

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

*in*Vídeo
uma aplicação de vídeo interactivo
sobre a Internet

Eduardo Miguel Dias Marques

Licenciado em Engenharia de Sistemas e Computadores
pela Universidade da Madeira

Dissertação submetida para satisfação parcial
dos requisitos do grau de
mestre em Tecnologias Multimédia

Dissertação realizada sob a supervisão de
Professor Eurico Manuel Carrapatoso,
do Departamento de Engenharia Electrotécnica e de Computadores
da Faculdade de Engenharia da Universidade do Porto

Porto, Dezembro de 2004

Resumo

A utilização de vídeo e áudio sobre a Internet tem ganho nos últimos anos um interesse crescente. Na verdade, o interesse não é recente, mas os limites tecnológicos de variadíssimas ordens impediam o desenvolvimento e a utilização alargada de aplicações com *media* ricos. Os avanços nas tecnologias relacionadas com, por exemplo, a compressão, as redes e as ferramentas multimédia possibilitaram o desenvolvimento e o funcionamento, de uma forma alargada, de sistemas multimédia.

O trabalho apresentado nesta dissertação descreve a especificação e a implementação de um sistema de formação à distância, designado *inVÍdeo*, que utiliza o vídeo como elemento central. O sistema tem como objectivo proporcionar formação por intermédio de uma aplicação que permite a visualização paralela de um vídeo com imagens ou animações, acompanhada com outros dados, como texto e acesso a ficheiros que podem ser descarregados. O enquadramento para esta aplicação surgiu das necessidades de formação contínua de um centro hospitalar, todavia a aplicação foi desenvolvida para poder ser utilizada noutras instituições.

A especificação passou pela descrição do problema e sua modelação utilizando um método para o desenvolvimento de aplicações interactivas, o WISDOM. A implementação descreve o sistema desenvolvido, a justificação das opções tecnológicas e os resultados de um inquérito feito aos utilizadores do sistema.

O desenvolvimento do sistema levou ao estudo e, em alguns casos, à utilização de um conjunto de tecnologias distintas. Os estudos iniciais abordaram as tecnologias que possibilitam a utilização do vídeo sobre a Internet e algumas das suas condicionantes. Seguiram-se a composição, sincronização e interactividade multimédia, e por fim o uso de metadados para conteúdos educacionais.

A resposta dada pelos utilizadores durante a utilização do sistema foi bastante satisfatória e revelou um grande interesse no uso do vídeo sobre a Internet. A construção de aplicações para Internet utilizando *media* exigentes deixou de ser a tarefa complexa que era há alguns anos, isto devido aos avanços tecnológicos. Todavia o desenvolvimento ainda necessita de muita ponderação e do enquadramento de muitos factores relacionados com a codificação dos *media* e com a heterogeneidade das redes, dos utilizadores e dos equipamentos.

Abstract

Using audio and video over the Internet has become increasingly popular in the last years. Actually, the interest has not been recent, but several technological limitations avoided a broader usage of rich media applications. The advances in the technologies related to, for example, compression, networking and multimedia tools, allowed the development and the use of multimedia systems by a larger number of persons.

The work presented in this thesis describes the specification and the implementation of a distance training system, *inVÍdeo*, which uses the video as central element. The main purpose of the system is to provide training with an application that presents a video in parallel with an image or animation. The scene is complemented with other elements, as texts, and with the access to additional files for downloading. This system is a result of a need of training detected in a medical centre, although the usage of this system in other institutions is possible.

The specification was made with the description of the system and its modulation, using a method to develop interactive applications. The implementation describes the system developed, justifies the several technological choices and presents the result of an inquiry given to system users.

The development of the system led us to the study and, in some cases, to different technology choices. The initial studies were about the technologies that allow the usage of video over the Internet and its limitations. The next study was the multimedia composition, synchronization and interactivity, and at last the use of metadata for educational contents.

The response given by system users was very positive and demonstrated the interest in video over the Internet. The development of applications for the Internet using demanding media no longer is the complex task that was a few years ago, thanks to the technological advances. Though, the development of this kind of applications still needs lots of perception in numerous factors related to media coding and network, users and terminal heterogeneity.

Résumé

L'utilisation de la vidéo et de l'audio sur l'Internet gagne de plus en plus d'intérêt, au long des dernières années. En fait, l'intérêt n'est pas récent, mais jusqu'à présent les limites technologiques les plus variées ont empêché le développement et l'utilisation élargie des applications avec les media riches. Les innovations technologiques, dans les domaines de la compression, des réseaux et des outils multimédias, ont rendu possible le développement et le fonctionnement des systèmes multimédias en générale.

Le travail présenté dans cette dissertation décrit la spécification et l'implémentation d'un système, nommé *inVideo*, qui utilise la vidéo comme l'élément central. Le système a comme objectif proportionner une formation avec l'aide d'une application qui permet la visualisation parallèle d'une vidéo avec des images ou des animations, accompagnée d'autres données, comme le texte et l'accès aux fichiers qui peuvent être téléchargés. L'encadrement pour cette application apparaît avec les besoins d'une formation continue d'un centre hospitalier, cependant l'application a été aussi développée de manière à être utilisée par d'autres institutions.

La spécification consiste à décrire le problème et son modelage en utilisant une méthode pour le développement des applications interactives, le WISDOM. L'implémentation décrit le système développé, la justification des options technologiques et les résultats d'un questionnaire réalisé aux utilisateurs du système.

Le développement du système a conduit à l'étude et, en cas spécifiques, à l'utilisation d'un ensemble de différentes technologies. Les études initiales ont abordé les technologies qui donnent la possibilité d'utiliser la vidéo sur l'Internet et quelques-unes de ses limitations. Ensuite apparaissent la composition, la synchronisation, et l'interactivité multimédia et finalement l'usage des métadonnées pour les contenus éducatifs.

La réponse présentée par les utilisateurs pendant l'utilisation du système a été très satisfaisante et a démontré un grand intérêt par l'usage de la vidéo sur l'Internet. La construction des applications pour l'Internet, en utilisant des médias exigeants, n'est plus la tâche complexe qui était auparavant, car les évolutions technologiques ont permis la diminution de ces difficultés. Néanmoins le développement doit encore être bien pondéré et dépend de l'encadrement de plusieurs facteurs en rapport avec la codification des media et avec l'hétérogénéité des réseaux, des utilisateurs et des équipements.

Agradecimentos

Agradeço ao Prof. Eurico Carrapatoso por ter aceite a orientação desta dissertação e a pela forma atenta e participada como a fez, mostrando grande disponibilidade e deixando sugestões e críticas, as quais considere interessantes e pertinentes, em todos os momentos do trabalho.

Agradeço à direcção do Departamento de Matemática e Engenharia da Universidade da Madeira, na pessoa do Prof. José Carmo, pelo constante interesse e apoio a vários níveis durante estes meses de trabalho. Agradecimento que estendo a todos os colegas do departamento pelo bom ambiente de discussão e reflexão que proporcionaram.

Ao Dr. Manuel Pereira e a toda a sua equipa no Centro Hospitalar do Funchal agradeço o tempo cedido e as opiniões relevantes que muito contribuíram para a especificação de um sistema mais completo e adequado.

Ao Prof. Carlos Oliveira agradeço as sugestões que muito apoiaram a realização deste trabalho.

Agradeço ao Prof. Nuno Nunes, Eng. Pedro Valente, Eng. Pedro Campos e Eng. Duarte Costa os esclarecimentos, as sugestões e as opiniões que deram no âmbito da especificação do sistema.

Um agradecimento à instituição Centro de Ciência e Tecnologia da Madeira (CITMA) o apoio dado sob a forma de uma bolsa individual de formação.

Agradeço à Eng.^a Lina Brito e aos alunos de Engenharia de Redes e Dados por se terem disponibilizado para ajudar na criação dos conteúdos para teste. Agradeço também a eles e a um restante grupo de pessoas a disponibilidade que tiveram para testar o sistema *inVÍdeo*, deixando opiniões e críticas construtivas.

À Dr.^a Carol Marques, à Dr.^a Elisabete Silva e ao Sr. Fernando Letra agradeço o tempo que tiveram para me apoiar na redacção deste trabalho.

Ao Sr. Nelson Vasconcelos agradeço a amizade e a ajuda prestada durante estes dois anos de mestrado.

Desejo ainda deixar um agradecimento especial aos meus tios, Luísa Alves e José Gonçalves, por me terem recebido tão bem e muito ajudado nas minhas constantes deslocações ao Porto.

Índice

Resumo	i
Abstract	ii
Résumé	iii
Agradecimentos	iv
Índice	v
Lista de Figuras	ix
Lista de Tabelas	xi
Acrónimos	xii
1. Introdução	1
1.1 Motivação	2
1.2 Objectivos	4
1.3 Organização	5
2. Vídeo sobre a Internet	6
2.1 Introdução	7
2.2 Modos de distribuição	7
2.2.1 <i>Download</i>	7
2.2.2 <i>Streaming</i>	8
2.2.2.1 <i>Unicast</i> e <i>multicast</i>	9
2.3 Condicionalismos	10
2.3.1 Enumeração das dificuldades	11
2.3.1.1 Largura de banda	11
2.3.1.2 Atraso	11
2.3.1.3 Perdas	11
2.3.1.4 Heterogeneidade	12
2.3.1.5 Segurança	12
2.3.2 Métodos de resolução	13
2.3.2.1 Controlo de congestão	13
2.3.2.2 Controlo de erros	14
2.4 Arquitectura de um serviço de <i>streaming</i>	15
2.4.1 Compressão de vídeo	15
2.4.2 Serviços de distribuição de <i>media</i> contínuos	16
2.4.3 Servidores de <i>streaming</i>	17
2.4.3.1 Sistema operativo	17
2.4.3.2 Armazenamento	17
2.4.4 Mecanismos de sincronização de <i>media</i>	18
2.5 Protocolos	18
2.5.1 RTP	19
2.5.1.1 Funcionamento	19

2.5.2	RTCP.....	20
2.5.3	RTSP	21
2.6	QoS centrada na rede	22
2.6.1	Serviços integrados	22
2.6.1.1	Nível de serviço	23
2.6.1.2	Interface do serviço	23
2.6.1.3	Controlo de admissão.....	24
2.6.1.4	Mecanismos de agendamento dentro da rede	24
2.6.1.5	Limitações.....	24
2.6.2	Serviços diferenciados	25
2.7	Conclusão	26
3.	Composição, sincronização e interacção multimédia	28
3.1	Introdução.....	29
3.2	Linguagem de integração de multimédia sincronizada.....	29
3.2.1	Estrutura básica de um documento SMIL	29
3.2.1.1	Cabeçalho	30
3.2.1.2	Corpo.....	30
3.2.2	Grupos Funcionais	31
3.2.2.1	Animação.....	32
3.2.2.2	Controlo de conteúdo	33
3.2.2.3	Composição.....	33
3.2.2.4	Ligação	34
3.2.2.5	Objectos.....	35
3.2.2.6	Meta-informação	36
3.2.2.7	Estrutura	36
3.2.2.8	Temporização e sincronização.....	36
3.2.2.9	Manipulação Temporal	37
3.2.2.10	Transições	38
3.2.3	Perfis de implementação	38
3.3	MPEG-4.....	39
3.3.1	Composição	40
3.3.1.1	BIFS versão 2	41
3.3.1.2	Formato textual.....	41
3.3.2	Sincronização	42
3.3.3	Interacção.....	43
3.4	Conclusões.....	43
4.	Metadados – Modelo de referência de conteúdos reutilizáveis.....	44
4.1	Introdução.....	45
4.2	Organização.....	47
4.3	Modelo de agregação de conteúdo	48
4.3.1	Componentes do modelo de conteúdo.....	49
4.3.1.1	Objectos simples	49
4.3.1.2	Conteúdos partilháveis.....	49
4.3.1.3	Organização de conteúdos.....	50
4.3.2	Empacotamento de conteúdos.....	51
4.3.3	Componentes de um manifesto.....	52
4.3.3.1	Metadados	52
4.3.3.2	Organizações.....	53

4.3.3.3 Recursos	54
4.3.3.4 Ficheiros	55
4.3.4 Perfis de pacotes de conteúdos	55
4.3.5 Metadados	57
4.4 Sequência e navegação	58
4.4.1 Sequência	58
4.4.1.1 Árvore de actividades	59
4.4.1.2 Clusters	59
4.4.1.3 Sub-manifestos	60
4.4.1.4 Actividade de ensino	61
4.4.1.5 Tentativa	61
4.4.1.6 Sessão de sequência	62
4.4.1.7 Monitorização do estado da actividade	62
4.4.1.8 Objectivos de ensino	62
4.4.2 Navegação	63
4.5 Ambiente de execução	64
4.6 Conclusão	65
5. Especificação do sistema <i>inVÍdeo</i>	66
5.1 Introdução	67
5.2 O método WISDOM	67
5.3 Descrição	69
5.4 Casos de utilização	71
5.5 Diagramas de actividades	72
5.6 Diagrama de análise	74
5.7 Arquitectura conceptual do sistema	76
5.8 Modelo de desenho	77
5.9 Modelo de apresentação	79
5.10 Refinamento dos espaços de interacção	79
5.11 Conclusão	82
6. Implementação do sistema <i>inVÍdeo</i>	83
6.1 Opções tecnológicas	84
6.1.1 Ferramentas e linguagens	84
6.1.1.1 Opções	84
6.1.1.2 Ferramentas e linguagens escolhidas	85
6.1.2 Norma para a importação e exportação	87
6.2 Apresentação do sistema	88
6.2.1 Diagrama de componentes	89
6.2.2 Ecrãs da aplicação cliente	90
6.2.2.1 Ecrã de acesso	90
6.2.2.2 Ecrã dos cursos	91
6.2.2.3 Ecrã dos módulos e das sessões	92
6.2.2.4 Ecrã da lição	93
6.2.3 Excertos do código	95
6.2.3.1 Implementação da sincronização	95
6.2.3.2 Implementação das comunicações	96
6.2.3.3 Implementação dos controlos	97
6.3 Utilização da aplicação	98
6.3.1 Utilizadores e características técnicas	99

6.3.2	Apreciação da utilização da aplicação.....	100
6.3.3	Análise aos resultados do inquérito.....	101
6.4	Conclusão	102
7.	Conclusões.....	103
7.1	Conclusões.....	104
7.2	Desenvolvimentos futuros	107
	Referências.....	109
	Bibliografia	112
	Anexo A – Documento SMIL.....	113
	Anexo B – Manifesto SCORM.....	115
	Anexo C – Resultados do inquérito	120

Lista de Figuras

Figura 2-1 Componentes de um sistema de <i>streaming</i>	9
Figura 2-2 Exemplo de <i>unicast</i> e <i>multicast</i>	10
Figura 2-3 Arquitectura para o <i>streaming</i> de vídeo.....	15
Figura 2-4 Pilha protocolar do <i>streaming</i>	19
Figura 2-5 Sinalização feita pelo RSVP	23
Figura 2-6 Âmbito dos serviços integrados e serviços diferenciados	25
Figura 3-1 Exemplo de temporização	31
Figura 3-3 Estrutura lógica de uma cena.....	40
Figura 3-3 <i>eXtensible</i> MPEG-4 <i>Textual Format</i> (XMT).....	42
Figura 4-1 Organização geral do SCORM.....	48
Figura 4-2 Exemplos de objectos simples.....	49
Figura 4-3 Exemplo de um conteúdo partilhável (SCO).....	50
Figura 4-4 Exemplo de organização de conteúdos	51
Figura 4-5 Diagrama conceptual do pacote de conteúdos.....	52
Figura 4-6 Componentes de um manifesto.....	52
Figura 4-7 Terminologia para a hierarquia de conteúdos	54
Figura 4-8 Recursos de um manifesto	55
Figura 4-9 Exemplo de um SCO representado como um elemento <resource>	55
Figura 4-10 Exemplo de um pacote de agregação de conteúdo.....	56
Figura 4-11 Exemplo de uma árvore de actividades	59
Figura 4-12 Exemplo de cluster	59
Figura 4-13 Conversão de um manifesto com sub-manifestos em árvore de actividades	60
Figura 4-14 Exemplo de uma actividade de ensino.....	61
Figura 4-15 Ambiente de execução conceptual do SCORM	64
Figura 5-1 Processo WISDOM [Nunes 01b].....	68
Figura 5-2 Organização dos modelos do WISDOM [Nunes 01b].....	68
Figura 5-3 Extensões WISDOM para o modelo de interacção	69
Figura 5-4 Casos de utilização da aplicação	71
Figura 5-5 Diagrama de actividade de 'Realizar Sessão'	73
Figura 5-6 Diagrama de actividades de 'Criar Sessão'	74
Figura 5-7 Diagrama de análise do caso de utilização 'Realizar Sessão'	75
Figura 5-8 Arquitectura conceptual do sistema	77
Figura 5-9 Diagrama de classes	78

Figura 5-10 Diagrama de apresentação do aluno	79
Figura 5-11 Componentes abstractos canónicos	80
Figura 5-12 Refinamento do espaço de interacção 'Realizar Lição'	80
Figura 5-13 Refinamento do espaço de interacção 'Criar Lição'	81
Figura 6-1 Diagrama de componentes do sistema <i>inVÍdeo</i>	89
Figura 6-2 Ecrã de acesso à aplicação	90
Figura 6-3 Ecrã com a lista de cursos	91
Figura 6-4 Ecrã da lista de módulos e das sessões	92
Figura 6-5 Ecrã da realização de uma lição	93
Figura 6-6 Conhecimentos dos utilizadores e características dos computadores e acessos	100
Figura 6-7 Gráfico com os valores obtidos por cada componente	101
Figura A-1 Cena SMIL	113
Figura B-1 Representação gráfica de um manifesto	115

Lista de Tabelas

Tabela 4-1 Eventos de navegação e sua descrição	63
Tabela 6-1 Comparação de servidores de <i>streaming</i>	85

Acrónimos

ADL	<i>Advanced Distributed Learning</i>
ADRIANE	<i>Alliance of Remote Instructional Authoring & Distribution Networks for Europe</i>
ADSL	<i>Asymmetric Digital Subscriber Line</i>
AICC	<i>Aviation Industry CBT Committee</i>
API	<i>Application Program Interface</i>
ASP	<i>Active Server Pages</i>
CAM	<i>Content Aggregation Model</i>
CBI	<i>Computer Based Instruction</i>
DCMI	<i>Dublin Core Metadata Initiative</i>
DIFFSERV	<i>Differentiated Services</i>
ECMA	<i>European Computer Manufacturers Association</i>
FEC	<i>Forward Error Correction</i>
FGS	<i>Fine Granularity Scalability</i>
HTML	<i>Hypertext Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IETF	<i>Internet Engineering Task Force</i>
IMS	<i>IMS Global Learning Consortium, Inc.</i>
INTSERV	<i>Integrated Services</i>
IP	<i>Internet Protocol</i>
ITS	<i>Intelligent Tutoring Systems</i>
LOM	<i>Learning Object Metadata</i>
LMS	<i>Learning Management System</i>
LTSC	<i>Learning Technology Standards Committee</i>
MPEG	<i>Motion Picture Experts Group</i>
PCM	<i>Pulse Code Modulation</i>
PDA	<i>Personal Digital Assistant</i>
PHP	<i>PHP Hypertext Pre-processor</i>
QoS	<i>Quality of Service</i>
REDIS	<i>Rede Digital com Integração de Serviços</i>
RTCP	<i>Real Time Transport Control Protocol</i>
RTE	<i>Run Time Environment</i>
RTMP	<i>Real Time Messaging Protocol</i>
RTP	<i>Real Time Transport Protocol</i>
RTSP	<i>Real Time Streaming Protocol</i>
RSVP	<i>Resource Reservation Protocol</i>
SCO	<i>Sharable Content Object</i>
SCORM	<i>Sharable Content Object Reference Model</i>

SLA	<i>Service Level Agreement</i>
SMIL	<i>Synchronized Multimedia Integration Language</i>
SN	<i>Sequência e Navegação</i>
SO	<i>Sistema Operativo</i>
SQL	<i>Simple Query Language</i>
SS	<i>Sequência Simples</i>
TCP	<i>Transmission Control Protocol</i>
UDP	<i>User Datagram Protocol</i>
UMa	<i>Universidade da Madeira</i>
UML	<i>Unified Modelling Language</i>
URI	<i>Universal Resource Identifier</i>
URL	<i>Universal Resource Locator</i>
VRML	<i>Virtual Reality Modeling Language</i>
X3D	<i>eXtensible 3D</i>
XML	<i>eXtensible Markup Language</i>
XMT	<i>eXtensible MPEG-4 Textual Format</i>
W3C	<i>World Wide Web Consortium</i>
WISDOM	<i>Whitewater Interactive System Development with Object Models</i>

1. Introdução

1.1 Motivação

A utilização da Internet, desde o seu aparecimento, tem-se reinventado e recriado continuamente. Tomando como exemplo a sua aplicação mais conhecida, a *web*, criada por Tim Berners-Lee em 1990, integrava inicialmente apenas texto e imagem. Rapidamente foram integrados outros *media*, como a animação, o áudio e o vídeo, devido a um interesse crescente dos utilizadores e como forma de cativar novos utilizadores. A animação é introduzida nas mais variadas aplicações, mas o áudio e o vídeo necessitaram de mais tempo para poderem ser utilizados, devido a serem muito maiores em tamanho relativamente aos outros *media* e às múltiplas redes que constituem a Internet não terem sido projectadas para *media* contínuos. Apenas nos últimos anos, devido às melhorias nas tecnologias de comunicação e compressão e ao aumento das capacidades de processamento e armazenamento, a criação de aplicações multimédia ricas sobre a Internet se tornou viável e atractiva para os criadores de conteúdos. Este facto não significa que o desenvolvimento de aplicações com áudio e vídeo não tenha existido. Apenas surgiam para implementações específicas e para um conjunto de utilizadores conhecido e limitado.

A criação de conteúdos para visualização nos navegadores *web* de um qualquer utilizador sempre seguiu o princípio do mínimo denominador comum, que visa garantir o acesso dos utilizadores à quase totalidade dos conteúdos presentes na Internet. E considerou-se o acesso telefónico, tipicamente tendo uma largura de banda máxima próxima dos 56 Kbps, como esse mínimo denominador comum. A produção de conteúdos seguindo esta regra permitiu que nenhum utilizador se sentisse tecnologicamente posto de lado no acesso à grande maioria da informação existente, mas também limitou a possibilidade de existirem aplicações mais ricas e interessantes. Unicamente os conteúdos sem exigências de taxas de transmissão e de garantias de atraso, isto é, os *media* que não são utilizados em tempo real, como o texto e a imagem, foram usados exaustivamente na produção de informação para a *web*.

A utilização de áudio e do vídeo apenas se tornaram viáveis com o surgimento de algumas tecnologias e com a sua crescente adopção pelos utilizadores. As tecnologias de rede de banda larga e a grande capacidade de processamento e armazenamento existentes nos computadores pessoais actuais foram as razões tecnológicas mais fortes para o incremento da utilização de *media* ricos.

A utilização do áudio e do vídeo sobre a Internet tem tido um crescimento considerável nos últimos anos. Já são muitas as empresas, principalmente as relacionadas com a produção de conteúdos, sejam informativos, documentais ou de entretenimento, que não abdicam do recurso a estes meios. Temos como exemplo os resumos dos filmes e blocos de notícias. O interesse na criação de aplicações criativas e apelativas utilizando o áudio e o vídeo é assim cada vez maior.

Os dados de áudio e vídeo podem ser obtidos de duas formas distintas. A primeira, que é como aceder a qualquer ficheiro na Internet, designa-se por modo *download* (modo que ainda se pode subdividir em normal e progressivo, este último permitindo descarregar o ficheiro e em paralelo visualizar os dados já presentes no terminal do cliente). A segunda é o modo *streaming*,

cada vez mais usado, recorrendo a um servidor especializado, designado por servidor de *streaming*. Os dados enviados por este servidor são codificados para formatos específicos e enviados para o cliente sobre protocolos especificamente definidos para o transporte de dados em tempo real. Estas duas formas de disponibilizar conteúdos são bastantes diferentes e apresentam, cada uma delas, vantagens e desvantagens. Destacam-se as seguintes vantagens: a simplicidade por parte do *download*, visto que não necessita de nenhum servidor específico, e a flexibilidade por parte do *streaming*, que permite a audição e/ou visualização quase imediata de qualquer parte do conteúdo.

A definição de cenas multimédia, que contempla a composição, sincronização e interacção, também tem tido desenvolvimentos nos últimos anos. O aparecimento de normas para cada *media* é demorado e envolve diversas disputas. O surgimento de uma norma que agregue os diferentes *media*, os sincronize e permita a sua interacção ainda tem sido mais difícil e complexo, com a consequência de neste momento não existir nenhuma norma amplamente seguida. Têm aparecido várias normas, umas proprietárias e outras públicas, mas nenhuma tem atingido uma implementação e aceitação alargadas. Por parte de cada criador de ferramentas para produção e apresentação de conteúdos tem existido uma filosofia de defesa dos seus formatos, mantendo uma incompatibilidade com outros produtos. Apesar disso, algumas normas públicas já começam a ter alguma maturidade e apresentam-se como opções credíveis para os vários intervenientes. Os autores e produtores visam a eficiência do seu trabalho procurando gerar conteúdos que possam ter uma base de utilização mais alargada. Os distribuidores procuram fornecer serviços de qualidade, mas a complexidade e o grande número de normas e formatos surgem como dificuldades para atingir esse fim. E os clientes querem simplicidade no acesso aos conteúdos multimédia, por intermédio de uma aplicação única que permita aceder a diferentes conteúdos.

As normas públicas, cada vez mais aperfeiçoadas, não descurem as funcionalidades que permitem uma experiência mais rica aos utilizadores. A visualização simples já não é a mais valia nos produtos actuais, o caminho mais utilizado é a interactividade. Os utilizadores de hoje não querem apenas receber, de forma passiva, a informação que alguém criou para eles, ao invés, procuram a possibilidade de ter uma atitude activa perante essa informação. A interacção poderá passar por alterar a sequência de visualização, optar por determinados elementos e ter acesso a complementos de informação.

Uma preocupação existente no mundo multimédia, e isto deve-se a um crescente aumento na produção de conteúdos, é a sua organização e referência, para acesso simplificado e futuras reutilizações. A necessidade de rapidamente se obterem conteúdos específicos não seria satisfeita se não existisse uma forma bem definida e clara de os descrever e armazenar. A produção de conteúdos multimédia, em relação a conteúdos mais simples como apenas texto, ou texto e imagem, é bastante mais demorada, complexa e dispendiosa. A reutilização, uma funcionalidade cada vez mais exigida devido aos ganhos de produtividade que traz, apenas é possível se houver uma forma simples de referir e referenciar os conteúdos já produzidos. Os metadados, método de associar informação variada aos dados, possibilitam efectuar sobre os dados acções de pesquisa e descoberta para novas utilizações. Além destas preocupações é também dada atenção à

reutilização dos conteúdos em diferentes ambientes e plataformas, como forma de otimizar a sua utilização.

O áudio e vídeo sobre a Internet, na sua vertente de *streaming*, têm já muitas e diversas possibilidades de aplicação como Rádio Web, Vídeo a Pedido, Ensino à Distância, Videoconferência, Cinema Digital, Câmaras de *Streaming* e outras ainda emergentes como *Streaming* Móvel e de Alta Definição. Se as utilizações já existem, a sua difusão e aceitação são ainda reduzidas, isto devido às limitações tecnológicas já referidas e ainda presentes em muitos casos. Onde estas limitações já foram ultrapassadas, a existência de aplicações adequadas e atraentes ainda é em pequeno número e com funcionalidade reduzida. De futuro espera-se o surgimento de aplicações que cativem e impulsionem a utilização do áudio e vídeo sobre a Internet.

1.2 Objectivos

O objectivo principal desta dissertação era o desenvolvimento de uma aplicação para formação/educação num ambiente multimédia, privilegiando o vídeo como meio central e tendo ainda presente uma componente de interacção. A aplicação deveria organizar os conteúdos em cursos e a apresentação do vídeo seria acompanhada com outros *media*, como imagem e texto. Para o sistema a construir estavam ainda previstas outras funcionalidades como: importação e exportação de conteúdos, acesso a informação complementar e integração de formas de comunicação síncrona e assíncrona entre os utilizadores.

Para conseguir atingir este objectivo, julgou-se que seria importante desenvolver estudos sobre: utilização do vídeo na Internet; algumas normas de composição, sincronização e interactividade multimédia; e o uso de metadados para os conteúdos educacionais. O estudo do vídeo sobre a Internet passava pela análise das tecnologias de suporte e pelos modos de poder ser disponibilizado. Na temática da composição, sincronização e interactividade sobre vídeo seriam estudadas algumas normas mais comuns e utilizadas nos *media players* e nos navegadores. Nos metadados seria abordada uma especificação vocacionada para a organização e reutilização de conteúdos educacionais em ambientes distribuídos.

Para o desenvolvimento da aplicação estavam previstas a sua especificação e implementação. Na especificação deveria ser usado um método para o desenvolvimento de *software* centrado nos utilizadores e guiado por casos de utilização essenciais. Para a implementação deveriam ser inicialmente avaliadas várias opções para ferramentas e linguagens para a codificação e funcionamento do sistema. Seguir-se-ia a implementação dos componentes essenciais para validar a solução proposta.

1.3 Organização

Esta dissertação encontra-se organizada em sete capítulos, estando no capítulo 1, “Introdução”, a enumeração e a descrição dos temas que irão ser abordados nesta dissertação.

No capítulo 2, “Vídeo sobre a Internet”, são abordados os modos de transmissão, as condicionantes que limitam a utilização do vídeo sobre a Internet, os componentes de uma arquitectura de vídeo, os protocolos para dados em tempo real e duas arquitecturas para a implementação de qualidade de serviço centrada na rede.

O capítulo 3, “Composição, sincronização e interactividade multimédia”, apresenta duas normas públicas, o SMIL e o MPEG-4, para a composição, sincronização e interactividade em cenas multimédia.

O capítulo 4, “Metadados – Modelo de Referência de Conteúdos Reutilizáveis”, descreve uma norma que permite a reutilização de conteúdos educacionais em ambientes distribuídos, o SCORM.

O desenvolvimento da aplicação é descrito nos capítulos 5 e 6, respectivamente, “Especificação do sistema *inVídeo*” e “Implementação do sistema *inVídeo*”. No capítulo 5 é modelado o sistema usando o método WISDOM. Destacam-se os casos de utilização, o diagrama de análise, o refinamento dos espaços de interacção e a arquitectura global do sistema. No capítulo 7 são descritas as opções de implementação, a nível das ferramentas e das tecnologias escolhidas; são apresentados alguns ecrãs e excertos de código desenvolvido; e analisados alguns resultados, baseados num inquérito, sobre a aplicação desenvolvida.

O capítulo 7 apresenta as conclusões desta dissertação bem como algumas propostas de trabalho futuro a desenvolver.

2. Vídeo sobre a Internet

O capítulo actual debruça-se sobre alguns aspectos relacionados com a utilização do vídeo sobre a Internet.

Inicialmente é feita uma comparação entre os diferentes modos de transmissão do vídeo, onde se enumeram as vantagens e desvantagens de cada um.

De seguida são referidos os elementos que condicionam a qualidade de serviço no uso do vídeo sobre a Internet e os avanços tecnológicos que vieram a facilitar o seu uso.

O capítulo continua com a apresentação de uma arquitectura para o *streaming* de vídeo e com a descrição dos seus principais componentes.

Surgem também os principais protocolos usados na Internet para a transmissão de dados em tempo real.

No final são apresentadas duas arquitecturas para a implementação da qualidade de serviço centrada nas redes.

2.1 Introdução

A utilização de vídeo sobre a Internet já existe há alguns anos, mas sempre com um número de utilizações bastante reduzido se comparado com o crescimento da Internet em termos de número de utilizadores e quantidade de informação. Esta pouca utilização deve-se, basicamente, a duas categorias de problemas: a produção e a distribuição. Na produção temos todo um processo complexo e exigente em termos tecnológicos (captura, edição e conversão do vídeo). Na distribuição temos a heterogeneidade das redes (diferentes larguras de banda, tempos e variações no atraso e perdas) e dos terminais (diferentes capacidades de processamento e armazenamento) que dificultam uma utilização mais alargada do vídeo sobre a Internet.

A produção de vídeo digital tem um ciclo onde se inclui a captura, a edição e a conversão. Na captura é necessário equipamento, que para a criação de vídeo de boa qualidade, dispendioso e de manuseamento complexo. A edição exige aplicações muito específicas, também dispendiosas e cujo tempo de aprendizagem e adaptação é extenso. A conversão exige equipamento com grandes capacidades de processamento e armazenamento.

A produção de vídeo digital está fora do âmbito desta dissertação, mas dada a sua importância foi deixado este apontamento no parágrafo anterior.

Os elementos que serão apresentados neste capítulo prendem-se com os aspectos que surgem depois da produção como: os modos de distribuição, os principais condicionantes, os componentes de uma arquitectura para o serviço de *streaming*, os protocolos mais usados e arquitecturas para a implementação de qualidade de serviço centrada nas redes.

2.2 Modos de distribuição

A distribuição de vídeo na Internet, actualmente, pode ser feita de duas formas distintas: via *download*, utilizando um qualquer servidor de documentos *web*; e por *streaming*, onde é utilizado um servidor especializado.

2.2.1 Download

O modo *download* dá ao vídeo o mesmo tipo de tratamento dado aos restantes documentos obtidos via *web*. O utilizador tem de esperar pela chegada completa do ficheiro ao seu computador e apenas nessa altura poderá visualizar o vídeo. As ligações lentas e os ficheiros normalmente grandes obrigavam a períodos de descarregamento longos e assim esta forma de entrega do vídeo nunca cativou, por razões óbvias, a grande maioria de utilizadores da Internet.

Um método que permitiu melhorar o *download* foi o *download* progressivo. Este método, que utiliza uma codificação específica, permite aos utilizadores visualizar o vídeo durante o processo de *download*, ou seja, os dados já descarregados podem ser decodificados e apresentados ao utilizador.

A melhoria obtida pelo método do *download* progressivo não permitiu ultrapassar um

conjunto de limitações na utilização do vídeo. Por um lado não permitia a implementação de uma gama de serviços, do qual é exemplo a emissão de eventos em tempo real; e por outro a dificuldade em garantir os direitos do autor do vídeo, uma vez que o vídeo era sempre descarregado para o computador do utilizador, possibilitando a cópia do ficheiro e a obtenção de uma cópia fidedigna do vídeo.

O novo método que surgiu para ultrapassar estas e outras limitações foi o *streaming* de vídeo, recorrendo a servidores especializados. Este método permitiu um conjunto de novos serviços sobre o vídeo, melhoria na eficiência da entrega e uma nova abordagem à resolução dos problemas relacionados com os direitos de autor.

Os problemas na obtenção do vídeo pelo método de *download* (simples ou progressivo) não são diferentes dos problemas de outros quaisquer ficheiros da *web*. Como o tema desta dissertação se foca apenas nas questões relacionadas com o vídeo não foi relevante explorar este caminho. Assim, este capítulo será dedicado às tecnologias relacionadas com o *streaming*, não deixando de haver pontos onde a discussão e a resolução de um problema ou limitação no vídeo por *streaming* não possa ser usado no *download*.

2.2.2 Streaming

O termo *streaming*, resultante de *stream*, pode-se traduzir como geração do fluxo (de informação). O *streaming* de vídeo é um método de distribuição que, contrariamente ao *download*, não necessita de obter o ficheiro por completo para permitir a sua visualização. O cliente pode, ao mesmo tempo, visualizar o primeiro pacote de vídeo, descomprimir o segundo e receber o terceiro [Liu 99].

Além desta vantagem, que já existia no *download* progressivo, o *streaming* não deixa nenhuma cópia dos dados no cliente, ou seja, as cenas após serem apresentadas são apagadas, não impondo grandes requisitos de espaço de armazenamento do lado do cliente. A este facto também podemos associar uma outra vantagem que é dificultar a cópia dos vídeos e assim proteger melhor os direitos do seu autor¹.

Outras grandes vantagens do *streaming* relativamente ao *download* são: permitir a emissão de eventos em tempo (quase) real, e melhoria na interactividade com o vídeo. Sobre o ponto da interactividade com o vídeo, o método de *streaming* permite “saltar” rapidamente para um momento específico do vídeo, possibilitando uma visualização não linear. Com o *download* progressivo se quisermos assistir aos segundos finais de um filme teremos sempre de esperar que todo o ficheiro seja descarregado.

O *streaming* para acontecer necessita do lado do servidor de uma aplicação específica que disponibilize o vídeo com as características referidas, designada servidor de *streaming*. Do lado do cliente necessita de uma aplicação que apresente o vídeo, designada *player*. A figura 2-1 apresenta as relações entre os componentes de um serviço de *streaming*. Os três componentes são: a

¹ Esta vantagem apenas dificulta, mas não impede por completo a cópia. Os programas clientes (os designados *players*) impedem a gravação local do vídeo, visto não terem essa funcionalidade implementada. Mas não impedem que utilizadores mais avançados, usando programas específicos, possam obter uma cópia integral e ilegal de qualquer vídeo disponibilizado por um servidor de *streaming*.

codificação, o cliente e o servidor.

A codificação usa formatos (estrutura do ficheiro de vídeo codificado) que o servidor de *streaming* entende e pode disponibilizar. Entre a codificação e o cliente terão de existir, respectivamente, codificadores e decodificadores (este par de termos é designado de forma corrente como *codecs*, da abreviação do inglês *COder/DECoder*) compatíveis. Finalmente, a entrega dos vídeos por parte do servidor ao cliente deverá ser baseada em protocolos (apresentados alguns na secção 2.5) conhecidos por ambos.

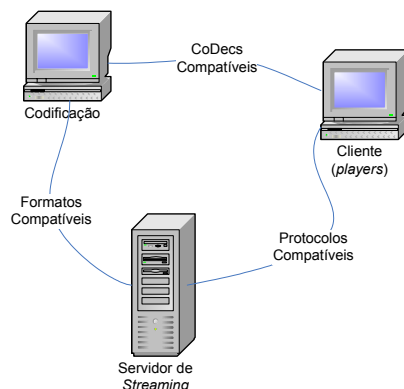


Figura 2-1 Componentes de um sistema de *streaming*

A utilização de um servidor especializado traz várias vantagens [Adobe 03]:

- **uso mais eficiente da largura de banda disponível:** de forma manual pelo cliente ou automática pelo servidor, pode ser escolhida a versão do vídeo mais adequada a uma certa largura de banda;
- **vídeo de melhor qualidade para o cliente:** no seguimento da vantagem anterior, a qualidade de visualização é melhor devido à escolha adequada da largura de banda que evitará variações na qualidade do vídeo recebido;
- **suporte para um grande número de utilizadores:** o servidor está melhor preparado para atender a grandes quantidades de pedidos de vídeos;
- **opções de entrega variadas:** os vídeos podem ser enviados via *multicast* e/ou *unicast* (apresentado na secção seguinte);
- **protecção dos direitos de autor sobre o conteúdo:** pelas razões já referidas no início desta secção.

2.2.2.1 *Unicast e multicast*

O servidor de *streaming* pode servir os fluxos de vídeo de duas formas: *unicast* e *multicast*. No *unicast* é estabelecida uma ligação ponto a ponto entre o servidor e um cliente, no caso no *multicast* é estabelecida uma ligação entre o servidor e vários clientes.

A figura 2-2 exemplifica os fluxos de informação em ambiente *unicast* (i) e *multicast* (ii). Neste exemplo e para o *unicast* temos quatro fluxos de 75 Kbps do servidor para os clientes, necessitando

portando de uma ligação ao servidor de 300 Kbps. No caso do *multicast* temos apenas um fluxo de saída do servidor com 75 Kbps até ao *router* (que terá de ter obrigatoriamente capacidades para o *multicast*) que se distribui para os diversos clientes. Este exemplo é muito simples mas já demonstra algumas das vantagens e desvantagens de cada modo.

A vantagem do *unicast* é, basicamente, a possibilidade de interacção com o servidor, visto que para cada cliente haverá uma ligação. Sendo as ligações independentes umas das outras, permite a cada cliente escolher a sua sequência de visualização (excepto emissões em tempo real). As desvantagens são: o processamento que impõe ao servidor, uma vez que cada ligação é gerida independentemente mesmo que várias ligações transportem exactamente os mesmos dados; e os requisitos de rede que relativamente ao modo *multicast* são maiores.

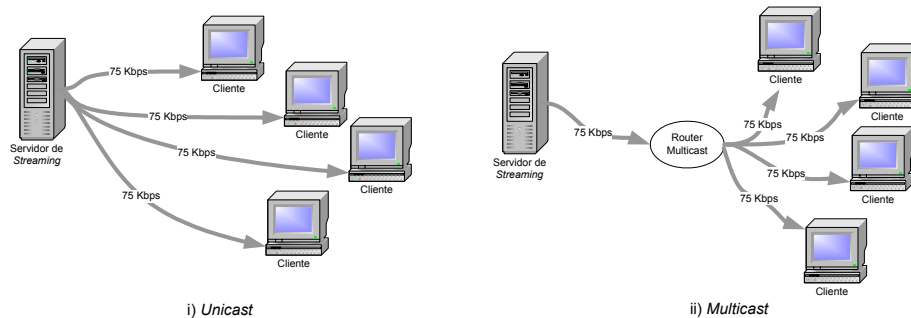


Figura 2-2 Exemplo de *unicast* e *multicast*

As vantagens do *multicast* são impor ao servidor uma carga menos pesada e diminuir os requisitos de rede, pois apenas existe um fluxo de dados (neste exemplo). A grande desvantagem é a impossibilidade de interactividade. Mas existe outra limitação, enquanto o *unicast* não impõe funcionalidades especiais na rede, o *multicast* necessita de equipamentos activos (*routers* e *switches*) que suportem essa função, aumentando os custos de um serviço de *streaming* que queira usar o *multicast*.

O *unicast* não é melhor que o *multicast* ou *vice-versa*. Para cada situação de *streaming* existe uma forma mais adequada que outra. O serviço designado por Vídeo a Pedido (um cliente poderá assistir a um filme na hora que pretender) apenas poderá funcionar com *unicast*. Mas um serviço noticioso em tempo real interno a uma empresa já poderá usar o *multicast*, e desta forma não sobrecarregar a sua rede, independentemente do número de utilizadores que estejam a assistir.

O *unicast* é mais flexível por permitir mais serviços, mas é mais exigente em termos de largura de banda e de carga sobre o servidor. O *multicast* é mais eficiente, em termos da rede e do servidor, mas a gama de serviços que permite é mais reduzida.

2.3 Condicionalismos

As dificuldades da distribuição de vídeo sobre a Internet podem ser resumidas na questão de o vídeo ser um *media* exigente num meio que não aceita exigências. A Internet permite a troca de qualquer tipo de ficheiro, mas não possibilita a definição de um tempo preciso ou aproximado

para essa troca. Para a maioria dos serviços sobre a Internet, como *e-mail* ou páginas *web*, os atrasos que acontecem não produzem problemas graves. Já os *media* em tempo real têm que ser entregues com um atraso pequeno e com poucas perdas, senão o risco da sua apresentação baixar um patamar mínimo de qualidade pode aumentar.

De seguida serão apresentadas as dificuldades no uso do vídeo e para aquelas relacionadas com a qualidade de serviço (QoS) serão apresentadas duas técnicas de resolução.

2.3.1 Enumeração das dificuldades

As dificuldades no uso do vídeo são cinco. As primeiras três - largura de banda, atraso, perdas - relacionam-se com a problemática da QoS e as últimas duas - heterogeneidade e segurança - também relevam, respectivamente, algumas questões da problemática da entrega do vídeo a tempo e com qualidade, e da questão da garantia dos direitos dos autores.

2.3.1.1 Largura de banda

A transmissão de vídeo em tempo real sobre a Internet necessita de uma largura de banda mínima para poder ser executada. Mesmo considerando a transmissão de um vídeo com uma qualidade mínima, serão usados 28 Kbps [Wu 01b]. Mas não temos na Internet actual a possibilidade de reservar uma largura de banda, mesmo tão baixa.

As limitações dos equipamentos activos, como *routers* e *switchs*, no que diz respeito ao controlo de congestão, leva a que um pico de tráfego pode causar a perda do fluxo de dados e comprometer a entrega do vídeo em tempo real com qualidade.

2.3.1.2 Atraso

A transmissão de informação textual e de imagens não impõe qualquer requisito no atraso, isto é, se chegar num determinado momento ou alguns segundos mais tarde, esse facto não representa nenhum problema grave. O mesmo não se passa com os dados em tempo real, como o vídeo. O atraso na entrega dos dados de um vídeo não pode ser acentuado, nem inconstante. Os dados têm de chegar ao terminal do utilizador a tempo de serem decodificados e apresentados. Um atraso pode provocar quebras na apresentação e se um pacote chegar também fora de sequência, já não será útil.

O atraso não levanta apenas o problema da sua dimensão, mas também o de ser ou não constante. Se o atraso for acentuado, mas também constante, o cliente e o servidor poderão ajustar-se na comunicação dos dados, com um atraso inicial, e a restante transmissão poderá funcionar bem. Se o atraso encurtar demasiado pode colocar problemas no cliente devido ao possível enchimento do *buffer* e consequente perda de pacotes.

Mais uma vez a Internet, como rede *best-effort* (rede que funciona da melhor forma possível), não permite que se defina o atraso, podendo este ser variável dentro de uma transmissão de dados.

2.3.1.3 Perdas

A perda de pacotes pode comprometer a boa visualização de uma sequência de vídeo. Os

olhos humanos são tolerantes a pequenas falhas, mas abaixo de um certo nível a qualidade da apresentação fica comprometida. As aplicações de vídeo impõem requisitos sobre a perda de pacotes, na forma de uma percentagem máxima, para garantir uma qualidade visual aceitável.

A Internet também não pode garantir esse nível de perdas. Num período de grande congestão as perdas poderão aumentar muito e causar a degradação do vídeo.

2.3.1.4 Heterogeneidade

O problema da heterogeneidade refere-se aos múltiplos e diferentes sistemas que existem para transporte e recepção do vídeo. Então a heterogeneidade pode-se dividir em dois tipos: de rede e de receptor.

A heterogeneidade de rede refere-se às sub-redes existentes na Internet que têm características diversas como largura de banda, processamento, políticas de controlo de congestão, entre outras. Esta diferença leva a que utilizadores diferentes tenham valores distintos para o atraso e para as perdas.

A heterogeneidade dos receptores refere-se às características dos receptores, como por exemplo, diferentes requisitos de qualidade e capacidades de processamento.

As abordagens para resolver as questões da heterogeneidade são duas, uma centrada na rede, outra nos sistemas finais (cliente e servidor).

A abordagem centrada na rede exige que os *routers* e *switchs* na rede suportem QoS, de forma a garantir uma dada largura de banda, um atraso limitado e poucas perdas, e assim possibilitar o funcionamento de aplicações com vídeo [Wu 01b]. Dois exemplos desta técnica são os serviços integrados e os serviços diferenciados, ambos apresentados na secção 2.6 .

A abordagem centrada nos sistemas cliente e servidor, isto é, os sistemas em cada um dos lados da transmissão do vídeo, não coloca nenhum requisito na rede. Aborda o problema com o uso de técnicas de controlo de congestão e de controlo de erro, de forma a maximizar a qualidade do vídeo sem qualquer envolvimento por parte da rede. Os métodos de resolução usados nesta abordagem são apresentados na secção 2.3.2 .

2.3.1.5 Segurança

As dificuldades referidas até ao momento prendem-se com a qualidade do vídeo. Mas a entrega do vídeo com qualidade não encerra a questão da transmissão de vídeo sobre a Internet.

A segurança de um vídeo digital, seja para ser usado na Internet ou não, passa pelos seguintes pontos [Lin 01]:

- **acesso condicional:** os sistemas de acesso condicional definem e reforçam um modelo de acesso. Este modelo é um conjunto de regras entre o fornecedor e o cliente no acesso aos *media* contínuos;
- **autenticação:** é o processo no qual a identidade do emissor, do receptor, e a integridade do vídeo podem ser verificadas;
- **controlo de cópias:** envolve a identificação de vídeos protegidos, a definição de condições para cópias legais e assegura que o vídeo seja acedido apenas através de terminais

específicos;

- **seguir o conteúdo:** é a integração de informação de identificação no vídeo, ou seja, no caso do *streaming* é criar uma versão única para cada receptor ou grupo de receptores que permita a sua identificação em utilizações posteriores.

As ferramentas existentes para implementar a segurança num vídeo são várias. O uso de apenas uma não assegura a protecção do vídeo; todavia a utilização combinada de várias cria sistemas mais ou menos completos. As ferramentas são:

- **criptografia:** a vantagem do uso de cifras é impedir a visualização dos vídeos, excepto aos receptores com a chave para inverter a cifra. O maior problema do uso da criptografia está na questão da protecção da chave;
- **assinaturas digitais:** conjunto de códigos que visam a autenticação de conteúdos digitais;
- **marca de água:** é um sinal embebido no vídeo que não compromete a sua qualidade e que serve para o transporte da informação sobre os direitos de autor, a autenticação e para o seguimento do vídeo. A cópia de vídeo digital protegida com marca de água também copia a marca de água introduzida.

2.3.2 Métodos de resolução

As dificuldades levantadas pelos aspectos da QoS – largura de banda, atraso e perdas – podem ser minoradas por uma abordagem centrada na rede e por uma outra abordagem centrada nos sistemas finais (cliente e servidor). Os métodos apresentados de seguida apresentam técnicas usadas nesta última abordagem. A abordagem centrada na rede é feita na secção 2.6 .

2.3.2.1 Controlo de congestão

O controlo de congestão visa reduzir a perda e o atraso dos pacotes de vídeo em tempo real. As perdas e os atrasos excessivos fazem baixar consideravelmente a qualidade da visualização do vídeo e são causados pela congestão da rede. Os mecanismos para o controlo da congestão são os seguintes [Wu 01b]:

- **controlo da taxa de transmissão:** é um método que tenta minimizar a congestão na rede e os pacotes perdidos igualando a taxa de transmissão do vídeo à largura de banda disponível na rede. Sem este método, os dados enviados acima das capacidades da rede serão descartados por esta. Pode ser implementado em três esquemas: baseado no emissor, baseado no receptor e híbrido (junção de ambos);
- **codificação do vídeo com taxa adaptável:** o envio de fluxos de vídeo pelo servidor na taxa encontrada pelo algoritmo de controlo da taxa de transmissão, implica que a codificação do vídeo tenha de ser feita de tal forma que permita o envio de diferentes fluxos com diferentes taxas de transmissão. A codificação feita desta forma deverá procurar

maximizar a qualidade da visualização em cada uma das taxas a usar (taxas que poderão ser fixas ou dinâmicas dependendo do grau de congestão da rede);

- **moldagem da taxa de transmissão:** esta é uma técnica para adaptar os fluxos de vídeo comprimido às taxas de transmissão pretendidas. Esta técnica não interfere com a codificação, logo, pode ser usada em qualquer esquema da codificação de vídeo e para vídeo ao vivo ou armazenado. Existem duas aproximações para a moldagem da taxa de transmissão: uma na perspectiva de transporte (um mecanismo representativo desta perspectiva é o descartar selectivo de *frames* pelo servidor) e outra na perspectiva da compressão (um mecanismo representativo desta perspectiva é a moldagem dinâmica da taxa de transmissão).

Apesar das técnicas para controlo da congestão, a perda de pacotes é inevitável na Internet e pode trazer perdas de qualidade consideráveis. O controlo de congestão é usado com outros mecanismos, como o controlo de erros que será apresentado na secção seguinte.

2.3.2.2 Controlo de erros

A perda de pacotes de um certo tipo de dados, como o texto, é inaceitável, mas o atraso é tolerável. Podem ser usados mecanismos para a correcção de erros ou então o pacote é reenviado para ultrapassar o erro. Para o vídeo em tempo real, a degradação da qualidade, se for pequena, é aceitável, mas o atraso tem que ser limitado. A diferença nas consequências dos erros nestes diferentes *media* irá implicar que os mecanismos de controlo de erro para aplicações de vídeo sejam diferentes daqueles usados nos *media* discretos, como o texto.

Os mecanismos de controlo de erro para aplicações de vídeo são [Wu 01b]:

- **forward error correction (FEC):** o princípio do FEC é adicionar alguma informação redundante num fluxo de dados de forma a permitir que o vídeo original possa ser reconstruído no caso de surgir perda de pacotes;
- **reenvio:** a repetição da transmissão de pacotes não é, normalmente, uma técnica muito usada para os dados em tempo real. O tempo de reenvio de um pacote é, basicamente, um atraso grande na sua entrega. Mas se acontecer que o tempo de reenvio de um pacote seja inferior ao atraso máximo permitido, então o uso desta técnica será possível;
- **resiliência a erros:** lida com a perda de pacotes, mas na perspectiva da compressão. Contrariamente ao FEC que corrige erros a nível do transporte ou perdas de pacotes, este método tenta corrigir o vídeo de maneira que a qualidade da imagem não se degrade muito. Aborda principalmente as questões relacionadas com o evitar da propagação do erro e limitar o âmbito do erro;
- **encobrimento do erro:** uma última técnica, para quando existe a perda de pacotes, é reconstruir o vídeo recorrendo à interpolação espacial e temporal de forma a manter a apresentação agradável aos olhos humanos (que são tolerantes a algumas distorções do

vídeo).

2.4 Arquitectura de um serviço de *streaming*

Um sistema que implemente o serviço de *streaming* de vídeo engloba vários elementos, tendo cada qual diferentes funcionalidades e objectivos.

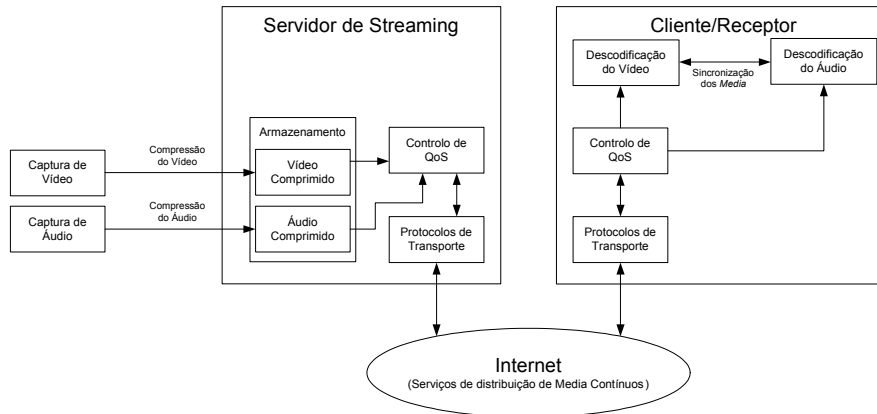


Figura 2-3 Arquitectura para o *streaming* de vídeo

A figura 2-3 mostra a arquitectura definida por [Wu 01a], enquadrando os seguintes elementos: compressão de vídeo, controlo da qualidade de serviço, serviços de distribuição de *media* contínuos, servidores de *streaming*, mecanismos de sincronização de *media*, e protocolos para o *streaming* de *media*.

Nas subsecções 2.4.1 , 2.4.2 , 2.4.3 e 2.4.4 , desta secção serão apresentados alguns dos elementos. Os restantes, como o controlo da QoS e protocolos de transporte, são tratados em secções próprias, respectivamente, na 2.3 e na 2.5 .

2.4.1 Compressão de vídeo

O vídeo digital sem qualquer tipo de compressão implica ficheiros de grande dimensão. Para ultrapassar este problema de armazenamento e, principalmente, a nível da transmissão é usada a compressão. A compressão é dividida em dois tipos: sem perdas e com perdas.

A compressão sem perdas permite que o ficheiro comprimido seja recuperado integralmente após a descompressão. A compressão sem perdas apresenta taxas de compressão pequenas e mantém, portanto, o problema dos ficheiros de vídeo serem ainda de dimensão considerável.

A compressão com perdas já não possibilita a recuperação fiel dos dados originais. Num ficheiro de texto ou num ficheiro executável este facto é grave, pois corta informação relevante e, assim, corrompe os dados. A compressão com perdas para os dados de *media* contínuos é feita para que estes se apresentem com ou sem perdas perceptíveis. As taxas de compressão com este método já são bastante maiores que a compressão sem perdas e estas taxas podem ser ajustadas de forma a obter um vídeo com a qualidade ou com a taxa pretendida.

Os esquemas de compressão de vídeo podem ser classificados em duas categorias: não

escaláveis e escaláveis. A compressão de vídeo não escalável cria, a partir do vídeo descomprimido, um ficheiro comprimido a uma dada taxa. A compressão de vídeo escalável produz um vídeo que pode ser transmitido a diferentes taxas. O princípio deste esquema está em criar um fluxo base que possa ser decodificado de forma independente e depois ir criando outros fluxos que vão trazendo melhorias ao fluxo base. Os fluxos não base não podem ser decodificados de forma independente, necessitando sempre de estarem acompanhados do fluxo de base. O fluxo base fornece o vídeo de pior qualidade e o uso adicional de todos os restantes fluxos permite ter o vídeo com a melhor qualidade.

A escalabilidade pode ser de qualidade (SNR - *Signal-to-Noise ratio*), temporal e espacial. Do ponto de vista temporal, a escalabilidade passa por alterar o ritmo dos *frames*, e do ponto de vista espacial passa por ajustar o tamanho da imagem. Além de mecanismos que combinam os tipos de escalabilidade referidos, existe outro, que surge com a norma MPEG-4, que é a escalabilidade de elevada granularidade (FGS de *fine granularity scalability*, onde o vídeo codificado consiste de uma camada de base e uma camada de melhoria, e onde esta última pode ser limitada de forma arbitrária de forma a encaixar o vídeo na largura de banda disponível [Koshy 02]). O FGS procura resolver os problemas de heterogeneidade do *streaming* por via de uma maior flexibilidade às diferenças encontradas nas redes e nos clientes.

2.4.2 Serviços de distribuição de *media* contínuos

Os serviços de distribuição de *media* contínuos são implementados por cima do protocolo IP e visam melhorar a eficiência do *streaming* de dados sobre a Internet. Estes serviços incluem a filtragem na rede, o *multicast* a nível da aplicação e a replicação de conteúdos [Wu 01a].

A filtragem na rede é uma técnica de controlo de congestão que procura maximizar a qualidade do vídeo durante um período de congestão na rede. Os filtros estão situados na rede e comunicam com o cliente e com o servidor. O filtro recebe os pedidos do cliente, que reencaminha para o servidor, e adapta o fluxo de dados gerado pelo servidor. A comunicação é feita por intermédio de dois canais virtuais: um para controlo, bidireccional; e um para os dados, unidireccional.

O *multicast* a nível da aplicação, conceptualmente semelhante ao *multicast* IP, visa a implementação de ligações de um para muitos, o que permite eficiência na utilização da largura de banda. A camada onde essa ligação se situa é a diferença entre os dois tipos de *multicast* referidos. O *multicast* a nível da aplicação permite aos vários intervenientes no fornecimento de conteúdos a construção de um serviço de *multicast* sobre a Internet e assim criar as suas próprias redes *multicast*, que podem ser ligadas a outras redes semelhantes.

A replicação de conteúdos, como o próprio nome indica, é a existência de cópias dos conteúdos, mais próximas dos clientes. A criação destas cópias visa: diminuir a necessidade de largura de banda na rede; reduzir a carga dos servidores de *streaming*; minimizar o tempo de início da apresentação para o cliente; e aumentar a disponibilidade. As formas de implementar a replicação de conteúdos passam pelo uso de *caches* e pela duplicação da informação.

Comment [D1]: Os intervenientes são CSPs (content Service Provider), ISPs (Internet Service Provider) e as empresas. Este método tem como a vantagem de quebrar as barreiras relacionadas com a escalabilidade, a gestão e o suporte ao controlo de congestão das redes, que existia no IP *multicast*.

2.4.3 Servidores de *streaming*

Os servidores de *streaming* são um componente muito importante. Para obter um serviço de qualidade os servidores têm de processar os pedidos e os fluxos de *media* contínuos com restrições apertadas de tempo, para evitar uma degradação da qualidade da visualização, e têm de suportar as operações de controlo interactivo, como pausa/avanço, e avanço e retrocesso rápido.

Um servidor de *streaming* é constituído pelos seguintes componentes: comunicador, sistema operativo e armazenamento. Estes dois últimos são tratados, respectivamente, nas secções 2.4.3.1 e 2.4.3.2. O comunicador envolve a camada de aplicação e os protocolos de transporte, e refere-se a todo o processo de comunicação entre o servidor e o cliente. Este componente é tratado nas secções 2.3 e 2.5.

2.4.3.1 Sistema operativo

O sistema operativo (SO) para um servidor de *streaming* é diferente de um SO comum, uma vez que necessita de responder a requisitos do tratamento de dados em tempo real. A gestão dos processos, dos recursos e dos ficheiros têm abordagens específicas.

A gestão dos processos refere-se ao uso do recurso principal, o processador. A gestão é o agendamento de cada processo criado, mediante um conjunto de políticas que têm por objectivo a utilização eficiente do servidor. São usadas técnicas de agendamento em tempo real e os algoritmos principais são o *earliest deadline first* e o *rate-monotonic scheduling*.

A gestão dos recursos envolve a definição do uso dos recursos do sistema. Esses recursos são o processador, a memória e os dispositivos de armazenamento. A gestão envolve o controlo de admissão e a atribuição de recursos. Em ambos os casos são usados algoritmos determinísticos e estatísticos para realizar a gestão.

A gestão dos ficheiros é o conjunto de funções para controlo e acesso à escrita e leitura dos ficheiros. As abordagens para otimizar a gestão dos ficheiros são duas. A primeira é guardar os ficheiros de *media* contínuos como se fossem dados discretos, mas recorrendo ao uso de algoritmos específicos para a sua gestão e pelo aumento na capacidade dos *buffers* para evitar atrasos. A segunda é organizar os ficheiros de dados contínuos em vectores de discos rígidos, isto é, dividir os ficheiros por diversos discos e assim diminuir os tempos de acesso (tema revisto na secção seguinte).

2.4.3.2 Armazenamento

Os requisitos de armazenamento e de envio de *media* contínuos também são diferentes de sistemas de armazenamento de dados comuns. Os problemas para o armazenamento são:

- **obter grandes taxas de transferência:** este problema pode ser minorado através da divisão do ficheiro por vários discos rígidos permitindo o acesso paralelo (diferente da replicação dos ficheiros, que é um método onde existem múltiplas cópias de um ficheiro e que permite, também, obter taxas de transferência maiores, mas com um aumento do custo);
- **capacidade do armazenamento:** que neste caso é um problema indirecto de custo, que

pode ser diminuído através do uso de memória terciária (como as *tapes*) e da hierarquização da informação, mantendo apenas nos discos os dados mais pedidos. Existem outras abordagens baseadas na distribuição dos dados como as arquitecturas *Storage Area Network* e *Network Attached Storage* [Wu 01a].

2.4.4 Mecanismos de sincronização de *media*

A sincronização é o mecanismo que visa manter as relações temporais dentro de um fluxo ou entre vários fluxos de dados. O envio de fluxos do servidor para o cliente tem ao longo do caminho muitos componentes (como pode ser verificado na figura 2-3). Cada um tem um certo objectivo e tem de realizar uma tarefa específica introduzindo, inevitavelmente, algum atraso. Alguns destes atrasos (como o introduzido pela rede) são aleatórios e inevitáveis e torna-se necessário recorrer a métodos de sincronização.

Existem três níveis de sincronização: *intra-stream*, *inter-stream* e *inter-object*.

A sincronização *intra-stream* refere-se à sincronização dentro de um fluxo de dados. Esta sincronização visa garantir uma continuidade temporal lógica na apresentação desse fluxo. A unidade considerada a este nível é o *frame* (de áudio ou vídeo).

A sincronização *inter-stream* refere-se à sincronização entre vários fluxos de dados, ou seja, à relação entre diferentes fluxos de dados. A este nível a unidade é o fluxo de dados, como um fluxo de vídeo ou um fluxo de áudio.

A sincronização *inter-object* integra fluxos de dados dependentes no tempo com dados independentes no tempo. A unidade neste nível é o objecto, que pode ser um vídeo, uma imagem, um texto, ou outro.

Os métodos de sincronização são especificações baseadas em: intervalos, gravação do tempo (*time-stamping*), controlo de fluxo e eventos [Wu 01a].

2.5 Protocolos

Os protocolos envolvidos no *streaming* de vídeo são métodos desenhados e normalizados para a comunicação entre os clientes e os servidores. Os serviços que os protocolos de *streaming* providenciam são endereçamento, transporte e controlo de sessão, o que permite classificar os protocolos em três categorias [Wu 01a]:

- **protocolo de rede:** é usado o protocolo *Internet Protocol* (IP), que fornece os serviços básicos de rede como o endereçamento;
- **protocolos de transporte:** fornecem funções de transporte de rede extremo a extremo para aplicações de *streaming*. Nesta camada os protocolos existentes são: o *User Datagram Protocol* (UDP) e o *Transmission Control Protocol* (TCP) num nível mais baixo desta camada; e o *Real Time Transport Protocol* (RTP) e *Real Time Transport Control Protocol* (RTCP), num nível mais alto desta camada;
- **protocolo para o controlo da sessão:** define as mensagens e os procedimentos para o

controlo da entrega de dados multimédia durante uma sessão. Um protocolo com essas características é o *Real Time Streaming Protocol* (RTSP).

A figura 2-4 apresenta a relação entre os diversos protocolos e várias áreas específicas. Os dados são colocados em pacotes e enviados pelo RTP. Estes pacotes levam informação de temporização, de sincronização e de sequência.

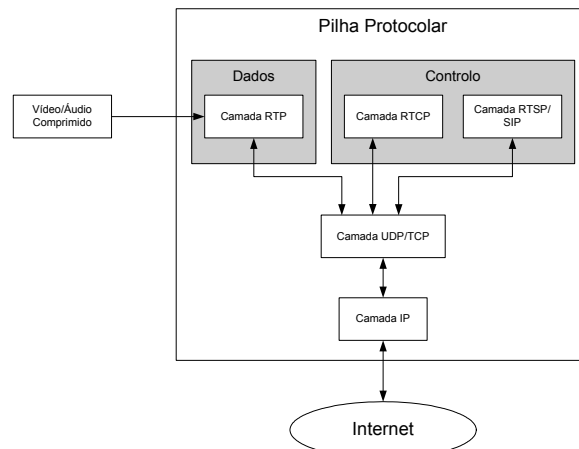


Figura 2-4 Pilha protocolar do *streaming*

Os pacotes são passados para a camada UDP/TCP e desta, por sua vez, para a camada IP. Após a transmissão sobre a Internet, no cliente o processo inverte-se com os mesmos elementos. Na área do controlo, os pacotes RTCP e RTSP são multiplexados na camada UDP/TCP, movidos para a camada IP e transmitidos sobre a Internet.

Os protocolos descritos de seguida são aqueles mais directamente relacionados com o transporte e controlo de dados em tempo real. As secções 2.5.1 , 2.5.2 e 2.5.3 , apresentam as principais características, respectivamente, dos protocolos RTP, RTCP e RTSP.

2.5.1 RTP

O RTP é um protocolo definido para o transporte de dados em tempo real, como o vídeo e o áudio. Os serviços fornecidos pelo RTP são a reconstrução temporal, detecção de perdas, segurança e identificação de conteúdo. Inicialmente o RTP foi desenhado para o *multicast* de dados em tempo real, mas também é possível a sua utilização no modo *unicast*. O uso deste protocolo, como foi referido, é essencialmente para o transporte de dados e, paralelamente, é usado outro protocolo para o controlo, o RTCP (descrito na secção 2.5.2).

2.5.1.1 Funcionamento

O RTP, como protocolo de transporte para dados em tempo real, necessita de eficiência e por isso usa essencialmente o UDP. O UDP é mais eficiente que o TCP, mas tem algumas desvantagens relevantes, como não garantir a entrega de todos os pacotes, nem a entrega dos pacotes na sequência correcta. Mas o RTP pode ser usado sobre o TCP, garantido assim a entrega de todos os pacotes, mas dificilmente mantendo uma apresentação sem quebras. A tolerância dos humanos na

visualização/audição de dados em tempo real é maior quando há perdas de qualidade, comparativamente a quando há atrasos ou interrupções.

O RTP não garante qualidade de serviço ou a entrega correcta dos pacotes mas fornece as seguintes funções de suporte sobre os pacotes: a marcação temporal, a sequência, a identificação do tipo de carga e a identificação da fonte. Estas funções são implementadas através de informação colocada no cabeçalho do pacote RTP.

A marcação temporal (expressão traduzida do termo inglês *timestamping*) é uma informação fundamental nos dados em tempo real. Todos os pacotes são marcados com o instante em que o primeiro octeto do pacote foi amostrado. Quando recebidos os pacotes, por intermédio da marcação feita no envio, é possível reconstruir correctamente, em termos temporais, o vídeo ou o áudio originais. Outra função importante da marcação temporal é possibilitar a sincronização entre diferentes fluxos de dados. Apesar do RTP providenciar estas funções, a reconstrução do fluxo e a sincronização é da responsabilidade da aplicação.

A numeração sequencial dos pacotes também é feita pelo RTP e permite a ordenação dos pacotes no cliente. Esta funcionalidade é necessária devido ao facto do UDP não garantir a chegada dos pacotes na sequência em que foram gerados. Como cada pacote poderá tomar caminhos diferentes a chegada ao destino poderá ocorrer numa ordem distinta ao do envio. A ordenação dos pacotes poderia ser executada através da marcação temporal, mas existem casos em que tal não é suficiente, como por exemplo, um *frame* de vídeo que necessite de mais que um pacote para ser transmitido (terá nos diversos pacotes a mesma marca temporal), logo, é necessário um método adicional para ordenar os pacotes, que é a numeração. A numeração também permite a detecção dos pacotes perdidos.

Outra funcionalidade do RTP é identificar o tipo de carga (tradução do termo inglês *payload*). Esta funcionalidade serve para identificar, relativamente aos dados transmitidos, o formato e os métodos que foram usados para a sua codificação e compressão. Através desta informação a aplicação cliente sabe como interpretar e apresentar os dados recebidos. Na norma estão já definidos alguns tipos de carga como vídeo e áudio MPEG-1/2, PCM e H.261, entre outros.

A identificação da fonte permite que a aplicação conheça a origem dos dados que está a receber e possa identificar diferentes fontes. A identificação é feita por intermédio de um campo no cabeçalho do pacote que se designa *Synchronization Source Identifier*.

2.5.2 RTCP

O RTCP é o protocolo de controlo desenhado para funcionar em conjunção com o RTP. Numa sessão RTP, os participantes enviam periodicamente pacotes a informar a qualidade dos dados entregues e informação sobre a associação à sessão [Liu 99]. Os tipos de pacotes definidos pelo RFC 1889 para o RTPC são os seguintes: *receiver report*, *sender report*, *source description items*, *bye*, e *application specific functions*.

Os *receiver reports* são gerados pelos participantes que não são emissores activos numa sessão. Contêm informação sobre a qualidade da recepção dos dados, nomeadamente, o pacote com o número mais elevado recebido, o número de pacotes perdidos, a diferença entre a chegada

de pacotes, e as marcações temporais (para cálculo do atraso entre o emissor e receptor).

Os *sender reports* são gerados por emissores activos. Além da mesma informação que os *receiver reports*, estes contêm uma secção onde existe informação sobre o emissor, nomeadamente: dados para a sincronização *inter-media*, contadores de pacotes cumulativos e quantidade de *bytes* enviados.

Os *source description items* contêm informação que descreve as fontes. O *bye* indica o fim da participação. As *application specific functions* servem para fins experimentais, com objectivo de apoiar o desenvolvimento de novas aplicações e funcionalidades.

O RTCP, através destes pacotes de controlo de informação, providencia os seguintes serviços:

- **monitorização da QoS e controlo de congestão:** funções principais do RTCP que fornecem retorno de informação a uma aplicação sobre a qualidade da distribuição dos dados. Esta informação é útil para emissores, receptores e outras entidades com funções de monitorização. O emissor pode ajustar a transmissão baseada no relatório do receptor. Os receptores podem determinar se uma certa congestão é local, regional ou global. Os gestores de rede podem avaliar o desempenho da rede em distribuição *multicast*;
- **identificação da fonte:** os pacotes de dados do RTP são identificados por números de 32 *bits* gerados aleatoriamente, forma de identificação que não é adequada para os humanos. Os pacotes RTCP *source description items* contêm informação textual, designada nomes canónicos. Servem como identificadores globais entre os participantes da sessão. Os dados que podem existir são nome, telefone, *e-mail* e outra informação;
- **sincronização *inter-media*:** os *sender reports* do RTCP contêm indicações do tempo real e da marcação temporal RTP correspondente. É usada para a sincronização entre vários *media*, como por exemplo o áudio e vídeo (sincronização do movimento dos lábios com o som);
- **informação de controlo escalonável:** os pacotes RTCP são enviados periodicamente entre participantes. Com o aumento do número de participantes, é necessário existir um equilíbrio entre ter informação de controlo actualizada e limitar o tráfego de dados de controlo. De forma a permitir um escalonamento para grandes grupos de *multicast*, o RTCP tem de evitar que o tráfego de controlo ocupe muitos recursos de rede. O RTP limita o tráfego de controlo para um máximo de 5% de todo o tráfego da sessão. Este ajuste é realizado alterando a taxa de geração de relatório mediante o número de participantes.

2.5.3 RTSP

O RTSP é um protocolo para o controlo de sessão de *streaming* sobre a Internet. Fornece funcionalidades do tipo “vídeo doméstico” (avanço, pausa, recuo e avanço rápido e posicionamento absoluto) para fluxos de áudio e vídeo. Os dados de *streaming* passíveis de envio

podem ser uma emissão “ao vivo” ou um ficheiro armazenado.

É um protocolo da camada da aplicação e foi desenhado para operar com protocolos de camadas mais baixas como o RTP e RSVP (ver secção 2.6.1.2) de forma a obter um serviço de *streaming* integrado sobre a Internet. Permite a escolha do canal de envio a utilizar (UDP, *multicast* UDP e TCP) e os mecanismos de entrega baseados no RTP. Funciona tanto em *multicast* como em *unicast*.

O RTSP estabelece e controla fluxos de áudio e vídeo entre os servidores e os clientes. Um servidor permite serviços de visualização ou gravação de fluxos de *media* enquanto os clientes fazem pedidos de dados ao servidor. As operações do RTSP são as seguintes:

- **obtenção de dados do servidor:** o cliente pode obter a descrição de uma apresentação e pedir ao servidor para iniciar uma sessão para o envio dos dados;
- **adicionar *media* a uma apresentação que já existe:** o servidor ou o cliente podem-se notificar sobre um qualquer *media* adicional que fique disponível.

No RTSP, as apresentações e os *media* são identificados por um URL. A apresentação e as propriedades do *media* são definidas num ficheiro de descrição da apresentação, que pode incluir codificação, linguagem, RTSP URL, endereço de destino, porta e outros parâmetros. Este ficheiro pode ser obtido por http, *e-mail*, ou de outras formas.

O RTSP mantém sessões para a troca de dados multimédia e o seu estado, permitindo ainda correlacionar os pedidos RTSP de uma sessão. No RTSP a emissão de pedidos pode tanto surgir por parte do cliente, como por parte do servidor.

2.6 QoS centrada na rede

A abordagem da QoS centrada nos sistemas cliente/servidor foi explorada na secção 2.3.2 . A secção actual apresenta dois métodos para a implementação da QoS, mas baseado nas redes.

A Internet e o protocolo IP suportam uma variedade cada vez maior de aplicações. Além das aplicações tradicionais, como o *e-mail* e a transferência de ficheiros, temos hoje também as aplicações com dados em tempo real, cujos requisitos sobre a rede são muito maiores e variados. Como nem a Internet nem o IP foram, inicialmente, projectados para lidar com o tráfego gerado por estas aplicações, desenvolveram-se várias técnicas para envolver a rede e para melhorar os protocolos de maneira a cumprir com as novas necessidades.

Esta secção aborda os principais conceitos de duas arquitecturas que tentam resolver a falta de garantias da actual Internet a nível da QoS. Esses métodos são os serviços integrados, baseado na reserva de recursos (secção 2.6.1), e os serviços diferenciados, baseado na marcação dos pacotes de acordo com classes de serviços predeterminados (secção 2.6.2) [Santos 99].

2.6.1 Serviços integrados

Os serviços integrados caracterizam-se por fazerem a reserva de recursos na rede e visam estabelecer uma QoS extremo a extremo. Antes de se iniciar a troca de dados, o emissor e receptor têm de fazer a alocação de recursos na rede que os liga. Os recursos reservados são a largura de

banda e tempo que deverá ser mantida a ligação.

Os serviços integrados estão a ser desenvolvidos por um grupo de trabalho do IETF, o INTSERV (serviços integrados), e neste foram identificados quatro componentes chave para uma arquitectura de serviços integrados para a Internet [Osso 99]: nível de serviço, *interface* de serviço, controlo de admissão e mecanismos de agendamento dentro da rede.

2.6.1.1 Nível de serviço

A definição passa por incluir toda a semântica do serviço: como é que os pacotes devem ser tratados dentro da rede, a forma da aplicação injectar tráfego na rede e o método de policiamento do serviço. Esta semântica terá de ser conhecida pelos *routers* e aplicações.

Foram definidos dois níveis de serviços: carga controlada e garantido.

O nível **carga controlada** é, em termos de qualidade de serviço, semelhante à actual Internet, ou seja, uma rede *best-effort*.

O nível **garantido** visa possibilitar a entrega de um fluxo de dados com garantias no ritmo de transmissão e no atraso.

2.6.1.2 Interface do serviço

A interface de serviço são um conjunto de parâmetros trocados entre a aplicação e a rede de forma a invocar uma dada qualidade de serviço dentro de um determinado nível de serviço, ou seja, é necessário um protocolo de sinalização e a definição de alguns parâmetros.

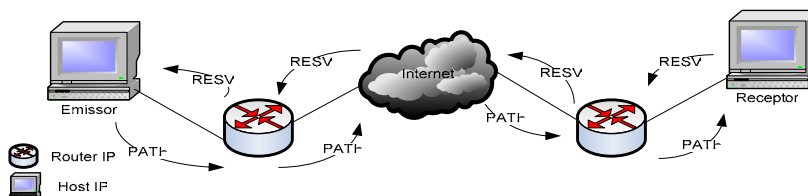


Figura 2-5 Sinalização feita pelo RSVP

O protocolo usado é o *ReSource reserVation Protocol* (RSVP). O pedido de reserva consiste num *FlowSpec*, que identifica as características do tráfego e o nível de serviço necessário. O procedimento de reserva (figura 2-5) é o seguinte: o emissor envia uma mensagem PATH anunciando os requisitos de QoS da sessão para o receptor.

Todos os *routers* RSVP que reencaminham esta mensagem guardam informação sobre o pedido (até um dos seguintes casos: cancelamento por parte do emissor, confirmação por parte do receptor ou *time-out*). O receptor então responde à mensagem PATH com uma mensagem RESV, enviada pelo mesmo caminho que a mensagem PATH tomou e, então, o emissor pode começar a enviar os dados. Para manter a reserva dos recursos são enviadas, regularmente, mensagens PATH e RESV por parte do emissor e receptor, respectivamente.

O RSVP faz a sinalização entre as aplicações do emissor e do receptor, cria reservas unidireccionais, é específico de uma sessão e necessita que as aplicações e a rede suportem os INTSERV e o RSVP.

2.6.1.3 Controlo de admissão

Serve para definir se um dado compromisso de serviço pode ser ou não satisfeito antes de se iniciar o envio dos dados. Usa a informação transmitida na invocação do serviço, juntamente com os dados de outras reservas activas. É também responsável por verificar o uso do serviço, garantindo que as aplicações não ultrapassam os recursos que foram pedidos.

O controlo de admissão é implementado nos *routers*.

2.6.1.4 Mecanismos de agendamento dentro da rede

A rede, após a aceitação das reservas, tem de empregar métodos que garantam que os pacotes de um certo fluxo recebam o nível de serviço que foi requerido para esse fluxo. Para isso, usa mecanismos de agendamento e gestão de filas nos *routers*.

O processo de encaminhamento dos pacotes feito pelos *routers* é complexo e envolve para cada pacote os seguintes passos:

- classificar o pacote de forma a identificar os requisitos de QoS;
- determinar o momento em que o pacote deve ser encaminhado;
- gerir as filas de saída sobre condições de congestão.

Estes passos exigem aos *routers* grandes capacidades de processamento, armazenamento e bons algoritmos para gestão das filas.

2.6.1.5 Limitações

A tarefa que os serviços integrados tentam resolver é complexa e por isso a sua implementação não é simples, havendo algumas limitações na sua utilização.

O protocolo RSVP tem alguns problemas, tais como [Osso 99]:

- a primeira passagem do estabelecimento de duas reservas pode encontrar um elemento de rede que aceita apenas uma delas, rejeitando a outra, mesmo que eventualmente a primeira venha a ser cancelada;
- no caso de uma ligação *multicast* é necessário manter uma grande quantidade de informação de estado;
- a manutenção das reservas leva à geração, pelos *routers* e aplicações, de tráfego extra;
- não é suportada uma heterogeneidade completa, por exemplo, numa conferência todos os receptores têm de partilhar o mesmo nível de serviço (*guaranteed* ou *controlled-load*);
- as aplicações têm de suportar o RSVP, o que não é simples de fazer em aplicações desenvolvidas com tecnologias antigas.

Estes problemas levam a que o INTSERV não seja muito apropriado para redes muito grandes, devido às elevadas capacidades necessárias de processamento e armazenamento nos *routers*. E como todos os *routers* participantes numa reserva têm de suportar o INTSERV levanta-se um problema de custos.

Mas os serviços integrados não deixam de ser uma opção interessante para as redes de

pequena dimensão.

2.6.2 Serviços diferenciados

Os serviços diferenciados, DIFFSERV, também desenvolvidos por um grupo de trabalho do IETF, abordam de forma diferente a utilização dos recursos de rede, comparativamente aos serviços integrados.

O modelo geral dos DIFFSERV passa por definir um sistema onde o tráfego que entra numa rede é classificado, condicionado, possivelmente, aos limites da rede, e agregado a diferentes conjuntos de comportamentos [Baker 98]. A classificação passa pela marcação dos pacotes com etiquetas bem definidas que identificam o nível de serviço onde será enquadrado o pacote. Este modelo é análogo ao utilizado nos correios tradicionais: as cartas/encomendas são classificadas como, por exemplo, registadas, correio azul e correio normal.

A grande diferença dos DIFFSERV relativamente aos INTSERV está na sua operação que é dirigida para um modelo de negócio, baseada em limites definidos de forma administrativa, e com os serviços agregados em utilizadores e grupos de utilizadores. Enquanto o RSVP actua numa base de fluxos individuais, as classes dos DIFFSERV podem ser usadas por muitos fluxos. Qualquer pacote dentro de uma classe partilha os recursos com todos os pacotes dessa mesma classe [Osso 99].

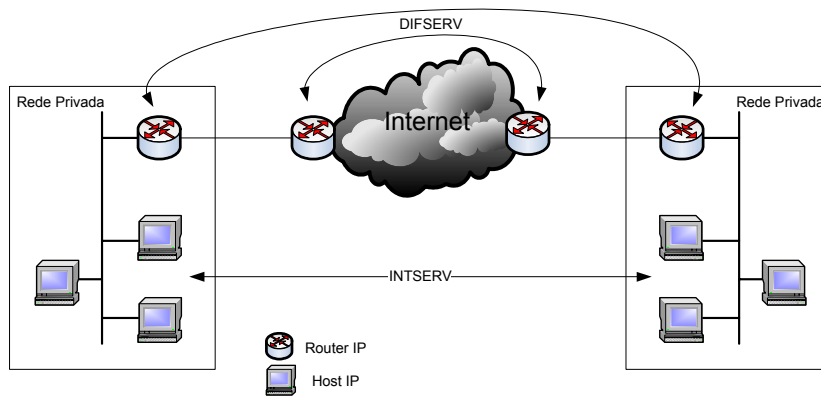


Figura 2-6 Âmbito dos serviços integrados e serviços diferenciados

A arquitectura dos DIFFSERV define que o serviço é implementado por intermédio de acordos entre domínios adjacentes (por exemplo, entre o utilizador do serviço e o fornecedor de serviço, ou entre fornecedores de serviço), designados por Acordos do Nível de Serviço (SLA de *Service Level Agreement*). O SLA determina as classes de serviços que serão suportadas e a largura de banda disponível entre os domínios. O acordo ainda pode incluir outros aspectos além dos relacionados com a QoS, como segurança, restrições de utilização em algumas horas do dia, etc.

A figura 2-6 mostra a diferença no âmbito destes serviços. Os INTSERV tentam fornecer a reserva de recursos às aplicações, ou seja, extremo a extremo. Os DIFFSERV tentam fornecer serviços baseados num contrato, SLA, entre as redes de serviços. Uma das grandes vantagens dos DIFFSERV é permitir a sua utilização sem serem necessárias alterações a nível das aplicações. Mas é sempre necessário dotar os *routers* da capacidade de suporte dos DIFFSERV. A marcação pode ser

feita nos *routers* da rede do utilizador do serviço ou nos *routers* do fornecedor de serviços. No primeiro caso, o fornecedor de serviços terá de proceder à verificação das marcações feitas nos pacotes para comprovar o respeito do SLA por parte do utilizador.

A implementação dos DIFFSERV nos pacotes passa pela utilização do campo *Type of Service* ou *Traffic-class* do cabeçalho de um qualquer pacote Ipv4 ou Ipv6, respectivamente. É feita uma alteração na sua composição, tendo uma nova designação: *Differentiated Service Field* (DS Field), mas mantendo a mesma semântica nas duas versões do IP. É neste novo campo que serão inscritas as classes definidas para os DIFFSERV.

Apesar das diferenças entre os DIFFSERV e os INTSERV, estes não são mutuamente exclusivos. Apesar de existirem acordos SLA genéricos entre domínios, se necessário, e para responder a um pedido pontual de uma certa aplicação (como uma videoconferência), podem ser usados mecanismos de reserva de recurso, recorrendo por exemplo o RSVP, para um certo fluxo.

2.7 Conclusão

O uso de vídeo sobre a Internet é uma tarefa que depende da conjugação de muitos factores, onde todos influenciam na qualidade final da visualização. A manutenção de um fluxo constante de dados é impossível de obter devido às características da Internet, logo é necessário recorrer a um conjunto de métodos e técnicas para ultrapassar esta limitação.

A distribuição por intermédio do *streaming* traz uma grande flexibilidade em termos de serviços possíveis e nas formas de distribuição. A possibilidade da visualização de eventos ao vivo, e por um grande número de utilizadores, são das maiores funcionalidades apenas possíveis com *streaming*.

Através de métodos de controlo de congestão e de erros é possível manter a qualidade a níveis aceitáveis e resolver, em parte, as limitações da Internet, como a pouca largura de banda, e a imprevisibilidade do atraso e das perdas. Outras dificuldades prendem-se, primeiro, com a heterogeneidade existente nas redes e nos sistemas e, segundo, com a protecção dos direitos dos autores dos conteúdos.

A criação de aplicações sobre a Internet com *media* contínuos necessita de um conhecimento alargado de todos os componentes envolvidos num serviço de *streaming*. Entre os mais importantes surge a compressão do vídeo, entre métodos escaláveis e não escaláveis, que permitem, respectivamente, uma maior ou menor flexibilidade na adaptação às flutuações nas características das redes. Os serviços de distribuição de *media* contínuos definem um conjunto de métodos, implementados, essencialmente, fora dos servidores e dos clientes, que permitem melhorar a qualidade do serviço de *streaming*. No servidor de *streaming*, elemento fulcral no serviço de *streaming*, podem ser implementadas diferentes políticas de gestão e afectação de recursos, procurando melhorar a eficiência no envio dos dados para os clientes. Por fim surgem ainda os mecanismos de sincronização, com diferentes níveis, fundamentais para uma apresentação lógica e coerente dos dados.

Os protocolos para o *streaming* são formas normalizadas de tratar o envio de dados que têm

características particulares. Estes dados, designados como dados em tempo real, têm condicionantes temporais e de qualidade diferentes dos restantes dados que circulam na Internet. Os protocolos referidos permitem a comunicação entre servidores e clientes, bidireccional, dos dados e de informação de controlo.

A abordagem de QoS sobre a rede é um elemento preponderante para a utilização futura da multimédia sobre a Internet. As duas arquitecturas descritas apresentam diferentes perspectivas sobre o uso dos recursos de rede, e cada uma apresenta-se com alternativa adequada para um dado caso.

Apenas com o conhecimento de todos estes aspectos e a sua correcta conjugação é possível obter uma aplicação onde a apresentação de *media* contínuos seja conseguida com qualidade e de uma forma apelativa. De entre os diversos *media*, o vídeo deverá merecer um cuidado especial por ser mais exigente em termos de recursos de rede.

3. Composição, sincronização e interacção multimédia

Neste capítulo serão descritas duas normas não proprietárias para a composição, sincronização e interacção em cenas multimédia.

A primeira norma será a segunda versão da Linguagem de Integração de Multimédia Sincronizada (SMIL) do *World Wide Web Consortium* (W3C). A exposição do SMIL passa pela apresentação dos seus grupos funcionais com os principais módulos e é complementada com alguns exemplos dos módulos mais utilizados.

A segunda norma a explorar será a MPEG-4, do *Motion Picture Experts Group*. Será feita uma pequena introdução à tecnologia num âmbito geral e será especificada a forma como estão definidas a composição, a sincronização e a interacção.

3.1 Introdução

As normas para composição, sincronização e interacção em cenas multimédia existem há alguns anos, mas apesar desse facto a sua utilização tem sido reduzida. As normas que vão ser apresentadas neste capítulo foram escolhidas devido à importância que têm adquirido e à sua crescente aceitação.

O SMIL 2.0 (*Synchronized Multimedia Integration Language 2.0*) aborda todos os aspectos referidos no título deste capítulo, e ainda mais alguns, e é uma linguagem declarativa desenhada para a construção de apresentações multimédia.

O MPEG-4 além de abordar os aspectos do título, é muito mais abrangente dentro da área multimédia, visto que é uma família de protocolos e normas que cobre muitos dos aspectos relacionados com objectos multimédia.

3.2 Linguagem de integração de multimédia sincronizada

A Linguagem de Integração de Multimédia Sincronizada (SMIL) é um formato para codificação de apresentações multimédia sobre a Internet. Esta norma permite compor, sincronizar e interagir com uma cena multimédia, ou seja, é um método de coordenar e controlar objectos multimédia no tempo e no espaço, por intermédio de elementos e atributos XML. O autor define cada objecto, a ser integrado durante uma apresentação, de forma independente.

A primeira versão do SMIL surgiu em Junho de 1998 e foi adoptada pelos principais *media players* e navegadores então existentes. Em Junho de 2001 o W3C produz a versão 2.0, com melhorias significativas. Os objectivos para esta nova versão foram os seguintes: [Bulterman 01]

- extensão às funcionalidades do SMIL 1.0. Esta versão foi propositadamente construída como uma linguagem simples e de fácil implementação. O SMIL 2.0 adiciona melhorias na interacção, semântica da temporização, controlo de conteúdo e novas primitivas de animação e transição;
- manter um formato declarativo, XML. Apesar de principalmente integrar conteúdo multimédia, o SMIL 2.0 é completamente declarativo e não procedimental, já que o SMIL não define pormenores de implementação da apresentação, deixando essa tarefa para o *player*. Mantém a completa compatibilidade com a linguagem XML;
- introdução de uma estrutura baseada em módulos. Utilizando uma estrutura modular pode-se integrar partes do SMIL 2.0 em outras linguagens baseadas em XML, sem ser necessário o suporte de toda a especificação SMIL 2.0.

3.2.1 Estrutura básica de um documento SMIL

O SMIL é baseado em XML e como tal contém um elemento raiz que se designa *smil*. Este é

composto por duas partes: o cabeçalho (elemento *head*) e corpo (elemento *body*). No cabeçalho estará toda a informação não temporal, principalmente os dados que definem as regiões onde os diversos objectos serão posicionados. No corpo estarão os objectos com os seus atributos, a sua organização temporal e, se as houver, as ligações.

No anexo A encontra-se um documento SMIL para a implementação de um ecrã onde poderá ser visualizado um vídeo paralelamente com uma imagem e algum texto. São desse anexo alguns dos excertos de código de SMIL que servem de exemplo nesta secção.

3.2.1.1 Cabeçalho

O cabeçalho poderá conter três tipos de informação: metadados (elemento *meta*), composição da cena (elemento *layout*) e controlo de conteúdo. O código seguinte exemplifica a estrutura de um cabeçalho com informação guardada em metadados e define uma cena com três regiões:

```
<head>
  <meta name="title"
        content="Apresentação Grupo 01"/>
  <meta name="author"
        content="Alunos da LEI/UMA"/>
  <layout>
    <root-layout width="800"
                  height="600"
                  background-color="#CCCCCC"/>
    <region id="region apresentacao"
            width="500"
            height="375"
            fit="meet"
            top="15"
            right="15"/>
    <region id="region video"
            top="15"
            width="240"
            height="180"
            left="15"/>
    <region id="region notas"
            height="360"
            width="240"
            left="15"
            top="220"/>
  </layout>
</head>
```

3.2.1.2 Corpo

O corpo contém essencialmente a informação sobre a posição e temporização dos objectos. A organização temporal do SMIL é uma estrutura hierárquica em árvore, onde existe a sincronização paralela e sequencial de objectos. Os operadores principais para implementar estas duas formas de sincronização são, respectivamente, *par* e *seq*.

O código seguinte apresenta um exemplo de um ecrã onde é apresentado um vídeo e paralelamente se vão seguindo imagens e textos:

```
<body>
  <par>
    <video
      region="region_video"
```

```

        fit="fill"
        src="filme 01 300k.rm"/>
    <seq>
    <par>
    
    <text
        region="region_notas"
        src="apresentacao 01/Diapositivo1.txt"
        end="6s"
        begin="0s"/>
    </par>
    <par>
    
    <text
        region="region_notas"
        src="apresentacao 01/Diapositivo2.txt"
        begin="0s"
        end="46s"/>
    </par>
    .....
    </seq>
</par>
</body>

```

O código anterior mostra, além de exemplos de utilização dos operadores, a relação com as definições feitas no cabeçalho, com o atributo *region*. Cada objecto é posicionado numa região referida explicitamente.

A figura 3-1 é a representação gráfica do código anterior. Temos o vídeo, que existe durante toda a apresentação, e paralelamente as imagens e os textos, que são trocados em momentos específicos.

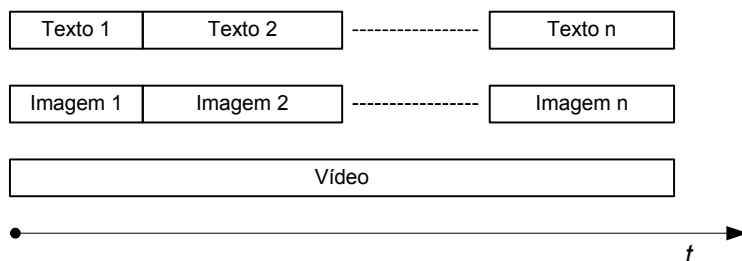


Figura 3-1 Exemplo de temporização

3.2.2 Grupos Funcionais

A segunda versão do SMIL define dez grupos funcionais de elementos e atributos, contendo aproximadamente cinquenta módulos. O conceito de módulo surge como a agregação de elementos e atributos XML semanticamente relacionados, representando uma unidade de

funcionalidades (por exemplo, o módulo *BasicAnimation*, do grupo funcional Animação). A junção de várias unidades de funcionalidades relacionadas constitui um grupo funcional (os módulos *BasicAnimation* e *SplineAnimation* formam o grupo funcional Animação). Quando os módulos são combinados num dado conjunto (não é obrigatória a combinação de grupos funcionais completos) designam-se como um perfil de implementação, o qual tem o objectivo de fornecer um conjunto de funcionalidades para uma aplicação específica. Na secção 3.2.3 são apresentados alguns dos perfis de implementação.

O resto desta secção terá uma pequena descrição de cada uma destas unidades e apresentará para as mais importantes alguns exemplos da utilização de alguns dos seus operadores.

3.2.2.1 Animação

A animação é uma função no tempo sobre um determinado elemento, mais precisamente sobre um dos seus atributos (como por exemplo a dimensão ou o posicionamento, entre outros). A animação define um mapeamento de valores no tempo do atributo.

O grupo Animação contém dois módulos, o *BasicAnimation* e o *SplineAnimation*. Estes módulos contêm elementos para introduzir animação sobre uma linha de tempo, bem como os mecanismos de composição entre diversas animações.

O módulo *BasicAnimation* fornece: a sintaxe dos elementos e dos atributos das funções de animação simples definidas por um conjunto de pontos a ser tratado sequencialmente. As funções definidas sobre a passagem por entre os diversos pontos usam interpolação discreta, linear ou *paced*. Os elementos existentes são os seguintes:

- *animate*: para animação genérica;
- *set*: para definir o valor de atributo;
- *animateMotion*: para definir o movimento num dado caminho;
- *animateColor*: para definir a alteração da cor no tempo.

O código seguinte apresenta um exemplo da utilização do *animate*. O elemento *rect* identifica um rectângulo e a animação definida fará a alteração da sua largura e comprimento.

```
<rect ...>
  <animate attributeName="width" from="10px" to="100px" begin="0s" dur="10s" />
  <animate attributeName="height" from="100px" to="10px" begin="0s" dur="10s" />
</rect>
```

O módulo *SplineAnimation* estende os modos de cálculo (discreto, linear e *paced*), permitindo um controlo adicional sobre a interpolação e a temporização. O código seguinte apresenta um exemplo onde é feita a alteração da cor de um objecto, iniciando a verde, passando a meio da animação para amarelo (0.5 visto que *keyTimes* define a relação temporal entre 0 e 1) e por fim para vermelho, até ao final da apresentação do objecto:

```
<animateColor attributeName="color" dur="10s"
  calcMode="discrete" values="green; yellow; red"
  keyTimes="0.0; 0.5; 1.0" />
```

3.2.2.2 Controlo de conteúdo

Os módulos do Controlo de Conteúdo contêm elementos e atributos que permitem variar o conteúdo a apresentar mediante as preferências do utilizador e otimizar a entrega do conteúdo. Os módulos são os seguintes:

- *BasicContentControl*: contém os elementos para a selecção de conteúdos e testes predefinidos aos atributos de sistema;
- *CustomTestAttributes*: contém elementos de teste e atributos definidos pelo autor do documento;
- *PrefetchControl*: contém os elementos e atributos para a optimização da apresentação;
- *SkipContentControl*: contém os atributos que suportam a avaliação selectiva de outros atributos.

O código seguinte apresenta um exemplo da selecção de conteúdo mediante a avaliação de um dado atributo (neste caso o *systemBitrate*) de uma lista dentro do elemento *switch*. Se este for igual ou superior a 40000 será apresentado o conteúdo dentro do elemento *par* onde é verificada essa condição (para o código apresentado é o primeiro caso) os restantes serão ignorados:

```
...
<par>
  <text .../>
  <switch>
    <par systemBitrate="40000">
      ...
    </par>
    <par systemBitrate="24000">
      ...
    </par>
    <par systemBitrate="10000">
      ...
    </par>
  </switch>
</par>
...
```

Este exemplo mostra a avaliação da largura de banda disponível detectada, mas existem outras possibilidades de teste como o tamanho do ecrã, a sua profundidade, a linguagem do sistema, a existência de legendas, entre outros. Estes testes já existiam no SMIL 1.0. Com o SMIL 2.0 também é possível avaliar o sistema operativo, o processador, incluir ou não descrições do áudio e outros componentes do sistema tem (como *codecs*, canais de áudio, etc.).

3.2.2.3 Composição

A Composição é uma colecção de módulos usados para explicitamente gerir a construção do espaço da apresentação e dos recursos audiovisuais. É composto por um módulo principal, o *BasicLayout*, e por outros três módulos construídos sobre este, adicionando cada qual diferentes funcionalidades.

O elemento raiz da Composição é o *layout* e cada zona desse ecrã é definida através do elemento *region*. Mediante o módulo usado poderão ainda ser usados outros elementos.

Os três módulos adicionais contêm elementos e atributos que permitem o posicionamento de objectos e o controlo da intensidade do áudio, e são os seguintes:

- *MultiWindowsLayout*: o uso deste módulo permite a criação e controlo de múltiplas janelas (nova funcionalidade com o SMIL 2.0). O elemento usado para definir cada uma das janelas é o *topLayout* dentro do elemento *Layout*;
- *HierarchicalLayout*: permite a declaração de regiões dentro de outras regiões. É semelhante ao conceito já usado entre *topLayout* e *region*, mas neste caso apenas com o elemento *region*;
- *AudioLayout*: suporta o controlo da intensidade do áudio de objectos posicionados numa dada região através da propriedade *soundLevel* do elemento *region*.

Na secção 3.2.1 é apresentado um exemplo da composição de um documento SMIL usando apenas o módulo *BasicLayout*.

3.2.2.4 Ligação

Os módulos da unidade de Ligação definem os elementos e atributos que permitem gerir as ligações temporais e não temporais. As ligações de uma apresentação SMIL podem ser desencadeadas pelo utilizador ou por eventos temporais.

Os módulos do grupo funcional Ligação são os seguintes:

- *BasicLinking*: é o conjunto de elementos básicos que definem as ligações;
- *LinkingAttributes*: é o conjunto de atributos usados para definir a semântica das ligações SMIL feitas pelos elementos básicos do módulo *BasicLinking*;
- *ObjectLinking*: inclui opções adicionais para funcionalidades de ligação que um dado perfil de linguagem pretenda incluir.

O código seguinte apresenta um exemplo da implementação de uma ligação dentro de uma apresentação SMIL. Se o utilizador seleccionar a imagem (source.jpg) localizado na metade superior da janela, desencadeará a apresentação do documento SMIL (embeddedSIML.smil) na parte inferior da janela. Uma segunda forma de iniciar esta acção é pressionando a tecla 'a'.

```
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
  <head>
    <layout>
      <region id="source"      height="%50"/>
      <region id="destination" top  ="%50"/>
    </layout>
  </head>
  <body>
    <a href="embeddedSMIL.smil" target="destination" accesskey="a">
      
    </a>
  </body>
</smil>
```

3.2.2.5 Objectos

O grupo funcional *Objectos* apresenta os módulos relacionados com a descrição dos *objectos media* que estarão presentes num documento SMIL. O módulo principal designa-se *BasicMedia* e existem cinco módulos adicionais cujas funcionalidades são complementares o módulo principal.

O módulo *BasicMedia* define os seguintes elementos para referir os mais diversos *objectos*:

- *ref*: referência genérica a um *media*;
- *animation*: gráficos vectoriais animados ou outro formato de animação;
- *audio*: ficheiro de áudio;
- *img*: ficheiro de imagem;
- *text*: referência a um texto simples;
- *textstream*: texto por *streaming*;
- *video*: ficheiro de vídeo;

A secção 3.2.1 apresenta um exemplo da utilização de vários elementos do módulo principal. Os cinco módulos já referidos no início desta secção são os seguintes:

- *MediaParam*: usa o elemento *param* e especifica o conjunto de valores que podem ser necessários a um *objecto* na sua apresentação. Podem existir vários elementos *param* para cada *objecto* e têm de estar no início do elemento deste;
- *MediaClipping*: contém dois atributos, *clipBegin* e *clipEnd*, que permitem a definição da apresentação de partes de um *media* contínuo (por exemplo: uma cena em vez de um filme completo);
- *MediaClipMarkers*: com *objectivos* similares aos anteriores mas utilizando marcadores para os momentos dos *media* contínuos, em vez do tempo como em *clipBegin* e *clipEnd*;
- *MediaAccessibility*: contém atributos (*alt* e *longdesc*) para informação suplementar que pode ser usada em casos de impossibilidade na apresentação do *objecto*. Define ainda um atributo (*readIndex*) para a ordenação do aparecimento da informação presente nos atributos *alt* e *longdesc*;
- *MediaDescription*: conjunto de elementos e atributos usados para descrever o *objecto*, como por exemplo: *abstract*, *title* e *copyright*.

O exemplo apresentado em 3.2.1 apenas faz uso dos elementos e dos atributos essenciais. Um exemplo com elementos mais diversos está apresentado no código seguinte:

```
<par>
  <video      id="carvideo"  src="car.rm"
             title="Car video"
             alt="Illustration of relativistic time dilation."
             longdesc="carvideodesc.html"
             readIndex="3"/>

  <audio      id="caraudio"  src="caraudio.rm"
             title="Car presentation voiceover"
             begin="bar.begin"/>
```

```

<animation id="cardiagram" src="car.svg"
  title="Diagram of the car"
  readIndex="2"/>


</par>

```

Neste exemplo, um equipamento especial faria apenas a apresentação dos títulos na ordem definida pelo atributo *readIndex*. O primeiro a ser apresentado seria a imagem, de seguida a animação e vídeo, e por fim o áudio por não ter o valor definido explicitamente.

3.2.2.6 Meta-informação

O módulo de Meta-Informação contém os elementos e atributos que permitem a descrição dos documentos SMIL.

Como existe uma dificuldade em automatizar de alguma forma a informação que existe na *web*, e se a este facto adicionarmos o volume de informação crescente, é importante que os conteúdos criados sejam acompanhados com informação que os descreva e assim permitirem acções de pesquisa, facilitando a sua localização e reutilização. Surgem os metadados que são ‘dados sobre os dados’, ou mais especificamente, dados a descrever os recursos *web*. O uso de metadados foi a forma definida no SMIL para a sua descrição.

A secção 3.2.1 apresenta um exemplo da forma mais simples de incluir metadados num documento SMIL. O elemento e os atributos usados no código já vêm da versão SMIL 1.0 e são, respectivamente, os seguintes: *meta*; *name* e *content*.

As empresas e os autores individuais são livres de definir os metadados que querem usar. A recomendação SMIL sugere o uso de metadados normalizados, como o *Resource Description Framework* e o *Dublin Core Metadata Initiative*, de forma a promover a troca e interoperabilidade.

3.2.2.7 Estrutura

A Estrutura define quais os elementos básicos para construir um documento SMIL. Os elementos básicos que formam a Estrutura são os seguintes: *smil* (como raiz), *head* (como cabeçalho) e *body* (como corpo principal do documento).

A secção 3.2.1 apresenta uma descrição mais extensa de cada um destes elementos, incluindo alguns exemplos da sua utilização.

3.2.2.8 Temporização e sincronização

Os módulos desta secção definem os elementos e os atributos para coordenar e sincronizar a apresentação dos objectos *media* no tempo. Os objectos *media* podem ser discretos, como as imagens e o texto, ou contínuos, como o vídeo, o áudio e a animação, estes últimos com duração.

Os seguintes elementos de Sincronização comuns podem ser inseridos uns nos outros sem restrições:

- *<seq>* : executa os elementos que contém uns atrás dos outros, sequencialmente;

- `<excl>`: executa apenas um dos elementos contidos de cada vez, mas sem impor qualquer ordenação;
- `<par>`: executa os elementos contidos como um grupo, existindo portanto apresentações paralelas.

O código apresentado na secção 3.2.1 inclui a utilização dos elementos *par* e *seq*. Este exemplo mostra um vídeo em paralelo com uma sequência de imagens e textos.

Os atributos disponíveis permitem definições complexas e avançadas. Podem ser definidas relações entre os objectos e fazer depender a apresentação de cada um deles num evento relacionado com o outro. O próximo código apresenta um exemplo simples deste aspecto:

```
<par begin="-10s" dur="20s">
  
  
</par>
```

No código anterior temos uma apresentação paralela de duas imagens (*slides* no exemplo). Apesar de ser paralela a apresentação, a segunda imagem apenas é apresentada três segundos após o término da primeira.

Uma última nota sobre este exemplo: a primeira imagem nunca será apresentada, apesar de ter definida uma duração de três segundos. Como o elemento *par* tem o início definido para menos dez segundos e uma duração de vinte segundos, a apresentação dos objectos dentro do *par* acontecerá apenas nos últimos dez segundos da sua duração. Isto é, como a primeira imagem dura apenas três segundos, fica dentro dos dez segundos sem apresentação. A segunda imagem seria apresentada ao sexto segundo (três segundos do *slide1*, mais os três segundos de deslocação relativo ao final do *slide1*), mas como este momento ainda está dentro dos primeiros dez segundos, apenas os últimos seis segundos da sua apresentação é que se realizarão.

Esta secção é das mais extensas do SMIL, totalizando dezanove módulos para definições de elementos e atributos.

3.2.2.9 Manipulação Temporal

A Manipulação Temporal, como o próprio nome indica, permite alterar o comportamento do tempo numa apresentação, ou seja, a sua velocidade. Estas manipulações estão pensadas principalmente para *media* discretos e a actuação sobre *media* contínuos nem sempre é suportada. Os atributos definidos para suportar esta funcionalidade são os seguintes: *accelerate*, *decelerate*, *speed* e *autoReverse*.

O código seguinte apresenta a utilização de um destes atributos:

```
<seq speed="2.0">
  <video src="movie1.mpg" dur="10s" />
  <video src="movie2.mpg" dur="10s" />
</seq>
```

O exemplo apresenta dois vídeos em sequência, e pela sua duração poderíamos deduzir que

a apresentação levará vinte segundos. Como o atributo *speed* está presente e com o valor dois, significa que a apresentação se fará no dobro da velocidade e levará metade do tempo, isto é, dez segundos.

3.2.2.10 Transições

As transições são efeitos visuais que ocorrem principalmente na passagem entre diferentes objectos imagem (discretos ou contínuos). Podem também ocorrer transições durante a apresentação de um objecto. As transições áudio não estão definidas nesta versão do SMIL.

O código seguinte apresenta a utilização de transições numa sequência de imagens:

```
...
<transition id="wipe1" type="barWipe" subtype="leftToRight" dur="1s"/>
...
<seq>
  
  
  
  
</seq>
```

O exemplo apresenta a sequência de quatro imagens, cada uma com a duração de cinco segundos. Antes da ocorrência do código onde está a sequência é feita a definição da transição com diversas características. Nos parâmetros da imagem são colocados os dados que identificam a transição a usar e de que modo. A passagem de uma imagem para outra terá agora sempre a inclusão da transição. Durante um segundo haverá o efeito visual do aparecimento da nova imagem e o desaparecimento da antiga por sobreposição, da esquerda para a direita.

A apresentação das diferentes áreas funcionais e módulos desta secção baseou-se numa selecção resumida dos aspectos mais importantes. A lista completa dos elementos e dos atributos, bem como a sua descrição e explicação, estão em [SMIL20].

3.2.3 Perfis de implementação

A estrutura modular do SMIL 2.0 permite aos construtores de *players* combinar os módulos que considerem relevantes e implementar nas suas aplicações apenas as funcionalidades que considerem mais adequadas. A selecção de um conjunto de módulos designa-se por um perfil de implementação.

Os perfis de implementação do SMIL 2.0 definidos até ao momento são os seguintes [Bulterman 02]:

- **linguagem**: usa praticamente todos os módulos definidos pelo SMIL 2.0 e os documentos criados segundo este perfil têm a estrutura definida na secção 3.2.2.7 ;
- **básico**: foi definido para permitir a utilização do SMIL em terminais simples, como telemóveis e PDAs. Mantém a estrutura definida no perfil linguagem e é diferente deste nos módulos que implementa. Este perfil apenas suporta os módulos cuja complexidade seja baixa;

- **XHTML+SMIL**: os módulos escolhidos para constituir este perfil permitem integrar a semântica da temporização e da apresentação noutros documentos XML.

3.3 MPEG-4

A norma MPEG-4, na sequência das normas MPEG-1 e MPEG-2, foi desenvolvida sobre o interesse crescente em três diferentes áreas: televisão digital, aplicações gráficas interactivas (com conteúdo sintético) e multimédia interactiva. O MPEG-4 visa fornecer a normalização dos elementos tecnológicos que integram a produção, a distribuição e o acesso aos conteúdos nestas três áreas [Koenen 00].

O MPEG-4 fornece um conjunto de tecnologias para satisfazer as necessidades dos autores, fornecedores de serviços e utilizadores.

Aos **autores**, o MPEG-4 possibilita a produção de conteúdos reutilizáveis e flexíveis. Permite ainda uma gestão mais eficaz dos direitos dos autores dos conteúdos.

Aos **fornecedores de serviços**, o MPEG-4 oferece informação transparente, que pode ser interpretada e traduzida para os sinais nativos de cada rede com a ajuda da normalização. Não inclui considerações para a qualidade de serviço (QoS), apenas fornecendo um descritor de QoS para diferentes tipos de *media* MPEG-4. A tradução dos parâmetros de QoS para cada um dos diferentes *media* está excluída do âmbito do MPEG-4 e é deixada para tratamento pelos fornecedores da rede.

Aos **utilizadores finais**, o MPEG-4 traz níveis mais elevados de interactividade com os conteúdos, dentro dos limites definidos pelo seu autor. Possibilita também a utilização da multimédia em novas redes, incluindo aquelas onde a taxa de transmissão é baixa, e nas redes móveis.

A todos os envolvidos, o MPEG-4 evita o uso de: diversas normas proprietárias, formatos não compatíveis e diversos *players* [Koenen 00].

Os objectivos do MPEG-4 são obtidos definindo um conjunto de normas que permitem a:

- codificação de objectos que podem ser conteúdos constituídos por elementos visuais, auditivos e audiovisuais. Os objectos podem ser de origem natural ou sintética (podem ser gravados com uma câmara de vídeo ou um microfone, ou gerados por intermédio de um computador);
- descrição da composição dos objectos de forma a criar uma cena audiovisual;
- multiplexagem e sincronização dos dados associados com os objectos, de forma a poderem ser transportados por canais de rede, com uma QoS apropriada, mediante a natureza do objecto;
- interacção com a cena audiovisual gerada do lado do receptor.

A restante secção relativa ao MPEG-4 mostrará alguns aspectos sobre as formas de composição (secção 3.3.1), de sincronização (secção 3.3.2) e de interacção (secção 3.3.3) possíveis no MPEG-4. A codificação e multiplexagem, por estarem fora do âmbito da dissertação, não serão

referidas.

3.3.1 Composição

O MPEG-4 além de possibilitar a codificação de objectos individuais, também permite a sua composição de forma a constituir uma cena. A informação de composição forma a descrição da cena, que é codificada e transmitida junto com os objectos de *media*. Partindo do *Virtual Reality Modeling Language* (VRML), o MPEG desenvolveu uma linguagem binária para a descrição das cenas, designada por *Binary Format for Scenes* (BIFS).

As descrições das cenas são codificadas independentemente dos fluxos dos objectos, para facilitar o desenvolvimento de ferramentas de autoria, manipulação e interacção. É dado um cuidado especial na identificação dos parâmetros pertencentes à descrição da cena (por exemplo, o posicionamento de um objecto), que são diferenciados daqueles que visam melhorar a eficiência da codificação de um objecto.

Apresenta-se de seguida alguns exemplos sobre a descrição de uma cena:

- **agrupar os objectos:** uma cena MPEG-4 segue uma estrutura hierárquica (ver figura 3-3) semelhante a uma árvore. A árvore não é necessariamente estática, podendo também ser dinâmica, visto que existe a possibilidade de alterar os parâmetros de cada nó e até adicionar, substituir ou apagar nós;

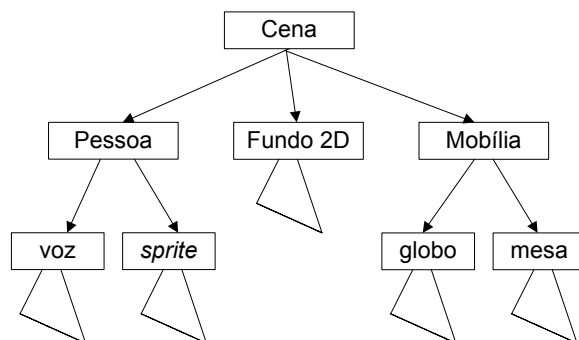


Figura 3-3 Estrutura lógica de uma cena

- **posicionamento dos objectos no espaço e no tempo:** no modelo MPEG-4, os objectos audiovisuais têm uma extensão espacial e temporal. Cada objecto tem um sistema de coordenadas local, que define uma localização e escala espacial e temporal fixas para esse objecto sistema este que serve para manipular o objecto no espaço e no tempo. Os objectos são posicionados numa cena especificando uma transformação das coordenadas locais do objecto com as coordenadas definidas numa descrição de cena de um nó superior da árvore;
- **selecção do valor dos atributos:** os objectos individuais e os nós de descrição de cena expõem um conjunto de parâmetros na camada de composição, alguns dos quais podem ser controlados. Alterar a intensidade de um som, a cor de um objecto sintético, são

exemplos de controlo sobre atributos de um objecto;

- **outras transformações sobre os objectos:** a estrutura da descrição da cena e a semântica de cada nó são muito influenciadas pelo VRML, incluindo também o seu modelo de eventos, o que permite ao MPEG-4 construir cenas complexas, devido ao conjunto rico de operadores construtores de cenas.

3.3.1.1 BIFS versão 2

O MPEG-4, na sua versão 2, melhorou o BIFS, adicionando um conjunto alargado de novas funcionalidades. O novo BIFS designou-se, então, *Advanced BIFS*, ou BIFS versão 2. De entre as melhorias destacam-se as seguintes:

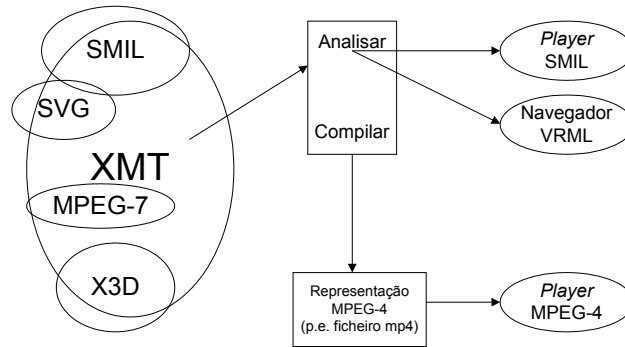
- **modelação avançada do ambiente áudio** em cenas virtuais interactivas; propriedades como reflexões em quartos, reverberação, efeito de *Doppler* e obstrução do som causada por objectos da cena entre o emissor e o receptor do som são processadas em tempo real para fontes de áudio em ambientes dinâmicos;
- **animação corporal** de um modelo corporal existente, por omissão, no decodificador ou de um modelo corporal que pode ser descarregado;
- **chroma keying** usado para gerar uma máscara e uma transparência para uma imagem ou sequência de vídeo;
- inclusão de **redes 3D hierárquicas** para cenas BIFS;
- associação de **comandos interactivos** aos nós de objectos: os comandos são levados para o servidor por um canal de retorno para processamento específico;
- **PROTO e EXTERNPROTO**, nós que permitem uma maior compatibilidade com o VRML e possibilitam a criação de macros que definem o comportamento dos objectos;
- **compressão avançada** dos dados BIFS, em particular a compressão óptima para redes e vectores de dados.

3.3.1.2 Formato textual

Uma outra forma para a descrição de cenas em MPEG-4 é o uso do *eXtensible MPEG-4 Textual Format* (XMT), que usa uma sintaxe textual.

O XMT permite a troca de conteúdos entre os autores, ferramentas ou fornecedores de serviços, e facilita a interoperabilidade com as linguagens X3D (*eXtensible 3D do Web3D Consortium*) e SMIL [Koenen 00].

O XMT pode ser usado nos *players* SMIL, VRML e MPEG-4. A figura 3-3 é uma descrição gráfica da relação entre o XMT e outras linguagens para descrição de cenas. O MPEG-4 usa: uma grande parte do SMIL, do *Scalable Vector Graphics*, do X3D, e também já prevê o uso das descrições de conteúdo da norma MPEG-7.

Figura 3-3 *eXtensible* MPEG-4 Textual Format (XMT)

3.3.2 Sincronização

A camada de sincronização (para mais informações sobre o modelo de camadas do sistema MPEG-4 ver [Koenen 00]) tem um conjunto de ferramentas para verificações de consistência, de enchimento (*padding*), de transporte de informação básica temporal e para carregar unidades de acesso com marcações temporais de um fluxo de um *media* elementar. Os fluxos elementares consistem em unidades de acesso com dados específicos sobre o tempo de descodificação e composição. A camada de sincronização necessita de uma detecção de erros fiável e de enquadrar cada pacote na camada inferior.

A associação de fluxos elementares a objectos de uma cena é feita por intermédio dos descritores dos objectos. Estes carregam informação sobre o número e as propriedades dos fluxos elementares relacionados com um certo objecto de *media*, e são transportados em fluxos elementares. Estes fluxos são considerados como a descrição dos recursos de *streaming* de uma apresentação. De forma semelhante, a descrição da cena também é transportada como um fluxo elementar, permitindo a modificação da composição espaço-temporal da apresentação no tempo.

Modelo avançado de sincronização

O modelo avançado de sincronização, também designado como *FlexTime*, melhora o modelo temporal do MPEG-4 de forma a permitir a sincronização de múltiplos fluxos e objectos, que podem ser originários de múltiplas fontes.

O modelo tradicional de sincronização foi definido basicamente para a simples difusão de aplicações onde a sincronização temporal entre unidade de acesso é conseguida via marcações temporais e relógios de referência. Com estes métodos era possível obter boa sincronização *intra-stream*, mas não permitia a sincronização *inter-stream* de fluxos provenientes de múltiplas fontes (e com diferentes relógios), como é o caso das aplicações sobre a Internet.

O modelo *FlexTime* permite aos autores de conteúdos especificar relações temporais simples entre objectos MPEG-4, como *CoStart*, *CoEnd* e *Meet*. Os autores também podem definir restrições de flexibilidade para os objectos MPEG-4, como se estes fossem “molas elásticas”. Isto permite a sincronização entre múltiplos objectos de acordo com relações temporais, podendo “encolher” e “esticar” os conteúdos dentro de alguns limites.

3.3.3 Interacção

O MPEG-4 permite a interacção do utilizador com o conteúdo e pode ser dividida em duas categorias: do lado do cliente e do lado do servidor.

A interacção do lado do cliente envolve a manipulação do conteúdo e pode tomar várias formas. A manipulação do lado cliente é executada localmente, no terminal do utilizador.

De entre as várias formas que a interacção pode tomar no lado cliente, salienta-se a alteração de um atributo de um nó da cena (como alterar a posição de um objecto, definir como visível ou invisível, ou alterar o tipo de letra, caso seja um texto sintético). A interacção é desencadeada pelo utilizador, utilizando os dispositivos de entrada disponíveis (por exemplo, o rato e o teclado). O terminal MPEG-4 pode processar os comandos de tal forma que o resultado sobre a apresentação, comparativamente a iguais comandos na fonte original do conteúdo, seja o mesmo. Assim, este tipo de interacção não necessita de normalização.

Outras formas de interacção do lado do cliente necessitam do suporte de sintaxe para a descrição da cena, e são especificadas por uma norma. Essa norma é o VRML, e é usada a sua estrutura de eventos, uma vez que fornece um modelo rico sobre o qual os autores de conteúdos podem trabalhar com vista à criação de materiais interactivos atraentes.

A interacção do lado do servidor envolve a manipulação do conteúdo na emissão, iniciada por um evento do utilizador. Apenas acontece se houver um canal de retorno entre o cliente e o servidor.

3.4 Conclusões

As duas normas apresentadas contêm os elementos referidos no título deste capítulo, a composição, sincronização e a interacção. Apesar de cobrirem elementos comuns, são normas com objectivos diferentes e com implementações muito distintas, e não estão em disputa por um espaço dentro das normas que visam definir apresentações multimédia.

O SMIL é uma colecção de elementos e atributos XML que podem ser usados para descrever a coordenação espacial e temporal de objectos de *media*.

O MPEG-4 é uma família de normas que trata um conjunto alargado de aspectos relacionados com os objectos de *media*, e não é apenas uma solução específica para uma qualquer forma de apresentação multimédia [Bulterman 01].

Estas duas normas complementam-se e esse esforço está a ser concretizado na definição do XMT, pertencente ao MPEG-4, mas que contempla o SMIL como uma das formas de descrição textual de uma cena multimédia.

4. Metadados – Modelo de referência de conteúdos reutilizáveis

Este capítulo aborda os metadados para conteúdos educacionais. Será feita uma abordagem à sua importância e às suas funções.

Com relação aos objectivos desta dissertação, foi escolhida uma especificação que tem por finalidade a criação de um ambiente de ensino distribuído, onde a organização dos mais variados conteúdos de ensino pode ser feita de forma a permitir a sua reutilização, independentemente da localização ou da plataforma utilizada.

A norma a explorar será o Modelo de Referência de Conteúdos Reutilizáveis (SCORM) e a abordagem passará por todas as suas componentes, com especial atenção ao Modelo de Agregação de Conteúdos, visto ser a componente usada na aplicação versada nesta dissertação.

4.1 Introdução

A introdução da tecnologia, como forma de incrementar a eficiência do ensino, iniciou-se com uma análise ao processo de aprendizagem. Foram realizados estudos em salas de aulas tradicionais e feitas comparações com o ensino individualizado (via tutor) e encontraram-se grandes diferenças, que se reflectiam no nível de êxito da aprendizagem:

- o ritmo de aprendizagem entre diferentes indivíduos é muito variável, mesmo em turmas que foram seleccionadas cuidadosamente [Gettinger 84];
- um aluno, em média, numa sala de aula faz 0,1 perguntas por cada hora; em ensino individualizado pode chegar às 120 perguntas [Graesser 94];
- a eficiência dos alunos no método individualizado pode exceder no dobro a dos alunos de uma sala de aula – considerando o percentil, os alunos de uma sala de aula normal poderiam, caso optassem pelo ensino individualizado, ver o seu percentil subir de 50% para 98% [Bloom 84].

Ficou claro que o ensino individualizado traz vantagens na eficiência da aprendizagem, mas a sua aplicação em ambientes de grande dimensão, como na administração pública, academias ou indústrias, tem um custo muito elevado e é complexo de implementar.

A utilização da tecnologia surgiu como uma forma de enfrentar esse problema, devido às suas capacidades de funcionamento em tempo real e de adaptação, visto que permitia fornecer instrução individualizada a um preço adequado e obter resultados de aprendizagem concretos. Outro conjunto de características importantes era a possibilidade de ajustar o ritmo, a sequência e o método de instrução que melhor se encaixavam nos interesses e objectivos de cada estudante [Fletcher 03].

As ferramentas que utilizam as novas tecnologias com fins de ensino começaram a surgir nos finais da década de sessenta do século passado e foram classificadas, basicamente, em duas categorias:

- *Computer Based Instruction (CBI)*: é um termo muito genérico e abrange qualquer sistema de ensino que utilize o computador como meio de aprendizagem;
- *Intelligent Tutoring Systems (ITS)*: baseada em estudos sobre inteligência artificial, esta abordagem avalia a forma como os humanos aprendem, desenvolvem capacidades e constróem definições. Os principais requisitos dos ITSs são: gerar instrução em tempo real e a pedido, necessária para estudantes individuais, e suportar diálogos, iniciados em qualquer dos lados, entre o sistema e o estudante.

Com o decorrer dos anos, devido às melhorias tecnológicas, principalmente à maior capacidade de processamento dos computadores pessoais, estas ferramentas e os seus custos de desenvolvimento foram, respectivamente, amadurecendo e decrescendo exponencialmente. Os

conteúdos rapidamente integraram capacidade multimédia e funcionalidades avançadas, mas este desenvolvimento surgiu sobre formatos proprietários e sistemas especializados que não permitiam a utilização dos conteúdos fora do contexto.

Com o aparecimento da Internet, e principalmente da *World Wide Web* (WWW), alteraram-se as capacidades dos CBIs e ITSs. A WWW foi desenvolvida sobre uma estrutura de comunicações largamente acessível e feita sobre normas comuns, permitindo assim um acesso facilitado, em qualquer instante e em qualquer lugar, à informação e ao conhecimento.

As estruturas dos, então actuais, sistemas de autoria CBI eram incompatíveis com a WWW. O conteúdo da WWW era neutro, em termos de plataforma, e guardado e gerido em servidores. Os conteúdos dos sistemas CBI eram armazenados e executados localmente usando linguagens de *script* e motores proprietários. Mas a comunidade CBI rapidamente compreendeu os benefícios da instrução baseada na *web*.

Numa primeira fase a transposição passou por uma adaptação directa dos conteúdos para o ambiente *on-line*, o que obrigou à utilização de *plug-ins* proprietários e o funcionamento não era consistente em ambientes diferentes.

Uma segunda geração de sistemas de autoria começou a seguir a ideia de separar os conteúdos da sua lógica de controlo e apresentação. Surgem então os *Learning Management Systems* (LMSs), cujas funções são: a entrega, o registo, o reporte, a gestão de conteúdos de ensino, a gestão do progresso dos alunos e das suas interações de ensino. Pela primeira vez os criadores de CBIs adoptaram conceitos mais próximos da comunidade ITS. Objectos de aprendizagem partilháveis e reutilizáveis e estratégias de ensino adaptativas começaram a ser uma base comum entre as comunidades CBI e ITS.

Os grupos, instituições e pessoas envolvidas no ensino à distância compreenderam, com a quase ubiquidade da Internet, o grande potencial que um sistema de ensino distribuído eficiente poderia ter. Surgiu, assim, o Modelo de Referência de Conteúdos Reutilizáveis (*Sharable Content Object Reference Model* - SCORM), desenvolvido pela *Advanced Distributed Learning* (ADL), que era um conjunto de especificações e normas com o objectivo de criar uma estrutura técnica comum que possibilitasse a interoperabilidade, a acessibilidade e a reutilização de conteúdos de ensino baseados na Internet.

Os arquitectos da ADL cedo reconheceram a necessidade de um conjunto de normas que especificasse os conteúdos de ensino e a sua descrição, armazenamento e apresentação num ambiente de ensino distribuído e cuja aceitação e implementação fossem generalizadas.

O SCORM definiu um conjunto de requisitos de alto-nível para garantir o cumprimento dos seus objectivos. Esses requisitos foram designados por as “ilidades” (do inglês *ilities*) da ADL, e constituem a base na qual assenta o SCORM. As “ilidades” são as seguintes [SCORM Overview 04]:

- **acessibilidade:** a possibilidade de localizar e aceder a conteúdos remotos que podem ser utilizados noutros pontos;
- **adaptabilidade:** conseguir modelar conteúdos para indivíduos e organizações;
- **custo** (do inglês *affordability*): poder aumentar a eficiência e produtividade, reduzindo o

tempo e os custos associados à disponibilização de conteúdos;

- **durabilidade:** não necessitar de novo projecto, configuração e codificação com a evolução e mudanças das tecnologias;
- **interoperabilidade:** a possibilidade de juntar um conjunto de componentes desenvolvidos num local com um conjunto de ferramentas específicas e/ou plataformas, e usá-los noutro local com um conjunto de ferramentas e plataformas diferentes;
- **reusabilidade:** a flexibilidade de poder utilizar conteúdos em múltiplas aplicações e contextos.

Para além desta lista existiu ainda mais um princípio fundador: o de ser baseado na *web*. As razões desta opção são:

- o ritmo de expansão acelerada das tecnologias e infra-estruturas *web* que cria uma base forte para as tecnologias de ensino;
- a inexistência, de forma abrangente, de normas para ensino via *web*;
- a possibilidade de recorrer a qualquer outro meio (como o CD, ambientes *stand-alone* ou de rede), para a difusão de conteúdos que tenham sido criados para a *web*.

4.2 Organização

O SCORM é um conjunto de livros técnicos independentes que contêm um conjunto de especificações e normas, quase todas de outras organizações (do IMS, da AICC, da ADRIANE e do IEEE LTSC). Os livros estão agrupados em três tópicos: Modelo de Agregação de Conteúdo (*Content Aggregation Model - CAM*), Sequência e Navegação (*SN*) e Ambiente de Execução (*RunTime Environment - RTE*).

Existe, ainda, um livro, designado Descrição, que tem por objectivo abordar os conceitos que envolvem o SCORM de uma forma resumida e de alto nível e fazer a introdução aos conceitos base e a toda a terminologia presente nos restantes livros.

A figura 4-1 apresenta os diversos livros e algumas especificações que os constituem.

O Modelo de Agregação de Conteúdo descreve os componentes usados numa experiência de ensino nos seguintes aspectos: na forma de os empacotar para troca entre diferentes sistemas, de os descrever para pesquisa e descoberta, e na definição de regras de sequência entre eles. Um pacote de conteúdos é descrito por um manifesto, definido num ficheiro XML designado *imsmanifest.xml*.

A Sequência e Navegação definem as possibilidades de sequência da apresentação do conteúdo através de eventos de navegação iniciados pelo aluno ou pelo sistema. A ramificação e fluxo desse conteúdo são descritos por um conjunto de actividades definidas, normalmente, na criação do pacote.

O Ambiente de Execução descreve: os requisitos do LMS para o processo de lançamento de conteúdos, as comunicações entre os conteúdos e os LMSs, e a normalização dos elementos do

modelo de dados usado para a passagem de informação sobre o aluno.

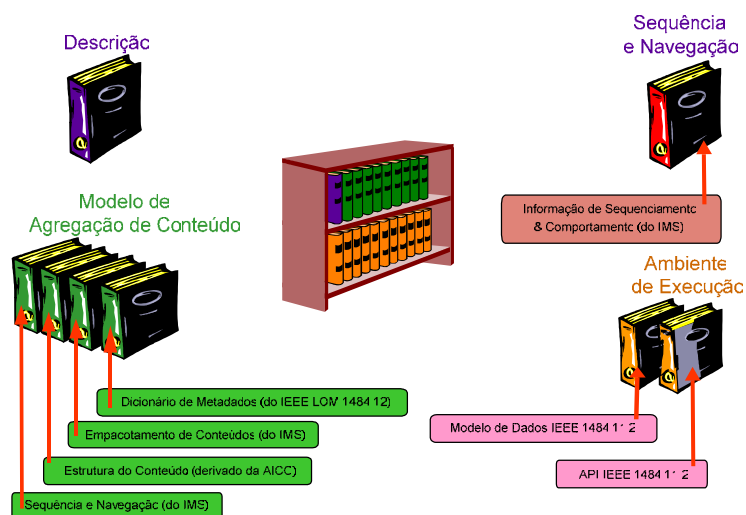


Figura 4-1 Organização geral do SCORM

Os vários livros são independentes entre si, mas tal não impede que haja áreas comuns. Por exemplo, enquanto o RTE trata, essencialmente, da comunicação entre o conteúdo e o LMS, não deixa muitas vezes de referir os tipos de objectos de conteúdos usados, os designados Conteúdos Partilháveis (*Sharable Content Objects* – SCO). A sua definição e descrição surgem no livro sobre o CAM.

As próximas secções deste capítulo dedicam-se a apresentar em maior pormenor os livros mencionados. O CAM será o componente com uma descrição mais alargada, por ter sido o único usado no âmbito do desenvolvimento do sistema abordado nesta dissertação.

4.3 Modelo de agregação de conteúdo

O CAM é um conjunto de livros, que faz parte integrante e fundamental do SCORM. Os livros SCORM CAM contêm os seguintes elementos:

- descrição dos componentes que estão presentes numa experiência de ensino;
- empacotamento desses componentes de forma a permitir a troca entre diferentes sistemas;
- caracterização dos componentes, para permitir a sua procura e descoberta (e assim facilitar a reutilização de conteúdos);
- definição de regras de sequência para os componentes.

O CAM promove o consistente armazenamento, rotulagem, empacotamento, troca e descoberta de conteúdos de ensino. Visa, também, definir responsabilidades e requisitos para a construção e organização de conteúdos (como por exemplo: cursos, módulos, lições, etc.). Ainda presente nesta secção apresenta-se o Empacotamento de Conteúdos que fornece uma forma consistente de descrever as estruturas dos conteúdos, os conteúdos de ensino, os metadados nos

seus componentes, e as regras de sequência e navegação.

4.3.1 Componentes do modelo de conteúdo

O modelo de conteúdos descreve os componentes SCORM usados na construção de experiências de ensino a partir de recursos educacionais. Define, também, como esses recursos de baixo-nível são partilhados e agregados em unidades de instrução de alto-nível.

Os componentes que existem são: os objectos simples, os conteúdos partilháveis (SCOs) e a organização de conteúdos.

4.3.1.1 Objectos simples

Os objectos simples, apresentados na figura 4-2, como o nome indica, são os conteúdos unitários e básicos. São a forma mais simples de um recurso educacional e podem ser um texto, uma imagem, um som, uma página *web* ou, ainda, qualquer outro tipo de dados que possa ser apresentado através da *web*. Pode existir a junção de objectos simples para a formação de um outro objecto simples [SCORM CAM 04].

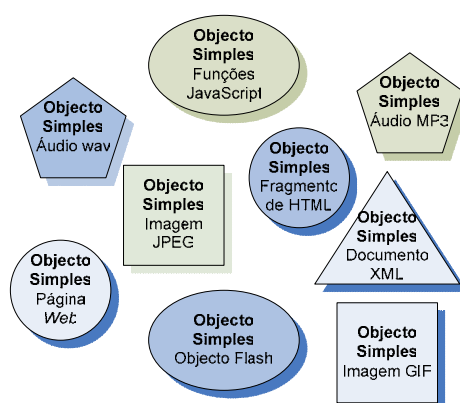


Figura 4-2 Exemplos de objectos simples

Estes componentes são descritos nos metadados (definidos na norma), de forma a permitir a procura e a localização em repositórios e a possibilitar a reutilização. A associação dos objectos simples aos seus metadados passa pelo empacotamento de conteúdos (a ser descrito na secção 4.3.2).

4.3.1.2 Conteúdos partilháveis

Os conteúdos partilháveis (SCO) (ver exemplo na figura 4-3) são uma colecção de um ou mais objectos simples que representam um recurso educacional que pode ser lançado utilizando o RTE para comunicar com o LMS. Os SCOs representam o nível mais baixo de granularidade de um recurso educacional que é seguido por um LMS usando o RTE. Um SCO pode comunicar com o LMS, mas tal não é possível aos objectos simples (devem-se tornar SCOs para isso possível).

Os SCOs devem ser independentes do contexto de ensino para que possam promover a reutilização. Outro pormenor para facilitar a reutilização é cada SCO ser pequeno e simples. Um SCO pode ser descrito com metadados pelas razões já enunciadas para os objectos simples

(procura, localização e reutilização). A associação de SCOs aos seus metadados passa igualmente pelo empacotamento de conteúdos.

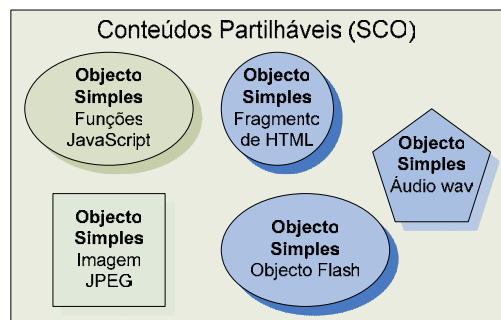


Figura 4-3 Exemplo de um conteúdo partilhável (SCO)

Os SCOs devem satisfazer um conjunto de requisitos definidos no RTE, com os seguintes benefícios:

- qualquer LMS que suporte o SCORM RTE pode lançar e seguir SCOs, independentemente de quem os gerou;
- qualquer LMS que suporte o SCORM RTE pode seguir qualquer SCO e saber quando foi iniciado e terminado;
- qualquer LMS que suporte o SCORM RTE pode lançar diferentes SCOs da mesma forma.

4.3.1.3 Organização de conteúdos

A organização de conteúdos é um mapa representativo do uso pretendido dos conteúdos organizados em unidades estruturadas de instrução (adiante designadas como actividades). O mapa da figura 4-4 representa a forma como as actividades (elementos *Item*) se relacionam entre elas.

As actividades podem ser constituídas por outras actividades (sub-actividades), que por sua vez podem conter outras actividades, não existindo limite ao número de níveis que podem ser criados. Normalmente existe uma estrutura de ensino comum (por exemplo, curso, capítulo, módulo, etc.), mas tal não é obrigatório e pode ser criada a estrutura que se achar mais adequada a um certo objectivo de aprendizagem.

As actividades que não são formadas por outras actividades (designadas actividades folha - *leaf* – por analogia com uma árvore) terão associado um recurso educacional, objecto simples ou SCO, que será usado para desenvolver a tarefa que estiver definida nessa actividade.

A organização de conteúdos e as actividades poderão ter associados metadados, com os fins já enunciados. A associação também ocorrerá da mesma forma, ou seja, utilizando o empacotamento de conteúdos.

A informação de sequência estrutura a relação entre as diferentes actividades, aplicando-se somente a estas, e faz parte da organização de conteúdos. A separação da informação de sequência dos recursos educacionais é um aspecto conceptual importante, visto que não limita a reutilização destes últimos, já que não estão directamente relacionados com um contexto de curso específico.

Este relacionamento apenas acontece via actividades. A reutilização de um recurso educacional depende da sua independência e unidade.

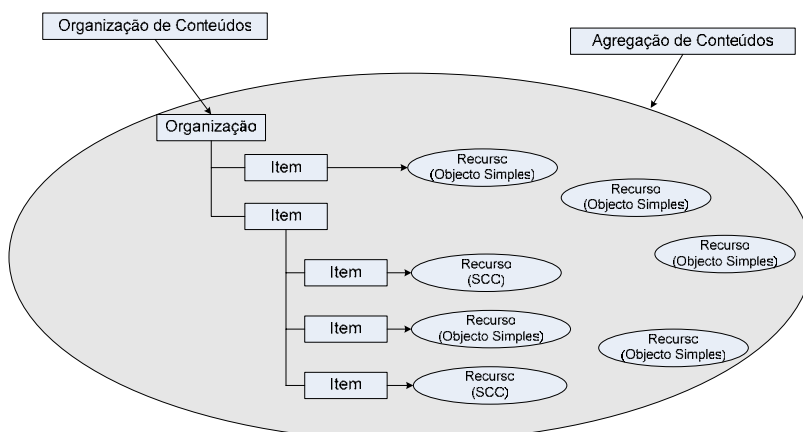


Figura 4-4 Exemplo de organização de conteúdos

O objectivo dos metadados é identificar, categorizar, assim como permitir pesquisas e localização num ou em vários sistemas para facilitar o uso e a reutilização de conteúdos.

A aplicação de metadados no empacotamento de conteúdos está definida em cinco pontos:

- manifesto: neste nível (como é o caso da agregação dos conteúdos) os metadados têm de ser concordantes com as ligações IEEE LTSC LOM, mas não têm mais restrições impostas pelo SCORM;
- organização: neste nível os metadados definem a organização como um todo, que pode ser um curso, uma unidade, uma lição ou outra qualquer unidade de instrução;
- item: descrevem a hierarquia de actividades de um forma sensível ao contexto;
- recurso: descrevem um SCO ou um objecto simples, independente do restante contexto;
- ficheiro: descrevem um objecto simples, independente do restante contexto.

4.3.2 Empacotamento de conteúdos

A especificação empacotamento de conteúdos do IMS foi desenhada para fornecer uma forma normalizada de estruturar e trocar conteúdos de ensino entre diferentes sistemas e ferramentas. O empacotamento de conteúdos contém dois componentes principais:

- um documento XML especial descrevendo toda a estrutura do conteúdo e dos recursos associados ao pacote, que se designa por ficheiro de manifesto (*imsmanifest.xml*). É obrigatório este documento estar na raiz do pacote;
- os ficheiros físicos que fazem parte do pacote.

A figura 4-5 mostra de que forma os componentes estão presentes num pacote de conteúdos.

O pacote representa uma unidade de aprendizagem. Esta unidade pode ser parte de um

curso que tem relevância fora da organização de outro curso e pode ser apresentada de forma independente, como um curso, parte de outro curso ou como sendo uma colecção de cursos.

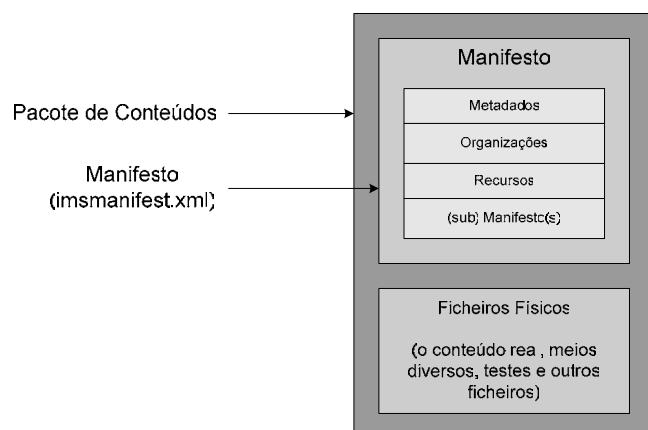


Figura 4-5 Diagrama conceptual do pacote de conteúdos

O manifesto é um documento XML que contém um inventário estruturado do conteúdo de um pacote. Cada pacote deverá ter apenas um manifesto principal e este então conter um ou mais sub-manifestos. O manifesto principal descreverá sempre o pacote e os sub-manifestos descreverão os conteúdos dentro do seu âmbito.

O Ficheiro de Intercâmbio de Pacote (*Package Interchange File* - PIF) é a junção de todos os componentes num ficheiro comprimido. Irá conter o *imsmanifest.xml*, todos os ficheiros de controlo e os recursos referenciados pelo pacote de conteúdos (apenas os ficheiros locais ao PIF). O uso deste formato é uma recomendação do SCORM devido à fiabilidade na transferência entre sistemas via Internet.

4.3.3 Componentes de um manifesto

O manifesto (apresentado na figura 4-6) contém toda a informação necessária para descrever os conteúdos do pacote e é composto por quatro componentes: metadados, organizações, recursos e, caso existam, sub-manifestos.



Figura 4-6 Componentes de um manifesto

4.3.3.1 Metadados

Os metadados são “dados sobre os dados” [SCORM CAM 04] que permitem conhecer um

pouco de um determinado conteúdo sem ser necessária a sua visualização. Os metadados do manifesto descrevem os conteúdos do pacote (por exemplo, a agregação dos conteúdos) como um todo. Além desta função, estes metadados, permitem a procura e localização de pacotes. A descrição mais detalhada dos metadados é feita na secção 4.3.5.

4.3.3.2 Organizações

O componente organizações é usado para descrever as organizações dentro do pacote de conteúdos. Podem existir várias organizações, cada qual descrevendo uma estrutura específica para os conteúdos, sendo considerada uma delas como a estrutura a ser usada por omissão por um LMS (acontece quando não existe informação a indicar que estrutura **usar**). A única estrutura definida pelo IMS é a estrutura em árvore ou hierárquica, sem representar qualquer tipo específico de organização de ensino (como curso, módulo, lição,...). Essa nomenclatura é da responsabilidade dos criadores de conteúdos.

Comment [D2]: Informação que pode existir que leve a escolher uma dada estrutura pode ser por exemplo o perfil do utilizador (iniciado, avançado) e existir uma estrutura adequada para cada um deles.

A organização de conteúdos não deve ser confundida com a organização dos ficheiros em directorias nem com a estrutura do manifesto. Também não pode ser entendida como um inventário dos recursos educacionais e da sua descrição. O propósito da organização de conteúdos é permitir ao criador de conteúdos especificar o(s) modo(s) como esses mesmos conteúdos poderão ser usados. A organização de conteúdos surge como uma hierarquia de actividades, sobre a qual poderão ser aplicadas regras e acções, a serem apresentadas de forma a produzir experiências de ensino semelhantes em qualquer ambiente LMS compatível com o SCORM. O LMS poderá mesmo criar a sua própria organização, não utilizando nenhuma criada previamente e presente no pacote. No caso de não existirem regras a organização de conteúdos é apenas um mapa que permitirá ao utilizador navegar à sua vontade pelos conteúdos nos quais tiver interesse.

A organização de conteúdos inclui um conjunto de componentes, cada qual definindo diferentes aspectos da estrutura dos conteúdos:

- **hierarquia de conteúdos:** é uma representação em forma de árvore que cria uma organização lógica para os recursos educacionais ou actividades que utilizem esses recursos (um exemplo na figura 4-7). Dependendo da estrutura optada, a hierarquia criada pode representar conceitos como cursos, capítulos, tópicos, ou outros semelhantes, para apresentação ao aluno. O IMS define um conjunto de termos (apresentados em inglês, visto reflectirem a nomenclatura que é usada no manifesto) usados para representar a hierarquia de conteúdos. O termo *organization* é o elemento raiz, do qual surgem os *items* que têm associados outros *items* ou *resources*. Os *resources* podem ser de dois tipos: objectos simples ou SCOs.

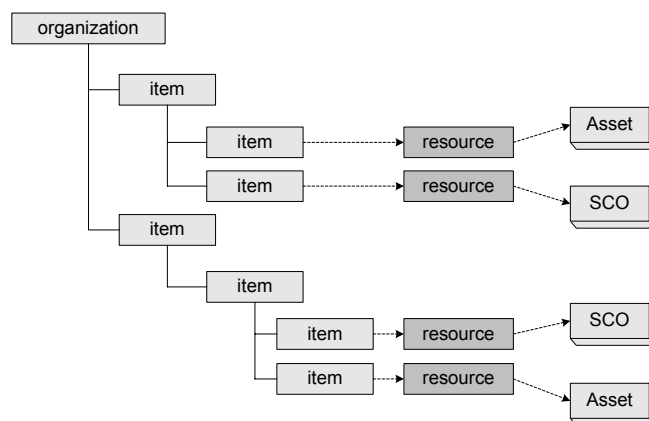


Figura 4-7 Terminologia para a hierarquia de conteúdos

- **metadados:** os metadados da organização de conteúdos são opcionais, descrevem uma actividade e designam-se por metadados específicos do contexto. Este tipo de metadados pode ser usado para descrever, por exemplo, as competências necessárias ou os objectivos a atingir com um recurso educacional usado numa organização de conteúdos particular.
- **sequência, sequência adaptativa e navegação:** indicações adicionais que podem ser inseridas na organização de conteúdos, se o seu autor quiser controlar a navegação dos alunos para obter um certo objectivo educativo. A indefinição de sequência e/ou navegação dará ao aluno a possibilidade de escolher a apresentação dos recursos que entender, na sequência que mais lhe interessar. A experiência de ensino ainda poderá ser adaptativa, com base no estado actual da apresentação dos conteúdos, em processamento mais complexo baseado no perfil do aluno ou ainda no resultado de testes. As regras a inserir são associadas a elementos da árvore da organização de conteúdos, cada um representando uma actividade. A aplicação de regras, tipicamente, resulta em lançar um dado conteúdo ou levar a uma escolha condicionada do aluno.

A organização de conteúdos representa uma variedade alargada de visões para os conteúdos de um pacote, desde pequenas unidades de ensino até conteúdos interactivos complexos monitorizados por LMSs.

4.3.3.3 Recursos

O componente recursos de um manifesto (figura 4-8) descreve e inventaria os ficheiros presentes no pacote ao qual o manifesto pertence. Pode, também, apontar para recursos remotos, externos ao pacote. Os ficheiros poderão ser de qualquer tipo: texto, vídeo, áudio, ou qualquer dado digital. Esta componente admite a formação de grupos conceptuais e a definição de relações entre diversos objectos. Um recurso é constituído por objectos simples, mas se o recurso pretende comunicar com um LMS então esses objectos simples terão de constituir um SCO.

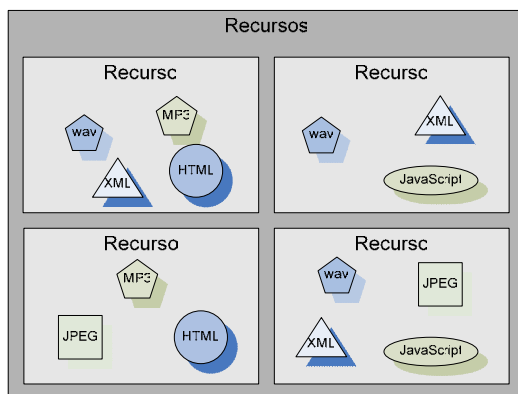


Figura 4-8 Recursos de um manifesto

4.3.3.4 Ficheiros

O componente ficheiros refere-se aos documentos digitais referenciados no componente recursos. Os ficheiros podem ser ficheiros locais, contidos no pacote de conteúdos, ou podem ser ficheiros externos referenciados pelo seu URI.

4.3.4 Perfis de pacotes de conteúdos

Os perfis de pacotes de conteúdos do SCORM descrevem a forma como o empacotamento de conteúdos do IMS (base dos pacotes de conteúdos) é aplicado no contexto geral do SCORM. O SCORM impõe ainda um conjunto de requisitos para assegurar a introdução de informação suficiente nos pacotes de forma a garantir interoperabilidade, importação e exportação de pacotes, entre sistemas utilizadores da norma SCORM.

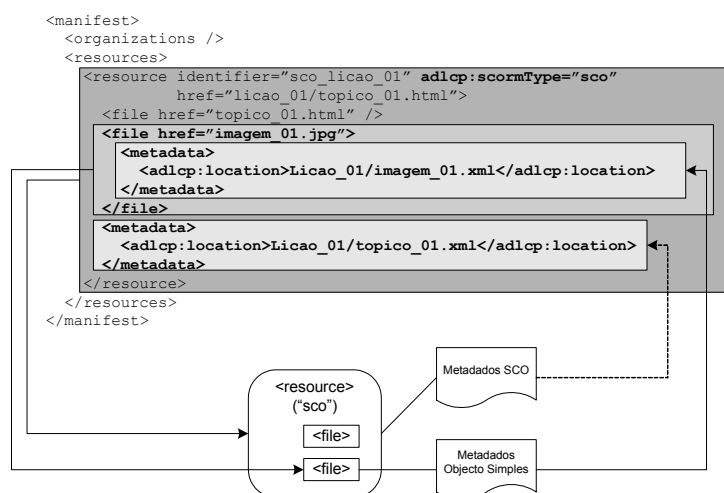


Figura 4-9 Exemplo de um SCO representado como um elemento <resource>

Existem, actualmente, dois perfis definidos para descrever o empacotamento dos componentes do modelo de agregação de conteúdos:

- **pacote de recursos:** este perfil define um mecanismo de empacotamento de objectos

simples e de SCOs (independente do número de ficheiros), sem ser necessário conter qualquer tipo de organização, contexto de ensino ou estrutura curricular. Visa, principalmente, o movimento de recursos entre sistemas. Os ficheiros estarão fisicamente no pacote ou não, dependendo se a referência é interna ou externa. O perfil pacote de recursos não permite que um LMS apresente estes recursos directamente a um utilizador. A figura 4-9 é um exemplo de um manifesto segundo este perfil, onde está definido um recurso do tipo SCO e onde são apresentados exemplos de metadados desse SCO, como também de um dos objectos simples que o constitui.

- **pacote de agregação de conteúdos:** este perfil deve ser usado quando se pretende agregar conteúdos e as suas possíveis estruturas, ou seja, constituir cursos, módulos, lições, etc. Desta forma um LMS entende como pode processar os conteúdos e apresentá-los a um aluno. A figura 4-10 é um manifesto, extensão do apresentado na figura 4-9, que respeita o perfil pacote de agregação de conteúdos e apresenta o detalhe de uma estrutura possível para os conteúdos declarado na sua secção `<resources>`.

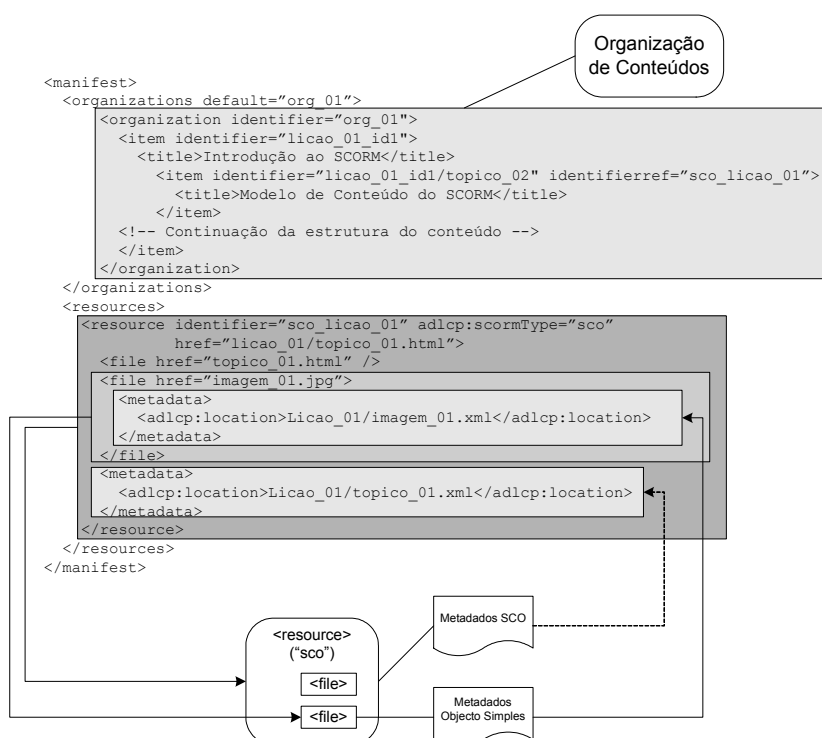


Figura 4-10 Exemplo de um pacote de agregação de conteúdo

A norma define claramente em ambos os pacotes quais os elementos do manifesto obrigatórios, opcionais e não permitidos.

4.3.5 Metadados

O propósito dos metadados é fornecer uma nomenclatura comum de descrição dos recursos educacionais para facilitar a pesquisa e descoberta em diversos sistemas e a consequente reutilização. Os metadados podem existir em catálogos ou junto dos recursos que descrevem. Uma outra utilização possível para os metadados é apoiar a decisão sobre qual o próximo recurso a ser apresentado num ambiente de execução.

O SCORM recomenda a norma IEEE *Learning Object Metadata* (LOM) para descrição dos recursos. Esta norma contém aproximadamente sessenta e quatro elementos e distingue os elementos obrigatórios dos opcionais. Podem ser usados outros esquemas de metadados, mas essa escolha pode não garantir o seu reconhecimento por sistemas diferentes.

O modelo de informação do LOM está dividido em nove categorias, apresentadas de seguida:

- **Geral (General)**: usada para descrever informação geral sobre o componente como um todo. Exemplo de elementos: <identifier>, <title> e <aggregation level>;
- **Ciclo de Vida (Life Cycle)**: usada para descrever o histórico e o estado actual do componente e dos envolvidos nas alterações que foram ocorrendo. Exemplos de elementos: <version>, <status> e <contribute>;
- **Meta-metadados (Meta-metadata)**: usada para descrever os metadados criados, ou seja, são metadados sobre os próprios metadados. São uma fonte de nova informação relevante que os restantes elementos dos metadados não descrevem. Exemplos de elementos: <metadataScheme> e <language>;
- **Técnica (Technical)**: usada para descrever os requisitos e características técnicas do componente. Exemplos de elementos: <format>, <otherPlatformRequirements> e <installationRemarks>;
- **Educacional (Educational)**: usada para descrever as características educacionais e pedagógicas do componente. Exemplo de elementos: <interactivityType>, <learningResourceType>, <semanticDensity> e <difficulty>;
- **Direitos (Rights)**: usada para descrever direitos de propriedade intelectual e condições de utilização do componente. Exemplo de elementos: <cost>, <description> e <copyrightAndOtherDescription>;
- **Relação (Relation)**: usada para descrever a relação entre o componente sobre o qual versa a descrição e outros componentes. Exemplo de elementos: <kind> e <resource>;
- **Anotação (Annotation)**: usada para deixar comentários sobre o uso educacional do componente, e informação sobre quem e quando o comentou. Exemplos de elementos: <entity> e <description>;
- **Classificação (Classification)**: usada para descrever o sistema de classificação onde será enquadrado o componente. Exemplos de elementos: <purpose> e <taxonPath>.

Os perfis de aplicação dos metadados em SCORM definem a integração do IEEE LOM no ambiente SCORM. Os perfis de aplicação definem os elementos obrigatórios e opcionais para cada componente quando os metadados são aplicados.

A norma ainda define, para os metadados, as formas de validação dos esquemas para o modelo de dados do LOM e as capacidades de extensão.

4.4 Sequência e navegação

A sequência e a navegação descrevem como os conteúdos compatíveis com a norma SCORM podem ser apresentados ao aluno, mediante a implementação de regras de sequência e definições de navegação. A apresentação pode ocorrer através de eventos de navegação iniciados pelo utilizador ou pelo sistema. A ramificação e o fluxo do conteúdo podem ser descritos por um conjunto de actividades.

No âmbito da norma SCORM, este tópico tem um livro próprio (*Sequencing and Navigation*), que cobre as obrigações que um LMS tem para apresentar conteúdos (SCOs ou objectos simples) e permitir pedidos de navegações sobre os SCOs.

A apresentação deste tema contempla duas secções: sequência e navegação. Os tópicos que serão, de uma forma resumida, apresentados são os seguintes:

- conceitos e terminologia da sequência;
- modelo de definição da sequência;
- modelo do comportamento da sequência;
- controlo e requisitos da navegação;
- modelo de dados da navegação.

4.4.1 Sequência

A sequência e navegação do SCORM (SN) são, em parte, baseadas na especificação Sequência Simples do IMS (IMS SS), que define uma forma de representar comportamentos pretendidos de um determinado conteúdo, de tal maneira que em diferentes LMSs a experiência de aprendizagem seja semelhante e consistente com a definida pelo autor. O IMS SS não aborda alguns tópicos relacionados com a sequência como: princípios de inteligência artificial, agendamento, dados de sistemas externos fechados e serviços, ensino parametrizado ou sincronização entre actividades de ensino paralelas. Também reconhece apenas a existência do aluno e não define qualquer tipo de relação com outros intervenientes (como alunos, instrutores ou mentores).

Os próximos parágrafos descrevem alguns dos conceitos associados à sequência, como: árvore de actividades, *clusters*, sub-manifestos, actividades de ensino, tentativa, sessão de sequência, monitorização do estado das actividades e objectivos de ensino.

4.4.1.1 Árvore de actividades

A SN descreve a ramificação e o fluxo de actividades de ensino em termo de árvore de actividades, baseada nos resultados das interacções do aluno com os conteúdos apresentados e com a estratégia de sequência criada.

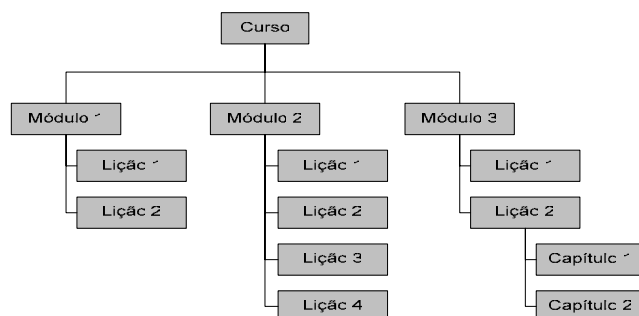


Figura 4-11 Exemplo de uma árvore de actividades

Esta árvore permite descrever requisitos de informação e processamento de uma forma independente da implementação. A figura 4-11 apresenta um exemplo de uma árvore de actividades.

Os diferentes sistemas que implementem a sequência deverão ter uma forma própria de implementar esta estrutura ou como uma árvore ou de uma outra maneira.

4.4.1.2 Clusters

O CAM define uma estrutura hierárquica de conteúdos de ensino. Esta organização de conteúdos é representada no pacote de conteúdos dentro de um único elemento *<organization>*. Cada item definido representa uma unidade relevante de aprendizagem. Esta forma de organização permite que a conversão para uma árvore de actividades seja simples. A organização de conteúdos tem no elemento *<organization>* a sua raiz e nos seus itens as diversas actividades de ensino.

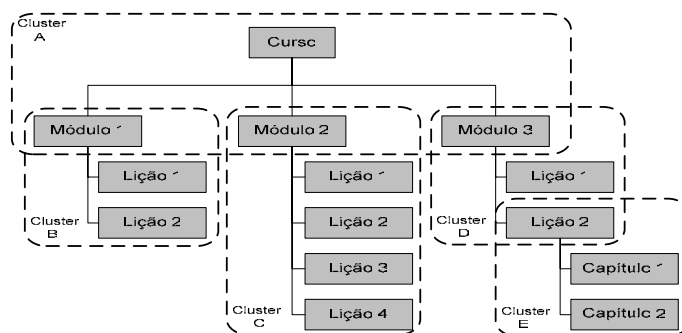


Figura 4-12 Exemplo de cluster

A árvore de actividades tem ainda uma forma especializada de se organizar. Em cada actividade de ensino que tenha sub-actividades, constitui-se um grupo de actividades designado *cluster*. Um *cluster* é formado por uma actividade “pai” e todos os seus “filhos”, que podem ser

“pais” noutros *clusters* ou actividades folha (associadas aos recursos, SCOs e objectos simples). Os descendentes dos “filhos” de um *cluster* já não fazem parte desse *cluster*.

A figura 4-12 apresenta cinco *clusters*, estando cada um definido por um quadrado de cantos redondo com linhas tracejadas.

O *cluster* superior, Curso, contém quatro actividades, o Curso e os seus três módulos, que são actividades “pai” dos *clusters* B, C e D. Cada módulo forma com as suas lições um novo *cluster* (B, C e D). Todas as lições são actividades folha, logo já não podem constituir *clusters*, excepto a “Lição 2” do “Módulo 3” que surge como o *cluster* E, tendo como “pai” a “Lição 2” e “filhos” os dois capítulos.

O *cluster* é considerado a unidade básica da árvore de actividades e muitos elementos do modelo de definição da sequência (como modos de controlo da sequência, controlo de escolha condicionada, entre outros) aplicam-se especificamente a estes agrupamentos. A estratégia de sequência do *cluster* estará presente na actividade “pai”.

Comment [E3]: Outros elementos:
- Regras de sequência;
- Condições limite;
- Controlo Rollup.

4.4.1.3 Sub-manifestos

O uso de sub-manifestos num pacote de conteúdos aparece associado aos elementos *<item>*. Os sub-manifestos terão as suas estruturas, que serão integradas na árvore formada a partir dos itens associados. A figura 4-13 mostra a conversão de uma organização com sub-manifestos numa árvore de actividades.

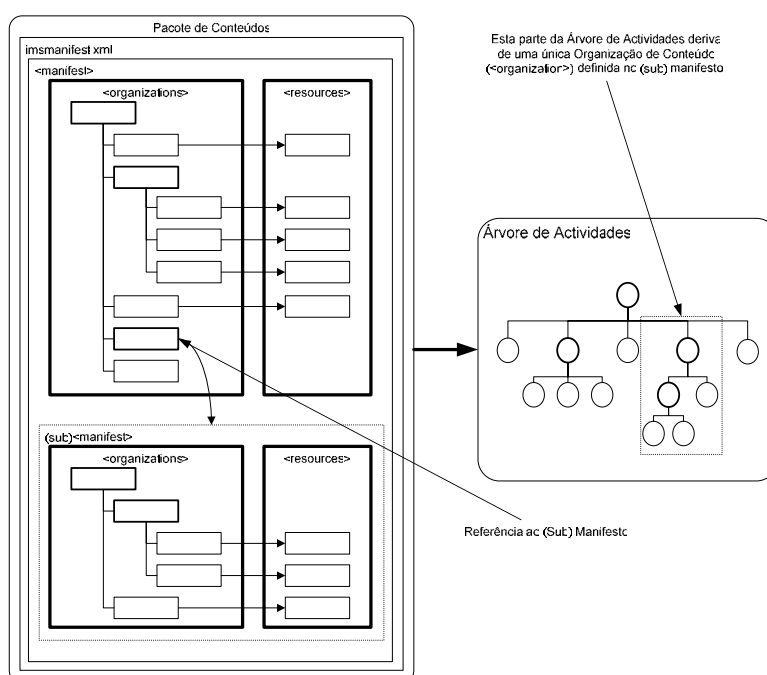


Figura 4-13 Conversão de um manifesto com sub-manifestos em árvore de actividades

Uma árvore de actividades pode ser construída a partir da organização de conteúdos presente no pacote de conteúdos, mas o SCORM não impõe o momento da conversão pelo LMS,

nem a sua representação interna.

4.4.1.4 Actividade de ensino

Uma actividade de ensino pode ser descrita como uma unidade de instrução coerente. É, conceptualmente, algo que faz o aluno progredir e pode ser constituída por um recurso de ensino ou por várias sub-actividades. O conceito actividade de ensino será referido adiante como actividade (este termo já foi usado na secção 4.3.1 para referir “unidades estruturadas de instrução” que tem o mesmo significado de actividade de ensino). A figura 4-14 apresenta uma actividade designada “Obter Lição”. É composta por três sub-actividades: “Pré-Teste”, “Visualizar Conteúdo” e “Teste Final”.

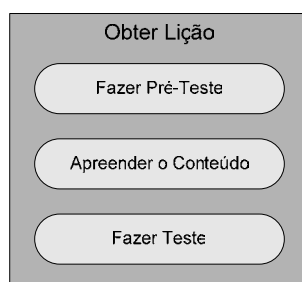


Figura 4-14 Exemplo de uma actividade de ensino

As sub-actividades apresentadas podem ser formadas por um conjunto de actividades e assim sucessivamente, sem limite da norma, mas de razoabilidade relativa ao conteúdo. Os conteúdos que serão visualizados aparecem associados a actividades folha. As actividades têm as seguintes características:

- têm um início e um fim;
- têm condições de conclusão e domínio bem definidas;
- podem ser constituídas por sub-actividades sem limite;
- ocorrem no contexto da actividade “pai”, caso exista.

4.4.1.5 Tentativa

O conceito tentativa é definido como o esforço para completar uma actividade e durante esse esforço podem ou não ser atingidos os objectivos de ensino. As tentativas num dado instante acontecem somente numa actividade folha e ocorrem sempre no contexto da sua actividade “pai”, que está com o seu estado “em progresso”. Deduz-se que os recursos associados à actividade folha foram lançados. No caso em que não seja possível iniciar e terminar uma tentativa, o aluno pode sempre suspender a apresentação e retomar mais tarde, sendo considerada a mesma tentativa. A informação variável do progresso do aluno na actividade é actualizada em cada tentativa, alterando o seu estado para essa actividade. Esta alteração deve afectar a actividade “pai”, acção designada por *Rollup*.

4.4.1.6 Sessão de sequência

Uma sessão de sequência é o tempo desde o início da tentativa, na actividade raiz de uma árvore de actividades, até ao seu final. O início, suspensão, retoma e final da sessão são da responsabilidade do LMS. Neste ponto o SCORM apenas define os pedidos possíveis que iniciam uma sessão de sequência.

4.4.1.7 Monitorização do estado da actividade

O utilizador, ao efectuar uma tentativa de uma actividade, deverá ter dados de monitorização associados. É nestes valores que o comportamento de sequência se baseia para controlo da sequência a apresentar.

A sequência diferencia os conteúdos em comunicativos ou não-comunicativos para cada actividade. No primeiro caso existe uma comunicação entre as interacções do utilizador e o ambiente de execução (descrito da secção 4.5), enquanto num conteúdo não-comunicativo não a utiliza.

A tentativa sobre uma actividade pode ser suspensa e retomada mais tarde, não contando como uma nova tentativa. Não existem impedimentos ao número de actividades suspensas para poder aceder a uma dada actividade.

A persistência dos dados de monitorização é dependente do LMS. O SCORM e o IMS SS não definem nenhuma forma de existir ligação dos dados entre diferentes sessões de sequência.

4.4.1.8 Objectivos de ensino

Os objectivos de ensino estão separados das actividades de ensino. O SCORM não impõe restrições sobre a forma como os objectivos estão associados com as actividades, nem define como os conteúdos utilizam esses mesmos objectivos. Numa perspectiva de monitorização, é mantido um conjunto de informações de estado para cada objectivo associado a uma actividade.

Uma actividade pode ter mais de um objectivo associado. Por omissão, o estado do objectivo de uma actividade é mantido local a essa actividade. A partilha da informação do estado do objectivo entre diferentes actividades é obtida via objectivos globais partilhados. Os objectivos globais podem ser partilhados dentro de uma mesma árvore de actividades ou entre várias árvores de actividades presentes num LMS.

A referência a objectivos globais partilhados por parte de uma actividade tem duas restrições:

- um objectivo local apenas pode obter o dado de estado de um e apenas um objectivo global partilhado;
- num conjunto de objectivos locais de uma actividade, apenas um pode alterar o dado de estado de um objectivo global partilhado.

A sequência ainda contempla mais algumas secções importantes, mais vocacionadas para a implementação e não para os conceitos gerais. Serão apresentados de seguida dois tópicos importantes de forma muito resumida:

- **modelo de definição da sequência:** define um conjunto de elementos usados para descrever diferentes comportamentos de sequência. Baseado na norma IMS SS do IEEE, é estendido por elementos novos definidos de forma a cobrir requisitos de comportamento e de restrição não contemplados por essa norma;
- **comportamentos de sequência:** define os comportamentos associados ao processo de sequência. Este processo utiliza informação de três modelos definidos: monitorização, estado de actividade e definição da sequência (modelo de definição da sequência).

4.4.2 Navegação

O tópico da navegação relaciona-se com os eventos de navegação iniciados pelo sistema e/ou pelo aluno. Este tópico aborda o conjunto e a forma dos eventos serem desencadeados e processados, procurando a próxima actividade de ensino a ser entregue. É um processo de cooperação entre o aluno e o LMS, que gere pedidos de navegação de forma a providenciar uma experiência de ensino.

A navegação pressupõe a existência de dispositivos de interface com o utilizador, fornecidos pelo LMS ou embebidos nos conteúdos, para desencadear os eventos de navegação. O livro do SCORM que contém as especificações de navegação descreve um modelo de execução que os SCOs podem usar para indicar pedidos de navegação ao LMS. Estes pedidos correspondem ao movimento pela árvore de actividades.

A tabela 4-1 apresenta o conjunto de eventos de navegação e define como são mapeados para pedidos de navegação. Existe ainda uma terceira coluna que indica quais os intervenientes que podem desencadear os eventos.

Tabela 4-1 Eventos de navegação e sua descrição

Evento de Navegação	Descrição do Comportamento	Fonte de Pedido
<i>Start</i>	Este evento indica a pretensão de identificar a primeira actividade ou a actividade inicial disponível na árvore de actividade. É, tipicamente, gerado de forma automática pelo LMS quando o aluno começa uma nova tentativa na actividade raiz.	Apenas LMS
<i>Resume all</i>	Este evento indica a pretensão em retomar uma tentativa previamente suspendida na actividade raiz.	Apenas LMS
<i>Continue</i>	Este evento indica a pretensão de identificar a “próxima” actividade de ensino lógica disponível na árvore de actividades.	LMS e SCO
<i>Previous</i>	Este evento indica a pretensão de identificar a actividade de ensino lógica “anterior” disponível na árvore de actividades.	LMS e SCO
<i>Choose</i>	Este evento indica a pretensão de “saltar” directamente para uma actividade de ensino específica da árvore de actividades.	LMS e SCO
<i>Abandon</i>	Este evento indica a pretensão de terminar, prematura ou anormalmente, a actual tentativa sobre o conteúdo corrente, sem intenção de recomeçar posteriormente.	LMS e SCO
<i>Abandon All</i>	Este evento indica a pretensão de terminar, prematura ou anormalmente, a actual tentativa sobre a actividade raiz, sem intenção de recomeçar posteriormente.	LMS e SCO

Evento de Navegação	Descrição do Comportamento	Fonte de Pedido
<i>Suspend All</i>	Este evento indica a pretensão de suspender a tentativa actual sobre a actividade raiz.	Apenas LMS
<i>Unqualified Exit</i>	Este evento indica que o término da tentativa actual sobre a actividade de ensino corrente foi normal, e que esse término não foi desencadeado por nenhum evento de navegação, como <i>Continue</i> , <i>Previous</i> , ou <i>Choose</i> .	LMS e SCO
<i>Exit All</i>	Este evento indica que o término da tentativa actual sobre a actividade de ensino corrente foi normal.	LMS e SCO

A norma, intencionalmente, não impõe qualquer requisito: no tipo ou no estilo de interface com o utilizador durante a execução, nos dispositivos que possibilitam a navegação e nos mecanismos para capturar as interações entre o utilizador e os LMS.

O modelo de navegação apenas se aplica à navegação entre actividades de ensino. A norma, neste momento, não possibilita a definição de sequência e navegação dentro de um SCO (o que não impede a existência de mecanismos internos ao SCO que possibilitem alguma navegação intra-SCO, como por exemplo um SCO com várias páginas).

4.5 Ambiente de execução

O livro ambiente de execução (RTE) do SCORM define os requisitos para o lançamento dos conteúdos, o estabelecimento da comunicação entre os LMSs e os SCOs, e a monitorização e a gestão da informação que possa ser trocada entre os SCOs e os LMSs [SCORM RT 04]. Descreve também um mecanismo comum de lançamento de conteúdos, de comunicação entre os objectos e os LMSs, e um modelo de dados comum para monitorizar as experiências de ensino dos alunos com os objectos.

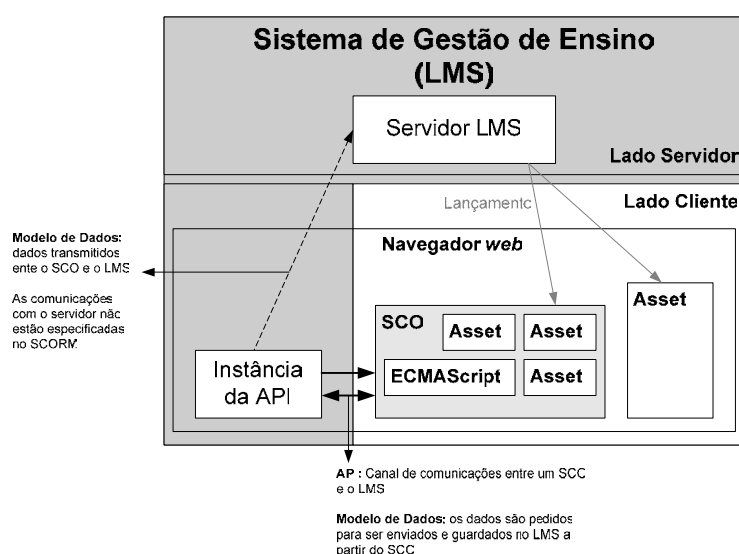


Figura 4-15 Ambiente de execução conceptual do SCORM

Os objectivos de reutilização e interoperabilidade de conteúdos que norteiam o SCORM são conseguidos com a definição de uma forma comum de lançar e gerir conteúdos, com um mecanismo comum de comunicação entre os conteúdos e os LMSs e com uma linguagem predefinida como base dessa comunicação. A figura 4-15 mostra os três aspectos principais do RTE: lançamento, interface de programação de aplicações (API) e modelo de dados.

O processo de lançamento define uma forma comum para os LMSs iniciarem conteúdos baseados na Internet, ou seja, define procedimentos e responsabilidades para o estabelecimento de comunicações entre o conteúdo lançado e o LMS. O mecanismo de comunicações é normalizado através de um API comum.

O API é o mecanismo de comunicação que informa o LMS do estado da comunicação entre o LMS e o conteúdo (por exemplo: iniciada, terminada ou em condição de erro), e é usada para leitura e guarda de dados (por exemplo: resultados, limites de tempo, etc.) entre o LMS e o SCO.

O modelo de dados é constituído pelos elementos normalizados, usados para definir a informação a monitorizar por um SCO, como por exemplo o estado actual ou o resultado de um teste. Os elementos definidos são conhecidos pelo SCO e pelo LMS.

4.6 Conclusão

A criação exponencial de informação se não for acompanhada com a criação de normas para a descrição e a identificação causará, eventualmente, longos períodos de pesquisa para alguns temas ou a impossibilidade de encontrar os dados pretendidos. O uso de metadados é, portanto, essencial para permitir a automatização da identificação dos recursos existentes actualmente na *web*.

O SCORM define os metadados para um tipo específico de dados: dados educativos. A definição passa pela descrição dos conteúdos educativos e pela sua organização. A descrição fará toda a caracterização da informação como autor, data, estado, entre muitas outras. A organização define a forma dos dados se estruturarem e constituírem unidades educacionais. Estes metadados irão facilitar a pesquisa, identificação e reutilização dos conteúdos em diferentes sistemas.

Além da definição dos metadados o SCORM ainda define uma plataforma de execução para a comunicação entre os conteúdos e os sistemas de gestão de informação.

5. Especificação do sistema *inVídeo*

O presente capítulo contém todos os aspectos da especificação do sistema *inVídeo*, desde a ideia inicial até à sua arquitectura conceptual.

A primeira secção introduz as necessidades que existiam no Centro Hospitalar do Funchal e que justificaram este sistema. Segue-se a apresentação do método de desenvolvimento usado para a especificação do sistema, o WISDOM.

De seguida inicia-se a especificação com a descrição dos requisitos encontrados para o sistema e a apresentação de diversos diagramas e modelos, nomeadamente, casos de utilização, diagramas de actividades, modelo de análise, arquitectura conceptual, diagrama de classes e modelo de apresentação. Por último é apresentado o refinamento de alguns espaços de interacção.

5.1 Introdução

O sistema especificado neste capítulo surgiu das necessidades de formação existentes numa instituição hospitalar, o Centro Hospitalar do Funchal, mais especificamente dos deveres do Núcleo de Apoio aos Centros de Responsabilidade. Estes constituem um nível intermédio de gestão na área da saúde, baseados no processo assistencial, agrupando serviços e/ou unidades funcionais, segundo critérios de homogeneidade, afinidade e complementaridade. Têm como objectivos melhorar o acesso e a qualidade de atendimento e dos cuidados prestados, como resultado de um melhor desempenho dos profissionais e de uma maior eficiência na utilização dos recursos humanos e materiais.

Foi no âmbito deste último objectivo que surgiu o enquadramento para o sistema *inVÍdeo*. Uma das formas de proporcionar o incremento das competências dos colaboradores passa por um sistema de ensino/formação à distância, onde o vídeo surge como meio educacional privilegiado. Foi baseado neste princípio que surgiu a designação do sistema.

O sistema desenvolvido, como conceito geral, visa proporcionar formação à distância, utilizando preferencialmente o vídeo e deve ter uma estrutura educacional simples baseada em cursos, que por sua vez são organizados em módulos e sessões. As sessões podem ser das seguintes categorias: lição, treino e avaliação.

5.2 O método WISDOM

O sistema *inVÍdeo* será desenvolvido e mantido, previsivelmente, por uma pessoa ou por um grupo pequeno de pessoas, e para estas era necessário um método que garantisse que o trabalho desenvolvido estivesse assente num processo e num conjunto de modelos bem definidos, para obter um produto final de qualidade, e cujo tempo de desenvolvimento não fosse muito penalizado por especificações exigentes e demoradas. Sendo o grupo de trabalho pequeno, o método a usar deverá ser adequado às exigências do projecto, mas mantendo uma leveza que permita um desenvolvimento rápido. O método de apoio escolhido para o desenvolvimento do sistema *inVÍdeo* foi o *Whitewater Interactive System Development with Object Models* (WISDOM). O WISDOM foi desenvolvido com o objectivo de minorar os principais problemas e dificuldades das pequenas empresas de desenvolvimento de *software*. Estes factores negativos são: o desenvolvimento caótico; o uso de processos proprietários, normalmente pouco maduros; e uma postura muito vincada em rapidamente apresentar resultados da fase de implementação, minorando as fases de análise e desenvolvimento da arquitectura. O WISDOM apresenta um método a ser usado por pequenas equipas de desenvolvimento que projectam, criam e mantêm sistemas interactivos, e procuram manter um grau elevado de qualidade [Nunes 01a]. Além da possível adequação do WISDOM a este projecto, também existe a vantagem de termos próximo o autor do método, que se disponibilizou para apoiar a especificação. O Processo Unificado

[Jacobson 99] ainda foi ponderado devido à sua grande utilização e adequação em muitos ambientes, mas por falta de experiência nossa e de ferramentas licenciadas que o suportassem este método não foi usado.

O WISDOM é constituído pelos seguintes componentes:

- **processo:** é baseado num modelo centrado nos utilizadores e que caracteriza-se por ser baseado num modelo de protótipos evolutivos (figura 5-1). O desenvolvimento está adaptado especificamente para equipas pequenas e para ambientes onde se podem tirar as seguintes vantagens: boa comunicação interna inclusivamente com parceiros externos; flexibilidade e capacidade de iniciativa não condicionada de forma rígida por processos administrativos; facilidade na gestão de projectos e equipas; e capacidade de resposta rápida por parte das equipas de desenvolvimento e das equipas de gestão a novas condições e/ou circunstâncias imprevistas;

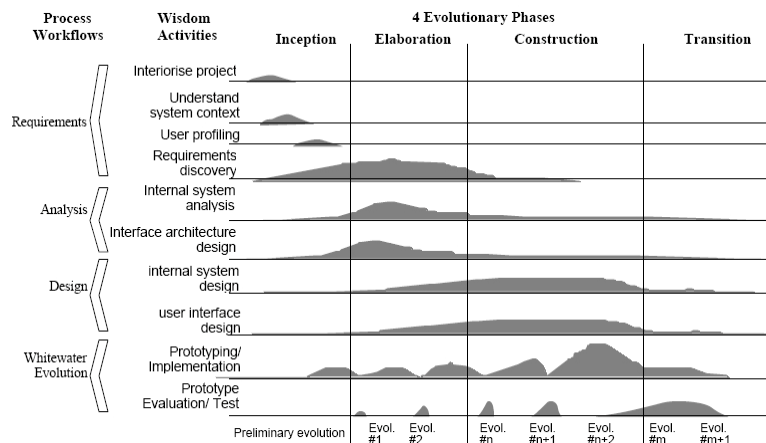


Figura 5-1 Processo WISDOM [Nunes 01b]

- **arquitectura:** é um conjunto de modelos UML (*Unified Modeling Language*) (figura 5-2) que suportam as diferentes fases de desenvolvimento envolvidas no processo WISDOM e são usados para a criação de sistemas interactivos em geral;

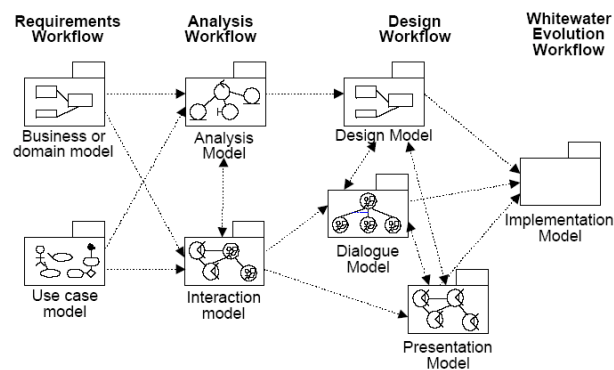


Figura 5-2 Organização dos modelos do WISDOM [Nunes 01b]

- **notação:** é um conjunto de notações baseadas num subconjunto do UML acrescido de extensões especificamente adaptadas para o desenvolvimento de sistemas interactivos.



Figura 5-3 Extensões WISDOM para o modelo de interacção

5.3 Descrição

A introdução deste capítulo referenciou alguns dados gerais sobre o sistema. Os parágrafos seguintes desta secção apresentam informação mais detalhada sobre quais as funcionalidades a especificar.

O sistema *inVÍdeo* tem como propósito principal transmitir vídeos formativos, sincronizados com apresentações (que podem conter textos, imagens e/ou animações) sobre a Internet. A estrutura da aplicação deverá ser baseada em cursos, que serão constituídos por módulos e estes últimos por sessões. A cada elemento referido na frase anterior deverá estar associada alguma informação, como: designação, descrição, objectivos, área, ou outra que se considere relevante.

As sessões, que é onde se concretizam os objectivos da aplicação, além da informação já referida, serão de um dos seguintes tipos:

- **lição:** sessão constituída basicamente por um vídeo em paralelo com uma apresentação, onde será possível interagir no decorrer da visualização usando comandos tipo leitor de vídeo doméstico. Outros elementos que poderão estar presentes são: o texto, sob a forma de legendas e de notas (públicas e privadas), e o(s) documento(s) de apoio à lição;
- **treino:** sessão onde serão executados treinos sobre diversos conteúdos. Os treinos poderão ser desde simples questionários até simulações interactivas complexas;
- **avaliação:** sessão onde será feita a avaliação de conhecimentos sobre um dado conteúdo. A avaliação poderá ser feita através de questões fechadas de resposta múltipla, cotadas automaticamente, e de questões abertas a serem cotadas por intervenção dos docentes.

Até ao momento, a explicação permitiu deduzir a existência de dois tipos de utilizadores: o aluno, que irá assistir aos cursos, e o docente, que os criará. Deverá ainda existir um terceiro tipo, o administrador. Ao aluno caberá a visualização e participação nos cursos, ao docente a responsabilidade de criar e manter os cursos e ao administrador a gestão das inscrições nos cursos e a criação e edição dos dados dos utilizadores.

A visualização dos cursos por parte dos alunos envolve a escolha das diversas sessões que o constituem e a realização das tarefas que estejam definidas. Na lição, o aluno apenas assiste à apresentação dos conteúdos, tendo a possibilidade de alguma interacção, mas no treino e na avaliação terá questões e tarefas com cotações associadas que serão usadas para cálculo do grau de

sucesso na realização dessa sessão. Aos alunos ainda é permitido executar pesquisas sobre os cursos em que estão inscritos e assistir de forma avulsa às lições relacionados com os tópicos sobre os quais tenham algum interesse.

A criação de um curso envolve a introdução dos dados relativos a esse curso e a associação dos módulos pretendidos. A criação de um módulo passa por escolher quais as sessões que o constituirão, além da introdução de informação complementar, como a descrição e os objectivos educacionais. A definição de uma sessão varia dependendo do tipo de sessão a criar. A sessão/lição terá a associação do vídeo e da apresentação, bem como dos dados para a respectiva sincronização, e de documentos auxiliares que serão disponibilizados para *download*. Poderão ainda existir legendas para o vídeo. A sessão/treino terá apenas a associação do ficheiro que contém as tarefas a praticar. A sessão/avaliação terá a definição do tipo de avaliação e a introdução das questões e, em alguns casos, das opções de resposta. Em qualquer um dos tipos de sessão haverá um registo das acções desenvolvidas (como por exemplo ser feito o registo da visualização da lição, da realização do treino e da avaliação com um certo grau de sucesso).

A responsabilidade dos administradores passa pela criação e edição dos utilizadores, pela gestão das inscrições dos alunos nos cursos e por resolver as dúvidas relacionadas com a utilização da aplicação colocadas pelos utilizadores.

A concretização do sistema que corresponde à descrição feita até ao momento está detalhada nas restantes secções deste capítulo utilizando alguns diagramas do método WISDOM (os diagramas apresentados são apenas um conjunto limitado de toda a especificação, mas escolhidos de forma a permitirem uma visão completa dos aspectos mais importantes da aplicação).

Para a fase de requisitos temos:

- o diagrama dos casos de utilização que visam a captura dos requisitos funcionais do sistema. Do conjunto de casos encontrados serão destacados aqueles que forem considerados essenciais;
- os diagramas de actividades, que mostram os passos básicos para a implementação de cada um dos casos de utilização. Serão apresentados os diagramas de actividades dos casos de utilização essenciais dos utilizadores aluno e docente.

Para a fase de análise temos:

- os diagramas de análise, onde é feita uma apresentação da visão interna do sistema para cada diagrama de actividades. Será apresentado o diagrama de análise do diagrama de actividades do 'Realizar Sessão'.
- a arquitectura conceptual do sistema, modelo criado a partir da exploração dos diagramas de análise. Este modelo visa apresentar uma visão geral do sistema a implementar.

Para a fase do desenho temos:

- o modelo de desenho, com a apresentação do diagrama de classes, que expõe as classes encontradas e definidas para a aplicação;
- o modelo de apresentação, com o diagrama de apresentação do utilizador aluno, onde é

exposta a estruturação dos espaços de interacção;

- os espaços de interacção refinados por meio do método de [Constatine 01], sendo feito o detalhe dos espaços de interacção ‘Realizar Lição’ e ‘Criar Lição’;

5.4 Casos de utilização

Nos casos de utilização foram definidas três categorias básicas de utilizadores: aluno, docente e administrador. Surge ainda a categoria Utilizador, mas esta apenas tem o significado de englobar o caso de utilização comum aos três utilizadores, 'Gerir Mensagem'. A figura 5-4 apresenta os casos de utilização encontrados para a aplicação.

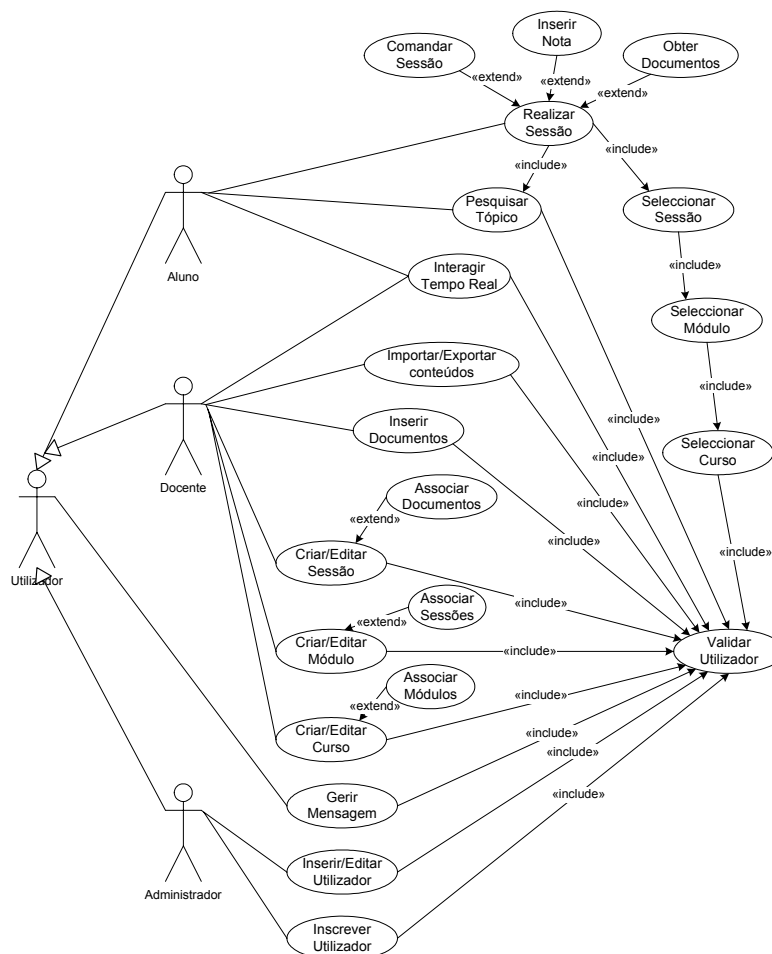


Figura 5-4 Casos de utilização da aplicação

O aluno tem como casos de utilização: o 'Realizar Sessão', o 'Pesquisar Tópico' e o 'Interagir Tempo Real'. O primeiro caso, essencial na aplicação, necessita, para ser efectuado, da passagem por um conjunto de casos de utilização, como a selecção de diferentes opções, e ainda da validação

do utilizador no sistema. Os casos de utilização até à realização da sessão não estão ligados a nenhum utilizador devido a não significarem por si só uma acção relevante para os objectivos do utilizador. Por exemplo, 'Seleccionar Curso' é um caso de utilização dessa categoria, que não cumpre nenhuma função essencial, mas é necessário para um fluxo de navegação lógico e estruturado. O 'Pesquisar Tópico' permitirá acesso às sessões relacionadas com o tópico escolhido. O 'Interagir Tempo Real', partilhado com o docente, permite conversação em tempo real com outros alunos e/ou docentes. Já referida para o primeiro caso, mas também presente para todos os restantes, é a necessidade de validação no sistema para poderem ser realizados todos os casos.

O docente tem a responsabilidade da criação e da manutenção dos cursos. Os casos de utilização reflectem essa tarefa, surgindo a criação/edição de cursos, de módulos e de sessões. A realização destes casos de utilização necessita da realização de outros casos, referidos por 'Associar...'. A lógica da criação de cursos passa pelo desenvolvimento independente de conteúdos, na sua unidade básica: o documento. A criação e edição de documentos estão fora dos objectivos desta aplicação, devendo ser realizadas noutras. O docente apenas fará o seu registo no sistema, da forma definida no caso de utilização 'Inserir Documentos'. Os documentos irão, então, constituir as sessões e para cada tipo de sessão haverá um conjunto possível de documentos. A agregação de um conjunto de sessões irá formar um módulo e a agregação de um conjunto de módulos formará um curso. O docente tem, ainda, a responsabilidade pela importação e exportação de conteúdos. Esta funcionalidade permitirá que os conteúdos sejam usados noutra sistema e permitirá receber conteúdos externos. Além dos princípios de compatibilidade e reutilização, também permitirá a criação e edição de cursos fora do sistema e sua posterior introdução e/ou actualização. A interacção em tempo real já foi explicada junto da descrição dos casos de utilização do aluno.

O administrador tem as seguintes responsabilidades: a gestão dos utilizadores, que passa pela sua introdução e actualização; e a gestão das inscrições dos alunos nos cursos.

A todos os utilizadores, como já foi referido, é possível a troca de mensagens dentro do sistema, declarada no caso de utilização 'Gerir Mensagem'.

A cada utilizador pode-se relevar um caso de utilização, considerando-o como caso de utilização essencial. O significado deste conceito é dar ênfase a um caso de utilização relativamente aos restantes por se considerar fundamental para atingir os objectivos principais do utilizador. A boa implementação deste caso de utilização é decisiva para o sucesso do sistema. Assim, para o aluno considera-se o caso 'Realizar Sessão', para o docente 'Criar Sessão' e para o administrador 'Inscrever Utilizador'.

5.5 Diagramas de actividades

A cada caso de utilização está associado um conjunto de acções e passos, quer por parte do utilizador, quer por parte do sistema. As figuras 5-5 e 5-6 mostram os diagramas de actividade de dois casos de utilização essenciais, respectivamente, do aluno e o do docente. A sua escolha deve-se a conterem os aspectos fundamentais do projecto. Como nota final sobre o caso 'Realizar

Sessão', este será implementado no âmbito desta dissertação, mas apenas a componente referente à lição (ver secção 6.2).

O caso de utilização 'Realizar Sessão' é o caso mais importante de toda a aplicação. A figura 5-5 apresenta as actividades básicas para realizar o caso de utilização, mas necessitará em fases posteriores da definição mais concreta de algumas das suas acções.

O acesso à sessão será sempre antecedido da selecção de um conjunto de elementos. Inicia-se com a selecção do curso, depois do módulo e por fim da sessão. Para cada tipo de sessão serão usados elementos diferentes. A lição será basicamente a exibição conjunta de uma apresentação e de um vídeo. O treino poderá variar muito, visto que se baseia na busca de um ficheiro onde estará inserido o treino e que poderá apresentar um qualquer conjunto de tarefas. A avaliação será a resolução de questões fechadas (cujas cotações serão feitas automaticamente) e abertas (a serem cotadas por um docente), o que implica a existência de ecrãs diferentes para cada tipo de sessão. Os documentos que estarão associados, por exemplo, numa lição e num treino têm fins diferentes. Ainda antes da realização da sessão será feito o registo do acesso para efeitos de controlo.

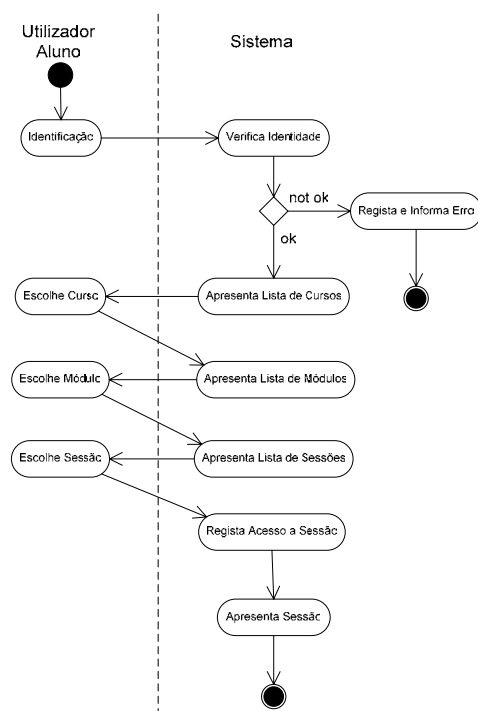


Figura 5-5 Diagrama de actividade de 'Realizar Sessão'

O diagrama de actividades da figura 5-5 é apresentado no âmbito da captura dos requisitos, e ainda não são declarados aspectos de pormenor. Um exemplo é a actividade 'Apresentar Sessão': a definição das acções nesta fase deve ser simples, ficando para fases posteriores um incremento do detalhe na especificação.

O diagrama da figura 5-6 mostra os passos básicos para a construção de uma sessão. As actividades para a construção de cursos e módulos são semelhantes.

A actividade inicial passa pela identificação do utilizador no sistema (actividade repetida em todos os casos de utilização). Neste caso a actividade principal passa pela escolha dos documentos a associar em cada sessão, os quais, dependendo do tipo de sessão escolhida, podem variar muito. Na nota da figura 5-5 são descritas as diferenças entre os diversos tipos de sessão, diferenças essas a nível dos documentos que cada uma terá associados.

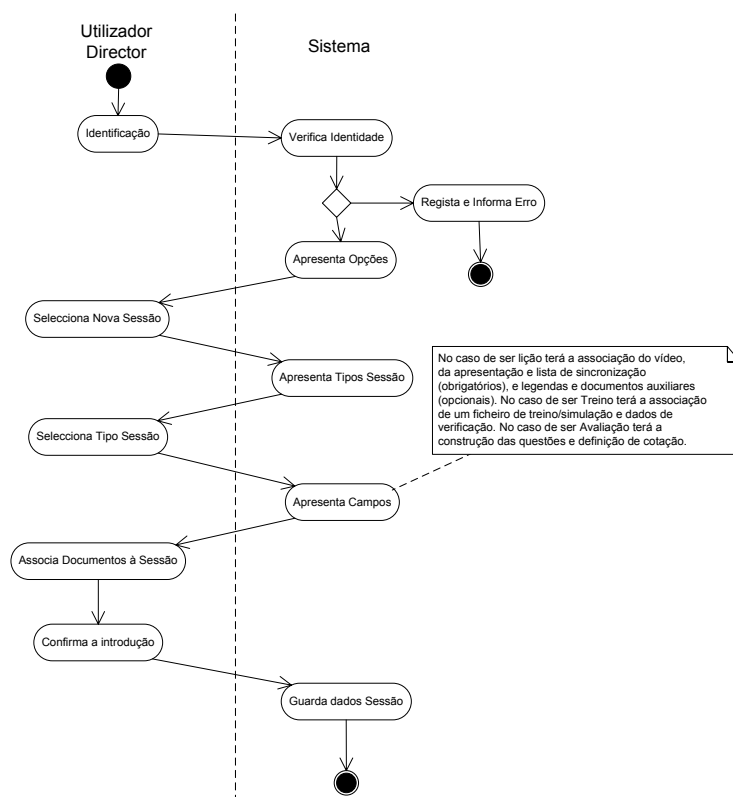


Figura 5-6 Diagrama de actividades de 'Criar Sessão'

5.6 Diagrama de análise

Os diagramas mostrados anteriormente (casos de utilização e diagramas de actividades) apresentam uma visão externa do sistema em termos das suas funcionalidades, ou seja, uma visão para ser entendida por todos os intervenientes na definição e construção da aplicação. As notações usadas até ao momento permitem que os esquemas criados possam ser compreendidos não apenas pela equipa de desenvolvimento mas também pelo cliente. Os diagramas de análise iniciam uma exploração às características internas do sistema, indicando os elementos necessários para cumprir as funcionalidades definidas. Estes diagramas são feitos para serem entendidos e usados pela equipa de desenvolvimento.

O diagrama de análise apresentado na figura 5-7 mostra a análise feita ao caso de utilização 'Realizar Sessão' por parte do utilizador aluno. Para cada actividade identificada anteriormente

para este caso são encontrados, primeiro, os espaços de interacção e as tarefas que devem ser cumpridas, e depois, a análise, onde se definem os controlos necessários e as entidades a que estão associadas.

A actividade do sistema 'Verifica Identidade' é a continuação da actividade 'Identificação' do utilizador e é realizada com a participação dos seguintes elementos: o espaço de interacção 'Acesso', a tarefa 'Inserir Identificação', o controlo 'Verifica Identidade' e a entidade 'Utilizador'.

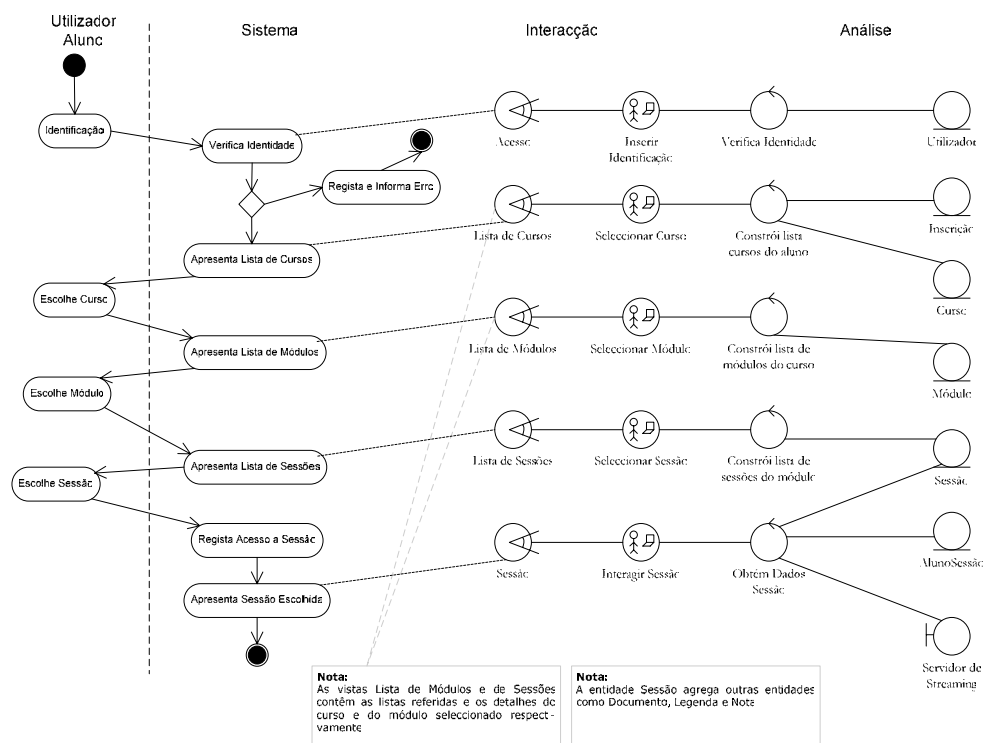


Figura 5-7 Diagrama de análise do caso de utilização 'Realizar Sessão'

O espaço de interacção 'Acesso' e a tarefa 'Inserir Identificação' são onde se concretiza a troca da informação entre o sistema e o utilizador (neste caso a introdução dos dados de identificação por parte do utilizador e a apresentação da informação do resultado da verificação por parte do sistema). O controlo 'Verifica Identidade' conterá as definições para realizar a actividade dentro do sistema (neste caso comparar os dados recebidos com os dados existentes e devolver o resultado da comparação), ou seja as definições para a coordenação, sequência, transacção e controlo de outros objectos podem ser designados como a lógica do negócio [Nunes 01a]. A entidade 'Utilizador' contém os dados relativos ao utilizador, que deverão ser mantidos e guardados.

A estrutura deste diagrama inicia uma separação lógica de diferentes conceitos. O espaço de interacção e a tarefa são uma visão externa do sistema disponibilizada através da interface entre o utilizador e o sistema. O controlo contém todas as funções necessárias para a concretização das funcionalidades definidas e separa a interface dos dados. A entidade define uma primeira versão

simples do modelo de dados que será implementado. Temos portanto uma arquitectura de três camadas com interface, lógica e dados. A importância desta separação é principalmente facilitar a reutilização, a manutenção e o desenvolvimento para diferentes plataformas e tecnologias de interfaces com o utilizador [Nunes 01a].

5.7 Arquitectura conceptual do sistema

Comment [EM4]: Coloca um dúvida sobre a localização deste esquema, que explico no primeiro parágrafo. A explicação que dou é insuficiente? Ou acha que esta arquitectura deveria estar noutra secção que não a que eu refiro (após análise)?

A construção e a exploração dos diagramas construídos e apresentados em parte neste capítulo, permitiram chegar à definição de uma estrutura global onde os elementos principais foram declarados e relacionados, constituindo desta forma uma arquitectura conceptual do sistema.

A figura 5-8 apresenta uma arquitectura do sistema *inVÍdeo*, e nela temos identificados todos os elementos a implementar, tais como: os espaços de interacção e as tarefas que neles se realizam; os controlos que conterão todas as funções que definem a forma de acesso aos dados; e as entidades que conterão os dados de forma persistente.

Os espaços de interacção e as tarefas reflectem, principalmente, a relação com os requisitos iniciais, definidos pelos casos de utilização.

Os controlos, componentes fundamentais da implementação, foram definidos após a construção dos diagramas de análise (apresentado um na secção 5.6 deste capítulo). Em cada um destes diagramas foram definidos os controlos necessários para responder às tarefas definidas e depois agrupados por proximidade na relação com as entidades. Para simplificar o seu tratamento criaram-se vários controlos principais, cada um dos quais reunindo um conjunto de funções com algum tipo de relação lógica entre si. Os controlos definidos foram Mensagem, Utilizador, MensagemRT, Nota, Curso e ImpExp (com a sufixação *Handler*, que se pode traduzir por responsável, o que no caso do CursoHandler, por exemplo, significa controlo responsável pelos Cursos).

A partir do diagrama de classes (figura 5-9) podemos ter uma definição mais completa de todas as entidades e de todos os dados que existem no sistema. As entidades apresentadas na figura 5-8 são apenas um conjunto das mais importantes, e permitem-nos identificar os principais dados que irão existir no sistema.

A arquitectura apresentada não é um esquema exaustivo de todo o sistema, mas apenas da sua parte mais importante e apresenta uma visão global: dos intervenientes; dos principais espaços de interacção, tarefas, controlos e entidades; e dos seus relacionamentos.

Uma outra função fundamental da arquitectura conceptual do sistema é apoiar, no caso da implementação ser faseada, a escolha dos elementos a implementar numa dada fase. Após a escolha das funcionalidades pretendida para uma dada fase, bastará desenvolver os espaços de interacção onde estas se concretizam, e as tarefas, os controlos e as entidades relacionadas.

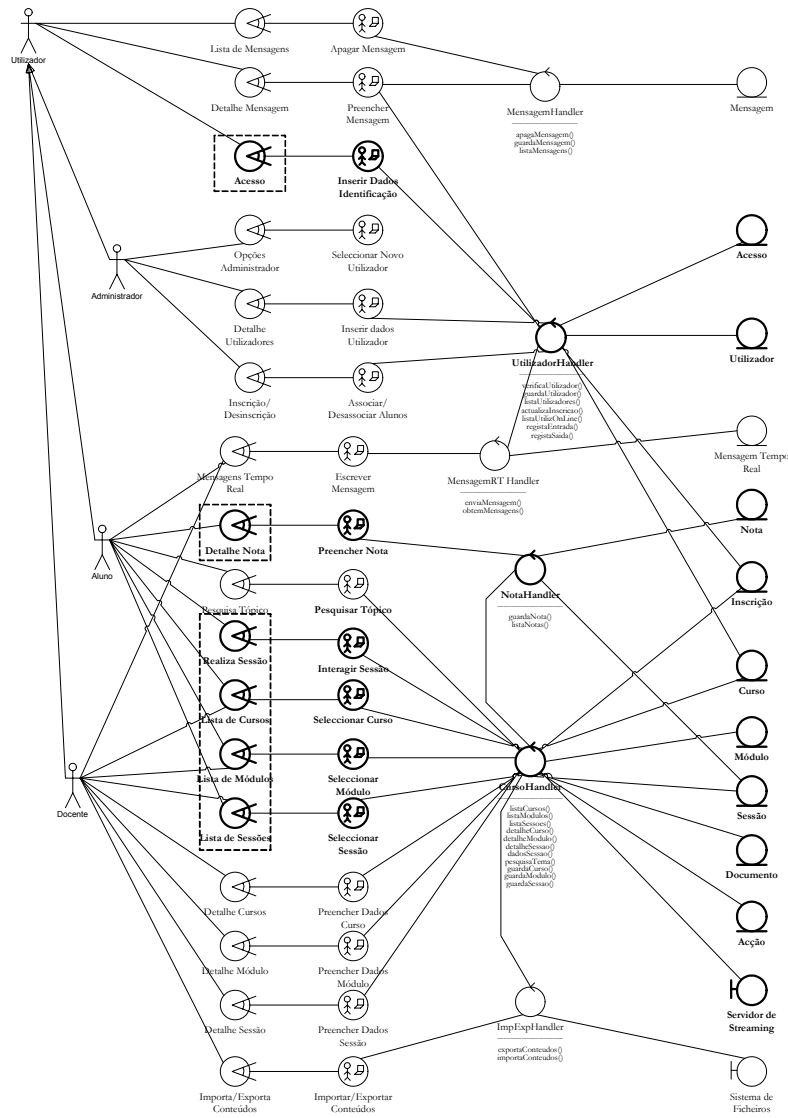


Figura 5-8 Arquitectura conceptual do sistema

5.8 Modelo de desenho

O diagrama escolhido para o modelo de desenho é o diagrama de classes. As classes encontradas são mostradas na figura 5-9 e definem as entidades da aplicação e os tipos de relações existentes entre elas. Surgem duas classes sem uma definição precisa, *Treino* e *Avaliação*, porque a implementação de ambas sai do âmbito deste trabalho. Os tipos de dados de cada atributo estão definidos no modelo de dados do sistema.

A análise ao diagrama de classes releva algumas divisões definidas para o sistema. A unidade mais simples é a relacionada com os utilizadores, conjunto composto pelas classes *Acesso*,

Utilizador, Aluno, Docente e Administrador, e é onde estão todos os dados relativos aos utilizadores e aos registos de utilização do sistema. Uma segunda unidade é a relativa aos documentos, isto é, aos conteúdos pedagógicos, com as classes: Documento, Vídeo e Apresentação. Aparecem ainda algumas classes necessárias mas menos importantes tais como: Tipo (de documento), Legenda e Slide. A terceira unidade é a estruturação dos conteúdos com as classes Curso, Módulo, Sessão, Lição, Treino e Avaliação.

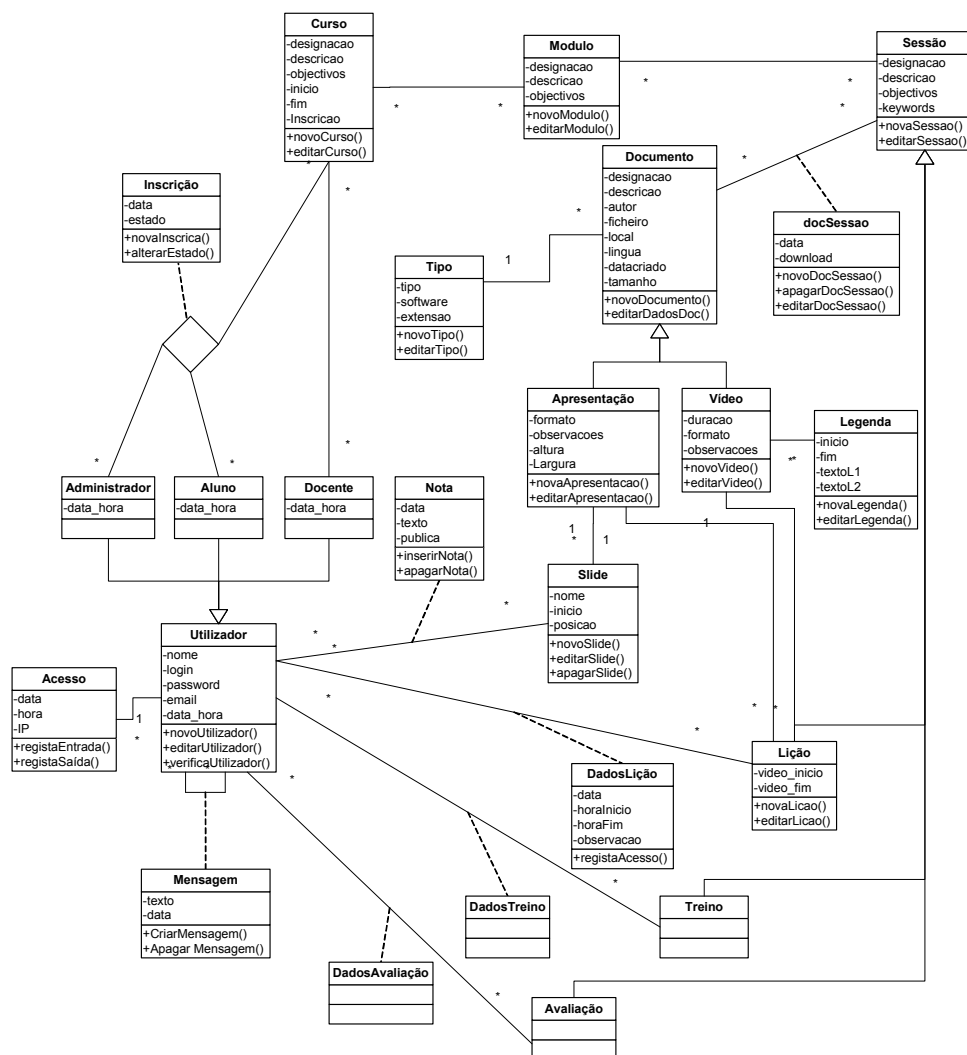


Figura 5-9 Diagrama de classes

A organização por unidades releva, assim, as relações não entre as classes de forma individual mas entre as diferentes unidades, designadas adiante por Utilizadores, Documentos e Cursos (no plural para diferirem das classes individuais, no singular). Surgem então a classe associação Inscrição e a relação Gere, como relações entre Cursos e Utilizadores, significando, respectivamente, a inscrição de alunos nos cursos, com a participação do administrador, e a definição dos docentes responsáveis pela criação e gestão dos cursos. Ainda entre as mesmas

unidades aparece o registo das sessões realizadas pelos alunos definidas nas classes associação DadosLição, DadosTreino e DadosAvaliação. Por fim, aparece a relação entre os Utilizadores e os Documentos com a classe associação Nota, que define os comentários por parte dos utilizadores aos *slides* das apresentações.

5.9 Modelo de apresentação

O propósito do modelo de apresentação do WISDOM é definir uma estrutura global dos espaços de interacção, ou seja, uma estrutura de entidades independentes da implementação que seja responsável pela “interacção física com o utilizador” [Nunes 01b].

O diagrama da figura 5-10 apresenta a estrutura referente ao utilizador aluno. As relações existentes entre as entidades, «navigates» e «contains», mostram, respectivamente, a navegação entre espaços de interacção existentes para o aluno, e quais os espaços de interacção contidos noutros.

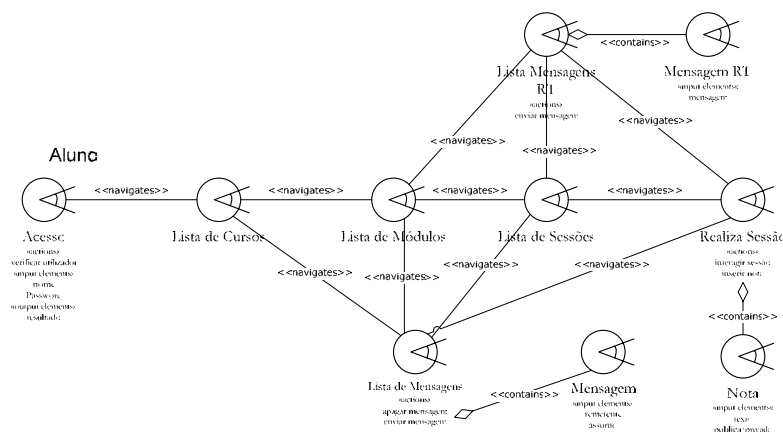


Figura 5-10 Diagrama de apresentação do aluno

Um exemplo de navegação entre espaços de interacção, é todo o percurso desde o ‘Acesso’ até ao ‘Realiza Sessão’, com passagem pelo ‘Lista de Cursos’, ‘Lista de Módulos’ e ‘Lista de Sessões’. Em cada espaço de interacção é possível navegar para a ‘Lista de Mensagens’ e para a ‘Lista de Mensagens RT’ (*Real Time*). O detalhe escolhido para este diagrama não apresenta todas as relações «contains» por razões de simplicidade. Estas relações surgem apenas para os espaços de interacção onde existe a introdução de dados, como ‘Mensagem’, ‘Mensagem RT’ e ‘Nota’. Na ‘Lista de Cursos’, por exemplo, estão contidos os espaços de interacção para apresentação de dados como a própria lista de cursos e o detalhe do curso que estiver seleccionado na lista.

5.10 Refinamento dos espaços de interacção

O modelo que deverá surgir paralelamente ao modelo de apresentação é o modelo de diálogo. A simplicidade da maioria das tarefas levou a que estes modelos não fossem apresentados neste documento, e optou-se por apresentar o detalhe dos espaços de interacção.

O método escolhido para modelar esse detalhe foi a utilização dos *Canonical Abstract Prototypes for Abstract Visual and Interaction Design* [Constantine 03]. Esta escolha surge de uma sugestão do autor do WISDOM, e após ser analisada revelou-se interessante e adequada para consolidar o detalhe de cada espaço de interação e a clarificação das tarefas que lá se realizam.

Um *Canonical Abstract Prototype* é uma extensão para o desenho centrado no utilizador e fornece um vocabulário formal para expressar desenho visual e interactivo, sem preocupações de aparência ou de comportamento.

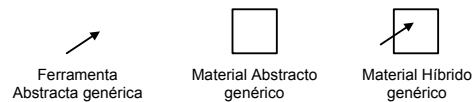


Figura 5-11 Componentes abstractos canónicos

O vocabulário é um conjunto normalizado e limitado de componentes abstractos que permitem um desenho fácil e rápido dos protótipos, que inclui as ferramentas, os materiais e os híbridos, também designados materiais activos (conjunto genérico na figura 5-11). A utilização destes componentes também facilita a comparação de diferentes modelos e o reconhecimento de padrões repetidos ou configurações comuns [Constantine 03].

Os diagramas de refinamento apresentados são os referentes aos espaços de interação 'Realizar Sessão' e 'Criar Sessão'. Em relação a 'Realizar Sessão', como foi referido anteriormente, uma sessão poderá ser do tipo lição, treino ou avaliação. Como o trabalho desta dissertação na sua fase de implementação, apenas se focou na lição, será apenas sobre componente do 'Realizar Sessão', isto é, o realizar de uma lição, que se apresentará com detalhe.

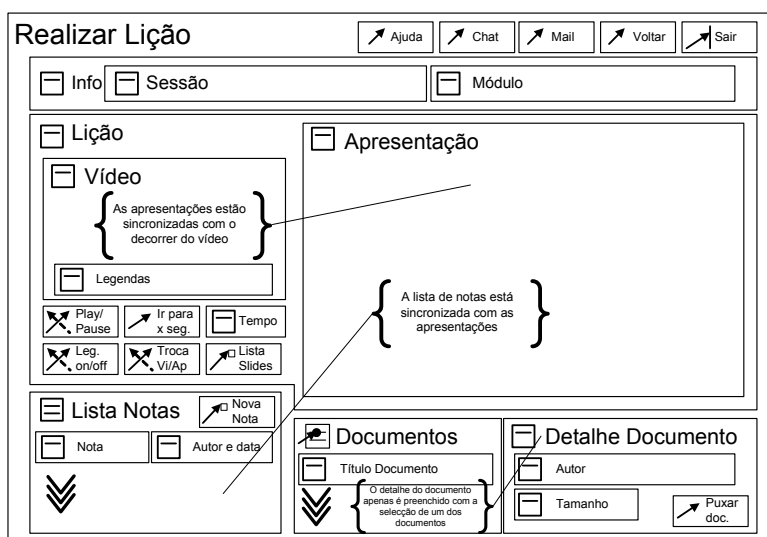


Figura 5-12 Refinamento do espaço de interação 'Realizar Lição'

Nesta fase, devido aos modelos criados até ao momento, já começa a existir uma ideia clara sobre os elementos que existirão em cada espaço de interação com o utilizador e as ações que este poderá executar em cada uma delas. O diagrama da figura 5-12 expõe os elementos que

deverão estar presentes numa lição. Assim definiram-se três áreas principais: a lição, a lista de notas e a lista de documentos. Surgem também mais alguns elementos como: designação da sessão e do módulo e acções diversas (como *Sair*, *Voltar*, *Mail*, *Chat* e *Ajuda*). Estes elementos são considerados gerais uma vez que surgem da análise conjunta de vários diagramas e são comuns aos vários diagramas.

A área da lição contempla os elementos vídeo, apresentação, notas e legendas, e acções diversas (interacção com a apresentação e com o vídeo, introdução de notas e *download* de documentos). A utilização do método de [Constantine 03] permite ao analista, como se pode verificar na figura 5-12, deixar algumas orientações para a composição do ecrã, definindo a estrutura global do espaço de interacção e o posicionamento relativo dos elementos. Um elemento que foi necessário avaliar foi a relação entre as dimensões do vídeo e da apresentação. O resultado apresentado não tem um valor definitivo, mas mostra a importância dada à apresentação, que tem assim um tamanho maior que o vídeo. Nas situações em que seja necessário dar ao vídeo um destaque maior, as posições e tamanhos são invertidos, graças à acção 'Troca Vi/Ap' (**V**ídeo/**A**presentação).

O diagrama da figura 5-12 apresenta, essencialmente, um conjunto de elementos e de acções simples. O diagrama 'Criar Lição', na figura 5-13, já contém uma maior variedade de elementos e de acções relacionadas com criar e apagar informação (o diagrama anterior apenas tinha uma acção para inserir notas). As acções gerais já encontradas na figura 5-12 estão repetidas.

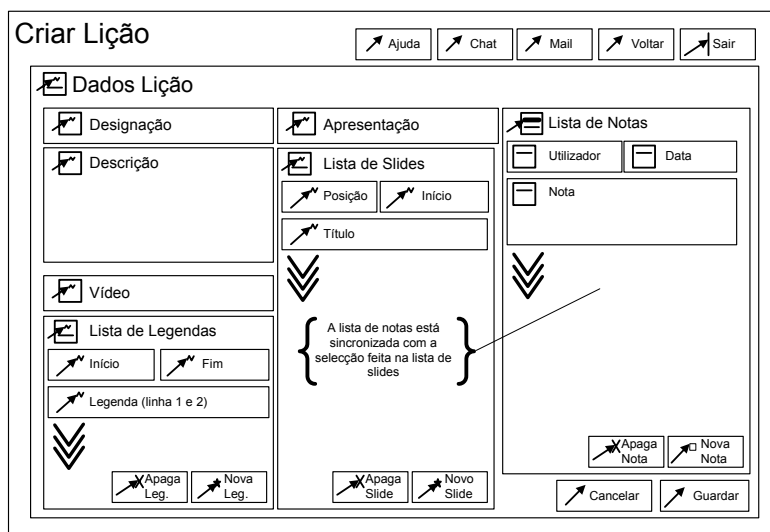


Figura 5-13 Refinamento do espaço de interacção 'Criar Lição'

Os elementos contidos no diagrama da figura 5-13 são os dados para a sessão/lição e todos aqueles que estão relacionados, como por exemplo o vídeo e a apresentação. A designação e descrição são dados da sessão, o vídeo e a apresentação já pertencem à lição. Por sua vez as legendas pertencem ao vídeo e os *slides* e as notas pertencem à apresentação. Estão também definidas as acções necessárias para fazer a edição de todos os dados.

5.11 Conclusão

O capítulo apresentado contém a especificação de uma aplicação interactiva utilizando a metodologia WISDOM. Os modelos WISDOM expostos foram escolhidos pela sua relevância para esta aplicação e para a descrição das funcionalidades mais importantes definidas para o sistema.

A detalhe do desenho dos modelos variou mediante a sua importância e/ou a necessidade revelada na altura da sua construção. Este foi um processo complexo e a opção por um detalhe maior ou menor, que por vezes pode implicar tempos de especificação demasiado prolongados, foi ponderada em cada modelo. Tentou-se detalhar com precisão onde tal era necessário e noutros casos apenas deixar indicações suficientes que permitissem uma compreensão alargada das ideias que se queriam expressar.

Os casos de utilização apresentaram as funcionalidades do sistema e a sua análise permitiu encontrar os objectivos principais da aplicação para cada utilizador através dos casos de utilização essenciais. Os casos de utilização foram depois complementados com os seus diagramas de actividades. O modelo de análise explorou a visão externa e interna do sistema do ponto de vista da equipa de desenvolvimento e permitiu identificar os elementos que seriam necessários implementar. De seguida apresentou-se uma arquitectura conceptual do sistema onde é possível ter uma visão geral, identificando os elementos e as suas relações. A construção da arquitectura conceptual também permite escolher de forma coerente e lógica o subconjunto de elementos a serem implementados numa determinada fase de um projecto. O diagrama de classes clarificou a informação a preservar e as suas relações. Por fim, o refinamento dos espaços de interacção e o modelo de apresentação caracterizaram, respectivamente, a interacção e a estrutura da navegação da aplicação.

6. Implementação do sistema *inVÍdeo*

O capítulo actual apresenta, para o sistema *inVÍdeo*: as principais opções tecnológicas no desenvolvimento; uma descrição de alguns elementos implementados; e os resultados de um pequeno inquérito aos utilizadores.

As opções tecnológicas dividem-se entre as diferentes ferramentas e linguagens utilizadas para desenvolvimento e a tecnologia utilizada na importação e exportação de conteúdos.

A descrição do sistema é feita através da apresentação de alguns dos seus ecrãs e de excertos de código de algumas funções, mostrando, em ambos os casos, a relação com a fase de especificação.

Por fim, apresentam-se os resultados de um pequeno inquérito feito aos utilizadores, mostrando o grau de satisfação na utilização da aplicação.

6.1 Opções tecnológicas

A escolha das tecnologias a usar para a implementação do sistema revelou-se um processo longo e, nalguns casos, complexo. A filosofia inicial para a implementação foi a de apenas utilizar *software* livre e normas não proprietárias. Tal não foi conseguido devido à impossibilidade de obter um conjunto completo de ferramentas, linguagens e normas enquadradas nessa categoria que garantisse os seguintes pontos: execução no maior número de plataformas e simplicidade na implementação e manutenção.

A avaliação das opções presentes nas diferentes categorias de *software* apresentou algumas dificuldades, devido à sua interdependência. Por questões de incompatibilidade a escolha não se resumiu simplesmente às melhores opções em cada uma das categorias.

O funcionamento da aplicação cliente em diferentes plataformas é um aspecto fundamental. Estudos indicam um aumento acentuado na diversidade de sistemas operativos [Legard 04], levando à utilização de ferramentas que permitam desenvolver aplicações que possam operar em diversos ambientes.

Outro factor de grande importância foi a nossa experiência na instalação e configuração dos servidores e no uso das ferramentas e linguagens de programação, de forma a garantir tempos mínimos de implementação.

A importação e exportação de dados, apesar de não ter sido implementada nesta fase de desenvolvimento, não foi descurada, tendo sido escolhida a norma que permitirá a utilização dos conteúdos deste sistema noutros sistemas e o inverso.

6.1.1 Ferramentas e linguagens

Apresentam-se de seguida as opções consideradas para cada uma das categorias de *software*, as respectivas vantagens e em alguns casos, por ser relevante, as desvantagens. No final são apresentadas as escolhas e feita a justificação das opções menos óbvias. As categorias são as seguintes: servidor de *streaming*, ferramenta para desenvolvimento do cliente, linguagem de interface com o servidor de base de dados, sistema operativo e motor de base de dados.

6.1.1.1 Opções

A escolha do servidor de *streaming* foi complexa, tendo envolvido quatro opções. Os servidores avaliados foram o *Helix DNA Server* da *Real Networks* [HDS], o *Darwin Straming Server* da *Apple* [DSS], o *Windows Media Services 9 Series* da *Microsoft* [WMS] e o *Flash Communication Server 1.5* da *Macromedia* [FCS]. As vantagens e as desvantagens principais dos diferentes servidores são apresentadas, resumidamente, na tabela 6-1.

As escolhas das linguagens de programação foram baseadas nas suas funcionalidades e, muito importante, na experiência da equipa de desenvolvimento. Foi ponderado o uso de PHP (*PHP Hypertext Pre-processor*) ou ASP (*Active Server Pages*) no lado do servidor e para o cliente o conjunto HTML/SMIL/JavaScript ou o *Macromedia Flash MX*.

Tabela 6-1 Comparação de servidores de *streaming*

Servidor	Vantagem	Desvantagem
<i>Helix DNA Server</i> da <i>Real Networks</i>	Maturidade e Qualidade	Custo do licenciamento
<i>Darwin Streaming Server</i> da <i>Apple</i>	Custo e flexibilidade	Pequena difusão do <i>player</i>
<i>Windows Media Series</i> da <i>Microsoft</i>	Difusão do <i>player</i> e do sistema operativo	Não permite uso noutra plataforma
<i>Flash Communication Server</i> da <i>Macromedia</i>	Difusão do <i>plug-in</i>	Qualidade dos codificadores e decodificadores

A apreciação entre o PHP e o ASP baseou-se numa maior ou menor experiência de programação na linguagem, tendo vantagem o PHP, enquanto que as facilidades de utilização e o apoio fornecido por parte das ferramentas de desenvolvimento deram vantagem ao ASP.

A ponderação entre o HTML/SMIL/JavaScript e o *Macromedia Flash MX* foi a de optar entre utilizar um conjunto de linguagens não proprietárias, tendo vantagem o trio HTML/SMIL/JavaScript, e utilizar um ambiente de desenvolvimento mais simples e integrado, aqui com vantagem para o *Macromedia Flash MX*.

As opções avaliadas para o sistema operativo foram o *Linux*, mais precisamente na sua distribuição *RedHat*, e o *Windows 2000 Server* da *Microsoft*. As vantagens do *Linux* reflectiam-se no custo, na fiabilidade de execução do servidor *web Apache*, e na integração simples de algumas das ferramentas e linguagens necessárias (base de dados, linguagem de programação e servidor de *streaming*). A grande vantagem do sistema operativo *Windows 2000 Server* da *Microsoft* é permitir a utilização do servidor de *media* da *Microsoft*, o *Windows Media Services 9 Series*, sem custos adicionais. A avaliação de robustez e confiança nos sistemas apenas foi baseada na nossa experiência, não havendo uma preferência clara por qualquer dos sistemas. Outras opções, como sistemas *Unix* e *MacOSX*, não foram avaliadas devido ao custo e requisitos de *hardware*.

Os motores de bases de dados considerados foram o *MySQL* e o *SQLServer* da *Microsoft* (MSSQL). As vantagens do *MySQL* encontram-se no custo e na simplicidade, enquanto que as do MSSQL foram o conjunto alargado de funcionalidades e robustez.

6.1.1.2 Ferramentas e linguagens escolhidas

O conjunto de ferramentas e linguagens escolhidas para o desenvolvimento e para ambiente de produção do sistema foram as seguintes:

- servidor de *streaming*: *Macromedia Flash Communication Server*;
- linguagem de programação do servidor: PHP;
- linguagem de programação do cliente: *Macromedia Flash MX* e *ActionScript*;
- sistema operativo: *Linux* (com servidor *web Apache*);
- motor de base de dados: *MySQL*.

A escolha mais importante recaiu no conjunto das ferramentas da empresa *Macromedia*, devido ao seu ambiente de desenvolvimento para a Internet ser estável e poderoso e o *software* cliente (ou *plug-in*) que suporta a aplicação cliente estar difundido pela maioria dos utilizadores e plataformas ligadas à Internet. Além destas vantagens, o servidor de *streaming* também é servidor de comunicações. Além do *streaming* de vídeo este servidor permite a implementação simplificada de interação de utilizadores em tempo real, tanto por mensagem de texto, como por áudio e videoconferência, através de um protocolo proprietário, o RTMP (*Real Time Messaging Protocol*) que funciona sobre o TCP/IP [Simmons 04]. Estas capacidades irão permitir no futuro aumentar as funcionalidades do sistema sem grandes dificuldades e sem alterações no servidor (por exemplo, a implementação de aulas em tempo real e a gravação automática das apresentações e a sua sincronização com os slides). A nível da aquisição e do licenciamento esta opção não é a mais económica, mas a nível da implementação este custo poderá ser diminuído visto que permitirá tempos de desenvolvimento mais curtos.

Inicialmente, a utilização do SMIL foi pensada para a apresentação das lições, ou seja, para a composição, sincronização e interação das cenas. Todavia, seria sempre necessário utilizar outro conjunto de ferramentas para implementar o resto da aplicação cliente. Após a análise do *Macromedia Flash* verificou-se que, além das vantagens já referidas no parágrafo anterior, também poderia realizar todas as funções do SMIL. Assim, por razões de simplicidade, foi tomada a opção por ter apenas *Macromedia Flash*. O SMIL, apesar de não ser utilizado neste ponto, será aproveitado na exportação dos dados, onde é necessário acompanhar os *media* com a informação da sua sincronização. Não sendo possível o envio de código *Flash* com a informação de sincronização, o uso de SMIL para transportar a composição e sincronização surge como uma solução simples e completa. Sendo o SMIL uma norma pública, permite a reutilização dos conteúdos criados no sistema *inVÍdeo* num outro ambiente que tenha, pelo menos, um navegador. O Anexo A apresenta um exemplo de uma lição codificada no formato SMIL e apresentada num *media player* (neste caso, o *real player* por apresentar menos problemas de compatibilidade).

O MPEG-4, por dificuldade em encontrar ferramentas e informação de desenvolvimento, não foi considerado, surgindo no capítulo três por ser uma norma pública importante e por conter na sua definição alguns conceitos relevantes para esta dissertação como a composição, sincronização e interação.

As restantes escolhas, *Linux/PHP/MySQL*, têm vantagens óbvias no custo, sendo todas elas licenciadas como *software* livre. Não menos importantes são também a sua fiabilidade, suporte e maturidade, garantidas pela sua generalizada implementação e para diferentes plataformas. O sistema operativo *Linux*, conjugado com o servidor *web Apache*, a linguagem PHP e o motor de base de dados *MySQL*, são uma das soluções mais usadas para a implementação de serviços sobre a Internet.

Ainda foram utilizados outros *softwares* auxiliares, cuja importância, se comparada com a dos já apresentados é pequena, mas que apoiaram de forma decisiva o desenvolvimento, facilitando o trabalho feito. Os *softwares* escolhidos foram todos da categoria *Open Source* e são os seguintes:

- SE|PY, para a escrita de código *ActionScript* [SEPY];

- *Reload* 1.2, para a construção de pacotes SCORM [Reload];
- *DBDesigner* 4, para modelação da base de dados e interacção com o motor de base de dados *MySQL* [DBDesigner];
- *LimSee* 2, para a construção de cenas multimédia no formato SMIL 1.0 e SMIL 2.0 [LimSee2];
- *Notepad*, para a codificação em PHP.

6.1.2 Norma para a importação e exportação

A componente de Importação/Exportação do sistema não foi implementada no âmbito desta dissertação. Apesar de muito importante, o interesse desta primeira fase de implementação focou-se no desenvolvimento da aplicação cliente (na função de visualização de uma lição).

A funcionalidade Importação/Exportação foi definida para cumprir os seguintes objectivos:

- reutilizar noutros sistemas os conteúdos construídos no *inVÍdeo*;
- receber conteúdos produzidos em aplicações externas ou em outros sistemas semelhantes.

Foi necessário procurar normas que pudessem descrever os dados que seriam produzidos pelo sistema, dados estes com características educacionais. Nas pesquisas feitas as normas encontradas foram as seguintes:

- *ADL Sharable Content Object Reference Model* (SCORM): esta norma visa facilitar a criação de conteúdos de ensino reutilizáveis, dentro de uma *framework* técnica comum para aprendizagem local ou sobre a Internet [SCORM Overview 04];
- *Dublin Core Metadata Initiative* (DCMI): esta é uma organização dedicada a promover a adopção de metadados normalizados e a desenvolver vocabulários de metadados especializados para descrever recursos, que, assim, permitam a existência de sistemas de procura de informação mais inteligentes [DCMI 04];
- *IEEE Learning Object Metadata* (LOM): esta norma visa a especificação da sintaxe e semântica de metadados para conteúdos educativos, definidos como atributos necessários para descrever de forma completa e adequada os recursos educacionais [LOM 04];

A opção recaiu sobre o SCORM da ADL por satisfazer de forma mais completa os requisitos, ou seja, a possibilidade de descrever e reutilizar conteúdos educativos. Este último aspecto, a reutilização, é muito importante por questões de produtividade. Os conteúdos produzidos neste sistema podem ser usados noutros sistemas que sigam a norma SCORM. Também será possível a importação de conteúdos externos, desde que estes tenham sido gerados respeitando o SCORM.

A especificação SCORM integra na sua definição o IEEE LOM, logo é uma opção mais abrangente e completa que este último. Sobre o DCMI, a vantagem do SCORM esteve na particularidade de estar mais vocacionada para conteúdos educacionais.

O capítulo quatro detalha em pormenor as diversas componentes que formam o SCORM e, em particular, a Agregação de Conteúdos (secção 4.4), que trata a organização e detalhe dos

recursos educacionais, área mais relevante no âmbito desta dissertação.

Um contributo importante que o estudo dos metadados para conteúdos educacionais trouxe para o desenvolvimento deste sistema foi o entendimento mais profundo das ligações que podem existir entre recursos educativos. O princípio da reutilização presente na actual aplicação apenas foi possível devido à exploração e conhecimento da norma SCORM. Nas versões iniciais da análise a estrutura educacional era semelhante ao resultado final, ou seja, tínhamos cursos constituídos por módulos e estes por secções, mas com uma limitação: poderiam ser reutilizados os recursos educacionais de forma individual (os documentos), mas as organizações onde esses recursos eram usados seriam sempre independentes, o que não permitia reutilizar as secções e os módulos. Por exemplo, podíamos ter um módulo introdutório a um qualquer tema, que podia fazer parte de diversos cursos sobre esse tema, sem termos de construir várias versões do módulo referido.

A estrutura de dados presente no diagrama de classes (secção 5.7) foi dividida em unidades, tendo sido feitos nessa secção o realce e a descrição das relações que existiam entre elas. Um aspecto muito importante também presente nessa divisão assenta numa interligação mais clara com o SCORM. As unidades definidas, então, foram: Cursos, Documentos e Utilizadores. O SCORM não define a transferência dos dados sobre os utilizadores, mas sim, de cursos e documentos. Assim, nesta estrutura é fácil identificar os Conteúdos Partilháveis e a Organização de Conteúdos (explicados na secção 4.3.1), respectivamente nas unidades Documentos e Cursos.

Apesar de não implementada a funcionalidade de importação/exportação, como referido no início desta secção, é apresentado no Anexo B um manifesto SCORM de uma possível exportação das apresentações de um curso presente na aplicação. Não foi desenvolvido o código que gere o manifesto e que cria o pacote de conteúdos, mas foram previstos os dados contendo a informação mínima para que, futuramente, tal seja possível implementar sem interferir no modelo de dados.

O manifesto SCORM apresentado na figura B-1 contém duas organizações para os conteúdos. A primeira com todas as apresentações dos diversos cursos e numa segunda organização apenas as apresentações referentes a um curso (LEI). Os recursos representam todas as apresentações com os seus materiais.

O ficheiro base é um documento SMIL, que contém toda a informação para a sincronização, a composição e a interacção, dos *media* da apresentação. Este ficheiro terá associado o vídeo, a apresentação (imagens) e outros documentos que porventura existam.

6.2 Apresentação do sistema

A presente secção visa mostrar o sistema desenvolvido, tendo por base as especificações do capítulo anterior. Por o tempo destinado a desenvolver um trabalho deste género apenas permitir algumas semanas de implementação, escolheu-se para esta componente do desenvolvimento uma parte relevante para os temas principais desta dissertação: o vídeo, a sua composição, sincronização e interactividade.

A opção de implementação para o tempo que durou esta dissertação recaiu sobre as componentes específicas do utilizador/aluno, e tendo-se escolhido os seguintes espaços de

interacção, identificados na figura 5-8 dentro de rectângulos com limite tracejado: ‘Realiza Sessão’, ‘Lista de Cursos’, ‘Lista de Módulos’, ‘Lista de Sessões’, ‘Detalhe Nota’ e ‘Acesso’. A escolha destes espaços de interacção levou à necessidade de implementar um conjunto mínimo de controlos e de entidades. Os controlos foram os seguintes: UtilizadorHandler, CursoHandler e NotaHandler. Os restantes controlos como não têm relação com estes espaços de interacção não foram implementados nesta fase. As entidades implementadas foram as relacionadas com os controlos escolhidos antes, ou seja, todas as entidades excepto as relacionadas com as mensagens. A figura 5-8 apresenta os elementos referidos a negrito e com uma linha mais espessa.

A apresentação dos componentes que constituem nesta fase o sistema *inVÍdeo* passará pelos ecrãs construídos para a aplicação cliente e alguns excertos do código também da aplicação cliente e das bibliotecas do servidor.

6.2.1 Diagrama de componentes

O título desta secção deveria ser Diagramas de implementação, e teria o objectivo de apresentar a estrutura dos componentes e do ambiente de execução, isto através dos diagramas de componentes e de instalação. O diagrama de componentes mostraria a organização e as dependências entre os componentes (que podem ser código fonte, executáveis, bibliotecas). O diagrama de instalação exporia a organização do *hardware* e da ligação do *software* ao *hardware*.

O diagrama de componentes apresentado na figura 6-1, por incluir também alguma informação sobre o *hardware* existente no sistema, é o único diagrama exposto nesta secção, dando-lhe o nome também.

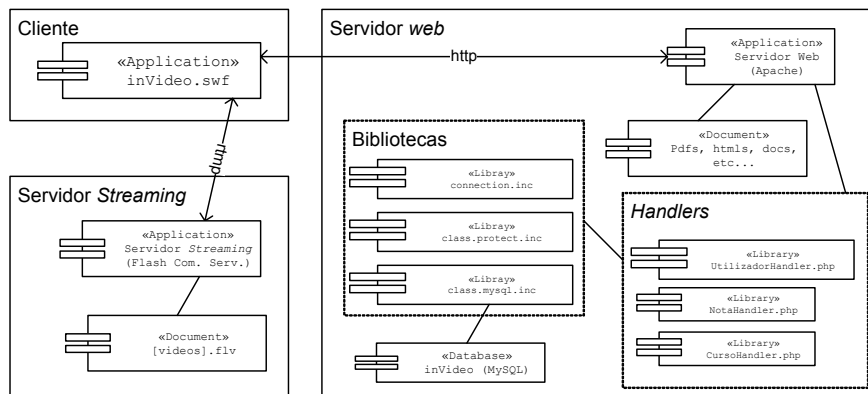


Figura 6-1 Diagrama de componentes do sistema *inVÍdeo*

A figura 6-1 mostra os três conjuntos de componentes do sistema: o cliente, o servidor de *streaming* e o servidor *web*. O cliente terá apenas um ficheiro, em *Flash*, que permitirá ao utilizador escolher e visualizar os conteúdos e que contém toda a parte da sincronização multimédia. O servidor *web* conterá a aplicação *Apache* (servidor de páginas) e o conjunto de documentos que constituem os *handlers* e as bibliotecas de apoio, principalmente aquela que faz a ligação à base de dados. Por fim, existe ainda o servidor de *streaming*, que disponibilizará os vídeos mediante os pedidos.

Os servidores *web* e de *streaming* podem ser integrados apenas num servidor único, uma vez que os primeiros usam normalmente a porta 80 e os segundos usam outras portas, que no caso do *Flash Communication Server* é a 1935 (mas é conveniente os servidores de *streaming* também poderem usar a porta 80 para evitar problemas de acesso a redes protegidas com *firewalls*). Esta opção traz a vantagem de reduzir o custo do equipamento servidor a comprar, pois só será necessário um. Mas a opção por uma máquina para todos os serviços também tem grandes desvantagens tais como: a eficiência do servidor, comprometida pela acumulação de serviços activos; e não permitir ao serviço de *streaming* o uso da porta 80 (ocupada pelo servidor de páginas), impedindo por vezes o acesso aos conteúdos por parte de utilizadores de redes cujas *firewalls* não permitam a troca de dados excepto em portas bem conhecidas, como é o caso da porta 80.

6.2.2 Ecrãs da aplicação cliente

Os ecrãs construídos para a aplicação cliente foram os seguintes: **acesso**, onde se faz a introdução dos dados de identificação do utilizador e se desencadeia a verificação, que, no caso de sucesso, permite aceder à restante aplicação; **lista de cursos**, onde se apresentam os cursos nos quais o utilizador está inscrito, permitindo a consulta do detalhe de cada um deles e o acesso aos módulos e sessões que o constituem; **lista de módulos e de sessões**, funcionalmente semelhante ao ecrã lista de cursos, mas neste caso com os módulos e as sessões; e **realiza sessão**, onde a visualização do vídeo e da apresentação acontece e onde existem as diferentes possibilidades de interacção.

De seguida apresentam-se as imagens das capturas dos ecrãs, enumeradas na lista anterior, de situações de utilização normal da aplicação por parte de um utilizador de teste.

6.2.2.1 Ecrã de acesso

O ecrã apresentado na figura 6-2 é o mais simples da aplicação, contendo poucos elementos e funções.

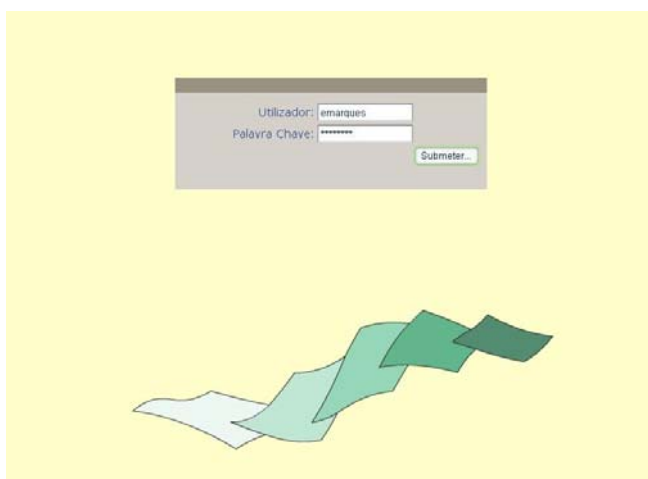


Figura 6-2 Ecrã de acesso à aplicação

A cada utilizador será atribuído um nome e uma palavra-chave, que lhe permitirá identificar-se perante o sistema. Este par de dados irá permitir ao utilizador aceder ao ecrã seguinte e a todos os ecrãs da aplicação definidos para a sua categoria de utilizador.

A análise à arquitectura conceptual, apresentada na figura 5-8, mostra que este ecrã implementa a vista ‘Acesso’ e a tarefa ‘Inserir Dados Identificação’. O controlo que contém os passos para a verificação é o ‘UtilizadorHandler’, e estes passos estão definidos através da função ‘verificaUtilizador’. Para este exemplo apenas são necessárias duas entidades, Utilizador e Acesso. A primeira para comparação dos dados e a segunda para registo do resultado e da data do acesso.

6.2.2.2 Ecrã dos cursos

O ecrã apresentado na figura 6-3 apresenta ao utilizador as seguintes informações: lista de cursos em que está inscrito, datas de início e de término dos cursos e tipo de inscrição¹. O utilizador poderá ainda consultar informações mais detalhadas sobre um curso, fazendo para isso a sua selecção na lista.



Figura 6-3 Ecrã com a lista de cursos

A relação da implementação deste ecrã com a arquitectura conceptual (figura 5-8) que o origina é de verificação simples. O espaço de interacção e a tarefa implementada são, respectivamente, ‘Lista de Cursos’ e ‘Seleccionar Curso’. Utiliza o controlo ‘CursoHandler’, nomeadamente as suas funções ‘listaCursos’ e ‘detalheCurso’, e as entidades Inscrição e Curso.

Surgem neste ponto elementos que estarão presentes em todos os ecrãs, como informação complementar e botões com funções diversas.

A informação complementar, no caso deste ecrã, é a identificação do utilizador e a data e hora do último registo de um acesso seu (possibilitando ao utilizador uma forma de confirmar que não existe utilização abusiva dos seus dados de acesso). Noutros ecrãs surge a identificação do curso ou do módulo e da sessão, conforme esteja, respectivamente, no ecrã de módulos e sessões

¹ que podem ser: livres e inscrição. Nos primeiros o acesso é livre a todos os utilizadores, para os segundos é necessário uma inscrição.

(figura 6-4) ou no ecrã da lição (figura 6-5).

Os botões terão algumas funções de navegação entre os ecrãs relacionados com o curso (curso, módulo e lição) e de acesso aos ecrãs das mensagens. Mediante o ecrã da aplicação em que o utilizador se encontre assim este terá mais ou menos botões. O botão mais à direita, sair (X), presente em todos os ecrãs excepto no 'Acesso', significa voltar ao ecrã de acesso e terminar a sessão. O botão voltar (<), presente na figura 6-4 e na figura 6-5, permite retornar ao ecrã anterior. Os dois botões seguintes permitem o acesso aos ecrãs das mensagens, não implementados nesta fase, mas já colocados nos ecrãs onde fazem mais sentido. E por fim o botão ajuda (?) activa/desactiva os textos que informam de forma curta os objectivos de cada botão.

6.2.2.3 Ecrã dos módulos e das sessões

A figura 6-4 apresenta o ecrã que contém os dados para a selecção do módulo que se pretende aceder e da sessão que se pretende realizar. O ecrã foi dividido na vertical em duas partes, existindo um espaço maior para a sessão, por conter dados mais relevantes e em maior número (como descrição, objectivos e duração da sessão). A lista de módulos contém os módulos do curso escolhido, e após a selecção de um deles podemos aceder a mais informações detalhadas sobre o módulo seleccionado. Esta acção desencadeia também o preenchimento da lista de sessões desse módulo. A forma de selecção e navegação desde o ecrã inicial (dos cursos) até à selecção da sessão a realizar mantém-se similar, facilitando a compreensão do funcionamento da aplicação.



Figura 6-4 Ecrã da lista de módulos e das sessões

Havia, inicialmente, dois ecrãs para esta informação, sendo cada ecrã semelhante ao dos cursos (figura 6-3), portanto separando o módulo e a sessão. Mas após a análise aos dados dos módulos e das sessões verificou-se que estes eram em número reduzido e que a passagem por três ecrãs para chegar à lição (ecrã fundamental da aplicação) era exagerado. Assim optou-se pela sua junção, simplificando a navegação e permitindo uma visão mais abrangente da estrutura do curso.

A opção inicial pelos dois ecrãs parte da definição dos espaços de interacção dos módulos e das sessões presentes na arquitectura conceptual (figura 5-8). Mas a arquitectura define espaços de

interacção e não ecrãs, e na construção dos espaços de interacção foram detectadas as dificuldades e limitações dessa opção, tendo-se decidido fazer a agregação dos dois espaços de interacção, com o resultado apresentado na figura 6-3. O modelo de apresentação para o aluno, secção 5.9, mostra ainda a opção dos dois ecrãs na estrutura de navegação, mas tal não condiciona nem obriga à existência de dois ecrãs, apenas aponta para a forma da navegação.

A restante relação com a arquitectura conceptual é feita de forma similar à exposta no caso da lista de cursos. As vistas e as tarefas implementadas são, respectivamente, as seguintes: 'Lista de Módulos' e 'Lista de Sessões', e 'Seleccionar Módulo' e 'Seleccionar Sessão'. O controlo utilizado é o mesmo do ecrã dos cursos, o 'CursoHandler', mas as funções diferem, sendo utilizado neste caso as seguintes: a 'listaModulos', a 'detalheModulo', a 'listaSessoes' e a 'detalheSessao'. As entidades usadas foram o Módulo e a Sessão.

6.2.2.4 Ecrã da lição

A figura 6-5 apresenta o ecrã onde se realiza a lição, e mostra, claramente, quatro zonas. As duas mais importantes, na parte superior do ecrã, são o vídeo e a apresentação. Na parte inferior encontram-se as notas e os documentos de apoio. As acções, ou botões, presentes encontram-se todas junto ao vídeo e estão basicamente agrupadas em três categorias: vídeo, lição e notas (neste último grupo apenas existe um botão para a introdução de notas).



Figura 6-5 Ecrã da realização de uma lição

As dimensões do vídeo e da apresentação foram ponderadas de forma a dar ênfase à informação visual mais importante, o que acontece com a apresentação. O ambiente de utilização base sobre o qual foi feita a análise é o orador a descrever um tema cujos tópicos e/ou descrições estão na apresentação. Mas no caso de existir uma lição que contenha um vídeo cuja visualização seja importante e a apresentação secundária, estes elementos podem ser trocados durante a visualização.

As notas estão sincronizadas com as apresentações, isto é, a informação de cada nota reflecte um comentário sobre a apresentação que estiver no momento no ecrã. Os documentos são simples

ficheiros que podem ser descarregados para o computador do utilizador, e são os documentos que o docente, que definiu a sessão, seleccionou.

Os comandos do vídeo são os botões normais de um objecto deste tipo. Têm a funcionalidade de avançar/pausar a imagem, a possibilidade de definir a intensidade do áudio e uma barra de navegação que permite passar directamente para um momento específico do vídeo. O grupo de três botões entre as notas e o vídeo têm a função de personalizar a composição dos elementos, mas restritos a algumas possibilidades. O botão mais à esquerda permite desactivar/activar as legendas, quando existem. O botão do meio permite trocar o vídeo com a apresentação. E o botão mais à direita dá acesso à lista das apresentações (não presente na figura 6-5). A selecção de um item dessa lista permite que a visualização “salte” para esse momento.

A entrada neste ecrã desencadeia a execução do vídeo (por omissão), e os elementos apresentação e notas vão sendo alterados continuamente. Podemos em qualquer altura da visualização usar um dos botões descritos. A introdução das notas é feita numa janela independente que surge integrada no ecrã e terá uma caixa de selecção para optar por manter a nota privada (opção por omissão) ou pública. Por fim, neste ecrã, existe a opção que permite descarregar os documentos que tenham sido introduzidos pelo docente que criou e/ou gere a sessão.

A análise comparativa deste ecrã com a fase da especificação será feita de duas formas, a primeira semelhante à que foi usada para os outros ecrãs e uma segunda forma por análise ao espaço de interacção apresentado na figura 5-11.

Na primeira forma, a análise com a arquitectura conceptual mostra que para definir este espaço de interacção, com algum simplismo, existem a vista ‘Realizar Sessão’ e a tarefa ‘Interagir Sessão’. Como foi descrito na secção 5.3 a sessão poderá ser de três tipos: lição, treino e avaliação. A especificação apenas foi definida até ao ponto da sessão e a especificação de cada uma das suas componentes apenas acontece em alguns modelos, como no espaço de interacção ‘Realizar Lição’, apresentado na figura 5-11. Os controlos usados neste caso são o ‘CursoHandler’ e o ‘NotaHandler’, e as funções usadas são, respectivamente, ‘dadosSessao’ e ‘listaNotas’, para carregar a sessão. Durante a visualização pode ainda ser invocado o ‘guardaNota’ do controlo ‘NotaHandler’. As entidades que participam na construção e durante a sessão são as seguintes: Documento, Sessão, Nota e o Servidor de *Streaming*.

O detalhe de cada modelo foi ponderado no momento do seu projecto (justificação na secção 5.11). No caso das entidades Documento e Sessão, na arquitectura conceptual (figura 5-8), estas não significaram uma entidade cada, mas um conjunto de entidades que nesse caso foram simplificadas. O conhecimento completo das entidades que constituem o Documento e Sessão pode ser obtido consultando o diagrama de classes do sistema, presente na secção 5.7 e na figura 5-8.

Quanto à segunda forma, o espaço de interacção da figura 5-10 e o ecrã apresentado na figura 6-5 são, respectivamente, a especificação e a implementação do modelo. A semelhança conceptual das figuras é grande e poderemos rapidamente identificar a forma com foi transposto o conceito para a realidade. O uso da notação de Constantine não pretende que os esquemas

reflectam na perfeição o interface a implementar, o que pretende é definir uma lista dos elementos que devem estar presentes no ecrã e deixar alguns apontamentos sobre a composição, mantendo desta maneira uma comunicação entre quem projecta e quem irá implementar.

6.2.3 Excertos do código

As linguagens escolhidas para a implementação do sistema foram várias. Para o desenvolvimento da aplicação cliente foi usado o ambiente gráfico *Macromedia Flash MX (Flash)*, que usa a linguagem *ActionScript* (baseada na norma ECMA-262) para a implementação de funcionalidades, principalmente as não gráficas. Para o servidor foi escolhida a linguagem *PHP Hypertext Preprocessor (PHP)* para a implementação das bibliotecas. Foi ainda usada a linguagem SQL para questionar o motor de base de dados *MySQL*.

O código mais relevante surge na implementação da sincronização, das comunicações e dos controlos. A implementação da sincronização é feita tendo como referência o vídeo, e agrega os *slides*, as legendas e as notas. A relevância do código das comunicações deve-se à implementação da ligação entre diferentes linguagens. O código dos controlos é importante porque implementa as funcionalidades do sistema definidas na especificação.

6.2.3.1 Implementação da sincronização

O código da sincronização, em *ActionScript*, foi introduzido na aplicação cliente e é utilizado apenas no ecrã da lição. É executado após a recepção dos dados da base de dados com a informação de sincronização para os diversos elementos, que são o vídeo, os slides (ou a animação), as legendas e as notas. O tempo de referência usado é o tempo do vídeo, e todo a animação dos diversos *media* acontece com o desenrolar do filme. Na entrada da lição o vídeo é iniciado imediatamente (por omissão), e é desencadeado um par de processos que numa base regular (segundo a segundo, porque é o detalhe mínimo do tempo do vídeo) actualiza os vários *media*. As linhas seguintes contêm a secção de código onde os processos são lançados:

```
...
if(GLB ar.slides.length>0){ // Lançar o processo que actualizará os slides e notas
    GLB int Apr = setInterval(actualizaApresentacao, 1000);
}
if(GLB ar.legendas.length>0){ // Lançar o processo que actualizará as legendas
    GLB int Leg = setInterval(actualizaLegendas, 1000)
}
...
```

Os procedimentos que actualizam os elementos obtêm o valor actual da posição do vídeo e mediante esse tempo definem qual o conteúdo a apresentar. O código seguinte apresenta a actualização dos *slides* e das notas (a razão do tratamento ser simultâneo é devido à relação existentes entre ambos, ou seja, uma nota está sempre associada a um *slide* – ver figura 5-8).

```
// Função que será chamada regularmente para actualizar a apresentação
function actualizaApresentacao(){
    var actualTime = myVideo md.playheadTime; //obter o tempo actual do vídeo
    var i, k;
    var vs_aux = "";
```

```

for (i=0;i<GLB ar slides.length;i++){ // Ciclo que percorre o array com os dados
    if (actualTime > GLB ar slides[i]["Inicio"]){ // Verificação do tempo actual
        GLB slide=GLB ar slides[i]["Posica"]; // e do tempo do slide
        GLB_ID_slide=GLB_ar_slides[i]["Sli_ID"];
        vs aux = "";
        for (k=0;k<GLB ar slides[i].length;k++){ // Ciclo para obter as notas
            vs_aux+= "<nome>"+GLB_ar_slides[i][k]["Autor"];
            vs_aux+= " (" +GLB ar slides[i][k]["Da Ho"]
                .substring(0,10)+"</nome>";
            vs_aux+= "<text>"+GLB_ar_slides[i][k]["Texto"]+"</text>";
        } // Texto das notas marcado como html e depois
        // é usada uma CSS para formatação
    }
}
if (vs notas slide.text != vs aux){ // Caso haja alteração nas notas
    vs notas slide.text = vs aux
}
myMovieClip mc.gotoAndStop(GLB slide); // comanda a apresentação a passar para um
} // determinado slide

```

6.2.3.2 Implementação das comunicações

O estabelecimento das comunicações foi dos aspectos mais complicados de implementar, por não existir uma forma directa do *Flash* questionar a base de dados; e, ainda, por factores relacionados com a maturidade da linguagem *ActionScript*. Apesar de largamente usada, o seu aparecimento é recente (2001) e muitos dos seus comandos têm sofrido alterações. A este facto associa-se a mudança do *plug-in Flash player* (forma de executar no navegador os ficheiros criados com o *Flash*), que não permite que alguns comandos mais antigos sejam executados em versões mais recentes do *player*.

A resolução da primeira limitação passou por introduzir uma linguagem intermédia que servisse de *interface* entre o *Flash* e o motor de base de dados *MySQL*. A linguagem escolhida para responder a essa necessidade foi o PHP. O PHP tem comandos nativos que permitem aceder aos dados existentes no *MySQL* (o desenvolvimento do PHP e *MySQL* está muito interligado, garantindo uma funcionalidade e eficiência muito grande). A comunicação entre o PHP e o *Flash* também é possível e foi executada por intermédio de texto simples sobre HTTP.

A implementação das comunicações do lado do cliente (*Flash*) com a base de dados teve de ser elaborada em dois passos:

- uma função que implementa a chamada do cliente para um ficheiro (PHP) localizado na máquina servidor da base de dados. O código seguinte apresenta um exemplo da implementação dessa chamada para o caso da verificação do utilizador, usada no ecrã 'Acesso'. A explicação está nos comentários ao código:

```

function verificaUtilizador(login,pass){
    var ar_param = new LoadVars(); // Novo objecto da classe LoadVars
    ar_param.onLoad = gereVerUtilizador; // Parâmetro que define qual função irá
    // gerir a chegada dos dados
    ar_param.metodo = "verificaUtilizador"; // Parâmetros a enviar para o PHP como
    ar_param.acesso = login; // qual o método a invocar, o login
    var enc_pass = new protect(pass); // e a password (cifrada)
    ar_param.chave = enc_pass.encrypt(); // Função sendAndLoad para o ficheiro PHP
    ar_param.sendAndLoad("utilizador handler.php",ar_param, "POST");
}

```

- uma função que fará a gestão dos dados recebidos, ficando à escuta dos dados que serão enviados do servidor. O código seguinte é um exemplo que complementa o código anterior, ou seja, recebe e trata os dados criados pelas funções PHP, e enviados pelo servidor. A explicação está nos comentários ao código:

```
function gereVerUtilizador() {
    vi result = this['resultado'];           // this refere-se ao array construído
                                           // com os dados enviados pelo servidor
    GLB nome = this['nome'];                 // Dados recebidos:
    GLB id = this['id'];                     // resultado: se acesso teve sucesso
    GLB entrada = this['entrada'];           // id e nome no caso de sucesso
    GLB ultAcesso = this['ult acesso'];       // ultimo acesso para informação
    if (vi result>0) {                       // Caso tenha sucesso avança
        gotoAndStop("ecra cursos", 1);      // para o ecrã de cursos
    } else {                                 // Senão é informado o erro e
        switch (vi_result) {                 // mantém-se no ecrã de acesso
            case "-1":
                txt avisos.text = "Utilizador Desconhecido!";
                break;
            case "-2":
                txt avisos.text = "Chave Errada!";
                break;
            default:
                txt avisos.text = "Erro de base de dados!";
                break;
        }
        txt pass.text = "";                  // Limpar as caixas de texto
        txt log.text = "";
    }
}
```

6.2.3.3 Implementação dos controlos

A linguagem escolhida para implementar os controlos foi o PHP. As funções de acesso aos dados e a forma como estes devem ser acedidos estão em ficheiros PHP, cada um representando um controlo definido na arquitectura conceptual (secção 5.10).

O código seguinte apresenta excertos da função 'verificaUtilizador', que faz parte do controlo 'UtilizadorHandler'. Os comentários explicam o funcionamento da função.

```
function verificaUtilizador ( $in_acesso, $in_chave ) {
    .....                               // Código para acesso à base de dados
                                           // Nova ligação à base de dados
    $mysql=new mysql($vs_host,$vs_user,$vs_pass,$vs_DB,0);
                                           // Questão da existência do utilizador
    $vs_sql="SELECT * FROM Utilizador WHERE acesso='".$in_acesso."'";
    $mysql->query($vs_sql);                // Executar a questão à base de dados
    $num=$mysql->num rows();                 // Número de registos encontrados
    .....                               // Dados para registo da entrada
    if ($num==0) {
        $rstring = "&resultado=-1";        // Devolver que o utilizador não existe
        registaEntrada(99999,-1,$vs_entrada,['.$in_acesso.']['.$in_chave.']);
    } else {
        .....                           // Utilizador existe, verificar password
                                           // Dados do utilizador (inclui password)
        if ( $vs_aux_chave != $in_chave ) { // Caso da Password Errada
            $vs_aux_result = -2; $rstring = "&resultado=$vs_aux_result";
        } else {                          // Caso da Password certa
            $rstring = "&resultado=4&id=$vs_aux_ID&nome=$vs_aux_nome&entrada=$vs_entr";
            $vs_aux_result = 4;
        }
        $rstring.="&ult_acesso=".ultimaEntrada($vs_aux_ID);
    }
}
```

```
registraEntrada($vs aux ID,$vs aux result,$vs entrada,'');  
}  
$mysql->end db(); // Terminar a ligação à base de dados  
echo $rstring; // Enviar a mensagem para o Flash  
}
```

Nesta função tem também o envio dos dados para o cliente *Flash* que assim completa o ciclo das comunicações iniciado com os excertos de código apresentados na secção 6.2.3.1. O texto construído e enviado será recebido no cliente sob a forma de um vector, do qual se poderá retirar a informação que se pretende. A *string* que resulta da linha “\$rstring=&resultado=4...”; do código anterior é enviado via HTTP e é interpretada no cliente *Flash* como um vector com os elementos ‘resultado’, ‘id’, ‘nome’ e ‘entrada’, como pode ser verificado no código sobre a gestão da recepção dos dados da secção 6.2.3.1, nas linhas onde aparece: `this['xpto']`.

No código anterior da função ‘verificaUtilizador’, algumas linhas foram omitidas por razões de espaço e de relevância. Enumeram-se de seguida os casos em que isso se verificou e qual o objectivo do código omitido:

- a obtenção dos dados de acesso à base de dados, por estarem definidos num ficheiro de parametrização, garantindo, assim, facilidade de configurar toda a aplicação para outro utilizador e/ou servidor;
- o registo da entrada, que guarda os dados de todos os acessos (quer tenham tido sucesso ou não);
- e a obtenção dos dados do utilizador que está a ser verificado, quer para validação da palavra chave (que será decifrada), quer para o envio dos dados do utilizador para o navegador e daí para a aplicação cliente.

Uma desvantagem na utilização deste conjunto de ferramentas é não existir um ambiente de desenvolvimento integrado, tornando por vezes o processo de *debug* (teste detalhado do código) bastante complexo e por vezes extremamente moroso. As linguagens escolhidas têm sintaxes semelhantes a tantas outras (como C, C++ e/ou *JavaScript*), o que permitirá uma adaptação rápida a programadores inexperientes.

6.3 Utilização da aplicação

O objectivo desta secção é apresentar alguns dados sobre a utilização do sistema. Para isso, recorreu-se à construção de um conjunto de lições e a um inquérito (presente no Anexo C juntamente com as respostas obtidas). Neste inquérito foi pedido a um conjunto de utilizadores que após a visualização das lições fornecessem informações sobre si, sobre os seus meios informáticos e, sobretudo, sobre diversos aspectos da sua experiência com o sistema.

Os conteúdos construídos basearam-se nas apresentações de um projecto prático da disciplina de Engenharia de Redes de Dados, presente em diversos cursos da Universidade da Madeira (UMa). A construção passou pela filmagem, produção e introdução das apresentações no

sistema e foram criados acessos para todos os alunos poderem rever o seu trabalho. Após a visualização foi pedido aos alunos que respondessem a um pequeno conjunto de questões inseridas num inquérito que procurava identificar os seguintes aspectos:

- as características técnicas mínimas para uma visualização aceitável;
- a experiência dos utilizadores na utilização de vídeo e áudio sobre a Internet;
- a qualidade da visualização do vídeo e da apresentação;
- a sua opinião sobre a relevância dos diferentes componentes da aplicação.

Para além dos alunos, pediu-se a um conjunto diverso de pessoas, de diferentes áreas, para visualizarem a aplicação e responderem ao inquérito. O domínio das pessoas que participaram nesta visualização foi restrito, o que determinou que a análise e apreciação dos resultados dos inquéritos fosse realizada tendo presente esta realidade. O domínio era constituído principalmente por pessoas da UMa: alunos, docentes e funcionários.

Os utilizadores criados no sistema, ou seja, as pessoas que participaram nesta avaliação foram trinta e quatro e, destas, onze responderam ao inquérito.

Nas figuras 6-6 B, C e D os resultados apresentam o número de respostas obtidas em cada categoria. Nas figuras 6-6 A e 6-7 os resultados reflectem o valor da soma das respostas obtidas, atribuindo a cada opção um valor entre -2 e 2 (o valor -2 foi para mau/pouco e 2 foi para bom/muito).

6.3.1 Utilizadores e características técnicas

A caracterização dos utilizadores (figura 6-6 A) revelou que, na sua quase totalidade, têm bons conhecimentos no uso dos computadores e da Internet. Também revelou que já têm alguma experiência no uso de vídeo e áudio sobre a Internet, mas muito menor, havendo mesmo casos cujo contacto era quase nulo.

O sistema operativo dominante foi o *Microsoft Windows*, havendo apenas um inquirido que acedeu com um sistema diferente, o *MacOS X*. Relativamente aos computadores, na sua generalidade, os utilizadores usaram na visualização máquinas com as seguintes características mínimas (figura 6-6 C e D): processadores de 500 a 1399 MHz, memórias superiores a 128MB e discos rígidos superiores a 40 GB. Mas na sua maioria tinham características superiores de processador e de memória.

O acesso à aplicação (figura 6-6 B) foi maioritariamente a partir da rede da UMa (100 Mbps) e de casa, usando tecnologias de banda larga (ADSL e rede Cabo). Apenas dois inquiridos usaram acessos diferentes: o telefónico e a RDIS (tendo identificado, respectivamente, as velocidades de 44 Kbps e 64 Kbps).

O *plug-in* necessário para a execução da aplicação estava instalado em todos os computadores e apenas dois inquiridos não tinham a última versão. Dos dois um fez a instalação e o outro não necessitou de proceder à actualização para poder executar a aplicação.

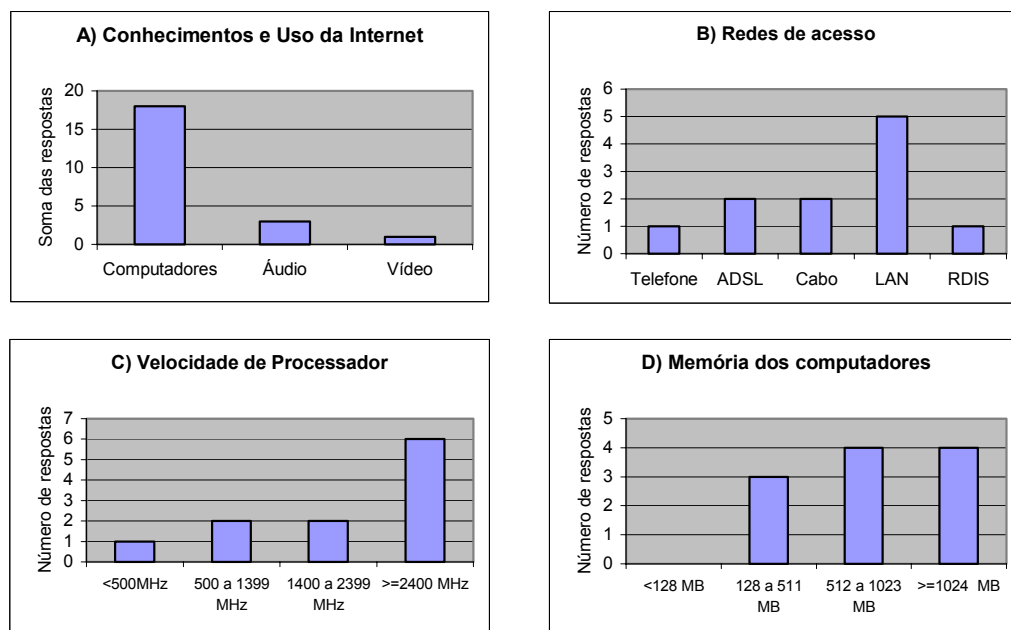


Figura 6-6 Conhecimentos dos utilizadores e características dos computadores e acessos

Os primeiros três pontos que se desejavam avaliar acabaram por estar parcialmente condicionados, uma vez que:

- a generalidade dos inquiridos são, como seria de esperar, experientes no uso do computador e Internet;
- acederam, na sua maioria, de uma rede interna a 100 Mbps;
- e com terminais cujas características são aceitáveis, ou seja, terminais com um ou dois anos no máximo.

Conhecer a opinião de pessoas com experiências e conhecimentos diversos (principalmente menores), com acessos mais fracos e computadores mais modestos, seria muito relevante, mas tal não foi possível. Os dados que se apresentam de seguida devem ser lidos tendo presente esta realidade.

6.3.2 Apreciação da utilização da aplicação

Os dados importantes foram divididos em duas categorias. Por um lado descobrir o tempo de resposta e a qualidade da visualização e, por outro, identificar os componentes mais relevantes.

Sobre a primeira categoria, e mais especificamente sobre o tempo de resposta, os resultados foram praticamente todos bons. Apenas uma resposta (o utilizador com o acesso telefónico) considerou médio tanto o tempo de resposta no aparecimento das listagens como do início da sessão. A qualidade da visualização teve resultados semelhantes, assim sendo, a qualidade da visualização do vídeo e da apresentação também foram consideradas genericamente boas, mas apresentando para o vídeo alguns resultados inferiores, nunca baixando da classificação média, ou seja, sempre positivos.

A segunda categoria, relevância dos diversos componentes, já apresentou resultados mais diversos. A maioria dos componentes teve resultados positivos, isto é, foram considerados como importantes nesta aplicação, e apenas diferiu o grau de importância que lhes foi atribuído.

Apresentam-se de seguida os resultados da análise à relevância dos componentes. O inquérito pedia aos inquiridos uma avaliação entre mau e bom. Relacionaram-se valores entre -2, para mau, e 2, para bom, e obteve-se os resultados apresentados na figura 6-7.

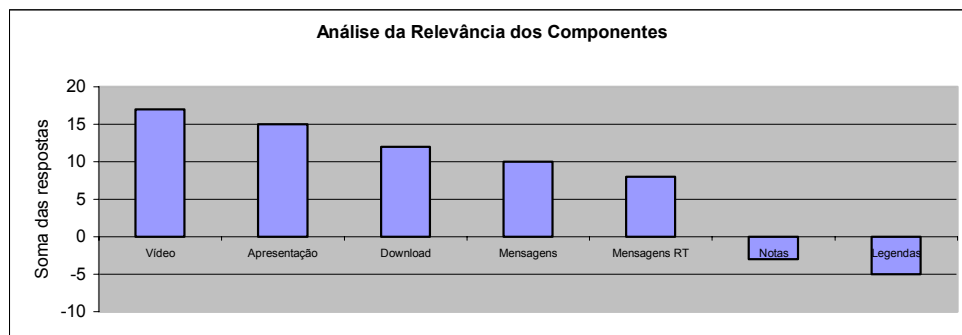


Figura 6-7 Gráfico com os valores obtidos por cada componente

Os componentes com resultados positivos foram os seguintes: vídeo, apresentação, *download* de documentos e mensagens. O vídeo e a apresentação foram os componentes com os melhores resultados enquanto que o *download* de documentos e as mensagens (*e-mail* e tempo real) obtiveram resultados ligeiramente inferiores. Os componentes com resultados negativos foram as notas e as legendas, tendo as últimas os piores resultados². Destes resultados deduz-se que estes elementos foram considerados pouco importantes para os utilizadores.

Alguns dados importantes retirados dos inquéritos partiram das sugestões e dos comentários. De entre as sugestões destacam-se propostas interessantes para novas funcionalidades que nunca estiveram previstas e que poderão ser consideradas em futuras implementações:

- avisos automáticos das alterações de conteúdos para os utilizadores;
- implementação de um fórum;
- visualizar o vídeo usando o ecrã inteiro.

6.3.3 Análise aos resultados do inquérito

Apesar das condicionantes já referidas, a opinião geral foi muito positiva, tanto na utilização como na classificação dos componentes e funcionalidades.

O acesso ao sistema a partir de redes de banda larga e de uma rede interna mostrou-se suficiente para as taxas de codificação usadas (máximo 300 Kbps). O acesso de tecnologias com larguras de banda menores aumenta os tempos de espera e as interrupções nos vídeos, degradando de forma considerável a qualidade da visualização.

² As legendas não foram introduzidas nos conteúdos produzidos para este teste, impossibilitando um contacto por parte dos utilizadores, ou seja, não puderam ver a forma como estas apareceriam na aplicação

Nenhum inquirido acedeu de computadores antigos, mas durante o desenvolvimento algumas experiências mostraram que máquinas com características de processamento e de memória abaixo, respectivamente, dos 500 MHz e dos 128 MB apresentavam algumas falhas na visualização e sincronização dos elementos. Todavia não podemos comprová-lo nestes inquéritos, mas ficou a garantia de que acima dos 500 MHz e 128 MB, no caso destes utilizadores, a aplicação funcionou sem grandes atrasos e problemas.

A notas e legendas deverão ser repensadas, por não colherem muitas opiniões positivas, comprovando-se também pelo pouco uso (apenas foram introduzidas algumas notas entre as dezenas de visualizações que foram efectuadas).

6.4 Conclusão

Este capítulo visou apresentar a concretização do sistema especificado no capítulo 5, iniciando-se com a fundamentação das escolhas tecnológicas para a sua implementação. Após a secção das opções tecnológicas apresentou-se o sistema desenvolvido, através do diagrama de componentes, de alguns ecrãs e de excertos de código. Terminou com a descrição e resultados de um inquérito feito a um conjunto de utilizadores que se disponibilizaram para testar a aplicação.

A escolha das tecnologias foi um processo moroso e complexo que obrigou a ponderar cada uma delas sob diferentes vertentes. As mais importantes foram as funcionalidades e a adequação de cada opção numa dada situação, seguidas do seu custo. O custo não se baseou apenas numa óptica da compra e/ou licenciamento, mas também no tempo que seria necessário para implementar o sistema com uma dada tecnologia, ou seja, nas nossas competências em cada ferramenta, linguagem e norma.

A apresentação do sistema foi feita recorrendo à descrição de alguns ecrãs e às funcionalidades que lhe estão associadas, relacionando-os com a especificação definida no capítulo anterior. Foram também apresentados excertos de código explicando como a interligação dos dados entre diferentes linguagens foi feita e, por fim, um diagrama com os componentes principais do sistema.

O inquérito realizado a um conjunto de utilizadores revelou o grau de satisfação (que foi positivo) na utilização do sistema. Ajudou também a detectar os elementos de maior e menor interesse, e, igualmente muito importante, identificou as características mínimas dos terminais dos utilizadores, que garantem um bom funcionamento da aplicação cliente.

7. Conclusões

O capítulo final apresenta as principais conclusões relativas ao desenvolvimento de aplicações multimédia, essencialmente aquelas em que o vídeo é utilizado como *media* preferencial.

Refere as facilidades técnicas existentes, actualmente, para o suporte de aplicações ricas, mas também mostra os limites que ainda condicionam a utilização dos *media* ricos como o áudio e o vídeo. Ainda refere a importância da utilização dos metadados em ambientes educacionais.

Apresenta algumas conclusões sobre o sistema especificado e desenvolvido, baseadas nas opções sobre os métodos, ferramentas e linguagens, e num inquérito aos utilizadores.

Termina por fazer referência a futuros desenvolvimentos, que passam, por um lado, pelo término da implementação do sistema e, por outro, pelo desenvolvimento de novas funcionalidades e novas aplicações.

7.1 Conclusões

A tarefa de desenvolver uma aplicação de vídeo sobre a Internet é um processo complicado que envolve muitos factores. Desde os modos de transmissão até à definição de acordos para o fornecimento de serviços de rede, os pormenores envolvidos são muitos e abrangem muitas áreas. Mas, nos dias de hoje, a utilização do vídeo já está bastante simplificada devido aos avanços tecnológicos em diversas áreas.

O *streaming*, codificação específica de dados e seu envio recorrendo a um servidor especializado, veio proporcionar novas formas de entrega e a difusão de novos conteúdos, como os eventos ao vivo e em tempo real e em simultâneo para muitos utilizadores, que não eram possíveis com o anterior modo, o *download*, nem com a sua versão melhorada, o *download* progressivo. O *streaming* além destas novas possibilidades veio também aumentar e reforçar os mecanismos de protecção dos direitos de autor dos conteúdos.

Por outro lado, a relação entre o tamanho dos ficheiros de vídeo e a sua qualidade tem vindo a alterar-se de forma positiva. Os novos métodos de compressão escaláveis, conjugados com novas formas de controlo de congestão e de erros, permitem que a qualidade perceptível na visualização dos vídeos seja cada vez maior. A utilização de serviços de distribuição de *media* contínuos pode melhorar a eficiência do *streaming* sobre a Internet com o uso de: filtros de rede, que podem mediar situações de congestão na rede; *multicast* a nível da aplicação, que permite a implementação do *multicast* sobre a Internet; e replicação de conteúdos, levando para mais próximo dos utilizadores os dados. A optimização do envio dos dados de vídeo pelo servidor de *streaming* assume também um papel importante na qualidade da apresentação do vídeo. As melhorias nas políticas de gestão dos sistemas operativos e dos sistemas de armazenamento que manipulam dados em tempo real, tornam cada vez mais eficiente o uso do *streaming*, evitando situações de congestionamento a nível do servidor.

O aproveitamento dos recursos da rede é cada vez mais eficaz, permitindo que num meio heterogéneo e que funciona como *best-effort*, como a Internet, possam passar dados com elevadas exigências sobre o atraso, variação do atraso e perdas. O uso de protocolos desenhados para o transporte e controlo de dados em tempo real – como o RTP, RTCP e RTSP – é uma forma de ultrapassar as dificuldades impostas pelas Internet.

O uso de arquitecturas mais complexas é por vezes necessário. Os serviços integrados permitem a reserva de recursos numa rede, garantido, assim, alguns requisitos mínimos para os dados em tempo real. As dificuldades dos serviços integrados com as redes de dimensão maior podem ser ultrapassadas com uma arquitectura de serviços diferenciados, onde os pacotes são marcados de uma forma específica possibilitando a alguns um tratamento preferencial na passagem por uma rede.

Se, e até agora, vimos as dificuldades em fazer chegar os dados com a melhor qualidade possível, outras dificuldades aparecem na utilização de vários fluxos de forma simultânea, ou dito de outra forma, na utilização de multimédia, ou seja, de dados com formatos e características

distintas. De entre vários aspectos, destacam-se os seguintes: a composição, a sincronização e a interacção multimédia.

A norma SMIL, por intermédio de um conjunto de elementos e atributos XML, visa definir uma linguagem declarativa que permite cumprir com os requisitos para a construção de aplicações multimédia, ou seja, conter definições para compor, sincronizar e permitir interacção com os *media*. Com um estilo tipo HTML, foi definida para ser uma linguagem simples, baseada em módulos e sobre os quais um construtor de aplicações poderá escolher as funcionalidades mais adequadas para um certo fim, recorrendo a um conjunto de perfis de implementação. Os variados grupos funcionais cobrem praticamente todas as necessidades para o desenvolvimento de ambientes multimédia ricos. Exemplos desses grupos são: a animação, o controlo do conteúdo, a ligação, as transições, e a temporização e sincronização, entre outros.

O MPEG-4, uma família de normas, surgiu após os esforços de normalização na área do vídeo digital, que resultaram no MPEG-1 e MPEG-2. Enquanto estes últimos tiveram como objectivos principais a definição de sistemas de vídeo específicos, o MPEG-4, com uma visão orientada ao objecto, visa normalizar os elementos tecnológicos da produção, da distribuição e da apresentação de conteúdos multimédia interactivos. Saímos da visualização de um vídeo rectangular para a integração de objectos multimédia no espaço e no tempo. O tratamento da composição das cenas é por intermédio de um formato binário designado por BIFS, cuja origem está no VRML. A sincronização básica é feita por marcação temporal dos fluxos, e existe um modo avançado, *FlexTime*, que permite sincronizar diversos fluxos de diferentes fontes. Esta última sincronização é conseguida com técnicas que permitem adaptar a duração dos *media*. A interacção do utilizador no MPEG-4 pode existir no lado do cliente e no servidor. No cliente podem ser alterados alguns parâmetros dos nós e objectos (como: tornar visível/invisível, trocar a cor, alterar o tamanho, etc.), dependem contudo dos equipamentos terminais; no servidor, a interacção, baseia-se na estrutura de eventos do VRML.

O SMIL tem como objectivo constituir uma linguagem completa para a descrição de cenas multimédia ricas. O MPEG-4 cobre muitos aspectos da utilização de ambientes multimédia, mas não é uma solução específica para a apresentação de cenas multimédia [Bulterman 01]. A relação entre estas normas surge no desenvolvimento coordenado de um formato textual, o XMT.

O crescimento na quantidade da informação torna fundamental a utilização de métodos que permitam a sua descrição e reutilização. Os metadados, como mecanismo que permite manter a descrição sobre os dados, são essenciais na caracterização e na identificação da informação. Além destes benefícios, permitem a implementação de pesquisas e localização sobre essa mesma informação.

O Modelo de Referência de Conteúdos Reutilizáveis (SCORM) é uma norma especificamente desenhada para gerir conteúdos educacionais, através da utilização da norma LOM do IEEE. Além da descrição completa dos conteúdos, a norma visa permitir a troca de conteúdos entre sistemas educacionais distintos, recorrendo à definição de uma forma de os empacotar (os conteúdos empacotados podem ser desde uma lição simples até um ou vários cursos completos). O SCORM ainda descreve a forma de sequenciar e navegar nos conteúdos. As regras de sequenciamento

baseiam-se na norma *Simple Sequencing* do IMS. Esta define uma forma de representar comportamentos pretendidos de um determinado conteúdo, de tal maneira que em diferentes LMSs a experiência de aprendizagem seja semelhante e consistente com a definida pelo autor. Além da definição dos metadados o SCORM ainda define uma plataforma de execução para a comunicação entre os conteúdos e os sistemas de gestão de informação.

A especificação do sistema *inVÍdeo* foi feita usando o método WISDOM [Nunes 01b]. Este permitiu chegar a uma arquitectura conceptual do sistema a implementar, definindo um modelo de dados, os controlos necessários e as interações com os utilizadores. Os esquemas apresentados não reflectem todo o trabalho feito, mas apenas os pontos essenciais e demonstrativos da especificação do sistema que se pretendia obter. Os casos de utilização mostram as funcionalidades que se pretendiam alcançar. A exploração das actividades identificou cada um dos componentes necessários. A partir da arquitectura obtida foi possível definir para a implementação quais os componentes a construir numa primeira fase. O modelo de apresentação e o refinamento dos espaços de interacção permitiram apoiar a definição da interacção com os utilizadores.

Ao WISDOM era pedido que fosse um método leve de utilizar, mas também que garantisse um bom nível de qualidade na especificação. Sendo o processo WISDOM baseado num modelo centrado nos utilizadores, rápido, evolutivo, e flexível a nível das alterações contínuas de requisitos, garantiu uma especificação completa e num tempo razoável, revelando-se, portanto, bastante adequado às características do projecto.

A implementação iniciou-se com a escolha das ferramentas e linguagens a utilizar. A dificuldade da escolha esteve na difícil integração de diferentes ferramentas e linguagens que poderiam ser as melhores opções nas suas áreas, e em obter um conjunto que permitisse um desenvolvimento mais simples e rápido. A análise de um grupo de ferramentas e linguagens, conjugadas de forma independente e em grupos, levou à escolha de um grupo delas para o desenvolvimento e para o ambiente de produção do sistema. As opções mais importantes recaíram sobre as ferramentas da *Macromedia*, devido ao seu ambiente de desenvolvimento e à facilidade de integração com o seu servidor de *streaming*. As restantes opções, como sistema operativo, linguagem de programação no servidor e motor de base de dados, resultaram de razões de integração, fiabilidade e custo.

Pretendia-se que o sistema *inVÍdeo* fosse uma ferramenta que permitisse uma aprendizagem à distância usando o vídeo como *media* central. A aplicação desenvolvida e o sistema implementado cumprem esse objectivo, permitindo assim que os utilizadores/alunos possam assistir a uma lição em vídeo acompanhada de informação complementar sob a forma de imagens ou animações, textos e do descarregamento de documentos auxiliares. Apesar de existirem algumas aplicações semelhantes (da *Macromedia* e da *Microsoft*, entre outras) para a apresentação de lições, não estão enquadradas em sistemas mais completos e apenas permitem a criação de apresentações 'avulsas'. O sistema *inVÍdeo* foi desenvolvido para se adequar a um ambiente onde o processo de autoria de lições fosse o mais simplificado possível, visto que o tratamento da componente de vídeo ainda é um processo pouco comum para a maioria dos utilizadores. Será ainda necessário algum esforço, como a codificação do vídeo para um formato adequado, mas para

a composição não é necessário fazer nada e para a sincronização apenas é necessário inserir os dados temporais para entrada e saída de um dado *media*.

A opção pela ferramenta *Macromedia Flash* para a maior parte do sistema trouxe facilidades no desenvolvimento e garantiu o cumprimento dos objectivos, mas a reutilização dos conteúdos ficou um pouco limitada, visto que a ferramenta não usa uma norma pública. O estudo de duas normas públicas, cada uma com um fim específico, permitiu definir o *inVÍdeo* de tal forma que futuramente os seus conteúdos possam ser usados em aplicações e sistemas mais difundidos.

A primeira norma pública, o SMIL, está adoptada, limitadamente, pela maioria dos *players*. Caso a compatibilidade da norma e dos múltiplos formatos de vídeo com os *players* se venha a concretizar, poderemos então ter a apresentação dos nossos conteúdos em qualquer um deles.

A segunda norma, o SCORM, permitirá que os conteúdos, entretanto construídos, sejam facilmente reutilizados em outras aplicações e o sistema possa receber conteúdos similares de outros sistemas. Com o incremento nas funcionalidades do sistema e com a completa implementação do SCORM, nomeadamente o Ambiente de Execução, será possível, de forma simples, o acesso e a partilha dos conteúdos em ambientes distribuídos.

Após o desenvolvimento do sistema foi construído e introduzido um curso e foi pedido aos seus utilizadores que respondessem a um pequeno inquérito para avaliar a experiência de visualização das diferentes lições. Este inquérito mostrou alguns aspectos fortes do sistema e outros que necessitam de ser repensados. A primeira análise definiu os requisitos técnicos mínimos para a utilização da aplicação a nível do terminal do utilizador e das características da rede. As respostas ao inquérito também mostraram que a visualização do vídeo e da apresentação teve boa aceitação, demonstrando o interesse existente em conteúdos nestes formatos. Mas também revelaram o pouco interesse nas notas e no acompanhamento do vídeo com legendas, o que levará à sua redefinição.

Os avanços tecnológicos nas mais diversas áreas, tais como na capacidade de processamento e de armazenamento, nas normas públicas para a multimédia, e na existência de redes com maiores garantias de qualidade, permitiram que o desenvolvimento de aplicações multimédia seja mais simples e rápido. O interesse por este tipo de aplicações é grande, mas as fracas características técnicas a nível dos terminais e das redes apenas permitiam aplicações de qualidade reduzida. Apesar das melhorias técnicas, ainda é necessário ter a consciência sob os diversos factores que envolvem os *media* mais ricos, como o vídeo e o áudio. As linguagens para o tratamento de dados multimédia são cada vez mais completas e permitem uma interacção mais poderosa ao utilizador. O uso de metadados irá permitir uma maior eficiência na utilização dos conteúdos produzidos.

7.2 Desenvolvimentos futuros

O trabalho a desenvolver após esta dissertação pode-se dividir em diferentes perspectivas. A primeira parte do trabalho futuro passará pela implementação dos componentes especificados, mas não implementados, como ecrãs dos docentes e dos administradores, e exportação/

importação de conteúdos. Numa segunda fase, será feita a especificação mais detalhada de alguns componentes previstos inicialmente, como os Treinos e as Avaliações.

Apesar de não terem sido planeados, ainda existem muitos incrementos que podem ser adicionados ao sistema. Alguns deles foram descobertos durante a especificação e no desenvolvimento do sistema, outros foram retirados das sugestões deixadas nos inquéritos e na troca de opiniões com os utilizadores, dos quais se destacam:

- Implementação do sequenciamento dos cursos, seguindo as regras definidas pelo SCORM;
- Gravação e disponibilização das lições em tempo real;
- Elaboração de relatórios pedagógicos para os docentes;
- Aviso das alterações dos conteúdos para os alunos;
- Personalização dos dados pessoais dos utilizadores.

Os desenvolvimentos futuros também poderão passar por adaptar o sistema a novas utilizações. No momento apenas se pretendia apresentar conteúdos educacionais, mas através de algumas variações o sistema pode ser usado em ambientes distintos, como apresentação de produtos (um imóvel ou um automóvel, entre outros) ou como portal de notícias. Estes últimos são apenas alguns exemplos da utilização do vídeo num ambiente multimédia e interactivo.

Referências

- [Adobe 03] ADOBE Dynamic Media Group (2003). *A Streaming Media Primer.*, disponível em <http://www.adobe.com/video>, Janeiro de 2004
- [Battista 04] BATTISTA, Stefano & CASALINO, Franco & LANDE, Claudio (2004). *MPEG-4: A Multimedia Standard for the Third Millennium, Part 1*, disponível em http://www.computer.org/multimedia/articles/MPEG4_2.htm, Outubro de 2004
- [Blake 98] Blake, S. & Black, D. & Carlson, M. & Davies, E. & Wang, Z. & Weiss, W. (1998). *An Architecture for Differentiated Services*, disponível em <http://www.ietf.org/rfc/rfc2475.txt>, Outubro de 2004
- [Bloom 84] BLOOM, B.S. (1984). *The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring*. Educational Researcher, 13, 4-16.
- [Bulterman 01] BULTERMAN, Dick C.A. (2001). *SMIL 2.0 Part 1: Overview, Concepts, and Structure*, IEEE MultiMedia, October–December 2001, 82-88
- [Bulterman 02] BULTERMAN, Dick C.A. (2001). *SMIL 2.0 Part 2: Overview, Concepts, and Structure*, IEEE MultiMedia, January–March 2002, 74-84
- [Constantine 03] CONSTANTINE, L (2003). *Canonical Abstract Prototypes for Abstract Visual and Interaction Design*, disponível em <http://www.foruse.com/articles/abstract.htm>, Abril de 2004
- [DBDesigner] DBDesigner 4, disponível em <http://www.fabforce.net/dbdesigner4>, Março de 2004
- [DCMI 04] DCMI (2004). *About the Dublin Core Metadata Initiative*, disponível em <http://dublincore.org/about/>, Outubro de 2004
- [DSS] Darwin Streaming Server. Apple, disponível em <http://developer.apple.com/darwin/projects/streaming/>, Novembro de 2004
- [FCS] Flash Communication Server MX, Macromedia, disponível em <http://www.macromedia.com/software/flashcom/>, Novembro de 2004
- [Fletcher 03] FLETCHER, J. D. (2003). *Evidence for Learning from Technology-Assisted Instruction*. In H. F. O'Neil Jr. and R. Perez (Eds.) *Technology Applications in Education: A Learning View*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- [Gettinger 84] GETTINGER, M. (1984). *Individual differences in time needed for learning: A review of the literature*. Educational Psychologist, 19,15-29.
- [Graesser 94] GRAESSER, A. C., e Person, N. K. (1994). *Question asking during tutoring*. American Educational Research Journal, 31, 104-137.
- [HDS] Helix DNA Server. RealNetworks, disponível em <https://helix-server.helixcommunity.org/>, Novembro de 2004
- [Jacklin 02] JACKLIN, Martin (2002). *MPEG-4 The Media Standard: The landscape of advanced multimedia*

- coding. mp4* – MPEG-4 Industry Forum, disponível em <http://www.m4if.org/resources>, 30 págs., 19 de Novembro de 2002
- [Jacobson 99] Jacobson, I. & G. Booch & J. Rumbaugh (1999). *The unified software development process*. The Addison-Wesley object technology series. 1999, Reading, Mass: Addison-Wesley. xxix, 463.
- [Koenen 00] KOENEN, Rob (2000). *Overview of the MPEG-4 Standard*, ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Audio, Beijing, Julho de 2000
- [Koshy 02] KOSHY, Philips & RAGHAVAN, S. V. (2002). *Quality adaptation for FGS MPEG-4 video streaming over Internet*, Proceedings of the 15th international conference on Computer communication, Editor: International Council for Computer Communication Washington, DC, USA, págs. 654 – 663, 11 a 14 de Agosto de 2002
- [Legard 04] LEGARD, David (2004). *IDC: Consolidation to Windows won't happen*, disponível em <<http://www.linuxworld.com.au/index.php/id;940707233;fp;2;fpid;1>>, Setembro de 2004
- [LimSee2] *LimSee 2*, disponível em <http://wam.inrialpes.fr/software/limsee2/>, Junho de 2004
- [Lin 01] LIN, Eugene T. & Cook, Gregory W. & Salama, Paul & Delp, Edward J. (2001) *An Overview of Security Issues in Streaming Video*, International Conference on Information Technology: Coding and Computing, Las Vegas, 2 de Abril de 2001, 345-348
- [Liu 99] LIU, Chunlei (1999). *Multimedia Over IP: RSVP, RTP, RTCP, RTSP*. Cap. 2 - Handbook of Emerging Communications Technologies: The Next Decade, CRC Press, Ed. Rafael Osso, Boca Raton, FL, 26 de Agosto de 1999
- [LOM 04] IEEE LOM (2004). *IEEE WG12: Learning Object Metadata*, disponível em <http://ltsc.ieee.org/wg12/index.html>, Abril 2004
- [Nunes 01a] NUNES, Nuno J. & Cunha, João F. (2001). *Object Modeling and User Interface Design: Designing Interactive Systems*, M.v. Harmelen, Editor. 2001, Addison-Wesley, cap. 6 *Wisdom-Whitewater Interactive System Development with Object Models*
- [Nunes 01b] NUNES, Nuno J. (2001). *Modelação por Objectos para o Desenvolvimento Centrado nos Utilizadores e o Desenho de Interfaces com o Utilizador*, Tese de Doutoramento, UMa, 2001
- [Osso 99] OSSO, Rafael (1999). *Handbook of Emerging Communications Technologies: The Next Decade*, CRC Press, Boca Raton, FL, 26 de Agosto de 1999, 416 páginas
- [Reload] *Reload 1.2*, disponível em <http://www.reload.ac.uk/>, Junho de 2004
- [Santos 99] SANTOS, Ana Paula S. (1999). *Qualidade de Serviço na Internet*, Boletim bimestral sobre tecnologia de redes, volume 3, número 6, Rede Nacional de Ensino e Pesquisa, disponível em <http://www.rnp.br/newsgen/9911/qos.html>
- [SCORM CAM 04] *SCORM Content Aggregation Model*. Advanced Distributed Learning, 30-01-2004, versão 1.3, disponível em <http://www.adlnet.org/>, Março de 2004
- [SCORM Overview 04] *SCORM 2004 Overview*, Advanced Distributed Learning, 30-01-2004, disponível em <http://www.adlnet.org/>, Março de 2004
- [SCORM RT 04] *SCORM Run-Time Environment*. Advanced Distributed Learning, 30-01-2004, Versão 1.3, disponível em <http://www.adlnet.org/>, Março de 2004

- [SCORM SN 04] *SCORM Sequencing and Navigation*. Advanced Distributed Learning, 30-01-2004, Versão 1.3, disponível em <http://www.adlnet.org/>, Março de 2004
- [SEPY] SE|PY, disponível em <http://www.sephiroth.it/python/sepy.php>, Março de 2004
- [Silva et al. 01] Silva, Alberto & Videira, Carlos (2001). *UML, Metodologias e Ferramentas CASE*, 1ª edição, colecção Tecnologias, Centro Atlântico, Portugal, Abril de 2001
- [Simmons 04] SIMMONS, David (2001). *Macromedia Flash Communications Server MX Security Overview*, disponível em http://www.macromedia.com/devnet/mx/flashcom/articles/security_overview.html, Setembro de 2004
- [SMIL20] *Synchronized Multimedia Integration Language (SMIL 2.0)*, W3C Recommendation, August 2001, <http://www.w3.org/TR/2001/REC-smil20-20010807/>, de 7 de Agosto de 2001
- [WMS] *Windows Media Services 9 Series*. Microsoft Windows, disponível em <http://www.microsoft.com/windows/windowsmedia/9series/server.aspx>, Fevereiro de 2004
- [Wu 01a] WU, Dapeng & Hou, Yiwei Thomas & Zhu, Wenwu & Zhang, Ya-Qin & Peha, Jon M. (2001). *Streaming Video over the Internet: Approaches and Directions*, IEEE Transaction on Circuits and Systems for Video Technology, Vol. 11, N° 3, March 2001
- [Wu 01b] WU, Dapeng & Hou, Yiwei Thomas & Zhang, Ya-Qin (2001). *Transporting Real-Time Video over the Internet: Challenges and Approaches*, Proceedings of the IEEE, Vol. 88, N° 12, December 2000

Bibliografia

- FLUCKIGER, François (1995). *Understanding networked multimedia: applications and technology*, 1ª Edição, Prentice Hall, Londres, 1995
- HARMELEN, M.V. (2001). *Object Modeling and User Interface Design: Designing Interactive Systems*, M.v. Harmelen, Editor. 2001, Addison-Wesley
- OSSO, Rafael (1999). *Handbook of Emerging Communications Technologies: The Next Decade*, CRC Press, Boca Raton, FL, 26 de Agosto de 1999, 416 páginas
- SILVA, Alberto & Videira, Carlos (2001). *UML, Metodologias e Ferramentas CASE*, 1ª edição, colecção Tecnologias, Centro Atlântico, Portugal, Abril de 2001
- TOPIC, Michael (2002). *Streaming Media Demystified*, MacGraw-Hill, Nova York, 2002

Anexo A – Documento SMIL

O exemplo da utilização do SMIL é uma cena com uma composição semelhante à existente no sistema *inVÍdeo*. A figura A-1 mostra a apresentação de um vídeo paralelamente com uma imagem e algum texto.

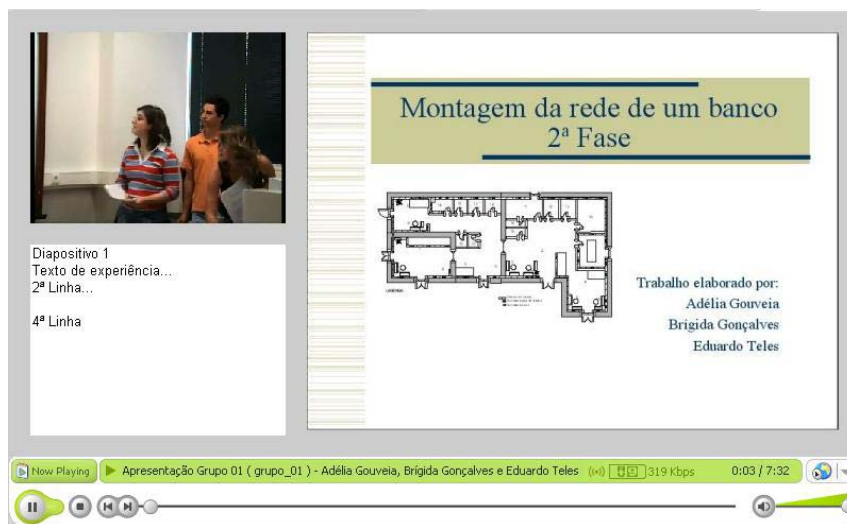


Figura A-1 Cena SMIL

O código seguinte apresenta o ficheiro SMIL que corresponde a toda a sequência multimédia onde se encaixa a cena apresentada na figura A-1.

```
<smil xmlns="http://www.w3.org/2001/SMIL20/Language"
      xmlns:rn="http://features.real.com/2001/SMIL20/Extensions">
  <head>
    <meta name="title" content="Apresentação Grupo 01" />
    <meta name="author" content="Adélia Gouveia, Brigida Gonçalves e Eduardo Teles" />
  </head>
  <layout>
    <root-layout
      width="800"
      height="420"
      backgroundColor="#CCCCCC"
      background-color="#CCCCCC" />
    <region top="20"
      width="500"
      height="375"
      fit="meet"
      id="region_apresentacao"
      right="20"
      z-index="0" />
    <region id="region_video"
      top="20"
      width="240"
      height="180"
      left="20"
      background-color="white"
      z-index="0" />
    <region top="220"
      height="180"
      width="240"
      left="20"
```

```

        id="region_notas"
        background-color="#FFFFCC" />
    </layout>
</head>
<body>
    <par>
        <video region="region_video" fit="fill" src="filme_01_300k.rm" />
    <seq>
        <par>
            
            <text region="region_notas" src="apresentacao_01/Diapositivo1.txt"
                end="6s" begin="0s" />
        </par>
        <par>
            
            <text region="region_notas" src="apresentacao_01/Diapositivo2.txt"
                begin="0s" end="44s" />
        </par>
        <par>
            
            <text region="region_notas" src="apresentacao_01/Diapositivo3.txt"
                begin="0s" end="36s" />
        </par>
        <par>
            
            <text region="region_notas" src="apresentacao_01/Diapositivo4.txt"
                begin="0s" end="26s" />
        </par>
        <par>
            
            <text region="region_notas" src="apresentacao_01/Diapositivo5.txt"
                begin="0s" end="46s" />
        </par>
        
        
        
        
        
        
    </seq>
</par>
</body>
</smil>

```

Anexo B – Manifesto SCORM

O exemplo colocado neste anexo mostra um manifesto com as apresentações dos trabalhos práticos de uma disciplina. A Figura B-1 mostra a árvore com todos os elementos (organizações, recursos e metadados). Alguns dos elementos são apresentados estendidos, mostrando um maior detalhe, os restantes têm nos seus ‘ramos’ uma estrutura semelhante. Em alguns casos existem metadados.

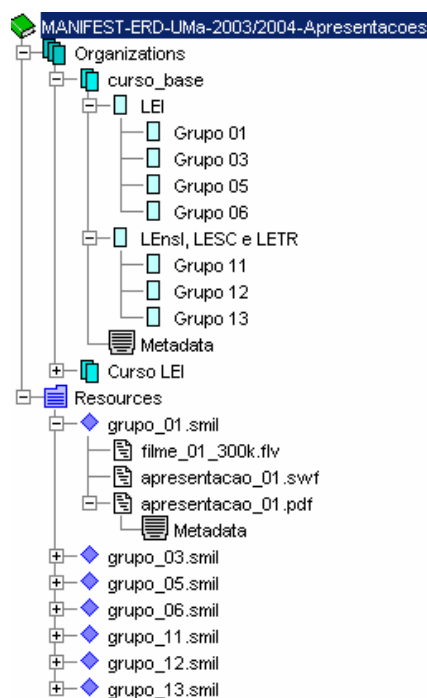


Figura B-1 Representação gráfica de um manifesto

De seguida é apresentado o código do manifesto que constitui a árvore apresentada na figura B-1.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- This is a Reload version 1.2 Content Package document -->
<!-- Spawned from the Reload Content Package Generator - http://www.reload.ac.uk -->
<manifest xmlns="http://www.imsglobal.org/xsd/imscp_v1p1"
  xmlns:imsmd="http://www.imsglobal.org/xsd/imsmd_v1p2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  identifier="MANIFEST-ERD-UMa-2003/2004-Apresentacoes"
  xsi:schemaLocation="http://www.imsglobal.org/xsd/imscp_v1p1 imscp_v1p1.xsd
    http://www.imsglobal.org/xsd/imsmd_v1p2 imsmd_v1p2p2.xsd" version="1.3">
  <organizations default="ORG-BF3382DE-7A48-8AEA-C2EB-8CF16DE41E4B">
    <organization identifier="ORG-ERD-UMa-2003/2004" structure="hierarchical">
      <title>curso_base</title>
      <item identifier="ITEM-ERD-UMa-2003/2004-LEI" isvisible="true">
        <title>LEI</title>
        <item identifier="ITEM-ERD-UMa-2003/2004-LEI-Grupo01" identifierref="RES-ERD-UMa-
          2003/2004-Grupo01" isvisible="true">
```

```

        <title>Grupo 01</title>
    </item>
    <item identifier="ITEM-ERD-UMa-2003/2004-LEI-Grupo03" identifierref="RES-ERD-UMa-
        2003/2004-Grupo03" isvisible="true">
        <title>Grupo 03</title>
    </item>
    <item identifier="ITEM-ERD-UMa-2003/2004-LEI-Grupo05" identifierref="RES-ERD-UMa-
        2003/2004-Grupo05" isvisible="true">
        <title>Grupo 05</title>
    </item>
    <item identifier="ITEM-ERD-UMa-2003/2004-LEI-Grupo06" identifierref="RES-ERD-UMa-
        2003/2004-Grupo06" isvisible="true">
        <title>Grupo 06</title>
    </item>
    <item>
    <item identifier="ITEM-ERD-UMa-2003/2004-LEnSI/LESC/LETR" isvisible="true">
        <title>LEnSI, LESC e LETR</title>
    <item identifier="ITEM-ERD-UMa-2003/2004-LEI-Grupo11" identifierref="RES-ERD-UMa-
        2003/2004-Grupo11" isvisible="true">
        <title>Grupo 11</title>
    </item>
    <item identifier="ITEM-ERD-UMa-2003/2004-LEI-Grupo12" identifierref="RES-ERD-UMa-
        2003/2004-Grupo12" isvisible="true">
        <title>Grupo 12</title>
    </item>
    <item identifier="ITEM-ERD-UMa-2003/2004-LEI-Grupo13" identifierref="RES-ERD-UMa-
        2003/2004-Grupo13" isvisible="true">
        <title>Grupo 13</title>
    </item>
</item>
</item>
<metadata>
    <imsmd:lom>
        <imsmd:general>
            <imsmd:language>pt</imsmd:language>
            <imsmd:description>
                <imsmd:langstring xml:lang="en">Apresentações dos trabalhos de Engenharias de
                    Redes e Dados da UMA - 2003/2004</imsmd:langstring>
            </imsmd:description>
            <imsmd:keyword>
                <imsmd:langstring xml:lang="en">redes, apresentacao</imsmd:langstring>
            </imsmd:keyword>
            <imsmd:structure>
                <imsmd:source>
                    <imsmd:langstring xml:lang="en">LOMv1.0</imsmd:langstring>
                </imsmd:source>
                <imsmd:value>
                    <imsmd:langstring xml:lang="x-none">Hierarchical</imsmd:langstring>
                </imsmd:value>
            </imsmd:structure>
            <imsmd:aggregationlevel>
                <imsmd:source>
                    <imsmd:langstring xml:lang="en">LOMv1.0</imsmd:langstring>
                </imsmd:source>
                <imsmd:value>
                    <imsmd:langstring xml:lang="x-none">1</imsmd:langstring>
                </imsmd:value>
            </imsmd:aggregationlevel>
        </imsmd:general>
        <imsmd:lifecycle>
            <imsmd:version>
                <imsmd:langstring xml:lang="en">1</imsmd:langstring>
            </imsmd:version>
            <imsmd:status>
                <imsmd:source>
                    <imsmd:langstring xml:lang="en">LOMv1.0</imsmd:langstring>
                </imsmd:source>
                <imsmd:value>
                    <imsmd:langstring xml:lang="x-none">Final</imsmd:langstring>
                </imsmd:value>
            </imsmd:status>
        </imsmd:lifecycle>
    </imsmd:lom>
</metadata>
</organization>
<organization identifier="ORG-ERD/LEI-UMa-2003/2004" structure="hierarchical">
    <title>Curso LEI</title>
    <item identifier="ITEM-A78943EE-264F-A8CB-8A99-C9CF357485D7" isvisible="true">
        <title>LEI</title>
    <item identifier="ITEM-F7AF6871-C1A6-D819-DC44-68C7EF7F5623" identifierref="RES-ERD-UMa-
        2003/2004-Grupo01" isvisible="true">

```

```

        <title>Grupo 01</title>
      </item>
      <item identifier="ITEM-22241667-7461-98EF-AD5F-2472B2A134AF" identifierref="RES-ERD-UMa-2003/2004-Grupo03" isvisible="true">
        <title>Grupo 03</title>
      </item>
      <item identifier="ITEM-5D139497-3448-63FF-12C4-0BB3BF321F51" identifierref="RES-ERD-UMa-2003/2004-Grupo05" isvisible="true">
        <title>Grupo 05</title>
      </item>
      <item identifier="ITEM-2DFBA81F-F044-DE44-AC94-E65D76116A1A" identifierref="RES-ERD-UMa-2003/2004-Grupo06" isvisible="true">
        <title>Grupo 06</title>
      </item>
    </organization>
  </organizations>
  <resources>
    <resource identifier="RES-ERD-UMa-2003/2004-Grupo01" type="webcontent" href="grupo_01.smil">
      <file href="filme_01_300k.flv" />
      <file href="apresentacao_01.swf" />
      <file href="apresentacao_01.pdf">
        <metadata>
          <imsmd:lom>
            <imsmd:general>
              <imsmd:language>pt</imsmd:language>
              <imsmd:description>
                <imsmd:langstring xml:lang="pt">Pdf's dos slides da apresentação do grupo 01</imsmd:langstring>
              </imsmd:description>
              <imsmd:keyword>
                <imsmd:langstring xml:lang="en">Redes, equipamento activo</imsmd:langstring>
              </imsmd:keyword>
              <imsmd:aggregationlevel>
                <imsmd:source>
                  <imsmd:langstring xml:lang="en">LOMv1.0</imsmd:langstring>
                </imsmd:source>
                <imsmd:value>
                  <imsmd:langstring xml:lang="x-none">4</imsmd:langstring>
                </imsmd:value>
              </imsmd:aggregationlevel>
            </imsmd:general>
            <imsmd:lifecycle>
              <imsmd:version>
                <imsmd:langstring xml:lang="en">1</imsmd:langstring>
              </imsmd:version>
              <imsmd:status>
                <imsmd:source>
                  <imsmd:langstring xml:lang="en">LOMv1.0</imsmd:langstring>
                </imsmd:source>
                <imsmd:value>
                  <imsmd:langstring xml:lang="x-none">Final</imsmd:langstring>
                </imsmd:value>
              </imsmd:status>
            </imsmd:lifecycle>
            <imsmd:technical>
              <imsmd:format>application/pdf</imsmd:format>
              <imsmd:requirement>
                <imsmd:type>
                  <imsmd:source>
                    <imsmd:langstring xml:lang="en">LOMv1.0</imsmd:langstring>
                  </imsmd:source>
                  <imsmd:value>
                    <imsmd:langstring xml:lang="x-none" />
                  </imsmd:value>
                </imsmd:type>
              </imsmd:requirement>
            </imsmd:technical>
            <imsmd:educational>
              <imsmd:interactivitytype>
                <imsmd:source>
                  <imsmd:langstring xml:lang="en">LOMv1.0</imsmd:langstring>
                </imsmd:source>
                <imsmd:value>
                  <imsmd:langstring xml:lang="x-none">Expositive</imsmd:langstring>
                </imsmd:value>
              </imsmd:interactivitytype>
              <imsmd:learningresourcetype>
                <imsmd:source>
                  <imsmd:langstring xml:lang="en">LOMv1.0</imsmd:langstring>
                </imsmd:source>
              </imsmd:learningresourcetype>
            </imsmd:educational>
          </imsmd:lom>
        </file>
      </resource>
  </resources>

```

```

</imsmd:source>
<imsmd:value>
  <imsmd:langstring xml:lang="x-none">Slide</imsmd:langstring>
</imsmd:value>
</imsmd:learningresourcetype>
<imsmd:interactivitylevel>
  <imsmd:source>
    <imsmd:langstring xml:lang="en">LOMv1.0</imsmd:langstring>
  </imsmd:source>
  <imsmd:value>
    <imsmd:langstring xml:lang="x-none">very low</imsmd:langstring>
  </imsmd:value>
</imsmd:interactivitylevel>
<imsmd:semanticdensity>
  <imsmd:source>
    <imsmd:langstring xml:lang="en">LOMv1.0</imsmd:langstring>
  </imsmd:source>
  <imsmd:value>
    <imsmd:langstring xml:lang="x-none">medium</imsmd:langstring>
  </imsmd:value>
</imsmd:semanticdensity>
<imsmd:intendedenduserrole>
  <imsmd:source>
    <imsmd:langstring xml:lang="en">LOMv1.0</imsmd:langstring>
  </imsmd:source>
  <imsmd:value>
    <imsmd:langstring xml:lang="x-none">Author</imsmd:langstring>
  </imsmd:value>
</imsmd:intendedenduserrole>
<imsmd:context>
  <imsmd:source>
    <imsmd:langstring xml:lang="en">LOMv1.0</imsmd:langstring>
  </imsmd:source>
  <imsmd:value>
    <imsmd:langstring xml:lang="x-none">University First
      Cycle</imsmd:langstring>
  </imsmd:value>
</imsmd:context>
<imsmd:typicalagerange>
  <imsmd:langstring xml:lang="en">20-23</imsmd:langstring>
</imsmd:typicalagerange>
<imsmd:difficulty>
  <imsmd:source>
    <imsmd:langstring xml:lang="en">LOMv1.0</imsmd:langstring>
  </imsmd:source>
  <imsmd:value>
    <imsmd:langstring xml:lang="x-none">medium</imsmd:langstring>
  </imsmd:value>
</imsmd:difficulty>
<imsmd:language>pt</imsmd:language>
</imsmd:educational>
<imsmd:rights>
  <imsmd:cost>
    <imsmd:source>
      <imsmd:langstring xml:lang="en">LOMv1.0</imsmd:langstring>
    </imsmd:source>
    <imsmd:value>
      <imsmd:langstring xml:lang="x-none">no</imsmd:langstring>
    </imsmd:value>
  </imsmd:cost>
<imsmd:copyrightandotherrestrictions>
  <imsmd:source>
    <imsmd:langstring xml:lang="en">LOMv1.0</imsmd:langstring>
  </imsmd:source>
  <imsmd:value>
    <imsmd:langstring xml:lang="x-none">no</imsmd:langstring>
  </imsmd:value>
</imsmd:copyrightandotherrestrictions>
</imsmd:rights>
<imsmd:relation>
  <imsmd:kind>
    <imsmd:source>
      <imsmd:langstring xml:lang="en">LOMv1.0</imsmd:langstring>
    </imsmd:source>
    <imsmd:value>
      <imsmd:langstring xml:lang="x-none" />
    </imsmd:value>
  </imsmd:kind>
</imsmd:relation>
<imsmd:classification>

```

```

        <imsmd:purpose>
          <imsmd:source>
            <imsmd:langstring xml:lang="en">LOMv1.0</imsmd:langstring>
          </imsmd:source>
          <imsmd:value>
            <imsmd:langstring xml:lang="x-none">Educational Level</imsmd:langstring>
          </imsmd:value>
        </imsmd:purpose>
      </imsmd:classification>
    </imsmd:lom>
  </metadata>
</file>
</resource>
<resource identifier="RES-ERD-UMa-2003/2004-Grupo03" type="webcontent" href="grupo_03.smil">
  <file href="filme_03_300k.flv" />
  <file href="apresentacao_03.swf" />
  <file href="apresentacao_03.pdf" />
</resource>
<resource identifier="RES-ERD-UMa-2003/2004-Grupo05" type="webcontent" href="grupo_05.smil">
  <file href="filme_05_300k.flv" />
  <file href="apresentacao_05.swf" />
  <file href="apresentacao_05.pdf" />
</resource>
<resource identifier="RES-ERD-UMa-2003/2004-Grupo06" type="webcontent" href="grupo_06.smil">
  <file href="filme_06_300k.flv" />
  <file href="apresentacao_06.swf" />
  <file href="apresentacao_06.pdf" />
</resource>
<resource identifier="RES-ERD-UMa-2003/2004-Grupo11" type="webcontent" href="grupo_11.smil">
  <file href="filme_11_300k.flv" />
  <file href="apresentacao_11.swf" />
  <file href="apresentacao_11.pdf" />
</resource>
<resource identifier="RES-ERD-UMa-2003/2004-Grupo12" type="webcontent" href="grupo_12.smil">
  <file href="filme_12_300k.flv" />
  <file href="apresentacao_12.swf" />
  <file href="apresentacao_12.pdf" />
</resource>
<resource identifier="RES-ERD-UMa-2003/2004-Grupo13" type="webcontent" href="grupo_13.smil">
  <file href="filme_13_300k.flv" />
  <file href="apresentacao_13.swf" />
  <file href="apresentacao_13.pdf" />
</resource>
</resources>
</manifest>

```

Anexo C – Resultados do inquérito

Os documentos presentes neste anexo são os seguintes:

- folha de inquérito enviada para os utilizadores;
- folha de inquérito com os resultados obtidos;
- lista de sugestões deixadas nos inquéritos;
- lista de comentários deixados nos inquéritos.

Inquérito sobre Aplicação inVÍdeo**Caracterização Técnica***Computador*

Processador: ☐ < 500 MHz ☐ 500 a 1399 MHz ☐ 1400 a 2399 MHz ☐ >= 2400 MHz
 Memória: ☐ < 128 MB ☐ 128 a 511 MB ☐ 512 a 1023 MB ☐ >= 1024 MB
 Disco Rígido: ☐ < 20 GB ☐ 20 a 39 GB ☐ >= 40 GB
 Sistema Operativo: ☐ Windows ☐ MacOS ☐ Linux/Unix

Rede

Tipo de ligação: ☐ Telefone ☐ ADSL ☐ Cabo ☐ LAN ☐ Outra
 Velocidade de Acesso: _____ bps

Plug-in

Flash Player 7 instalado? ☐ Sim ☐ Não Se não, instalou? ☐ Sim ☐ Não

Local

Acesso à aplicação de: ☐ Casa ☐ Uma ☐ Trabalho ☐ Outro: _____

Caracterização do Utilizador*Dados Pessoais*

Sexo: M ☐ F ☐ Idade: _____ Nome (opcional): _____

Conhecimentos Informáticos

Computadores e Internet em geral: Mau ☐ ☐ ☐ ☐ Bom

Uso regular de serviços de áudio e vídeo pela Internet?

Áudio: Pouco ☐ ☐ ☐ ☐ Muito ☐
 Vídeo: Pouco ☐ ☐ ☐ ☐ Muito ☐

Caracterização da Experiência*Tempo de resposta*

Listagens cursos, módulos e sessões: Mau ☐ ☐ ☐ ☐ Bom ☐
 Início da sessão: Mau ☐ ☐ ☐ ☐ Bom ☐

Qualidade da

Visualização do vídeo: Má ☐ ☐ ☐ ☐ Boa ☐
 Visualização da apresentação: Má ☐ ☐ ☐ ☐ Boa ☐

Relevância dos seguintes componentes

Vídeo: Pouca ☐ ☐ ☐ ☐ Muita ☐
 Apresentação: Pouca ☐ ☐ ☐ ☐ Muita ☐
 Notas: Pouca ☐ ☐ ☐ ☐ Muita ☐
 Legendas: Pouca ☐ ☐ ☐ ☐ Muita ☐
 Documentos para *download*: Pouca ☐ ☐ ☐ ☐ Muita ☐
 Mensagens tipo e-mail (a implementar): Pouca ☐ ☐ ☐ ☐ Muita ☐
 Mensagens em tempo real (a implementar): Pouca ☐ ☐ ☐ ☐ Muita ☐

Sugestões de novas funcionalidades _____

Espaço para outras observações e comentários

Muito obrigado pelo tempo e atenção!

Inquérito sobre aplicação inVÍdeo - Eduardo Marques, trabalho de Dissertação, Julho de 2004

Inquérito sobre Aplicação inVÍdeo

Caracterização Técnica

Computador

Processador: **1** < 500 MHz **2** 500 a 1399 MHz **2** 1400 a 2399 MHz **6** >= 2400 MHz
 Memória: **0** < 128 MB **3** 128 a 511 MB **4** 512 a 1023 MB **4** >= 1024 MB
 Disco Rígido: **2** < 20 GB **1** 20 a 39 GB **8** >= 40 GB
 Sistema Operativo: **10** Windows **1** MacOS **0** Linux/Unix

Rede

Tipo de ligação: **1** Telefone **2** ADSL **2** Cabo **5** LAN **1** Outra
 Velocidade de Acesso: **entre 44kbps e 100Mbps**

Plug-in

Flash Player 7 instalado? **9** Sim **2** Não Se não, instalou? **1** Sim **1** Não

Local

Acesso à aplicação de: **6** Casa **5** UMa **0** Trabalho **0** Outro: _____

Caracterização do Utilizador

Dados Pessoais

Sexo: M **10** F **1** Idade: **entre 21 e 31 anos**

Conhecimentos Informáticos

Computadores e Internet em geral: Mau **0** **0** **1** **2** **8** Bom

Uso regular de serviços de áudio e vídeo pela Internet?

Áudio: Pouco **2** **1** **2** **4** **2** Muito

Vídeo: Pouco **1** **2** **4** **3** **1** Muito

Caracterização da Experiência

Tempo de resposta

Listagens cursos, módulos e sessões: Mau **0** **0** **1** **2** **6** Bom

Início da sessão: Mau **0** **0** **1** **4** **6** Bom

Qualidade da

Visualização do vídeo: Má **0** **0** **2** **6** **3** Boa

Visualização da apresentação: Má **0** **0** **1** **4** **6** Boa

Relevância dos seguintes componentes

Vídeo: Pouca **0** **0** **1** **3** **7** Muita

Apresentação: Pouca **0** **0** **2** **3** **6** Muita

Notas: Pouca **2** **2** **4** **3** **0** Muita

Legendas: Pouca **2** **5** **1** **2** **1** Muita

Documentos para *download*: Pouca **0** **2** **0** **4** **5** Muita

Mensagens tipo e-mail (a implementar): Pouca **0** **0** **3** **6** **2** Muita

Mensagens em tempo real (a implementar): Pouca **0** **0** **3** **6** **1** Muita

Lista de sugestões para novas funcionalidades:

- “sistema de aviso automático quando apareçam actualizações usando por exemplo mensagens tipo email”
- “montar um fórum de forma a ser possível discutir todo e qualquer conteúdo”
- “Porque é que a UMa não adopta isto para os próprios docentes/investigadores? Novas funcionalidades... assim de repente só me lembro de uma caixa de procura (por assunto ou natureza do item: exames, aulas teóricas/práticas). Isto consoante a dimensão do módulo em causa, claro. Se me lembrar de mais coisas eu digo”
- “Notas para o docente para colocação de informação como p.e. Classificação do grupo, etc...”
- “No caso de esta aplicação ser usada pelos alunos tb, poderiam ser colocadas notas tb para o aluno.”
- “Na continuidade da aplicação a estruturação da informação introduzida neste campo poderia ser relevante.”
- “Permitir dar mais ênfase a determinadas partes, da aplicação, por exemplo mostrar apenas filme e legendas, maximizando a área de visão (embora já seja possível fazendo o zoom).”

Lista de comentários e observações

- “As legendas, se fossem utilizadas, onde ficariam? Se fosse na própria imagem de vídeo seria difícil de ler”
- “Desde já congratulo pelo trabalho efectuado, visto ser um projecto com muita utilidade académica e de grandes funcionalidades. Tirando o que falta implementar, diria que fica muito perto de um óptimo trabalho. Parabéns e Felicidades”
- “Na minha opinião as componentes acima implementadas já são suficientes para o fornecimento de um bom serviço”
- “Acho a funcionalidade deveras interessante e com muito potencial. Muito útil para todo o tipo de cadeiras com apresentações.”
- “Apenas não consegui ouvir som. O vídeo decorreu sem problemas de maior”
- “Achei muito interessante a organização e as funcionalidades disponibilizadas pela aplicação e considero que seria benéfico para nós alunos se todas as apresentações realizadas fossem filmadas e disponibilizadas posteriormente, para que possamos nos apercebermos dos erros que cometemos de forma a evitar repeti-los.”
- “Tá muito fixe a cena, gostei da parte das notas acompanharem o decorrer das apresentações e o interface se lembrar de onde estávamos quando recuamos, permitiu percorrer todas as apresentações. Vê-se que o esqueleto está montado e há agora espaço para evolução gráfica. Está de parabéns!”
- “Para uma Universidade com as características da UMa parece-me uma aplicação

excelente e com enorme potencial (por exemplo, trabalhadores estudantes ou estudantes que moram em sítios longínquos). Parabéns!”

- “Com a minha ligação levou um bom bocado a transferir o filme em *Streaming* se calhar devia existir a possibilidade de fazer o download do filme, visto que não é em directo e é muito chato ver aquilo aos bocadinhos sempre está a fazer o download.”
- “A disponibilização das notas não é imediata, é necessário carregar o filme (apenas um detalhe não prejudica a visualização do vídeo)”
- “Acho que as funcionalidades escolhidas permitem uma vasta possibilidades na utilização da aplicação, o que está muito bom.”
- “As funcionalidades (com o botão direito) activas, algumas podem introduzir acções indesejadas.”