instância, mas também nos *dossiers* das outras instâncias com que está relacionada. Daqui surge a necessidade de integrar no sistema de gestão de inovação componentes de gestão de informação que:

- Apresentem o *dossier* da instância em visualização e que permitam gerir o conteúdo do mesmo, e que será designado por Explorer+;
- Permitam aceder de forma expedita às instâncias associadas e ao conteúdo dos respectivos *dossiers*, e que será designado por Viewer.

Introdução

A figura 1.5 esquematiza o relacionamento entre entidades, instancias e os *dossiers* associados.

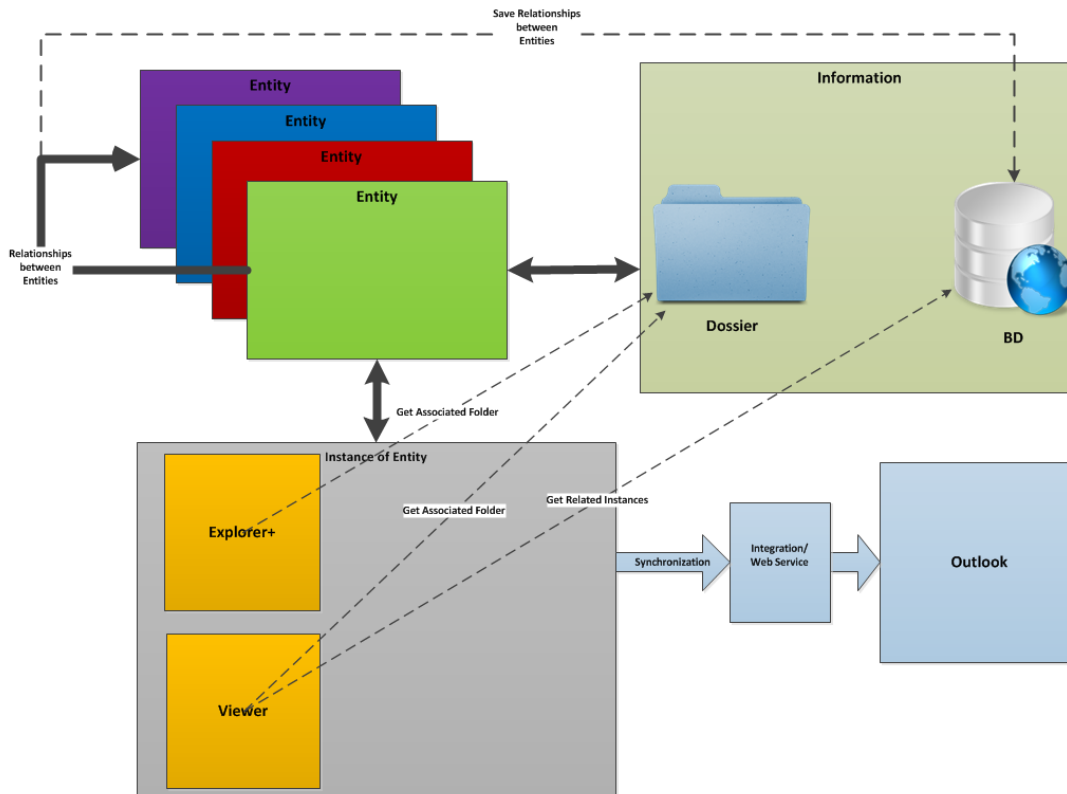


Figura 1.5: Esquema entidades, instâncias e dossiers associados.

1.2.3 Gestor de Colaboração

A plataforma inclui ainda um conjunto de módulos para gerir a colaboração entre utilizadores. Os módulos de colaboração disponibilizam ferramentas de comunicação e partilha de conhecimento entre colaboradores. Por exemplo, placares de notas, salas de conversação ou fóruns.

1.3 Objectivos

O trabalho apresentado neste documento teve por objectivos específicos a análise, especificação e implementação de dois dos módulos de gestão da informação. Os dois módulos a desenvolver serão integrados na plataforma de gestão de inovação, que está actualmente a ser desenvolvida no âmbito do projecto *I9Source*.

O primeiro desses módulos consiste num gestor de conteúdos organizado em pastas hierárquicas e oferece uma experiência de utilização próxima dos gestores de ficheiros convencionais como, por exemplo, o *Windows Explorer*¹. No entanto, e relativamente a estes gestores, o módulo desenvolvido oferece um conjunto de funcionalidades alargado, nomeadamente a gestão de versões de ficheiros, um esquema flexível de gestão de permissões, a integração de diferentes tipos de conteúdos e a possibilidade de especificação de *templates* para os repositórios documentais. A análise detalhada deste módulo será apresentada no capítulo 3.

O segundo módulo consiste num visualizador de conteúdos que agrega numa única interface conteúdos relativos a diferentes entidades da aplicação e cuja análise detalhada será apresentada no capítulo 4.

1.4 Metodologia

A metodologia de desenvolvimento utilizada no desenvolvimento dos módulos de gestão de informação foi o modelo em espiral², como está representado na figura seguinte.

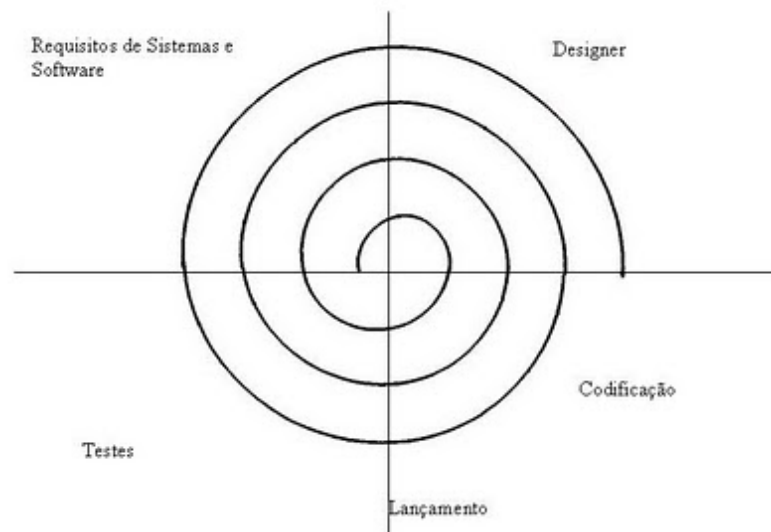


Figura 1.6: Modelo de desenvolvimento em espiral.

Este modelo de desenvolvimento consiste em desenvolver consecutivamente novas versões, sendo que as novas versões são desenvolvidas sobre as anteriores adicionando novas funcionalidades. É um modelo de natureza iterativa e incremental.

O desenvolvimento de cada versão segue um conjunto de fases. A fase de análise de requisitos serve para identificar novas funcionalidades a adicionar a cada versão. Na fase

¹http://en.wikipedia.org/wiki/Windows_Explorer

²http://en.wikipedia.org/wiki/Spiral_model

de desenho todo o sistema modelado e com especial foco sobre as novas funcionalidades identificadas na análise de requisitos. Depois, segue-se a codificação. Para terminar, lançamento em produção piloto para respectivo teste das funcionalidades adicionadas.

O ciclo de vida no desenvolvimento de uma versão tem as seguintes fases:

1. Análise requisitos
2. Desenho
3. Codificação
4. Lançamento em produção (protótipo)
5. Teste em produção

1.5 Estrutura do documento

Este documento está organizado em seis capítulos, o primeiro dos quais é a introdução.

O segundo capítulo expõe as definições de trabalho do conhecimento e de processos de trabalho, com especial atenção sobre a gestão da informação. São apresentadas também as soluções actuais de plataformas de suporte ao trabalho baseado no conhecimento, com suporte a gestão de informação, existentes no mercado.

No terceiro capítulo é apresentado o primeiro módulo, designado Explorer+. Sendo detalhadas, um exemplo de utilização, a análise de requisitos, a modelação e a implementação do mesmo.

No quarto capítulo é apresentado o segundo módulo, designado Viewer. Sendo detalhadas, um exemplo de utilização, a análise de requisitos, a modelação e a implementação do mesmo.

O quinto capítulo apresenta as conclusões de todo o trabalho efectuado ao longo deste projecto e sugere algumas perspectivas de desenvolvimento para o futuro.

Capítulo 2

Sistemas de trabalho baseados em conhecimento intensivo

Os módulos de gestão de informação desenvolvidos no âmbito desta dissertação integrarão uma plataforma de suporte a actividades de gestão de inovação, que conforme já referido, são actividades baseadas em conhecimento intensivo.

Neste capítulo definimos trabalho do conhecimento e processos de trabalho relacionados com o mesmo, dando especial atenção a gestão de informação, pois é o tema da dissertação em questão. O capítulo termina com a análise e comparação de soluções existentes no mercado.

2.1 Trabalho do conhecimento

Os utilizadores finais do sistema de gestão de conhecimento e informação são, todo o tipo de trabalhadores que baseiam as suas acções no seu conhecimento intrínseco. Neste contexto, é necessário perceber o que é conhecimento, como o gerir e o tipo de trabalho associado.

2.1.1 Conhecimento

A forma mais correcta de entender o que é o conhecimento é compreender a relação entre dados, informação e conhecimento.

A figura [2.1](#) apresenta, de forma simples, o relacionamento e a hierarquia existente entre, dados, informação e conhecimento, com o objectivo de aquisição de sabedoria.

O conhecimento divide-se em dois tipos, o conhecimento tácito e o conhecimento explícito.

O conhecimento tácito é o saber fazer implícito e não documentado em nenhum formato, está associado a cada pessoa, é um conhecimento implícito e é baseado na experiência adquirida ao longo dos anos. É aquele que o indivíduo adquiriu ao longo da vida, que está na cabeça das pessoas. Geralmente é difícil de ser formalizado a outra pessoa, pois é subjectivo, não mensurável e impossível de ensinar sem existir contacto com o mesmo.



Figura 2.1: Pirâmide do conhecimento.

O conhecimento tácito é mais valioso devido a sua difícil captura, registo e divulgação, exactamente por ele estar ligado as pessoas. É o que algumas pessoas chamam de verdadeiro conhecimento.

O conhecimento explícito é formal, claro, com regras e fácil de ser comunicado. Pode ser formalizado em textos, desenhos ou diagramas, assim como guardado em bases de dados ou publicações. Geralmente está guardado em artigos, revistas, livros e documentos, logo este tipo de conhecimento é confundido com a própria informação, na sua forma mais simples.

Os dois tipos de conhecimento expostos anteriormente não são opostos, eles complementam-se. Neste contexto, um sistema de gestão de conhecimento e informação deve contemplar esta complementaridade.

[Swe10, Ind09, Has07, NC99]

2.1.2 Gestão do conhecimento

A gestão do conhecimento é todo o sistema que gere as informações e conhecimentos, tanto internos como externos de uma organização. Está assente em processos de localização, extração, partilha e criação de conhecimento, através das ferramentas e tecnologias de informação e comunicação.

Vários autores, como *Drucker (1993)*, *Davenport (1996)*, *Staples (2001)* ou *Holsapple (2008)* afirmam que boas iniciativas e práticas de gestão do conhecimento contribuem para a sustentabilidade das vantagens competitivas das organizações que as empreendem.

Uma boa gestão do conhecimento pode ainda gerar redução dos custos, redução de tempo de produção, rápida comercialização de novos produtos, aumento do valor das ações, melhoria dos processos internos e maior fluidez nas operações, processos de tomada de decisões mais eficientes e melhores tomadas de decisão.

Os principais objectivos da gestão do conhecimento numa organização é tornar acessíveis grandes quantidades de informação organizacional, compartilhando as melhores práticas e tecnologias, organizar e acrescentar lógica aos dados de forma a torná-los compreensíveis. O sistema deve também apoiar a geração de novos conhecimentos, propiciando o estabelecimento de vantagens competitivas.

[Swe10, Ind09, Has07, NC99, KRO06]

2.1.3 Trabalho baseado no conhecimento

Para *Drucker*, o trabalho baseado no conhecimento caracteriza-se da seguinte forma:

"This is an orientation or management practice that Drucker has labeled "management by objectives"(MBO). With this approach, knowledge workers and management establish specific goals or objectives within the organization so that the outcome is mutually understood, but the specific course of actions and decision making is left to the knowledge worker. ACM provides for guidance and measurement of outcomes, as well as the long-term maintaining of data surrounding the process to demonstrate how objectives were realized and decisions were made." [Swe10, cap. Introduction]

O trabalho baseado no conhecimento não pode ser visto como um conjunto de processos de negócio tradicional (onde existe um plano e nos limitamos a segui-lo), o trabalho baseado no conhecimento procede conforme os acontecimentos vão surgindo no dia-a-dia, as decisões dependem dos acontecimentos e de quem as toma.

A cada acontecimento que surge deve se elaborar um plano, que a cada novo acontecimento muda ou adapta-se com base no conhecimento do trabalhador, neste modo de trabalho o negócio fica muito dependente da experiência do trabalhador. É necessário um planeamento, mas ao mesmo tempo uma flexibilidade, que determina a forma de actuar nos diferentes casos que podem surgir.

O plano é pôr a jogar os melhores indivíduos em cada caso (cada acontecimento), só assim podemos tirar o máximo partido dos trabalhadores em cada uma das situações.

O armazenamento e partilha rápida de conhecimento também são essenciais, pois só dessa forma é que possível ajudar o utilizador na avaliação de cada situação.

O trabalho baseado na experiência não surgiu agora, mas existe desde sempre, é um trabalho sustentado no conhecimento intrínseco das pessoas que o executam.

O exemplo do jovem capitão dos bombeiros que tem que tomar decisões na hora, para salvar o alpinista em perigo é um bom exemplo de trabalho baseado no conhecimento,

pois o jovem capitão vai tomando várias decisões ao longo do salvamento baseadas na sua experiência e conhecimento. Devido a necessidade de resposta rápida aos acontecimentos sucessivos e as mudanças repentinas de contexto, ele é forçado a agir com destreza e celeridade tomando inúmeras decisões baseadas no seu conhecimento intrínseco.

O que é novo é a categorização do trabalho dessas pessoas em trabalho baseado no conhecimento, unindo práticas e profissões que dantes eram vistas separadamente e agora são vista como trabalho baseado no conhecimento (médicos, advogados, bombeiros, etc. são pessoas que baseiam as suas acções em conhecimento intrínseco adquirido).

Então não existe uma forma metódica de resolver o problema do trabalho baseado na experiência, mas existem regras, leis, regulamentações e dados históricos que restringem e servem de orientador para o trabalhador na hora de tomar a decisão.

As regras ajudam e restringem o trabalhador na sua tomada de decisão, por isso a aproximação destas regras e restrições na hora de tomada de decisão é também um requisito para um futuro sistema de suporte.

O trabalho baseado no conhecimento não é repetitivo o suficiente para ser possível automatizar, mas é necessário uma espécie de guia ou descrição formal para se estar sempre a par de toda a situação do processo de negócio. Se esse acompanhamento não for efectivo, corre-se o risco de o processo se alongar em demasia, aumentando os custos (tempo, dinheiro e recursos) drasticamente.

A automatização de tudo não é a forma de resolução dos problemas de trabalho baseado no conhecimento, porque o investimento na automatização do processo de resolução deste tipo de problemas pode não compensar a resolução de futuros problemas do mesmo tipo. Estes problemas são imprevisíveis e possivelmente não se repetem, logo perde-se tempo e dinheiro a automatizar um sistema que não voltaria a ser utilizado e não reembolsaria os gastos na automatização.

Uma plataforma de suporte do trabalho baseado no conhecimento deve responder as particularidades associadas a este tipo de trabalho, pois só dessa forma é possível responder às reais necessidades dos utilizadores finais da plataforma.

Ao nível da gestão de informação, uma plataforma de suporte do trabalho baseado em conhecimento deve possuir os seguintes requisitos:

- Suportar informação dispersa em múltiplos suportes;
- Apresentar informação relevante independentemente do seu suporte;
- Especificar a informação a apresentar em cada contexto de utilização (Configurável);
- Especificação dos níveis de estruturação da informação (Configurável);
- Definição de estrutura de *dossier*;

- Gestão de permissões associada ao *dossier*;
- Controlo de versões

[Swe10, Has07, BFP⁺08]

2.2 Processos de trabalho

Até agora, os maiores ganhos de produtividade nos processos de negócio foram alcançados através da automatização de processos (processos estruturados). No entanto, muitos processos não dão para seguir esta abordagem, e nesses casos a solução muitas vezes tem passado pela utilização de ferramentas *ad hoc* para coordenar todo o trabalho, como correio electrónico ou mensagens instantâneas. Embora inegavelmente úteis, estas ferramentas são desligadas do processo e podem levar ao sobre carregamento e improdutividade dos processos. Desta forma, é necessário definir os diversos tipos de processos de trabalho e sistemas para suportar os mesmos. [Swe10]

Um processo de trabalho é um conjunto de acções ou tarefas executadas com o objetivo de atingir um dado resultado.

Os processos de trabalho podem ter diferentes níveis de estruturação, isto é, os processos podem ser estruturados, semi-estruturados ou não estruturados.

Os processos estruturados têm associado um fluxo de trabalho pré-definido e devem seguir esse fluxo, os executantes desses processos não têm que tomar nenhuma decisão apenas seguem o plano de trabalhos, pois essa decisão já foi tomada por alguém previamente.

A metodologia utilizada nos processos estruturados é a metodologia *BPM*¹.

Os processos não estruturados ou colaborativos não têm qualquer tipo estruturação, nem métodos definidos, é ideal para processos ágeis, com poucas pessoas. As *Wiki's*² são um bom exemplo de sistemas para gestão de processos de trabalho colaborativo.

Os processos semi-estruturados são processos de trabalho que requerem diferentes níveis de estruturação, os processos de trabalho baseado no conhecimento são processos semi-estruturados com algumas particularidades.

[Swe10]

2.2.1 Processos de trabalho baseado no conhecimento

Os processos de trabalho baseado no conhecimento são processos de trabalho que requerem um misto de estruturação e adaptabilidade ao caso. Um sistema para gerir estes processos de trabalho deve fornecer outro tipo de liberdade ao utilizador e suportar os diferentes tipos de processos de trabalho.

¹http://en.wikipedia.org/wiki/Business_process_management

²<http://en.wikipedia.org/wiki/Wiki>

Estes processos necessitam de métodos de gestão mais ágeis e flexíveis que as soluções *BPM* convencionais, a metodologia *ACM* revela-se mais adequada a este tipo de processos. As características principais da metodologia *ACM* são:

- Orientada a objectivos – trabalho focado no objectivo;
- Imprevisibilidade - difícil prever o que vai acontecer, difícil prever o futuro e os passos seguintes;
- Orientado a um *template* do processo - pequeno guia/orientador para o fluxo de trabalho;
- Restrições - Definir restrições no fluxo de trabalho, que não podem ser executadas;
- Biblioteca de *templates* - juntar e partilhar *templates* de resolução de problemas pelos vários membros da organização;
- Novos casos podem utilizar diferentes partes de *templates* diferentes, criando outro *template*;
- Diminuição de esforço na realização de tarefas;
- Vista do gestor - providenciar uma vista para o gestor ou administrador do caso para ele conseguir gerir efectivamente o caso;
- Acompanhamento diário das tarefas - o acompanhamento sobre as tarefas a realizar, fará aumentar o rendimento e poupar tempo para tarefas mais criativas e dispendiosas em termos de tempo.

[Swe10]

2.2.2 Sistema de suporte ao trabalho baseado no conhecimento

Sistema de trabalho entende-se por um conjunto de processos e actividades executadas numa dada organização ou unidade organizacional, e de pessoas e recursos que os executam.

Os sistemas de suporte ao trabalho baseado no conhecimento necessitam de um sistema de partilha e armazenamento de experiências e casos, pois só assim é possível internalizar a informação por toda a organização. Com este sistema de partilha de experiências é possível divulgar por toda a organização as melhores práticas e as melhores formas de agir em diferentes casos. Esta técnica permite que o conhecimento fique armazenado e passe a pertencer a empresa, não exclusivamente ao trabalhador.

Mesmo que a curto prazo o sistema de suporte ao trabalho baseado no conhecimento seja a causa de gastos desnecessários, este tipo de sistemas a longo prazo pode ser um

verdadeira fonte de valor e conhecimento, indispensável para os KWs e para a organização.

A principal dificuldade de um sistema de suporte ao trabalho do conhecimento é ser aceite pelos KWs, então para ser aceite o sistema deve possuir um elevado grau de usabilidade e flexibilidade. Elevada flexibilidade, de modo a ser possível adoptar diferentes paradigmas de gestão conforme as características de cada tipo de processo, ou até de cada instância; e ainda para se adaptar às diferentes características dos KWs. Elevada usabilidade, de modo a facilitar a utilização do sistema aos KWs, como por exemplo gestão do sistema ser baseada nas ferramentas *Microsoft Office*³.

Estes sistemas devem ainda permitir uma visibilidade e controlabilidade aos diferentes utilizadores do sistema, isto é, deve permitir aos KWs uma visão das tarefas a eles afectadas e deve permitir aos responsáveis pela organização uma visão dos processos em curso.

O nível de estruturação destes sistemas deve corresponder ao melhor compromisso entre, flexibilidade e usabilidade por um lado, consistência e controlabilidade global por outro. Tendo em conta que o sistema de trabalho tem tendência a mudar de projecto para projecto em função das suas características (processos e pessoas associadas). Em suma, estes sistemas devem ser um misto de estruturação e não estruturação.

[Swe10, HYJK06]

2.2.3 Síntese

O trabalho baseado no conhecimento requer uma aproximação diferente do trabalho rotineiro, diferente da automatização de trabalho, pois o método de automatização a maioria das vezes não funciona para o trabalho baseado no conhecimento. Este tipo de trabalho não deve ser tratado como um conjunto de processos para um determinado fim mas sim um conjunto de informações e recursos para ajudar o trabalhador a actuar na hora certa.

A imprevisibilidade associada ao trabalho baseado no conhecimento requer um sistema que suporte todo o tipo de processos de trabalho, e ainda que os processos suportem diferentes níveis de estruturação.

Os sistemas de suporte ao trabalho baseado no conhecimento devem suportar componentes mais estruturadas e componentes menos estruturadas. As componentes mais estruturadas são políticas, leis, regras e procedimentos da organização ou mesmo de um país. As componentes menos estruturadas são as componentes que proporcionam aos KWs a autonomia necessária para tomar decisões na hora certa.

[Swe10, Has07, McC08]

³<http://office.microsoft.com/>

2.3 Comparação de soluções existentes

Como se viu, este projecto tem por objectivo a construção de dois módulos de gestão de informação, então a comparação das soluções existentes incidiu sobretudo sobre plataformas de apoio à tomada de decisão com um sistema de gestão de recursos de informação. A grande parte destas aplicações são realizadas por empresas privadas que não querem ver revelada a arquitectura e desenho dos seus sistemas. De seguida são apresentados plataformas e sistemas que suportam gestão de conteúdos empresariais.

Neste ponto, são analisadas plataformas de suporte a trabalho com um sistema de gestão de conteúdos de informação incorporado.

Microsoft SharePoint 2010

Segundo a própria *Microsoft*, o *Microsoft SharePoint 2010*⁴ é:

"Uma Plataforma de colaboração empresarial para a Empresa e para a Internet. Facilita o trabalho em conjunto. Os seus colaboradores podem configurar sítios *Web* para partilhar informações, gerir documentos do início ao fim e publicar relatórios para ajudar a tomar melhores decisões. O *SharePoint 2010* gere documentos, registos, conteúdos da *Web* e multimédia numa única plataforma." [Mic10]

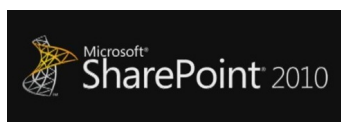


Figura 2.2: Sharepoint.

⁴<http://sharepoint.microsoft.com/>

Segundo a figura 2.3, na comunidade AIIM⁵, o *SharePoint* é visto como um sistema de colaboração.

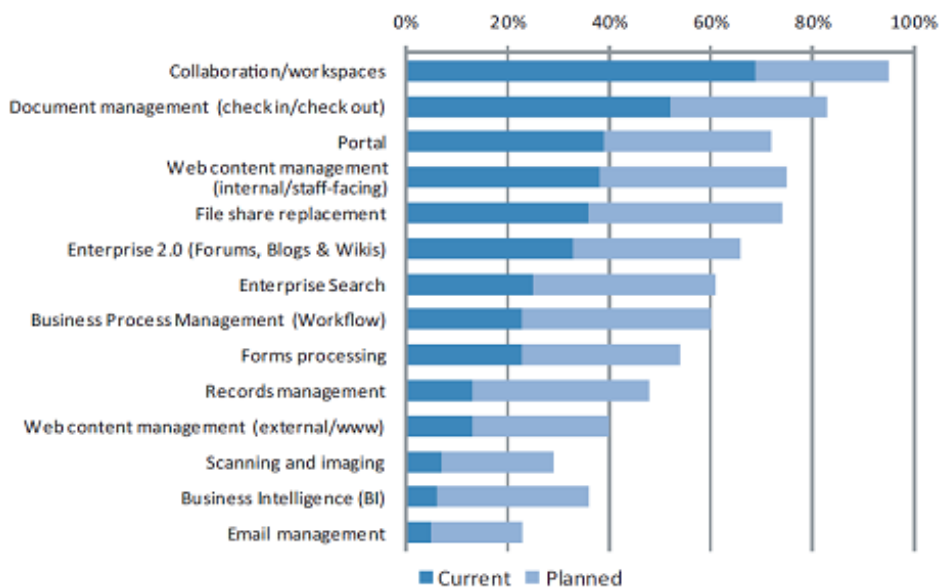


Figura 2.3: Gráfico de respostas a "Aplicações do SharePoint?"

A plataforma *SharePoint* utiliza o *SharePoint 2010 Content*, para gestão de conteúdos empresariais. O *SharePoint 2010 Content* disponibiliza inúmeras funcionalidades para gestão de conteúdos empresariais, como:

- Armazenar, organizar e localizar conteúdos;
- Garantir a consistência dos conteúdos;
- Gestão de meta-dados para conteúdos;
- Ajudar a proteger conteúdos de acesso ou utilização não autorizados;
- Garantir processos de negócios consistentes para a forma como os conteúdos são tratados;
- Bibliotecas de *templates* de conteúdos;
- Localização rápida e facilmente de conteúdos.

⁵<http://www.aiim.org>

Segundo a figura 2.4, na comunidade AIIM⁶, a maioria das empresas planeia utilizar o *SharePoint* para gestão documental.

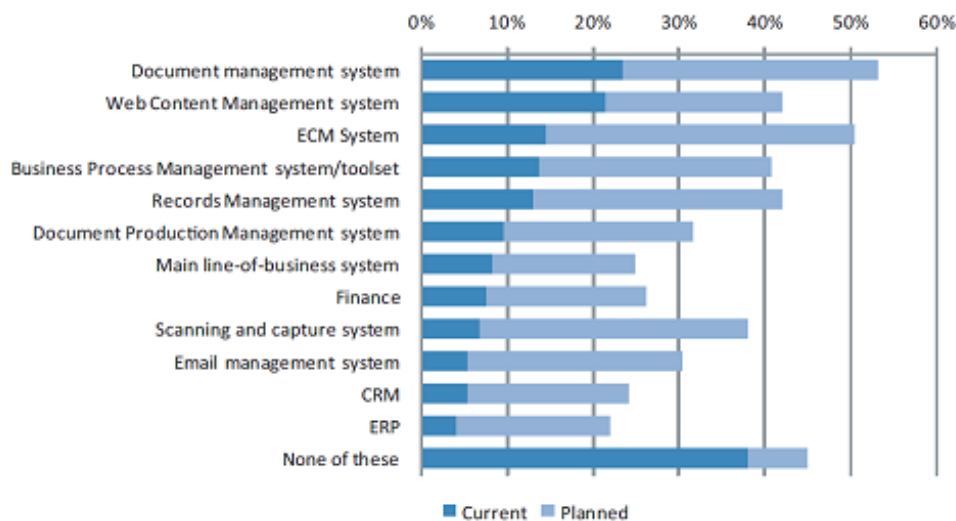


Figura 2.4: Gráfico de respostas a "Quais dos seguintes sistemas deseja integrar com o SharePoint?"

[Mic10, AI10]

FileDoc 2.0

Segundo a *Webuild*⁷, o File Doc 2.0 é:

"O File Doc é um gestor documental, para sistema de trabalho colaborativos que permite a gestão facilitada de documentos e fluxos de trabalho. A fácil utilização é a característica principal do File Doc. O sistema foi desenvolvido em Microsoft para ambiente *Web*, possibilitando o fácil acesso a partir de um navegador de Internet." [Web10]



Figura 2.5: FileDoc.

As principais funcionalidades do *File Doc* são:

- Acesso, autenticação e segurança;
- Edição e gestão de versões;

⁶<http://www.aiim.org>

⁷<http://www.webuild.pt/>

- Gestão de fluxos de trabalho;
- Gestão de permissões;
- Gestão de Documento e de *e-mail*;
- Classificação e reclassificação de conteúdos.

O *File Doc* é um sistema de gestão documental, de processos e *workflow* que integra as funcionalidades de arquivo e gestão do ciclo de vida dos mais diversos formatos de conteúdos. O sistema foi desenvolvido em tecnologia *Microsoft*⁸ para ambiente *Web* e é totalmente integrável com as ferramentas *Microsoft Office*⁹.

O *File Doc* é constituído por um conjunto de módulos que cumprem objectivos distintos e que podem ser implementadas de modo integral ou de forma faseada. A figura 2.6 apresenta o conjunto de módulos do sistema.

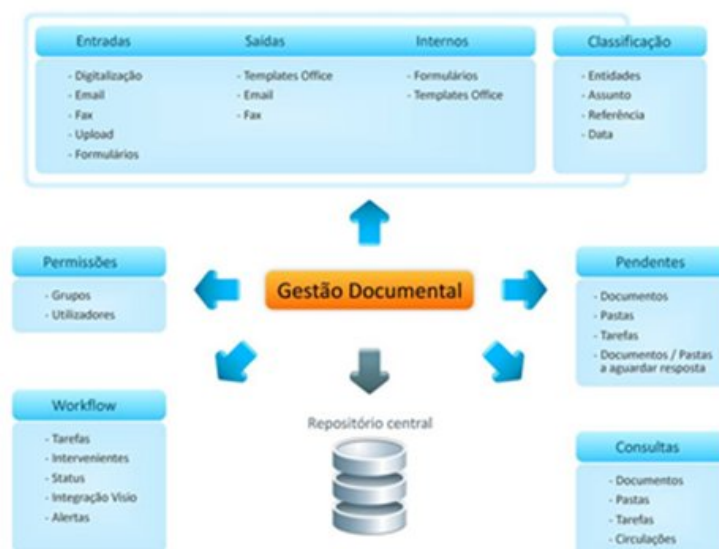


Figura 2.6: Módulos do File Doc.

[Web10]

⁸<http://www.microsoft.com/>

⁹<http://office.microsoft.com/>

Aconex

O *Aconex*¹⁰ é um sistema de gestão de documentos *on-line* e de colaboração via *Web*¹¹ para gestão de projectos de construção, engenharia e instalações. O sistema fornece o armazenamento e a gestão de toda a documentação e comunicação do projecto, acessível a partir de qualquer ligação a Internet.

As suas funcionalidades principais são:

- Gestão de documentos;
- Gestão de *e-mail*;
- Segurança de acesso *on-line*;
- Pesquisa de informação;
- Gestão de projectos;
- Colaboração;
- Controlo de versões;
- Gestão de fluxos de trabalho;
- Gestão de concursos.

O sistema *Aconex* é uma plataforma para suporte a projectos de construção e engenharia.

[Aco10]

KnowledgeTree

O *KnowledgeTree*¹² é um sistema de gestão do conhecimento, com gestão documental e controlo de fluxos de trabalho através de alertas sobre documentos. Este sistema permite carregar documentos, edição dos mesmos com controle de versões, criação de alertas sobre documentos para controlo de processos de trabalho e ainda criação de pastas partilháveis.



Figura 2.7: KnowledgeTree.

¹⁰<http://www.aconex.com/>

¹¹http://en.wikipedia.org/wiki/World_Wide_Web

¹²<http://www.knowledgetree.com/>

As suas funcionalidades principais são:

- Partilha de ficheiros e colaboração;
- Pesquisa de informação;
- Controlo de fluxo através de alertas sobre documentos;
- Integração com aplicações *Microsoft* e aplicação *desktop*. [Kno10]

iPortalDoc

O *iPortalDoc*¹³ é uma plataforma de gestão documental e gestão de fluxos de trabalho *on-line*. A sua utilização intuitiva permite uma curva de aprendizagem rápida.



Figura 2.8: iPortalDoc.

“O iPortalDoc Light surge com a necessidade de trocar informação entre uma empresa e os seus clientes, fornecedores e parceiros: uma organização que tenha o sistema de Gestão Documental e Workflow, iPortalDoc, com todos os documentos devidamente organizados e classificados, pode disponibilizar o acesso a informação específica por parte de entidades externas. Ou seja, depois de estabelecer os critérios de acesso a essa informação (definindo que entidade pode aceder a que informação), a organização permite que outras entidades possam introduzir, encaminhar e consultar documentos e integrem workflows, através de um interface Web chamado iPortalDoc Light.” [iPo10]

As características centrais do *iPortalDoc* são, facilidade de utilização, formação reduzida, integração no ambiente de trabalho dos utilizadores, redução do *paperware*, controlo de processos com utilização de *workflows*, segurança no acesso e acções sobre documentos.

As suas funcionalidades principais são:

- Hierarquia de documentos;
- Listagem de acções pendentes;
- Introdução de *templates* de documentos;
- Menu de pesquisa rápida sobre documentos;

¹³<http://www.iportaldoc.com/>

- Encaminhamento e download de ficheiros;
- Consulta de informação de documentos;
- Inserção de documentos facilitada;
- Filtro de pesquisa avançada;
- Gestão de *workflows*.

[iPo10]

Document Locator

O *Document Locator* consiste num sistema de gestão de conteúdos empresariais, para aumentar a eficiência colaborativa e comunicativa das organizações.



Figura 2.9: Document Locator.

As suas principais funcionalidades são:

- Gestão de documentos;
- Gestão de *e-mail*;
- Gestão de conteúdos;
- Pesquisa de conteúdos;
- Controlo de versões;
- Controlo de conteúdos;
- Armazenamento de conteúdos;
- Gestão de fluxos de trabalho;
- Associação de fluxos de trabalho a conteúdos;
- Unificação de conteúdos.

[Loc10]

eDocLink

O *eDocLink* é um sistema vocacionado para suporte ao trabalho de administração pública.



Figura 2.10: eDocLink.

As funcionalidades principais do *eDocLink* são:

- Gestão documental com as aplicações de *office*;
- Pesquisa de conteúdos;
- Gestão de *e-mail*;
- Alarmes e Controlo de Prazos;
- Encaminhamento / Distribuição de documentos;
- Organização Processual dos documentos;
- Gestão de processos;
- Documentação associada ao processo;
- Integração com *templates MS Word*;
- Recepção de alarmes e notificações com origem na Gestão Documental;
- Formulários *Web*.

[[MMC10](#)]

Alfresco

O *Alfresco* é uma plataforma para gestão de conteúdos empresariais, esta plataforma suporta:

- Gestão documental;
- Colaboração e Partilha;
- Gestão de conteúdos *Web*;
- Gestão de fluxos de trabalho.



Figura 2.11: Alfresco.

As principais funcionalidades do *Alfresco*, ao nível da gestão documental são as seguintes:

- Gestão de conteúdos;
- Funcionalidades ECM;
- Controlo de versões automático;
- Partilha de conteúdos facilitada;
- *Drag and Drop* a partir da caixa de correio electrónico;
- Associação de pastas a fluxos de trabalho;
- Integração *Microsoft Office*;
- Gestão de permissões;
- Gestão de acções e papéis;
- Associação de meta-dados a informação;
- Segurança.

O *Alfresco* tem a particularidade de ser *Open Source*¹⁴ e poder ser ainda integrado em ferramentas *Microsoft Office*¹⁵, proporcionando assim uma adaptabilidade mais efectiva aos diferentes tipos de utilizadores.

[Alf10]

¹⁴<http://www.opensource.org/>

¹⁵<http://office.microsoft.com/>

DropBox

O *Dropbox*¹⁶ é um sistema para gerir ficheiros, após a instalação e registo *on-line*. O utilizador tem total liberdade para criar, adicionar, editar, mover e eliminar pastas e ficheiros, todas estas acções podem ser executadas no sítio *Web*¹⁷ após efectuada a autenticação ou por uma aplicação instalada num qualquer computador ligado à Internet¹⁸ (bastando a autenticação), essa aplicação não é mais que uma pasta ligada ao servidor de ficheiros do *Dropbox* e cada vez que é efectuada alguma alteração nessa pasta o servidor efectua o carregamento automático da mesma. Todos os ficheiros e pastas estão armazenados num servidor de ficheiros do *Dropbox*, é possível partilhar (dar permissões para visualizar, editar e eliminar) pastas entre os utilizadores do sistema. Este sistema não permite a edição simultânea de ficheiros, é um sistema ideal para partilha de documentos.



Figura 2.12: DropBox.

[Dro10]

GoogleDocs

O *GoogleDocs*¹⁹ é um sistema para gerir ficheiros, após registo. O utilizador tem total liberdade para criar, adicionar, editar, mover e eliminar pastas e ficheiros, todas estas acções podem ser executadas *on-line* e em simultâneo. É possível partilhar pastas ou ficheiros com restantes utilizadores. Permite ainda a edição simultânea de ficheiros, esta funcionalidade não é necessária no gestor de recursos de informação a desenvolver. É um sistema ideal para partilha e edição simultânea de documentos em processos de trabalho.

[Goo10]

2.3.1 Resumo

O conceito de gestão de conteúdos de informação é uma necessidade em qualquer área de negócio. Neste ponto, foram analisadas diferentes soluções de aplicação de gestão de informação em diversas áreas, cada uma delas com as suas especificidades e dificuldades próprias.

Sistemas como, a *Aconex*, *iPortalDoc* ou *eDocLink* são sistemas de suporte vocacionados a um tipo de sistema de trabalho. O *DropBox* ou *GoogleDocs* permitem apenas

¹⁶<http://www.dropbox.com/>

¹⁷http://en.wikipedia.org/wiki/World_Wide_Web

¹⁸<http://en.wikipedia.org/wiki/Internet>

¹⁹<http://docs.google.com/>

a gestão e partilha de diversos tipos de documentos. O *Alfresco* ou *Document Locator* são plataformas baseadas ECM²⁰ e assim sendo, são sistemas de suporte bem mais orientados ao suporte de trabalho baseado no conhecimento que todos os outros inumerados anteriormente. Por último, o *SharePoint* é uma plataforma ideal para suporte a trabalho do conhecimento, mas perde em termos de usabilidade.

Com base na análise dos vários sistemas existentes de gestão documental e na metodologia ACM, os módulos de gestão de informação devem suportar uma gestão unificada.

A gestão unificada, referida anteriormente, consiste em gerir os diferentes tipos de recursos de informação (dados, ficheiros e *e-mail*) utilizando:

- Os mesmos conceitos e métodos de análise e modelação;
- As mesmas interfaces de configuração e de consulta.

A aplicação da gestão unificada, deve possuir as seguintes funcionalidades:

- Configuração *in place*;
- Hierarquia de recursos;
- Meta-dados;
- Apresentação e organização da informação;
- Permissões de acesso;
- Inserção, edição e eliminação de recursos facilitada;
- Filtro de pesquisa avançada;
- Restrições de integridade;
- Pesquisa de informação;
- Controlo de versões;
- Classificação e reclassificação de pastas e recursos, automático ou semi-automática(*Drag and Drop* entre pastas ou classificação individual);
- Criação de *templates* de estruturas de recursos para sistemas típicos;
- Criação de *dossier* de projecto;
- Associação a processos de trabalho;
- Integração com ferramentas *Microsoft Office*²¹.

²⁰http://en.wikipedia.org/wiki/Enterprise_content_management

²¹<http://office.microsoft.com/>

Também neste contexto, é possível afirmar que uma plataforma que suporte todo o tipo de processos de trabalho e todo o tipo de necessidades associadas ao trabalho baseado no conhecimento será uma mais-valia para o mundo empresarial, pois desse modo é possível utilizar a mesma plataforma independentemente do projecto, do sistema de trabalho ou do processo.

Conhecimento gera conhecimento, então qualquer meio que permita uma localização, extracção, partilha, criação, armazenamento e consulta de conhecimento é sempre uma ferramenta de grande valor. Neste contexto, a gestão de informação ou conhecimento é muito importante para o projecto *I9Source*.

[[Swe10](#), [Cra09b](#), [Cra09a](#)]

Capítulo 3

Módulo Explorer+

Conforme já referido, o Explorer+ foi um dos módulos desenvolvidos no âmbito desta dissertação. Este capítulo contém a análise de requisitos, o desenho e modelação dos dados, e a implementação do módulo Explorer+.

O módulo apresenta o conteúdo do *dossier* de uma instância em forma de uma árvore de pastas hierárquicas. A cada instância está associado um *dossier*, que corresponde a uma pasta numa biblioteca de documentos na plataforma de suporte *Sharepoint*. O conteúdo pode ser *e-mails*, ficheiros, ligações, anotações e pastas. A apresentação deverá ser em árvore, para garantir uma estruturação dos conteúdos. Em suma, o módulo Explorer+ consiste numa espécie de *Tree View* habilitada com todas as funcionalidades que um explorador de *Windows*¹ possui, como organização em pastas hierárquicas, *Drag and Drop* ou menus de contexto nos objectos. Contem ainda funcionalidades suplementares, como controlo de versões, controlo de *Check-in/out* ou gestão de permissões.



Figura 3.1: Módulo Explorer+.

¹http://en.wikipedia.org/wiki/Microsoft_Windows

3.1 Exemplo de aplicação módulo: Página projectos

O módulo Explorer+ poderá ser aplicado a qualquer página, onde seja apresentada uma instância da aplicação que possua um *dossier* associado. Como, a maioria das instâncias do sistema de gestão de inovação está nestas condições, o Explorer+ poderá ser aplicado na maioria das páginas da plataforma *I9Source*.

A título de exemplo do módulo Explorer+, é apresentada de seguida a utilização deste módulo na página projectos. Esta página apresenta instâncias da entidade projectos.

A figura 3.2 apresenta um exemplo de uma instância da página projectos, com o módulo Explorer+.

The screenshot shows the I9Source platform interface. On the left is a navigation menu with categories like 'Entrada', 'Programas de Financiamento', 'Candidaturas', 'Projectos', 'Arquivo', 'Cooperação', 'Vigilância', 'IDO', 'Base de Conhecimento', and 'Organização Interna'. The main area is titled 'Adapt' with a date '13-03-2011 | 27-05-2011'. A red box highlights the 'ExplorerMails' module, which shows a file explorer view with folders like 'Documentos', 'Arquivo', 'Piano do projecto', 'Article.pdf', 'Google link', and 'Novos contactos.note'. Below this, there are sections for 'E-mails' (Inbox) and 'Entidades relacionadas' (Candidates, Parcels, Programs of Financing, Protocols). On the right, there is a yellow box with instructions for providers, followed by a table of 'Assuntos' (Topics) and a table of 'Reuniões' (Meetings).

Assuntos	Estado	Data
swertyl		27-05-2011
new	Em curso	07-05-2011
Preparar viagem a França	Em curso	11-04-2011
Alteração do módulo de comunicação	Concluído	23-03-2011

Reuniões	Estado	Data
fgfjk	Agendada	27-05-2011
new	Agendada	07-05-2011
Reunião @ INESC	Agendada	11-04-2011
Reunião na FEUP	Agendada	11-04-2011
Reunião de kickoff	Concluído	19-02-2011

Relatórios	Estado	Data
zsofhgij	Em curso	27-05-2011
new rel	Em curso	07-05-2011
Relatório semanal de evolução	Em curso	30-03-2011

Figura 3.2: Página projecto com o módulo Explorer+

Módulo Explorer+

O projecto exemplo utilizado para apresentar a página projectos é o projecto *Aventura*. Em seguida, são apresentadas algumas das acções mais comuns numa página projectos. A figura 3.3 ilustra a página do projecto *Aventura*.

Aventura
5/15/2011 | 11/30/2011

Dossier

- Documentos
 - Consórcio
 - Plano
 - plano Adapt_v4.xlsx
 - Itens

E-mails

- Inbox

Conteúdos relacionados

- Candidaturas
- Parceiros
- Programas de Financiamento
- Protocolos

Ligar para o fornecedor a respeito dos novos componentes

- Pedir informações sobre capacidade
- Pedir orçamento

Preparar reunião - URGENTE!!

- Reservar sala e equipamentos
- Enviar e-mails de confirmação
- Preparar documentação

2Do's

	Estado	Data
Contratação bolsistas	Em curso	6/29/2011
Aprovação do acordo de consórcio	Concluído	6/29/2011
Aquisição licenças de software	Em curso	6/29/2011

Reuniões

	Estado	Data
Steering Committee - Maio 11	Agendada	6/17/2011
Technical Committee	Aguardar acta	6/9/2011
Kickoff	Concluído	5/17/2011

Relatórios

	Estado	Data
Relatório de arquitectura	Em curso	9/12/2011
Especificação requisitos	Em curso	7/5/2011
Estado da arte	Concluído	6/9/2011

Figura 3.3: Página projecto *Aventura*

Uma página de projecto pode ter associado vários assuntos, relatórios e reuniões, que são listados na página do projecto com ligações para os mesmos. A página disponibiliza também formas de adicionar novos assuntos, relatórios e reuniões ao projecto, criando automaticamente a pasta associada no *dossier* do projecto. A figura 3.4 ilustra a adição de um assunto ao projecto.

Plataforma de Gestão da Inovação

iSource2

Aventura

Dossier

- Entrada
- Financiamento
- Candidaturas
- Projectos
 - Adapt
 - Aventura**
 - Multiwave
 - Netcare
 - Skin Monitor
 - Arquivo
 - Cooperação
 - Organização
 - Vigilância
 - IDO
 - Indicadores
 - Base de Conhecimento

E-mails

- Inbox

Conteúdos relacionados

- Candidaturas
- Parceiros
- Programas de Financiamento
- Protocolos

2Do

Título:

Estado:

Descrição:

Submeter ☒ Cancelar

Figura 3.4: Adicionar assunto a projecto

A figura 3.5 ilustra a adição de uma reunião ao projecto.

The image shows a web application interface for 'iSource2'. A modal window titled 'Reunião' is open, allowing the user to add a meeting to a project. The form includes the following fields:

- Título:** A text input field.
- Data:** A date selection field with a calendar icon.
- Estado:** A dropdown menu currently showing 'Agendada'.
- Descrição:** A large text area for the meeting description.
- Ordem trabalhos:** A section with a rich text editor for listing tasks.
- Acta:** Another section with a rich text editor for the meeting minutes.

At the bottom right of the modal are 'Submeter' and 'Cancelar' buttons. The background interface shows a sidebar with a tree view under 'Projectos', where 'Aventura' is the selected project.

Figura 3.5: Adicionar reunião a projecto

A figura 3.6 ilustra a adição de um relatório ao projecto.

The image shows the same 'iSource2' web application interface. A modal window titled 'Relatório' is open, allowing the user to add a report to a project. The form includes the following fields:

- Título:** A text input field.
- Estado:** A dropdown menu currently showing 'Em curso'.
- Descrição:** A large text area for the report description.

At the bottom right of the modal are 'Submeter' and 'Cancelar' buttons. The background interface is consistent with the previous figure, showing the 'Projectos' sidebar with 'Aventura' selected.

Figura 3.6: Adicionar relatório a projecto

Uma página de projecto através do modulo Explorer+ disponibiliza diversas acções para gerir os conteúdos do *dossier* associado. As acções de gestão de *dossier* disponibilizadas por menu de contexto são:

- Adicionar pasta, pasta versionada, anotação ou ligação;
- Arquivar na base de conhecimento (apenas disponível para ficheiros);
- Propriedades (apenas disponível para pastas);
- Explorar;
- Actualizar;
- Alterar nome;
- Eliminar.

A figura 3.7 apresenta as acções possíveis sobre um ficheiro no *dossier*.

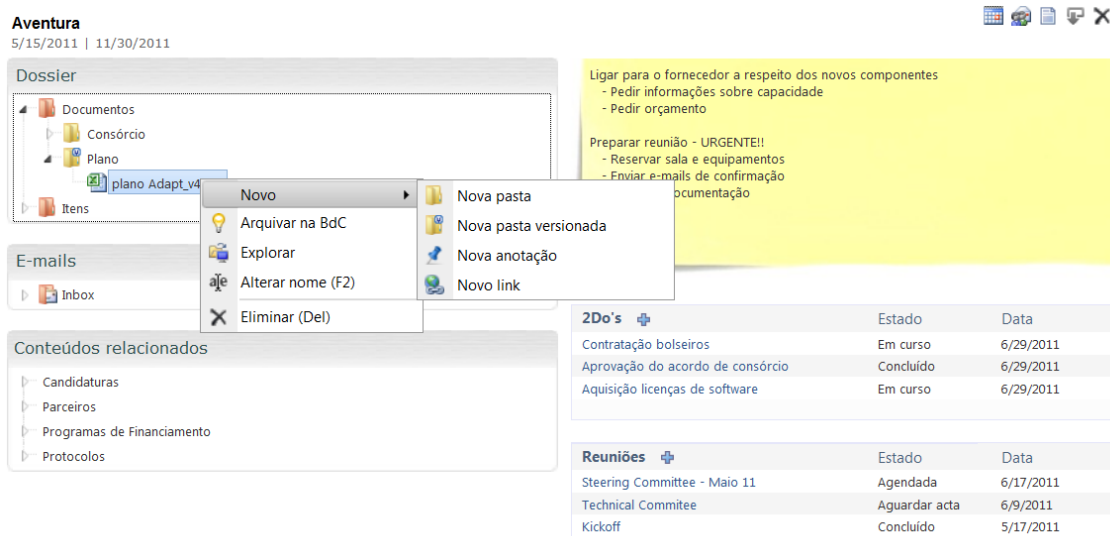


Figura 3.7: Gestão de conteúdos de *dossier*

De seguida, são apresentadas a arquitectura, o modelo de dados e a estrutura do *dossier* da página projectos.

3.1.1 Arquitectura página proyectos

A arquitectura da página de projectos está representado na figura 3.8.

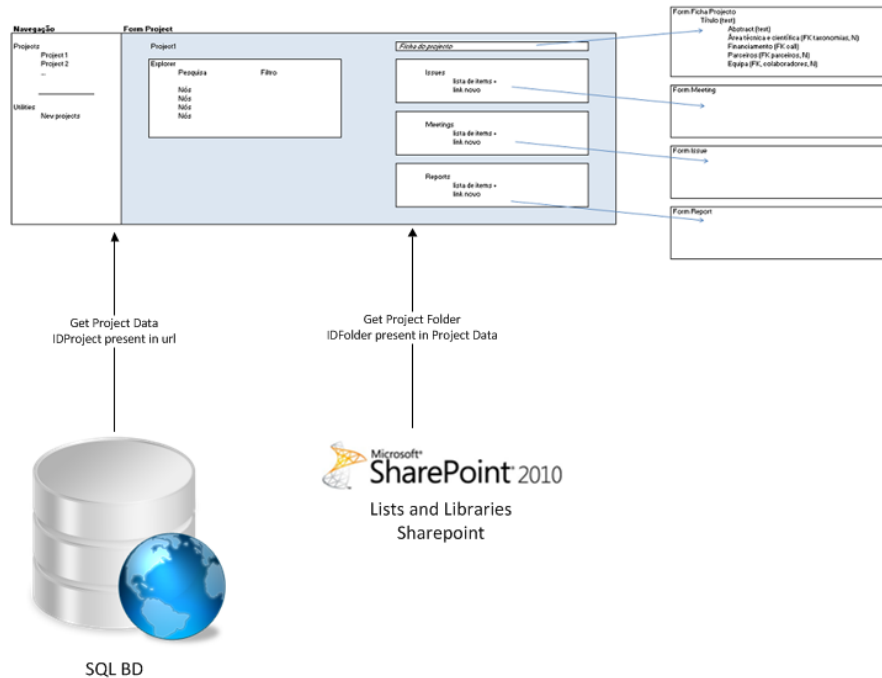


Figura 3.8: Arquitectura página proyectos

A arquitectura da página proyectos, demonstra que uma instância de um projecto tem um *dossier* e um registo em base de dados. O *dossier* guarda os conteúdos associados ao projecto. O registo em base de dados serve para guardar os meta-dados referentes ao projecto instanciado.

3.1.2 Modelo de dados página proyectos

O modelo de dados demonstra a relação do projecto instanciado pela página, com reuniões, relatórios e assuntos associados ao mesmo. Também é identificada a existência de um pasta relacionada com cada um dos itens.

O modelo de dados dos projectos, está esquematizado na figura 3.9.

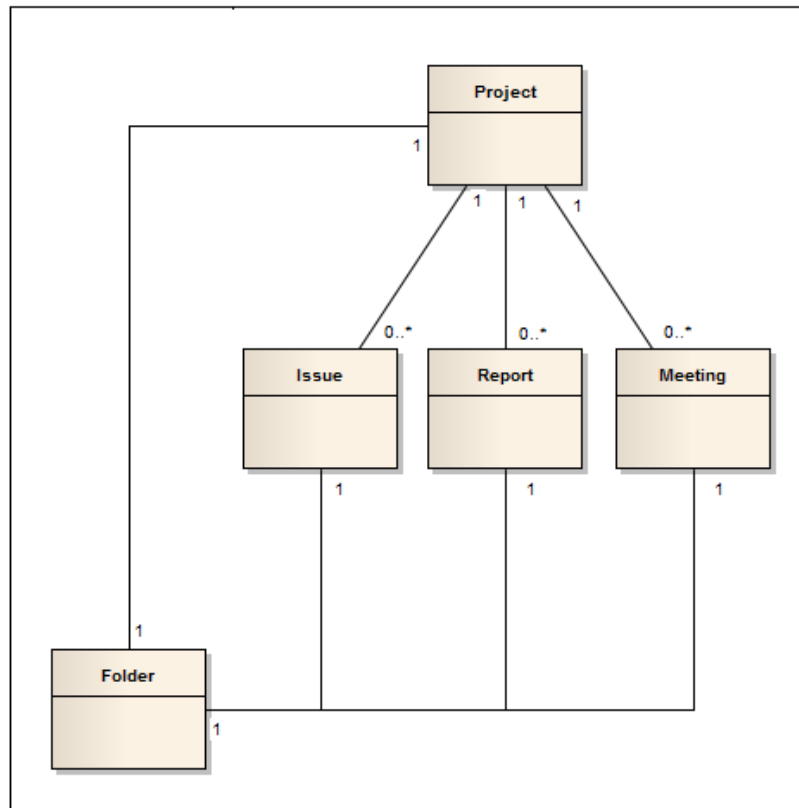


Figura 3.9: Modelo de dados projectos.

3.1.3 Estrutura do *dossier* projectos

Como já referido, a cada instância de um projecto corresponde um registo na base de dados (tabela projectos) e uma pasta numa biblioteca *Sharepoint*. O *ID* do registo em base de dados coincide com o nome da pasta na biblioteca. O *ID* da pasta também é um campo a preencher no registo em base de dados.

O *dossier* associado a um projecto inclui uma pasta por cada tipo de item, onde são arquivados os *dossiers* dos itens associados ao projecto. Os itens são assuntos, reuniões e relatórios. Estas pastas possuem um conjunto de restrições, geridas pelo módulo Explorer+.

Nos anexos A.1.2 está apresentada a estrutura de um *dossier* do projecto.

As pastas criadas por defeito no *dossier* do respectivo projecto, não são editáveis, pois cada novo relatório, assunto ou reunião adiciona uma pasta dentro de uma dessas pastas. Sendo que, a pasta adicionada será o *dossier* associado ao item adicionado.

As pastas criadas automaticamente pelo sistema são pastas do tipo *Managed Folder*, este tipo de pasta contém quatro campos para gestão das acções possíveis sobre os conteúdos, no módulo Explorer+. Os campos são definidos na tabela 3.1.

Tabela 3.1: Campos adicionais *Managed Folder*

Campos adicionais <i>Managed Folder</i>	
EditOwn	Booleano que define se é possível a edição da própria pasta ou não
EditSub	Booleano que define se é possível a edição dos conteúdos da pasta ou não
Version	Texto que define o nome do ficheiro versionado na pasta, em caso de <i>Null</i> não é uma pasta versionada
Ordering	Inteiro que define a ordem das pastas ao mesmo nível, em caso de <i>Null</i> a ordenação é alfabeticamente pelo nome da pasta

O anexo A.1.3 representa a estrutura de *Managed Folders* na pasta de projectos, e o valor dos atributos *EditOwn* e *EditSub* em cada uma.

As pastas do tipo *Folder* são tratadas com pasta do tipo *Managed Folder* com as características apresentadas na tabela 3.2.

Tabela 3.2: Valores campos *Managed Folder* quando pasta do tipo *Folder*

Valores campos <i>Managed Folder</i> quando pasta do tipo <i>Folder</i>	
EditOwn	<i>True</i>
EditSub	<i>True</i>
Version	<i>Null</i>
Ordering	<i>Null</i>

Assim, define-se os diferentes tipos de pastas no *dossier* de projectos, pois os mesmos serão o principal modo de implementação da gestão de permissões sobre os conteúdos no *dossier*.

3.2 Análise requisitos

A análise de requisitos consiste no levantamento e especificação do conjunto de características que o módulo deverá contemplar para corresponder às necessidades dos seus futuros utilizadores. Os requisitos podem ser classificados em funcionais, e não-funcionais.

3.2.1 Requisitos não funcionais

Os requisitos não-funcionais especificam as restrições a que o módulo tem que responder. Os requisitos não-funcionais identificados são:

- Facilidade de Utilização;
- Desempenho;
- Robustez;
- Flexibilidade.

Facilidade de utilização

Os utilizadores deste módulo serão maioritariamente trabalhadores do conhecimento, por isso e para que esta aplicação fosse facilmente aceite e posteriormente se tornasse uma ferramenta de uso diário dos utilizadores, uma das principais preocupações no seu desenvolvimento foi a facilidade de uso. Todas as funcionalidades da aplicação devem surgir de forma intuitiva e de fácil utilização para os seus utilizadores.

Para responder a este requisito, no desenvolvimento da aplicação, optou-se por disponibilizar os mesmos métodos de acção disponibilizados pelo *Windows Explorer*² na aplicação, pois desse modo a grande maioria dos utilizadores já estará familiarizado. Outro exemplo que a aplicação desenvolvida segue este requisito é a disponibilização de diferentes métodos para executar a mesma acção.

Desempenho

O módulo deve responder de forma rápida e eficaz a cada acção sobre o mesmo, pois o tipo de utilizadores do sistema tem como característica, a falta de paciência, levando a um retorno aos velhos métodos.

Robustez

É muito importante que a aplicação desenvolvida seja robusta, pois é outro factor que servirá como principal argumento para cativar o utilizador a confiar e explorar todas as funcionalidades que esta aplicação lhe fornece. Se a aplicação não garantir robustez e frequentemente ocorrem erros ou comportamentos inesperados, os utilizadores nunca vão construir uma relação de dependência com a aplicação, deixando-a cair em desuso.

²http://en.wikipedia.org/wiki/Windows_Explorer

Flexibilidade

A flexibilidade é outra característica bastante importante neste tipo de aplicações, pois se esta conseguir facilmente englobar mais processos de negócio, que poderão vir a estar à responsabilidade da organização, garante-se assim que esta ferramenta acompanha a expansão da própria organização e evitando a adopção de outras ferramentas, ou de uma nova fase de desenvolvimento para se adaptar à nova realidade.

3.2.2 Requisitos funcionais

A especificação dos requisitos para este módulo foi fundamentada nos estudos sobre trabalho baseado em conhecimento intensivo, gestão de conteúdos de informação e nas soluções existentes no mercado, abordadas no capítulo 2.

Tal como já foi referido, as ideias para este módulo concentravam-se no desenvolvimento de um sistema que permita o acesso e a gestão simples, rápida e eficaz da informação existente numa biblioteca SP associada a uma dada instância.

Lista de requisitos funcionais:

- Suporte a diferentes tipos de conteúdos;
- Gestão de permissões;
- Controlo de versões;
- Ordenação de pastas;
- *Drag and Drop*;
- *Upload e Download* de conteúdos;
- Pesquisa por texto;
- Filtragem por tipo de conteúdo.

Suporte a diferentes tipos de conteúdos

O módulo suportará diversos tipos de itens associados a cada pasta. Os tipos de itens suportados pelo Explorer+ serão:

- Ficheiros;
- Ligações;
- Notas.

Uma nota serve para reter algum tipo de anotação e tem como grande vantagem ser guardada no *dossier*.

Uma ligação serve para guardar uma hiperligação útil e o *click* sobre a mesma navega para o *url* associado.

As ligações e as notas devem ser editáveis, ambas são guardadas na pasta da biblioteca SP, como ficheiros XML³. As ligações têm como campos o título e a ligação. As notas têm como campos o título e o texto.

Gestão de permissões

A gestão de permissões de acesso aos conteúdos, permite determinar as acções possíveis sobre cada item nos conteúdos. A gestão de permissões permite determinar o menu de contexto associado a cada item na pasta e consequentemente restringindo as acções disponibilizadas. Assim, é possível associar a cada pasta a possibilidade ou não, de alterar a própria pasta e/ou alterar subpastas existentes. As permissões serão guardadas em meta-dados associados a cada pasta nos conteúdos.

Controlo de versões

O controlo de versões, permite a gestão de ficheiros versionados numa pasta. Uma pasta para ser versionada deverá estar devidamente sinalizada em meta-data associada a mesma.

O módulo apresenta apenas o ficheiro com a versão mais recente. As outras versões podem encontrar-se na pasta mas não são apresentadas pela interface, sendo possível visualizar as mesmas acedendo a pasta via servidor.

Ordenação de pastas

A ordenação de pastas ao mesmo nível na estrutura é também um requisito funcional do módulo. Este requisito deverá ser suportado com recurso a um campo nos meta-dados da pasta em questão. As pastas devem ordenadas primeiramente pelo campo de ordenação em meta-data e em seguida alfabeticamente. A ordenação deverá ser facilmente editável.

Drag and Drop

O módulo deverá contemplar o *Drag and Drop* de ficheiros e pastas, na árvore de conteúdos do módulo, de acordo com as permissões associadas as pastas.

³[urlhttp://en.wikipedia.org/wiki/XML](http://en.wikipedia.org/wiki/XML)

Upload e Download de conteúdos

O *Upload e Download* de conteúdos é disponibilizado através do *Drag and Drop* entre o ambiente de trabalho e uma pasta na biblioteca *Sharepoint*. A possibilidade de *Drag and Drop* depende das permissões associadas as pastas no servidor.

Pesquisa por texto

A pesquisa por texto sobre os itens nos conteúdos é outro requisito a disponibilizar no módulo. A pesquisa incide sobre o nome ou título dos conteúdos nas pastas.

Filtro por tipo de conteúdo

O módulo disponibilizará também um modo de filtragem dos tipos de itens a apresentar pelo módulo. Assim, é possível apresentar conteúdos de apenas de um tipo nas pastas.

3.2.3 Casos de utilização

A figura 3.10 apresenta o diagrama de casos de uso do módulo Explorer+.

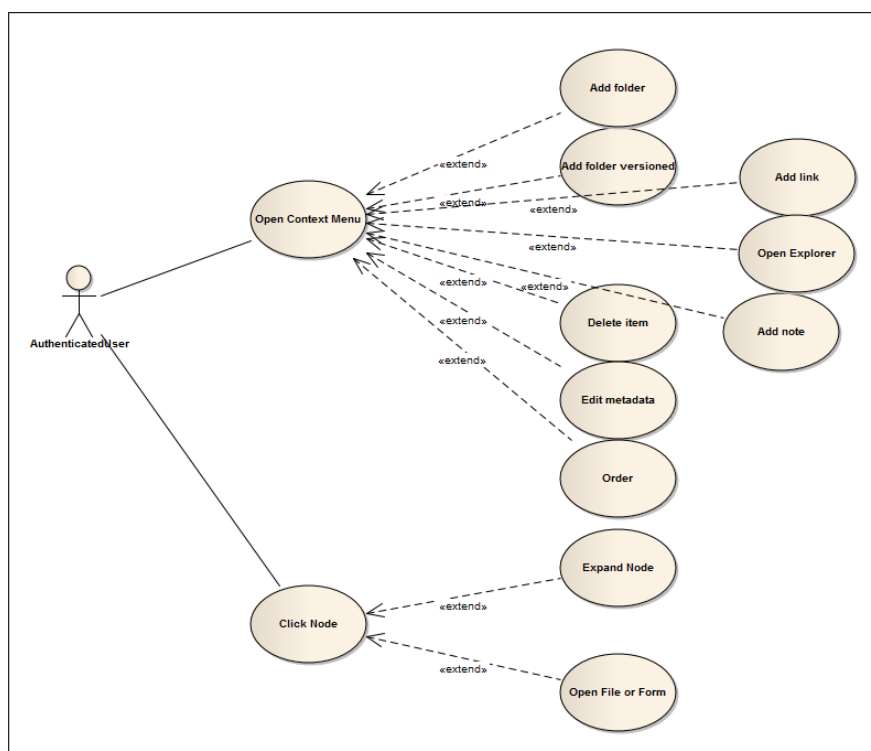


Figura 3.10: Diagrama casos do módulo Explorer+.

As tabelas de descrição dos casos de uso, apresentados na figura 3.10, estão nos anexos A.1.4.

A forma de executar a maioria dos casos de utilização é com recurso ao menu de contexto sobre os conteúdos. A figura 3.11 apresenta um exemplo de iniciação do menu de contexto sobre uma pasta.

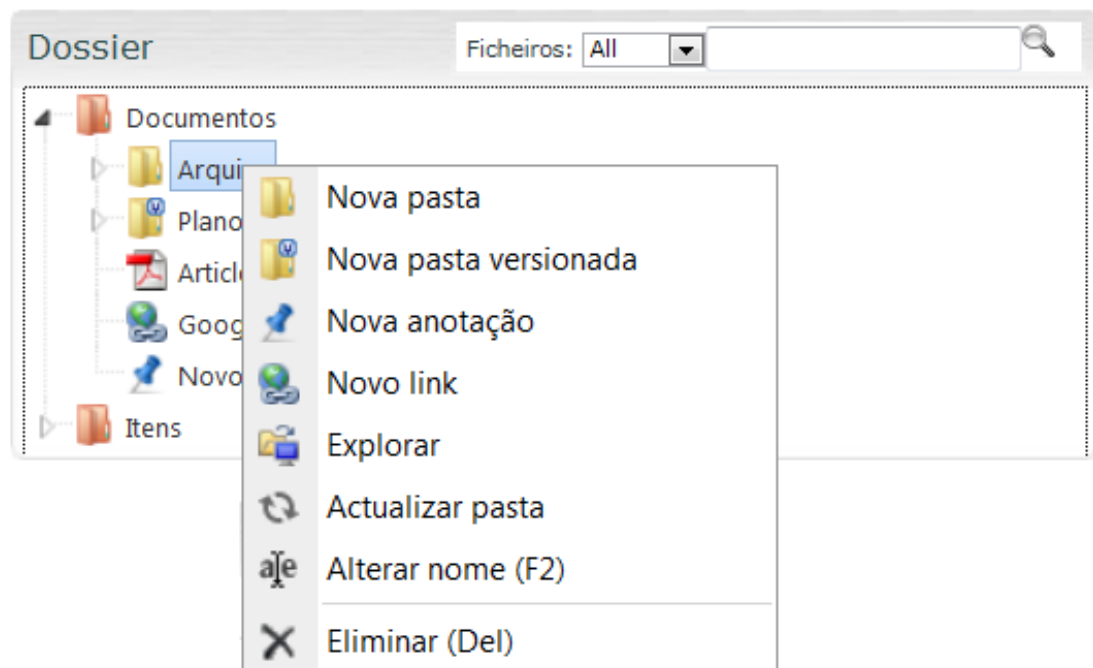


Figura 3.11: Menu de contexto do módulo Explorer+ sobre uma pasta

3.3 Desenho do módulo

Neste ponto, é apresentada a arquitectura e o modelo de dados utilizado para desenvolver o módulo Explorer+.

3.3.1 Arquitectura

A arquitectura do módulo desenvolvido divide-se em três níveis, camada de acesso a dados, camada lógica e por fim a camada de apresentação.

A camada de acesso a dados é responsável por gerir os conteúdos do *dossier* associados ao Explorer+, e ainda o acesso aos registos em base de dados da instância e da entidade associadas a página relacionada.

Módulo Explorer+

A camada lógica é o próprio módulo Explorer+. Gere todas as acções possíveis sobre os conteúdos associados ao módulo.

A camada de apresentação é responsável por apresentar a página com o módulo Explorer+.

A arquitectura em camadas do módulo é apresentada na figura 3.12.

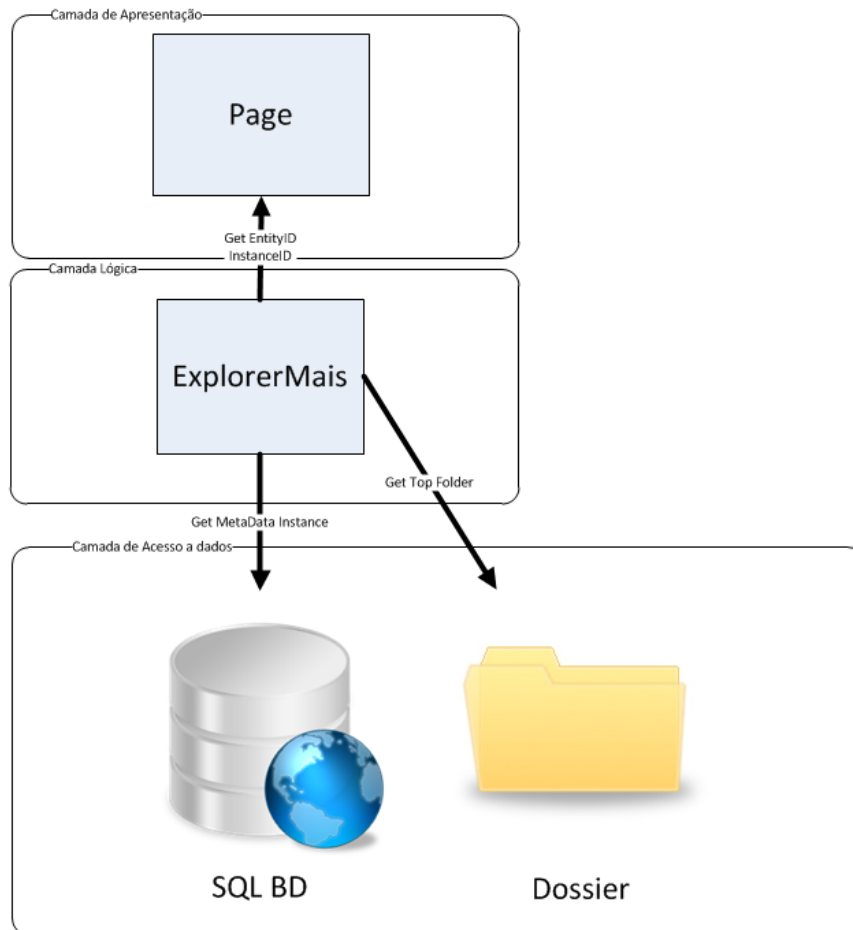


Figura 3.12: Arquitectura do módulo Explorer+

3.3.2 Modelo de dados

O modelo de dados do módulo Explorer+ está representado na figura 3.13.

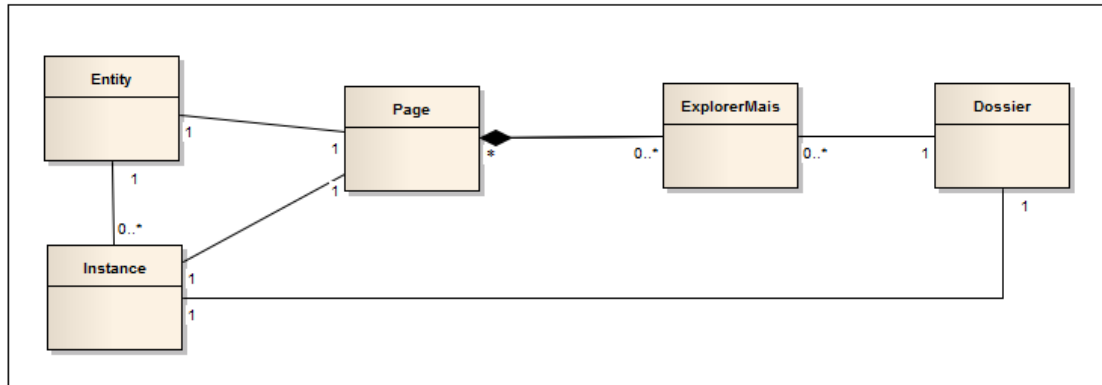


Figura 3.13: Modelo de dados do módulo Explorer+.

Ao nível da modelação de dados, cada módulo Explorer+ é integrado numa *Web Part* de uma página. E cada página está associada a uma entidade e uma instância, desse modo obtém os meta-dados necessários para identificar o *dossier* associado ao módulo Explorer+.

3.4 Implementação

Este ponto, descreve a implementação das funcionalidades associadas ao módulo Explorer+. Preferiu-se mostrar esquemas onde se explica o funcionamento dos demais trechos de códigos que estão por detrás das funcionalidades, em vez de expor o código desenvolvido. Assim torna-se mais fácil transmitir as soluções escolhidas na implementação.

Inicialização do módulo

O Explorer+, em primeiro lugar identifica a pasta a partir da qual o módulo deverá apresentar os conteúdos. A identificação da pasta, passa por uma análise do *url* da página. Essa análise, inicialmente verifica o tipo de página que chama o controlo e assim identifica o tipo de item, depois de identificado o tipo de item e a respectiva tabela em base de dados a que se refere o ID no *url*, identifica-se o registo na tabela e obtém-se o *ID* da pasta sobre a qual o módulo Explorer+ se inicia.

Apresentação hierárquica de conteúdos

O módulo Explorer+ é uma vista em árvore de conteúdos de uma pasta. A árvore utilizada para desenvolver o módulo Explorer+ é a *RadTreeView* [Tel02]. A vista em árvore é a forma de hierarquizar os conteúdos.

Gestão de permissões

A implementação da gestão de permissões sobre os conteúdos no módulo Explorer+, foi efectuada com recurso a um tipo de pasta, *Managed Folder*, este tipo de pasta apresenta dois campos em meta-data para gestão de permissões, esses campos são *EditOwn* e *EditSub*. As quatro combinações possíveis dos campos *EditOwn* e *EditSub* dão origem a diferentes menus de contexto sobre os conteúdos no módulo Explorer+.

Controlo de versões

A implementação das pastas versionadas no módulo, é tratada recorrendo a um campo *Version* nas pastas do tipo *Managed Folder*. Se o campo *Version* não for *Null*, o conteúdo do campo é o nome do ficheiro versionado. A pasta versionada apresentará o ficheiro mais recente, isto é, com índice de versão maior.

A figura 3.14 apresenta a estrutura do nome dos ficheiros dentro de uma pasta versionada.

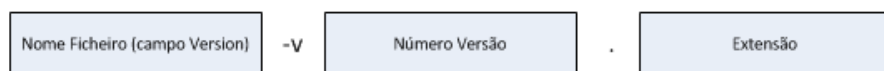


Figura 3.14: Estrutura do nome dos ficheiros em pasta versionada.

Este sistema manual de controlo de versões, depende de um nível mais elevado de atenção por parte do utilizadores, para funcionar correctamente. Os ficheiros dentro de uma pasta versionada, que o nome não siga a estrutura anteriormente apresentada são ignorados.

Notas e ligações

Os ficheiros do tipo nota são implementados com recurso a ficheiros XML. Uma nota é um ficheiro XML com dois campos, título e nota. O nome do ficheiro é o título (do campo título no ficheiro XML) mais *.note*. No módulo Explorer+ a extensão *.xml* não é apresentada.

A figura 3.15 é um exemplo de um ficheiro XML representativo de uma nota.

```
<?xml version="1.0"?>
<NOTE>Nota associada a pasta
<título>nota 2</título>
<nota>nota2</nota>
</NOTE>
```

Figura 3.15: Exemplo de uma nota em XML.

O *click* sobre um ficheiro tipo nota dá origem a uma *popup* de apresentação da nota.

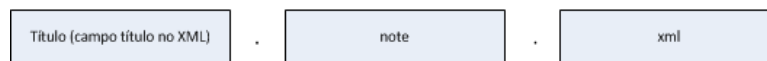


Figura 3.16: Estrutura do nome dos ficheiros nota.

Os ficheiros do tipo ligação são implementados com recurso a ficheiros XML. Uma ligação é um ficheiro XML com dois campos, título e ligação. O nome do ficheiro é o título (do campo título no ficheiro XML) mais *.link*. No módulo Explorer+ a extensão *.xml* não é apresentada. A figura 3.17 é um exemplo de um ficheiro XML representativo de uma ligação.

```
<?xml version="1.0"?>
<LINK>Ligação
<título>RADTREE</título>
<url>http://www.telerik.com/products/aspnet-ajax/treeview.aspx</url>
</LINK>
```

Figura 3.17: Exemplo de uma ligação em XML.

O *click* sobre um ficheiro tipo ligação faz o *browser* abrir uma nova aba com *url* da ligação.



Figura 3.18: Estrutura do nome dos ficheiros ligação

Ordenação de pastas

A ordenação manual de pastas, recorre a um campo *Ordering* nas pasta do tipo *Managed Folder*. As pastas com esse campo vazio são ordenadas alfabeticamente pelo nome.

O módulo ordena as pastas, ao mesmo nível na estrutura, primeiro pelo campo *Ordering* e depois alfabeticamente pelo nome.

Drag and Drop

O *Drag and Drop* de ficheiros e pastas, na árvore de conteúdos do módulo Explorer+, tem que verificar três condições.

Primeiro, a pasta de destino dos conteúdos a mover tem que permitir largar conteúdos sobre o mesmo (*Drop = True*), senão o sistema não permite a movimentação dos conteúdos para essa pasta. No caso do destino dos conteúdos a mover ser um ficheiro é necessário identificar a pasta a que pertence esse ficheiro, e se pasta o permitir, a movimentação será para essa pasta. Os conteúdos a mover para poderem ser movidos necessitam ter como característica, *Drag = True*.

Depois, só é possível entre pastas e ficheiros com nomes diferentes, pois não é possível numa mesma pasta existirem ficheiros com a mesma identificação, como também acontece num sistema de ficheiros.

Por último, o sistema tem que verificar as permissões das pastas do tipo *Managed Folder*. A pasta de destino tem que possuir o campo *EditSub* a verdadeiro, e as pastas dos conteúdos a mover têm que possuir o campo *EditOwn* a verdadeiro.

Upload e Download de conteúdos

Uma funcionalidade importante do Explorer+ é o *Drag and Drop* directamente do ambiente de trabalho para uma pasta numa biblioteca *Sharepoint*. Para esse efeito o menu de contexto do Explorer+, oferece a opção explorar, que quando seleccionada abre uma janela *explorer* posicionada na pasta do servidor seleccionada no módulo. Sendo então possível efectuar *Upload* ou *Download* de conteúdos entre pasta no servidor e o ambiente

de trabalho. Como o módulo prevê essa funcionalidade algumas condições ao nível das permissões das pastas terão de ser satisfeitas.

A figura 3.19 ilustra o resultado da opção explorar sobre uma pasta na árvore de conteúdos do módulo Explorer+.

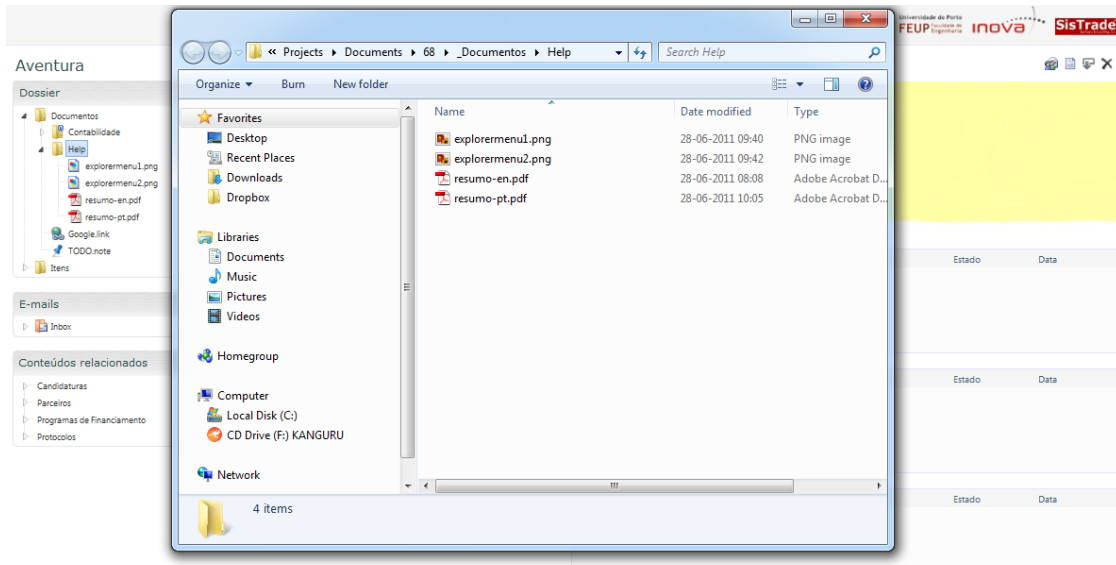


Figura 3.19: Exemplo de acção explorar sobre pasta no módulo Explorer+

As condições a ser satisfeitas são tratadas ao nível dos *EventHandlers* disparados no momento de movimentação de conteúdos entre ambiente de trabalho e pasta no servidor.

Os *EventHandlers* verificam as permissões das pastas do tipo *Managed Folder*. As condições a verificar são:

- A pasta de destino - campo *EditSub* a verdadeiro;
- As pastas dos conteúdos a mover - campo *EditOwn* a verdadeiro.

Pesquisa por texto

A pesquisa por texto sobre os conteúdos do módulo Explorer+, limita-se a apresentar os ficheiros que contenham no nome o texto inserido no campo de pesquisa.

Filtro por tipo de conteúdo

A filtragem dos ficheiros a apresentar pelo módulo Explorer+ consiste na verificação da extensão do ficheiro. Se a extensão do ficheiro nos conteúdos corresponde ao seleccionado na *drop*, esse ficheiro é adicionado à árvore do módulo. Em caso de seleccionar todos os itens na *drop*, não faz essa verificação e apresenta todos os ficheiros nos conteúdos.

Capítulo 4

Módulo Viewer

O segundo módulo desenvolvido no contexto desta dissertação designado por Viewer. Tal como o anterior é um componente reutilizável, que permite apresentar ou aceder na página de uma dada instância aos *dossiers* das instâncias que lhe estão associados. Por exemplo, se está numa página de um projecto, para poder aceder aos parceiros associados ao mesmo, a página necessita incluir um módulo Viewer com um nó configurado para visualizar parceiros. A expansão do nó começa por apresentar as instâncias de parceiros relacionadas com o projecto, e a expansão de uma destas instâncias apresenta o respectivo *dossier*, conforme esquematizado na figura 4.1.

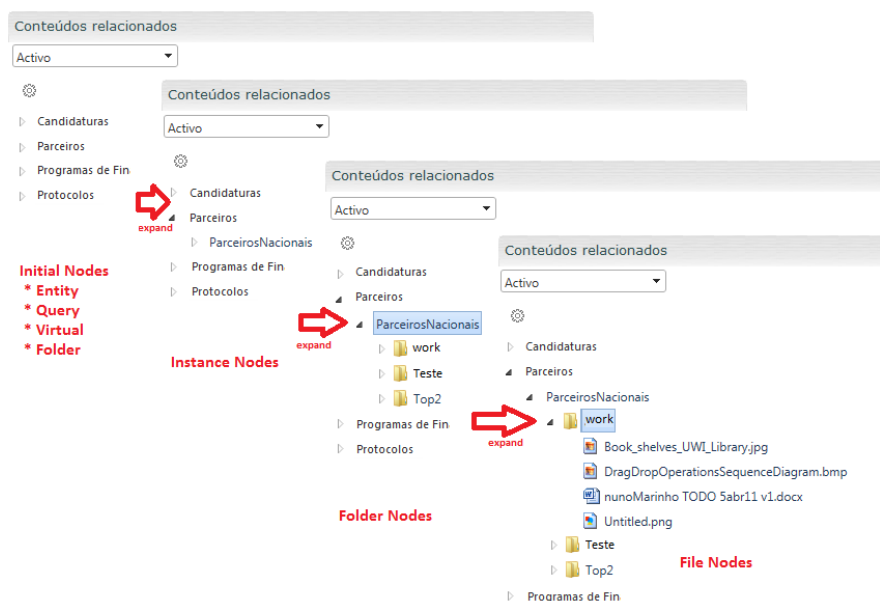


Figura 4.1: Conteúdos de instâncias relacionadas com a página

Assim, os conteúdos são apresentados em árvore, em que os nós de primeiro nível correspondem às entidades tipo, os de segundo às instâncias dessas entidades e os de terceiro nível (e restantes) aos conteúdos do *dossier*. Com a expansão até ao terceiro nível, é possível visualizar e gerir o *dossier* associado à instância de segundo nível. Além de permitir a visualização de conteúdos, na versão actual, o módulo Viewer permite mover conteúdos entre pastas por *Drag and Drop*.

A figura 4.1 representa a sequência de acções necessárias para visualizar e gerir conteúdos de instâncias relacionadas com a página.

4.1 Exemplo de aplicação módulo: Página projectos

O módulo Viewer é um módulo reutilizável, pois poderá ser aplicado a qualquer página da plataforma. Para ilustrar o funcionamento deste módulo considera-se a página de projectos, já utilizada anteriormente para exemplificar o módulo Explorer+.

As entidades relevantes, no âmbito da página de projectos, são as representadas na figura 4.2.

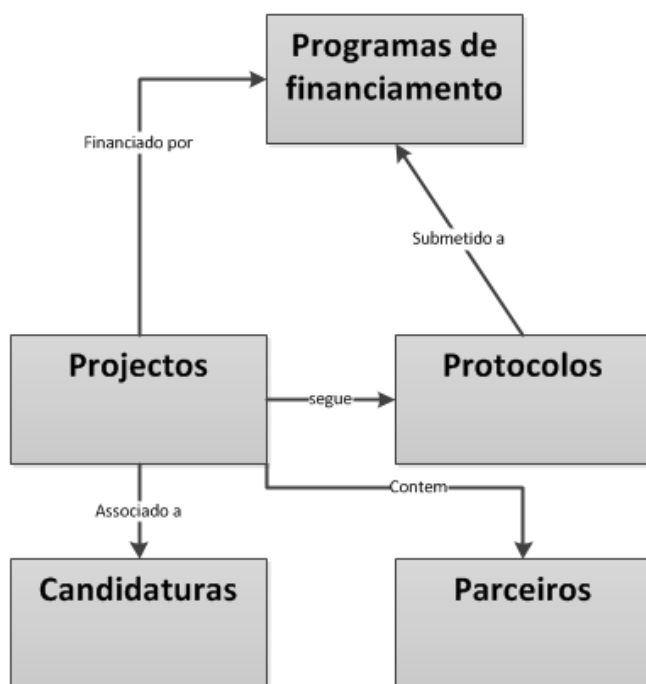


Figura 4.2: Entidades relacionadas com projectos

De acordo com o modelo apresentado na figura 4.2, as instâncias de projectos estão associadas a instâncias de parceiros, protocolos, programas de financiamento e candidaturas.

Como já referido, os nós de primeiro nível correspondem às entidades tipo a visualizar. Estes nós iniciais do Viewer são configuráveis, através de um botão de configuração associado ao módulo. A expansão destes nós dá origem às instâncias relacionadas com a página, da entidade tipo associada ao nó.

Para aceder aos *dossiers* das instâncias destas entidades a partir da página de projectos, deverá ser instalado um Viewer contendo um nó configurado para cada entidade tipo.

A figura 4.3 ilustra o resultado da expansão de um nó inicial dos nós de instâncias relacionadas.



Figura 4.3: Instâncias relacionadas

Os nós de segundo nível correspondem às instâncias relacionadas. A expansão de instâncias devolve a pasta raiz do *dossier* associado à instância. A figura 4.4 ilustra o resultado da expansão de um nó de segundo nível.



Figura 4.4: *Dossier* associado à instância

Para preencher cada nível da árvore de conteúdos do Viewer é necessário aceder a diferentes repositórios de informação, conforme esquematizado na figura 4.5.

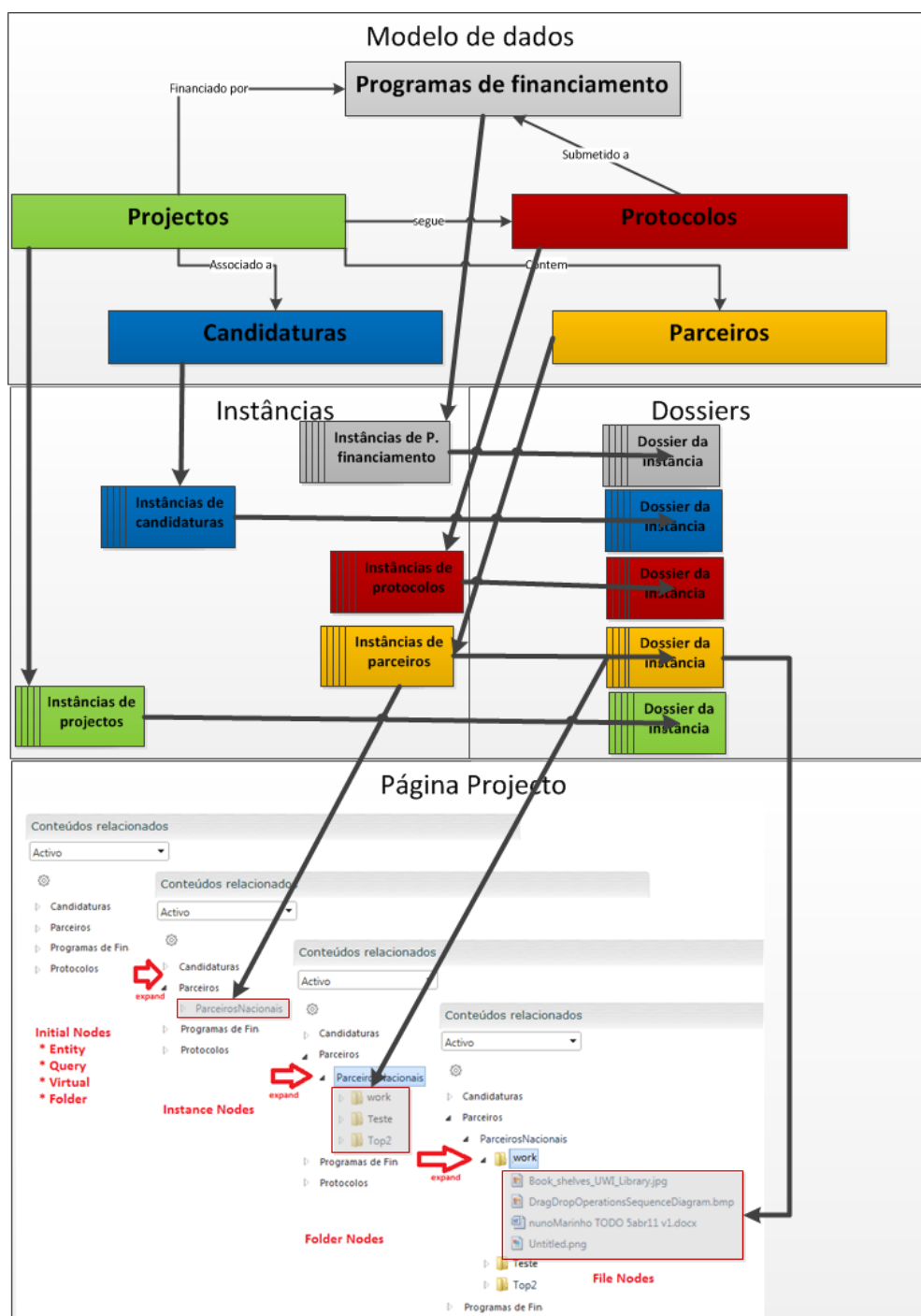


Figura 4.5: Diferentes repositórios de informação

Outra característica fundamental deste módulo, é o facto de poder ser utilizado em páginas de diferentes entidades requerendo o mínimo de configuração, por tanto para

instalar o Viewer numa página, apenas é necessário especificar o conjunto de nós de topo e para cada nó a entidade tipo que lhe está associada.

A configuração do Viewer não requer qualquer conhecimento da implementação física dos repositórios de informação, que tanto podem residir em listas *Sharepoint*, bibliotecas, sistemas de ficheiros externos, bases de dados e *e-mails*. Por tanto, para garantir esta independência, entre a configuração e a localização física dos dados foi criado um gestor de entidade para cada entidade tipo. Este gestor recebe as acções do utilizador na interface e retorna os resultados num formato normalizado para serem consumidos pela camada de apresentação. A figura 4.6 representa o relacionamento entre a configuração, a localização física dos dados e os gestores de entidade.

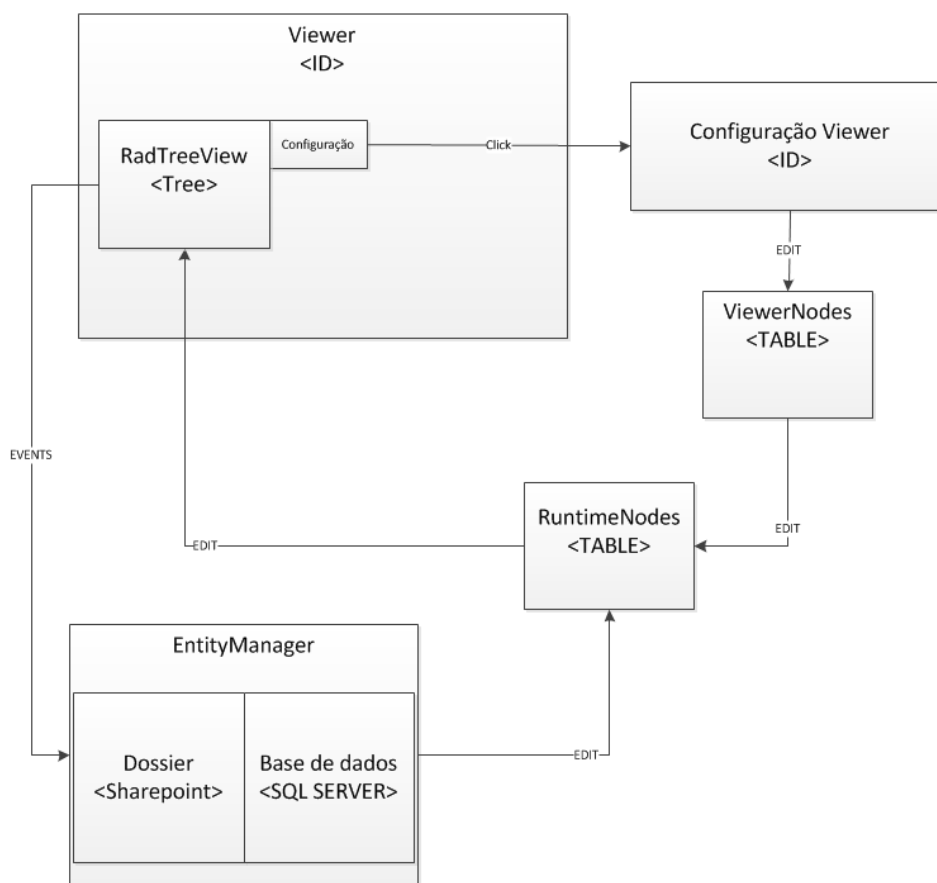


Figura 4.6: Relacionamento entre configuração, dados e gestores de entidade

Embora a finalidade principal do Viewer, seja permitir navegar nas entidades, instâncias e *dossiers* relacionados com a página actual.

O Viewer oferece ainda um conjunto adicional de funcionalidades, entre as quais as seguintes:

- A possibilidade de gerir os conteúdos, nomeadamente mover conteúdos entre pastas (*drag and drop*). O drag and drop pode ser dentro do mesmo Viewer, ou entre Viewers diferentes na mesma página.
- A possibilidade de configurar diferentes tipos de nós de topo, como nós entidade, nós *folder*, nós *query* e nós virtuais.

Depois desta apresentação dos principais conceitos associados ao Viewer, é efectuada a especificação dos requisitos, o desenho, o modo de tratamento de eventos e finalmente os aspectos de implementação do módulo.

4.2 Análise requisitos

A análise de requisitos consiste no levantamento e especificação do conjunto de características que o módulo deverá contemplar para corresponder às necessidades dos seus futuros utilizadores. Os requisitos podem ser classificados em funcionais, e não-funcionais.

4.2.1 Requisitos não funcionais

Os requisitos não-funcionais identificados são os mesmos que os especificados no ponto 3.2.1 para o módulo ExplorerMais.

4.2.2 Requisitos funcionais

A especificação dos requisitos para este módulo foi fundamentada nos estudos sobre trabalho baseado em conhecimento intensivo, gestão de conteúdos de informação e nas soluções existentes no mercado, abordadas no capítulo 2.

O módulo Viewer tem como finalidade a apresentação de conteúdos de diferentes instâncias, relacionadas com a instância associada à página.

Configuração do módulo

A configuração *in place* do módulo, consiste em desenvolver uma interface de configuração dos nós iniciais do Viewer e essa interface deve ser facilmente acessível através do módulo Viewer.

A interface de configuração dos nós do Viewer disponibilizará diversas funcionalidades ao nível da gestão desses mesmos nós. Essas funcionalidades são:

- Adicionar nós;
- Eliminar nós;
- Editar meta-data de nós.

A configuração *in place* do módulo Viewer serve para gerir a tabela de nós iniciais associados ao Viewer, isto é, a tabela *ViewerNodes*.

Suportar diferentes tipos de nós

Os diferentes tipos de nós suportados pelo módulo são:

- Nó *Virtual*;
- Nó *Query*;
- Nó *Entity*;
- Nó *Folder*.

O nó *Virtual* tem como funcionalidade agrupamento/estruturação de nós, isto é, serve como separador por a agrupar de forma mais conveniente os nós no módulo. Por exemplo, cria-se um nó *Virtual* com o nome parceiros e associa-se ao mesmo dois nós *Query*, um nó com o nome parceiros nacionais e com a *Query* nesse sentido, e outro, para parceiros internacionais.

O nó *Query* é um nó que, além de uma entidade tipo, permite associar uma *query* que especifica condições de filtragem, agrupamento e ordenação das instâncias apresentadas. A implementação deste nó requer a implementação de um modelo que simplifique a criação de *queries* sobre entidades relacionadas, sem conhecer o modelo de dados físico. É necessário criar uma abstracção sobre o modelo de dados físico.

O nó *Entity* tem como funcionalidade relacionar a entidade associada ao nó com a entidade associada a página que contem o módulo. A expansão de um nó *Entity* apresenta as instâncias relacionadas com a instância associada à página.

O nó *Folder* apresenta a pasta relacionada com o nó. A expansão de um nó *Folder* apresenta os conteúdos da pasta.

Suportar vários Viewers

Outro requisito, é permitir a presença de mais que um viewer na mesma página ou mesmo na mesma *Web Part*, com recurso a um gestor de *Web Part*.

Drag and Drop

O módulo permitirá o *Drag and Drop* de pastas e ficheiros dos diferentes conteúdos relacionados, mesmo esses conteúdos se encontrando em diferentes bibliotecas ou *sites*. O *Drag and Drop* é apenas possível entre pastas e ficheiros.

Filtragem e pesquisa de conteúdos

Por último, tal como no ExplorerMais, o Viewer tem que fornecer ferramentas de pesquisa sobre os conteúdos e/ou filtragem por tipo de conteúdos.

4.2.3 Casos de utilização

A figura 4.7 apresenta o diagrama de casos de uso do módulo Viewer.

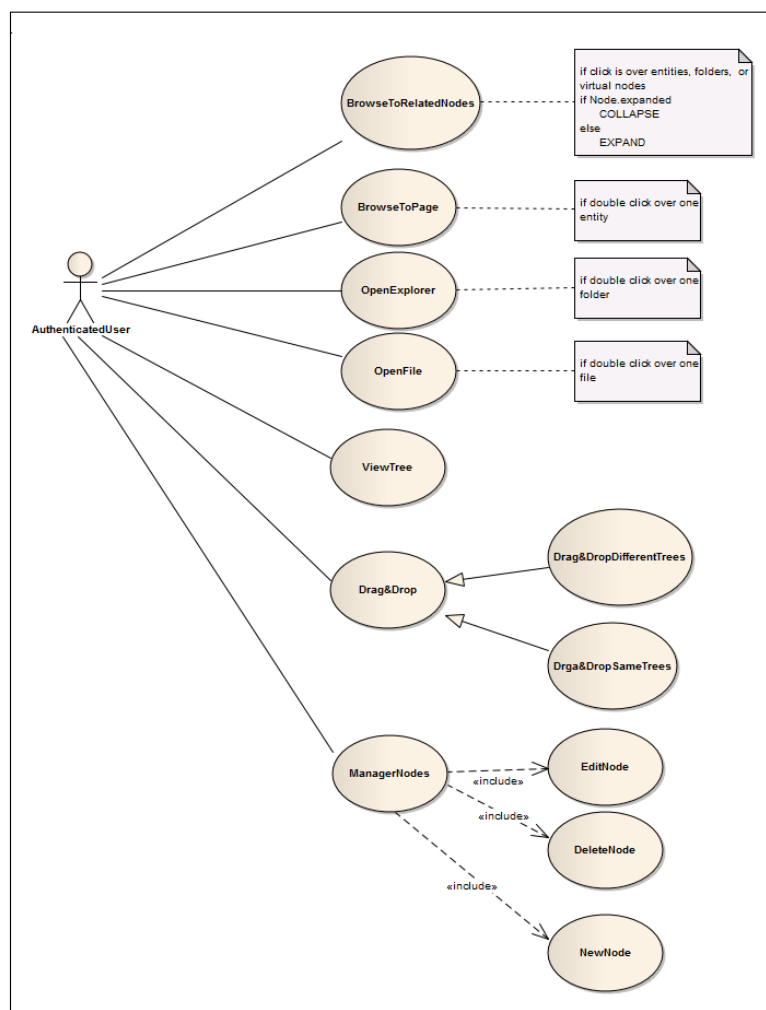


Figura 4.7: Diagrama casos do módulo Viewer.

As tabelas de descrição dos casos de uso, apresentados na figura 4.7, estão nos anexos B.5.0.1.

4.3 Modelo de dados e arquitectura

Neste ponto, é apresentado o modelo de dados e a arquitectura do módulo.

4.3.1 Modelo de dados

O modelo de dados é o representado na figura 4.8.

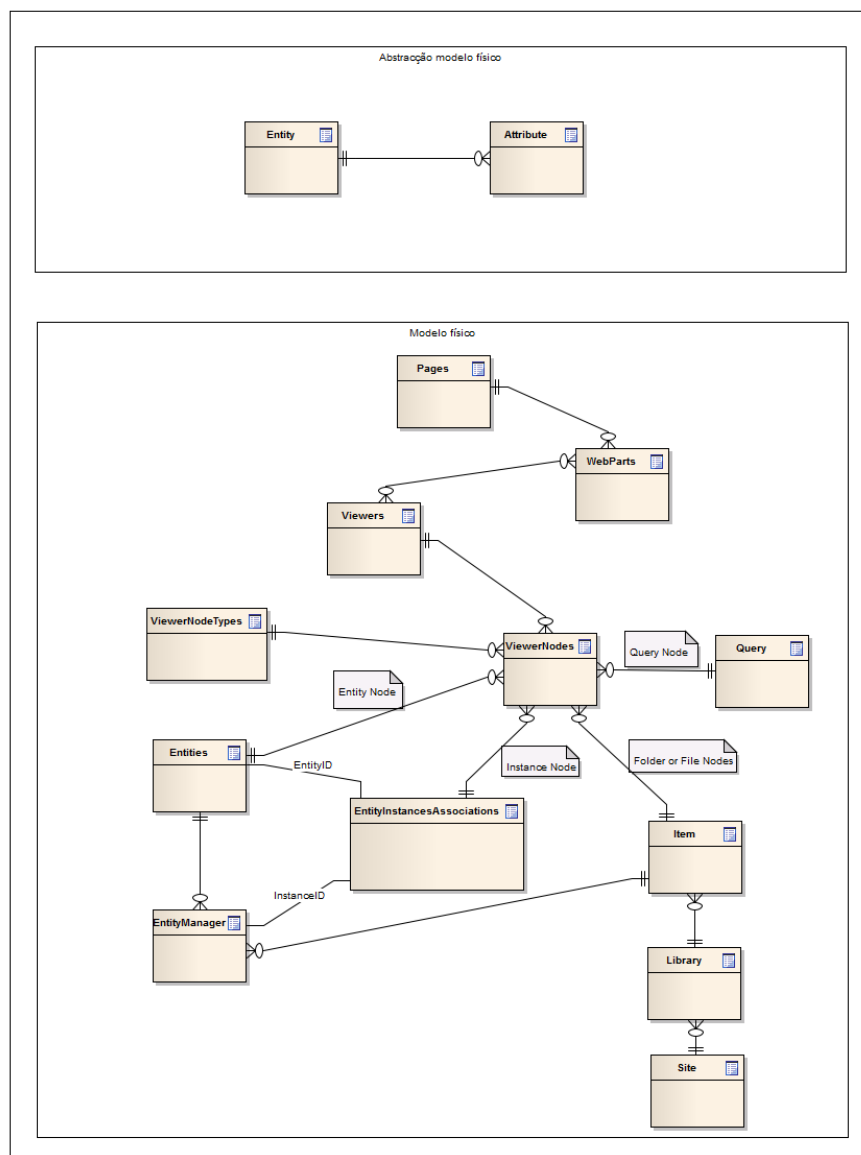


Figura 4.8: Modelo de dados do módulo Viewer

Abstracção modelo físico

Para permitir especificar a configuração do Viewer sem necessidade de conhecer a estrutura física dos repositórios de informação foi criada uma abstracção sobre o modelo físico da informação centrada em dois conceitos, entidades e atributos, que serão utilizados na especificação das *queries* do Viewer.

O conceito entidade contém as entidades presentes na aplicação e suas características. O conceito atributo representa os atributos associados às entidades. Um atributo tem que estar sempre associado a uma entidade. Estes conceitos servem apenas para guardar meta-dados sobre as entidades e respectivos atributos associados da plataforma.

Modelo físico

O modelo físico dos dados esquematiza a associação entre os diversos repositórios de informação. O módulo *Viewer* está integrado numa *WebPart* e tem associado nós de topo. A cada nó está associada uma entidade, que por sua vez está associada a um gestor. O gestor de cada entidade tem acesso aos diversos repositórios de informação associados a uma dada instância de uma entidade, dessa forma é possibilitada a gestão e visualização dos mesmos no Viewer.

O conceito mais importante, no âmbito do módulo Viewer, é o *EntityInstancesAssociations*. Neste repositório são guardadas as relações entre as diversas instâncias das várias entidades da plataforma. A expansão de um nó de topo utiliza o repositório *EntityInstancesAssociations* para determinar as instâncias associadas.

4.3.2 Arquitectura

A figura 4.9 representa a arquitectura em camadas do módulo.

A arquitectura do módulo Viewer, divide-se em três camadas, camada de acesso a dados, camada lógica e camada de apresentação.

Camada de apresentação

Os módulos da camada de apresentação são responsáveis apresentação da *Web Part* com o(s) respectivos Viewer(s), na página. A criação e formatação da árvore, botão de configuração associado, painel e tabela, e ainda a definição de estilos, títulos e cores de toda a *Web Part* em questão, é efectuada pelas classes associadas a esta camada.

A camada de apresentação contém a classe *WebPartManager* e a classe *Viewer*.

A classe *WebPartManager* tem como função a gestão dos controlos numa *WebPart*. Nesta classe é criado o Viewer com as respectivas definições, é possível criar outros controlos (inclusive outros Viewers). Os controlos criados no mesmo *WebPartManager* podem interagir, e é nesse contexto que surgiu a necessidade de criar esta classe.

Módulo Viewer

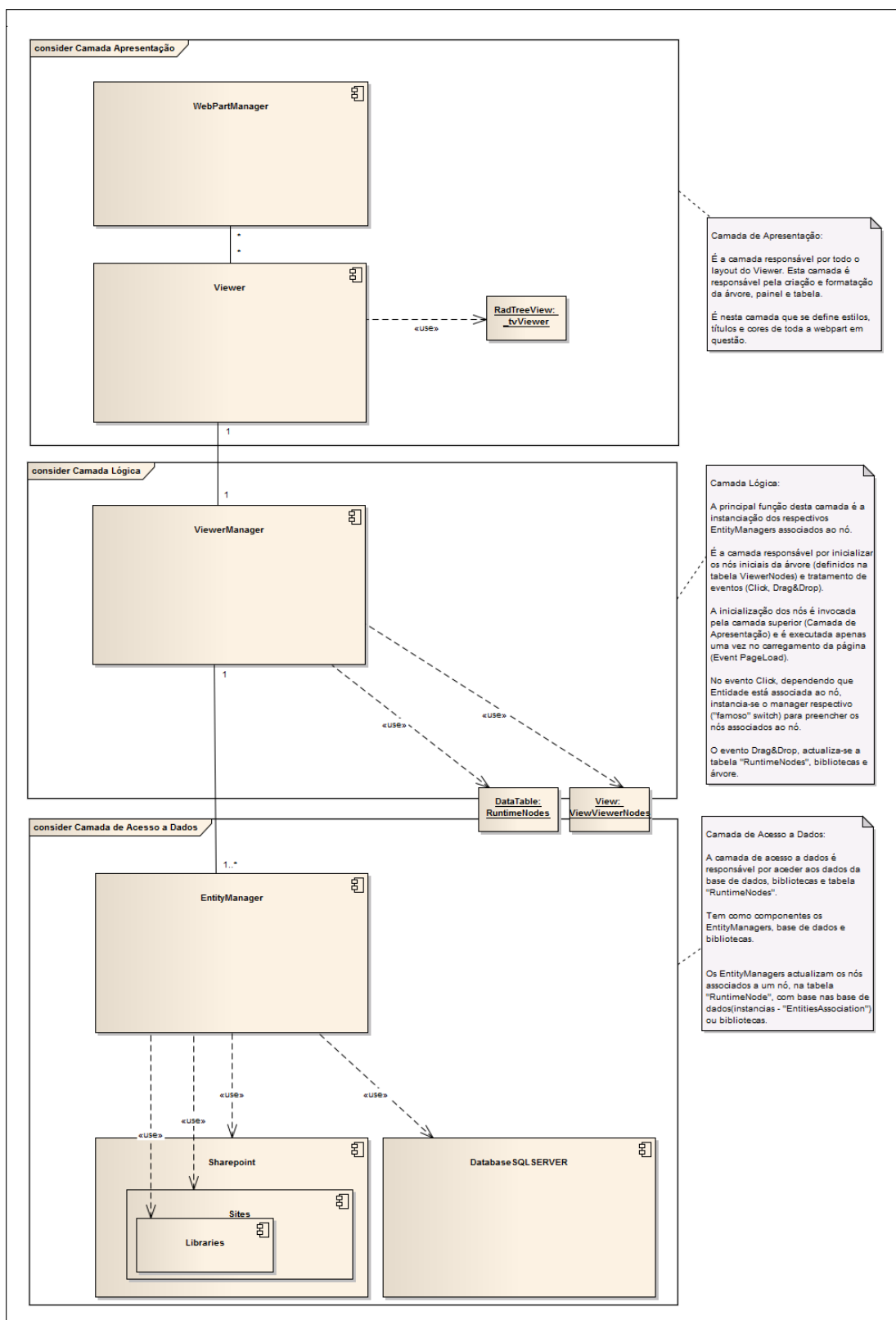


Figura 4.9: Arquitectura em camadas utilizada pelo módulo Viewer.

A classe *Viewer* é a definição do próprio módulo Viewer. Os eventos sobre o módulo são recebidos por esta classe e passados ao gestor da mesma, isto é, ao *ViewerManager*.

Camada lógica

Os módulos da camada lógica são responsáveis por inicializar os nós iniciais da árvore, instanciar os respectivos *EntityManagers* associados a cada nó, e por tratar eventos *Click* e *Drag and Drop* sobre nós.

A instanciação dos respectivos *EntityManagers* associados a cada nó, utiliza um *switch* de selecção para determinar o *EntityManager* associado e instanciar o mesmo. O evento *Click* é utilizado o *EntityManager* instanciado para preenchimento de nós da árvore.

A inicialização dos nós iniciais da árvore, definidos no repositório *ViewerNodes* associado ao Viewer também é da responsabilidade dos módulos desta camada. A inicialização dos nós é invocada pela camada superior (camada de apresentação) e é executada apenas uma vez no carregamento da página. O modelo de dados utilizado pelo repositório *ViewerNodes* é o esquematizado na figura 4.10.

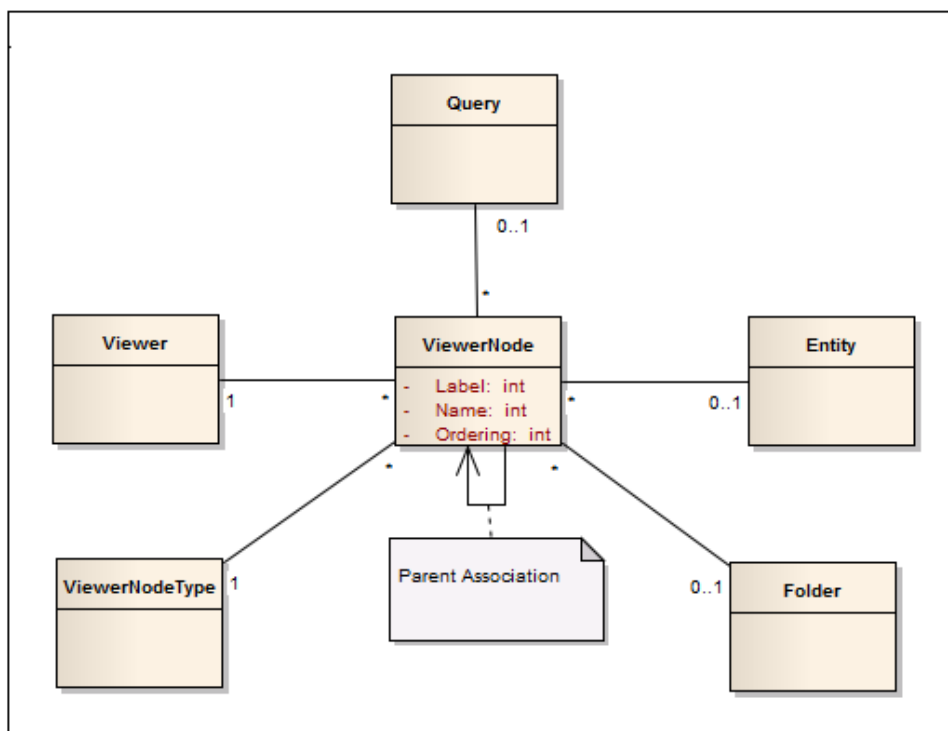


Figura 4.10: Modelo de dados *ViewerNodes*

O evento *Drag and Drop*, actualiza o repositório *RuntimeNodes* em variável de sessão, bibliotecas e árvore. O *RuntimeNodes* está a ser constantemente actualizado, de forma a acompanhar a árvore de conteúdos com meta-data sobre os nós da mesma. Este evento é tratado pelos módulos desta camada.

A camada lógica é constituída pela classe *ViewerManager*, e ainda pelos repositórios *ViewerNodes* e *RuntimeNodes*.

O conceito mais importante a reter na camada lógica é a existência de um gestor associado ao Viewer, o *ViewerManager*. O *ViewerManager* faz a instanciação dos nós iniciais do módulo Viewer e trata os eventos ocorridos sobre o Viewer, criando todos os *EntityManager* necessários para a execução dos mesmos. Esta classe contém um *switch* de selecção do *EntityManager*, para determinar qual o *EntityManager* a ser criado, para o preenchimento da árvore de conteúdos relacionados.

Camada de acesso a dados

Os módulos da camada de acesso a dados são responsáveis pelo acesso aos diversos repositórios de informação.

O repositório *RuntimeNodes* vai guardando a meta-data associada a cada nó carregado na árvore de visualização. A localização do mesmo na arquitectura é entre a camada lógica e a camada de acesso a dados, porque as duas camadas necessitam de aceder ao repositório.

O repositório *ViewViewerNodes* está localizado entre a camada lógica e a camada de acesso a dados, porque a camada lógica necessita de carregar os nós iniciais associados e a camada de acesso a dados necessita gerir esses mesmos nós. Assim, é necessário que esse repositório esteja acessível às duas camadas.

A camada lógica é constituída pela classe *EntityManager*, que utiliza os diversos repositórios de informação sempre que necessário.

A classe *EntityManager* representa o gestor de cada entidade na lógica de negócio. É necessário um gestor próprio para cada entidade, porque cada entidade está relacionada com diferentes repositórios de informação com diferentes formas de acesso.

4.4 Tratamento de eventos

Neste ponto, descreve-se a forma utilizada pelo módulo para tratar os eventos sobre a árvore de conteúdos. Preferiu-se mostrar esquemas onde se explica o funcionamento dos demais trechos de códigos que estão por detrás das funcionalidades, em vez de expor o código desenvolvido.

Evento *load*

O evento *load* no módulo Viewer consiste na criação do interface do módulo e o preenchimento dos nós iniciais associados ao Viewer.

Inicialmente, o evento dá origem à criação de um objecto *Viewer* passando o controlo do evento para o objecto. O objecto acede ao *url* da página, retira o identificador da entidade e da instância. Em seguida, é(são) identificado(s) o(s) identificador(es) do(s) Viewer(s) associados a *Web Part*. Para terminar, o objecto *Viewer* passa o controlo do evento ao *ViewerManager* criado com os dados determinados nas tarefas anteriores. O objecto *Viewer* cria um *ViewerManager* por cada *Viewer* associado à *Web Part*.

O objecto *ViewerManager* é inicializado com a configuração anterior, e identifica a vista *ViewViewerNodes* associada. Em seguida, é criada a tabela *RunTimeNodes* em variável de sessão e é efectuada a leitura dos nós associados ao identificador do *Viewer*. Por último, a tabela *RunTimeNodes* e a árvore de conteúdos são actualizadas com nós identificados anteriormente.

Neste contexto, é perceptível que a vista *ViewViewerNodes* com os nós associados ao Viewer em questão, é o principal repositório de informação. A partir da leitura dos dados da vista *ViewViewerNodes* é que todos os outros repositórios são carregados. Também é possível reter que o preenchimento dos nós iniciais é efectuado pelos métodos da classe *ViewerManager*.

A figura 4.11 esquematiza a sequência de métodos invocados pelas diferentes classes associadas ao Viewer. Enquanto que a figura 4.12 descreve o algoritmo associado ao tratamento do evento *Load*, descrito no parágrafos anteriores.

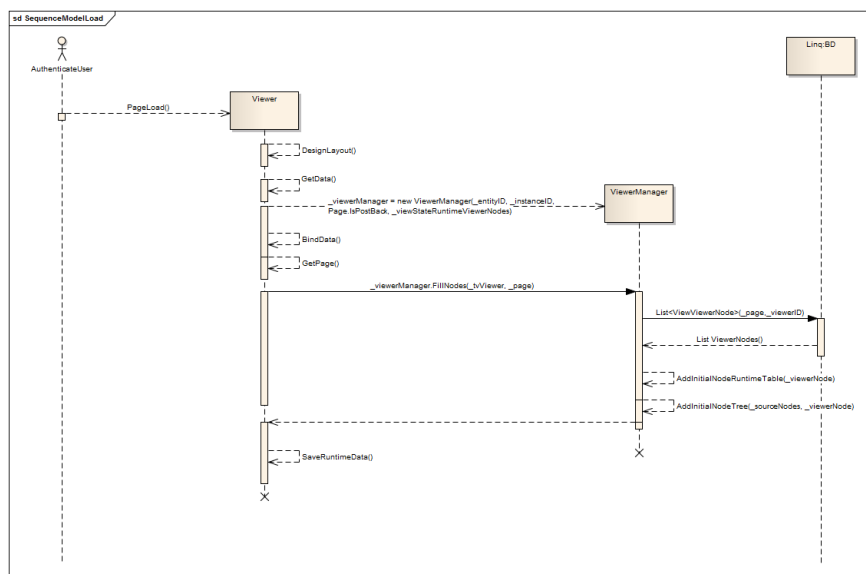


Figura 4.11: Diagrama de sequência evento *Load*

Módulo Viewer

No anexo B.3 estão os restantes diagramas de análise do evento de carregamento de nós iniciais.

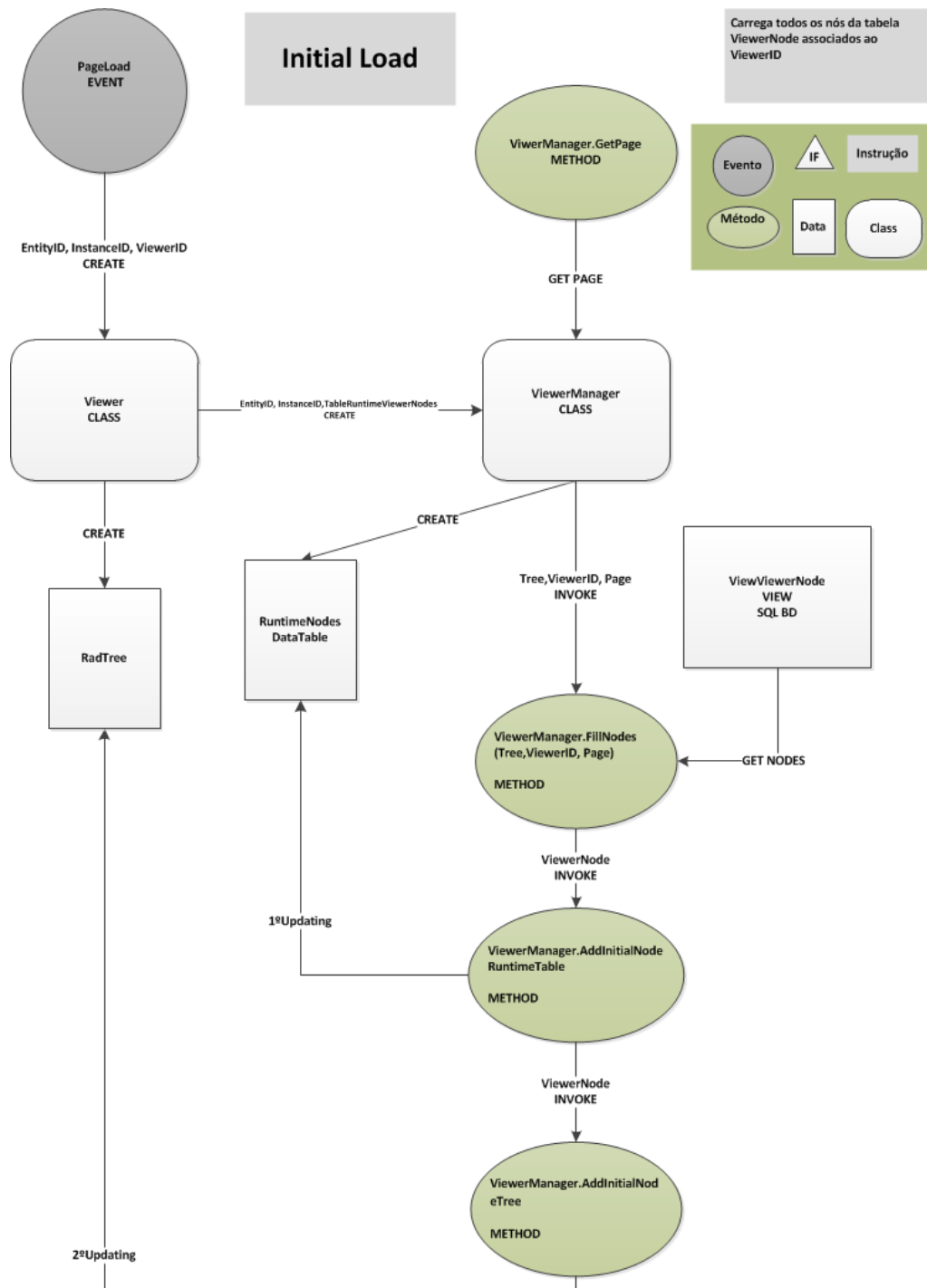


Figura 4.12: Esquema representativo do tratamento do evento *load*

Evento Click

O evento *click* carrega todos os nós associados ao nó clicado, dependendo do tipo de nó.

O objecto *Viewer* recebe o evento e limita-se a passar o controlo do evento para o *ViewerManager* associado. O *ViewerManager* acede aos meta-dados associados ao nó clicado, e determina o *EntityManager* (criando-o se necessário) relacionado. A responsabilidade em relação ao tratamento do evento passa a ser do *EntityManager* identificado.

O *EntityManager* invoca o método de preenchimento de nós associados, conforme os meta-dados associados ao nó clicado. O preenchimento poderá ser ao nível das instâncias associadas ou ao nível do *dossier* associado.

Se os nós a preencher estão ao nível das instâncias, é necessário aceder a vista *ViewEntitiesAssociation*. O carregamento de instâncias verifica o tipo de nó e utiliza a vista *ViewEntitiesAssociation*. Em caso de nó *Query* é necessária uma análise da *Query* e a consequente execução da mesma. Nos outros casos, a instanciação é apenas baseada na vista *ViewEntitiesAssociation*. Este tipo de preenchimento, necessita de um gestor de entidade, pois o *EntityManager* é que possui os métodos de acesso a vista *ViewEntitiesAssociation* e consequentes nós associados.

Se os nós a preencher estão ao nível do *dossier*, é necessário aceder a uma pasta na biblioteca *Sharepoint*. Este tipo de preenchimento, necessita de um gestor de pastas, pois o *FolderManager* é que possui os métodos de acesso a biblioteca *Sharepoint*.

Por último, são actualizadas consecutivamente, a tabela em variável de sessão e a árvore de conteúdos.

A figura 4.13 esquematiza a sequência de métodos invocados pelas diferentes classes associadas ao Viewer. Enquanto que a figura 4.14 descreve o algoritmo associado ao tratamento do evento *Click*, descrito no parágrafos anteriores.

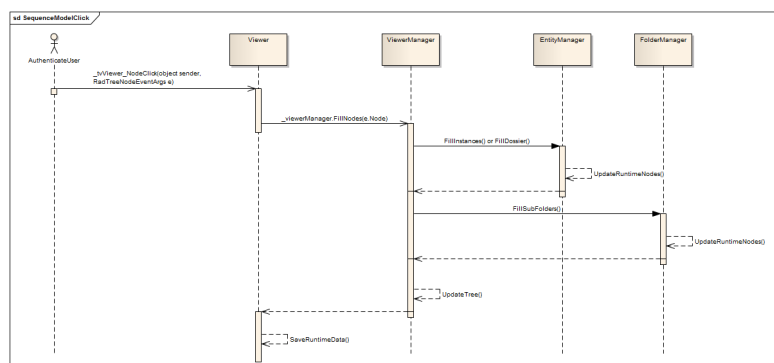


Figura 4.13: Diagrama de sequência evento *Click*

No anexo B.4 estão os restantes diagramas de análise do evento de *Click*.

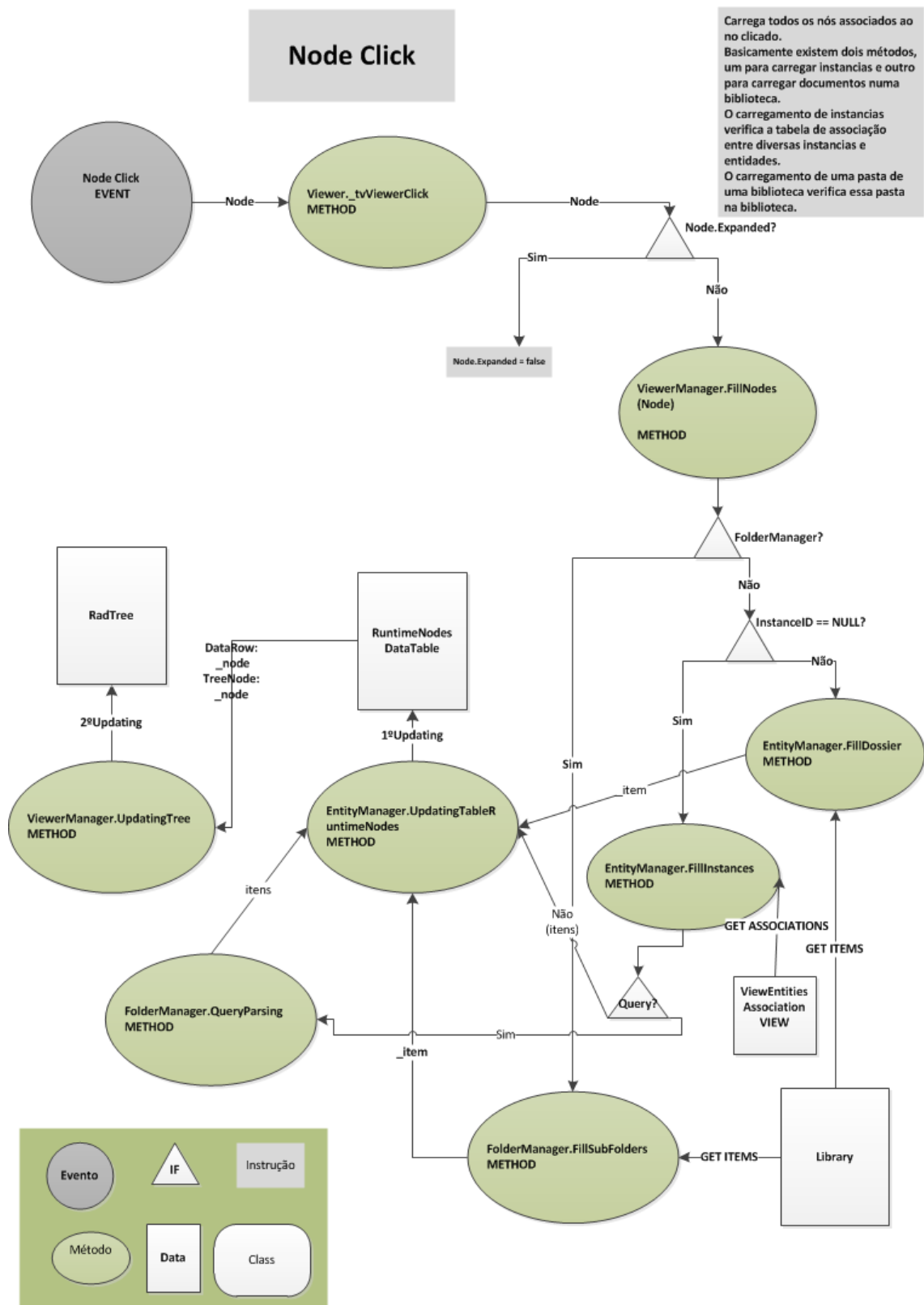


Figura 4.14: Esquema representativo do tratamento do evento *Click*

Evento *Drag and Drop*

O evento *Drag and Drop* de conteúdos consiste em disponibilizar a funcionalidade arrastar e largar sobre conteúdos na mesma página (podendo estar em Viewers diferentes).

O objecto *Viewer* recebe o evento e verifica, se existem diferentes Viewers na página, se é necessário aceder a diferentes bibliotecas e se existe o(s) ficheiro(s) movidos na pasta de destino. Em seguida, o controlo do evento é passado ao *ViewerManager* e é executado o método de movimentação de conteúdos com as restrições retiradas inicialmente.

O método de movimentação de conteúdos é o método principal na sequência de métodos a executar para o evento. Inicialmente, tenta processar a movimentação ao nível da(s) biblioteca(s), com o método *MoveDocItem* da classe *ViewerManager*. Em seguida, o método identifica as tabelas em variável de sessão e actualiza as mesmas.

Por último, é necessário actualizar as árvores de conteúdos associadas as tabelas actualizadas.

A figura 4.11 esquematiza a sequência de métodos invocados pelas diferentes classes associadas ao Viewer. Enquanto que a figura 4.12 descreve o algoritmo associado ao tratamento do evento *Drag and Drop*, descrito no parágrafos anteriores.

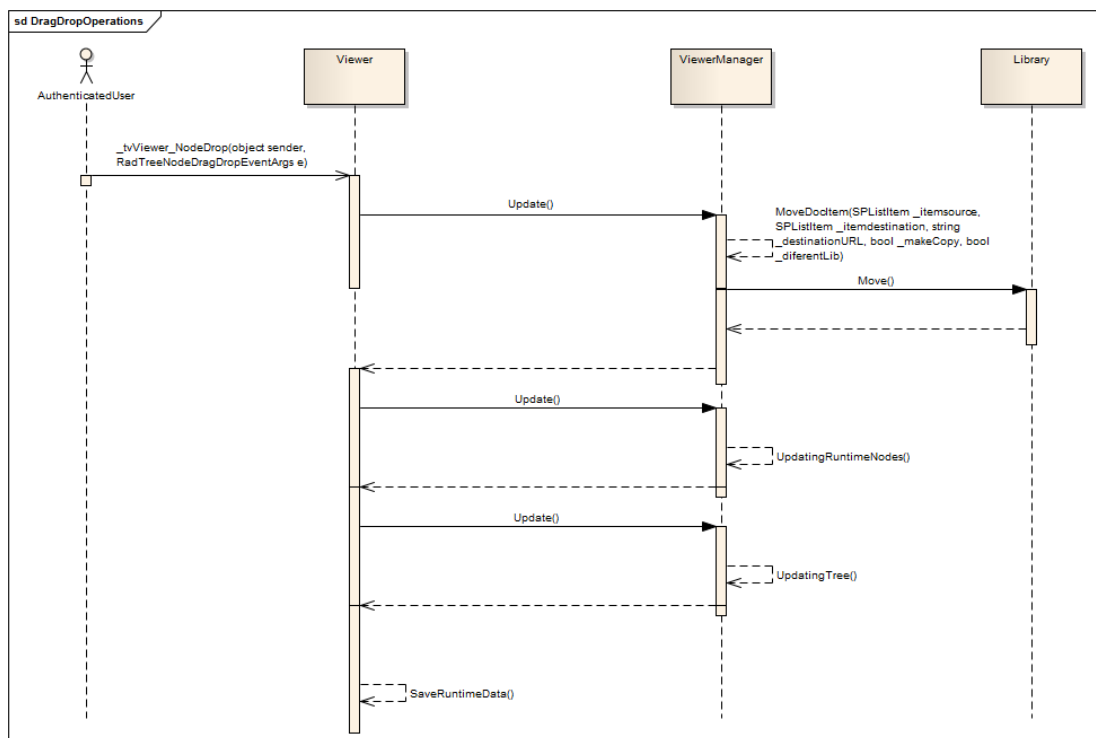


Figura 4.15: Diagrama de sequência evento *Drag and Drop*

No anexo B.5 estão os restantes diagramas de análise do evento de *Drag and Drop*.

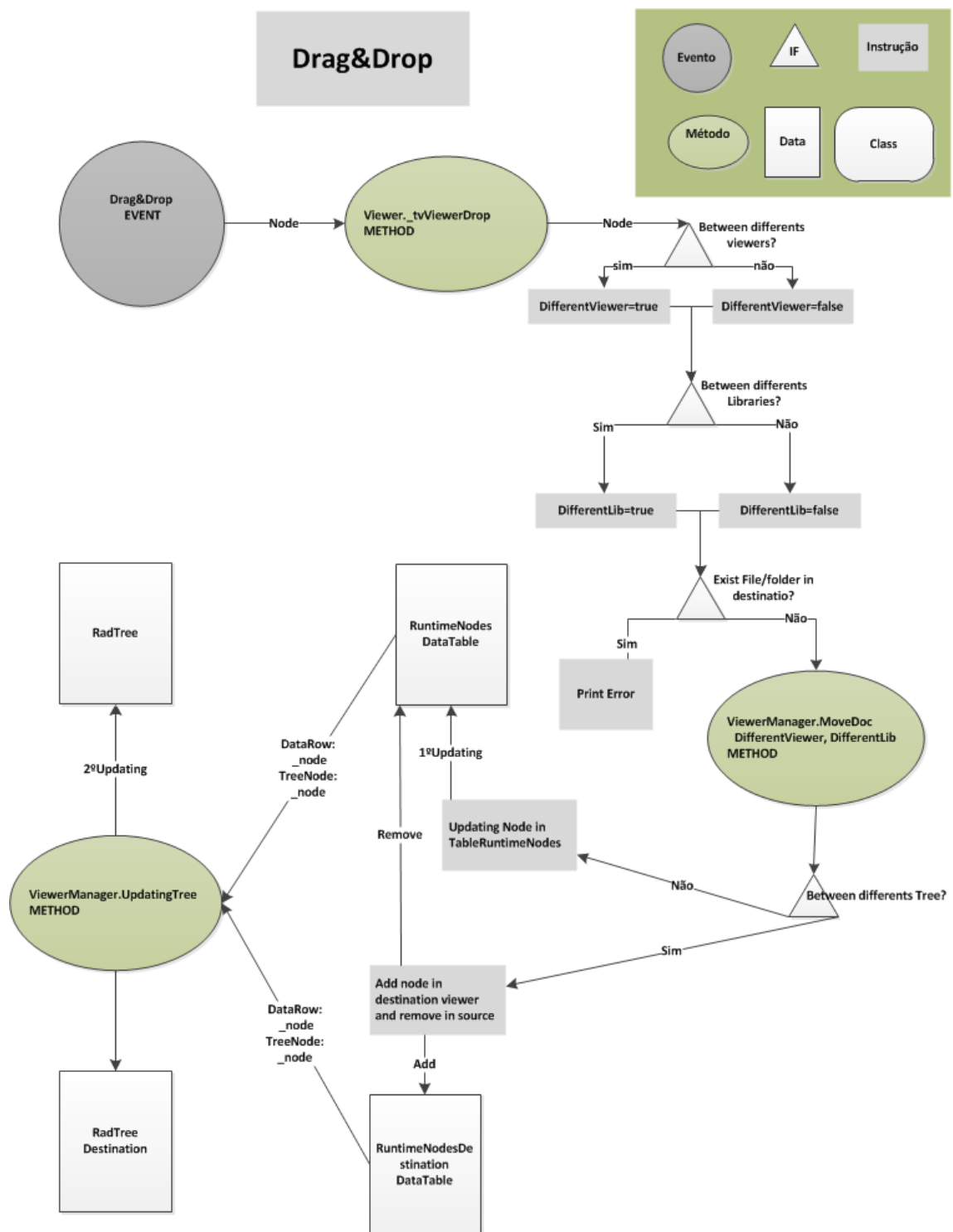


Figura 4.16: Esquema representativo do tratamento do evento *Drag and Drop*

4.5 Implementação

Este ponto detalha aspectos da implementação do módulo. São apresentados a plataforma de desenvolvimento, a configuração *in place* do módulo, a implementação dos diferentes tipos de nós, a pesquisa de conteúdos e o filtro por tipo de conteúdos.

Plataforma de desenvolvimento

Como referido anteriormente, a plataforma foi desenvolvida em *Sharepoint* e consequentemente todas as páginas da plataforma. Neste âmbito, os módulos desenvolvidos foram integrados em *Web Parts*, que seguidamente foram adicionadas as páginas.

Para implementar o módulo foi utilizada a plataforma *Sharepoint* e a biblioteca de componentes *.NET*, designada *telerik*. Foi utilizado o componente *RadTreeView* da biblioteca de componentes referida anteriormente, este componente possui uma estrutura composta por uma árvore de nós, como a *TreeView* disponível da plataforma *.NET*. Em contra-partida, a *RadTreeView* possui uma série de funcionalidades e controlos não disponibilizados pela *TreeView*, por esse motivo é a árvore utilizada na criação do Viewer.

As funcionalidades disponibilizadas pela *RadTreeView* suportam uma interactividade entre utilizador e árvore bem mais efectiva, essas funcionalidades são:

- Menus de contexto sobre nós;
- Edição *in place* do próprio;
- Atributos customizados;
- Selecção múltipla de nós;
- *Drag and Drop* entre nós;
- Animações.

As funcionalidades disponibilizados pela *RadTreeView*, têm de ser configuradas e tratadas pelo módulo desenvolvido, pois a árvore só disponibiliza as funcionalidades. [[Tel02](#)]

Configuração do módulo

Como já referido, o módulo permite uma configuração *in place* do mesmo, isto é, permite gerir os nós de topo associados ao Viewer através de uma interface de configuração. A interface de configuração está acessível através de um botão disponível na interface do módulo. A configuração dos nós de topo consiste em definir para cada nó um conjunto de parâmetros que especificam os dados a apresentar de baixo desse nó.

A interface de configuração contém uma árvore apenas com os nós de topo da árvore e disponibiliza três acções possíveis sobre cada nó (em caso de ser um nó virtual disponibiliza quatro). As acções possíveis são:

- Editar
- Adicionar nó
- Eliminar
- Adicionar nó-filho (disponível apenas se nó for virtual)

A figura 4.17 apresentada o menu de contexto disponibilizado na configuração *in place*, que são consequentemente as acções possíveis sobre os nós iniciais do Viewer.

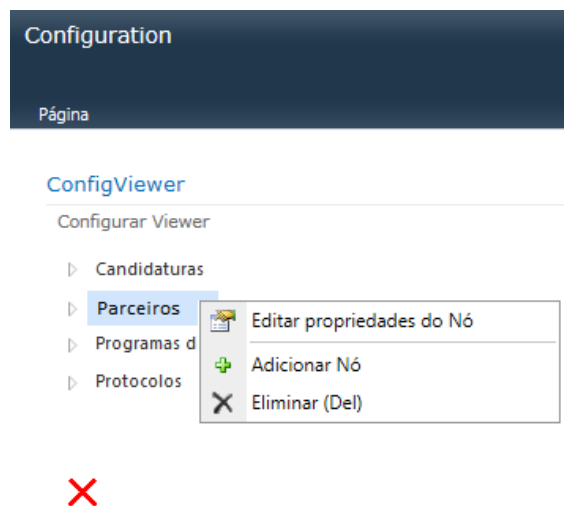


Figura 4.17: Menu de contexto disponibilizado na configuração *in place* do Viewer

Como já referido, a adição de nós disponibiliza quatro tipos de nós, em seguida são apresentadas as interfaces de configuração dos diferentes tipos de nós.

Implementação de nós

Neste ponto, descreve-se a forma de implementação dos diferentes tipos de nós disponibilizados pelo módulo Viewer.

A figura 4.18 apresentada os tipos de nós que é possível adicionar ao Viewer na configuração do mesmo. Os outros tipos de nós implementados não disponíveis na configuração do Viewer, são nós que resultam da expansão dos nós iniciais.

Figura 4.18: Tipo de nós passíveis de adicionar ao Viewer

Nó *Virtual*

O nó *Virtual* é nó que serve como forma de agrupar ou separar nós, de forma a incutir uma melhor estruturação dos nós na árvore.

A forma de implementação de um nó *Virtual* é bastante simples, pois basta criar um registo na tabela *ViewerNodes* com um nome, o tipo de nó e a identificação do nó-pai. A partir deste momento, o sistema já sabe que esse nó é virtual e que o seu gestor é o *VirtualManager*. No gestor *VirtualManager* o preenchimento das instâncias baseia-se apenas na pesquisa dos nós-filho do nó que se pretende preencher, a pesquisa incide sobre a tabela *ViewerNodes*.

A figura 4.19 apresenta a forma de adicionar um nó virtual ao Viewer.

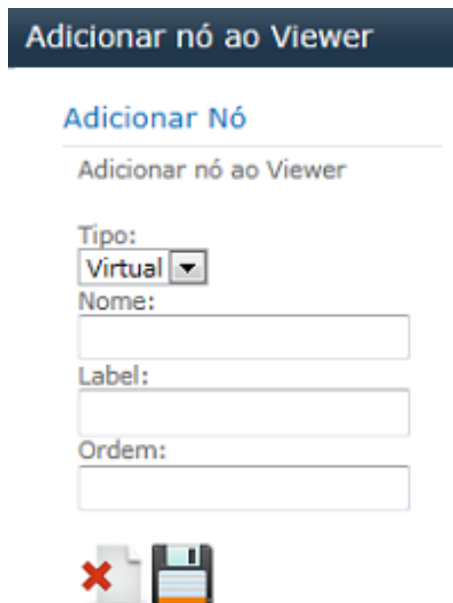


Figura 4.19: Adicionar nó virtual ao Viewer

A tabela 4.1 representa um exemplo de um nó virtual na tabela *RuntimeNodes*.

Tabela 4.1: Exemplo nó virtual na tabela em variável de sessão (*RuntimeNodes*)

Exemplo nó virtual na tabela <i>RuntimeNodes</i>	
UniqueID	12
ParentID	-1
Label	VirtualNode Label
ViewerNodeTypeID	5
EntityID	
Manager	VirtualManager
InstanceID	
TopFolderID	
ItemID	
IconUrl	
Query	

Nó *Entity*

O nó *Entity* é um nó que tem uma entidade associada como meta-data, o preenchimento de um nó entidade resulta nas instancias relacionadas, entre a página em que o módulo está a correr e a entidade associada ao nó.

A figura 4.20 apresenta a forma de adicionar um nó entidade ao Viewer.

Figura 4.20: Adicionar nó entidade ao Viewer

O preenchimento dos nós-filhos de um nó entidade são nós instância.

A tabela 4.2 representa um exemplo de um nó entidade na tabela *RuntimeNodes*.

Tabela 4.2: Exemplo nó entidade na tabela em variável de sessão (*RuntimeNodes*)

Exemplo nó entidade na tabela <i>RuntimeNodes</i>	
UniqueID	12
ParentID	
Label	EntityNode Label
ViewerNodeTypeID	1
EntityID	1
Manager	ProtocolManager
InstanceID	
TopFolderID	
ItemID	
IconUrl	
Query	

Nó *Query*

Para implementar os nós *Query* foi especificada uma estrutura prova específica de *queries* que depois são interpretadas. Cada *EntityManager* saberá interpretar a *Query* associada e executá-la, para determinar as instâncias associadas a esse nó.

A estrutura prova específica das *queries* está esquematizado na figura 4.21.

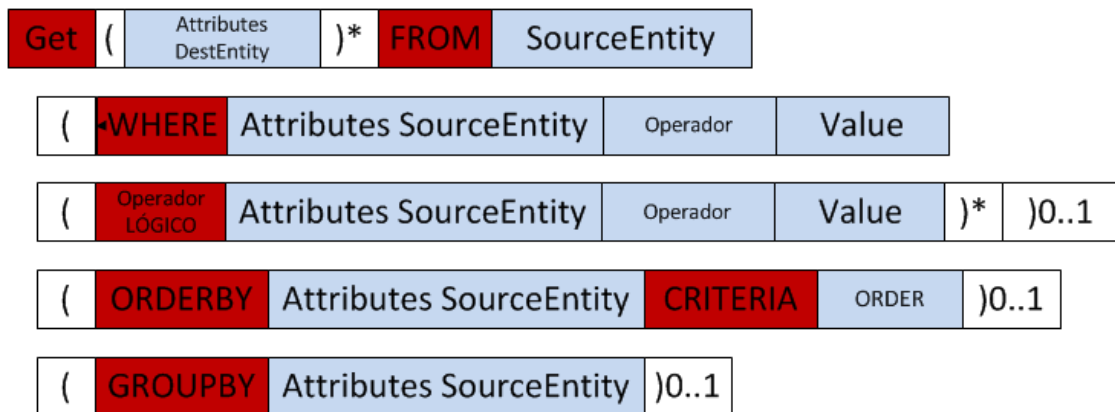


Figura 4.21: Estrutura *queries*

Analisando, as *queries* verifica-se que contêm diferentes campos, e cada campo tem características e dados específicos do mesmo. O campo *Attributes DestEntity* define os atributos das instâncias a apresentar pela *query*. O campo *SourceEntity* define a entidade a relacionar com a página. O campo *Attributes SourceEntity* pode conter qualquer um dos atributos da entidade *SourceEntity*. O campo *Operador* dependendo do tipo de atributo seleccionado anteriormente, pois se o atributo for simples o campo pode conter = / > / <>, se o atributo for múltiplo o campo pode conter *contains*. O campo *value* pode ser um valor constante ou um valor associado a própria *Web Part*, como o *thiswebpartinstance* e o *thiswebpartattribute*. Por último, o campo *order* pode tomar o valor de *ASC* ou *DESC*.

O protótipo de interface para configurar *queries* está apresentado na figura 4.22.

Adicionar nó ao Viewer

Adicionar Nó

Adicionar nó ao Viewer

Tipo:

Nome:

Label:

Ordem:

Entidade:

Condição 1:

Atributo 1:

Operador 1:

Valor 1:

Operador Lógico:

Condição 2:

Atributo 2:

Operador 2:

Valor 2:

Agrupar por:

Ordenar por:

Tipo de Ordenação:

Figura 4.22: Protótipo de interface para representar um configurador de *queries*

Para se implementar o editor de *queries* também foi necessária criar uma abstracção sobre o modelo físico da base de dados, de forma a unificar os diferentes tipos de recursos de informação (diferentes entidades têm diferentes origens). No fundo, é criar um modelo de dados sobre o próprio modelo de dados físico, é meta-data sobre os dados em base de dados.

O modelo de dados que oferece a abstracção na especificação das *queries* na interface com o utilizador é o apresentado na figura 4.23.

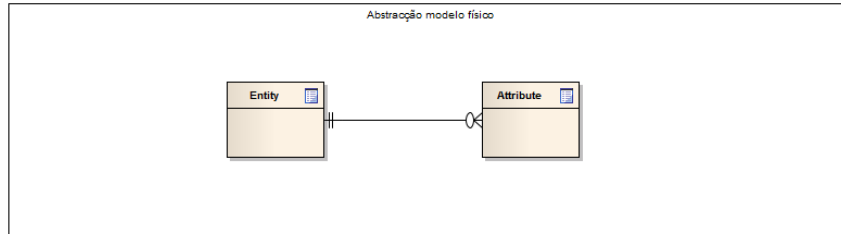


Figura 4.23: Modelo de abstracção na especificação das *queries* por parte do utilizador

Ao nível dos *EntityManagers*, cada um faz o análise da *Query*.

Depois do tratamento e análise da *Query*, o *EntityManager* transforma a *Query* em várias *queries* SQL sobre o modelo de dados físico. A execução das duas restrições possíveis podem ser simultâneas, depois é que é necessário combinar as mesmas conforme o operador. Se a *Query* possuir ordenação, em seguida é necessário aplicar essa ordenação no resultado da combinação das condições.

Em caso, da *Query* possuir agrupamento o restante conteúdo da *Query* é ignorado e são criados vários nós *Query*, representado cada grupo no agrupamento. Estes nós *Query* criados herdam a *Query* anterior sem o agrupamento e com a restrição do grupo que representam.

Este foi o método encontrado para restringir as instâncias relacionadas com o nó, a todos os níveis e não apenas a entidade como é possível com os nós *Entity*.

A tabela 4.3 representa um exemplo de um nó *Query* na tabela *RuntimeNodes*.

Tabela 4.3: Exemplo nó *Query* na tabela em variável de sessão (*RuntimeNodes*)

Exemplo nó <i>Query</i> na tabela <i>RuntimeNodes</i>	
UniqueID	12
ParentID	
Label	Query1NodeLabel
ViewerNodeTypeID	6
EntityID	1
Manager	ProtocolManager
InstanceID	
TopFolderID	
ItemID	
IconUrl	
Query	query.query

Nó *Instance*

O nó *Instance* é nó que aponta para uma instância de uma entidade, logo tanto a entidade como a instância têm que estar em meta-data.

O preenchimento de um nó instância da origem a uma pasta de topo associada a instância referenciada pelo nó.

A tabela 4.4 representa um exemplo de um nó instância na tabela *RuntimeNodes*.

Tabela 4.4: Exemplo nó instância na tabela em variável de sessão (*RuntimeNodes*)

Exemplo nó instância na tabela <i>RuntimeNodes</i>	
UniqueID	12
ParentID	
Label	InstanceNode Label
ViewerNodeTypeID	2
EntityID	1
Manager	ProtocolManager
InstanceID	12
TopFolderID	35
ItemID	
IconUrl	
Query	

Nó *File*

O nó *File* é nó que aponta para um ficheiro na biblioteca de documentos. Um nó file não é expansível.

A tabela 4.5 representa um exemplo de um nó ficheiro na tabela *RuntimeNodes*.

Tabela 4.5: Exemplo nó ficheiro na tabela em variável de sessão (*RuntimeNodes*)

Exemplo nó ficheiro na tabela <i>RuntimeNodes</i>	
UniqueID	12
ParentID	
Label	file Label
ViewerNodeTypeID	4
EntityID	1
Manager	ProtocolManager
InstanceID	12
TopFolderID	35
ItemID	33
IconUrl	word.png
Query	

Nó Folder

O nó *Folder* é um nó que aponta para uma pasta na biblioteca de documentos.

O preenchimento de um nó *Folder* resulta na instanciação das pastas e documentos filhos da pasta apontada pelo na biblioteca de documentos. Os nós resultantes do preenchimentos de um nó *Folder* são outros nós *Folder* ou nós *File*.

Os nós *Folder* normalmente aparecem a partir do terceiro nível da árvore de conteúdos do módulo, mas o módulo Viewer inclui uma funcionalidade adicional que permite adicionar nós *Folder* ao primeiro nível da árvore. Para definir um nó *Folder* no primeiro nível da árvore é necessário especificar o site, a biblioteca, a pasta de topo e a respectiva pasta a visualizar.

Os sites, bibliotecas e pastas de topo estão associados respectivamente pelas tabelas *Sites*, *Libraries* e *TopFolders*. Estas tabelas e o relacionamento entre as mesmas é ilustrado na figura 4.24.

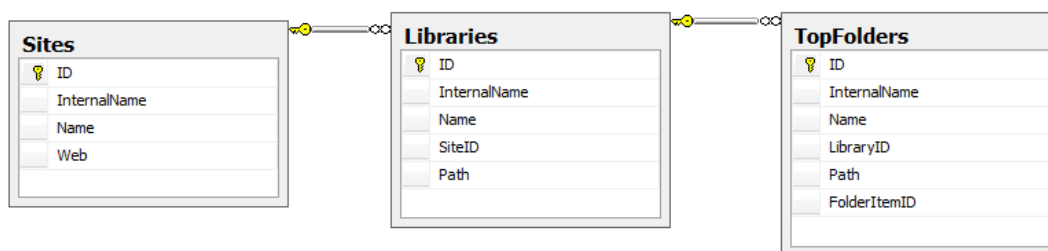


Figura 4.24: Relacionamento entre tabelas sites, bibliotecas e pastas de topo

A tabela 4.6 representa um exemplo de um nó pasta na tabela *RuntimeNodes*.

Tabela 4.6: Exemplo nó pasta na tabela em variável de sessão (*RuntimeNodes*)

Exemplo nó pasta na tabela <i>RuntimeNodes</i>	
UniqueID	12
ParentID	
Label	FolderNode Label
ViewerNodeTypeID	3
EntityID	
Manager	FoldersManager
InstanceID	
TopFolderID	13
ItemID	159
IconUrl	folder.png
Query	

A figura 4.25 apresenta a forma de adicionar um nó pasta ao primeiro nível da árvore no Viewer.

Figura 4.25: Adicionar nó pasta ao Viewer

Filtragem e pesquisa

O módulo Viewer permite efectuar pesquisas nos conteúdos dos *dossiers* apresentados. Para esse efeito, o módulo inclui um campo de texto, onde o utilizador pode inserir texto livre. O controlo retorna os conteúdos do *dossier* que contenham no nome o texto inserido.

A filtragem dos ficheiros a apresentar pelo módulo Viewer consiste na verificação da extensão do ficheiro. Se a extensão do ficheiro nos conteúdos corresponde ao seleccionado na *drop*, esse ficheiro é adicionado à árvore do módulo. Em caso de seleccionar todos os itens na *drop*, não faz essa verificação e apresenta todos os ficheiros nos conteúdos.

Módulo Viewer

A figura 4.26 apresenta as funcionalidade de pesquisa e filtragem de conteúdos no módulo. A *drop* serve de filtro sobre o tipo de conteúdos a apresentar, e o campo de texto como pesquisa sobre o nome dos conteúdos.



Figura 4.26: Funcionalidade de pesquisa e filtragem de conteúdos

Capítulo 5

Conclusões e trabalho futuro

Neste capítulo encontra-se a revisão e a avaliação do trabalho desenvolvido, e são ainda apresentadas algumas perspectivas de desenvolvimento futuro.

5.1 Revisão

A criação de módulos de gestão de informação para incluir numa plataforma de suporte ao trabalho baseado em conhecimento intensivo, consistiu na criação de dois módulos de gestão de informação, o módulo ExplorerMais e o módulo Viewer.

O módulo ExplorerMais consiste num gestor de informação organizado em pastas hierárquicas e oferece uma experiência de utilização próxima dos gestores de ficheiros convencionais como, por exemplo, o Windows Explorer¹, com algumas funcionalidades suplementares. As funcionalidades suplementares implementadas foram:

- Pasta *Managed Folder*;
- Gestão de permissões;
- Pasta versionada ou ficheiro versionado;
- Ficheiro ligação;
- Ficheiro nota;
- Ordenação manual de pastas;
- *Drag and Drop* de ficheiros e pastas;
- Filtragem e pesquisa de conteúdos.

¹http://en.wikipedia.org/wiki/Windows_Explorer

O módulo Viewer é a forma de apresentar os conteúdos relacionados de entidades diferentes. Esses conteúdos podem ter origens completamente diferentes, como servidores de *e-mails*, listas de documentos *Sharepoint* ou registos em base de dados, mas mesmo assim a sua gestão no módulo é exactamente igual. Esta característica é conseguida devido a uma subida no nível de abstracção da base de dados, com tabelas de meta-dados sobre as próprias tabelas existentes em base de dados.

O Viewer ainda agrega outras funcionalidades, como:

- Configuração *in place*
- *Drag and Drop* entre conteúdos
- Criação de diferentes tipos de nós
- Pesquisa sobre os conteúdos
- Filtragem por tipo de conteúdos.

Os dois módulos implementados poderão ser integrados numa plataforma de gestão do conhecimento intensivo. Assim, a integração dos módulos desenvolvidos numa plataforma depende das necessidades do cliente.

5.2 Avaliação final do trabalho

Os indicadores de avaliação deste trabalho são a concretização dos objectivos inicialmente definidos. Neste contexto, foi construída uma ferramenta para gerir de forma unificada diferentes recursos de informação. A ferramenta teria que ser fácil de utilizar, robusta e flexível.

Todos os interfaces com o utilizador foram desenvolvidos com o objectivo de serem bastante intuitivos, para tornar a experiência com a ferramenta agradável e simples para o utilizador.

A forma de gerir conteúdos é similar com o explorador de *Windows*, dessa forma também se obteve uma aceitação e uma facilidade de interacção, por parte dos utilizadores, mais efectivas.

A robustez e flexibilidade assentam na configuração independente dos módulos, isto é, para cada plataforma, os módulos e suas funcionalidades são facilmente configuráveis. Cada funcionalidade é um módulo independente que se associa aos *templates* das páginas.

A gestão unificada dos diferentes tipos de recursos de informação foi conseguida através de meta-data sobre a base de dados. A abstracção da base de dados associa um gestor de entidade a cada instância. Assim, apenas cada gestor de entidade é que consegue aceder aos diferentes tipos de recursos associados à instância.

Por isso se pode concluir, que os objectivos propostos foram atingidos.

5.3 Trabalho Futuro

Estes módulos, poderiam evoluir com a inclusão de novas funcionalidades como, por exemplo receber os e-mails directamente na aplicação de gestão de projectos em vez de receber na aplicação cliente de e-mail.

Também se poderia incluir, uma forma de sincronização com o *Microsoft Outlook*² tanto no lado do *Microsoft Outlook* como no lado do *Microsoft Sharepoint*³. A sincronização poderia ser ao nível das tarefas, *e-mails* ou contactos. Assim, a sincronização de tarefas, *e-mails* ou contactos consistia numa exportação dos dados do *Sharepoint* para o *Outlook* ou vice-versa.

A arquitectura de uma aplicação de sincronização de dados *Microsoft Outlook* com *Microsoft Sharepoint* necessita de um *WebService* para gerir a comunicação entre as duas aplicações.

A aplicação destes módulos em diferentes plataformas pode implicar o desenvolvimento de novas funcionalidades, que serão de fácil integração nos mesmos.

²http://en.wikipedia.org/wiki/Microsoft_Outlook

³http://en.wikipedia.org/wiki/Microsoft_Sharepoint

Conclusões e trabalho futuro

Referências

- [Aco10] Aconex. Aconex, 2010. acessado em Dezembro 2010 e disponível em <http://www.aconex.com/>.
- [AII10] AIIM. AIIM, association for information and image management, 2010. acessado em Janeiro 2011 e disponível em <http://www.aiim.org/>.
- [Alf10] Alfresco. Alfresco, 2010. acessado em Dezembro 2010 e disponível em <http://www.alfresco.com/>.
- [BFP⁺08] Tom Butler, Joseph Feller, Andrew Pope, Bill Emerson e Ciaran Murphy. Designing a core it artifact for knowledge management systems using participatory action research in a government and non-government organization. *Journal of Strategic Information Systems*, 2008.
- [Bra99] Ascensão Braga. A gestão da informação, 1999. acessado em Janeiro 2011 e disponível em http://www.ipv.pt/millennium/19_arq1.htm.
- [Cra09a] Kevin Craine. Computhink why document management. 2009.
- [Cra09b] Kevin Craine. The strategic importance of email archive management. 2009.
- [Dro10] Dropbox. Dropbox, 2010. acessado em Dezembro 2010 e disponível em <http://www.dropbox.com/>.
- [Goo10] Google. Google docs, 2010. acessado em Dezembro 2010 e disponível em <http://docs.google.com>.
- [Has07] Lewis Hassell. A continental philosophy perspective on km. 2007.
- [HYJK06] C. Hill, R. Yates, C. Jones e S. L. Kogan. Beyond predictable workflows: Enhancing productivity in artful business processes. *IBM SYSTEMS JOURNAL*, pages 1–12, 2006.
- [Ind09] Indigo. Dekks, who knows what, 2009. acessado em Janeiro 2011 e disponível em <http://www.dekks.com/>.
- [iPo10] iPortalMais. iportaldoc, 2010. acessado em Dezembro 2010 e disponível em <http://www.iportaldoc.com/>.
- [Kno10] KnowledgeTree. Knowledgetree, 2010. acessado em Dezembro 2010 e disponível em <http://www.knowledgetree.com/>.

REFERÊNCIAS

- [KRO06] Hilary Kane, Gillian Ragsdell e Charles Oppenheim. Knowledge management methodologies. *The Electronic Journal of Knowledge Management*, pages 141–152, 2006.
- [Loc10] Document Locator. Document locator, 2010. acedido em Dezembro 2010 e disponível em <http://www.documentlocator.com/>.
- [McC08] Dermot McCauley. Singularity, the singularity process platform. 2008.
- [Mic10] Microsoft. Microsoft sharepoint 2010, 2010. acedido em Janeiro 2011 e disponível em <http://sharepoint.microsoft.com/>.
- [MMC10] MMC. edoclink, 2010. acedido em Dezembro 2010 e disponível em <http://www.edoclink.mccomputadores.com.com/>.
- [NC99] Brian (Bo) Newman e Kurt W. Conrad. A framework for characterizing knowledge management methods, practices, and technologies. *George Washington University*, pages 1–12, 1999.
- [Swe10] Keith Swenson. *Mastering the Unpredictable, How Adaptive Case Management Will Revolutionize the Way That Knowledge Workers Get Things Done*. Meghan-Kiffer Press, 2010.
- [Tel02] Telerik. Telerik,radtreeview for asp.net ajax, 2002. acedido em Junho 2011 e disponível em <http://www.telerik.com/products/aspnet-ajax/treeview.aspx>.
- [Web10] Webuild. File doc 2.0, 2010. acedido em Janeiro 2011 e disponível em <http://www.webuild.pt/>.

Anexo A

Controlo ExplorerMais

A.1 Página Projectos

A.1.1 Diagrama casos de uso Página Projectos

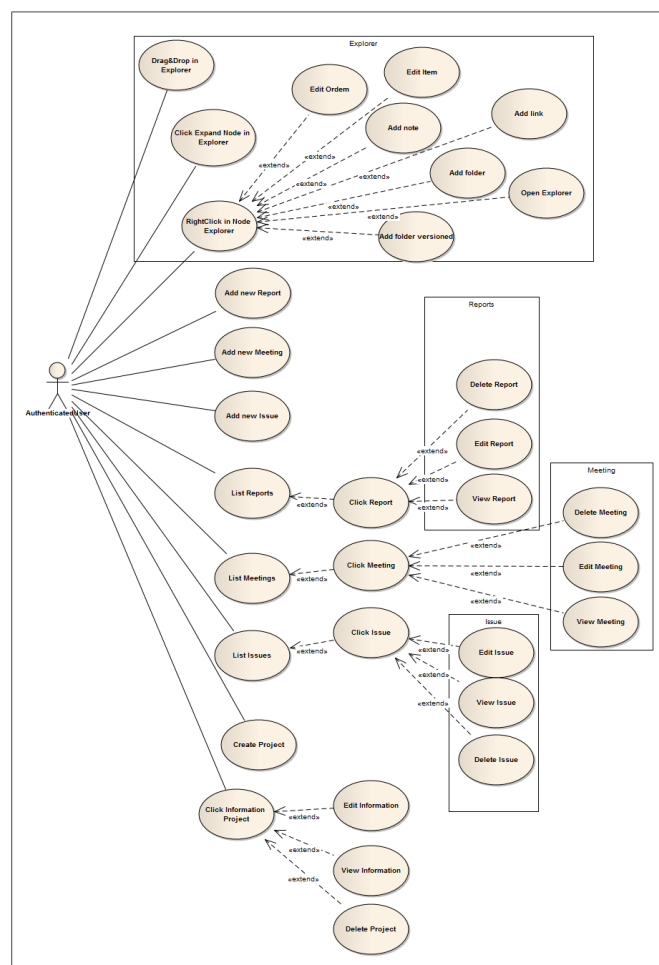


Figura A.1: Diagrama casos de uso página projecto

A.1.2 Estrutura pastas projectos

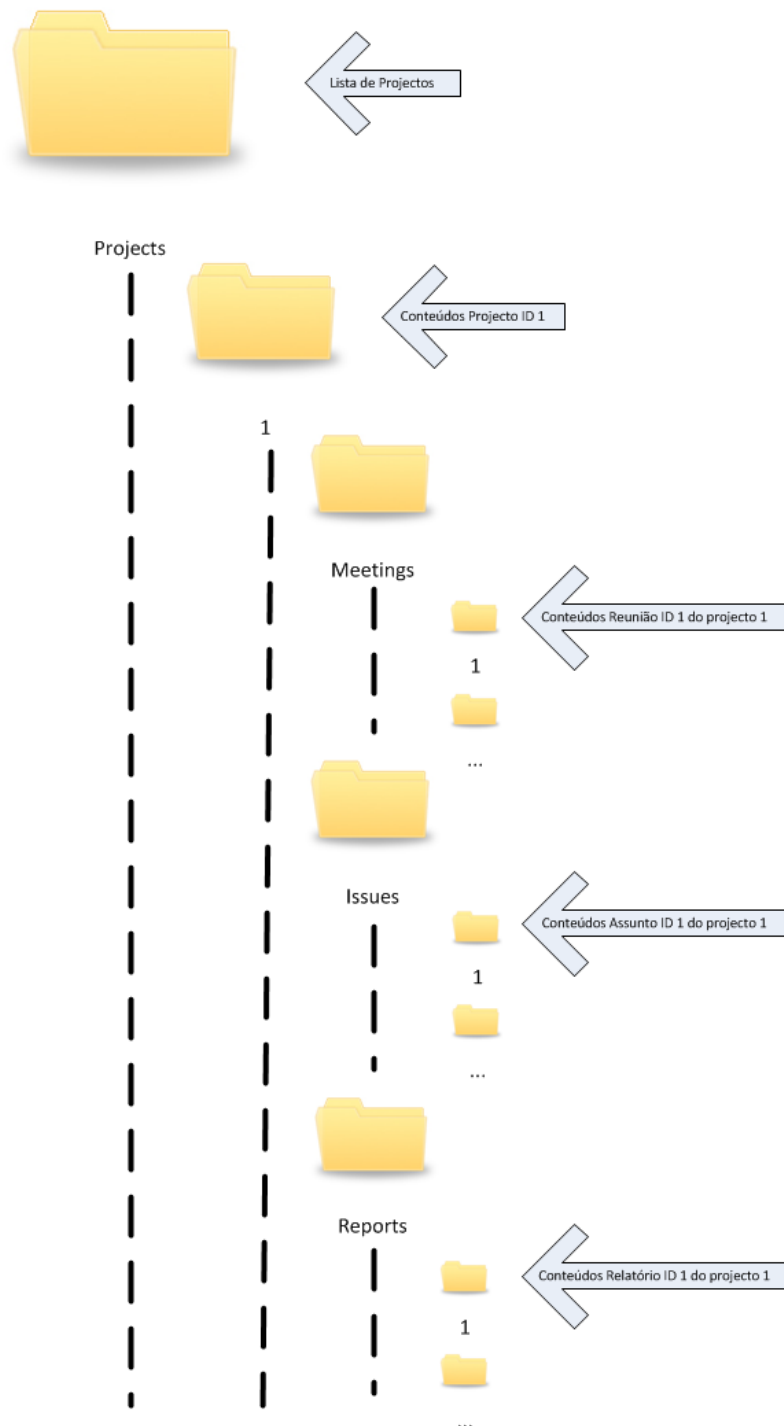


Figura A.2: Estrutura de pastas para projectos

A.1.3 Estrutura pastas projectos e atributos *Managed Folder*

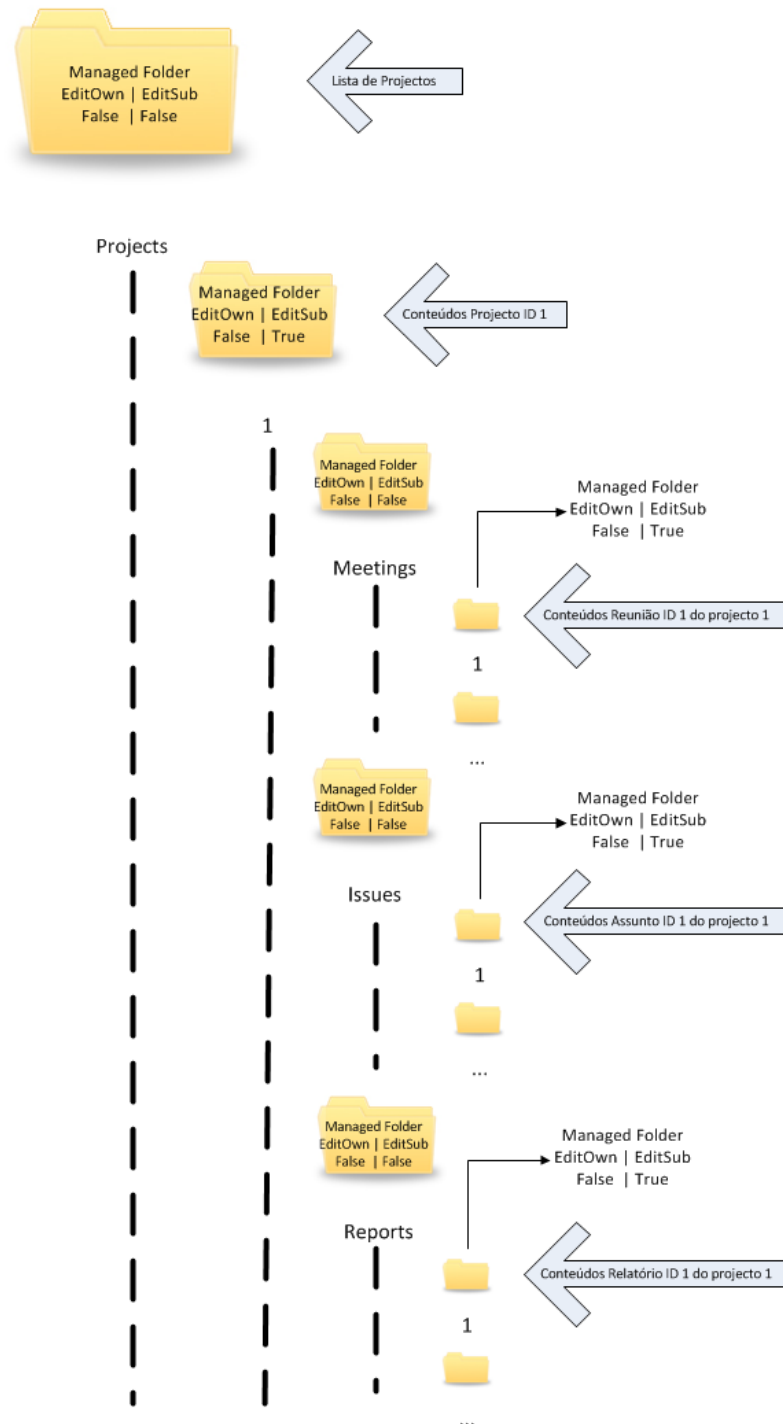


Figura A.3: Estrutura de pastas para projectos e atributos *Managed Folder*

A.1.4 Tabelas descrição dos casos de uso

Tabela A.1: Caso de uso *Drag and Drop* de ficheiros e pastas

Caso de uso <i>Drag and Drop</i> de ficheiros e pastas	
Nome	<i>Drag and Drop</i> de ficheiros e pastas
Descrição	O controlo ExplorerMais permitirá efectuar o <i>Drag and Drop</i> de ficheiros e pastas, dentro dos conteúdos do controlo
Actores	Responsáveis e colaboradores devidamente autenticado
Pré-condições	O utilizador terá de se autenticar
Sequência	O utilizador deverá seleccionar os itens a mover e arrastar os mesmos até uma pasta ou ficheiro com permissão para receber itens
Pós-condições	

Tabela A.2: Caso de uso adicionar nota

Caso de uso adicionar nota	
Nome	Caso de uso adicionar nota
Descrição	O controlo ExplorerMais permitirá a adição de notas, com extensão <i>.note</i> , aos conteúdos do controlo. Uma nota é um ficheiro XML ¹ nos conteúdos, com dois campos, título e texto. Dependendo das permissões das pastas, os ficheiros e pastas dentro da mesma podem ou não disponibilizar o menu de contexto adicionar nota
Actores	Responsáveis e colaboradores devidamente autenticado
Pré-condições	O utilizador terá de se autenticar
Sequência	Abrir menu de contexto sobre ficheiro ou pasta, e seleccionar adicionar nota. Em seguida, preencher os campos título e texto da nota, e guardar
Pós-condições	

Tabela A.3: Caso de uso editar nota

Caso de uso editar nota	
Nome	Caso de uso editar nota
Descrição	O controlo ExplorerMais permitirá editar notas, com extensão <i>.note</i> , nos conteúdos do controlo. Uma nota é um ficheiro XML ² nos conteúdos, com dois campos, título e texto, estes dois campos são os campos editáveis
Actores	Responsáveis e colaboradores devidamente autenticado.
Pré-condições	O utilizador terá de se autenticar
Sequência	Abrir a nota, editar campos e guardar
Pós-condições	

Tabela A.4: Caso de uso eliminar nota

Caso de uso eliminar nota	
Nome	Caso de uso eliminar nota
Descrição	O controlo ExplorerMais permitirá eliminar notas, com extensão <i>.note</i> , nos conteúdos do controlo. Uma nota é um ficheiro XML ³ nos conteúdos, com dois campos, título e texto
Actores	Responsáveis e colaboradores devidamente autenticado
Pré-condições	O utilizador terá de se autenticar
Sequência	Seleccionar no menu de contexto sobre a nota o item eliminar e depois confirmar o mesmo
Pós-condições	

Tabela A.5: Caso de uso adicionar ligação

Caso de uso adicionar ligação	
Nome	Caso de uso adicionar ligação
Descrição	O controlo ExplorerMais permitirá a adição de ligações, aos conteúdos do controlo. Uma ligação é um ficheiro XML ⁴ nos conteúdos, com dois campos, título e texto. Dependendo das permissões das pastas, os ficheiros e pastas dentro da mesma podem ou não disponibilizar o menu de contexto adicionar ligação
Actores	Responsáveis e colaboradores devidamente autenticado
Pré-condições	O utilizador terá de se autenticar
Sequência	Abrir menu de contexto sobre ficheiro ou pasta, e seleccionar adicionar ligação. Em seguida, preencher os campos título e <i>url</i> da ligação, e guardar
Pós-condições	

Tabela A.6: Caso de uso editar ligação

Caso de uso editar ligação	
Nome	Caso de uso editar ligação
Descrição	O controlo ExplorerMais permitirá editar ligações, nos conteúdos do controlo. Uma ligação é um ficheiro XML ⁵ nos conteúdos, com dois campos, título e texto, estes dois campos são os campos editáveis
Actores	Responsáveis e colaboradores devidamente autenticado.
Pré-condições	O utilizador terá de se autenticar
Sequência	Abrir a ligação, editar campos e guardar
Pós-condições	

Tabela A.7: Caso de uso eliminar ligação

Caso de uso eliminar ligação	
Nome	Caso de uso eliminar ligação
Descrição	O controlo ExplorerMais permitirá eliminar ligações, nos conteúdos do controlo. Uma ligação é um ficheiro XML ⁶ nos conteúdos, com dois campos, título e texto
Actores	Responsáveis e colaboradores devidamente autenticado
Pré-condições	O utilizador terá de se autenticar
Sequência	Seleccionar no menu de contexto sobre a ligação o item eliminar e depois confirmar o mesmo
Pós-condições	

Tabela A.8: Caso de uso adicionar pasta

Caso de uso adicionar pasta	
Nome	Caso de uso adicionar pasta
Descrição	O controlo ExplorerMais permitirá a adição de pastas, aos conteúdos do controlo. Dependendo das permissões das pastas, os ficheiros e pastas dentro da mesma podem ou não disponibilizar o menu de contexto adicionar pasta
Actores	Responsáveis e colaboradores devidamente autenticado
Pré-condições	O utilizador terá de se autenticar
Sequência	Abrir menu de contexto sobre ficheiro ou pasta, e seleccionar adicionar pasta
Pós-condições	

Tabela A.9: Caso de uso eliminar pasta

Caso de uso eliminar pasta	
Nome	Caso de uso eliminar pasta
Descrição	O controlo ExplorerMais permitirá eliminar pastas, nos conteúdos do controlo. Dependendo das permissões das pastas, os ficheiros e pastas dentro da mesma podem ou não disponibilizar o menu de contexto eliminar
Actores	Responsáveis e colaboradores devidamente autenticado
Pré-condições	O utilizador terá de se autenticar
Sequência	Abrir menu de contexto pasta, e seleccionar eliminar
Pós-condições	Confirmar a eliminação da pasta

Tabela A.10: Caso de uso adicionar pasta versionada

Caso de uso adicionar pasta versionada	
Nome	Caso de uso adicionar pasta versionada
Descrição	O controlo ExplorerMais permitirá a adição de pasta versionadas, aos conteúdos do controlo. Dependendo das permissões das pasta versionadas, os ficheiros e pasta versionadas dentro da mesma podem ou não disponibilizar o menu de contexto adicionar pasta versionada
Actores	Responsáveis e colaboradores devidamente autenticado
Pré-condições	O utilizador terá de se autenticar
Sequência	Abrir menu de contexto sobre ficheiro ou pasta versionada, e seleccionar adicionar pasta versionada
Pós-condições	

Tabela A.11: Caso de uso eliminar pasta versionada

Caso de uso eliminar pasta versionada	
Nome	Caso de uso eliminar pasta versionada
Descrição	O controlo ExplorerMais permitirá eliminar pasta versionadas, nos conteúdos do controlo. Dependendo das permissões das pasta versionadas, os ficheiros e pasta versionadas dentro da mesma podem ou não disponibilizar o menu de contexto eliminar
Actores	Responsáveis e colaboradores devidamente autenticado
Pré-condições	O utilizador terá de se autenticar
Sequência	Abrir menu de contexto pasta versionada, e seleccionar eliminar
Pós-condições	Confirmar a eliminação da pasta versionada

Anexo B

Controlo Viewer

B.1 Base de dados completa Viewer

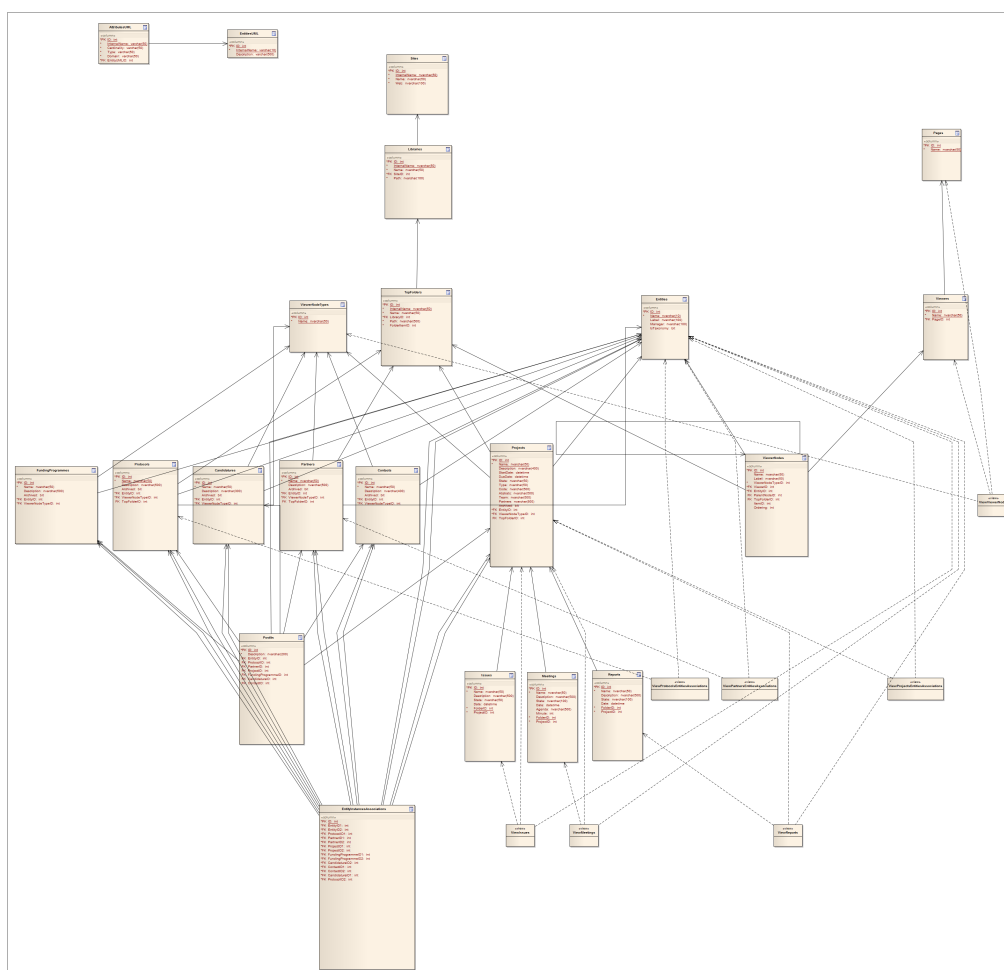


Figura B.1: Base de dados completa do controlo Viewer.

B.2 Configuração *WebPart* com controlos Viewer



Figura B.2: Controlo Viewer



Figura B.3: Controlo Viewer - botão modo edição

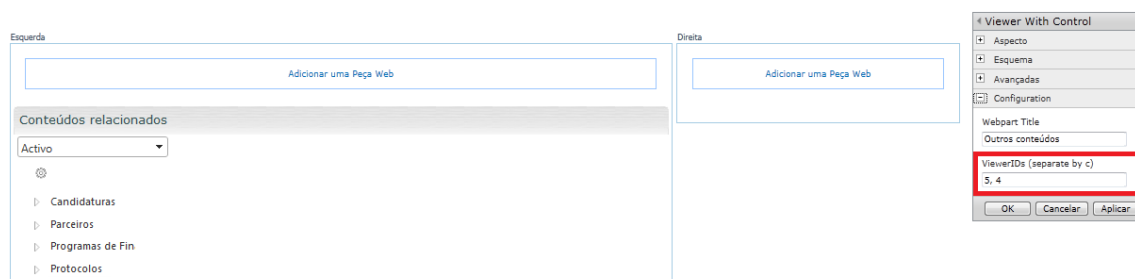


Figura B.4: Controlo Viewer em modo edição

B.3 Evento *Load*

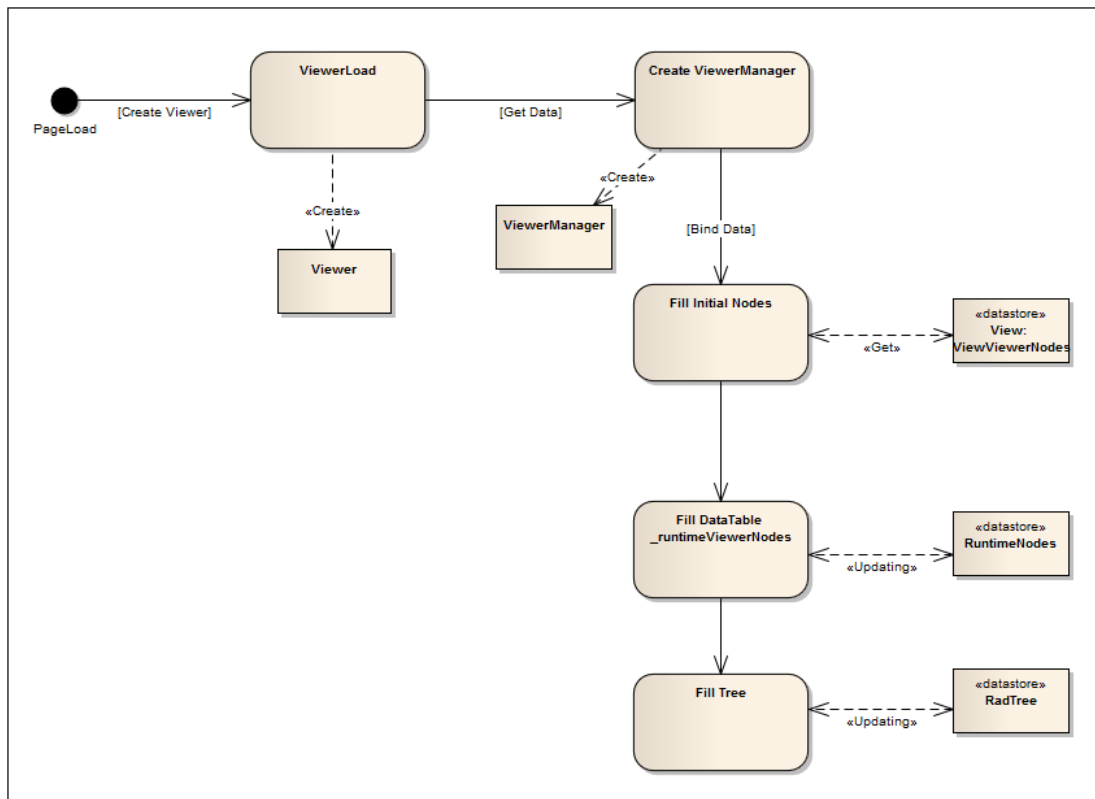


Figura B.5: Diagrama de actividade evento *Load*

B.4 Evento *Click*

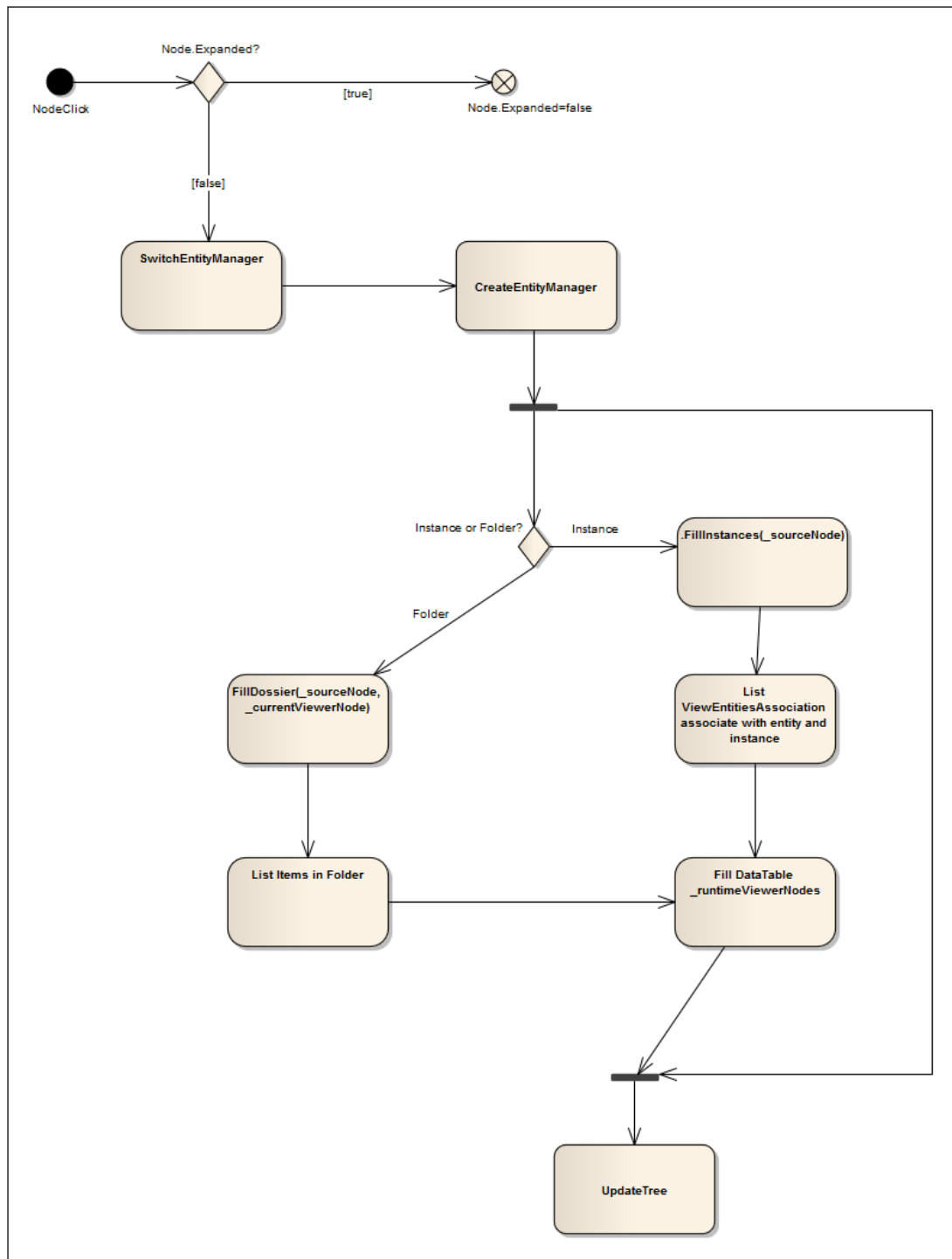


Figura B.6: Diagrama de actividade evento *Click*

B.5 Evento *Drag and Drop*

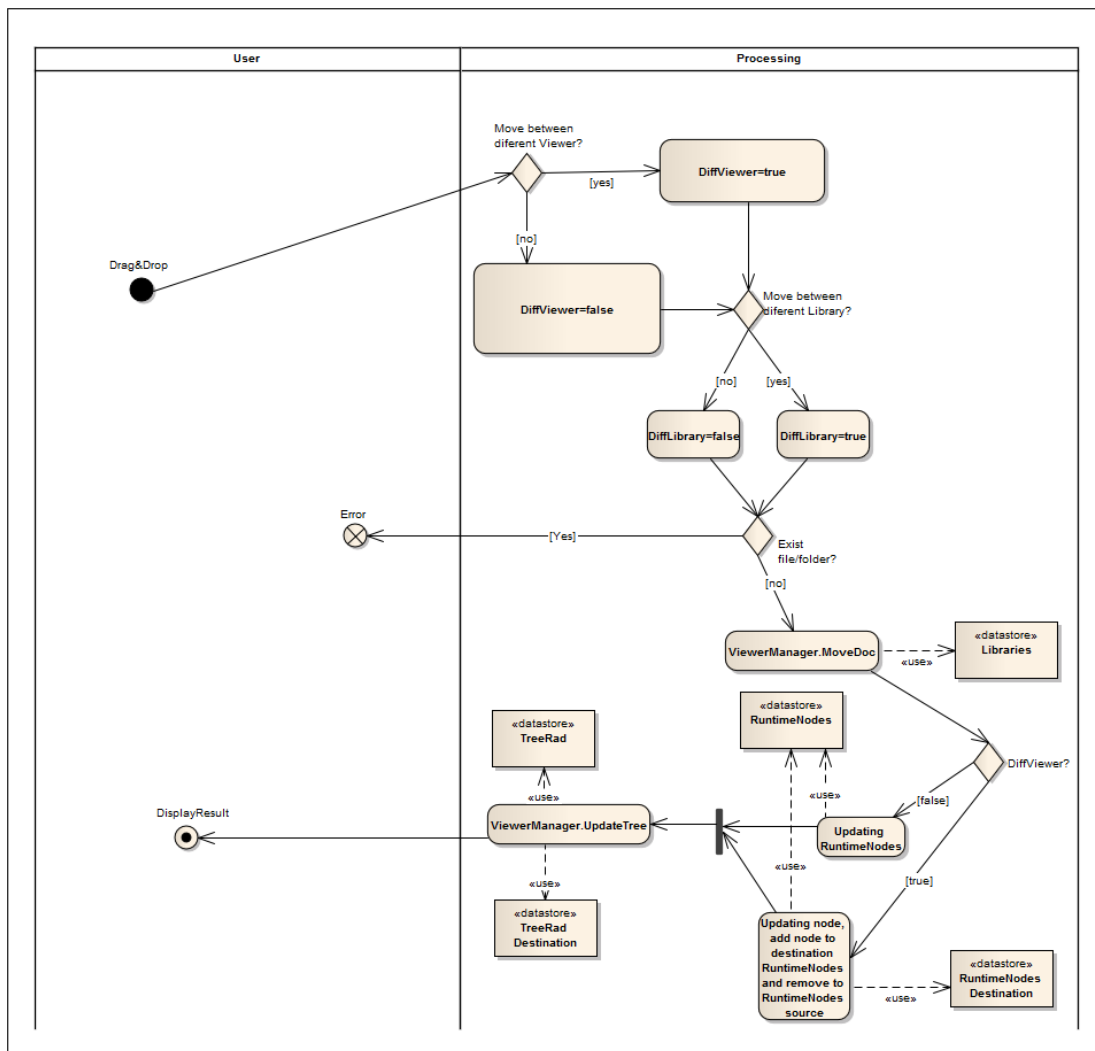


Figura B.7: Diagrama de actividade evento *Drag and Drop*

B.5.0.1 Descrição de casos de uso

Tabela B.1: Caso de uso abrir ficheiro

Caso de uso abrir ficheiro	
Nome	Abrir ficheiro
Descrição	O <i>Click</i> sobre um nó tipo ficheiro dá origem a abertura do ficheiro
Actores	Responsáveis e colaboradores devidamente autenticado
Pré-condições	O utilizador terá de se autenticar
Sequência	<i>Click</i> sobre um nó do tipo ficheiro
Pós-condições	

Tabela B.2: Caso de uso expandir nó

Caso de uso navegar para nós relacionados	
Nome	Navegar para nós relacionados
Descrição	Ao clicar sobre o nó, são apresentados os nós relacionados, e no caso de o nó já ter sido clicado os nós relacionados são eliminados.
Actores	Responsáveis e colaboradores devidamente autenticado
Pré-condições	O utilizador terá de se autenticar
Sequência	<i>Click</i> sobre um nó
Pós-condições	

Tabela B.3: Caso de uso abrir explorador

Caso de uso abrir explorador	
Nome	Abrir Explorador
Descrição	Abrir Explorador na pasta que está associada ao nó
Actores	Responsáveis e colaboradores devidamente autenticado
Pré-condições	O utilizador terá de se autenticar
Sequência	<i>Double-Click</i> sobre um nó do tipo pasta
Pós-condições	

Tabela B.4: Caso de uso navegar para instância

Caso de uso navegar para instância	
Nome	Navegar para instância
Descrição	O <i>Double-Click</i> sobre um nó do tipo instância faz com que o sistema seja redireccionado para a página dessa instância
Actores	Responsáveis e colaboradores devidamente autenticado
Pré-condições	O utilizador terá de se autenticar
Sequência	<i>Double-Click</i> sobre um nó do tipo instância
Pós-condições	

Tabela B.5: Caso de uso adicionar nó

Caso de uso adicionar nó	
Nome	Adicionar nó
Descrição	Adicionar nó de configuração ao Viewer
Actores	Responsáveis e colaboradores devidamente autenticado
Pré-condições	O utilizador terá de se autenticar
Sequência	<i>Click</i> no botão de configuração, <i>Left-Click</i> sobre nó para abrir menu de contexto e seleccionar adicionar nó
Pós-condições	

Tabela B.6: Caso de uso eliminar nó

Caso de uso eliminar nó	
Nome	Eliminar nó
Descrição	Eliminar nó de configuração do Viewer
Actores	Responsáveis e colaboradores devidamente autenticado
Pré-condições	O utilizador terá de se autenticar
Sequência	<i>Click</i> no botão de configuração, <i>Left-Click</i> sobre nó para abrir menu de contexto e seleccionar eliminar
Pós-condições	

Controlo Viewer

Tabela B.7: Caso de uso editar nó

Caso de uso editar nó	
Nome	Editar nó
Descrição	Editar nó de configuração do Viewer
Actores	Responsáveis e colaboradores devidamente autenticado
Pré-condições	O utilizador terá de se autenticar
Sequência	<i>Click</i> no botão de configuração, <i>Left-Click</i> sobre nó para abrir menu de contexto e seleccionar editar
Pós-condições	

Tabela B.8: Caso de uso *Drag and Drop* entre ficheiros e pastas

Caso de uso <i>Drag and Drop</i> entre ficheiros e pastas	
Nome	<i>Drag and Drop</i> entre ficheiros e pastas
Descrição	O sistema permite o <i>Drag and Drop</i> entre ficheiros e pastas existentes, nos conteúdos das diferentes instâncias. Esses conteúdos podem estar em diferentes bibliotecas ou mesmo diferentes <i>sites</i>
Actores	Responsáveis e colaboradores devidamente autenticado
Pré-condições	O utilizador terá de se autenticar
Sequência	Arrastar ficheiros ou pastas para outra pasta ou ficheiro, pois no caso de ser ficheiro o sistema identifica a pasta-pai
Pós-condições	