



**João Filipe Ferreira
Rodrigues**

**Servidor IPTV com Personalização Automática de
Canais**

**IPTV Server with Automatic Channels
Personalization**



**João Filipe Ferreira
Rodrigues**

**Servidor IPTV com Personalização Automática de
Canais**

**IPTV Server with Automatic Channels
Personalization**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores de Telemática, realizada sob a orientação científica do Doutor Paulo Salvador e do Doutor António Nogueira, Professores Auxiliares do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

Dedico esta dissertação à minha família pelo apoio incondicional durante todos estes cinco anos de estudo na Universidade de Aveiro, à minha namorada Susana pela inspiração concedida, carinho e compreensão demonstrados e aos meus amigos que me ajudaram quando eu precisei. Por isso, obrigado a todos!

o júri

presidente

Doutor Paulo Miguel Nepomuceno Pereira Monteiro

Universidade de Aveiro

Doutor Paulo Jorge Salvador Serra Ferreira

Universidade de Aveiro

Doutor António Manuel Duarte Nogueira

Universidade de Aveiro

Doutor Joel José Puga Coelho Rodrigues

Faculdade de Engenharia da Beira Interior

agradecimentos

Agradeço especialmente aos meus orientadores, Professor Doutor Paulo Salvador e Professor Doutor António Nogueira, pela excelente orientação dada no âmbito desta tese, por todo o apoio e compreensão demonstrada mesmo quando me atrasava a cumprir um ou outro objectivo.

palavras-chave

IPTV, Televisão Digital, Personalização, Personalização Automática, Agregação, Disponibilidade, Aprendizagem, Serviços, Universalidade

resumo

Desde a invenção da "caixa mágica" que mudou o mundo (TV), esta tem vindo a evoluir constantemente sofrendo transformações não só ao nível físico e electrónico mas também do ponto de vista do conceito subjacente. O aparecimento da televisão digital (DTV) enquadra-se numa das mais significativas evoluções da televisão nos últimos anos que permite actualmente ao utilizador usufruir de uma experiência multimédia de grande qualidade áudio-visual mas ainda muito pouco interactiva. Actualmente o telespectador exige muito mais do que estar restrito a seleccionar um dos canais disponíveis num equipamento de TV que não interage consigo ao nível do conteúdo e não é personalizável. Tem-se verificado que o conceito de aparelho de TV está cada vez mais generalizado a qualquer equipamento com as condições mínimas que permitam ao utilizador assistir a conteúdos multimédia. São exemplos disso dispositivos como alguns telemóveis, PDAs e computadores pessoais (PCs). Estes, quando ligados à Internet, permitem ao utilizador obter uma experiência multimédia fixa ou móvel muito mais interactiva e personalizável usufruindo de um leque diversificado de conteúdos. Assim sendo, porque não integrar a televisão na Internet e a Internet na televisão? É neste âmbito que surge o conceito de IPTV como sistema de televisão acessível, interactiva, personalizável, global e heterogénea tanto ao nível dos conteúdos como ao nível da rede de comunicação e características dos dispositivos cliente, funcionando sobre uma rede IP global.

Nesta Dissertação de Mestrado foi projectado e implementado um protótipo de um servidor IPTV que se prende com a agregação e gestão de uma multiplicidade e diversidade de conteúdos multimédia existentes na Internet. O trabalho desenvolvido conduziu também à criação de condições para a introdução de uma componente de aprendizagem dinâmica inteligente do perfil do utilizador, tendo em vista a sugestão automática de conteúdos. A agregação e gestão de conteúdos tem como objectivo final a disponibilização destes a um cliente IPTV genérico (que não faz parte dos objectivos deste trabalho). Este cliente deverá ter como função tratar de todo o processo de interacção com o utilizador e acesso efectivo aos conteúdos num ambiente de utilização completamente heterogéneo em termos de *hardware* e necessidades do utilizador, orientado à convergência e mobilidade dos meios de visualização. O sistema desenvolvido tenta ser o mais modular possível tendo em vista distribuir o processamento em rede e permitir facilmente a integração de novos serviços ou a alteração dos existentes. O servidor desenvolvido não contempla qualquer mecanismo de codificação, compressão e optimização dos fluxos multimédia nem qualquer política de gestão e transporte dos mesmos na rede de comunicação.

keywords

IPTV, Digital Television, Customization, Automatic Customization, Aggregation, Availability, Learning, Services, Universality

abstract

Since the invention of the “magic box” that changed the world (TV), it has constantly evolved not only at the physical and electronic levels, but also in its underlying concept. The appearance of digital television is one of the most significant developments in television in recent years and has allowed users to enjoy a multimedia experience with much more quality but without significantly more interactivity. Currently, the viewer requires much more than being only restricted to select one of the available channels on the TV equipment that can't interact very much with him and that is not customizable. The TV equipment concept is becoming more generalized to any equipment that has the minimum capabilities that allow the user to view multimedia contents with sound and image. Mobile phones, PDAs and personal computers (PC) are some examples of systems that allow the user, once connected to the Internet, to obtain a fixed or mobile multimedia experience that is more interactive and customizable, benefiting from a diverse range of available contents. So, why not extend these features of personalization and interactivity to “conventional” television systems and turn devices such as PCs and mobile phones into non dedicated TV systems? This constitutes the concept of IPTV, as an accessible, interactive, customizable, comprehensive and heterogeneous environment, both in its contents, networking level, communication features and client devices. In this Master's Thesis, a prototype of an IPTV server system was designed and implemented having in mind the aggregation and management of a multiplicity and diversity of media contents that are available on the Internet. The work has also created the conditions for introducing a learning component of the user profile in an intelligent and dynamic way, in order to suggest contents in an automatic and inteligente form. The aggregation and content management has the final objective of making all contents available to an IPTV client (whose development is not the aim of this work). The client system must be designed to handle the whole process of interacting with the user and access the multimedia contents, in an usage environment that is focused on convergence and mobility in terms of visualization and completely heterogeneous in terms of hardware and user's needs. The developed system tries to be as modular as possible, with the objective of distributing the processing across the network and allow an easy integration of new services or the modification of existing ones. The server has no mechanisms for encoding, compression and optimization of multimedia streams, nor has any management mechanisms for the multimedia transport over the communication network.

Conteúdo

1	Introdução	1
1.1	Motivação e Contexto	1
1.2	Objectivos	3
1.3	Estrutura e Organização desta Dissertação	4
1.4	Visão Geral	7
2	Conceitos Base e Estado Actual da Tecnologia	13
2.1	Introdução	13
2.2	Tecnologias de Televisão	14
2.2.1	Televisão Analógica e História da TV em Portugal	14
2.2.2	Televisão Digital	21
2.2.3	Televisão de Alta Definição	27
2.2.4	Televisão sobre Internet	29
2.2.5	IPTV	34
2.3	Arquitecturas de Redes IPTV	49
2.4	Tecnologias de Desenvolvimento	53
2.5	Conclusão	56
3	A Arquitectura do Servidor: Análise Funcional	59
3.1	Introdução	59
3.2	Requisitos do Sistema	59
3.3	A Arquitectura	60

3.4	Modelo de Comunicação	65
3.5	Modelo de Classificação	68
4	A Arquitectura do Servidor: Análise Modular	77
4.1	Introdução	77
4.2	<i>IPTV Server Core</i>	78
4.2.1	Arquitectura Geral do Funcionamento Interno	87
4.3	<i>Content Link Sources</i>	88
4.3.1	Arquitectura Geral do Funcionamento Interno	92
4.4	<i>User Profile Learning</i>	95
4.5	<i>IPTV Database</i>	97
4.5.1	Tecnologias Utilizadas	97
4.5.2	Modelo de Base de Dados	99
4.5.3	As Entidades da Base de Dados	102
4.5.4	Tipos de Dados Definidos	107
4.5.5	Sistema de Inicialização Automática	107
4.5.6	<i>Triggers</i>	109
4.5.7	Configurações e Excepções	109
4.5.8	Arquitectura Geral do Funcionamento Interno	110
4.6	Gestão e Monitorização: <i>IPTV Server GUI</i>	110
4.6.1	Arquitectura Geral do Funcionamento Interno	117
5	API em <i>Java</i> para Cliente IPTV	119
5.1	Introdução	119
5.2	Sugestões para o Desenvolvimento do Cliente IPTV	119
5.3	A API	121
6	Requisitos e Modelos de Instalação do Servidor	125
6.1	Introdução	125
6.2	Requisitos	125

6.3	Instalação	126
7	Testes Gerais de Desempenho e Comportamento	131
7.1	Introdução	131
7.2	Descrição dos Procedimentos	132
7.3	Resultados	134
7.4	Conclusão	138
8	Conclusões e Trabalho Futuro	141
A	Diagramas de Base de Dados	147
B	Descrição das Mensagens XML e Respectiva Sintaxe XSD	153
B.1	Mensagem <i>User Authentication</i>	154
B.2	Mensagem <i>User Deauthentication</i>	160
B.3	Mensagem <i>New User</i>	163
B.4	Mensagem <i>New User Validation</i>	172
B.5	Mensagem <i>Session Keep Alive Request</i>	175
B.6	Mensagem <i>New Link Request</i>	178
B.7	Mensagem <i>New Link Response</i>	183
B.8	Mensagem <i>Link List Request</i>	190
B.9	Mensagem <i>Link List Response</i>	195
B.10	Mensagem <i>Link Feedback Request</i>	204
B.11	Mensagem <i>Command Request</i>	208
B.12	Mensagem <i>Command Response</i>	212
B.13	Estrutura geral das mensagem XML	218
C	Exemplo de Utilização da API Cliente	219

Lista de Figuras

1	Grafismo exemplificativo dos domínios de QoE e QoS.	xxvi
2	Exemplos de transmissões em <i>unicast</i> , <i>multicast</i> , <i>broadcast</i> e <i>anycast</i> , respectivamente.	xxviii
2.1	Espectro electromagnético.	16
2.2	Sistemas de codificação televisiva a cores agrupados por país. (século XX).	17
2.3	Exemplo gráfico da evolução dos sistemas de televisão.	18
2.4	Distribuição das tecnologias de transmissão de televisão digital terrestre agrupadas por país.	24
2.5	Exemplo de configuração de um sistema típico de DTV via satélite.	25
2.6	Comparação entre um fluxo de dados da TV “convencional” e um fluxo IPTV, respectivamente.	27
2.7	Exemplo comparativo das resoluções obtidas nas tecnologias HDTV e SDTV, respectivamente.	28
2.8	Arquitectura do serviço <i>Triple Play</i>	37
2.9	Representação gráfica do serviço <i>Quadruple Play</i>	37
2.10	Mapa da distribuição dos serviços de IPTV no mundo até meados de 2009.	41
2.11	Esquemas exemplificativos das arquitecturas cliente–servidor e P2P, respectivamente.	50
2.12	Exemplo esquemático de uma rede P2PTV.	51
2.13	Domínios funcionais do IPTV segundo o ITU-T.	52

3.1	Requisitos globais do servidor IPTV. A azul: para futura implementação.	60
3.2	Casos de utilização genéricos do servidor IPTV. A azul: para futura implementação.	61
3.3	Diagrama geral da arquitectura do servidor IPTV implementado.	62
3.4	Enquadramento da API de acesso à base de dados do servidor IPTV.	63
3.5	Interface principal da ferramenta de gestão do servidor.	65
3.6	Enquadramento geral da API de comunicação com o servidor num cliente IPTV.	68
3.7	Exemplo gráfico de herança simples e herança múltipla, respectivamente.	71
3.8	Exemplo da representação gráfica do perfil multimédia de um utilizador num instante específico, incluindo limites mínimos e máximos.	73
4.1	Diagrama de sequência relativa ao módulo <i>IPTV Server Core</i> no caso de execução de uma tarefa do tipo <i>User Authentication</i>	84
4.2	Diagrama de pacotes do módulo <i>IPTV Server Core</i>	87
4.3	Diagrama das classes principais do módulo <i>IPTV Server Core</i> e seus relacionamentos.	88
4.4	Diagrama de pacotes do módulo <i>Content Link Sources</i>	94
4.5	Diagrama das classes principais do módulo <i>Content Link Sources</i> e seus relacionamentos.	95
4.6	Exemplo do <i>layout</i> da aplicação <i>phpMyAdmin</i>	98
4.7	Diagrama do processo iterativo de geração dos modelos de base de dados.	100
4.8	Diagrama de pacotes da API de acesso à base de dados.	110
4.9	Diagrama das classes principais da API de acesso à BD.	111
4.10	Janela principal da ferramenta <i>IPTV Server GUI</i>	113
4.11	Janela das opções de configuração da ferramenta <i>IPTV Server GUI</i>	113
4.12	Menu rápido da ferramenta <i>IPTV Server GUI</i> na bandeja de sistema.	114
4.13	Janelas da ferramenta <i>IPTV Server GUI</i> relativas à gestão e monitorização do módulo <i>IPTV Server Core</i>	115

4.14	Diagrama de pacotes do módulo <i>IPTV Server GUI</i>	118
4.15	Diagrama das classes principais do módulo <i>IPTV Server GUI</i> e seus relacionamentos.	118
5.1	Diagrama de pacotes da API de comunicação cliente–servidor.	122
5.2	Diagrama das classes principais da API de comunicação cliente–servidor.	123
6.1	Diagrama de instalação do servidor IPTV com todos os módulos a funcionar na mesma máquina.	128
6.2	Diagrama de instalação do servidor IPTV com os módulos distribuídos por várias máquinas sem replicação de nenhum módulo.	129
7.1	Gráfico referente aos atrasos produzidos utilizando HTTP e validação XSD das mensagens recebidas pelo servidor.	135
7.2	Gráfico referente aos atrasos produzidos utilizando HTTP sem validação XSD das mensagens recebidas pelo servidor.	137
A.1	Diagrama físico das novas entidades a introduzir na BD para permitir a criação de “pacotes de conteúdos multimédia”.	148
A.2	Diagrama do modelo conceptual da base de dados implementada em notação <i>Barker</i>	149
A.3	Diagrama do modelo lógico da base de dados implementada em notação <i>Barker</i>	150
A.4	Diagrama do modelo físico da base de dados implementada em notação relacional.	151
B.1	Diagrama das relações entre os elementos e atributos para mensagens do tipo <i>User Authentication</i>	156
B.2	Diagrama das relações entre os elementos e atributos para mensagens do tipo <i>User Deauthentication</i>	161

B.3	Diagrama das relações entre os elementos e atributos para mensagens do tipo <i>New User</i>	164
B.4	Diagrama das relações entre os elementos e atributos para mensagens do tipo <i>New User Validation</i>	173
B.5	Diagrama das relações entre os elementos e atributos para mensagens do tipo <i>Session Keep Alive</i>	176
B.6	Diagrama das relações entre os elementos e atributos para mensagens do tipo <i>New Link Request</i>	180
B.7	Diagrama das relações entre os elementos e atributos para mensagens do tipo <i>New Link Response</i>	187
B.8	Diagrama das relações entre os elementos e atributos para mensagens do tipo <i>Link List Request</i>	192
B.9	Diagrama das relações entre os elementos e atributos para mensagens do tipo <i>Link List Response</i>	197
B.10	Diagrama das relações entre os elementos e atributos para mensagens do tipo <i>Link Feedback Request</i>	206
B.11	Diagrama das relações entre os elementos e atributos para mensagens do tipo <i>Command Request</i>	210
B.12	Diagrama das relações entre os elementos e atributos para mensagens do tipo <i>Command Response</i>	215

Lista de Tabelas

2.1	Lista de alguns sistemas P2PTV mais conhecidos.	40
3.1	Tabela dos diferentes tipos de mensagens implementadas	75
4.1	Alguns serviços multimédia que disponibilizam API de acesso.	93
4.2	Entidades implementadas pela base de dados do sistema e respectiva descrição.	106
4.3	Lista dos domínios da base de dados.	108
4.4	Lista das entidade com inicialização.	108
7.1	Tabela da carga média nos processadores do servidor IPTV para testes utilizando HTTP e validação XSD.	135
7.2	Tabela do atraso médio para cada número de clientes considerado, utilizando HTTP com validação XSD das mensagens XML recebidas pelo servidor. . .	136
7.3	Tabela da carga média nos processadores do servidor IPTV para testes utilizando HTTP sem validação XSD.	136
7.4	Tabela do atraso médio para cada número de clientes considerado, utilizando HTTP sem validação XSD das mensagens XML recebidas pelo servidor. . .	137
B.1	Tabela dos elementos e atributos para mensagens do tipo <i>User Authentication</i>	156
B.2	Tabela dos elementos e atributos para mensagens do tipo <i>User Deauthentication</i>	162
B.3	Tabela dos elementos e atributos para mensagens do tipo <i>New User</i>	166
B.4	Tabela dos elementos e atributos para mensagens do tipo <i>New User Validation</i>	174

B.5	Tabela dos elementos e atributos para mensagens do tipo <i>Session Keep Alive</i>	177
B.6	Tabela dos elementos e atributos para mensagens do tipo <i>New Link Request</i>	180
B.7	Tabela dos elementos e atributos para mensagens do tipo <i>New Link Response</i>	186
B.8	Tabela dos elementos e atributos para mensagens do tipo <i>Link List Request</i>	192
B.9	Tabela dos elementos e atributos para mensagens do tipo <i>Link List Response</i>	198
B.10	Tabela dos elementos e atributos para mensagens do tipo <i>Link Feedback</i> <i>Request</i>	206
B.11	Tabela dos elementos e atributos para mensagens do tipo <i>Command Request</i>	210
B.12	Tabela dos elementos e atributos para mensagens do tipo <i>Command Response</i>	214

Lista de Acrónimos

AAA	<i>Authentication, Authorization and Accounting</i>
ADSL	<i>Asymmetric Digital Subscriber Line</i>
AM	<i>Amplitude Modulation</i>
AOL	<i>America OnLine</i>
API	<i>Application Programming Interface</i>
APP	<i>Atom Publishing Protocol</i>
ASF	Atom Syndication Format
AVI	<i>Audio Video Interleave</i>
BBC	<i>British Broadcasting Corporation</i>
BD	Base de Datos
CATV	<i>Community Antenna Television</i>
CBS	<i>Columbia Broadcasting System</i>
codec	codificador/descodificador
CPU	<i>Central Processing Unit</i>
CRT	<i>Cathode Ray Tube</i>

CSS	<i>Cascading Style Sheets</i>
CSV	<i>Comma-Separated Values</i>
DBMS	<i>Database Management System</i>
DivX	<i>DivX Video Codec</i>
DNS	<i>Domain Name System</i>
DoS	<i>Denial-of-Service attack</i>
DTD	<i>Document Type Definition</i>
DTTV	<i>Digital Terrestrial Television</i>
DTV	<i>Digital Television</i>
DVB Group	<i>Digital Video Broadcasting Group</i>
DVB-C	<i>Digital Video Broadcasting – Cable</i>
DVB-H	<i>Digital Video Broadcasting – Terrestrial for Handheld Devices</i>
DVB-S	<i>Digital Video Broadcasting – Satellite</i>
DVB-SH	<i>Digital Video Broadcasting – hybrid Satellite/terrestrial for vehicles and Handheld devices</i>
DVB-T	<i>Digital Video Broadcasting – Terrestrial</i>
DVD	<i>Digital Video Disc</i>
EPG	<i>Electronic Program Guide</i>
FLV	<i>Flash Video</i>
FM	<i>Frequency Modulation</i>

GPS	<i>Global Positioning System</i>
GSM	<i>Global System for Mobile Communications</i>
H.264	<i>H.264/MPEG-4 AVC video compression</i>
HD	<i>High-Definition</i>
HDTV	<i>High-Definition Television</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
HTTPS	<i>HyperText Transfer Protocol Secure</i>
IDE	<i>Integrated Development Environment</i>
IETF	<i>Internet Engineering Task Force</i>
IMDB	<i>The Internet Movie Database</i>
IMS	<i>IP Multimedia Subsystem</i>
IP	<i>Internet Protocol</i>
IPTV	<i>Internet Protocol Television</i>
ISMA	<i>Internet Streaming Media Alliance</i>
ISP	<i>Internet Service Provider</i>
ITV	<i>Internet Television</i>
ITU	<i>International Telecommunication Union</i>
JS	<i>JavaScript</i>
JSP	<i>JavaServer Pages</i>

JSF	<i>JavaServer Faces</i>
MD5	<i>Message-Digest algorithm 5</i>
LCD	<i>Liquid Crystal Display</i>
MIB	<i>Management Information Base</i>
MMS	<i>Multimedia Messaging Service</i>
MoD	<i>Music on Demand</i>
MP3	<i>MPEG-1 Audio Layer 3 codec</i>
MPEG	<i>Moving Picture Experts Group</i>
MPEG-4	<i>ISO/IEC Moving Picture Experts Group video codec layer 4</i>
NBC	<i>National Broadcasting Company</i>
NGN	<i>Next Generation Network</i>
nPVR	<i>network Personal Video Recording</i>
NTSC	<i>National Television System Committee</i>
NYT	<i>New York Times</i>
NSV	<i>Nullsoft Streaming Video</i>
OLED	<i>Organic Light-Emitting Diode</i>
OO	<i>Orientado a Objetos</i>
P2P	<i>Peer-to-Peer</i>
P2PTV	<i>Peer-to-Peer Television network</i>
PAL	<i>Phase Alternating Line</i>

PC	<i>Personal Computer</i>
PDA	<i>Personal Digital Assistant</i>
PHP	<i>Hypertext Preprocessor</i>
PVR	<i>Personal Video Recording</i>
QoE	<i>Quality of Experience</i>
QoS	<i>Quality of Service</i>
RAM	<i>Random Access Memory</i>
RCA	<i>Radio Corporation of America</i>
RFID	<i>Radio-Frequency IDentification</i>
RGB	<i>Red, Green and Blue color model</i>
RMI	<i>Remote Method Invocation</i>
RSS	<i>Really Simple Syndication</i>
RTSP	<i>Real Time Streaming Protocol</i>
RTP	Radiotelevisão Portuguesa
SATA	<i>Serial Advanced Technology Attachment</i>
SDTV	<i>Standard-Definition Television</i>
SECAM	<i>Séquentiel Couleur Avec Mémoire</i>
SGBD	Sistema Gestor de Base de Dados
SIC	Sociedade Independente de Comunicação
SMIL	<i>Synchronized Multimedia Integration Language</i>

SMS	<i>Short Message Service</i>
SNMP	<i>Simple Network Management Protocol</i>
SO	Sistema Operativo
SQL	<i>Structured Query Language</i>
SSL	<i>Secure Sockets Layer</i>
STB	<i>Set-Top Box</i>
TDT	Televisão Digital Terrestre
TCP	<i>Transmission Control Protocol</i>
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>
TLS	<i>Transport Layer Security</i>
TOD	<i>Television On the Desktop</i>
TV	Televisão (<i>Television</i>)
TVI	Televisão Independente
UE	União Europeia
UHDV	<i>Ultra High Definition Video</i>
UHF	<i>Ultra High Frequency</i>
UMTS	<i>Universal Mobile Telecommunication System</i>
UML	<i>Unified Modeling Language</i>
URI	<i>Uniform Resource Identifier</i>
URL	<i>Uniform Resource Locator</i>

VHF	<i>Very High Frequency</i>
VC-1	<i>Microsoft/SMPTE 421M Video Codec</i>
Vlog	<i>Video blogging</i>
VoD	<i>Video-on-Demand</i>
VoIP	<i>Voice over IP</i>
W3C	<i>World Wide Web Consortium</i>
WTVML	<i>Worldwide TV Mark-up Language</i>
XML	<i>eXtensible Markup Language</i>
XSD	<i>XML Schema Definition</i>
XviD	<i>MPEG-4 Part 2 Advanced Simple Profile (ASP)</i>

Glossário

Anycast Um pacote enviado para um endereço de *anycast* é entregue a um qualquer endereço de um conjunto de endereços que são indicados como possíveis pontos de passagem no percurso do pacote até ao seu destino. Esta arquitectura de *routing* prevê que a informação seja encaminhada pelo destino “mais perto” ou “o melhor” com base na topologia de rede. Em *anycast* existe, à semelhança do *multicast* e *broadcast*, uma associação de um-para-muitos entre o endereço de rede e os destinos possíveis. Cada endereço destino identifica um conjunto de pontos de recepção finais, mas somente um deles é escolhido em cada altura para receber os dados de um qualquer emissor e os transmitir para o próximo nó. Este mecanismo confere balanceamento e redundância relativa aos múltiplos caminhos possíveis na rede para atingir um mesmo destino. (ver figura 2);

Árvore No contexto das Ciências da Computação, uma árvore é uma estrutura de dados em que os mesmos estão dispostos de forma hierárquica. Uma árvore é composta por um elemento principal chamado raiz, que possui ligações para outros elementos, que são denominados de “filhos”. Estes “filhos” podem levar a outros elementos que, por sua vez, também podem possuir “filhos”. Os elementos que não possuem “filhos” são denominados de “folhas” ou nós terminais;

Atom O *Atom* é baseado em *eXtensible Markup Language* (XML) possuindo uma funcionalidade muito semelhante ao conhecido *Really Simple Syndication* (RSS), mas sendo o seu desenvolvimento considerado mais sofisticado do que o do RSS. O nome *Atom* aplica-se a um conjunto de normas associadas. O Atom Syndication

Format (ASF) é uma linguagem XML utilizada para *web feeds*, enquanto que o *Atom Publishing Protocol* (APP) é um protocolo baseado em *Hypertext Transfer Protocol* (HTTP) simples para criar e actualizar recursos *web*. *Web feeds* permitem que programas verifiquem se há actualizações publicadas num determinado *site*. Para fornecer uma *web feed*, o dono ou gestor do *site* disponibiliza um conjunto de *links* para as mesmas *feeds* que não são mais que ficheiros *XML* que vão sendo actualizados pelo *site* traduzindo as alterações relativas a este ao longo do tempo. As *feeds* podem então ser transferidas através do *site* que distribui o conteúdo ou por um programa específico para esta função denominado *Feed Reader* (leitor de *feeds* ou agregador). Este permite aos utilizadores deste protocolo a subscrição de *feeds* e a leitura do seu conteúdo. Uma *feed* pode conter diversos tipos de informações desde notícias textuais, artigos completos, *links* para conteúdo na Internet entre outro tipo de dados sempre acompanhados por dados extra (*metadata*) tais como título, data de publicação da *feed*, autores, assunto, palavras chave, etc. O formato *Atom* foi desenvolvido como uma alternativa ao protocolo RSS. Ben Trott, um defensor deste novo formato, acredita que o RSS possui limitações e falhas, tais como a actual falta de inovação e a necessidade de permanecer retrocompatível, sendo que haveria grandes vantagens em elaborar uma nova concepção [45]. Apoiantes do formato *Atom* formaram o *IETF Atom Publishing Format and Protocol Workgroup*. O formato *Atom Syndication* (ASF) foi publicado como proposta de norma IETF na RFC 4287 e o *Atom Publishing Protocol* (APP) foi publicado como RFC 5023. Este grupo que trabalha neste projecto tem o apoio de grandes empresas como a Google;

BitTorrent É um protocolo de partilha de dados que utiliza uma arquitectura *Peer-to-Peer* (P2P) usado para distribuição de grandes quantidades de dados por um elevado número de utilizadores. *BitTorrent* é um dos protocolos actualmente mais comuns para transferência de arquivos de grande dimensão em todo o mundo. O protocolo *BitTorrent* permite aos seus utilizadores distribuir grandes quantidades de dados sem sobrecarregar nenhum sistema central de armazenamento. Um utilizador ao mesmo

tempo que obtém determinado ficheiro, partilha os “pedaços” que já possui deste e de outros que já possua e deseje disponibilizar na rede. Deste modo, uma determinada partilha não se encontra armazenada num só ponto crítico da rede mas distribuída por múltiplos utilizadores da mesma de forma indiscriminada. Com esta abordagem é possível aumentar a disponibilidade dos dados, a sua acessibilidade, a tolerância a falhas, a largura de banda dos *downloads* possibilitando a transferência simultânea a partir de várias fontes e ainda a robustez e escalabilidade da rede;

bps, Kbps, Mbps; Bps, KBps, MBps Em telecomunicações, o termo *bit rate* ou *data transfer rate* refere-se à quantidade de informação especificada em número médio de *bits* (“b” minúsculo – unidade mínima de informação) por unidade de tempo que é transferida entre dois equipamentos que comuniquem entre si. Esta informação é tipicamente medida em múltiplos da sua unidade (*bit*) por segundo. Um *byte* corresponde a 8 *bits* e representa-se por um “B” maiúsculo. Ainda é possível acrescentar como prefixo multiplicadores como o “K” que significa “Kilo”, “M” que significa “Mega”, “G” com o significado de “Giga”, e assim sucessivamente. Estes multiplicadores podem possuir diferentes valores consoante a unidade que multiplicam, ou seja, sempre que se trata da unidade *bytes* é usado o multiplicador 1024^n enquanto que no caso da unidade *bits* é usado 1000^n . Assim sendo, por exemplo $1Gbps = 1000^3bps$, enquanto que $1GBps = 1024^3Bps$;

Broadcast Em oposição ao *unicast*, a arquitectura *broadcast* refere-se à transmissão de dados (pacotes) numa rede em que esta será recebida (conceptualmente) por todos os dispositivos na rede. Na prática, o alcance de uma transmissão *broadcast* está limitado ao domínio de *broadcast*. Do ponto de vista da associação subjacente ao endereço de rede, esta é do tipo um-para-muitos entre o endereço de rede e os pontos de entrega finais (ver figura 2);

Broadcatching É a denominação utilizada para o acto de descarga de conteúdo digital que foi disponibilizado na Internet utilizando *feeds* RSS. A ideia geral deste termo

é a utilização de um mecanismo automatizado para agregar múltiplas *feeds web* de modo a possibilitar o *download* ou *streaming* de conteúdos solicitados para propósitos de visualização (ver página <http://en.wikipedia.org/wiki/Broadcatching>);

Clustering É uma técnica de *Data Mining* que tem como objectivo criar agrupamentos automáticos de dados (*cluster*) segundo um grau de semelhança. O critério que identifica o grau de semelhança faz parte da definição do problema e depende dos algoritmos utilizados. Os mecanismos de *clustering* podem ser aplicados a diversos tipos de problemas, desde que seja possível extrair desses problemas um padrão nos seus dados;

Data Mining Traduzido normalmente como “Prospecção de dados”, é o acto ou processo de explorar grandes quantidades de dados à procura de padrões consistentes. Estes padrões podem ser dados por regras de associação ou sequências temporais que visam detectar relacionamentos sistemáticos entre variáveis, criando assim novos subconjuntos de dados. Esta temática é relativamente recente em ciência da computação mas utiliza várias técnicas da estatística, recuperação de informação, inteligência artificial e reconhecimento de padrões amplamente estudadas. É normalmente utilizada numa grande variedade de problemas e é transversal a qualquer área de estudo;

Frequência Portadora É um sinal analógico tipicamente sinusoidal que será modulado (alterado) para representar a informação a ser transmitida. A frequência portadora é, geralmente, superior à do sinal modulador que é o sinal que contém a informação a ser transmitida. Os sinais de modulação em frequência (*Frequency Modulation* (FM)) e de modulação em amplitude (*Amplitude Modulation* (AM)) são ambos transmitidos com o auxílio de frequências portadoras específicas. A frequência própria de, por exemplo, uma estação de rádio é na verdade, a frequência central da “portadora”;

Grafo Em Matemática e Ciência da Computação, um grafo é o objecto básico de estudo da Teoria dos Grafos. Tipicamente, um grafo é representado como um conjunto de pontos (vértices) ligados por linhas (as arestas). Dependendo da aplicação, as arestas

podem ser direccionadas, sendo representadas por "setas", ou não direccionadas, sendo representadas por linhas simples;

Jitter e latência *Jitter* é a medida da variação estatística do tempo de entrega de dados numa rede, ou seja, pode ser definida como a medida de variação do atraso sucessivo na entrega de pacotes de características semelhantes. Por outro lado, a latência é a diferença de tempo entre o início de um evento e o momento em que os seus efeitos se tornam perceptíveis. Em redes, a latência de uma ligação é uma medida estatística que mede o atraso médio de pacotes com características semelhantes desde o momento em que são enviados até ao instante em que são recebidos no seu destino;

Kernel O *Kernel* de um sistema operativo é entendido como o núcleo deste ou, numa tradução literal, o seu cerne. Este representa a camada de software mais próxima do *hardware*, sendo responsável por gerir os recursos do sistema computacional como um todo;

Middleware É o termo usado para designar camadas de software que não constituem directamente aplicações, mas que facilitam o uso, coerência e interoperabilidade de aplicações e sistemas distribuídos complexos. A camada de *middleware*, tal como o próprio nome indica, situa-se no "meio" entre aplicações distintas que necessitam de comunicar entre si. Um exemplo comum de *middleware* é o Sistema Gestor de Base de Dados (DBMS) que faz a ponte entre a base de dados fisicamente criada e as aplicações que a desejam aceder através de uma linguagem de alto nível. Aplicações actuais delegam e centralizam serviços comuns em camadas de *middleware* tornando-se mais escaláveis e modulares fomentando a interoperabilidade;

Multithreading Capacidade de suportar a execução de múltiplos fluxos de execução de um mesmo processo (*threads*), de forma concorrente, num mesmo processador, utilizando os recursos de forma escalonada com o objectivo de otimizar o desempenho da execução dos processos. Os *threads* pertencentes a um mesmo processo partilham recursos de sistema como a memória, o processador, a *cache*, etc. Caso o sistema

computacional possua múltiplos processadores, é possível estabelecer um paralelismo real de execução dos *threads* aumentando ainda mais o desempenho dos processos e consequentemente do sistema computacional;

Multicast Em *multicast* a entrega de informação (pacotes) não é feita de forma singular (como no caso do *unicast*) mas sim relativa a um grupo de destinatários que recebem os dados de forma “simultânea”. Esta arquitectura usa uma estratégia muito mais eficiente de entrega de informação quando esta é igual para múltiplos destinatários (por exemplo, *streaming* de um mesmo vídeo em tempo-real), ocupando a ligação de transmissão com um único fluxo de pacotes, criando cópias do fluxo somente quando as ligações para os destinatários se separam. Do ponto de vista da associação subjacente ao endereço de rede, esta é do tipo um-para-muitos entre o endereço de rede e os pontos de entrega finais (ver figura 2);

P2P e P2PTV *Peer-to-Peer* é o nome dado a uma arquitectura de rede para sistemas distribuídos caracterizada pela descentralização de funções na rede. Os elementos numa rede P2P tanto podem realizar funções de servidor como de clientes, partilhando os dados que transferem na rede. O *Peer-to-Peer Television network* é uma técnica de disseminação de conteúdos multimédia (televisão) através da Internet utilizando uma arquitectura P2P, onde é permitido a cada utilizador assistir a conteúdos audiovisuais e ao mesmo tempo partilhar estes com outros utilizadores segundo uma política de rede cooperativa. Este tipo de sistemas permite o balanceamento natural da carga na rede e uma maior redundância e disponibilidade dos dados, face a sistemas cliente–servidor. A figura 2.11 representa as arquitecturas cliente–servidor e P2P, permitindo visualizar as diferenças topológicas entre ambas. A figura 2.12 representa um exemplo esquemático de uma arquitectura *Peer-to-Peer Television network* (P2PTV);

Podcasts e Videocasts Estes termos são utilizados para referir a entrega *online* de conteúdos multimédia *on demand* via *Atom* ou RSS. O termo *videocast* é uti-

lizado para distinguir a distribuição de conteúdos de vídeo dos *podcasts* que mais comumente contém ficheiros áudio. Ambos utilizam a norma RSS como forma de disseminação dos conteúdos, em que o utilizador subscreve a *feed* RSS que lhe interessa e passa assim a receber os *links* actualizados para o tipo de conteúdo pretendido. Utilizando as tecnologias actuais, a subscrição do RSS e visualização do conteúdo respectivo pode ser efectuada utilizando um *Personal Computer* (PC), um dispositivo móvel e até uma *Set-Top Box* (STB) que possua ligação á Internet. Um dos conteúdos frequentemente mais distribuídos na Internet utilizando vídeo *podcasts* (*videocasts*) são as séries televisivas. O termo *enhanced podcast* refere-se a um *podcast* (de áudio) em que é possível mostrar imagens simultâneamente e de forma sincronizada com o áudio. Uma lista de clientes para *podcasts* pode ser encontrada em <http://www.buzzmaven.com/podcast-software-list.html>;

QoS e QoE O termo *Quality of Service*, em português “Qualidade de Serviço”, designa, aplicado a redes de computadores, a capacidade de fornecer um serviço de comunicação (dados, voz, imagem, etc.) de acordo com as exigências impostas em matéria de tempos de atraso/resposta, perda de pacotes na rede e de largura de banda disponível. Por outro lado, *Quality of Experience*, em português “Qualidade de Experiência do Utilizador”, é uma medida subjectiva que visa quantificar a qualidade de um serviço aos olhos dos utilizadores. Este conceito está intrinsecamente relacionado com o conceito de *Quality of Service* (QoS), pois a qualidade de experiência do utilizador pode depender em muito da qualidade de serviço do fornecedor, sendo ambas métricas distintas. O QoS tenta medir objectivamente o serviço prestado por um fornecedor enquanto que o *Quality of Experience* (QoE) está muito dependente de tipo de utilizador. A figura 1 exemplifica, de forma gráfica, os domínios destes dois conceitos;

RSS Tipicamente traduzido como *Really Simple Syndication* e outras vezes como *Rich Site Summary*, é um sistema que possibilita a publicação de alterações frequentes em *web*

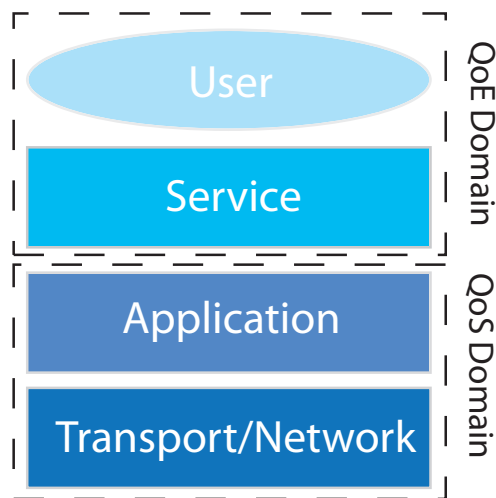


Figura 1: Grafismo exemplificativo dos domínios de QoE e QoS.

sites (páginas) ou serviços como *blogs*, páginas de notícias, serviços de multimédia (áudio e vídeo), entre outros, de uma forma padrão.[40] Um documento RSS (também denominado *feed*, *web feed* ou canal) inclui o texto relativo ao *web site* de uma forma total ou sumária, juntamente com dados extra (*metadata*) como data de publicação da *feed*, autores, assunto, etc. Este mecanismo beneficia quem o disponibiliza quem o subscreve devido a permitir a disseminação da informação de uma forma automática, rápida, cómoda e incisiva em quem está interessado nela. Os utilizadores passam a não ser obrigados a consultar dezenas de *web sites* ou serviços apenas para verificar se existe alguma actualização nos mesmos. Estas *feeds* podem ser lidas utilizando programas denominados de *RSS Readers* (ou *feed reader* ou agregadores). Estes programas permitem a subscrição e leitura de *feeds* e podem ser eles próprios *web sites*/serviços *web*, programas de *desktop* ou programas para dispositivos móveis. O protocolo RSS tem como especificação um formato XML padrão, permitindo que a informação seja publicada uma vez e posteriormente lida em diversos sistemas e programas completamente distintos. O utilizador subscreve então a(s) *feed(s)* que deseja relativas às páginas e assuntos que lhe interessa simplesmente introduzindo o endereço *Uniform Resource Identifier* (URI) da respectiva *feed* ou página no leitor de RSS. Outra forma comum de subscrever uma *feed* RSS é através do *browser*,

pressionando o *icon* relativo ao RSS que, normalmente, permite ao *browser* identificar e iniciar o processo de subscrição. O programa leitor de RSS verifica regularmente e de forma automática todas as subscrições do utilizador, verificando se existe para alguma delas actualizações disponíveis. Este programa permite normalmente não só a visualização do texto presente directamente na *feed* mas também a transferência (automática) de imagens e vídeos relativos a *links* que se encontram na mesma de forma a possibilitar uma visualização completa do seu conteúdo;

Streaming Fluxo de dados multimédia distribuídos aos seus clientes através de uma rede de telecomunicações. Este conceito é frequentemente utilizado para distribuir conteúdos multimédia através da Internet. A utilização de *streaming* possibilita que o utilizador acesse ao conteúdo multimédia de uma forma rápida e faseada em que não necessita de transferir todo o conteúdo para começar a assistir ao mesmo. Tal como o nome indica, trata-se de um fluxo de dados que é usado para apresentar o conteúdo no momento ao utilizador sendo depois os dados, tipicamente, perdidos. A informação pode ser transmitida utilizando diversas arquitecturas, através de *unicast*, *multicast* ou *broadcast*;

Thread Em ciências da computação, um *thread* resulta da divisão do fluxo de execução de um processo em duas ou mais tarefas concorrentes. A implementação dos *threads* pode ser feita ao nível do *kernel* ou ao nível do utilizador. A implementação do mecanismo de *threads* pode diferir de sistema operativo para sistema operativo, mas geralmente, um *thread* está contido num processo. Múltiplos *threads* podem existir dentro de um mesmo processo, partilhando o mesmo espaço de endereçamento de memória (entre outros recursos), podendo desta forma comunicar e partilhar dados através desta. Ao contrário destes, diferentes processos não partilham o mesmo espaço em memória, não podendo assim comunicar através desta região partilhada;

Unicast Em redes de computadores, este termo refere-se ao número de utilizadores alvos de uma transmissão de dados (pacotes) de forma simultânea. São transmissões *uni-*

cast aquelas cujo destino final dos pacotes *Internet Protocol* (IP) seja um único nó na rede. É um exemplo desta arquitectura a transferência de um ficheiro anexado a um e-mail. Do ponto de vista da associação subjacente ao endereço de rede, esta é do tipo um-para-um entre o endereço de rede e o ponto de entrega final (ver figura 2);

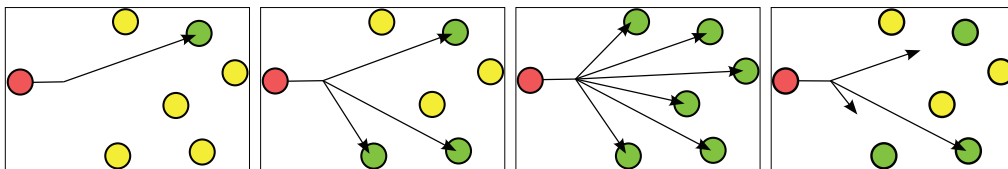


Figura 2: Exemplos de transmissões em *unicast*, *multicast*, *broadcast* e *anycast*, respectivamente.

(Figura baseada em: <http://en.wikipedia.org/wiki/Anycast>)

Web crawler Um *web crawler* é um programa de computador que navega na Internet de forma metódica e automática em busca de um conjunto de dados específicos. Este conceito pode ser traduzido por diferentes termos como indexadores da *web*, indexadores automáticos, *bots*, *worms*, *web spiders* ou *web robot*;

Capítulo 1

Introdução

1.1 Motivação e Contexto

Internet Protocol Television (IPTV) designa genericamente um serviço de distribuição de conteúdos multimédia em tempo real sobre uma rede IP como é o caso da Internet. As fontes de conteúdos multimédia são cada vez mais abundantes (como por exemplo: *Videocasts*, Vídeos YouTube, etc.) mas também cada vez mais heterogéneas e dispersas. A dispersão e variedade de conteúdos torna mais difícil fazer com que todos estes cheguem ao seu público alvo. Para além disso, os utilizadores têm cada vez mais dificuldade em obterem os conteúdos multimédia que mais se adaptam às suas preferências sem que para isso “esbarrem” em conteúdos que não lhes interessam. Existe assim a necessidade de desenvolver uma infraestrutura de classificação de conteúdos e utilizadores bem como de distribuição personalizada desses conteúdos que os torne mais acessíveis.

Por outro lado, as empresas de telecomunicações estão a investir cada vez mais para oferecer aos seus clientes serviços de *triple-play* e, começando já a antever a fundamental integração com serviços móveis, serviços *quadruple-play* (ver secção 2.2.5). Este facto faz com que haja uma convergência ao nível da rede para o protocolo IP, de forma a oferecer múltiplos serviços sobre a mesma. Numa rede *all-IP*, serviços personalizados e interactivos podem ser oferecidos muito facilmente aos seus utilizadores. Existe assim um forte potencial de integração e convergência de várias tecnologias e serviços através da

Internet como rede IP global, em que os sistemas IPTV poderão vir a ter uma posição preponderante, criando novos conceitos e novas oportunidades de negócio.

Na infraestrutura referida anteriormente, o sistema servidor deverá manter uma listagem dos conteúdos multimédia residentes noutros sistemas (servidores), actualizada via RSS ou outras tecnologias como, por exemplo, as *Application Programming Interfaces* (APIs) genéricas ou específicas de acesso aos vários provedores de conteúdos, caso estas existam e estejam disponíveis. Deverá também manter uma classificação dinâmica dos conteúdos multimédia que se encontram na sua base de dados, classificação essa construída e refinada de forma gradual em função das interacções dos clientes com os conteúdos multimédia no sistema. Para além disso, o sistema deverá manter um perfil dos utilizadores e das reacções dos mesmos aos diferentes conteúdos (*feedback*), de modo a esculpir dinamicamente as preferências destes.

Esta característica visa agregar o máximo número de conteúdos possível, distribuídos através de diferentes serviços e provedores. Actualmente, cada provedor de conteúdos multimédia possui os seus próprios conteúdos e meios de os distribuir, bem como *softwares* cliente específicos que somente são compatíveis com o seu serviço. Isto leva a uma dispersão dos conteúdos visto a interacção entre os diferentes provedores ser diminuta e logo ser necessário diferentes sistemas ou interfaces para aceder aos conteúdos multimédia disponibilizados.

A interacção com o sistema projectado será efectuada através de um cliente IPTV construído para o efeito, preferencialmente transversal a qualquer tecnologia e dispositivo no mercado. Este cliente IPTV deve permitir ao utilizador aceder a todos os conteúdos sequencialmente ou através de *playlists* personalizadas (programação) contendo os mesmos. A visualização de conteúdos “em tempo real” ao estilo da “televisão convencional” também deverá ser possível, com o acréscimo de um conjunto de funcionalidades que permitam um elevado grau de controlo e interactividade com os conteúdos apresentados.

Os conteúdos recomendados poderão ser sugeridos tanto pelo sistema como por outros utilizadores, assim como o utilizador poderá sugerir conteúdos a terceiros. Realizar a procura de um conteúdo específico com recurso a palavras-chaves e critérios múltiplos, é

uma funcionalidade natural neste tipo de sistemas e que o servidor IPTV implementado contempla actualmente.

Não tão importante mas não menos interessante é a possibilidade de receber e efectuar comentários aos conteúdos vistos pelo próprio e por outros utilizadores, assim como a existência de vários modos de visualização. Estes modos de visualização permitem criar sub-perfis do utilizador tendo em conta a sua companhia no momento, o ambiente em que se encontra, o nível de controlo parental, entre outros.

Em suma, no que respeita à motivação e contexto, um sistema deste tipo poderá integrar um enorme conjunto de funcionalidades pertencente a áreas completamente disjuntas, funcionando como elemento transversal a estas, aglutinando novas funcionalidades e reinventando as actualmente existentes. Todas essas funcionalidades e serviços serão acessíveis através de uma rede IP (a Internet), criando uma abordagem *all-IP*.

1.2 Objectivos

De uma forma genérica, os objectivos desta Dissertação de Mestrado centraram-se na projecção e implementação de um protótipo funcional de um servidor IPTV, no que diz respeito à agregação, gestão e disponibilização de conteúdo multimédia existente na Internet. A criação de condições para a futura implementação de métodos de classificação dinâmica do perfil do utilizador e classificação indirecta de conteúdos com vista à sugestão automática dos mesmos ao utilizador também foi considerada.

De uma forma mais objectiva, os pontos propostos podem ser sintetizados em:

1. Estudar as diferentes fontes de conteúdos multimédia existentes;
2. Estudar o actual estado da tecnologia no que respeita a sistemas de televisão mais especificamente sistemas IPTV;
3. Estudar as diferentes tecnologias de implementação de sistemas que comunicam em rede;

4. Desenvolver uma metodologia inicial de classificação dinâmica de conteúdos e utilizadores;
5. Desenvolver um desenho de base de dados que permita a persistência de toda a informação necessária ao sistema;
6. Desenvolver o protocolo de interacção cliente-servidor em rede *Transmission Control Protocol/Internet Protocol* (TCP/IP);
7. Desenvolver o servidor de agregação e composição de conteúdos utilizando como fontes *feeds* RSS;
8. Desenvolver um protótipo funcional do servidor IPTV integrando os vários módulos funcionais;
9. Realizar testes gerais de integração e desempenho do sistema com clientes genéricos.

Tal como já foi referido, foram criadas as condições para que este sistema permitisse não só a classificação dinâmica do utilizador mas também uma aprendizagem dinâmica inteligente do perfil. Para isso, é possível utilizar dados como o correlacionando dos utilizadores no sistema com perfis semelhantes, grupos virtuais de amigos, *feedbacks* e, se disponível, dados dinâmicos provenientes do dispositivo (móvel ou fixo) cliente como a posição *Global Positioning System* (GPS) do utilizador, a temperatura ambiente, condições meteorológicas, entre outros. Deste modo, seria possível implementar um sistema de sugestão de conteúdos multimédia que iria de encontro às necessidades específicas e locais do utilizador, sendo estas inerentemente dinâmicas e voláteis.

1.3 Estrutura e Organização desta Dissertação

Esta Dissertação de Mestrado está organizada em capítulos. Cada capítulo possui secções de temática mais específica, e estas, por sua vez, podem conter subsecções. Todos estes elementos estruturais podem conter texto relativo à temática respectiva, tabelas de dados e grafismos da mais variada ordem.

É parte integrante deste documento um índice dos conteúdos ordenado pelo número da página, uma lista de figuras, uma lista de tabelas, uma lista de acrónimos e um glossário com alguns termos e expressões que podem não ser do conhecimento geral, estando este elementos presentes antes do actual capítulo.

Em termos de conteúdo, é feita uma introdução geral no capítulo 1 aos tópicos abordados e subjacentes à temática desta dissertação, na qual consta esta descrição da estrutura da mesma.

De seguida é possível encontrar aquilo que em investigação científica é normalmente denominado por “Estado da Arte”, mas que aqui, devido a não possuir um carácter tão formal e elaborado, foi denominado de “Conceitos Base e Estado Actual da Tecnologia” (capítulo 2). Este capítulo faz uma abordagem quer histórica quer contemporânea dos tópicos de maior importância relacionados com o trabalho desenvolvido.

Posteriormente, é possível encontrar o primeiro capítulo que se refere especificamente ao trabalho desenvolvido no âmbito desta Dissertação de Mestrado. O seu nome é “Arquitectura do Servidor: Análise Funcional” (capítulo 3). Aqui são apresentados de uma forma breve e geral a arquitectura do sistema, sendo que a especificação exaustiva de cada módulo que o constitui é enunciada no capítulo 4. É também especificado neste capítulo os requisitos propostos que levaram à adopção da arquitectura do sistema, bem como enunciada a metodologia geral do modelo de comunicação.

O capítulo 4, cujo nome é “A Arquitectura do Servidor: Análise Modular”, especifica e descreve de forma mais profunda e exaustiva cada módulo funcional desenvolvido que compõem o sistema projectado. Desde especificações técnicas até aos diagramas comportamentais, passando pela descrição das metodologias associadas, é possível encontrar neste capítulo todas as especificidades de cada módulo construído. Para um maior conhecimento das especificidades do sistema ao nível da programação subjacente, é favor consultar a documentação *Javadoc* fornecida com o código do servidor IPTV.

Dado que durante a construção do servidor IPTV foi tido sempre como objectivo criar código o mais genérico e robusto possível, foi facilmente implementada, a partir das bibliotecas *Java* construídas para o servidor, uma API de comunicação com o servidor que

funciona em total harmonia com este. O capítulo que descreve essa mesma API é o capítulo 5 (cujo nome é “API para Cliente IPTV em Java”). É de notar que à semelhança do que acontece com o servidor IPTV, foi também criada toda a documentação referente ao código da API, podendo esta documentação ser encontrada junto do código da mesma.

Os penúltimos dois capítulos referem-se, respectivamente, aos requisitos e procedimentos para instalação do sistema em uma única máquina (forma centralizada) ou em múltiplas máquinas (forma distribuída) e aos testes gerais de desempenho e comportamento.

O último capítulo antes dos anexos apresenta algumas conclusões retiradas do desenvolvimento e pesquisa efectuada no âmbito deste trabalho e apresenta também algumas propostas e ideias de trabalho futuro, tanto ao nível do servidor IPTV como também ao nível do cliente IPTV.

No final deste documento, em termos estruturais, podem ser encontrados os anexos ao trabalho. Estes especificam detalhes mais técnicos e específicos relativos ao sistema desenvolvido. O anexo A é referente ao módulo *IPTV Database*, o anexo B especifica as mensagens de comunicação cliente<->servidor utilizadas pelo módulo *IPTV Server Core* e por fim o anexo C foca a utilização da API em *Java* para estabelecimento da comunicação entre um cliente IPTV genérico e o servidor implementado.

O anexo B é o mais longo de todos os anexos e contém informações muito importantes sobre as mensagens construídas. Por este facto, a sua leitura é especialmente aconselhada, visto permitir a obtenção de um conhecimento mais bem fundamentado sobre alguns pormenores importantes do servidor desenvolvido.

É de notar que este documento encontra-se recheado de *hyperlinks* para recursos *web* considerados estáveis e fidedignos para complementar o entendimento e esclarecimento dos conceitos associados. Como será de esperar, na versão impressa deste documento estes são inúteis. Assim sendo, aconselha-se a leitura do mesmo na versão digital, até porque é ambientalmente menos dispendioso.

1.4 Visão Geral

Os sistemas de televisão digital, *Digital Television* (DTV), enquadram-se numa das mais significativas evoluções da televisão nos últimos anos que actualmente permitem ao utilizador usufruir de uma experiência multimédia com uma superior qualidade visual quando comparados com sistemas analógicos. Mesmo depois de todas as recentes e importantes evoluções tecnológicas nesta área, ainda existe uma significativa falta de agregação e heterogeneidade de fontes de conteúdos multimédia, alguma carência de convergência, diversidade e interoperabilidade dos meios nos quais podemos assistir aos conteúdos, bem como bastantes lacunas no campo da interacção e personalização dos conteúdos.

Neste âmbito, esta Dissertação de Mestrado visou projectar e implementar um protótipo de um sistema servidor IPTV que se prende com a agregação e gestão de uma multiplicidade e diversidade de conteúdos multimédia existentes na Internet (com a possibilidade de se estender a serviços de multimédia dedicados). Esta agregação e gestão de conteúdos tem como objectivo final a disponibilização dos mesmos, de forma inteligente, a um sistema cliente IPTV que irá tratar de todo o processo relativo à acessibilidade dos conteúdos multimédia. Este acesso aos conteúdos será feito num ambiente de utilização orientado à convergência, interoperabilidade e mobilidade dos meios de visualização, através de uma abordagem heterogénea e dinâmica em termos de *hardware* e necessidades do utilizador final. Para isso, a construção do servidor recorreu a uma arquitectura modular distribuída, pensada para operar sobre uma rede TCP/IP, com múltiplos módulos de funções específicas, cujo o elemento central e agregador destes é o sistema de base de dados. Esta Base de Dados (BD) contém toda a informação imprescindível ao funcionamento de cada módulo (ou sub-sistema) e, conseqüentemente, do sistema no seu todo.

Ao longo de toda esta Dissertação, o sistema implementado será sempre apelidado de “sistema IPTV”, “serviço IPTV” ou “servidor IPTV”. Não foi definido nenhum nome específico para o sistema/aplicação/serviço desenvolvido, apesar de terem surgido alguns. A título de curiosidade e por ordem cronológica, alguns dos nomes que surgiram para o sistema foram: *Universal TV*, *iTV4You*, *iTV4All*, *iTVBox*, *iBox4You*, *iNTV* - *intelligent*

Network TeleVision, *iNMedia*, *iNBox*, Arca de Noé - por conter tudo-em-um (*all-in-one*), *MyTEA* (*on my way*) - *My Television Everywhere, Anywhere* (*on my way*), entre outros.

Foram criados quatro sub-sistemas funcionais e um sub-sistema de gestão remota dos anteriores. Cada um dos quatro módulos tem somente uma dependência externa com o sistema, sendo esta com a BD. Segue-se uma descrição muito sucinta das funcionalidades de cada um.

O módulo que trata de toda a lógica de comunicação e sinalização XML sobre HTTP ou *HyperText Transfer Protocol Secure* (HTTPS) entre o cliente e o servidor IPTV é o “Núcleo do Servidor IPTV” – *IPTV Server Core* (Secção 4.2). Este módulo é exclusivamente passivo, ou seja, responde sempre a uma mensagem recebida mas nunca envia uma mensagem sem receber uma solicitação para isso. A sua lógica de operações interna foi concebida segundo uma política *multithreading* em que cada mensagem recebida dá origem ao início de um novo *thread* prestador de serviço que termina quando as tarefas a efectuar estiverem todas concluídas e a resposta à solicitação for enviada ao respectivo cliente. É garantida a entrega das mensagens através da utilização do protocolo HTTP que funciona sobre *Transmission Control Protocol* (TCP). Caso um ou mais clientes enviem uma ou mais mensagens simultaneamente, a exclusão mútua das operações e actualizações a realizar na BD é garantida ora pelo Sistema Gestor de Base de Dados (SGBD) (ou *Database Management System* (DBMS)) ora pelo próprio sistema através do referido módulo *IPTV Server Core*.

O módulo “Fontes de *Links* para Conteúdos Multimédia” – *Content Link Source* (Secção 4.3) captura e armazena na BD de forma automática os *links* para os conteúdos multimédia e toda a informação existente associada a estes (autor, data, tipo de conteúdo, descrição, etc.) bem como dá início à sua caracterização fazendo uso da informação inicialmente disponível. Actualmente, este sub-sistema utiliza como forma de pesquisa e acesso às fontes de conteúdos as APIs dos provedores de conteúdos multimédia directamente suportados (por exemplo, o *Youtube*) e, de forma completamente genérica, *feeds* RSS (versão 2.0). A utilização do protocolo RSS permite estabelecer a comunicação de forma padrão entre este módulo funcional do servidor IPTV e qualquer fonte de conteúdos multimédia. É

de salientar que a arquitectura do sistema contempla que um *link* para um novo conteúdo possa ser inserido não só de forma automática pelo próprio sistema fazendo uso do mecanismo anteriormente mencionado, mas também directamente por um utilizador registado. No caso de ser o utilizador a introduzir um *link* para um novo conteúdo, é solicitado a este o fornecimento de algumas informações base sobre o conteúdo de modo a iniciar a sua caracterização e classificação pelo sistema.

“Aprendizagem do Perfil do Utilizador” – *User Profile Learning* (Secção 4.4) é o subsistema capaz de interagir com a BD a fim de criar e refinar a informação relativa ao perfil do utilizador através da sua interacção com o sistema. Mesmo não sendo um objectivo desta Dissertação de Mestrado, o sistema foi projectado tendo em conta a utilização de uma componente muito interessante no domínio da distribuição de conteúdos (bem como em outras áreas bastante diversificadas), que é a *aprendizagem dinâmica e inteligente de perfis dos utilizadores*. Esta aprendizagem dinâmica assenta tipicamente na correlação de toda a experiência de utilização do utilizador no sistema. É considerado pelo servidor informações distintas como o *feedback* discreto relativo aos conteúdos vistos pelo utilizador, a correlação com outros utilizadores do sistema com perfis semelhantes, a correlação com os membros do seu grupo virtual de amigos (caso este exista), entre outros factores que possam pesar na construção do perfil do respectivo utilizador. Estes dados permitem situar as preferências do utilizador no centro de um sub-espço no universo de todas as temáticas consideradas passíveis de possuir conteúdos multimédia.

Esta construção do perfil do utilizador tem como aplicação directa a sugestão automática de conteúdos seguindo uma metodologia muito mais inteligente do que a utilização única de modelos matemáticos de carácter probabilístico através da correlação do histórico dos conteúdos acedidos pelos utilizadores. Em Engenharia, muitas vezes a solução para os problemas mais complexos reside no mais trivial dos processos quotidianos. Analisemos um exemplo do comportamento social humano que corresponde à oferta de um presente numa qualquer situação festiva. Tipicamente, para oferecer o presente “ideal” que seja útil e simultaneamente economicamente viável, possuindo *a priori* um vasto leque de opções passíveis de serem escolhidas, primeiro é necessário conhecer o destinatário da prenda

relativamente bem no que diz respeito à sua personalidade. Deve saber-se também o enquadramento da oferta, a época do ano gregoriano, o enquadramento social e pessoal do visado, saber o que, genericamente, o sujeito já possui e necessita actualmente, o que outras pessoas pensam oferecer nessa mesma ocasião, saber um pouco da situação psicologia actual da pessoa, entre outras condicionantes. Reportando-nos à “oferta” de um conteúdo multimédia, podemos utilizar como analogia a situação anterior com as devidas alterações e simplificações, sendo que em ambas as situações temos algo nuclear em comum: aquisição e sintetização de conhecimento sobre a personalidade da pessoa e estabelecimento de restrições dinâmicas dependentes do contexto. Assim, relativamente a esta componente de sugestão automática de conteúdos, o sistema contempla uma grande diversidade de fontes de informação de apoio à decisão dos conteúdos a sugerir. Estas fontes vão desde o conhecimento das áreas de interesse do utilizador, até, caso esteja disponível, a actual posição GPS do mesmo, passando por dados tão diversificados como as condições meteorológicas da sua localização, o *hardware* de visualização que detém no momento, conhecimentos linguísticos do mesmo ou o seu sexo, idade, profissão, hora do corrente dia em que o serviço é requisitado, entre outros.

O sistema prevê guardar todos estes dados devidamente organizados para posterior relacionamento e sintetização. Esta informação, em conjugação com a informação que a BD detinha anteriormente para o mesmo utilizador, servirá para o sistema refinar a base de dados de conhecimento sobre o utilizador em questão num ambiente de utilização completamente dinâmico.

A construção de um só módulo contendo ambas as funcionalidade de sugestão automática de conteúdos e aprendizagem do perfil do utilizador deveu-se ao facto de ambas estarem profundamente relacionadas, sendo que estas são lógica e matematicamente separáveis. É de notar que a sugestão de conteúdos específicos a um utilizador em particular também pode ser efectuada de forma directa entre utilizadores registados no sistema desde que estes pertençam a um mesmo grupo virtual de “amigos”.

O sub-sistema funcional que pode ser considerado como o “ponto crítico” de todo este servidor, devido a possuir funções de agregação e centralização de informação fundamentais,

chama-se “Base de Dados do Servidor IPTV” – *IPTV Database* (Secção 4.5). Esta é a BD relativa a toda a informação no sistema. O SGBD utilizado foi o MySQL[37], fazendo uso de uma política de acesso Orientado a Objectos (OO) através da biblioteca *Java Hibernate* [32] [22] para o mapeamento objecto-relacional escrito na linguagem *Java* [43] [33], o que torna a API de acesso à base de dados construída em *Java* [43] [33] independente do *SGDB* utilizado, conferindo-lhe portabilidade.

O módulo de gestão remota de todo o sistema, permite o controle e monitorização de cada módulo funcional. Este denomina-se “Ferramenta de Gestão do Servidor IPTV” – *IPTV Server GUI* (Secção 4.2) e não era um objectivo inicial desta Dissertação de Mestrado mas, devido à necessidade de controlo e monitorização remotas de todos os componentes do servidor através de uma interface de alto nível, foi implementado de uma forma simplificada. Este segue uma política de aplicação para *desktop*, permitindo o controle e monitorização de todo o servidor de uma forma simples e intuitiva. Construído utilizando *Java* [43] [33] através das suas bibliotecas de elementos gráficos *Swing* e *SwingX*, usufruindo de uma política de comunicação baseada em *Remote Method Invocation* (RMI) de forma directa, permite o controlo e monitorização multi-utilizador em tempo real de um módulo em particular ou de vários simultaneamente.

O sistema projectado não persiste qualquer conteúdo multimédia no que respeita ao conteúdo em si, indexando somente os meta-dados relativos a estes na base de dados do servidor. Para além disso, o sistema não contempla mecanismos de codificação, compressão e optimização dos *streamings* multimédia nem qualquer política de gestão de transporte dos mesmos na rede, ficando estes a cargo de outros sistemas a projectar ou já existentes. Assim o servidor projectado vai ao encontro de um dos objectivos desta abordagem: um sistema de forte componente distribuída.

Capítulo 2

Conceitos Base e Estado Actual da Tecnologia

2.1 Introdução

Neste capítulo serão explicados alguns conceitos base relativos às temáticas inerentes a este trabalho bem como mencionado o estado actual da tecnologia no que respeita a sistemas de televisão. Assim sendo, é possível encontrar aqui informações relativas aos mais recentes desenvolvimentos no âmbito dos sistemas de televisão, mais especificamente, sistemas de televisão digital (DTV), televisão de alta qualidade (*High-Definition Television* (HDTV)), *Internet Television* (ITV) e, obviamente, IPTV.

Como não se pode falar do presente e das mais recentes evoluções de algo de forma completa e coerente sem referir as bases construídas no passado, será também enunciado de uma forma sucinta as características e os acontecimentos históricos mais importantes relativos aos sistemas analógicos de televisão.

Neste capítulo será ainda mencionado o significado e características genéricas de um sistema de IPTV, a sua história, bem como apresentadas algumas das vantagens e desvantagens desta tecnologia face as restantes existentes. Para além disso, são mencionadas algumas funcionalidades que os sistemas IPTV poderão integrar conferindo-lhes uma maior utilidade e aplicabilidade, concentrando vários serviços e tecnologias num só sistema.

2.2 Tecnologias de Televisão

2.2.1 Televisão Analógica e História da TV em Portugal

Foi com o primeiro sistema de televisão analógica que surgiu o conceito de “televisão” que viria mudar a forma como as pessoas viam o mundo, interagiam com ele, e também viria condicionar as invenções e evoluções tecnológicas subsequentes. A palavra “televisão” tem como origem etimológica “tele” do Grego, cujo significado é “distante” e “visione”, do Latim, que significa “visão”. Esta é muitas vezes usada em Português quer para designar o aparelho electrónico receptor de som e imagens (televisor, aparelho de televisão ou sistema de televisão) quer para designar a entidade que é responsável pela emissão dos conteúdos (estação televisiva). Seja um sistema analógico ou digital, o conceito de Televisão (*Television*) (TV) explora as propriedades do olho humano, criando a ilusão de movimento devido à apresentação de “fotografias” sequenciais (chamados *frames*) a uma taxa de actualização que impossibilita o olho humano de ver as transições devido à persistência de visão relativamente à imagem anteriormente vista.

O televisor (ou sistema de TV como será chamado nesta Dissertação por ser mais genérico) nasceu como sendo um dispositivo electromecânico analógico, passando a sistema electrónico analógico (com a invenção das válvulas electrónicas e dos transístores), e actualmente é, no caso dos dispositivos mais recentes, um dispositivo electrónico completamente digital. Sistemas de TV analógicos sofreram, ao longo da sua evolução, grandes modificações na forma como eram fabricados, passando de *Cathode Ray Tubes* (CRTs) a sistemas de *retroprojectão* e depois a *Liquid Crystal Displays* (LCDs), sendo que a sua metodologia de funcionamento interno e formas de actualização das imagens permaneceram semelhantes.

Do ponto de vista funcional, os primeiros televisores completamente electrónicos (que eram CRTs a preto-e-branco), eram constituídos essencialmente por uma caixa (madeira ou plástico), um cinescópio (formado por um painel frontal em vidro (ecrã), uma máscara de sombra, um cone de vidro, um canhão de electrões, um cone metálico interno e uma

bobine de deflexão), um circuito controlador de todos os componentes, alguns botões e potenciómetros e uma antena.

Todo o processo analógico começa na captura das imagens da cena a transmitir através de uma câmara de filmar analógica, cujo mecanismo baseia-se na análise e conversão da luz e do som em sinais eléctricos. Estes sinais são posteriormente convertidos e modelados em ondas electromagnéticas segundo uma frequência portadora que permitirá seleccionar um sinal específico (sintonizar) de entre os restantes que possam existir. As ondas electromagnéticas por sua vez são emitidas por antenas emissoras específicas segundo essa mesma frequência portadora que define um canal de TV específico. Numa outra posição geográfica completamente distinta (mas no raio de alcance da emissão) encontra-se um ou mais receptores que recebem “em tempo real” de forma idêntica os sinais emitidos. Ao receber estes sinais através das suas antenas, estes dispositivos receptores procedem à reconversão das ondas electromagnéticas em sinais eléctricos que, posteriormente, são utilizado pelo aparelho para serem novamente transformam em sinais luminosos de modo a formar num ecrã uma replica da imagem capturada inicialmente pela câmara de filmar. A transmissão dessas mesmas ondas electromagnéticas de televisão desde o emissor até aos receptores, foi padronizada segundo duas secções do espectro electromagnético: *Very High Frequency* (VHF) e *Ultra High Frequency* (UHF). A figura 2.1 representa o espectro electromagnético.

Reportando-nos à história da televisão, foi na cidade de Londres em Fevereiro de 1924 que o primeiro sistema semi-mecânico de televisão analógica foi demonstrado. Posteriormente, foi possível demonstrar imagens em movimento num sistema semelhante ao anterior a 30 de Outubro de 1925. Um sistema de televisão electrónico completo foi apresentado por John Logie Baird, Philo Farnsworth e Philo Taylor Farnsworth em 1927. A primeira estação televisiva analógica foi inaugurada em Nova Iorque a 11 de Maio de 1928 e ficou a cargo da WGY em Schenectady. O primeiro serviço considerado na altura como “de alta definição” apareceu na Alemanha em Março do ano de 1935, mas a sua disponibilidade estava muito restrita. Uma das primeiras grandes transmissões televisivas foi dos Jogos Olímpicos de Berlim de 1936. A disseminação e uso da televisão aumentou de forma ex-

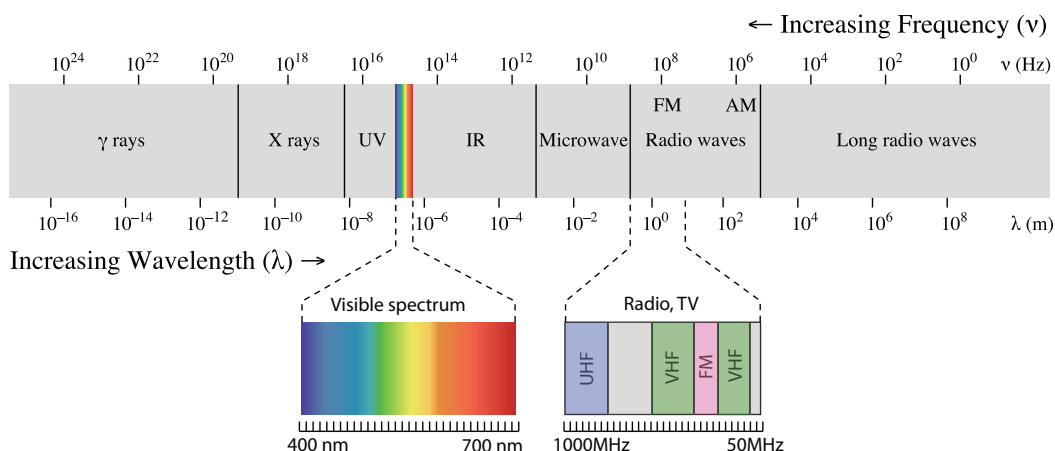


Figura 2.1: Espectro electromagnético.

(Figura baseada em: http://en.wikipedia.org/wiki/File:EM_spectrum.svg)

ponencial depois da Segunda Grande Guerra Mundial devido aos avanços tecnológicos que esta propiciou. Como já vem fazendo parte da historia, é em períodos de crise e guerra que os maiores e mais importantes avanços tecnológicos se conseguem, bem como as mais importantes invenções são materializadas, como é o caso da Internet, que surgiu a partir de pesquisas militares nos períodos áureos da Guerra Fria na década de 60.

A televisão a cores surgiu em 1954 na rede norte-americana *National Broadcasting Company* (NBC). Um ano antes o governo dos Estados Unidos da América aprovou o sistema de transmissão analógica a cores proposto pela rede *Columbia Broadcasting System* (CBS), mas quando a *Radio Corporation of America* (RCA) apresentou um novo sistema que mantinha a retrocompatibilidade com os aparelhos posteriores a preto-e-branco, a CBS abandonou a sua proposta em favor da nova. Este sistema proposto pela RCA adicionava informação relativa à Saturação (*saturation*) e *Hue* da cor ao sinal básico monocromático de forma que os dispositivos a preto-e-branco ignorassem esta modificação. No que respeita à codificação dos sinais analógicos da “nova” TV a cores transmitidos por meio das ondas electromagnéticas, temos os conhecidos NTSC, PAL ou SECAM. Cada país optou pela sua tecnologia de codificação dos sinais, dando origem à distribuição mostrada na figura 2.2 relativa ao século XX.

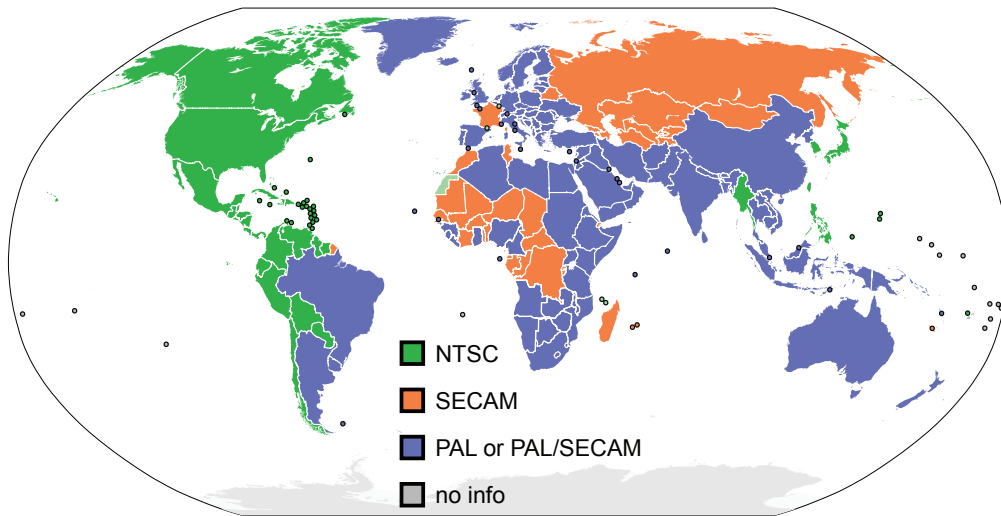


Figura 2.2: Sistemas de codificação televisiva a cores agrupados por país. (século XX).
(Figura retirada de: <http://pt.wikipedia.org/wiki/Ficheiro:PAL-NTSC-SECAM.svg>)

Também no que respeita ao som dos dispositivos receptores de TV, existiu uma evolução associada, tendo-se transitado do som mono-canal (um canal de áudio), para um som *stereo* (dois canais de áudio). Em 1960 a japonesa SONY introduz no mercado os receptores de TV com transístores substituindo a tecnologia posterior que utilizava válvulas electrónicas. Dois anos mais tarde, em 1962, o satélite Telstar inicia a transmissão de sinais de televisão sobre o Oceano Atlântico. A capacidade de tornar os dispositivos de TV muito mais pequenos e portáteis chegou em 1979 quando a Matsushita (actual Panasonic) registou a patente da televisão de bolso com um aspecto que é o utilizado nos nossos dias: um ecrã plano.

Com o passar dos anos assistiu-se a uma constante evolução tecnológica no campo dos monitores de vídeo que continua nos dias de hoje. Esta evolução é actualmente fomentada não só pela aplicação em sistemas de TV mas também em sistemas computacionais, dispositivos móveis ou em áreas tão distintas como a decoração de espaços ou a iluminação artística. Desde os primeiros televisores CRT a preto e branco até aos *Organic Light-Emitting Diodes* (OLEDs) e OLED de matriz activa muito finos, flexíveis e potencialmente muito baratos já existentes no mercado, passámos pelos sistemas CRT a cores, monitores

de retroprojectão, LCDs e monitores de plasma, evoluindo constantemente no que respeita à qualidade de imagem, área de visualização, eficiência e espessura dos mesmos. A figura 2.3 representa os passos mais significativos da evolução dos sistemas de TV.



Figura 2.3: Exemplo gráfico da evolução dos sistemas de televisão.

No que respeita à evolução da televisão em Portugal, esta deu os primeiros passos, ainda a preto e branco, a 4 de Setembro de 1956. As emissões regulares tiveram início a 7 de Março de 1957. Nessa altura, ainda só podia ser captada na região de Lisboa. Nos anos seguintes, a Radiotelevisão Portuguesa (RTP) chegaria ao Porto, à Madeira e aos Açores, e depois cobriria todo o território nacional, com delegações nas diversas regiões do país. Em 1968 tiveram início as emissões do segundo canal da RTP denominado de RTP2.

As fases da evolução da televisão em Portugal confundem-se com os ciclos políticos, económicos e sociais. Durante o Estado Novo, a RTP pertencia ao Estado, à Igreja e à Rádio Renascença, sendo o aparelho ideológico do regime. Depois do 25 de Abril de 1974, procedeu-se à nacionalização da empresa. Foi através da RTP que o país assistiu ao maior duelo político a seguir ao 25 de Abril, a 6 de Outubro de 1975 e durante quatro horas, entre o líder do PS, Mário Soares, e o líder do PCP, Álvaro Cunhal. É em 1977 que se inaugura a era das telenovelas, com a transmissão da telenovela brasileira “Gabriela”, que fez literalmente parar o país. A transmissão a cores começou em 1980.

Em 1992, com Cavaco Silva como primeiro-ministro, arrancaram as emissões da Sociedade Independente de Comunicação (SIC), o primeiro canal privado de TV Português, e em 1993 tiveram início as emissões da Televisão Independente (TVI), canal também privado e inicialmente de inspiração religiosa cristã. Estes canais conseguiram em pouco tempo impor mudanças no estilo da informação e da programação em geral.

A transmissão da televisão por cabo surgiu em Portugal em 1994 e permitiu o aparecimento de mais canais televisivos nacionais e estrangeiros, proporcionando ao telespectador um leque mais variado de escolhas televisivas.

Para completar a enumeração dos principais acontecimentos na história da televisão em Portugal, mais recentemente em Outubro de 2008 tiveram início as primeiras emissões experimentais do serviço de Televisão Digital Terrestre (TDT) (secção 2.2.2) a partir do emissor de Palmela. O arranque oficial das emissões do serviço de TDT em Portugal deu-se em 29 de Abril de 2009, sendo que a *Portugal Telecom* ganhou o concurso para a implementação desta (ver página <http://tdt.telecom.pt/>).

A IPTV (secção 2.2.5) surgiu em Portugal pela primeira vez em 2006, com o serviço *SmarTV* da *Clix* (ver página [web http://acesso.clix.pt/](http://acesso.clix.pt/)). Mais tarde, em 2007, surgiu o *Meo*, serviço de IPTV da *Portugal Telecom* (ver página [web http://www.meo.pt/](http://www.meo.pt/) criado depois da separação forçada entre a *PT Comunicações* e a *PT Multimédia*, cuja última mudou o seu nome para *Zon Multimédia*, com o objectivo de combater a mais recente adversária. O serviço *Meo* foi disponibilizado numa primeira fase em Lisboa, Porto e em Castelo Branco, sendo que actualmente já se encontra disponível em várias localidades do país através de IPTV fazendo uso das linhas telefónicas e disponível em todo o país via satélite (ver página [web http://wapedia.mobi/pt/Meo](http://wapedia.mobi/pt/Meo)). Em Julho de 2009, a *Vodafone* lançou o serviço de IPTV designado *Vodafone Casa TV* (ver página [web http://www.vodafone.pt/main/Particulares/vodafonecasa/IPTV/](http://www.vodafone.pt/main/Particulares/vodafonecasa/IPTV/)) com condições de adesão e mensalidade bastante atractivas face às restantes operadores, o que veio relançar este mercado.

No endereço <http://www.youtube.com/watch?v=CkyTL7d9xlE> é possível encontrar um vídeo da autoria da revista *Exame Informática* comparativo entre o serviço *SmarTV* da *Clix* e o *Meo* da *Portugal Telecom*.

Voltando ao tema da televisão analógica, as principais desvantagens e problemas desta tecnologia de televisão face aos sistemas IPTV são:

- Limitações físicas da electrónica analógica no que respeita à resolução da imagem,

frequência de actualização, qualidade e nitidez desta, levando a uma imagem de notória baixa qualidade em formatos de recepção com maior área de imagem, mesmo com boas condições de recepção;

- Não é possível comprimir com eficiência o actual sinal analógico e logo não é possível enviar uma maior quantidade de informação que permitiria aumentar a qualidade de imagem e som;
- Canal de comunicação unidireccional, não permitindo a interactividade dos fluxos multimédia com o utilizador;
- Sistema de som *stereo* de baixa qualidade e sujeito a uma elevada quantidade de ruído;
- Sujeito intrinsecamente a ruído, interferências e atenuações electromagnéticas difíceis de corrigir e compensar;
- Interoperabilidade com sistemas digitais dependente de sistemas auxiliares adaptadores, dificultando a obtenção de imagens com qualidade e impondo um custo adicional;
- Perda de potência com o aumento da distância ao emissor e aparecimento de “fantasmas” na imagem resultantes de reflexões electromagnéticas e outros fenómenos electromagnéticos;
- Necessidade da existência de retransmissores dos sinais analógicos de TV para que seja possível cobrir maiores áreas territoriais.

Por todos estes inconvenientes e devido à emergente utilização de outros sistemas de televisão como é o caso da televisão digital terrestre (TDT ou *Digital Terrestrial Television* (DTTV)), os sinais de televisão analógica já deixaram de ser transmitidos nos Estados Unidos da América[42], e em 2012 essa ruptura com o passado da televisão também terá lugar na Europa, mudança esta na qual Portugal também fará parte em Abril de 2012[30].

É de notar que a partir de 2011 terá início em Portugal o processo de transição entre o velho sistema analógico e o actual sistema digital de TV [26].

2.2.2 Televisão Digital

Todo o processo digital começa com a aquisição das imagens através de câmaras digitais funcionando estas com uma resolução muito superior comparativamente com as câmaras analógicas. No que respeita à transmissão, a televisão digital (DTV) usa um modelo de modulação e compressão de sinais discretos (digitais) para transmitir os sinais de vídeo, áudio e dados para os sistemas compatíveis com esta tecnologia, proporcionando assim uma recepção com muito melhor qualidade de imagem e som, podendo mesmo transmitir em alta definição (*High-Definition* (HD)), utilizando a mesma frequência (canal). Os padrões de DTV em uso comercial, são capazes de transportar aproximadamente 19 Mbps de dados sobre a forma digital. Em termos práticos e de forma aproximada, isto é o equivalente a um conteúdo em HD, que ocupa aproximadamente 15 Mbps, ou quatro em definição padrão, que consomem em média 4 Mbps de largura de banda cada um.

Comparativamente com a televisão analógica, a qualidade de imagem e som usufrui de grandes melhorias. Relativamente à resolução de imagem possível, recordemos os primeiros sistemas receptores de TV que possuíam apenas 30 linhas de vídeo, que, ao longo das décadas de 1930 e 1940, foram evoluindo passando a possuir 240 linhas. Actualmente, um monitor analógico de boa qualidade apresenta entre 480 e 525 linhas. Na televisão digital de alta definição, chega-se a 1080 linhas com o padrão HDTV 1080p (*p*-*progressive*). É também genericamente conhecido o padrão 1080i (*i*-*interlaced*), apresentando na prática uma qualidade inferior ao padrão 720p, por este último ser progressivo em oposição ao 1080i ser entrelaçado.

Relativamente ao formato da imagem, a televisão digital vem consolidar o que já vinha sendo apresentado como uma tendência padrão dos fabricantes de aparelhos de TV e monitores de vídeo: um ecrã plano que passará do formato de proporção 4:3 (típico da TV analógica) para o formato 16:9, mais próximo do formato panorâmico.

A qualidade de som também sofreu grandes alterações, evoluindo dos típicos dois canais (*stereo*) para seis canais (padrão utilizado por equipamentos de som *Home Theaters*).

Devido às características digitais do sinal, a recepção de DTV ocorre sem a presença de fantasmas e com qualidade de áudio e vídeo livres de ruídos e interferências, sendo que podem ocorrer erros esporádicos nos *bits* dos fluxos multimédia, mas sem que estes afectem de forma grave e contínua a imagem ou o som.

Quanto à interactividade entre o utilizador e o conteúdo multimédia disponibilizado, este sistema com qualidade de imagem e som excepcionais peca por não contemplar de uma forma padrão um canal de retorno de informação com largura de banda suficiente para permitir uma elevada interacção e personalização dos conteúdos. Em relação à interactividade possível, pode definir-se esta segundo três níveis:

Interactividade Local O conteúdo é transmitido unilateralmente para o receptor, de uma só vez. A partir daí, o utilizador pode interagir livremente com os conteúdos recebidos que ficam armazenados, tipicamente, na STB do cliente. Um novo fluxo de dados ocorre apenas quando é solicitada uma actualização ou quando é necessário o acesso a um diferente serviço;

Com Canal de Retorno Não-Dedicado A interactividade é estabelecida a partir da troca de informações através de uma rede independente do sistema de televisão, como por exemplo, uma ligação *Asymmetric Digital Subscriber Line* (ADSL). A recepção dos dados relativos ao fluxo dos conteúdos multimédia (*streaming*) ocorre através da infraestrutura de DTV, sendo que o possível retorno de informação que possibilite alguma interactividade dá-se sempre através de outra rede;

Com Canal de Retorno Dedicado Esta seria a interactividade mais desejada, contemplando um canal de retorno dedicado com largura de banda suficiente para a operacionalização de uma interacção de alto nível com o sistema. Para isso, o cliente de DTV necessitaria não apenas de dispositivos receptores, mas também de dispositivos transmissores. Para além disso, os sistemas de DTV teriam de possuir uma interface

de alto nível com o utilizador e a capacidade de enviar os sinais de retorno para a central de transmissão através do canal dedicado.

Mesmo sem possuir um canal de retorno dedicado, os sistemas de DTV permitem que a televisão se torne mais acessível e personalizável, na medida em que se um utilizador não pode assistir a um programa de determinado canal no momento em que este é transmitido, pode configurar o seu gravador de vídeo ou a sua STB para gravar automaticamente esse programa quando este se iniciar e terminar a gravação quando o mesmo acaba. Isto deve-se à existência de sinais digitais específicos, síncronos com os fluxos multimédia, que sinalizam o início e fim dos fluxos multimédia dos conteúdos de forma progressiva. Esta acessibilidade aos conteúdos também é conseguida através da transmissão de um mesmo conteúdo em diversos canais (ou sub-canais) de forma repetitiva, com início separado por um determinado tempo (por exemplo, de 20 em 20 minutos), dando aos utilizadores diversas oportunidades para assistir ao conteúdo sem necessidade de o gravar.

A tecnologia digital possibilita ainda uma flexibilidade e escalabilidade no ajuste de parâmetros de transmissão de acordo com as características geográficas locais. Em áreas acidentadas ou com muitos obstáculos, pode ser utilizado o recurso a uma transmissão hierárquica [27]. Com este recurso, um programa pode ser transmitido com um sinal mais composto de modo a ser recebido em locais mais favoráveis (através de antenas externas, por exemplo), enquanto outro programa ou o mesmo programa do mesmo canal é transmitido com um sinal menos composto e logo com uma menor resolução de imagem para recepção em todos os pontos da área de prestação do serviço. Isto permite que terminais portáteis ou móveis (instalados em veículos, por exemplo) possam receber sem problemas as transmissões conferindo assim um nível de escalabilidade ao sinal.

No que respeita aos meios de transmissão, assim como na tecnologia de TV analógica, o sinal digital pode viajar utilizando diferentes formas de propagação. Algumas destas são:

Terrestre (DTTV ou TDT) Transmitidos através de ondas electromagnéticas, os sinais digitais viajam no ar necessitando de emissores e receptores apropriados com antenas para uma correcta transmissão. Este é o meio de transmissão de DTV com um

custo mais reduzido para o utilizador final, sendo que não há necessidade de pagar uma mensalidade relativa ao serviço à semelhança do que acontecia com a televisão analógica. Esta nova tecnologia necessita de sistemas de recepção de TV digitais, logo, se o utilizador não possuir um destes equipamentos, terá de adquirir um novo ou comprar um adaptador que permita a conversão do sinal digital recebido em analógico no caso de não querer substituir o seu aparelho de televisão analógica. Em Portugal, a televisão digital terrestre foi inaugurada em 29 de Abril de 2009, adoptando, tal como o resto da Europa, o sistema *Digital Video Broadcasting – Terrestrial* (DVB-T), cobrindo 40% da população [26]. Para visualizar a distribuição das tecnologias de transmissão digital terrestre por países, ver figura 2.4;

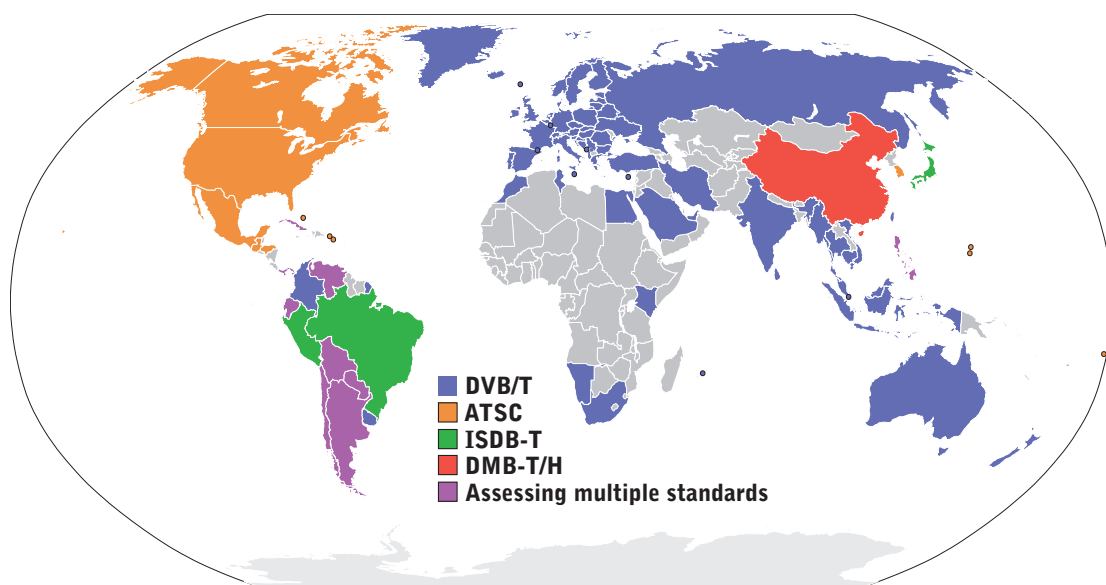


Figura 2.4: Distribuição das tecnologias de transmissão de televisão digital terrestre agrupadas por país.

(Figura retirada de: http://en.wikipedia.org/wiki/File:Digital_broadcast_standards.svg)

Cabo (*Digital Video Broadcasting – Cable* (DVB-C)) Neste caso os sinais de DTV

têm como meio de transmissão não o espaço aberto mas cabos concebidos para o efeito. Este método garante uma melhor qualidade do sinal aumentando a imunidade a erros, interferências e atenuações, permitindo a comunicação a maiores distâncias

exigindo a instalação de uma infraestruturas física de suporte;

Satélite (*Digital Video Broadcasting – Satellite (DVB-S)*) Sistema de transmissão que se baseia na emissão dos sinais de televisão digital a partir de um satélite e recepção por uma antena parabólica. O primeiro serviço mundial de televisão digital por satélite foi lançado na Tailândia e Sul de África no final de 1994. Desde então tornou-se o mais popular sistema de transmissão digital de televisão, estendendo-se rapidamente a mais de 100 milhões de receptores em todo o mundo. Este sistema permite a recepção de televisão digital em locais isolados e remotos que de outra forma não teriam acesso ao serviço [27]. A figura 2.5 exemplifica, de forma muito simplificada, uma configuração de um sistema típico de televisão digital via satélite.

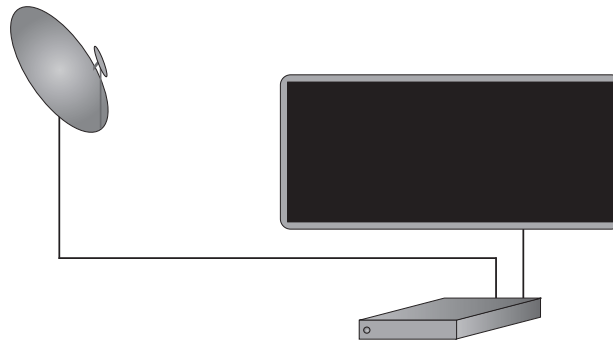


Figura 2.5: Exemplo de configuração de um sistema típico de DTV via satélite.
(Figura baseada em: <http://en.wikipedia.org/wiki/File:Hdpictorialsatellite.gif>)

Terrestre Móvel (*DVB-H*) Sistema de transmissão de televisão digital orientado para a recepção a partir de dispositivos móveis (instalados em veículos, por exemplo). *Digital Video Broadcasting – Terrestrial for Handheld Devices (DVB-H)* foi criada originalmente para uso na banda UHF, actualmente ocupada na maioria dos países por serviços de televisão analógicos e digital terrestre DVB-T. Para colmatar esta dificuldade, foi criada a norma *Digital Video Broadcasting – hybrid Satellite/terrestrial for vehicles and Handheld devices (DVB-SH)*, explorando maiores frequências onde há menor congestionamento do que no UHF. O DVB-H pertence às normas estabelecidas pela União Europeia (UE) para harmonização das telecomunicações em

toda a UE [28];

Híbrida Satélite/Terrestre Móvel (DVB-SH) Sistema de transmissão padrão que permite a entrega de serviços de vídeo, áudio e dados para veículos e dispositivos móveis. Este sistema é híbrido, permitindo a utilização de um satélite para garantir a cobertura de grandes regiões ou até mesmo países inteiros. Nas áreas onde a recepção directa do sinal via satélite não é possível, é utilizado um sistema terrestre para permitir a cobertura. O DVB-SH foi concebido utilizando frequências abaixo dos 3GHz, normalmente em torno dos 2.2GHz. O *Digital Video Broadcasting Group* (DVB Group) começou a trabalhar na especificação do DVB-SH em 2006 [27].

As principais desvantagens da televisão digital face aos sistemas IPTV são:

- Pode possuir qualidade de imagem e som inferior aos sistemas IPTV, apesar de ambos serem bastante equivalentes a este nível;
- Limitações ao nível da interacção e personalização dos serviços devido à não existência padrão de um canal de retorno dedicado. Não permite a organização e classificação automática de canais segundo as preferências do utilizador;
- Possui um número máximo de canais limitado por questões físicas, visto todos os canais serem enviados simultaneamente. Ao contrário deste, no serviço IPTV os fluxos desejados são pedidos e só depois são enviados. Isto confere ao sistema um número de canais virtualmente ilimitado. A figura 2.6 mostra esta diferença no que respeita ao envio dos fluxos de dados;
- Dificuldade na convergência e acesso a fontes de conteúdos multimédia disponíveis na Internet como é o exemplo do *Youtube*;
- Impossibilidade de personalização da programação através da utilização dos serviços de *Video on Demand* ou *Music on Demand*, visto o fluxo de dados ser partilhado por todos os utilizadores;

- Impossibilidade de aceder à Internet com um sistema de DTV;
- Possibilidade de acesso aos conteúdos a partir de dispositivos portáteis e móveis não dedicados mas com necessidade de utilização de um sistema auxiliar adaptador de captura e conversão dos sinais de TV digitais impondo, entre outras desvantagens, um custo extra associado;
- Impossibilidade de interoperabilidade com outros dispositivos e sistemas com o objectivo de atingir uma convergência total (chamada telefónica *Voice over IP* (VoIP) no ecrã do televisor, por exemplo);

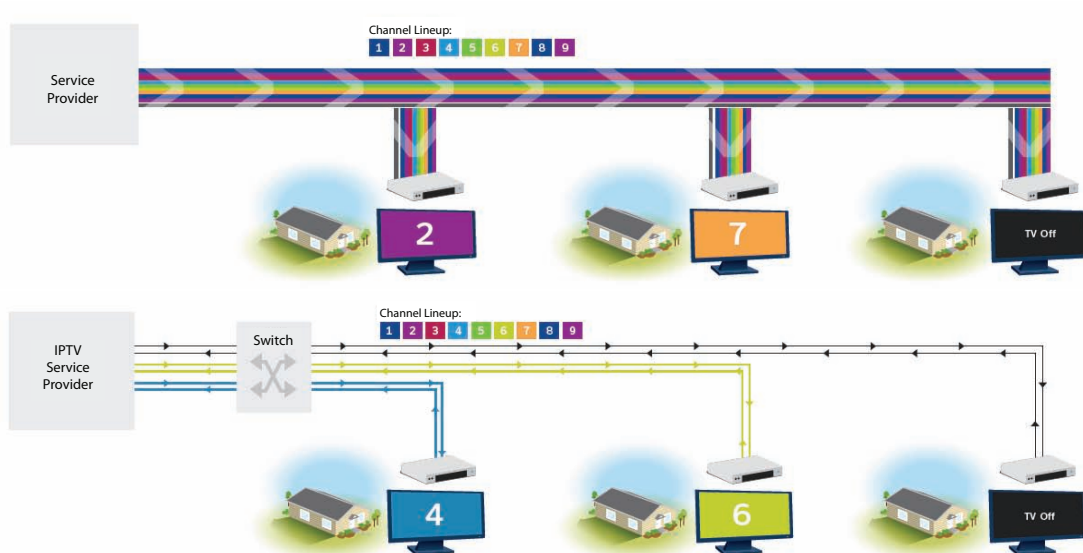


Figura 2.6: Comparação entre um fluxo de dados da TV “convencional” e um fluxo IPTV, respectivamente.

2.2.3 Televisão de Alta Definição

A televisão de alta definição, também conhecida como *High-Definition Television* (HDTV), é uma norma associada à transmissão de televisão com uma resolução significativamente superior à dos formatos tradicionais *Standard-Definition Television* (SDTV) como *National*

Television System Committee (NTSC), *Phase Alternating Line* (PAL) ou *Séquentiel Couleur Avec Mémoire* (SECAM). A DTV pode utilizar tanto SDTV como HDTV nas suas emissões digitais.

As primeiras implementações de televisão de alta definição usavam uma transmissão analógica dos sinais, mas actualmente é usada a televisão digital, requerendo assim menos largura de banda graças à compressão digital de vídeo.

Apesar de terem sido propostos vários padrões de televisão de alta definição, os padrões HDTV actuais são definidos pelo ITU-R BT.709 como 1080i (*interlaced*), 1080p (*progressive*) ou 720p usando uma proporção de ecrã 16:9. No formato 1080i as linhas são actualizadas ordenadamente, uma vez as linhas de índice par e no momento seguinte as linhas de índice ímpar. Já nos formatos 1080p e 720p todas as linhas são actualizadas simultaneamente, evitando que numa cena de movimento rápido uma das sequências de actualização se atrase.

O termo “alta definição” pode se utilizado tanto para referir a própria especificação da resolução ou aparelho de TV, como ao meio de transmissão do conteúdo (DTV) e ainda ao conteúdo em si, capaz de tal definição.

Na figura 2.7 pode ser visto um grafismo ampliado exemplificativo da comparação entre a resolução das tecnologias HDTV e SDTV.

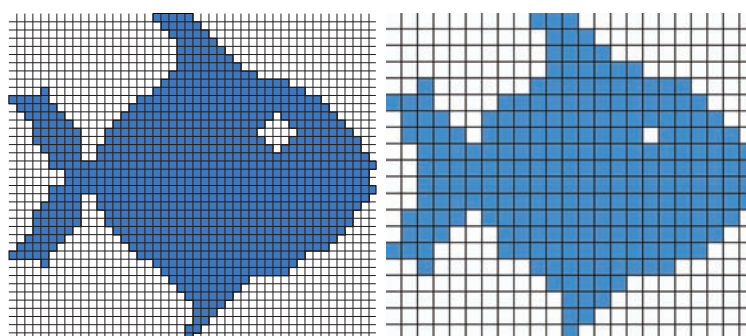


Figura 2.7: Exemplo comparativo das resoluções obtidas nas tecnologias HDTV e SDTV, respectivamente.

(Figura retirada de: http://en.wikipedia.org/wiki/High-definition_television)

2.2.4 Televisão sobre Internet

A televisão sobre Internet (ou ITV) é um serviço de televisão, como o próprio nome indica, distribuído via Internet. Este tornou-se muito popular nos últimos anos, devido a serviços como o *TVCatchup* criado na Inglaterra e que estava (e ainda está) limitado à sua utilização nesse mesmo país. Outros serviços como o *Joost* (<http://www.joost.com/>), o *Veoh TV* (<http://www.veoh.com/>) ou o *Miro* (<http://getmiro.com/>) são também exemplos de serviços de televisão sobre a Internet possíveis de ser utilizados de forma gratuita. É de notar que o sistema *Miro* é um sistema *video podcast* muito interessante e possui muitas funcionalidades que promovem a integração de serviços e tecnologias que os sistemas IPTV também têm em vista como objectivo.

Um exemplo mais actual de uma outra plataforma muito interessante que permite a qualquer utilizador não só assistir a um canal de ITV mas também criar um canal de televisão sobre a Internet de forma gratuita é o *Livestream* (<http://www.livestream.com/>). Este sistema permite a criação de canais de ITV em directo e/ou em diferido e também a sua visualização com recurso a um *player* bastante avançado de nome *Mogulus*.

Criado a partir do serviço *Livestream*, um exemplo de um canal ITV bem conhecido da comunidade académica da Universidade de Aveiro, é o *Seca2TV* (<http://www.seca2.tv/>) onde é possível assistir a vários conteúdos relacionados com a Academia.

A televisão sobre a Internet tem muitas semelhanças com os sistemas IPTV. Além de ambos poderem utilizar as mesmas normas de codificação e os mesmos protocolos de comunicação em rede, ambos também podem ser distribuídos via Internet. Esta rede global é um meio de transmissão de dados actualmente sem garantias de QoS, o que se pode traduzir para o utilizador na carência de QoE de visualização, onde o *multicast* ao nível do IP é tipicamente bloqueado pelos *Internet Service Providers* (ISPs).

O conceito associado à ITV baseia-se em permitir que os utilizadores escolham o conteúdo ou o canal que desejam assistir de uma lista de conteúdos ou de canais disponíveis utilizando uma página de Internet. Assim, estes podem visualizar os conteúdos em qualquer dispositivo que possua acesso à Internet e também suporte o *player* multimédia utilizado

que, tipicamente, é baseado na tecnologia *Flash* da *Adobe*.

Alguns destes serviços disponibilizam também um *player* multimédia que pode ser descarregado e instalado no computador, eliminando a necessidade da existência de um *browser* com o *plugin* do *player* para assistir aos conteúdos disponibilizados.

A SIC e a TVI são exemplos de estações televisivas que disponibilizam legalmente alguns dos seus conteúdos em directo ou em diferido na Internet, criando assim um serviço de televisão sobre Internet. É possível encontrar na Internet alguns sites que mantêm listas bastantes extensas de estações televisivas *online* e conteúdos multimédia distribuídos de forma ilegal utilizando o conceito de TV sobre Internet.

Outros nomes são utilizados para designar tecnologias de televisão sobre Internet, tais como:

Television On the Desktop (TOD) Televisão no *Desktop*;

Video blogging (Vlog) *Blogs* de vídeo em que os utilizadores partilham vídeos e comentários com *feedback* sobre os conteúdos no sistema. O site mais popular deste tipo é o *Youtube*, podendo-se também destacar o *Vimeo*;

Podcast, Vodcast, Video Podcast ou Vidcast São termos para *Video-on-Demand* – distribuição de vídeos pela Internet via RSS em que o utilizador subscreve um ou mais canais (através do seu RSS), recebendo depois as actualizações referentes aos conteúdos publicados nesses canais, visualizando somente os que lhe interessar;

Web TV Televisão *Web* (não confundir com o serviço *Microsoft/MSN WebTV*);

NET TV Televisão sobre a Internet criada pela *Philips* para os seus aparelhos de televisão topo de gama.

Os métodos de entrega dos conteúdos usados pela ITV podem ser de dois tipos:

Broadcatching Significa a transferência de um conteúdo digital que foi disponibilizado através da Internet utilizando o protocolo RSS. A ideia geral é utilizar um mecanismo automático para agregar várias *feeds* RSS para depois transferir o conteúdo

referenciado por estas com o objectivo de ser apresentado posteriormente. Sistemas P2PTV tipicamente fazem uso deste conceito;

Streaming O fluxo correspondente ao conteúdo pretendido provém de um só servidor *web*.

Algumas das tecnologias utilizadas pela televisão sobre Internet quer ao nível de rede de comunicação, quer ao nível da codificação digital dos conteúdos a disseminar são:

BitTorrent É um protocolo de processamento rápido que permite ao utilizador efectuar *download* de arquivos indexados em *web sites*. Esta rede é do tipo P2P, fazendo uso do conceito de partilha do que acabou de transferir da Internet, maximizando assim muito o desempenho global e possibilitando *downloads* bastante rápidos. Foi criado por Bram Cohen em 2003 e estima-se que no passado ano de 2005, este protocolo tenha sido responsável por 35% dos dados transferidos via Internet a nível mundial;

Dirac É um codificador/descodificador (codec) de vídeo *open source* desenvolvido pelo departamento de investigação da *British Broadcasting Corporation* (BBC). Destina-se a permitir compressão de vídeo em HD até ao *Ultra High Definition Video* (UHDV), competindo com codificadores já existentes tais como H.264 e VC-1. O codec de vídeo *Dirac* foi utilizado internamente pela BBC para transmitir as imagens em HDTV referentes aos Jogos Olímpicos de Pequim em 2008. Esta especificação foi finalizada em 21 de Janeiro de 2008;

Nullsoft Streaming Video (NSV) É um formato de multimédia desenhado para facilitar a transferência de conteúdos multimédia através de *streaming*, suportando virtualmente qualquer codec de áudio e vídeo e podendo ser utilizado em quase qualquer plataforma. Actualmente, o NSV utiliza *MPEG-1 Audio Layer 3 codec* (MP3) para a codificação do áudio e VP3 (actualmente pertencente e base do codec *Theora*) para a codificação do vídeo. Esta é a tecnologia usada pelo *America OnLine* (AOL)–Video para entrega de conteúdo de vídeo através da Internet;

Theora É um codec de vídeo desenvolvido pela Fundação *Xiph.org* sendo uma parte do projecto Ogg. Baseado no codec VP3 e doado pela *On2 Technologies*, o *Theora* é semelhante aos codecs MPEG-4, XviD e DivX.

Worldwide TV Mark-up Language (WTVML) É um formato baseado em XML desenhado para permitir aos operadores de *web sites* desenvolver e implementar de forma fácil serviços de TV interactiva. Tipicamente, esta tecnologia reduz o tempo despendido na criação do *TV site* e resulta na compatibilidade deste com uma larga quantidade de dispositivos, sendo capaz de ser automaticamente validado, testado e transformado segundo as necessidades do operador. O formato de descrição segue, de forma estrita, uma sintaxe XML através de regras especificadas por um ficheiro *Document Type Definition* (DTD). O WTVML possui o potencial para uniformizar a criação de conteúdo para televisão interactiva, que está recheada de *middlewares* proprietários, existindo falhas também relativas à padronização dos *browsers*. O serviço WTVML, que segue o modelo *Web2.0*, pode ser automática e dinamicamente transformado em várias formas de HTML/JS/CSS, tornando este compatível com os típicos *web browsers* bem como com *browsers* WTVML nativos. Permite assim ao operador do sistema gerir as funcionalidades específicas da sua plataforma independentemente do padrão utilizado na construção do *web site*;

HTTP Tendo como significado *Hypertext Transfer Protocol*, é o protocolo de comunicação que permite a troca de mensagens *web* na camada da aplicação. Este protocolo é amplamente usado na Internet para comunicação de dados entre vários sistemas de diferentes tipos;

RSS Protocolo de agregação e disseminação de conteúdos que utiliza o padrão XML como forma de organização da informação transportada. Através das *feeds* RSS e dos leitores de *feeds* que as interpretam, o utilizador pode manter-se informado à cerca das actualizações de um ou mais *sites*, *blogs*, jornais *online*, entre outros serviços, sem que para isso tenha de os aceder um a um directamente. Os serviços que utilizam

RSS são tipicamente aqueles cujo conteúdo é actualizado regularmente, como é o exemplo de jornais *online* ou sites de ITV;

RSS *Enclosure* Estas *feeds* são semelhantes aos RSS comuns, sendo que permitem a inclusão de referências para conteúdos multimédia dentro dos mesmos, fornecendo a URL para o arquivo associado, sendo este por exemplo, um ficheiro MP3. Ao contrário dos protocolos de e-mail, o RSS *enclosure* não agrega o conteúdo indexado, possuindo meramente um *hyperlink* para o conteúdo. Um agregador de *feeds* RSS ao receber um RSS *enclosure* pode transferir automaticamente o conteúdo e exibi-lo, fornecer um link para o mesmo ou simplesmente ignorá-lo;

RTSP Tendo como significado *Real Time Streaming Protocol*, é um protocolo ao nível da aplicação desenvolvido pela *Internet Engineering Task Force* (IETF) em 1998, para controle durante a transferência de dados multimédia sobre a forma de *streaming* em tempo real sobre a Internet;

SMIL O *Synchronized Multimedia Integration Language* pertence ao padrão *World Wide Web Consortium (W3C) recommended XML markup language* para descrição de apresentações multimédia. Este define a temporização, *layout*, animações, transições visual, elementos multimédia embebidos, entre outros. O *Synchronized Multimedia Integration Language* (SMIL) permite a apresentação de vários elementos tais como texto, imagens, vídeo e áudio, bem como *links* para outras apresentações SMIL e outros arquivos vários. Este usa o XML para descrever os seus elementos e tem algumas semelhanças com o *HyperText Markup Language* (HTML).

A ITV difere da IPTV no que respeita ao que ambas tem para oferecer e na forma de acesso e qualidade dos seus conteúdos. Mesmo sendo ambas baseadas no protocolo IP, enquanto que na ITV não é efectuada uma gestão e monitorização de rede ao nível da qualidade de serviço (QoS), em IPTV este ponto é de extrema importância de modo a garantir um serviço de boa qualidade com uma largura de banda necessária à correcta operação do sistema sem falhas. Outra diferença entre ITV e IPTV é que o primeiro é

orientado à visualização em computadores “convencionais”, enquanto que a IPTV pretende ser o mais independente possível do equipamento físico.

Assim, o conceito de IPTV implica normalmente o conceito de rede com gestão de tráfego e QoS, criando um serviço dedicado à difusão de multimédia. Apesar disso, algumas definições para IPTV, tais como as referidas pela *International Telecommunication Union* (ITU) e pela DVB Group, usam o termo IPTV como um super-conjunto de ambos os sistemas IPTV com gestão de rede e ITV sem gestão de rede.

Mais uma vez, a IPTV pretende juntar o melhor de todos os mundos e afirmar-se como o sistema de TV do futuro, tendo como objectivos a apresentação dos conteúdos com a máxima qualidade e interactividade possíveis, sem descuidar a interoperabilidade de recursos e tecnologias num ambiente global.

2.2.5 IPTV

IPTV, cujo acrónimo é referido como “televisão do futuro”, pode ter interpretações muito diversificadas. Algumas delas são, *Interactive Personal TV*, devido a esta permitir interactividade ao nível da personalização da programação segundo os desejos do utilizador, *Intelligent Personal TV* por esta nova forma de televisão ser “inteligente” não devido ao equipamento de TV em si mas sim à rede à qual está ligada, ou *Internet Protocol TV* devido a esta ter como meio de comunicação de dados a rede global que é a Internet e esta ser baseada no protocolo IP. Ao longo de toda esta Dissertação de Mestrado, a interpretação para este acrónimo será *Internet Protocol Television* não só por ser o mais genérico dos disponíveis, mas também por ser este o comumente mais utilizado.

Os sistemas IPTV estão a mudar a forma como a TV é distribuída e vista, oferecendo enormes vantagens aos emissores televisivos e aos utilizadores. Não só o baixo custo de disseminação e transporte mas também o facto de possibilitar a oferta de serviços interactivos diferenciados como *on-demand TV* (*Video-on-Demand* (VoD), *Music on Demand* (MoD), jogos, etc.) torna os sistemas IPTV muito apetecíveis tanto para os utilizadores como para os operadores.

A IPTV pretende juntar o melhor de todos os mundos no que respeita aos diversos tipos de sistemas de televisão actualmente existentes. A universalidade e baixo custo da televisão analógica, a excelente qualidade multimédia, interoperabilidade e estabilidade do mundo da radiodifusão digital fazendo uso da alta definição (HD) ou ultra-alta definição (UHDV) são alguns dos objectivos da IPTV. A elevada capacidade de interacção, integração, inovação, multiplicidade, disponibilidade e heterogeneidade da televisão sobre Internet são também aspectos que os sistemas IPTV têm como objectivo concentrar num só.

Com este novo conceito, cada utilizador cria a sua própria televisão, na qual define quais os conteúdos que deseja assistir, quando os deseja assistir e onde os deseja assistir conferindo extrema mobilidade ao utilizador. Assim sendo, transitamos de um sistema baseado na selecção de canais específicos, rígidos no que respeita ao seu conteúdo, para um sistema baseado na selecção do tema ou conteúdo que se deseja assistir, conferindo-lhe um elevado grau de versatilidade e personalização.

Enquanto que IPTV é inerentemente DTV pois utiliza sinais digitais, DTV não é equivalente a IPTV. A rede de distribuição de DTV utiliza como meio de transmissão dos sinais digitais ondas electromagnéticas segundo um meio guiado ou não guiado, enquanto que IPTV usa uma rede IP sobre um qualquer meio físico com ou sem fios. No que diz respeito à interactividade, a DTV disponibiliza tipicamente um só canal unidireccional dedicado que é utilizado pelo cliente do serviço digital para recepção (*download*) do fluxo multimédia. Por oposição a este, a IPTV utiliza inerentemente um canal de comunicação bidireccional (*download e upload*) disponibilizados, de forma natural, por uma rede IP. Esta diferença é crucial no que respeita à criação de interactividade padrão entre o serviço de televisão e o utilizador. Mesmo considerando que alguns sistemas de DTV possam actualmente prever um política de comunicação bidireccional, esta utiliza um canal de retorno extra não dedicado (por exemplo, uma ligação ADSL), o que limita a sua utilização e obriga à utilização de mais uma rede de telecomunicações. As principais semelhanças entre IPTV e DTV estão na necessidade de ter-se uma STB para utilização em sistemas de televisão não concebidos de forma inata para o efeito e na qualidade de áudio e vídeo que são próximas, ambos podendo transmitir conteúdos em HD.

Os operadores de telecomunicações introduziram-se nos tempos modernos do acesso à Internet através de campanhas de *marketing* onde anunciavam a oferta de soluções *Dual Play* que se caracterizavam pelo fornecimento de serviços fixos de Internet + Telefone mas em regime de utilização em exclusão mútua. Com as evoluções tecnológicas, a utilização do telefone simultaneamente com a Internet deixou de ser um problema. Seguindo as tendências do mercado, os operadores passaram a oferecer soluções fixas *Triple Play* que contemplavam Internet + Telefone + Televisão, sendo esta última distribuída através de uma rede *Community Antenna Television* (CATV) (ver figura 2.8). Surgiu por esta altura a televisão digital e com ela a televisão de alta definição que veio impor mais qualidade multimédia ao serviço. Numa das mais recentes abordagens, os operadores de telecomunicações começaram a disponibilizar soluções *Quadruple Play*. Esta, tal como a figura 2.9 tenta representar, contempla os serviços fixos habituais (Internet + Telefone + Televisão) com o acréscimo do acesso móvel dual à Internet, utilizando a rede *Global System for Mobile Communications* (GSM)/*Universal Mobile Telecommunication System* (UMTS) quando fora do alcance da rede *WiFi* fornecida pelo acesso fixo à Internet, fazendo *roaming* entre estas de forma, teoricamente, transparente para o utilizador. Para usufruir deste serviço é obviamente necessário possuir um telefone móvel com as capacidades de comunicação necessárias. Esta recente abordagem *Quadruple Play* vem sem dúvida facilitar e promover a convergência dos meios de acesso aos serviços disponíveis e a disponibilização dos mesmos a utilizadores móveis independentemente do local onde se encontram, da rede de transporte que utilizem e do equipamento que possuem.

Ao nível do meio físico de transmissão, as redes de telecomunicações acessíveis ao utilizador comum estão também a mudar. A actual tendência de transformação das redes que funcionavam sobre meios de transmissão de cobre para sistemas de fibra óptica traduz-se num ganho superior a 10 vezes mais velocidade para o utilizador final. Este espantoso aumento de velocidade das redes de acesso à Internet possibilita que esta seja utilizada, de forma intensiva, por outras formas de comunicação, entre elas os sistemas de televisão digital completamente interactiva que o IPTV pretende implementar. Actualmente é possível cenários em que um mesmo utilizador numa mesma rede doméstica pode possuir múltiplos

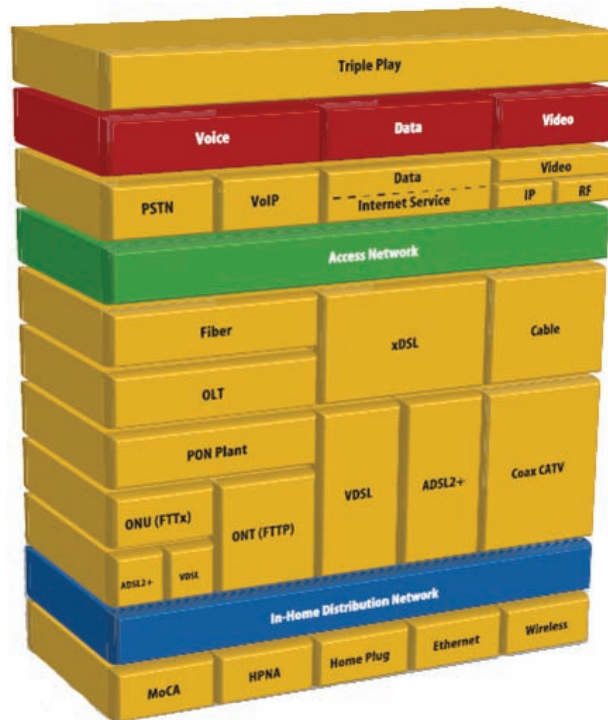


Figura 2.8: Arquitectura do serviço *Triple Play*.
(Figura retirada de: <http://www.seratoptik.com/?p=121>)

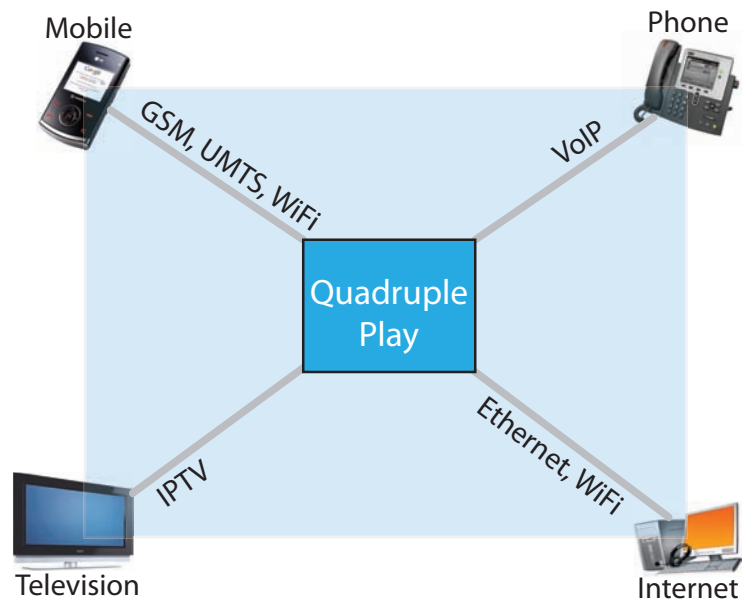


Figura 2.9: Representação gráfica do serviço *Quadruple Play*.

dispositivos receptores de IPTV activos simultaneamente, exibindo diferentes conteúdos e todos eles usufruindo de uma muito boa qualidade de serviço. É também ainda possível a utilização simultânea de terminais de acesso convencional à Internet e ainda a existência de chamadas de voz sobre IP (VoIP) a decorrem com fluidez tudo sobre a mesma infraestrutura de rede. Este facto pode ser constatado não só matematicamente mas também devido ao exemplo que o serviço *Meo* da *PT* oferece.

Mesmo com os actuais desenvolvimentos relativos ao aumento das velocidades de acesso à Internet, os sistemas IPTV possuem inerentemente uma necessidade de gestão da rede de transporte e QoS associados. Com o aumento da largura de banda no acesso à Internet, os seus utilizadores mudam a forma de a utilizar, criando ainda mais tráfego na rede. Isto invalida a implementação de sistemas IPTV com garantia de boa qualidade de serviço sem a utilização de mecanismos de QoS e controlo de rede, para um serviço contínuo sem quebras de qualidade, com latência e *jitter* reduzidos. É este um dos pontos que distinguem IPTV do conceito de ITV. A não necessidade de presença de um *browser* e a possibilidade de utilização de múltiplos tipos de clientes ou de um único cliente escalável de modo a abranger uma gama diversificada de tipos de dispositivos, é outro ponto que afasta a IPTV da ITV. Os sistemas ITV não se interessam pelo tipo e características do dispositivo receptor nem pela qualidade da rede que fornece o serviço. A segurança e personalização dos serviços que os sistemas de IPTV oferecem requer também políticas específicas baseadas em protocolos de *Authentication, Authorization and Accounting* (AAA), que permitam ao utilizador usufruir do sistema em qualquer local onde se encontre disponível, de uma forma segura e controlada de acordo com os seus privilégios.

Visto o tráfego IPTV poder ser transportado por uma rede de larga escala já existente como é a Internet, coexistindo com outro tipo de tráfego muitas vezes menos prioritário, torna num grande desafio a monitorização das redes e sua optimização de forma a gerir eficientemente a largura de banda disponível criando um serviço sustentável, escalável e viável não só tecnologicamente mas também do ponto de vista económico.

Analisando as consequências de tal implementação generalizada de novos serviços IPTV mais os sistemas e serviços existentes actualmente, é normal que os ISPs se mostrem

preocupados devido à carga intensiva de fluxos que a rede teria de suportar. A preocupação é tanta que estes apelidam os sistemas de IPTV como “os assassinos das redes do futuro”. É de notar que a distribuição de um serviço de IPTV pode ser efectuada recorrendo a diferentes arquitecturas de rede, desde sistemas cliente-servidor até sistemas P2P (P2PTV) complexos. Por sua vez, estes sistemas P2P podem ser estruturados ou não estruturados, fazendo uso das características de disponibilidade, escalabilidade global, heterogeneidade e descentralização próprias deste tipo de arquitectura de rede. Alguns sistemas P2PTV foram já implementados com sucesso e a sua análise revelou resultados promissores no que respeita à carga na rede.

Na tabela 2.1 encontra-se uma pequena lista de alguns dos serviços mais conhecidos que disponibilizam gratuitamente conteúdos multimédia, segundo uma política P2P. É de notar que alguns destes tipos de serviços possuem conteúdos ilegais que violam regras de direitos de autor. Isto não é de estranhar dada a índole livre e, por vezes, não controlada deste tipo de serviços. A maioria deles são baseados no sistema *BitTorrent*, possibilitando a partilha de qualquer conteúdo entre os *peers* de uma forma distribuída. Para mais informações sobre qualquer um dos programas listados em particular, é favor consultar a página *web* respectiva. Uma curiosidade sobre os sistemas P2PTV listados é que a maioria é de origem Chinesa e em alguns casos a sua documentação ou o próprio *software* cliente não se encontram traduzidos em Inglês ou outra língua mais acessível, dificultando assim em muito a sua utilização por um grupo linguisticamente mais heterogéneo de utilizadores.

No que respeita a um breve apanhado da sua história em termos mundiais, sabe-se que o termo IPTV surgiu pela primeira vez em 1995. Em 1998 a empresa *AudioNet* inicia o primeiro *webcasts* ao vivo com conteúdo televisivo. Por volta de 2001, foi a empresa *Kingston* uma das primeiras a introduzir um sistema IPTV sobre a tecnologia ADSL. Em 2003 a empresa *Total Access Networks Inc* lança o seu serviço de IPTV, que compreendia 100 canais IPTV internacionais de acesso gratuito. Este serviço foi utilizado em mais de 100 países e tinha canais em 26 línguas distintas. Em 2005 a empresa *Bredbandsbolaget* lançou o seu serviço de IPTV, tendo sido o primeiro a surgir na Suécia.

Mais recentemente, em 2006, a *AT&T* lançou nos Estados Unidos da América um

Nome do software	Página web
<i>PPLive</i>	http://www.pplive.com/en/
<i>Coolstreaming</i>	http://www.coolstreaming.us/hp.php?lang=en
<i>TVUPlayer</i>	http://www.tvunetworks.com/
<i>PPStream</i>	http://www.ppstream.com/
<i>PPMate</i>	http://ppmate-nettv.en.softonic.com/
<i>TVAnts</i>	http://www.tvants-ppstream.com/
<i>Sopcast</i>	http://www.sopcast.com/
<i>QQLive</i>	http://tv.qq.com/
<i>UUSee</i>	http://www.uusee.com/
<i>Feidian</i>	http://www.pprecorder.com/iptv-feidian.htm
<i>VGO</i>	http://vgo.21cn.com/index.html
<i>TVKoo</i>	http://www.tvkoo.com/en/aboutus.htm
<i>SstreamTorrent</i>	http://groups.google.com/group/streamtorrent
<i>Afreeca</i>	http://www.afreeca.com/
<i>Veetle</i>	http://www.veetle.com/
<i>Zattoo</i>	http://zattoo.com/

Tabela 2.1: Lista de alguns sistemas P2PTV mais conhecidos.

serviço de IPTV de nome *U-Verse*, contendo mais de 300 canais em 11 diferentes cidades Americanas. Em 2009 a *AT&T* anunciou a introdução de mais de 100 canais de TV em HD no seu serviço de IPTV. Utilizando sempre o protocolo IP, a *AT&T* construiu a sua própria rede de transporte, exclusiva para transportar conteúdos multimédia, garantindo assim uma excelente qualidade de serviço[34].

Na figura 2.10 é possível observar a distribuição dos serviços de IPTV no mundo até meados do corrente ano de 2009.

No que respeita a Portugal, actualmente três grandes empresas exploram o conceito de IPTV, incluindo este numa oferta *Quadruple Play*. Estas permitem aos seus clientes, através do seu sistema de televisão, alugar/comprar filmes, ver canais de televisão em qualidade HD, gravar programas televisivos na sua STB com agendamento remoto, entre outras opções. Este tipo de serviços ganharam uma grande popularidade, mas ainda possuem uma falta de convergência, heterogeneidade, personalização e interactividade que a cada novo produto são melhoradas.

Estas três empresas são a *Portugal Telecom* com o serviço *Meo e Meo Fibra*, a *Zon* com

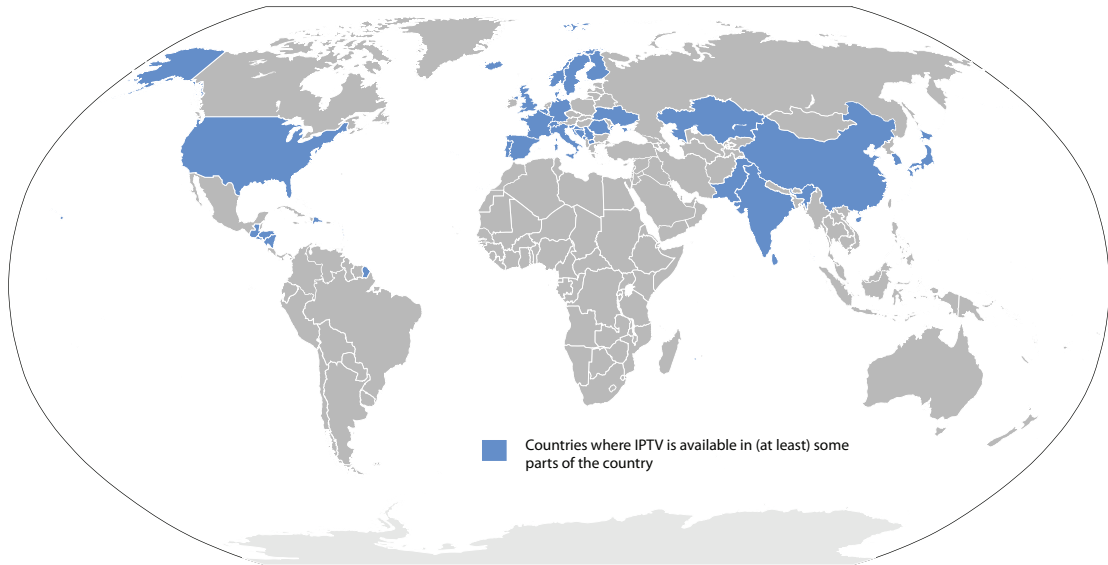


Figura 2.10: Mapa da distribuição dos serviços de IPTV no mundo até meados de 2009.

(Figura retirada de: <http://en.wikipedia.org/wiki/File:IPTV-Countries.svg>)

o serviço *Zon TVCabo e Zon Fibra* e a *Vodafone* com o serviço *Vodafone Casa TV*.

Como seria de esperar, o imergir dos sistemas IPTV criou novas oportunidades de negócio e desenvolvimentos tecnológicos relacionados com sistemas de televisão, telecomunicações e multimédia. Alguns conceitos e funcionalidades inovadoras no campo dos sistemas de televisão foram introduzidos bem como funcionalidades já existentes foram reinventadas. É o caso da experiência de utilização de suportes multimédia como *Digital Video Discs* (DVDs) e de vídeos *online*, onde o utilizador tem total controlo sobre o fluxo do conteúdo, podendo realizar operações tão simples como parar, pausar, avançar ou retroceder para um local específico do conteúdo. Mudar o idioma das legendas e do áudio ou até o ângulo de visualização são outras operações também possíveis. Assim sendo, algumas das funcionalidades possíveis de implementar em sistemas IPTV são:

Alta definição Imagem HD e som multi-canal cristalinos de excelente qualidade mesmo em formatos de visualização de grande dimensão;

Controlo Parental Restrição dinâmica da visualização de canais e conteúdos a grupos específicos de espectadores (por exemplo, crianças), de modo a filtrar possíveis

conteúdos impróprios para estes. Ao contrário das restrições estáticas impostas pelos sistemas actuais, é possível implementar restrições baseadas não somente no canal mas também na semântica dos conteúdos multimédia exibidos;

Multiview Possibilidade de ver múltiplos diferentes conteúdos ou canais no mesmo ecrã ao mesmo tempo, seleccionando dinamicamente um destes como principal. Este conceito também se aplica a um mesmo canal mas com a possibilidade de visualização do conteúdo sobre diferentes vistas ou ângulos, o que aliado às restantes funcionalidades de controlo, confere ao sistema uma poderosa capacidade de interacção e personalização;

Start Over TV ou Time-shifted TV Caso o utilizador não comece a ver um programa que lhe interesse transmitido “em directo” desde o início, é possível voltar ao seu início e ver todo este (com uma deslocação no tempo relativa ao original) sem que para isso o utilizador tenha previamente iniciado a sua gravação;

Pause Mesmo sendo um programa “em directo”, é possível efectuar uma pausa na sua exibição e retomar depois sem perder um único segundo deste;

Fast Forward and Rewind É possível avançar ou retroceder dentro dos limites temporais do conteúdo multimédia mantendo a sua causalidade. No caso de um conteúdo multimédia em tempo real (por exemplo, um jogo de futebol transmitido “em directo”), obviamente é possível avançar somente até o ponto em que este se encontra no fio temporal real;

Personal Video Recording (PVR) Esta funcionalidade já existe em diversos sistemas actualmente no mercado, e resume-se na possibilidade do utilizador gravar no seu dispositivo receptor (ou STB) um determinado conteúdo multimédia de forma a poder visualiza-lo mais tarde sem necessitar de recorrer à rede para o fornecer. Genericamente falando, qualquer conteúdo que seja emitido/disponibilizado uma vez deverá estar disponível na rede sempre a partir desse momento, caso isto não viole leis de

direito de autor. Caso o utilizador não desejar recorrer à rede para o rever, poderá guardá-lo localmente. Este sistema deverá ter a capacidade para gravar simultaneamente múltiplos conteúdos provenientes de diferentes fontes. Uma funcionalidade muito útil aliada à gravação de conteúdos é o controlo remoto desta funcionalidade, possibilitando colocar remotamente o receptor a gravar um qualquer programa em determinado momento;

network Personal Video Recording (nPVR) Diferencia-se do PVR na medida em que o conteúdo é armazenado pela rede de forma a disponibiliza-lo posteriormente aos utilizadores. Estes conteúdos gravados na rede são fundamentais para o funcionamento dos mecanismos de *Fast Forward and Rewind*, *Start Over TV* e *Pause*. Neste tipo de armazenamento em rede, ao utilizador não interessa saber onde está o conteúdo mas sim se está disponível quando ele deseja aceder-lhe. Este mecanismo pode também ser usado para o utilizador colocar os seus próprios vídeos e partilha-los com a família e amigos de uma forma controlada e segura;

Video-on-Demand Refere-se ao acesso ou compra ágil de conteúdo multimédia através da Internet e posterior visualização sem que para isso o utilizador tenha de levantar do sofá. Com este serviço o utilizador acede aos conteúdos que deseja quando desejar, sem estar restringido à programação rígida de um canal de televisão. O VoD tem como analogia os tradicionais “vídeo-clubes”, sendo que aqui o utilizador não precisa de se preocupar em entregar o filme dentro do limite estabelecido para não pagar multas! Se o utilizador deseja, por exemplo, assistir a um filme que estreou recentemente e não é de visualização gratuita, pode o obter comprando-o de uma forma permanente ou temporária directamente no sistema de TV. Desde o momento em que o conteúdo é adquirido, é possível realizar todas as operações de visualização e navegação sobre ele, como se um DVD se tratasse, podendo as operação de gravação não ser permitidas devido a leis de direitos de autor (tal como acontece normalmente nos DVDs legais);

Music on Demand Este serviço é de índole muito semelhante ao serviço VoD e visa permitir a aquisição ágil e legal de músicas e sua futura reprodução a partir do sistema IPTV;

Bookmark Caso exista uma parte específica de um conteúdo multimédia que por algum motivo o utilizador gostaria de assinalar para futuramente encontrar facilmente, assume-se a possibilidade de colocação de uma marcação neste. É feita assim uma analogia com um marcador de um livro, em que o utilizador pode marcar de alguma forma uma frase, parágrafo ou página de um livro que lhe suscite mais interesse;

Anúncios publicitários localizados Abre uma nova forma de publicidade e negócio em que um determinado conteúdo pode possuir produtos e/ou serviços facilmente adquiríveis através do sistema de televisão em uso, recorrendo à compra *online* via Internet do mesmo, ou simplesmente, possibilitando a consulta de informação mais pormenorizada sobre estes. Desde bilhetes de cinema até ao vestuário ou mobiliário, tudo poderá suscitar interesse ao utilizador ao ver esse bem ou serviço em determinado conteúdo multimédia. Para além de serviços de publicidade profissional, também poderá suportar, de uma forma controlada, publicidade colocada directamente por determinado utilizador com vista a vender um determinado produto;

Informações e notícias personalizadas Barra de notícias e informações personalizadas colocada estrategicamente no ecrã sem interromper ou comprometer de forma significativa a visualização de conteúdos multimédia. Esta informação poderá usar dados como a localização do utilizador, temperatura, hora do dia e outros para alertar o utilizador para notícias importantes, eventos locais, meteorologia local, etc;

Emergências e Monitorização Serviços de emergência de vária índole poderão estar ligados ao utilizador 24 horas por dia via IPTV, quer por intermédio de dispositivos fixos ou móveis. Deste modo é possível a monitorização de situações consideradas críticas *a priori* bem como o fácil despoletar de alarmes em situações anómalas pontuais;

Electronic Program Guide Disponibilidade de um guia electrónico de programação dos conteúdos televisivos que permite ver e/ou seleccionar os conteúdos multimédia a assistir em determinada hora;

Sistemas de geolocalização Introdução de opções de procura de bens ou serviços (por exemplo, nas *páginas amarelas*), que permita a localização geográfica dos mesmos bem como a captura das informações disponíveis e o contacto telefónico ou de e-mail. É também interessante a possibilidade de localização geográfica de elementos que o utilizador deseje, como por exemplo, o seu próprio cão, o seu carro ou amigos e familiares do mesmo (se estes o permitirem). O acompanhamento da rota do auto-carro escolar dos filhos ou do avião onde a esposa viaja também são funcionalidades interessantes. Para além destes serviços, o planeamento de rotas e viagens torna-se muito mais cómodo quando efectuado de forma natural em frente ao sistema de TV socializando com a família ou amigos, usufruindo de vídeos dos locais, sugestões, informações e todo um conjunto de ferramentas disponíveis;

Interactividade e comunicação entre utilizadores Possibilidade nativa de troca de mensagens e outras formas de comunicação entre utilizadores presentes no sistema. Esta funcionalidade cria a oportunidade de utilização de um sistema que, até então, para pouco mais servia que ver televisão. Assim os sistemas de TV tornam-se em verdadeiros sistemas de comunicação e entretenimento multi-utilizador de alta qualidade, disponibilizando, entre outros serviços, jogos em rede (com possibilidade de serem computados remotamente abstraindo-se assim das capacidades gráficas e de processamento do cliente), *karaoke*, vídeo-conferência, álbuns fotográficos, acesso a redes sociais (*facebook*, *twitter*, etc.), utilização de *chats* (*Google Talk*, *Skype*, *Microsoft Messenger*, etc.), entre outros;

E-learning Os sistemas de ensino à distância fazem cada vez mais parte da educação dos cidadãos. Os sistemas IPTV podem permitir o fácil e cómodo acesso a complexos sistemas de *e-learning* à distância, livros electrónicos, tutoriais em vídeo, entre outros;

Acesso à Internet Esta funcionalidade surge de uma forma perfeitamente natural, dado que os sistemas IPTV são sistemas que utilizam a Internet. Assim sendo, tal como é possível com IPTV ver televisão no PC, também poderá ser possível aceder a todos os recursos da Internet a partir do aparelho convencional de TV, usufruindo do mesmo nível de interactividade de um computador típico;

Partilha de dados Partilha de fotografias, música, documentos e outros ficheiros com outros utilizadores IPTV e com outros sistemas computacionais pertencentes à rede local ou global;

Procura avançada de conteúdos Possibilidade de procura de conteúdos não só de forma textual mas também fazendo uso dos mecanismos de reconhecimento de padrões e análise facial por comparação com outros conteúdos (imagens, fotos, vídeos, músicas, etc.). Este mecanismo também poderá ser utilizado na procura de produtos em questões de publicidade;

Heterogeneidade e convergência dinâmica Com os sistemas IPTV a utilizarem uma rede *IP Multimedia Subsystem* (IMS) controlada é possível, entre outras coisas:

- Permutar a visualização dos conteúdos multimédia entre dispositivos distintos de forma fácil e com persistência do estado anterior. Por exemplo, caso o utilizador esteja a assistir um filme no seu sistema de TV em casa mas precisa de sair para ir apanhar o comboio, ele poderá parar a exibição do conteúdo multimédia e continuar a assistir o mesmo, por exemplo, no seu *iPhone* desde o ponto em que parou anteriormente, mantendo toda a interactividade com o conteúdo;
- Utilizar sistemas de TV para interagir e controlar outros sistemas distintos ligados à Internet. Por exemplo, caso o utilizador possua uma “casa inteligente”, é possível utilizar o seu actual sistema de TV (dedicado ou não dedicado, fixo ou móvel) para controlar e verificar os recursos e parâmetros da sua habitação. Este pode, por exemplo, verificar se todas as janelas estão fechadas pois começou a chover, visualizar imagens do seu bebé que se encontra a dormir no quarto do

piso acima ou ligar o sistema de rega do jardim. Esta funcionalidade também se reveste de muita utilidade no caso do utilizador abandonar a sua casa por um período indeterminado mas desejar continuar a controlá-la como se lá estivesse, assim como vigiar a mesma ao mesmo tempo que vê as notícias ou um filme, por exemplo;

- Comunicar com utilizadores de outros sistemas com diferentes arquitecturas como GSM, UMTS, *Internet Messaging*, entre outros, através de *Short Message Services* (SMSs), chamadas de áudio, chamadas de vídeo com dispositivos móveis ou fixos que as suportem, áudio ou vídeo-chamadas com outros utilizadores de IPTV, etc. Desde o momento em que o acesso aos recursos esteja centralizado e todos os subsistemas estejam interligados formando um sistema de comunicação “universal”, uma enorme quantidade de funcionalidades e serviços são passíveis de serem implementados;

Foi anunciado recentemente pela empresa *Microsoft Corporation* o desenvolvimento de um novo projecto, aplicado à consola *XBOX*, chamado *Project Natal* (<http://www.xbox.com/en-US/live/projectnatal/>). Este projecto visa a criação de um sistema revolucionário na área da interacção humano-computador, aplicado neste caso a jogos de computador. A *Microsoft* propõe que não sejam necessários quaisquer controlos remotos excepto os movimentos físicos e a capacidade vocal do utilizador. Tal como se pode encontrar na sua página *web*: “See a ball? Kick it, hit it, trap it or catch it. If you know how to move your hands, shake your hips or speak you and your friends can jump into the fun – the only experience needed is life experience.”.

O *Projecto Natal* é mais um exemplo de um sistema, que aliado ao conceito de IPTV, poderia despoletar uma profunda mudança na forma como os utilizadores interagem e descrevem os sistemas de TV. A televisão sem dúvida que mudou a forma como as pessoas vêem o Mundo desde que foi criada. Esta tem agora espaço e possibilidade para crescer em termos de funcionalidade, utilidade e criatividade, que podem não só transformar o mundo da televisão mas também o Mundo e a sociedade global.

Analisando serviços como o *iTunes* da Apple que disponibiliza um variado leque de música e *software* (para dispositivos *Apple*) de forma legal e a preços bastante acessíveis, é possível determinar o porquê da sua tão grande popularidade. Serviços semelhantes seriam, possivelmente, muito bem vistos pelos utilizadores no que respeita aos sistemas de distribuição de conteúdos em IPTV. Enquanto que conteúdos gratuitos seriam disponibilizados de forma completamente livre, conteúdos não gratuitos como filmes ou séries televisivas recentes seriam disponibilizadas segundo uma política de AAA, pagos através de importâncias mínimas, que incentivassem à redução da pirataria informática relativa a conteúdos multimédia.

Genericamente e salvo restrições de direitos de autor, qualquer conteúdo que seja emitido/disponibilizado uma vez deverá estar disponível na rede sempre a partir desse momento. Esses conteúdos poderão estar localizados em servidores dedicados e/ou nos próprios sistemas de televisão clientes (STBs e afins), criando uma rede P2PTV onde os conteúdos estão mais distribuídos e logo mais acessíveis aos utilizadores, sobrecarregando menos sistemas específico. O acesso aos conteúdos por parte dos utilizadores poderá ser efectuado de uma forma completamente livre ou segundo políticas de acesso restrito quando o conteúdo só pode ser acedido por um grupo específico de utilizadores com determinados privilégios, sejam estes devidos a políticas de pagamento pelo conteúdo ou políticas de privacidade. A evolução das redes de telecomunicações estão a levar a que no futuro o utilizador não possua o suporte físico do conteúdo multimédia. Este irá se encontra *online* ou na memória do sistema de televisão (STB e afins).

No futuro da IPTV, cada utilizador poderá criar o seu próprio canal de televisão, podendo também visualizar canais típicos e até fundi-los com conteúdos provenientes de fontes diversificadas que deseja assistir.

Existem actualmente algumas entidades internacionais empenhadas em desenvolver a tecnologia IPTV bem como em tornar padrão as tecnologias utilizadas. Algumas dessas entidades são: *Internet Streaming Media Alliance*, *IPTV Focus Group*, o *Open IPTV Forum*.

2.3 Arquitecturas de Redes IPTV

Existem múltiplas formas de visualizar conteúdos multimédia utilizando o protocolo IP. O modo mais comum na Internet é baseado nas arquitecturas cliente–servidor, dos quais é possível referir dois tipos:

Download–and–Play Arquitectura em que o utilizador necessita de fazer o *download* do conteúdo multimédia e só depois pode visualiza-lo. Com este método o utilizador tem, tipicamente, de esperar pela conclusão do *download* antes de começar a assistir ao conteúdo. Este tipo de arquitectura é inviável para conteúdos em tempo–real;

Streaming Arquitectura em que o utilizador recebe progressivamente o conteúdo enviado pelo servidor, podendo assistir simultaneamente a este enquanto o recebe, tipicamente com um atraso devido à codificação/descodificação, transporte na rede, etc. Com este método o utilizador não tem de esperar pela transferência completa do conteúdo antes de começar a usufruir do mesmo, podendo assistir a conteúdos em tempo–real.

Para além de arquitecturas cliente–servidor, uma outra forma de transferir dados muito utilizada hoje em dia na Internet é conhecida por P2P. No caso desta arquitectura ser utilizada na transferência de *streamings* com conteúdos multimédia (televisão), apelida-se normalmente de P2PTV. A figura 2.11 esquematiza as diferenças entre uma rede do tipo cliente–servidor e uma rede do tipo P2P.

De uma forma geral, as redes P2P são baseada em redes cooperativas, onde os utilizadores estão ligados entre si e partilham informação ao mesmo nível, substituindo a noção de clientes e servidores pela noção de *peers*. Ao mesmo tempo que um utilizador pode estar a receber um ficheiro, este poderá estar também a partilhar o mesmo e/ou outros na rede com múltiplos utilizadores. Deste modo cria-se uma arquitectura de rede totalmente distribuída onde os dados se encontram replicados ao longo dos nós da mesma.

As redes P2PTV podem diferir de uma rede P2P típica na medida em que podem possuir nós que actuam como *super–peers*, colocando e actualizando os conteúdos na rede. Estes *super–peers* actuam como servidores numa primeira instância, disseminando inicial-

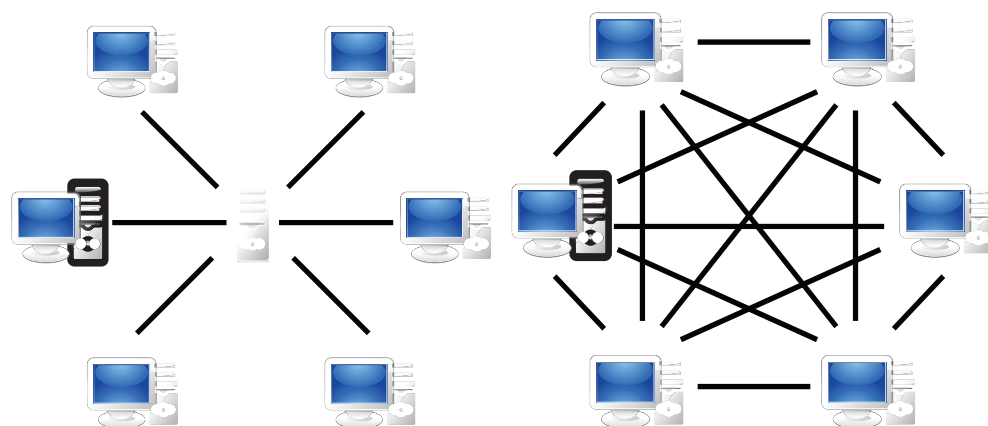


Figura 2.11: Esquemas exemplificativos das arquitecturas cliente-servidor e P2P, respectivamente.

mente os conteúdos pelos *peers*. Esta arquitectura híbrida P2P e cliente-servidor permite não sobrecarregar um servidor único de *streaming*, tornando assim o serviço mais distribuído e logo menos susceptível a falhas. A figura 2.12 representa esquematicamente uma rede P2PTV.

Sistemas P2P como o *BitTorrent* foram alvo de testes para funcionamento com conteúdos multimédia numa política de *streaming* distribuído, tendo-se obtido resultados muito positivos, revelando uma carga na rede bem distribuída aliada a uma qualidade e disponibilidade multimédia muito satisfatórias. O problema mais notório destas arquitecturas P2P é a possível falta de segurança e o atraso no início do *streaming*. Sistemas como o popular *PPLive* ou o *CoolStreaming*[48], implementam um sistema IPTV sobre uma rede *mesh-pull P2P*[31] criando assim um sistema P2PTV com uma tecnologia semelhante à utilizada no protocolo *BitTorrent*. Nesta tecnologia os utilizadores partilham os conteúdos ao mesmo tempo que os estão a assistir ou descarregar. Tipicamente, existe um pequeno número de *super-peers* (ou “servidores”), funcionando como uma *proxy* de vídeo, que têm um papel muito importante na colocação inicial e actualização dos conteúdos. Isto possibilita a disseminação de conteúdos de uma forma distribuída balanceando a carga na rede e melhorando quer as medidas de desempenho quer a disponibilidade dos dados.

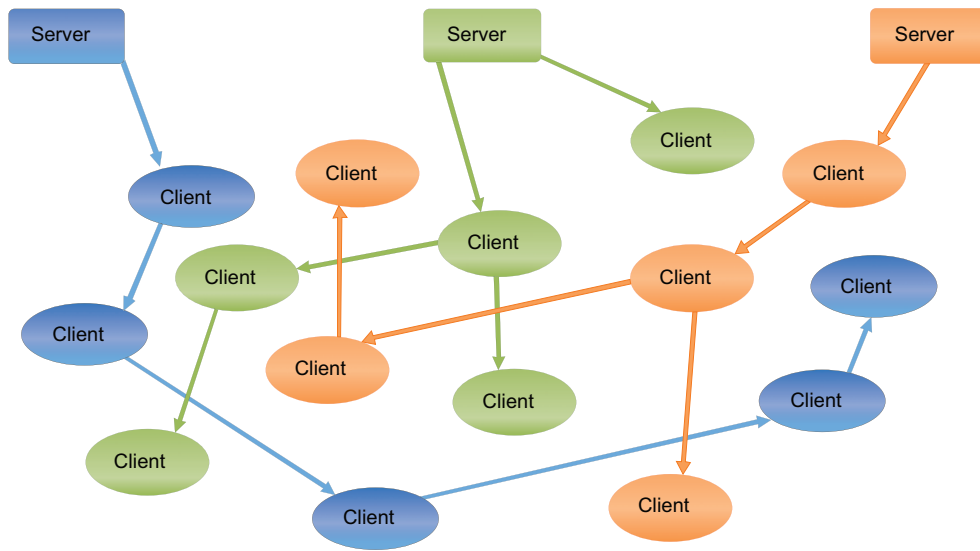


Figura 2.12: Exemplo esquemático de uma rede P2PTV.

(Figura retirada de: <http://upload.wikimedia.org/wikipedia/commons/6/6e/P2PTV.svg>)

Mesmo utilizando um sistema P2P, os sistemas IPTV podem necessitar da utilização de políticas de QoS para garantir de uma forma inequívoca um serviço fiável e constante com elevado QoE.

Devido ao constante crescimento na utilização dos sistemas P2P para partilha de ficheiros, estes são actualmente um dos maiores responsáveis pelo tráfego na Internet a nível mundial. Isto leva a que alguns ISPs tentem aplicar políticas de rede que retiram alguma prioridade aos dados enviados por este tipo de programas, condicionado de forma prejudicial o bom funcionamento deste tipo de redes.

Segundo a ITU, a arquitectura funcional de um sistema IPTV pode ser baseada nos componentes e na tecnologia de redes existentes, ou numa arquitectura *Next Generation Network* (NGN). Esta abordagem conduz a três possíveis opções de arquitecturas IPTV[20]:

- Arquitectura funcional de IPTV para componentes de redes *non-NGN* (incluindo as redes existentes);
- Arquitectura funcional de IPTV baseada numa arquitectura NGN, mas não baseada

em IMS;

- Arquitectura funcional de IPTV baseada numa arquitectura NGN e IMS.

Os quatro domínios funcionais de um sistema IPTV, conforme ilustrado na figura 2.13, são os seguintes:

Provedor de Conteúdos Entidade que possui ou está habilitada para distribuir os conteúdos;

Prestador de Serviço Entidade que fornece o serviço IPTV ao utilizador final. Tipicamente, adquire ou licencia os conteúdos pertencentes a um provedor de conteúdos e coloca-os num pacote que disponibiliza ao utilizador final. Este servidor fica também incumbido de toda a gestão e monitorização do serviço IPTV;

Provedor de Rede Entidade que liga o utilizador final ao Prestador de Serviço. Pode ocorrer facilmente que o Prestador de Serviço e o Provedor de Rede sejam a mesma entidade. Como exemplo disso temos o serviço IPTV *Meo* que utiliza a rede da PT, entidade à qual pertence;

Utilizador Entidade final que consome e paga pelos conteúdos IPTV disponibilizados.

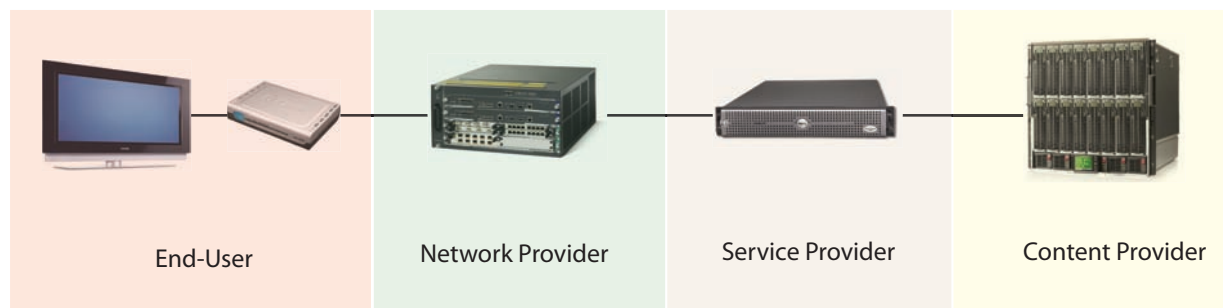


Figura 2.13: Domínios funcionais do IPTV segundo o ITU-T.

(Figura baseada em:

http://www.itu.int/dms_pub/itu-t/oth/19/04/T19040000020001MSWE.doc)

Como é possível verificar, a arquitectura funcional ilustrada em 2.13 é bastante geral e simplificada. Cada domínio funcional pode ser expandido na sua arquitectura específica

que pode variar consoante a implementação. No caso desta abordagem, o servidor IPTV implementado enquadra-se no domínio funcional apelidado por “Prestador de Serviço”.

À data de escrita desta Dissertação, muitas outras entidades se dedicavam ao desenvolvimento de soluções IPTV, quer de carácter mais científico ou mais comercial. Assim sendo, torna-se muito difícil mencionar e explicar a arquitectura específica utilizada por cada uma, não sendo também um objectivo desta Dissertação de Mestrado o estudo exaustivo das mesmas.

De qualquer forma, salientam-se empresas como a gigante *Cisco* que possui soluções integradas de IPTV *end-to-end*. É possível ver a arquitectura desta solução consultando o *site* http://www.cisco.com/en/US/netsol/ns610/networking_solutions_solution_category.html.

2.4 Tecnologias de Desenvolvimento

Em termos de tecnologias disponíveis para a construção de um sistema deste tipo, é possível encontrar um leque variado de diferentes soluções para as várias componentes do servidor IPTV implementado.

Foram valorizadas tecnologias que possibilitam um alto nível de escalabilidade, que permitam definir arquitecturas distribuídas e que fossem o mais independentes possível de pormenores de implementação bem como da arquitectura computacional utilizada.

Tal como já foi referido, o sistema desenvolvido não contempla os mecanismos de codificação dos conteúdos multimédia bem como o seu transporte na rede. Este foca-se na agregação e aquisição de informações relativas aos conteúdos para futura classificação, bem como na persistência de todos os dados disponíveis quer sobre os conteúdos quer sobre os utilizadores para aplicação nos algoritmos de aprendizagem dinâmica do perfil destes.

A índole base deste trabalho assenta na personalização automática de canais, que é possível graças à persistência dinâmica sintetizada das preferências do utilizador, bem como o seu histórico e variáveis relativas a um conjunto de parâmetros que reflectem o estado do utilizador.

Assim sendo, é crucial a existência de uma base de dados para alojar todas os dados

necessários. As tecnologias de base de dados consideradas foram:

- XML (<http://www.w3.org/XML/>)
- *MySQL* (<http://www.mysql.com/>)
- *Oracle Database* (<http://www.oracle.com/database/index.html>)
- *PostgreSQL* (<http://www.postgresql.org/>)
- *Firebird* (<http://www.firebirdsql.org/>)
- *Microsoft SQL Server* (<http://www.microsoft.com/SQL/default.msp>)

De entre a lista de opções consideradas, o SGBD escolhido foi o *MySQL*. As principais razões assentam no facto de ser de utilização gratuita, muito estável para sistemas de pequena e média dimensão, de fácil utilização e manutenção e ainda muito disseminado globalmente, o que significa que muitos sistemas e aplicações o suportam. Para mais informações sobre *MySQL*, é possível consultar a sua página oficial em <http://www.mysql.com/>. A secção 4.5 refere-se ao módulo de persistência de dados do sistema e contém mais informações sobre as tecnologias de base de dados adoptadas.

Mesmo não sendo utilizado para efectuar persistência dos dados relativos aos conteúdos ou aos utilizadores, foi usado XML para dar persistência a algumas configurações específicas do sistema bem como na política de comunicação com os clientes.

Para além do sistema de base de dados, foi também necessário escolher qual a linguagem de programação das funcionalidades nucleares pretendidas. As linguagens de programação consideradas foram *C/C++*, *Java* e *C#*.

Analisadas as opções consideradas, foi determinada a utilização da linguagem *Java* da *Sun*, devido a ser uma linguagem (teoricamente) independente da arquitectura do sistema, estável, muito intuitiva e clara, de funcionalidade facilmente extensível e actualmente de uso muito comum nos mais variados tipos de sistemas. Os motivos da escolha desta linguagem em detrimento das restantes consideradas não se ficam por aqui. A existência da *framework Hibernate* para *Java* que possibilita o mapeamento objecto-relacional no

acesso a bases de dados relacionais (como é o caso do *MySQL*), também influenciou muito a escolha. Para mais informações sobre o *Hibernate*, consultar a página oficial em <https://www.hibernate.org/>. A secção 4.5 refere a utilização do *Hibernate* no sistema, articulando-o com as restantes tecnologias utilizadas.

Relativamente ao ambiente de desenvolvimento (*Integrated Development Environment* (IDE)), foi utilizado o *NetBeans*. Esta escolha foi efectuada tendo em conta a existência de um grande número de ferramentas deste tipo, como é o caso do *Eclipse*, do *JCreator*, *JBuilder*, *DrJava*, entre outras. As razões que levaram à utilização do *NetBeans* como IDE no desenvolvimento do servidor IPTV basearam-se nos seguintes factores: é uma ferramenta muito intuitiva, fácil de utilizar e versátil; possui todas as ferramentas para arquitectar e dimensionar um *software* de média e grande complexidade, integrando num só ambiente um grande número de funcionalidades, permitindo tarefas desde a modelação do sistema a implementar, passando pelo seu desenvolvimento, instalação e *debug*/testes; o autor desta Dissertação possuía *à priori* alguma experiência e conhecimento no uso da ferramenta.

Na secção 4.2 é possível encontrar mais informações sobre as tecnologias utilizadas na concepção do núcleo do sistema.

No que respeita ao protocolo de comunicação entre o cliente e o servidor IPTV, foram definidas mensagens utilizando XML e *XML Schema Definition* (XSD) que possibilitam a comunicação via HTTP entre ambos de forma organizada e estruturada. Este modelo de comunicação é independente da arquitectura do cliente, tendo por base um sistema cliente-servidor em que o cliente despoleta sempre a comunicação. Este tópico é abordado de forma mais extensa na secção 3.4.

Desde o início da projecção do sistema que ficou definido que a comunicação seria efectuada com recurso a uma estrutura XML sobre HTTP, tal como acontece tipicamente nos *Web Services*. Assim, é possível a construção, de forma natural, de clientes IPTV segundo abordagens tão distintas como soluções *Desktop*, *Web* ou *Mobile*, mantendo a mesma política de interacção com o servidor.

Relativamente ao modo como o sistema adquire os *links* para os conteúdos multimédia

e toda a sua informação, foram consideradas três abordagens que funcionam no sistema simultaneamente: *feeds* RSSs provenientes dos provedores de conteúdos; APIs dos provedores dos conteúdos se estas estiverem disponíveis; Introdução directa dos dados dos conteúdos por parte dos utilizadores dos sistema.

Em qualquer uma das formas anteriores, o sistema adquire toda a informação disponível sobre o conteúdo multimédia mas não guarda o conteúdo propriamente dito. Informações mais detalhadas sobre esta componente do sistema podem ser encontradas na secção 4.3.

No que se refere à ferramenta de gestão e monitorização do sistema (*IPTV Server GUI*), foram consideradas inicialmente as seguintes tecnologias:

- HTML + PHP + JS (*JQuery*)
- *Zk* (<http://www.zkoss.org/>)
- *JavaFX* (<http://javafx.com/>)
- JSP ou JSF (<http://java.sun.com/>)
- *Java Swing* + *SwingX* (<http://java.sun.com/javase/technologies/desktop/>)

Devido à simplicidade requirida e também dado que a ferramenta de gestão e monitorização é actualmente utilizada maioritariamente para *debug* do sistema, foram utilizadas as tecnologias *Java Swing* e *SwingX* segundo uma política de aplicação *Desktop* multiutilizador. Para um esclarecimento mais profundo sobre a ferramenta desenvolvida, consultar a secção 4.6.

2.5 Conclusão

É facilmente verificável que no decorrer dos anos os sistemas de TV têm evoluído constantemente a todos os níveis. Estes têm cada vez mais impacto nos hábitos da sociedade moderna, verificando-se mesmo que muitas pessoas já quase não conseguem passar sem ver os seus conteúdos de interesse, sejam eles conteúdos noticiosos, cinematográficos, políticos, desportivos ou outros.

Em traços gerais, é possível afirmar que os sistemas IPTV actuais podem ser considerados como somente um vislumbre daquilo que poderão ser no futuro, dadas todas as possibilidades que estes permite.

É de considerar que nos últimos anos tem-se assistido a um maior investimento nesta área, o que a permitiu passar do conceptual testado nos laboratórios para sistemas reais em utilização por todo o mundo e em especial em Portugal. Mesmo assim o estado tecnológico actual destes sistemas ainda está muito aquém do espectável, e sendo a IPTV considerada a televisão do futuro, a tendência deverá ser para investir nesse mesmo futuro.

É visível que as principais entidades que estão por detrás dos avanços tecnológicos nos sistemas de IPTV já pensam na essencial necessidade de convergência dos meios de visualização, heterogeneidade e agregação dos conteúdos e funcionalidades, para que deste modo seja possível servir mais e melhor as necessidades do utilizador. Só deste modo será possível expandir o conceito de televisão a dispositivos e tecnologias que até então não possuíam essa funcionalidade, bem como transpor o conceito de televisão e criar novas funcionalidades em diferentes tecnologias e equipamentos.

Durante toda a pesquisa sobre tecnologias de TV e mais especificamente sobre IPTV, foi concluído que existem múltiplas soluções de um vasto número de empresas (algumas delas de grande dimensão) a trabalhar nessa área, bem como de grupos com parcerias tecnológicas várias. A grande maioria destas focam-se essencialmente nas formas de transmissão *end-to-end* dos conteúdos na rede, preocupando-se com temáticas como o transporte sobre IP, QoS, QoE, controlo e codificação dos sinais de vídeo, descuidando a classificação, caracterização e agregação de conteúdos multimédia. É nesta componente que este trabalho visa dar uma modesta contribuição e, mais que isso, fomentar algum desenvolvimento futuro.

Mesmo não fazendo parte dos objectivos desta dissertação, a mesma é sensível a temáticas como a integração de novos serviços e tecnologias nos sistemas IPTV, bem como a criação de novas funcionalidades e reinvenção dos conceitos já existentes, a alta disponibilidade do serviço e a heterogeneidade dos meios de acesso ao mesmo. Todas estas características são factores preponderantes para o sucesso e proliferação do serviço.

Capítulo 3

A Arquitectura do Servidor: Análise Funcional

3.1 Introdução

Este capítulo tem como principal objectivo a apresentação de forma breve e geral da arquitectura do sistema, sendo que a especificação exhaustiva de cada módulo que o constitui será enunciada no capítulo 4. São também aqui especificados os requisitos propostos para o sistema que levaram à adopção da sua arquitectura, bem como a metodologia geral do modelo de comunicação utilizado.

3.2 Requisitos do Sistema

Antes de formular a arquitectura do sistema, foi necessário analisar as necessidades e requisitos deste de forma a desenhar a melhor solução possível que reúna as características pretendidas pelo sistema proposto no âmbito desta Tese de Mestrado. A figura 3.1 mostra de forma gráfica os requisitos globais pretendidos depois da análise efectuada.

É de notar que esta análise de requisitos prende-se com as necessidades de suporte globais do sistema, numa visão futura de trabalho, e não necessariamente com os requisitos obrigatórios para esta dissertação. Os requisitos obrigatórios para esta tese encontram-se no

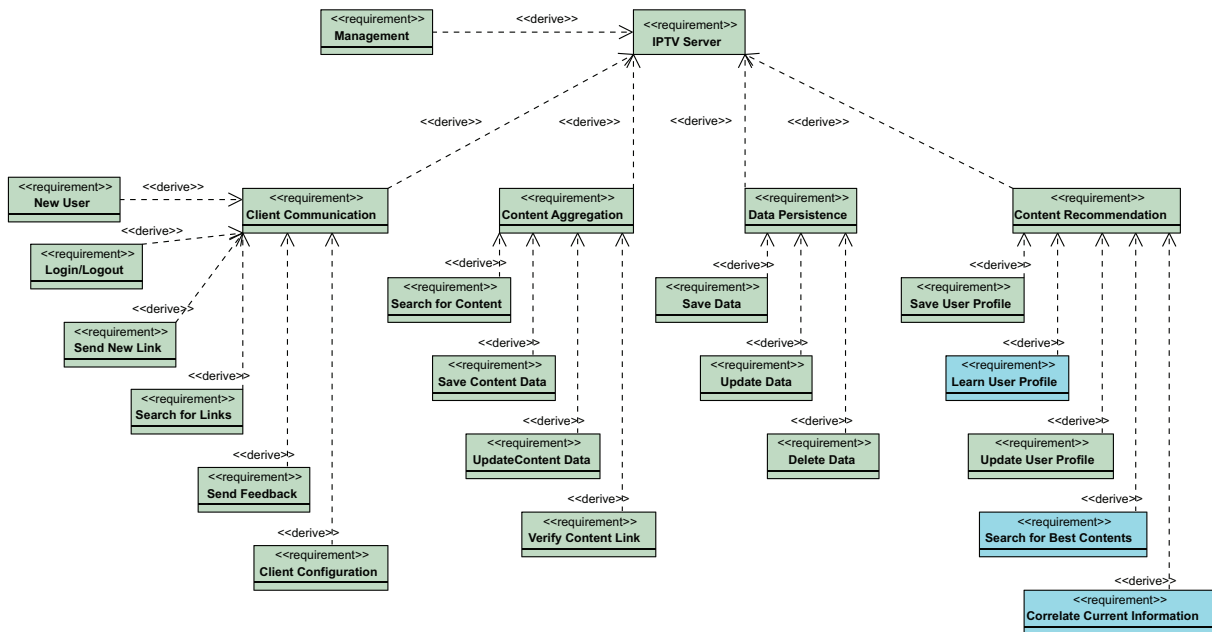


Figura 3.1: Requisitos globais do servidor IPTV. A azul: para futura implementação.

grafismo 3.1 com fundo verde. A azul é possível observar os requisitos cuja implementação não faz parte deste trabalho mas que o mesmo criou condições para futura introdução.

Depois de elaborada a lista de requisitos do sistema, procedeu-se à construção do diagrama de casos de utilização (*use cases*) em *Unified Modeling Language* (UML) que representa de uma forma genérica as funcionalidades do servidor no seu todo.

O diagrama elaborado encontra-se nesta Dissertação na figura 3.2. À semelhança do diagrama de requisitos do sistema (3.1), os casos de utilização representados com fundo azul indicam funcionalidades que ainda não foram implementadas, enquanto que os casos de utilização com fundo verde dizem respeito a funções completamente operacionais no servidor.

3.3 A Arquitectura

A arquitectura base do sistema desenvolvido baseia-se em quatro módulos distintos com funcionalidades específicas, com a possibilidade de serem distribuídos por várias máquinas,

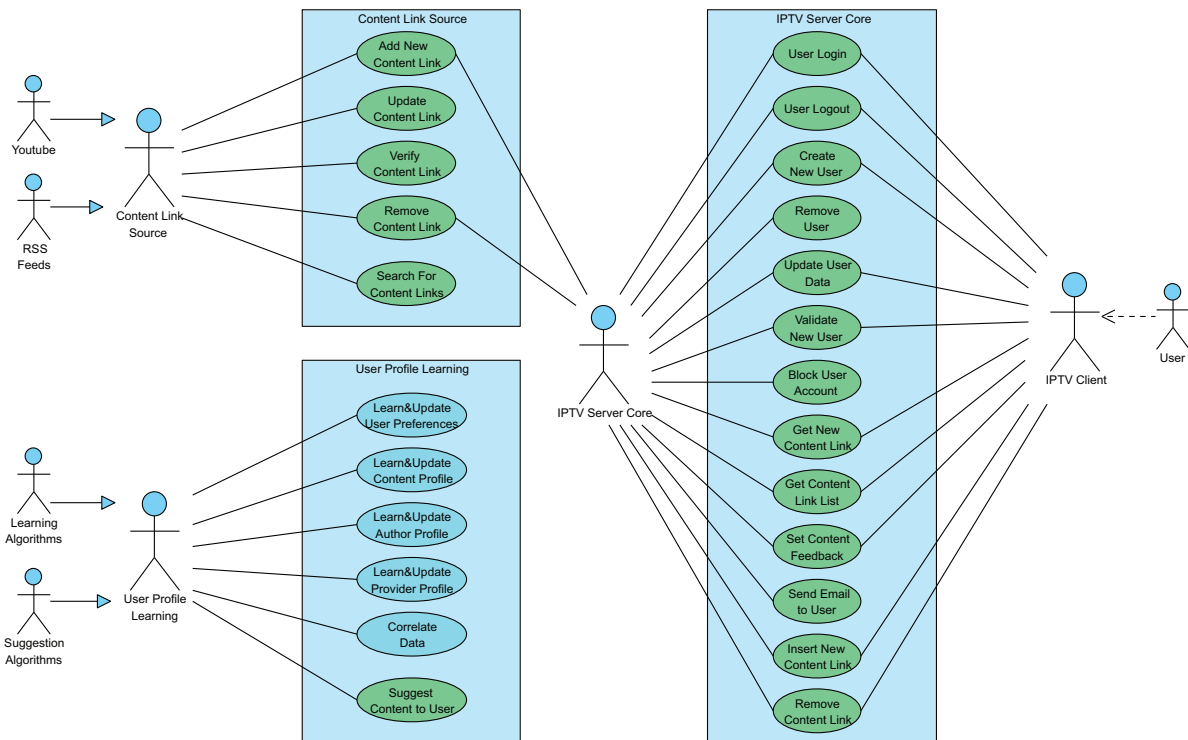


Figura 3.2: Casos de utilização genéricos do servidor IPTV. A azul: para futura implementação.

interligadas através de uma rede TCP/IP. De entre os módulos construídos existe um que possui uma maior importância funcional para o correcto funcionamento do sistema, sendo mesmo considerado o “ponto crítico” deste. Este é a base de dados do servidor IPTV ou também denominado nesta dissertação por *IPTV Database* (secção 4.5). Usado pelos restantes módulos para partilha da informação referente a todos os actores presentes no sistema, este possui um papel crucial no que diz respeito à sincronização e partilha dos dados. A figura 3.3 mostra o diagrama da arquitectura geral do servidor no qual é possível ver o carácter modular do mesmo.

Na figura 3.3, a orientação das setas referentes às ligações entre os diferentes módulos mostram o sentido das dependências e as etiquetas das mesmas o protocolo base de comunicação. Ainda nestas, o estereótipo etiquetado como `<< import >>` significa que a dependência é interna (a funcionalidade está embebida no módulo), enquanto que o es-

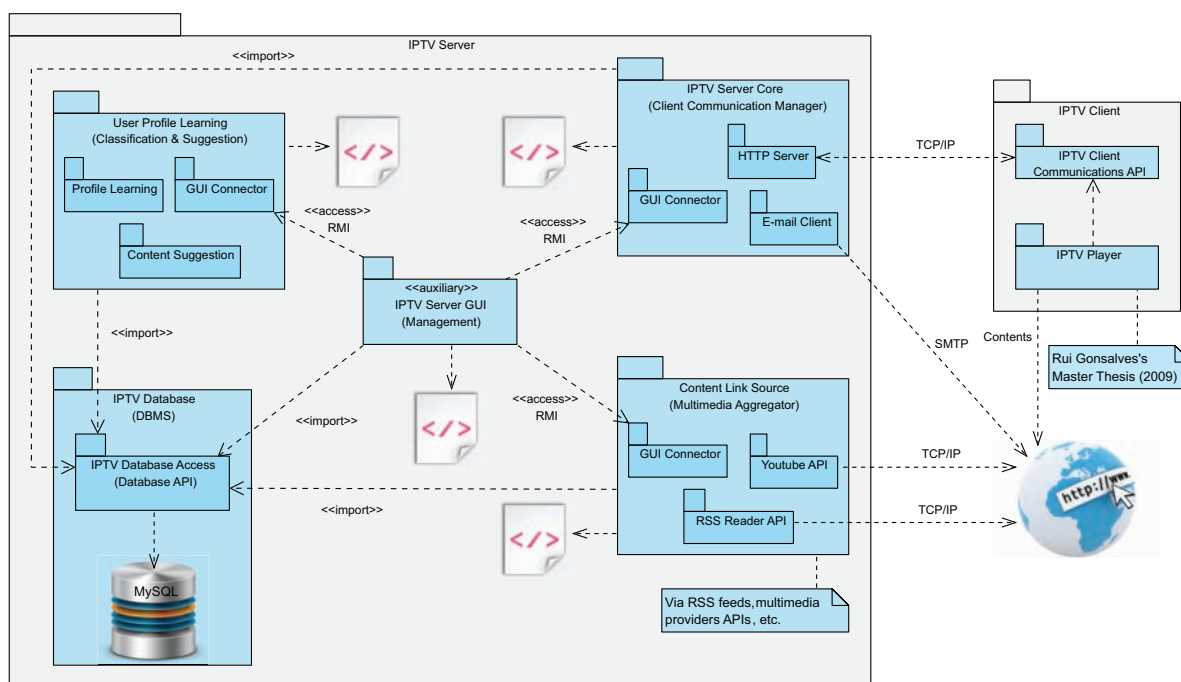


Figura 3.3: Diagrama geral da arquitectura do servidor IPTV implementado.

tereótipo `<< access >>` significa que a dependência existe devido ao módulo em questão necessitar de aceder e/ou modificar dados remotos que não lhe pertencem.

Cada módulo possui uma instância da API de acesso à base de dados (*IPTV Database Access*). O acesso aos dados presente na base de dados é efectuado de forma exclusiva com recurso a essa mesma API, criada para o efeito e que mapeia todas as entidades em objectos *Java*. O DBMS utilizado actualmente é o *MySQL*, sendo que devido ao facto de toda a arquitectura do sistema se basear na utilização da API de acesso à base de dados, é extremamente fácil a mudança de DBMS, sendo necessário somente alterações mínimas na configuração da API de acesso. Isto cria uma camada lógica que abstrai as especificidades de implementação da base de dados dos módulos que a utilizam. A figura 3.4 traduz, de forma gráfica, o enquadramento da API de acesso à base de dados no sistema. No que respeita à introdução de novas funcionalidades ou alteração das existentes, esta camada intermédia de acesso aos dados mostra-se novamente de extrema utilidade e importância no que respeita à agilização e correcção dos processos, restringindo a responsabilidade de

possíveis erros e problemas de acesso aos dados a uma só entidade.

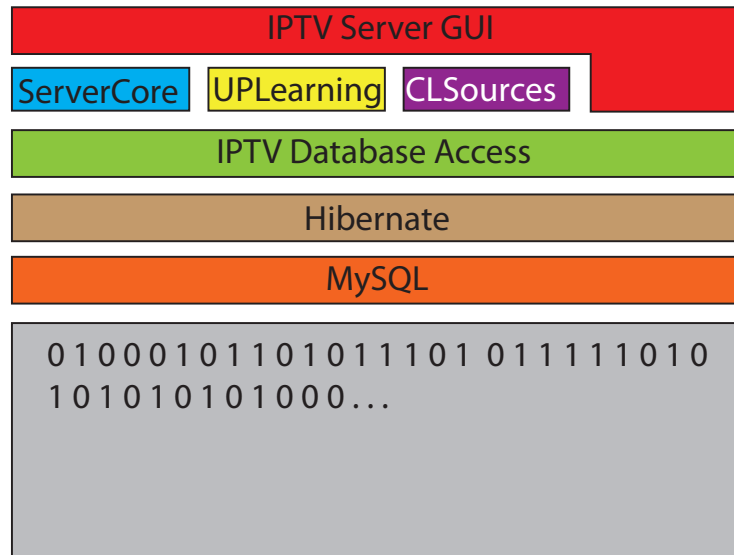


Figura 3.4: Enquadramento da API de acesso à base de dados do servidor IPTV.

Excepto no caso da ferramenta de gestão remota do sistema, todos os dados fluem sempre entre o módulo que os requer e a base de dados, sendo que nunca existe comunicação directa entre diferentes módulos. Cada um dos módulos possui um ficheiro XML de configuração do mesmo (representado no diagrama 3.3 pelo símbolo `< / >` a cor vermelha), sendo que este pode ser alterado de forma directa (editado num editor de texto) ou através da ferramenta de gestão remota do sistema. De qualquer forma, para que as novas configurações sejam utilizadas, é sempre necessário reiniciar o módulo em causa. Este ficheiro de configuração confere a cada módulo persistência estruturada das suas configurações específicas, o que se traduz num sistema facilmente reconfigurável, robusto e flexível. Seguidamente é possível encontrar um exemplo de um ficheiro XML de configuração do módulo *IPTV Server Core* a título exemplificativo.

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <ServercoreSettings xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
3   <serverExternalAddress value="iptvserver.servehttp.com" />
4   <configFilePath value="C:\IPTVFiles\config\" />
5   <xmlFilePath value="C:\IPTVFiles\xml\" />
6   <clientPort value="2323" />
7   <sslclientPort value="2322" />
8   <sslpassword value="jffr3559" />
9   <guiPort value="2324" />
10  <databasePort value="3306" />
11  <databaseName value="iptvdatabase" />
12  <databaseAddress value="localhost" />
13  <databaseUser value="joaoffr" />
14  <databasePass value="jffr3559" />
15  <rspDelay value="0" />
16  <userSessionTimeout value="5000" />
17  <tooManyAttemptsLockTime value="5000" />
18  <remoteObjUser value="root" />
19  <remoteObjPass value="admin" />
20 </ServercoreSettings>
```

A ferramenta de gestão remota permite, utilizando uma interface gráfica intuitiva e simples, alterar todas as configurações correntes de cada módulo do servidor, iniciar, parar ou reiniciar um módulo específico. Para além disso, permite monitorizar não só os eventos ocorridos nos módulos mas também os recursos disponíveis/utilizados no sistema computacional onde cada módulo está a ser executado, entre outras funcionalidades. Aquando a acção de “salvar” determinadas configurações de um qualquer módulo, estas são reescritas no respectivo ficheiro de configuração sendo que a sua aplicação efectiva no respectivo módulo só acontece depois deste ser reiniciado.

A aplicação de gestão remota foi construída orientada segundo uma metodologia de programação *Desktop*, e pode ser utilizada em qualquer sistema computacional que suporte a máquina virtual de *Java*. É possível o controle de um só módulo em particular ou de todos simultaneamente, sendo também possível que este controlo seja efectuado através um número virtualmente infinito de instâncias da ferramenta de gestão executadas no mesmo computador ou em vários. Isto deve-se à política multi-utilizador implementada. A monitorização de eventos ocorridos nos módulos acontece em tempo real e a alteração das configurações presentes só poderá ser levada a cabo depois de guardadas através de

uma acção específica na interface. A figura 3.5 mostra a interface principal da ferramenta de gestão sem estar ligada a qualquer módulo, onde é possível seleccionar o(s) módulo(s) a gerir. Esta imagem foi capturadas durante a execução da ferramenta em ambiente *Windows 7 build 7100*.

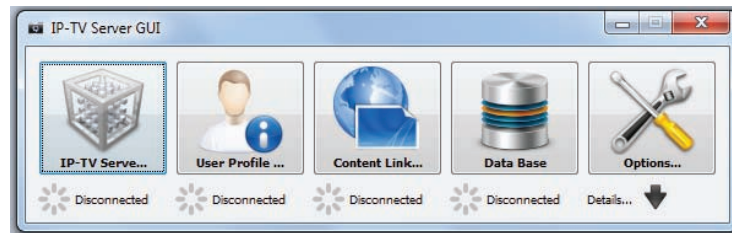


Figura 3.5: Interface principal da ferramenta de gestão do servidor.

Para além dos módulos atrás mencionados e das APIs cliente (secção 5 e de acesso à base de dados (secção 4.5), foram ainda construídas algumas APIs com pequenas funcionalidades, que reduzem a redundância e facilitam a reutilização de código produzido. Tarefas como o envio de *e-mails*, *parsing* e construção de mensagens XML ou a modelação de objectos auxiliares, são alguns exemplos de funcionalidades do sistema implementadas através destas pequenas APIs.

3.4 Modelo de Comunicação

Neste capítulo será apresentado e discutido o protocolo de comunicação *XML* sobre *HTTP/HTTPS* desenvolvido. Foram criados diversos formatos de mensagens XML com uma estrutura constante bem definida, que permitem a comunicação cliente-servidor. Na tabela 3.1 é possível encontrar a lista das mensagens implementadas e qual a sua função. A especificação pormenorizada dessas mesmas mensagens encontra-se no anexo B.

Os principais motivos da adopção de uma comunicação baseada em XML sobre HTTP (ou HTTPS) foram:

- O formato XML é actualmente padrão na comunicação e persistência de dados nos mais variados sistemas;

- Por si só, o XML estabelece uma relação hierárquica complexa entre os dados, o que permite uma excelente organização;
- Tanto no HTTP como no XML a informação é codificada em texto o que facilita a sua compreensão e detecção de possíveis erros;
- O XML suporta validação sintática e semântica (através da linguagem XSD, por exemplo), permitindo a definição de regras de validação;
- Existe actualmente um enorme conjunto de bibliotecas para manipulação de conteúdos XML em quase todas as linguagens de programação;
- O protocolo HTTP está fortemente implementado e qualquer ferramenta pode utilizá-lo;
- O XML em conjunto com o HTTP permite a fácil construção de clientes utilizando múltiplas abordagens (*Desktop, Web, Mobile, etc.*);
- O HTTP garante a entrega e correcção dos dados que transporta, podendo ainda ser utilizada encriptação *Secure Sockets Layer* (SSL) ou *Transport Layer Security* (TLS) (HTTPS);
- Criação de um nível de abstracção entre os pormenores das implementações das entidades que desejam comunicar.

No contexto das Tecnologias de Informação, segurança entende-se basicamente como quatro propriedades da informação: integridade, disponibilidade, autenticidade e confidencialidade. Não era um dos objectivos deste trabalho a implementação de políticas de segurança na comunicação cliente-servidor. Mesmo assim, foi implementado um mecanismo básico de criação, validação e início de sessão de utilizadores baseados nas credenciais *user name* e *password* (guardada no servidor encriptada através do algoritmo *Message-Digest algorithm 5* (MD5)).

É de notar que cada conta criada pelos utilizadores terá de ser validada antes da sua primeira utilização. Para isso, o servidor IPTV procede ao envio de um e-mail para o

utilizador que se acabou de registar com o seu *username* e *password* actual codificada em MD5. O cliente IPTV terá de fornecer um interface ao utilizador onde este possa colocar o seu *username* e *passwords* antiga, nova e confirmação da nova de modo a activar a conta. Quando a conta estiver activa, o servidor notificará novamente o utilizador, informando agora este novo estado da sua conta no sistema.

Consta também da implementação a atribuição de um identificador único a cada sessão do utilizador e a possibilidade de comunicação através de uma ligação segura HTTPS, que exige a partilha do certificado de acesso. Assim sendo, pode-se admitir a existência de um nível básico de autenticidade e confidencialidade. A integridade dos dados é garantida pelo próprio servidor IPTV mas a disponibilidade deste não está assegurada, podendo ser posta em causa por ataques *Denial-of-Service attack* (DoS), por exemplo.

À primeira vista, o sistema parece não requerer especiais mecanismos de segurança, mas se atentarmos nos dados e serviços que este pode conter e fornecer, o caso pode mudar de figura. Não só o servidor poderá possibilitar o acesso a serviços pagos, o que por si só exige uma forte componente de segurança, mas também poderá conter dados privados sensíveis. Tendo em conta que o servidor foi projectado para guardar e aprender o perfil do utilizador e que isso implica guardar, de alguma forma, o histórico dos conteúdos multimédia visualizados, o acesso de forma indevida por parte de terceiros a esta informação poderá representar uma importante invasão da privacidade.

Tal como já foi mencionado, a comunicação cliente–servidor pode ser efectuada segundo duas políticas: sem encriptação dos dados – HTTP – e com encriptação dos dados – HTTPS. Em ambas as situações a sintaxe e semântica das mensagens XML trocadas é exactamente a mesma. O servidor está preparado para funcionar em ambos os modelos, podendo atender pedidos em paralelo do mesmo ou de diferentes clientes, utilizando tanto HTTP como HTTPS. Ao responder a uma mensagem, o servidor usa o mesmo protocolo que foi utilizado no envio do pedido que despoletou a resposta.

O serviço de atendimento HTTP é conceptualmente independente do serviço de atendimento HTTPS, funcionando em portos diferentes e configuráveis. Ambos são parte integrante do módulo *IPTV Server Core* (secção 4.2). De uma forma simplificada, para esta-

belecer uma comunicação HTTPS, terá de se definir previamente o certificado a utilizar e partilhá-lo com os clientes IPTV que desejam comunicar por esta via. Tanto num como no outro protocolo, é garantida a entrega e correcção das mensagens no seu destinatário, dado ambos funcionarem sobre TCP/IP.

No capítulo 5 é possível encontrar a especificação geral da API cliente construída e alguns exemplos da sua utilização. Esta API em *Java* permite a comunicação HTTP e/ou HTTPS com o servidor com recurso ao mesmo modelo de programação utilizado no servidor. É criada assim uma abstracção de alto nível entre o cliente e o servidor IPTV, permitindo desconhecer pormenores de implementação do modelo de comunicação e utilizando evocações simples para realizar tarefas complexas. A figura 3.6 enquadra a utilização desta API num cliente IPTV.

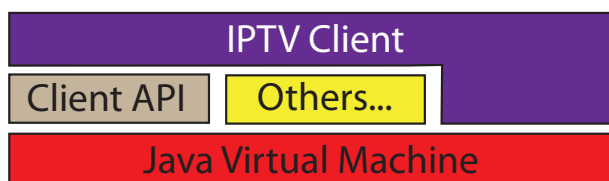


Figura 3.6: Enquadramento geral da API de comunicação com o servidor num cliente IPTV.

3.5 Modelo de Classificação

Decorrente da análise do problema apresentado pela aprendizagem dinâmica do perfil do utilizador com vista a futura sugestão automática e inteligente de conteúdos, o desafio remete-nos para a seguinte questão: Como representar e guardar o conhecimento relativo aos conteúdos multimédia e ao perfil do utilizador em termos dos seus gostos/preferências multimédia, sendo que este conhecimento deve evoluir de forma cumulativa com a utilização do sistema? A resposta encontrada para esta questão foi baseada no que acontece em muitas outras áreas como é o caso da química. Se efectuarmos uma análise da matéria que nos rodeia, do ponto de vista químico, é possível descrever qualquer composto através de uma

fórmula química que articula valores reais e elementos básicos numa organização específica, elementos os quais podemos encontrar na *Tabela Periódica dos Elementos* que é finita e está bem definida e organizada. De uma forma simplista, construído um novo composto à base dos compostos básicos, é possível utilizar este novo composto mais específico e complexo para construir novos compostos ainda mais complexos e refinados.

Do mesmo modo, segundo o ponto de vista aqui apresentado, é possível descrever os gostos do utilizador com recurso a um conjunto finito de áreas de interesse adequado, definido *a priori*, cujos elementos mantêm relações de generalização (herança) entre si. É assim utilizado o princípio de que com um número finito de elementos base é possível criar uma infinidade de configurações (novos elementos diferentes dos que lhe deram origem e mais complexos), se for possível conjugá-los e variar pelo menos um parâmetro que os pese de uma forma teoricamente infinita.

Um outro exemplo da utilização generalizada desta abordagem é o sistema de cores *Red, Green and Blue color model* (RGB). Com este é possível representar todas as cores visíveis ao olho humano pela adição ponderada das componentes *Red, Green* e *Blue*.

As relações de generalização estabelecidas entre as áreas de interesse permite ao sistema sugerir conteúdos de uma forma inteligente, mesmo quando o sistema ainda não possui muito conhecimento sobre as preferências do utilizador. Caso a árvore/grafos de áreas de interesse esteja bem construído e organizado, possuindo um número suficiente de relações entre as várias áreas e sub-áreas consideradas, é possível conferir ao sistema a capacidade de relacionar áreas de interesse distintas que mantenham relações entre si.

Do ponto de vista efectivo, esta característica permite que o sistema classifique o utilizador tendo em conta não só cada área de interesse específica mas também aquela ou aquelas que directamente lhe deram origem bem como aquela ou aquelas que possuam a anterior como área “pai”. Considere-se o exemplo hipotético de que o utilizador *U* demonstrou gostar da área 0.1.3 – “Futebol”. O sistema, ao saber esta pequena informação, pode concluir *a priori* que o utilizador gosta de todas as sub-áreas dos seguintes níveis que tenham como área pai “Futebol”, sendo estas por exemplo 0.1.3.0 – “Futsal”, 0.1.3.1 – “Futebol Americano”, 0.1.3.2 – “Futebol de Praia” e 0.1.3.3 – “*Rugby*”. Para além disso, o

sistema pode considerar ainda que a área “pai” de “Futebol” também deverá ser considerada como interessante para o utilizador. Neste caso seria a área 0.1 – “Desporto”. É de notar que para relações mais complexas deverá ter-se em conta algoritmos que estabeleçam critérios ponderados na classificação e sugestão.

Para além das áreas de interesse definidas inicialmente, o sistema possibilita a construção dinâmica de novas áreas de interesse com base nas anteriores. Para isso, são articuladas múltiplas áreas de interesse “base” ponderadas através de valores reais no intervalo $[0.0, 1.0]$ (que significa $[0\%, 100\%]$), definindo assim novos elementos classificativos que podem, por sua vez, servir de “base” a elementos futuramente criados.

Todos estes elementos classificativos podem ser utilizados, de forma dinâmica, tanto para descrever as preferências do utilizador como para classificar os conteúdos multimédia presentes no sistema. Cada utilizador, conteúdo multimédia, provedor de conteúdos, “grupo virtual de amigos” e outros, são representado por um tuplo (ponto próprio) majorado e minorado através de um conjunto de variáveis. Todos estes parâmetros que identificam cada entidade classificada no universo considerado são ajustáveis.

O ajuste e afinação dos parâmetros numéricos que definem o perfil do utilizador é fundamental para o correcto funcionamento dos algoritmos de sugestão. Estes ajustes podem ser conseguidos de varias formas, sendo que o método pensado faz uso de uma rede neuronal, cujos pesos são os valores anteriormente descritos, e cujos os ajustes se processam quando esta mesma rede evolui gradualmente, aprimorando a definição do perfil do utilizador.

Excepto a área mais global denominada de “raíz” (todo o universo considerado), cada área de interesse tem, obrigatoriamente, pelo menos uma área considerada mais abrangente (área “pai”). Pode-se então considerar que as áreas de interesse estão organizadas de uma forma hierárquica com base em relações de generalização, que, do ponto de vista mais simples, são relações de herança simples representadas em árvore. De forma natural é possível ter também relações de herança múltipla (grafo orientado), onde uma mesma área “filha” poderá possuir múltiplas áreas “pais”. A figura 3.7 mostra um exemplo esquemático de generalização simples (esquema à esquerda) e de generalização múltipla (esquema à

direita).

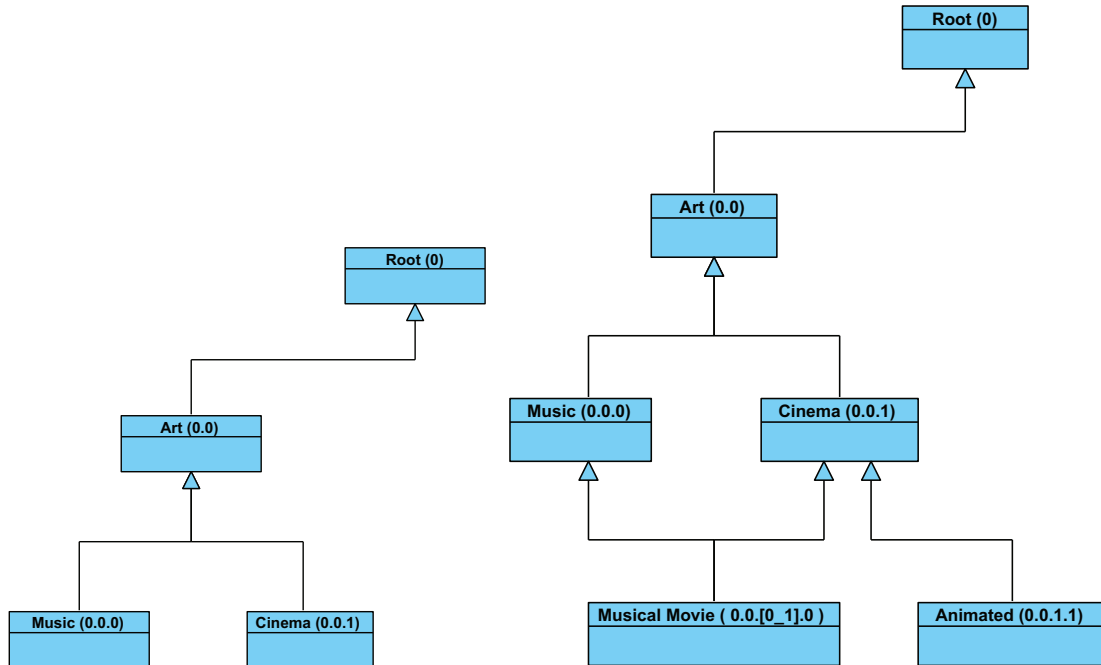


Figura 3.7: Exemplo gráfico de herança simples e herança múltipla, respectivamente.

Esta abordagem não é mais que uma sugestão de modelação do problema utilizando uma estratégia multidimensional (multi-variável) de ordem genérica n . Pretende-se definir um ponto de dimensão n (ponto próprio) específico para cada entidade a classificar, e com essa localização e mais alguma informação dada por variáveis dinâmicas extra, definir uma “nuvem” probabilística em seu redor. Considerando a “nuvem” assim definida para um qualquer utilizador, a probabilidade do mesmo estar interessado num conteúdo cuja “nuvem” intersecte a sua, depende do grau de intersecção. Este grau deve ser grande o suficiente para que o conteúdo em questão seja sugerido ao utilizador com grande probabilidade de sucesso.

O método classificativo utilizado na construção do perfil do utilizador é o mesmo utilizado para classificar os conteúdos, autores e provedores de conteúdos ou grupo de utilizadores, estabelecendo no sistema uma matriz composta por tuplos que classificam especificamente cada entidade relativamente às suas áreas de interesse multimédia.

Do ponto de vista matemático, temos um conjunto de áreas de interesse, organizadas segundo relações de generalização que reflectem a realidade “social”, e que são identificadas inequivocamente por variáveis com a sintaxe $a_i, i \in \mathbb{N}^{0+} \wedge 0.0 \leq a \leq 1.0$. Cada utilizador pode ser assim identificado pelo tuplo $(a_0, a_1, a_2, \dots, a_{n-1}) \in \mathbb{R}^n : 0.0 \leq a \leq 1.0$ definindo assim um ponto num espaço unitário positivo a n dimensões. Se considerarmos também dois diferentes tuplos de igual cardinalidade n , um dos quais serve para identificar um limite superior aditivo e outro um limite inferior subtractivo de a_i , ambos nunca inferiores a 0.0 (0%) nem superiores a 1.0 (100%), definimos assim limites superior e inferior para cada variável associada a uma área de interesse. Estes limites bem como o “ponto próprio” são ajustadas dinamicamente segundo um método específico, reflectindo a classificação da entidade, dependendo do elemento que se está a classificar.

Visto estar a ser considerado uma abordagem multidimensional de ordem n , não é possível proceder à sua representação gráfica de uma forma natural se $n > 3$, o que facilmente se verifica. Assim sendo, foi encontrada uma forma de representar graficamente as variáveis classificativas das áreas de interesse de cada entidade utilizando somente 2 dimensões. Na figura 3.8 está representado de forma exemplificativa o tuplo $(0.150, 0.750, 0.460, 0.375, 0.916, 0.708)$ referente ao “ponto próprio” de um utilizador e também os seus valores limite recorrendo à representação encontrada. É de notar que este tipo de grafismo pode representar áreas de interesse de diferentes níveis hierárquicos, mas é claramente mais ilustrativo e útil se representar um só nível hierárquico simultaneamente. A este tipo de representação foi chamado de “Pegada de Classificação Multimédia” (*Multimedia Classification Footprint*).

Na figura 3.8 as etiquetas “0.1”, “0.2”, “0.3” e assim sucessivamente, identificam as áreas de interesse multimédia, utilizando como identificador a sua composição hierarquia, fazendo uso actualmente de uma estrutura em árvore. Assim sendo, cada dígito separado por ponto (“.”) corresponde a um elemento específico do nível indicado pela sua posição lida da esquerda para a direita, ao estilo dos identificadores das *Management Information Bases* (MIBs) utilizadas pelo protocolo *Simple Network Management Protocol* (SNMP). É de notar que o primeiro elemento é sempre “0” (zero), que corresponde à raiz da estrutura

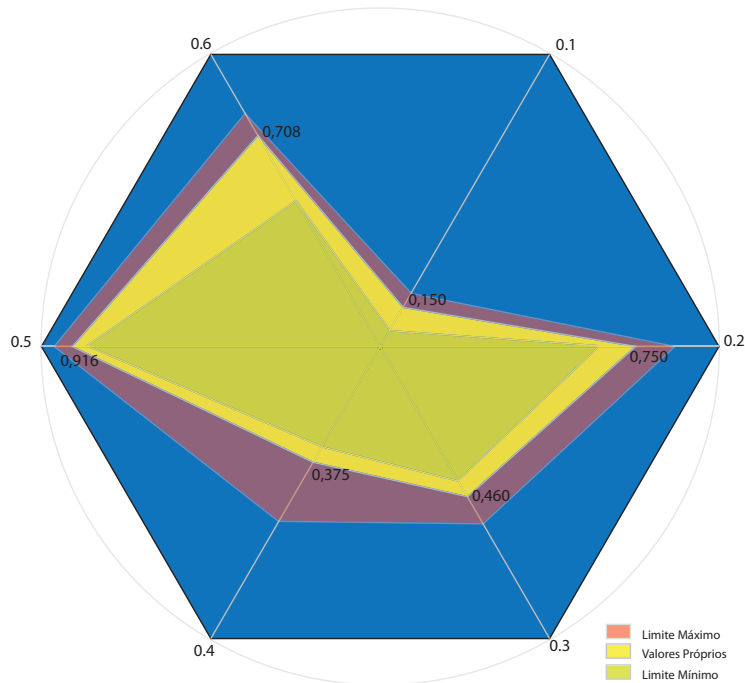


Figura 3.8: Exemplo da representação gráfica do perfil multimédia de um utilizador num instante específico, incluindo limites mínimos e máximos.

de dados. Mais formalmente, para identificar um elemento c genérico pertencente, por exemplo, ao nível 3 da árvore de áreas de interesse, ter-se-ia a sequência $0.a.b.c$, que indicaria que o elemento c possui uma relação de herança directa com o elemento b pertence ao nível 2 e uma relação de herança indirecta com o elemento a pertence ao nível 1. Um exemplo desta representação pode ser visto na figura 3.7 (esquema à esquerda).

Por outro lado, caso se tenha herança múltipla, a sintaxe do identificador varia um pouco. Considerando que o elemento c herda de a e de b simultaneamente enquanto que a e b herdam somente da raíz, o identificador padrão de c seria $0.[a_b].c$. Um exemplo desta representação pode ser visto na figura 3.7 (esquema à direita).

Para além da classificação e aprendizagem do perfil do utilizador tendo em conta a sua experiência de utilização no sistema, é ainda possível utilizar mecanismos de correlação dos perfis de múltiplos utilizadores. Deste modo é possível sugerir conteúdos a um utilizador com base nos conteúdos vistos por outros utilizadores com perfis semelhantes, que irão influenciar a aprendizagem dinâmica do perfil do utilizador caso este imita *feedbacks* rela-

tivos aos conteúdos. Parte-se assim do pressuposto que existe uma grande probabilidade de um utilizador gostar de um conteúdo que outro utilizador já assistiu e gostou se ambos possuírem perfis “minimamente” coincidentes.

De uma forma muito superficial, a descoberta de perfis semelhantes entre utilizadores pode ser conseguida através de técnicas de *data mining*, mais especificamente, através de *clustering*. Para isso, pode ser usado como critério a distância *Euclidiana* entre os pontos próprios de cada utilizador, definidos pelos pesos de ponderação das áreas de interesse. Distâncias abaixo de um determinado limiar indicam perfis “minimamente” semelhantes e logo podem partilhar sugestões de conteúdos multimédia.

Nome da mensagem	Direcção	Descrição da função
<i>User Authentication</i>	cliente ->servidor	Permite que um utilizador se autentique e inicie uma sessão no servidor IPTV
<i>User Deauthentication</i>	cliente ->servidor	Permite que um utilizador termine a sua sessão de forma regular no servidor IPTV
<i>New User</i>	cliente ->servidor	Permite que sejam adicionados novos utilizadores ao sistema de uma forma interactiva
<i>New User Validation</i>	cliente ->servidor	Permite que um utilizador previamente registado efectue a validação e activação da sua nova conta
<i>New Link Request</i>	cliente ->servidor	Permite que um cliente IPTV peça o próximo <i>link</i> para um conteúdo sugerido ao utilizador autenticado no cliente IPTV
<i>New Link Response</i>	servidor ->cliente	Esta mensagem é a resposta enviada pelo servidor IPTV aquando a recepção de uma mensagem <i>New Link Request</i> caso toda a operação ocorra com sucesso
<i>Link List Request</i>	cliente ->servidor	Possibilita que um cliente IPTV peça uma lista de um ou mais <i>links</i> para conteúdos, segundo vários critérios de procura e selecção
<i>Link List Response</i>	servidor ->cliente	Possibilita que o servidor IPTV responda a uma mensagem do tipo <i>Link List Request</i> previamente enviada por um cliente IPTV
<i>Link Feedback Request</i>	cliente ->servidor	Permite que um cliente IPTV envie um <i>feedback</i> relativo a um conteúdo específico para o servidor IPTV
<i>Command Request</i>	cliente ->servidor	Permite que um cliente IPTV envie um pedido de comando ou operação que não exija a autenticação de um utilizador para ser executada
<i>Command Response</i>	servidor ->cliente	Permite ao servidor IPTV responder aos mais variados pedidos de uma forma padrão bem como informar situações de erro

Tabela 3.1: Tabela dos diferentes tipos de mensagens implementadas

Capítulo 4

A Arquitectura do Servidor: Análise Modular

4.1 Introdução

No capítulo 3 foi referida, de uma forma geral, a arquitectura do servidor IPTV implementado. Aqui serão apresentados, discutidos e analisados de forma mais profunda, cada módulo (ou sub-sistema) que compõe a arquitectura do sistema. Nos anexos a este documento é possível encontrar alguma informação mais pormenorizada à cerca de tópicos mais específicos, como por exemplo a especificação das mensagens XML criadas.

É de notar que para uma compreensão mais profunda do ponto de vista técnico e específico relativo à lógica de programação utilizada, é favor consultar a documentação do código *Java* (*Javadoc*) fornecida com o mesmo (ver directoria referente ao código do servidor).

4.2 IPTV Server Core

Neste capítulo será apresentado o módulo que efectua a comunicação cliente–servidor, módulo este denominado de *IPTV Server Core*.

Este módulo possui funções nucleares no que diz respeito ao correcto funcionamento do sistema, visto ser ele que efectua a comunicação entre o servidor e os possíveis múltiplos clientes IPTV.

Este módulo foi o primeiro a ser desenhado e implementado, seguindo uma política de comunicação típica de um sistema cliente–servidor, em que o servidor é passivo na medida em que será sempre o cliente a tomar a iniciativa no envio de pedidos, sendo estes sobre a forma de mensagens XML.

Cada pedido de uma operação ou comando enviado por um qualquer cliente dá origem a uma, só uma e necessariamente uma resposta por parte do servidor. Temos assim um sistema que, no caso do seu correcto funcionamento, nunca deixa de responder a uma mensagem recebida (mesmo que esta não faça sentido) mas também não envia mais que uma mensagem em resposta a um pedido único efectuado.

É possível ter mais que uma instância deste módulo em funcionamento simultâneo, desde que essas mesmas instâncias sejam configuradas para funcionar em portos TCP disjuntos.

Todos os parâmetros passíveis de serem configurados dinamicamente (sem edição do código fonte *Java*) encontram-se num ficheiro XML. Este ficheiro efectua a persistência das configurações deste módulo e pode ser editado manualmente ou então com recurso à utilização da ferramenta de gestão e monitorização do sistema (secção 4.6). Segue-se um exemplo desse mesmo ficheiro de configuração.

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <ServercoreSettings xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
3   <serverExternalAddress value="iptvserver.servehttp.com" />
4   <configFilePath value="C:\IPTVFiles\config\" />
5   <xmlFilePath value="C:\IPTVFiles\xml\" />
6   <clientPort value="2323" />
7   <sslclientPort value="2322" />
8   <sslpassword value="jffr3559" />
9   <guiPort value="2324" />
```

```

10 <databasePort value="3306" />
11 <databaseName value="iptvdatabase" />
12 <databaseAddress value="localhost" />
13 <databaseUser value="joaoffr" />
14 <databasePass value="jffr3559" />
15 <rspDelay value="0" />
16 <userSessionTimeOut value="5000" />
17 <tooManyAttemptsLockTime value="5000" />
18 <remoteObjUser value="root" />
19 <remoteObjPass value="admin" />
20 </ServercoreSettings>

```

É de notar que as *tags* XML utilizadas no ficheiro de configuração são auto-explicativas no que toca à sua semântica. Qualquer questão relativa às mesmas, é favor consultar a documentação *Javadoc* deste módulo presente na directoria do código *Java* do servidor IPTV.

Tal como já foi referido (secção 2.4), foi escolhida como linguagem de programação do servidor a linguagem *Java* da *Sun* devido à sua robustez, flexibilidade, fiabilidade e também por ser uma das linguagem de programação mais populares e usadas em inúmeros campos da computação, trazendo isso enormes vantagens. Outras linguagem de programação foram consideradas (secção 2.4), tendo sido preferido o uso de *Java* em detrimento destas devido também à experiência em *Java* do autor desta Dissertação de Mestrado.

Graças ao mecanismo de *logging* implementado que informa ao gestor do sistema todos os eventos que sucedem no servidor, sejam estes eventos normais ou eventos anormais de baixa, média ou alta gravidade, é possível saber o que se passa ou passou no servidor a nível interno em qualquer momento. A ferramenta de gestão e monitorização do sistema (secção 4.6) faz uso deste mecanismo para informar remotamente e em tempo real sobre os eventos que sucedem no servidor. As informações de cada evento antes de serem descartadas passam por um buffer de registo de eventos que permite recuperar alguns eventos ocorridos anteriormente.

Devido à sua arquitectura concorrente, o sistema tenta sempre recuperar de um erro ou “morte” de um dos seus *threads* prestadores de serviço, lançando outro para o seu lugar. Deste modo, os *threads* que efectuem serviços de comunicação ou execução de tarefas críticas nunca executam eles próprios os serviços, lançando outros *threads* que os

prestam. Isto permite não só a concorrência na execução de tarefas mas também uma maior imunidade e facilidade na recuperação de falhas.

A abordagem utilizada na criação do servidor IPTV pressupõe que o *software* cliente não necessite de saber qualquer configuração ou dados específicos do sistema *a priori*, para além do conhecimento do protocolo de comunicação utilizado, isto se o mesmo não fizer uso da API de comunicação desenvolvida.

Esta característica é conseguida graças ao facto de qualquer configuração poder ser enviada ao cliente IPTV através de uma mensagem *Command Response*, despoletada por uma mensagem *Command Request* (ver anexo B) específica. Este mecanismo tem como objectivo controlar e padronizar as configurações e dados presentes nos múltiplos (tipos de) clientes IPTV, visto este sistema pretender abarcar o maior número de serviços e tecnologias possível.

O módulo aqui referido contempla também o envio de mensagens electrónicas (*e-mails*) para os seus utilizadores em diversas situações. Esta funcionalidade é possível graças à utilização da API *JavaMail 1.4.2* que implementa um cliente de *e-mail* em *Java*.

Algumas das situações passíveis de serem contempladas com o envio de *e-mails* para os utilizadores são:

- Criação de uma nova conta de utilizador no sistema (conta ainda por validar);
- *Feedback* enviado relativo ao acto de validação com sucesso da nova conta (conta validada);
- Recuperação de *password* ou *user name* (actualmente ainda não implementado);
- Excesso de tentativas de *login* falhadas. Para além do utilizador ter de esperar um tempo determinado até poder tentar novamente o *login* no sistema, este será notificado do ocorrido;
- Envio de felicitações no dia de aniversário de cada utilizador (actualmente ainda não implementado);

- Demasiado tempo sem aceder ao sistema (actualmente ainda não implementado);
- *Broadcast* de informações/notícias gerais sobre o sistema enviadas para um grupo de utilizadores. Possibilidade de criação de *mailing lists* segundo diferentes critérios relativos as características dos utilizadores;
- Possibilidade do utilizador enviar *e-mails* a partir do sistema para o gestor do mesmo com sugestões, críticas, dúvidas ou problemas relativas à sua utilização.

É de notar que sempre que o sistema procede ao envio de um qualquer *e-mail* para um ou mais utilizadores, este fica registado na conta de *e-mail* do sistema para futura verificação.

Todas as mensagens descritas no anexo B foram implementadas e encontram-se operacionais. Do mesmo modo, a API de comunicação desenvolvida faz uso de todas estas mensagens segundo a sintaxe descrita no anexo anteriormente referido.

É de notar que para optimizar o funcionamento do módulo *IPTV Server Core*, poder-se-á desactivar as verificações sintáticas das mensagens XML contra os esquemas XSD respectivos descritos no anexo B. Isto levaria a que o sistema confie *à priori* nos dados presentes nas mensagens, efectuando o *parsing* das mesmas com base nestes. Caso uma qualquer mensagem possua um conteúdo inconsistente, o sistema não irá detectar a inconsistência aquando a recepção da mensagem mas sim quando tentar efectuar o seu *parsing* tendo em conta o seu conteúdo, devolvendo o respectivo erro.

Esta alteração tornaria o sistema menos imune a erros e logo menos robusto e flexível, tendo como vantagem um tempo de processamento de cada pedido ligeiramente mais reduzido.

Para além das verificações ao nível da estrutura e semântica XML, é efectuada uma validação sintática e semântica de alguns dos dados presentes nas mensagens depois de feito o seu *parsing* XML. Assim sendo, a fim de evitar a geração desnecessária de erros relacionados com tipos de dados incorrectos, endereços *web* ou *e-mail* com caracteres inválidos/sintaxe não permitida, entre outros, o cliente IPTV deve efectuar a verificação

dos dados a enviar antes destes serem efectivamente enviados para o servidor, sob pena de receber uma mensagem de resposta a reportar erros detectados.

Descrevendo o comportamento geral deste módulo face a cada pedido implementado, têm-se que:

User Authentication A autenticação é efectuada por comparação das credenciais definidas aquando a validação da conta com as fornecidas na tentativa de *login*. Estas têm de coincidir de forma exacta. Caso o utilizador falhe a autenticação três vezes ou mais, fica impossibilitado de aceder ao sistema durante um curto período de tempo e é notificado via *e-mail* do ocorrido;

User Deauthentication O fecho da sessão no sistema poderá ser manual ou automático, dependendo se é o utilizador o efectuar voluntariamente ou se é o próprio sistema a despoletá-lo. Caso o utilizador efectue *login* no servidor em outro sistema computacional, a anterior sessão é encerrada e a nova é iniciada. Caso a sessão esteja inactiva durante um longo período de tempo, este módulo efectua também o fecho automático da sessão para permitir a libertação dos recursos alocados;

New User Esta tarefa permite criar um novo utilizador no sistema bem como iniciar a construção do seu perfil de preferências multimédia. Caso as credenciais e/ou os dados relativos ao perfil já existam no sistema, é devolvido uma resposta que indica que a criação do registo falhou e a descrição dos motivos. Um novo utilizador do sistema necessita sempre de validar a sua nova conta. Para isso, o servidor envia um *e-mail* para o utilizador com o *user name* criado e uma *password* que deverá alterar ou confirmar;

New User Validation Depois de criada uma nova conta no servidor, o utilizador terá de a validar. Este requisito é executado por uma mensagem/tarefa do tipo *New User Validation*. Logo após a conta ser validada, é enviado um *e-mail* para o utilizador respectivo a informar que este poderá efectuar o seu primeiro *login*;

Session Keep Alive Request Do ponto de vista efectivo, esta mensagem somente leva

o servidor a actualizar a data e hora da última actividade de determinado utilizador no sistema, evitando temporariamente que o *trigger* que efectua periodicamente as verificações de *session timeout* faça *logout* automático do utilizador;

New Link Request No que respeita à resposta enviada a uma mensagem *New Link Request*, os *links* para os conteúdos são enviados ao utilizador por ordem decrescente de data de sugestão, ou seja, do mais recentemente sugerido para o mais antigo. Estas sugestões continuam presentes na respectiva tabela da BD até ser recebido pelo servidor algum *feedback* do utilizador respectivo sobre o conteúdo assistido. O cliente IPTV pode optar por enviar um *feedback* por defeito (contemplado pela BD) mesmo que o utilizador não seleccione o nível de agrado relativo ao conteúdo que acabou de assistir. Deste modo evita-se que de futuro o conteúdo respectivo seja novamente sugerido a esse mesmo utilizador;

Link List Request Ao contrario de um pedido *New Link Request*, uma mensagem/tarefa *Link List Request* permite obter uma lista de tamanho variável de *links* para conteúdos e não somente um *link* para um conteúdo. Para além disso, é possível especificar critérios de procura, critérios de ordenação e qual a origem do conteúdo em termos de organização interna do servidor, ou seja, se o conteúdo pertence à lista de conteúdos sugeridos, vistos pelo utilizador respectivo ou se, por outro lado, pode ser qualquer conteúdo presente na BD;

Link Feedback Request Dado que o sistema pretende aprender dinamicamente o perfil do utilizador, este deverá especificar sempre que possível um *feedback* relativo ao conteúdo assistido. Esta tarefa actualiza a classificação global do conteúdo assim como insere uma nova entrada no histórico de *feedbacks* do utilizador respectivo, para posterior tratamento pelo módulo *User Profile Learning* (secção 4.4);

Command Request Uma tarefa realizada com recurso a este tipo de mensagem pode ser de índole muito variada, desde um comando de verificação de disponibilidade do servidor (*Ping Request*) até a um pedido de uma configuração específica ou execução

de uma tarefa que não necessite da autenticação de qualquer utilizador no sistema;

Com a excepção das evocações específicas dos métodos respectivos, a execução de tarefas relativas às mensagens recebidas segue sempre a mesma abordagem geral no que respeita à sequência nuclear de sub-tarefas a realizar. A figura 4.1 serve de exemplo demonstrativo da sequência básica de sub-tarefas utilizadas no caso específico de mensagens/tarefas do tipo *User Authentication*. Desde o momento em que uma tarefa é despoletada até ao instante em que é enviada a resposta ao respectivo cliente IPTV, um único *thread* prestador de serviço assegura a integridade, a execução e a monitorização das sub-tarefas necessárias (que também podem ser modeladas por outros *threads*).

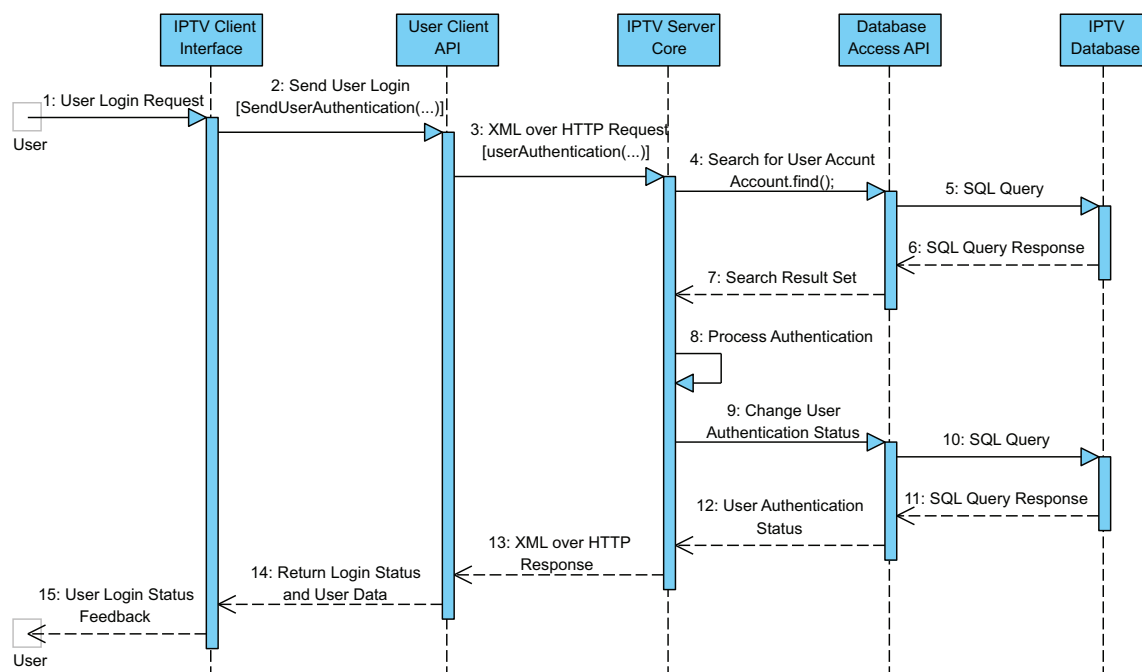


Figura 4.1: Diagrama de sequência relativa ao módulo *IPTV Server Core* no caso de execução de uma tarefa do tipo *User Authentication*.

É de notar que para todas as mensagens implementadas (ver anexo B), foi criado o seu respectivo diagrama de sequência das tarefas a executar no sistema, à semelhança da figura 4.1.

Para além das mensagens/tarefas implementadas (anexo B), o sistema suporta de forma

natural a implementação futura de muitos outros tipos de mensagens com semântica vária. Algumas destas poderão ser:

- Envio de mensagens ou comentários a conteúdos entre utilizadores do sistema;
- Recuperação das credenciais de acesso ao sistema referentes a um utilizador específico;
- Actualização dos dados e perfil de um utilizador;
- Desactivação da conta de um utilizador;
- Adição (por parte de um utilizador) de um novo *link* para um conteúdo;
- Criação de um “grupo virtual de amigos”;
- Adicionar/convidar um utilizador para um “grupo virtual de amigos”;
- Requerer a entrada num “grupo virtual de amigos”;
- Sugestão de um conteúdo (por parte de um utilizador) a um outro utilizador pertencente a um dos seus “grupos virtual de amigos”;
- Permitir a um utilizador ver os conteúdos assistidos anteriormente (em modos não privados) de qualquer elemento do seu “grupo virtual de amigos”;
- Permitir a um utilizador ver os conteúdos que foram sugeridos (em modos não privados) a qualquer elemento do seu “grupo virtual de amigos”;

A criação de novos “grupos virtuais de amigos” é completamente livre. Para isso, baste que o utilizador use a opção disponível no cliente IPTV para criar um novo grupo virtual, sendo que fica automaticamente associado a este como o seu administrador. Deste modo é este utilizador o único que pode alterar as suas definições internas, aceitar um novo elemento no grupo, remover um elemento do grupo ou até eliminar o próprio grupo. Todos os restantes elementos têm iguais privilégios dentro do grupo.

Com a implementação actual, o utilizador poderá visualizar qualquer conteúdo multimédia no sistema, sendo que o *link* de acesso a este poderá ser escondido do cliente. Tendo

em vista futuras alterações, é possível restringir facilmente o acesso dos utilizadores/clientes IPTV a certos conteúdos, introduzindo algumas alterações à BD e a sua respectiva repercussão no código *Java* do servidor. A figura A.1 exemplifica essas possíveis alterações na BD de modo a que um utilizador só tenha acesso a alguns conteúdos com base em pacotes de conteúdos afectados. Esta restrição pode ser muito útil e até necessária no caso de conteúdos e canais pagos ou de acesso restrito a determinados utilizadores.

Neste caso, um utilizador só poderia aceder a conteúdos que estejam inseridos em pelo menos um dos pacotes subscritos por esse mesmo utilizador (conteúdo restrito). Caso um conteúdo não esteja inserido em pelo menos um pacote, então este poderá ser acedido por qualquer utilizador presente no sistema (conteúdo livre).

Este módulo é também responsável pela criação, lançamento e gestão da execução de alguns *triggers* temporais relacionados com a interacção utilizador–servidor. Actualmente esses *triggers* são:

Session Time Out Trigger Efectua o *logout* automático do utilizador quando este permanece um tempo excessivo (configurável) sem efectuar qualquer acção no sistema. Esta acção permite não só uma monitorização mais fidedigna dos utilizadores activos no sistema como também libertar recursos desnecessariamente alocados no mesmo;

Too Many Attempts Unlock Trigger Desbloqueio automático temporizado da conta dos utilizadores que efectuaram excessivas tentativas falhadas de *login* no sistema e ficaram impossibilitados de aceder ao servidor durante um período de tempo (configurável). Esta acção tenta eliminar a possibilidade de descoberta das credenciais dos utilizadores através de ataques por “força bruta” (*brute force*) mas, por outro lado, possibilita a criação de condições para ataques de “negação de serviço” (*denial of service*). Numa fase posterior o bloqueio de utilizadores não será feito somente tendo em conta as suas credenciais (*user name* e *password*) mas também utilizando o endereço IP em conjugação com o endereço físico do dispositivo de acesso.

Para mais informações técnicas sobre o modelo de programação utilizado neste módulo, é favor consulta a documentação *Javadoc* fornecida com o servidor.

4.2.1 Arquitectura Geral do Funcionamento Interno

Esta secção pretende dar uma ideia geral do funcionamento e arquitectura interna do módulo *IPTV Server Core*. Para isso, recorreu-se à apresentação de alguns dos diagramas UML criados na fase de modelação deste módulo funcional e que foram sofrendo algumas pequenas alterações até ao momento em que se deu esta fase do desenvolvimento do servidor por terminada. A figura 4.2 mostra o diagrama UML do encapsulamento interno dos pacotes *Java* criados, bem como algumas das relações de dependência entre estes. Os pacotes que se encontram fora do pacote com fundo cinza não pertencem a este módulo, sendo partilhados com todo o sistema.

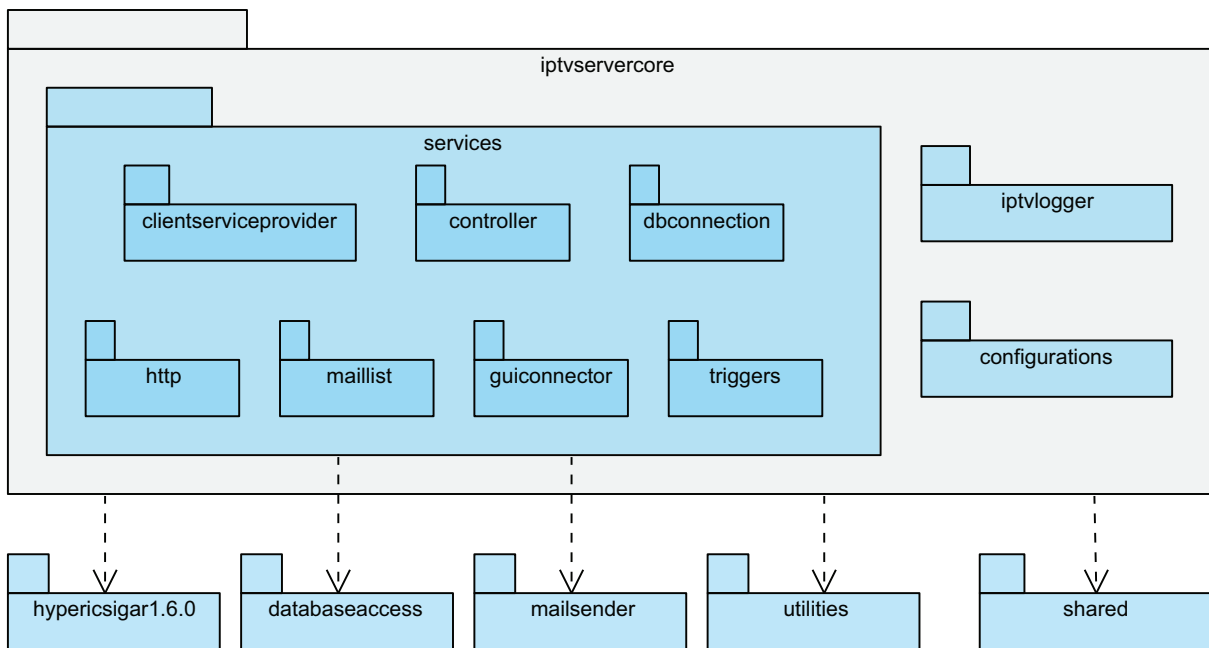


Figura 4.2: Diagrama de pacotes do módulo *IPTV Server Core*.

De modo a permitir um conhecimento mais pormenorizado sobre o funcionamento interno deste módulo em particular, a figura 4.3 mostra o diagrama UML das classes *Java* criadas, representando também algumas das associações e dependências entre estas.

É de notar que nos diagramas apresentados nesta secção não se encontram representados todos os pacotes/classes deste módulo nem todas as dependências entre estes. Somente os

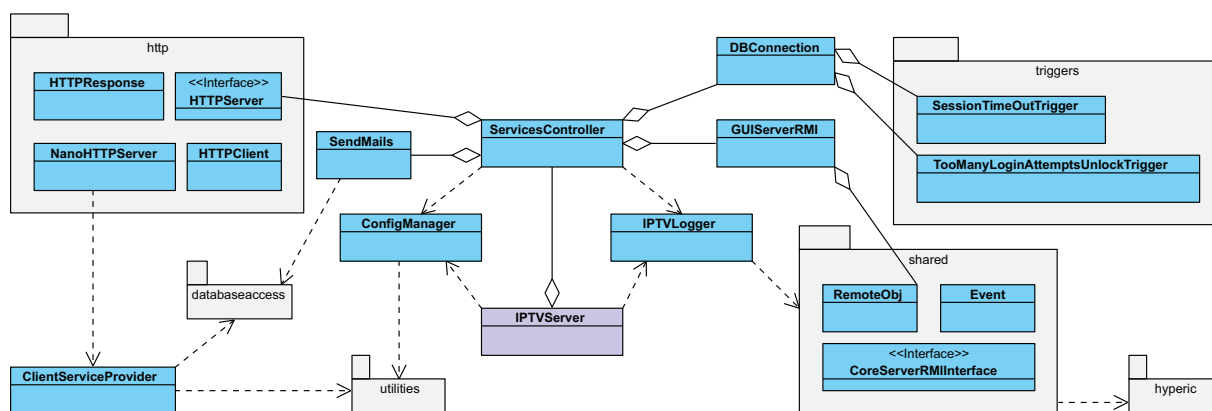


Figura 4.3: Diagrama das classes principais do módulo *IPTV Server Core* e seus relacionamentos.

aspectos mais importantes estão evidenciados.

4.3 Content Link Sources

Neste capítulo será apresentado o módulo *Content Link Source*.

Este módulo estabelece a ligação entre o sistema implementado e os recursos multimédia externos disponíveis. De uma forma genérica, este sub-sistema do servidor efectua não só a procura de novos conteúdos multimédia mas também a gestão dos já existentes na base de dados. É de notar que em nenhuma situação o sistema guarda os conteúdos multimédia propriamente ditos, sendo que a sua persistência não depende do servidor implementado e por isso a utilização do conceito de “link” quando se fala do acesso a um conteúdo multimédia.

Sendo então a metodologia deste servidor IPTV orientada à persistência de todas as informações do conteúdo multimédia inclusive o seu *link*, significando isto que o conteúdo poderá se encontrar em qualquer sistema computacional sobre um qualquer protocolo de transmissão. O acesso correcto aos conteúdos é da exclusiva responsabilidade do cliente IPTV.

Em complemento ao objectivo proposto relativo à utilização de RSS, foi ponderada a utilização de um sistema totalmente automático baseado numa *web crawler* que “varreria” a Internet à procura de conteúdos multimédia, indexando-os com uma ferramenta poderosa como o *Apache Lucene*. Este tipo de procura automática em profundidade torna-se pouco controlável e controlada quando o volume de dados aumenta, introduzindo de forma quase indiscriminada “lixo” na base de dados de conteúdos multimédia.

Dada essa grande desvantagem, optou-se pela utilização da tecnologia RSS por forma que somente os provedores desejados forneçam conteúdos multimédia ao sistema. Cria-se assim à partida não só uma pré-classificação em termos de áreas de interesse dos conteúdos (pois o provedor e/ou o RSS subscrito têm geralmente uma área de interesse associada) mas também um controlo do que o sistema disponibiliza.

Uma outra forma de procura e acesso aos dados relativos aos conteúdos multimédia são as APIs de acesso dos provedores dos serviços. Estas podem ser disponibilizadas não só por certos provedores multimédia aos quais também é possível interagir através de RSS (sendo dada primazia ao acesso via API) mas fundamentalmente por provedores cujo o acesso ao serviço não se encontra disponível de forma natural ou gratuita na Internet e logo não é possível o acesso via RSS, necessitando de processos de AAA.

Assim sendo, o acesso genérico aos dados dos conteúdos multimédia é feito via RSS e o acesso especializado é conseguido através das APIs do respectivo provedor do conteúdo.

Do ponto de vista automático, este módulo é a única entidade responsável pela introdução de novos *links* para conteúdos multimédia. Para além do mecanismo automático, é ainda possível introduzir novos *links* para conteúdos através da adição destes por parte de utilizadores registados no sistema. Assim sendo, um utilizador tem a liberdade para adicionar um novo *link* bem como as informações subjacentes e também políticas básicas de restrição de acesso (*link* público ou privado). Do mesmo modo, é permitido aos utilizadores removerem os *links* que adicionaram previamente, mantendo a integridade da base de dados.

A base de dados prevê a persistência de vários tipos de *links* para conteúdos multimédia no que respeita à sua frequência de actualização. Conteúdos estáticos (que não sofrem

actualizações periódicas), conteúdos com actualizações periódicas ou ainda conteúdos distribuídos em “tempo real”. Para fazer esta distinção, a base de dados possui na tabela *ContentLink* o campo *RefreshRate* que contém o valor -1 caso o conteúdo seja estático, 0 caso seja em *real-time* ou então um valor positivo em milisegundos referente à taxa de actualização.

O suporte dos múltiplos protocolos de comunicação e *streaming* multimédia utilizados na Internet (RTP, RTSP, MMS, etc.) bem como das codificações dos mesmos (AVI, FLV, H.264, MPEG, etc.) são da exclusiva responsabilidade do cliente IPTV.

Este módulo funcional é internamente orientado a uma metodologia de programação concorrente no que respeita à execução das suas tarefas. É por esta razão que possibilita a busca paralela de conteúdos em múltiplos provedores.

Por defeito, o sistema executa simultaneamente somente uma instância deste módulo que centraliza a captura e gestão de conteúdos. Não obstante das vantagens do processamento distribuído, é possível a execução simultânea de múltiplas instâncias deste módulo, por forma a organizar e distribuir em rede a procura e actualização dos recursos.

À semelhança dos restantes módulos funcionais, este também utiliza um ficheiro XML que permite a persistência das suas configurações dinâmicas. Segue-se um exemplo desse mesmo ficheiro de configuração.

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <ContentLinkSourcesSettings xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
3   <serverExternalAddress value="iptvserver.servehttp.com" />
4   <configFilePath value="C:\IPTVFiles\config\" />
5   <xmlFilePath value="C:\IPTVFiles\xml\" />
6   <guiPort value="2326" />
7   <remoteObjUser value="root" />
8   <remoteObjPass value="admin" />
9   <databasePort value="3306" />
10  <databaseName value="iptvdatabase" />
11  <databaseAddress value="localhost" />
12  <databaseUser value="joaoffr" />
13  <databasePass value="jffr3559" />
14 </ContentLinkSourcesSettings>
```

É de notar que as *tags* XML utilizadas no ficheiro de configuração são auto-explicativas no que toca à sua semântica. Qualquer questão relativa às mesmas, é favor consultar a

documentação *Javadoc* deste módulo que acompanha o código do servidor.

De modo a guardar as *feeds* RSS subscritas bem como páginas base que podem conter *links* para *feeds*, foi utilizado um ficheiro com uma sintaxe muito simples. Caso o *link* diga respeito a uma *feed* RSS, é utilizado o elemento XML *rssdirectlink*. Caso contrario, utiliza-se o elemento *baselink*. Ambos podem ser vistos no exemplo seguinte.

```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!--
3   'baselink' - link to a web page that have links to RSS
4   'rssdirectlink' - direct link to link to RSS feed
5   NOTE: use the &amp; instead of just & (XML syntax)
6 -->
7 <rsslinks xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
8   <baselink value="http://movies.yahoo.com/rss" interestarea="movies" description="Yahoo
9     Movies" />
10  <baselink value="http://news.bbc.co.uk/2/hi/help/3223484.stm" interestarea="BBC"
11    description="BBC" />
12  <rssdirectlink value="http://videos.sapo.pt/diarioeconomico/rss"
13    interestarea="economy" description="Diario Economico" />
14 </rsslinks>

```

Para processar *links* indirectos, este módulo possui um mecanismo que procura numa página HTML todos os *links* validos e tenta extrair deles uma *feed* RSS. Caso não consiga, descarta estes.

A biblioteca *Java* que se utiliza para servir de cliente RSS é o *Java ROME v1.0 (Release Mar/12/2009)*.

No que respeita ao *parsing* das *feeds* RSS, é utilizada a mesma biblioteca *Java* usada no *parsing* dos dados em XML: *JDOM XML parser*.

Dado que o protocolo RSS é bastante flexível, para interpretar correctamente todas as *feeds* dos diferentes provedores é necessário proceder a um estudo e análise prévio do conteúdo destas, visto poder variar sensivelmente. Aquando a escrita deste documento, este módulo funcional estava a utilizar os campos padrão do protocolo RSS como fonte de dados para qualquer provedor de conteúdos multimédia.

Relativamente ao acesso aos conteúdos disponibilizados pelo *Youtube*, foi utilizada a API *Java Youtube Data API* disponibilizada pela *Google*. A utilização desta necessitou de um estudo prévio, sendo que todo o código que a compõe encontra-se muito bem documentado.

Actualmente o sistema utiliza as categorias e subcategorias presentes no grafo de áreas de interesse armazenado na BD para procurar vídeos no *Youtube*. Depois de encontrar os conteúdos disponíveis, o módulo *Content Link Sources* tira partido das relações entre vídeos estabelecidas pelo *Youtube* para indexar mais e mais conteúdos, ramificando-se de uma forma exponencial até atingir o equilíbrio. É de notar que o sistema não adiciona múltiplos conteúdos com o mesmo *link*, mesmo que possuam diferentes *metadados* associados. Se assim ocorrer, os dados do conteúdo são actualizados de modo a reflectir o mais recente.

Existe actualmente uma estrutura em árvore de áreas de interesse bastante extensa. Devido à sua dimensão, não é possível mostrar a mesma neste documento. Para a consultar, verificar a directoria “InterestAreaTree” dentro da directoria do código fonte do servidor IPTV.

Alguns serviços relacionados com conteúdos multimédia (neste exemplo, só relacionados com vídeos) que disponibilizam uma API podem ser encontrados na tabela . É de notar que nesta tabela consta uma lista reduzida dos serviços mais conhecidos que disponibilizam API de acesso, sendo que muitos mais estão disponíveis e não foram aqui mencionados. A tabela foi baseada na informação disponibilizada em <http://www.programmableweb.com/apis/directory/1?apicat=Video>, à data da escrita desta Dissertação de Mestrado.

Com é possível verificar pela análise da tabela 4.1, existe um grande número de serviços *online* que efectuem tarefas desde o fornecimento de conteúdos até à codificação e transcodificação *on demand*, passando pela disponibilização de legendas ou informações sobre os conteúdos multimédia. Deste modo é possível a utilização de alguns destes para criar um conjunto de funcionalidades extra no o sistema IPTV.

4.3.1 Arquitectura Geral do Funcionamento Interno

Esta secção pretende dar uma ideia geral do funcionamento e arquitectura interna do módulo *Content Link Sources*. Para isso, recorreu-se à apresentação de alguns dos diagramas UML criados na fase de modelação deste sub-sistema e que foram sofrendo

Nome do serviço	Descrição
<i>12seconds.tv</i>	<i>12-second videoblogging</i> ;
<i>Ankoder</i>	Serviço de transcodificação de vídeo;
<i>AOL Video</i>	Procura de vídeos AOL;
<i>Apideo</i>	Serviço de <i>streaming</i> de vídeo;
<i>Blinkx</i>	Serviço de procura de vídeos;
<i>Blip.tv</i>	Serviço de <i>upload</i> e partilha de vídeos;
<i>Brightcove</i>	Serviço de publicação de vídeos;
<i>Encoding.com</i>	Serviço de codificação de vídeos;
<i>ffwd</i>	Construção de canais, partilha e procura de vídeos;
<i>Filmcrave</i>	Base de dados de revisões de filmes e <i>trailers</i> ;
<i>Floobs</i>	Serviço de <i>Streaming</i> de vídeos;
<i>Hey Spread</i>	Serviço de distribuição e promoção de vídeos;
<i>HeyWatch</i>	Serviço de codificação de vídeo;
<i>Howcast</i>	Serviço de procura de vídeos “como fazer” determinadas tarefas;
<i>Internet Video Archive</i>	<i>Trailers</i> de filmes/jogos e vídeos musicais;
<i>Joost</i>	Serviço de televisão <i>online</i> ;
<i>Justin.tv</i>	Serviço de <i>Live video</i> ;
<i>Kyte</i>	Serviço de <i>streaming</i> interactivo de televisão;
<i>Mefedia</i>	<i>Video feeds</i> e comunidades de vídeos;
<i>Mydeo Media Manager</i>	Serviço de armazenamento e <i>streaming</i> de vídeo;
<i>Netflix</i>	Serviço de aluguer digital de vídeos;
<i>NY Times Movies</i>	Arquivo de revisão de vídeos e filmes do NYT;
<i>Ooyala</i>	Plataforma compreensiva de vídeo;
<i>Opensubtitles</i>	Serviço de legendas para filmes;
<i>Revver</i>	Serviços de partilha de vídeos;
<i>Sesame Vault</i>	Serviço de partilha de vídeo;
<i>StreamAPI</i>	Serviço de <i>streaming</i> de vídeo;
<i>Stupeflix</i>	Serviço de criação de vídeos;
<i>Ustream.TV</i>	Serviço de <i>streaming</i> de vídeos;
<i>Veoh</i>	Televisão sobre a Internet, vídeos e plataforma social;
<i>Viddler</i>	Serviço de vídeos <i>online</i> ;
<i>VideoBloom</i>	Soluções de distribuição de vídeo <i>end-to-end</i> e publicidade;
<i>VideoSurf</i>	Serviço de procura de vídeos;
<i>Viewdle</i>	Serviço de reconhecimento facial em vídeo;
<i>Vimeo</i>	Serviço de partilha de vídeos;
<i>Vodpod</i>	Serviço de partilha e procura de vídeos;
<i>Vuclip Video</i>	Partilha e procura de vídeos para dispositivos móveis;
<i>vzaar</i>	Serviço de armazenamento de vídeos;
<i>Yahoo Video Search</i>	Serviço de procura de vídeos;
<i>YouTube</i>	Serviço de partilha e procura de vídeos;

Tabela 4.1: Alguns serviços multimédia que disponibilizam API de acesso.

algumas pequenas alterações até ao momento em que se deu esta fase do desenvolvimento do servidor por terminada. A figura 4.4 mostra o diagrama UML do encapsulamento interno dos pacotes *Java* criados, bem como algumas das relações de dependência entre estes. Os pacotes que se encontram fora do pacote com fundo cinza não pertencem a este módulo, sendo partilhados com todo o sistema.

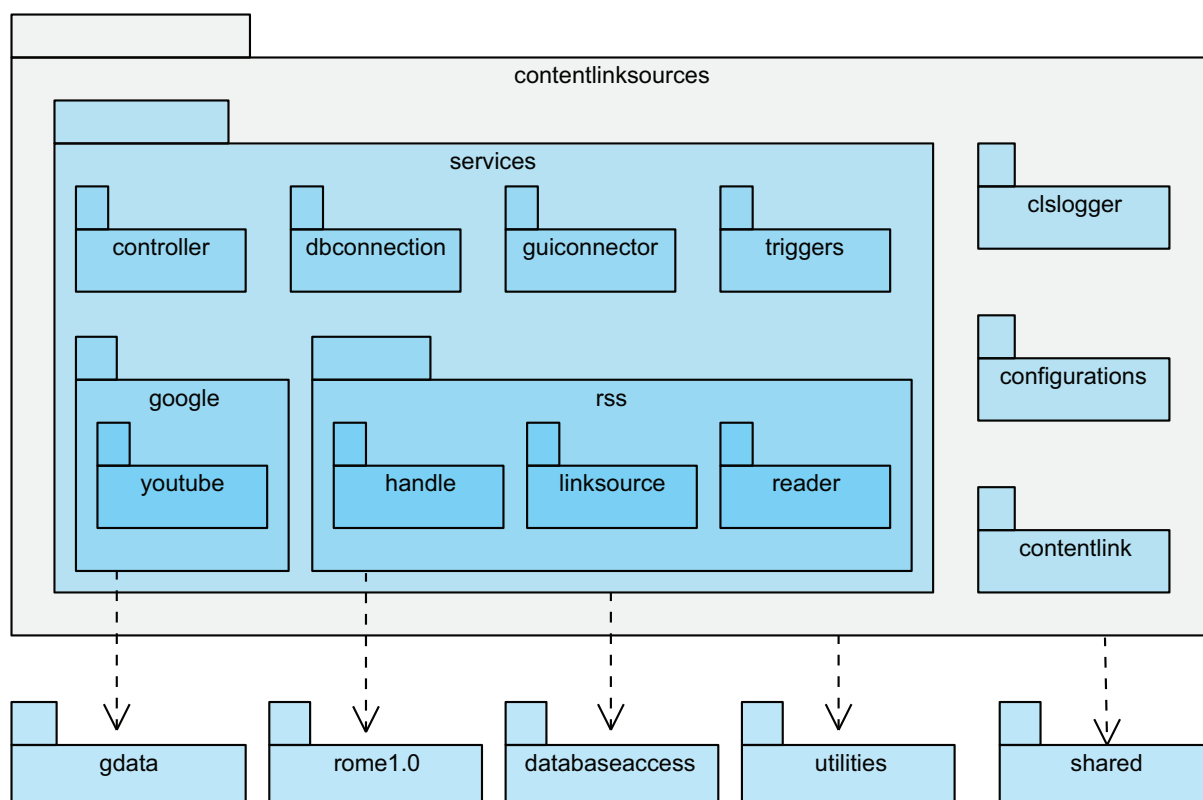


Figura 4.4: Diagrama de pacotes do módulo *Content Link Sources*.

De modo a permitir um conhecimento mais pormenorizado sobre o funcionamento interno deste módulo em particular, a figura 4.5 mostra o diagrama UML das classes *Java* implementadas, representando também algumas das associações e dependências entre elas.

É de notar que nos diagramas apresentados nesta secção não se encontram representados todos os pacotes/classes deste módulo nem todas as dependências entre estes. Somente os aspectos mais importantes estão evidenciados.

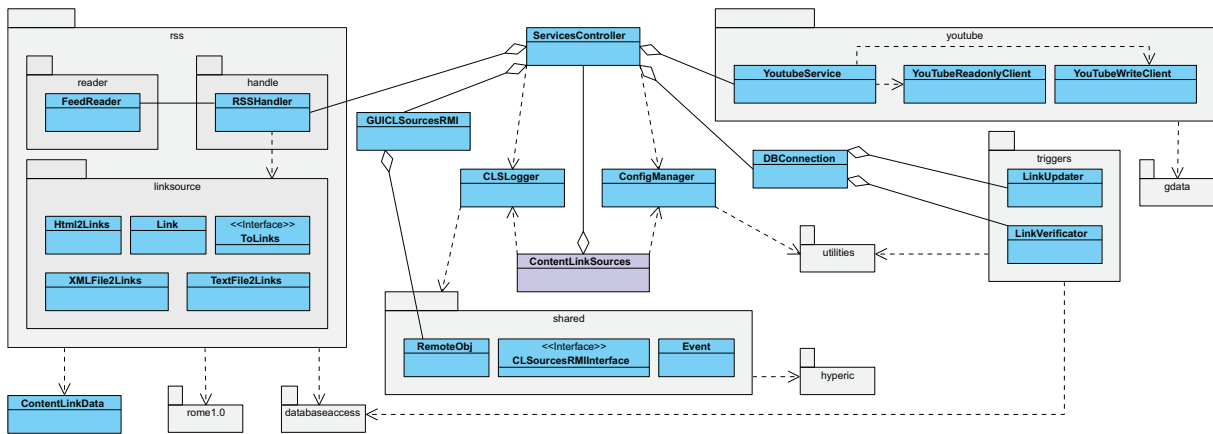


Figura 4.5: Diagrama das classes principais do módulo *Content Link Sources* e seus relacionamentos.

4.4 *User Profile Learning*

Neste capítulo será apresentado o módulo *User Profile Learning*.

Faz parte deste módulo duas componentes distintas: Sugestão de conteúdos (*Content Suggestion*) e aprendizagem dinâmica do perfil do utilizador (*Profile Learning*). Estas duas componentes encontram-se dentro do mesmo módulo devido a uma forte relação que as une, em que uma depende intrinsecamente da outra.

A componente de aprendizagem do perfil do utilizador é, por si só, suficiente para a elaboração de uma ou mais Teses de Mestrado, dada a sua possível complexidade e diversidade de abordagens e implementações possíveis. É proposto aqui que o sistema possua a habilidade de capturar, analisar e correlacionar dados provenientes tanto do perfil estático do utilizador como da sua experiência de utilização no sistema (perfil dinâmico) ou dados correntes como a sua posição geográfica, condições meteorológicas, temperatura, velocidade, equipamento de visualização em uso, etc.

No âmbito desta Dissertação de Mestrado foi implementado um sistema de aprendizagem de perfil do utilizador muito básico que se limita a guardar todos os dados relevantes relativos ao utilizador para posterior tratamento. Não é actualmente feito nenhum correlacionamento entre estes dados nem existe qualquer algoritmo de aprendizagem implemen-

tado, mas todas as condições para a sua implementação estão criadas tanto no código *Java* como na base de dados.

No que respeita à componente de sugestão automática, esta possui uma dependência inata com a componente de aprendizagem dinâmica. Isto acontece devido ao facto de que para sugerir conteúdos é necessário possuir conhecimento *a priori* sobre o utilizador. Por outro lado, para possuir conhecimento *a priori* sobre o perfil do utilizador é fundamental a obtenção do *feedback* relativo aos conteúdos vistos (sejam eles sugeridos ou não) bem como a informação corrente e passada (histórico) associada.

Assim sendo, à semelhança do que foi implementado na componente de aprendizagem do perfil do utilizador, o sistema de sugestão automática é actualmente muito simples, sugerindo sempre os conteúdos mais recentes e com melhor classificação, descuidando actualmente o seu tema (semântica) bem como o actual perfil do utilizador (aprendizagem).

É de notar que mesmo que não estejam implementados os algoritmos de sugestão e aprendizagem dinâmica, todos os dados relativos a *feedbacks* enviados para o servidor são guardados de uma forma estruturada e organizada na base de dados para futura utilização.

A sugestão de conteúdos a um determinado utilizador pode também ter origem em outro utilizador do sistema desde que pertençam ao mesmo “grupo virtual de amigos”. Do mesmo modo, qualquer utilizador pode sugerir qualquer conteúdo a um utilizador em particular desde que pertença a um dos seus “grupos virtuais de amigos”, bem como a todos os elementos de um grupo no qual esteja inserido.

A futura aprendizagem e classificação dos utilizadores deverá seguir a abordagem proposta no capítulo 3.5, sob pena do trabalho desenvolvido actualmente não poder ser compatível com uma nova abordagem.

A ferramenta de controlo e gestão do servidor IPTV (*IPTV Server GUI*) prevê e suporta a interacção com o módulo neste capítulo descrito tal como foi projectado. Actualmente o controlo sobre o módulo *User Profile Learning* é reduzido pois as funcionalidades do mesmo também são reduzidas.

A execução de múltiplas instâncias deste módulo foi prevista aquando a sua projecção, de forma a agilizar o processo de aprendizagem, podendo utilizar-se diferentes algoritmos

de aprendizagem complementares entre si, desde que garantido que convirjam para uma solução.

4.5 *IPTV Database*

Neste capítulo será apresentado o módulo *IPTV Database* no que respeita ao seu modelo, funcionalidades, tecnologias utilizadas, bem como todas as suas características relevantes.

Este módulo é o conjunto formado pela base de dados propriamente dita (o SGBD) e a API de acesso à mesma, sendo esta última utilizada em cada módulo que necessite de aceder aos dados na base de dados.

4.5.1 Tecnologias Utilizadas

Este módulo reveste-se de uma importância extrema, não só por ser o módulo agregador de todos os outros módulos funcionais mas também por conter a informação vital ao funcionamento de todo o sistema.

Para tornar a implementação do servidor IPTV independente do SGBD em uso, foi utilizado a tecnologia *Hibernate* que trabalha em perfeita harmonia com o *Java* (ambos da Sun Microsystems). Este torna a comunicação com qualquer SGBD relacional suportado mais fácil, robusta e intuitiva, fazendo uso do mapeamento objecto-relacional. A sua política de comunicação com os diferentes SGBD no mercado baseia-se em *drivers*, sendo que basta trocar o *driver* para mudar o tipo do SGBD alvo da comunicação.

Com esta tecnologia, cada entidade dá origem a uma classe *Java*, a qual possui os atributos referentes aos campos da entidade e os métodos que sejam necessários para, através de evocações a métodos específicos do *Hibernate*, executar as tarefas pretendidas na BD. É efectuada também validação e controlo dos dados a inserir, fazendo uso inúmeras vezes de expressões regulares que definem uma sintaxe válida para estes. Cria-se assim uma abstracção sobre a BD que permite desconhecer os pormenores da sua implementação e do

seu *Structured Query Language* (SQL) específico, deixando essa tarefa a cargo do *Hibernate*, tornando assim o sistema mais flexível e robusto.

Como SGBD (ou DBMS) escolhido, optou-se pelo *MySQL* (agora também pertencente à empresa Sun Microsystems), não só por ser gratuito mas também por ser estável, simples e robusto o suficiente para esta primeira fase de implementação e testes.

No que toca à gestão e monitorização da base de dados, é utilizada a ferramenta *phpMyAdmin* (<http://www.phpmyadmin.net/>), que possibilita o controlo e gestão de bases de dados cujo o SGBD é o *MySQL* de forma remota e completamente integrada, utilizando um qualquer *browser* de Internet. A figura 4.6 mostra o *layout* desta aplicação no que respeita à gestão da base de dados do servidor.

Table	Action	Records ¹	Type	Collation	Size	Overhead
account		0	MyISAM	utf8_general_ci	1.0 KiB	-
accountstatus		0	MyISAM	utf8_general_ci	1.0 KiB	-
author		0	MyISAM	utf8_general_ci	1.0 KiB	-
authorinterestsprofile		0	MyISAM	utf8_general_ci	1.0 KiB	-
contentfeedback		0	MyISAM	utf8_general_ci	1.0 KiB	-
contentformat		0	MyISAM	utf8_general_ci	1.0 KiB	-
contentinterestsprofile		0	MyISAM	utf8_general_ci	1.0 KiB	-
contentlink		0	MyISAM	utf8_general_ci	1.0 KiB	-
contenttype		0	MyISAM	utf8_general_ci	1.0 KiB	-
friendsgroup		0	MyISAM	utf8_general_ci	1.0 KiB	-
friendsgroupinterestsprofile		0	MyISAM	utf8_general_ci	1.0 KiB	-
gender		0	MyISAM	utf8_general_ci	1.0 KiB	-
interestarea		0	MyISAM	utf8_general_ci	1.0 KiB	-
language		0	MyISAM	utf8_general_ci	1.0 KiB	-
profession		0	MyISAM	utf8_general_ci	1.0 KiB	-
provider		0	MyISAM	utf8_general_ci	1.0 KiB	-
providerinterestsprofile		0	MyISAM	utf8_general_ci	1.0 KiB	-
relationshipstatus		0	MyISAM	utf8_general_ci	1.0 KiB	-
sessionstatus		0	MyISAM	utf8_general_ci	1.0 KiB	-
suggestedcontent		0	MyISAM	utf8_general_ci	1.0 KiB	-
systemvalue		0	MyISAM	utf8_general_ci	1.0 KiB	-
usergroup		0	MyISAM	utf8_general_ci	1.0 KiB	-
userinfriendsgroup		0	MyISAM	utf8_general_ci	1.0 KiB	-
userinterestsprofile		0	MyISAM	utf8_general_ci	1.0 KiB	-
userlanguage		0	MyISAM	utf8_general_ci	1.0 KiB	-
usermessage		0	MyISAM	utf8_general_ci	1.0 KiB	-
userprofile		0	MyISAM	utf8_general_ci	1.0 KiB	-
usersessiondata		0	MyISAM	utf8_general_ci	1.0 KiB	-
visitedcontent		0	MyISAM	utf8_general_ci	1.0 KiB	-
visitedcontentfeedback		0	MyISAM	utf8_general_ci	1.0 KiB	-
watchingmode		0	MyISAM	utf8_general_ci	1.0 KiB	-
31 table(s)	Sum	0	MyISAM	utf8_general_ci	31.0 KiB	0 B

Figura 4.6: Exemplo do *layout* da aplicação *phpMyAdmin*.

Neste momento, o servidor IPTV projectado é meramente académico, mas é facilmente compreensível que no caso do seu uso mais real este seria sujeito a uma enorme carga de pedidos por parte dos seus utilizadores/clientes. Sendo a base de dados um ponto crítico do

sistema, o SGBD *MySQL* não é considerado ideal para grandes sistemas e sistemas críticos. Depois de uma análise aos SGBDs disponíveis, aquele que é considerado ideal para este sistema é o SGBD da Oracle (empresa que, só por curiosidade, comprou recentemente a empresa Sun Microsystems), pelas suas características de extrema robustez, flexibilidade e fiabilidade. O mesmo não foi utilizado no servidor logo de início devido, essencialmente, a não ser gratuito, a ser mais complexo e exigir um muito maior poder computacional para ser executado eficientemente. Os testes ao servidor utilizando *Oracle* e a sua real implementação é deixada para uma segunda fase do desenvolvimento do servidor, não constituindo para já a utilização do *MySQL* uma limitação.

4.5.2 Modelo de Base de Dados

O modelo de base de dados implementado foi desenhado com recurso ao programa *PowerDesigner 15* da Sybase.

Todo o processo de *design* da base de dados iniciou-se na construção manual do modelo conceptual inicial pretendido. Depois de criada uma versão estável e coerente deste, o modelo físico foi gerado pelo programa de acordo com as regras para normalização dos modelos de base de dados. A partir deste momento, o modelo físico da base de dados passou a ser usado como sendo o modelo base para todas as alterações subsequentes, tendo sido gerados os modelos lógicos e o modelo orientado a objectos a partir deste, bem como re-gerado o modelo conceptual. Todos estes modelos serviram para refinar iterativamente o desenho da base de dados, até chegar ao modelo de dados actual.

A evolução do modelo de base de dados sucedeu sempre de forma iterativa, em que as alterações necessárias eram adicionadas ao modelo físico e depois gerados todos os outros para confirmação dos resultados pretendidos. A figura 4.7 esquematiza este processo.

Os diagramas conceptual, lógico e físico da base de dados implementada encontram-se em anexo (ver anexo A).

A BD actualmente implementada não contempla a definição de índices nas suas tabelas, exceptuando os definidos por defeito para os campos referentes a chaves primárias e chaves

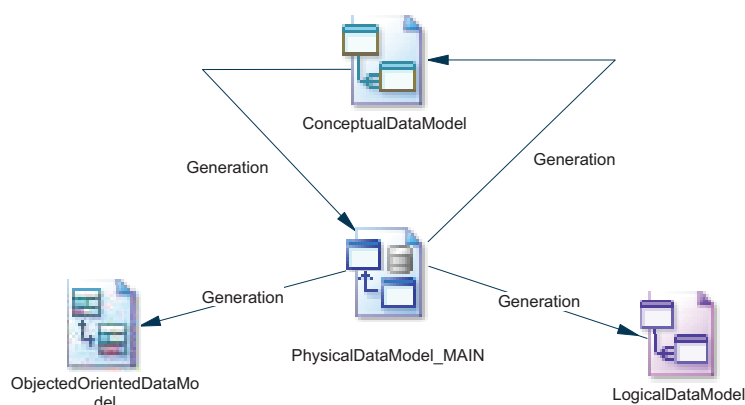


Figura 4.7: Diagrama do processo iterativo de geração dos modelos de base de dados.

estrangeiras. Isto deveu-se ao facto do sistema ainda não possuir carga suficiente de modo a ser significativo a determinação de índices, bem como ainda não ser possível determinar com rigor os campos que poderão necessitar de indexação devido a serem alvo de um elevado número de *querys* de pesquisa e ordenação.

No que respeita à existência de campos nulos, genericamente estes não são permitidos. Os únicos campos em que valores nulos podem existir são:

SessionEndDate Tabela *UserSessionData* - Contém o instante temporal que marca o final da sessão do utilizador no sistema. É nulo enquanto o utilizador estiver *online*;

InsertedByUserID Tabela *ContentLink* - Contém o identificador numérico do utilizador que inseriu um *link* específico. É nulo no caso do *link* ter sido inserido pelo sistema (módulo *Content Link Sources*);

SuggestedByUserID Tabela *SuggestedContent* - Contém o identificador numérico do utilizador que sugeriu um *link* específico a outro utilizador. É nulo no caso do *link* ter sido sugerido pelo sistema.

No que respeita a valores definidos por defeito, estes foram especificados no código *Java* referente à API de acesso à BD. À semelhança destes, a verificação de gamas e tipos admissíveis foi também programada dentro da API.

É de notar que, exceptuando na passagem de um *link* da tabela de conteúdos sugeridos (*SuggestedContent*) para a tabela de conteúdos visitados (*VisitedContent*), nunca são eliminadas entradas das tabelas do sistema. Todas as entidades susceptíveis de eliminação de registos possuem um campo *active* que determina se a entrada respectiva está activa (valor a *true*) ou inactiva (valor a *false*). Isto permite não só manter sempre todos os dados na BD como também manter a consistência da base de dados não necessitando de efectuar eliminação de registos em cascata nem com recurso a outros algoritmos, aumentando assim o desempenho do sistema. Por outro lado, esta estratégia pode penalizar ligeiramente o desempenho da BD devido a esta possuir uma dimensão superior à mínima necessária, não sendo para já este pormenor considerado um problema.

Assim sendo, pode perfeitamente ocorrer que, por exemplo, um *link* para um conteúdo exista na BD mas não esteja activo e logo não usável devido ao respectivo campo *active* da tabela *ContentLink* possuir o valor *false*.

A base de dados foi projectada tendo em vista um grande conjunto de funcionalidades, cujas algumas delas, à data da elaboração desta Dissertação, o servidor ainda não explora. De entre estas é possível destacar o mecanismo de persistência e envio de mensagens entre utilizadores referentes a um determinado conteúdo multimédia, o mecanismo de criação e gestão de grupos virtuais de utilizadores bem como a sua classificação, a classificação de autores e provedores de conteúdos, entre outras.

Deste modo, para explorar as capacidades e funcionalidades da base de dados actualmente não aproveitadas pelo servidor, basta criar os modelos das mensagens XML necessárias caso a funcionalidade seja passível de ser requisitada por um cliente IPTV, bem como programar a lógica funcional referente às tarefas a executar no respectivo módulo funcional.

Foi também alvo de reflexão a utilização de uma base de dados distribuída e o uso de replicação de dados. Estes não foram considerados devido ao carácter experimental da actual implementação do sistema. Ambas as abordagens podem ser implementadas sem qualquer alteração à arquitectura utilizada, tornando o sistema mais rápido e menos susceptível a falhas.

4.5.3 As Entidades da Base de Dados

A base de dados implementada conta actualmente com 31 tabelas (entidades) e respectivas relações entre as mesmas, definindo assim o modelo relacional de persistência de dados implementado no servidor IPTV. Cada tabela possui um papel bem definido no sistema assim como cada campo comporta dados específicos que geralmente não podem ser nulos, excepto em algumas situações pontuais que favorecem o funcionamento do sistema.

A definição e desenho da base de dados teve como uma das premissas a não criação de dados redundantes ou desnecessários à partida, encontrando-se esta normalizada de acordo com as regras e boas práticas de modelação de sistemas de base de dados. A semântica das relações estabelecidas entre as entidades possibilitam tirar um elevado partido dessas mesmas relações e dependências entre os dados de forma a melhor correlacioná-los.

No desenho da base de dados, sempre que é necessário utilizar uma relação do tipo *muitos-para-muitos* é imprescindível criar uma tabela de relação entre as partes. Normalmente, essa tabela contém apenas como campos a chave primária composta que são as chaves estrangeiras da relação. No entanto, no desenho da base de dados implementada, essas tabelas de relação são sempre aproveitadas para conter mais alguns dados que pertençam à relação, definindo assim não só a relação mas também parâmetros da relação.

A lista das tabelas implementadas e respectiva função no sistema encontra-se definida na tabela 4.2. A especificação das entidades, seus campos e diagramas das relações estabelecidas encontra-se em anexo (anexo A).

Nome da Entidade	#	Descrição
<i>Account</i>	8	Entidade utilizada para guardar os dados privados específicos relativos à conta do utilizador;
<i>AccountStatus</i>	3	Entidade que lista os estados possíveis em que se pode encontrar uma conta do utilizador;
<i>continua na página seguinte</i>		

<i>continuação da página anterior</i>		
Nome da Entidade	#	Descrição
<i>Author</i>	5	Entidade que detém os dados de todos os autores de conteúdos multimédia no sistema;
<i>AuthorInterestsProfile</i>	5	Entidade que persiste os dados relativos ao perfil de interesses actual dos autores de conteúdos multimédia;
<i>ContentFeedback</i>	4	Entidade que lista os tipos de <i>feedback</i> passíveis de ser submetidos pelo utilizador relativos à visualização de conteúdos multimédia;
<i>ContentFormat</i>	4	Entidade que lista os formatos de conteúdos multimédia suportados (exemplos: AVI, MPEG, MP3, etc.);
<i>ContentInterestsProfile</i>	5	Entidade que guarda os dados relativos ao perfil actual de cada conteúdo multimédia no sistema;
<i>ContentLink</i>	16	Entidade que modela e persiste todos os dados específicos relativos a um conteúdo presente no servidor;
<i>ContentType</i>	3	Entidade que lista os tipos de transmissão suportados para os conteúdos (exemplos: <i>real time streaming</i> , <i>non realtime streaming</i> , etc.);
<i>FriendsGroup</i>	7	Entidade que modela e persiste os vários grupos sociais de utilizadores que o sistema possui;
<i>FriendsGroupInterestsProfile</i>	5	Entidade que guarda os dados relativos ao perfil geral de interesses dinâmicos de cada grupo social de utilizadores;
<i>Gender</i>	3	Entidade que lista todos os sexos passíveis de serem considerados pelos utilizadores;
<i>continua na página seguinte</i>		

<i>continuação da página anterior</i>		
Nome da Entidade	#	Descrição
<i>InterestArea</i>	5	Entidade que guarda todas as áreas de interesse e suas evoluções/composições suportadas pelo sistema. O identificador de cada área de interesse reflecte as relações de generalização que deram origem à mesma;
<i>Language</i>	3	Entidade que lista todos os idiomas suportados pelo sistema;
<i>Profession</i>	3	Entidade que lista todas as profissões passíveis de serem consideradas pelos utilizadores;
<i>Provider</i>	5	Entidade que detém os dados de todos os provedores de acesso aos conteúdos multimédia presentes no sistema;
<i>ProviderInterestsProfile</i>	5	Entidade que persiste os dados relativos ao perfil actual baseado nas áreas de interesses dos provedores de conteúdos multimédia;
<i>RelationshipStatus</i>	3	Entidade que lista todos os estados conjugais passíveis de serem considerados pelos utilizadores;
<i>SessionStatus</i>	3	Entidade que lista os estados possíveis de uma sessão iniciada por um utilizador;
<i>SuggestedContent</i>	5	Entidade que persiste a referência para os conteúdos multimédia sugeridos (pelo sistema ou por outro utilizador) ao utilizador em questão;
<i>continua na página seguinte</i>		

<i>continuação da página anterior</i>		
Nome da Entidade	#	Descrição
<i>SystemValue</i>	5	Entidade que guarda determinados parâmetros e valores dinâmicos do sistema;
<i>UserGroup</i>	5	Entidade que lista os diversos grupos de utilizadores em termos do seu tipo de conta (exemplo: administrador);
<i>UserInFriendsGroup</i>	5	Entidade que relaciona um utilizador com o seu grupo social de utilizadores. Esta entidade é a responsável pela persistência da lista de utilizadores pertencentes a um determinado grupo de “amigos”;
<i>UserInterestsProfile</i>	6	Entidade que persiste os dados relativos ao perfil actual com respeito às áreas de interesses de determinado utilizador registado no sistema;
<i>UserLanguage</i>	5	Entidade que relaciona cada utilizador com os múltiplos idiomas que este domina segundo determinado peso medido de 0.0 (0%) a 1.0 (100%);
<i>UserMessage</i>	8	Entidade que permite a persistência e troca de mensagens entre utilizadores do sistema, estando estas sempre relacionadas com determinado conteúdo multimédia;
<i>UserProfile</i>	10	Entidade que modela e persiste os dados específicos relativos ao perfil geral do utilizador;
<i>UserSessionData</i>	15	Entidade que guarda toda a informação considerada relevante de cada sessão iniciada pelos utilizadores registados no servidor;
<i>continua na página seguinte</i>		

continuação da página anterior		
Nome da Entidade	#	Descrição
<i>VisitedContent</i>	7	Entidade que persiste a referência para os conteúdos multimédia assistidos pelo utilizador em questão;
<i>VisitedContentFeedback</i>	6	Entidade que persiste todos os <i>feedbacks</i> recebidos pelo sistema enviados por parte de cada utilizador utilizadores relativos a determinado conteúdo multimédia assistido;
<i>WatchingMode</i>	5	Entidade que lista todos os modos de visualização (exemplos: sozinho, com a família, etc.) passíveis de serem seleccionados pelos utilizadores;

Tabela 4.2: Entidades implementadas pela base de dados do sistema e respectiva descrição.

Este módulo, para além de comportar as entidades e métodos de introdução, acesso e manipulação dos dados guardados, possui também outras funcionalidades embebidas. De entre elas, é possível destacar o sistema de inicialização (subsecção 4.5.5) que coloca os dados iniciais na base de dados, alguns eles estáticos outros dinâmicos. Faz também parte integrante da API definida a especificação do tamanho dos tipos de dados personalizados (subsecção 4.5.4), *triggers* (subsecção 4.5.6), configurações e controle de erros/excepções (subsecção 4.5.7).

A API de acesso à base de dados implementada permite a utilização dos mecanismos de persistência definidos recorrendo à invocação de métodos de alto nível, permitindo descuidar os pormenores de implementação.

Mais uma vez, para tornar o servidor totalmente independente do SGBD, não foram implementadas rotinas de tratamento de dados e eventos (*triggers*, *store procedures*, etc.) em SQL. Optou-se então pela criação de um mecanismo em *Java* baseado na implementação de *threads* específicos para a monitorização e execução deste tipo de tarefas.

4.5.4 Tipos de Dados Definidos

Durante a modelação iterativa da base de dados, através do *software* mencionado na subsecção 4.5.2, foram considerados tipos de dados padronizados na especificação dos campos das diversas entidades. Estes tipos de dados foram definidos com base nos tipos básicos existentes. Deste modo é possível atribuir tipos de dados “virtuais” de uma forma indirecta e assim permitir propagar de forma natural possíveis alterações aos mesmos, conferindo uma maior robustez à modelação.

A tabela 4.3 enumera e descreve cada tipo de dados personalizado. A sua utilização pode ser verificada nos diagramas presentes no anexo A.

A API em *Java* construída para possibilitar o mapeamento objecto-relacional permite não só manipular a base de dados da forma desejada, mas também permite criar esta e inicializa-la (subsecção 4.5.5), sendo que a sua criação obedece aos diagramas especificados no anexo A.

4.5.5 Sistema de Inicialização Automática

A criação desta funcionalidade surgiu para colmatar o trabalho despendido sistematicamente na introdução de dados cada vez que se reformulava e reinstalava a base de dados.

Deste modo, é possível colocar os dados iniciais de uma forma automática, o que reduz o tempo de instalação do sistema.

A tabela 4.4 enumera as tabelas que são alvo de inicialização e indica quantos registos, actualmente, são introduzidos durante essa mesma inicialização.

É de notar que depois da inicialização, a base de dados fica preparada para uma utilização normal. Inicialmente esta não possui qualquer conteúdo multimédia (que são introduzidos pelo módulo *Content Link Sources* ou pelos próprios utilizadores). A inicialização não prevê também a sugestão de qualquer conteúdo a qualquer utilizador ou a classificação em termos de áreas de interesses de qualquer entidade (que é da exclusiva responsabilidade do sistema).

O utilizador introduzido pelo mecanismo de inicialização é, por defeito, do tipo admin-

Nome	Tipo de Dados	Tamanho	Precisão
<i>DateTime</i>	timestamp	–	–
<i>Decimal</i>	double	8	4
<i>Description</i>	varchar	1024	–
<i>GPSPos</i>	double	14	10
<i>LongName</i>	varchar	64	–
<i>NumberOf</i>	int	–	–
<i>NumericID</i>	int	–	–
<i>Password</i>	varchar	32	–
<i>Path</i>	varchar	255	–
<i>PathID</i>	varchar	64	–
<i>Rate</i>	double	8	6
<i>ShortName</i>	varchar	24	–

Tabela 4.3: Lista dos domínios da base de dados.

Nome da entidade	Número de registos
<i>Account</i>	1
<i>AccountStatus</i>	4
<i>Author</i>	3
<i>ContentFeedback</i>	4
<i>ContentFormat</i>	46
<i>ContentType</i>	5
<i>Gender</i>	3
<i>InterestArea</i>	9
<i>Language</i>	124
<i>Profession</i>	6
<i>Provider</i>	10
<i>RelationshipStatus</i>	5
<i>SessionStatus</i>	4
<i>SystemValue</i>	5
<i>UserGroup</i>	6
<i>UserLanguage</i>	4
<i>UserProfile</i>	1
<i>WatchingMode</i>	7

Tabela 4.4: Lista das entidade com inicialização.

istrador e permite o acesso directo a todo o sistema mediante a introdução das correctas credenciais.

4.5.6 *Triggers*

O sistema actualmente prevê alguns comportamentos automáticos no que respeita à execução de algumas tarefas. Exemplos destas são o *Logout* automático devido ao utilizador ter estado tempo demasiado sem actividade no sistema ou o bloqueio da conta do mesmo, durante um período temporal, por demasiadas tentativas de *login* sem sucesso.

Actualmente, somente as duas tarefas anteriormente mencionadas estão implementadas, sendo que o mecanismo de programação adoptado possibilita a fácil e rápida adição de novas tarefas automáticas de índole temporal.

Os *triggers* implementados são totalmente configuráveis. Caso seja necessário desactivar algum deles ou adicionar outro, foi implementado um gestor destas tarefas sobre a forma de um *thread* gestor que coordena *threads* que são os *triggers* propriamente ditos.

4.5.7 Configurações e Excepções

No que diz respeito às configurações da API, estas são maioritariamente efectuadas pelos diversos módulos que a utiliza aquando a sua inicialização.

No caso do sistema de inicialização da base de dados, dado que este usa a API de acesso à BD como se de um módulo funcional distinto se tratasse, existem parâmetros por defeito que estão escritos no próprio código. Estes não são dinâmicos devido ao facto da inicialização da BD ser feita muito esporadicamente por pessoas específicas e devido a motivos bem definidos, não exigindo uma configuração dinâmica.

Quando acontece algum erro ou excepção interna na API, são disparadas excepções. Estas podem ser as existentes por defeito na linguagem *Java* ou outras que foram definidas especificamente para certas situações anómalas. Para saber mais informações sobre este tópico, por favor consultar a documentação *Javadoc* fornecida com o código do servidor.

4.5.8 Arquitectura Geral do Funcionamento Interno

Esta secção pretende dar uma ideia geral do funcionamento e arquitectura interna da API de acesso à base de dados. Para isso, recorreu-se à apresentação de alguns dos diagramas UML criados na fase de modelação desta API. A figura 4.8 mostra o diagrama UML do encapsulamento interno dos pacotes *Java* criados, bem como algumas das relações de dependência entre estes. Os pacotes que se encontram fora do pacote com fundo cinza não pertencem directamente ao código da API, sendo partilhados com todo o sistema.

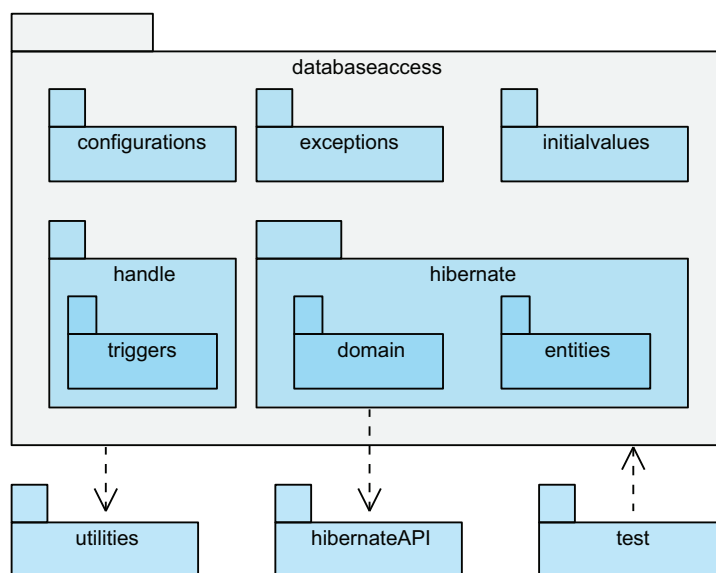


Figura 4.8: Diagrama de pacotes da API de acesso à base de dados.

De modo a permitir um conhecimento mais pormenorizado sobre o funcionamento interno desta API, a figura 4.9 mostra o diagrama UML das classes *Java* criadas, representando também algumas das associações e dependências entre elas.

4.6 Gestão e Monitorização: *IPTV Server GUI*

Neste capítulo será apresentado o módulo *IPTV Server GUI*.

Esta ferramenta foi desenvolvida no intuito de facilitar a gestão e manutenção do servidor de forma remota. Esta encontra-se ainda numa fase de desenvolvimento muito precoce

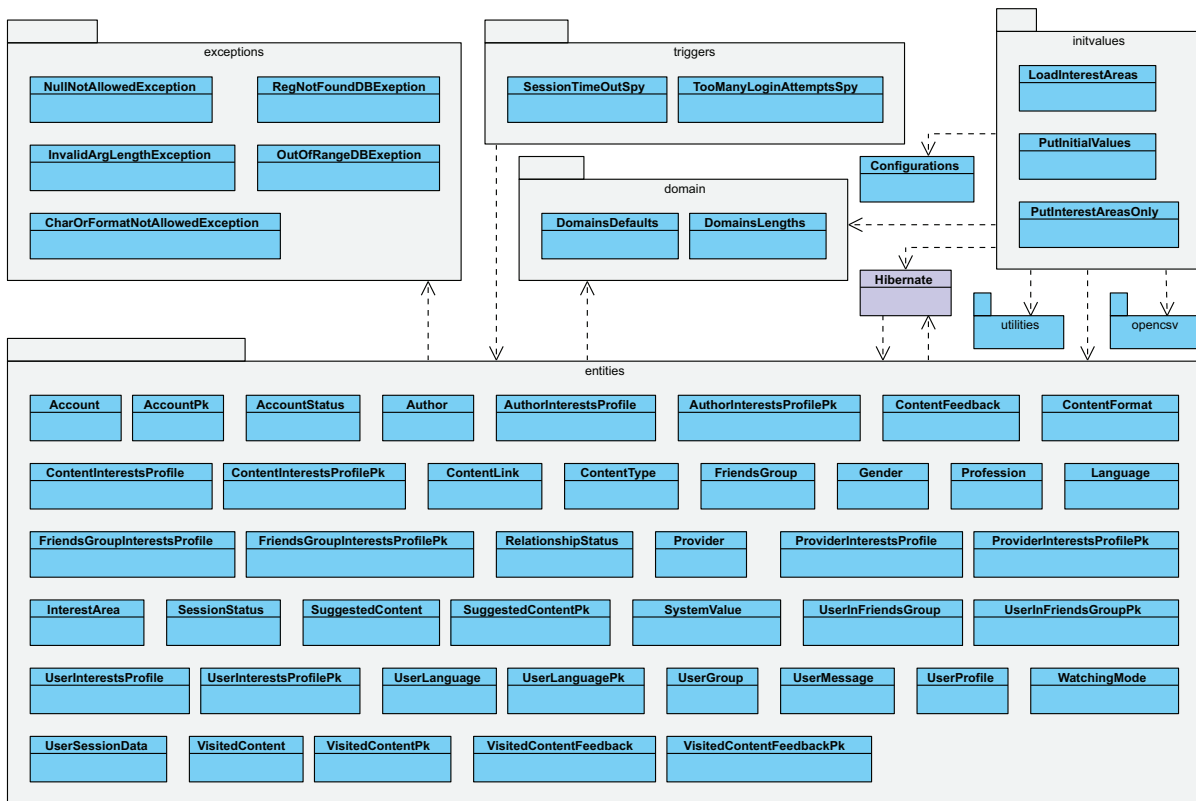


Figura 4.9: Diagrama das classes principais da API de acesso à BD.

na medida em que, essencialmente, somente o módulo *IPTV Server Core* é controlado de uma forma total. O controlo disponibilizado sobre o módulo *Content Link Sources* ainda não é tão amplo e profundo como desejado, sendo que possui actualmente as operações básicas de gestão e monitorização.

Por outro lado, relativamente aos módulos *IPTV Database* e *User Profile Learning*, a ferramenta prevê o seu controlo e gestão mas ainda não possui muitas funcionalidades específicas para isso. No caso do módulo *IPTV Database*, actualmente não é essencial criar no módulo *IPTV Server GUI* opções de gestão e monitorização específicas devido a ser utilizada a ferramenta *phpMyAdmin* que é suficiente para a gestão e monitorização da BD. No que respeita ao módulo *User Profile Learning*, as suas funcionalidades são, actualmente, muito reduzidas e logo o controlo e monitorização não faz sentido.

De qualquer forma, foram criadas todas as condições para a integração de mais fun-

cionalidades no controlo e monitorização de todos os módulos, tendo sido utilizado um estilo de programação organizado, bem documentado e facilmente alterável e estendível a outras funcionalidades.

Um dos principais objectivos no desenvolvimento desta ferramenta baseou-se na gestão robusta do sistema com simplicidade nas operações. Na figura 4.10 é possível ver a janela principal da ferramenta. Esta permite não só o início/fim da ligação a um módulo funcional específico do sistema mas também a configuração de todos os parâmetros relativos à ferramenta de gestão (figura 4.11). É de notar que todos os botões e campos passíveis de serem editados possuem um *tool tip text* com ajuda sobre o mesmo. No que respeita à configuração de caminhos para directorias, estes devem sempre possuir uma barra no seu final (“\” no MS Windows, “/” no Linux).

Para uma maior acessibilidade, sempre que o Sistema Operativo (SO) o permitir, é colocado um *icon* na barra de sistema, no qual é possível efectuar as acções normais que este tipo de funcionalidade disponibiliza. De entre elas pode-se destacar a funcionalidade de mostrar/trazer a janela para primeiro plano (duplo *click*) ou mostrar o menu rápido (*click* direito) com as operações mais importantes. Na figura 4.12 é mostrado o menu rápido da aplicação.

É também de destacar que as operações mais importantes e frequentes possuem atalhos de teclado com o intuito de agilizar a interacção do utilizador com a ferramenta.

Como será de esperar, é possível ter-se um número virtualmente ilimitado de instâncias da ferramenta a serem executadas simultaneamente na mesma máquina ou em máquinas diferentes, ambas as instâncias ligadas (ou não) a qualquer um dos módulos funcionais do servidor IPTV.

Essencialmente, a operação desta ferramenta foca-se em possibilitar controlo e monitorização remotos dos diversos módulos implementados, desde que estes estejam em funcionamento e a ferramenta esteja ligada a eles.

Em relação à monitorização, é possível tanto monitorizar os recursos físicos (o *hardware* de suporte) como as acções lógicas do sistema (o *software* implementado). É assim possível ver em tempo real a carga do *Central Processing Unit* (CPU), a quantidade de memória

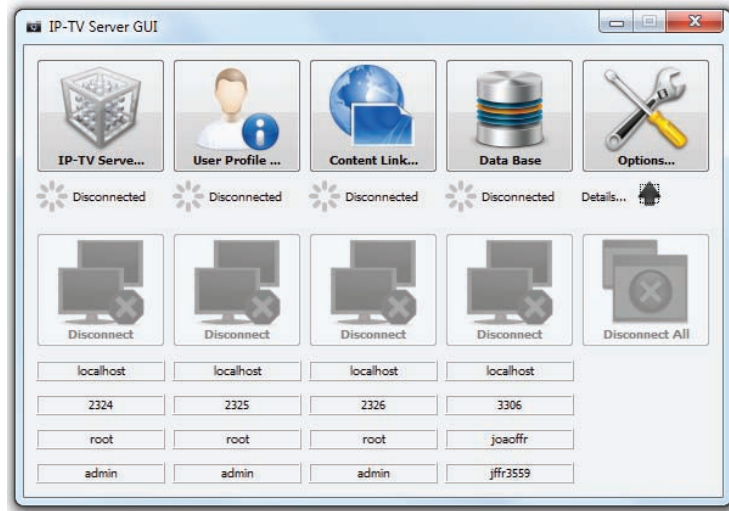


Figura 4.10: Janela principal da ferramenta *IP-TV Server GUI*.

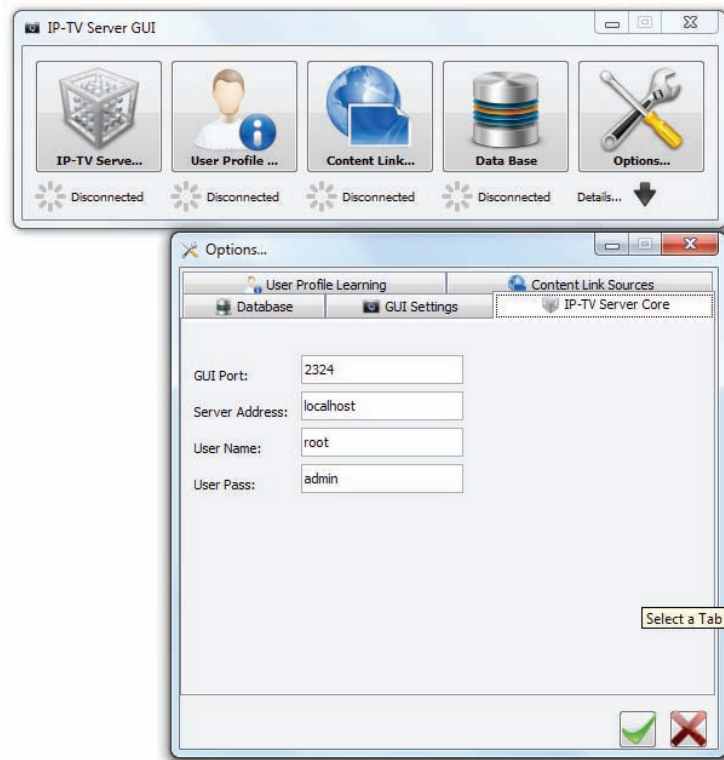


Figura 4.11: Janela das opções de configuração da ferramenta *IP-TV Server GUI*.

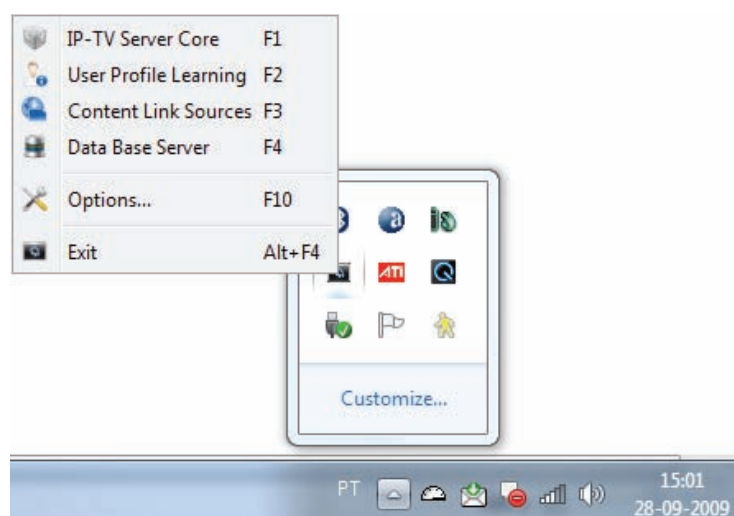


Figura 4.12: Menu rápido da ferramenta *IPTV Server GUI* na bandeja de sistema.

ocupada e disponível, detalhes da interface de rede, entre outros.

Por outro lado, é possível monitorizar em tempo real e de forma assíncrona os eventos que cada módulo despoleta, mostrando os mesmo numa janela reservada para o efeito. Estes eventos podem ser de diversa ordem, desde a recepção de uma mensagem específica até a um alerta ou erro despoletado pelo módulo.

Do ponto de vista do controlo exercido, genericamente é possível iniciar, parar e reiniciar cada módulo, bem como alterar qualquer parâmetro dinâmico presente no ficheiro XML específico de cada módulo. É de notar que a aplicação de novos parâmetros dinâmicos, regra geral, exige que o módulo seja reiniciado. Caso o *restart* do módulo não seja exigido, é aconselhável.

Todas estas operações e funcionalidades estão representadas graficamente na figura 4.13.

Quando se fala em reiniciar um módulo, isto não implica reiniciar todo o servidor. Isto deve-se ao carácter distribuído do sistema, em que o "todo" é constituído por cada parte de forma autónoma.

À semelhança dos módulos funcionais, esta ferramenta também utiliza um ficheiro XML para possibilitar a persistência das configurações dinâmicas próprias. Segue-se um exemplo desse mesmo ficheiro de configuração.

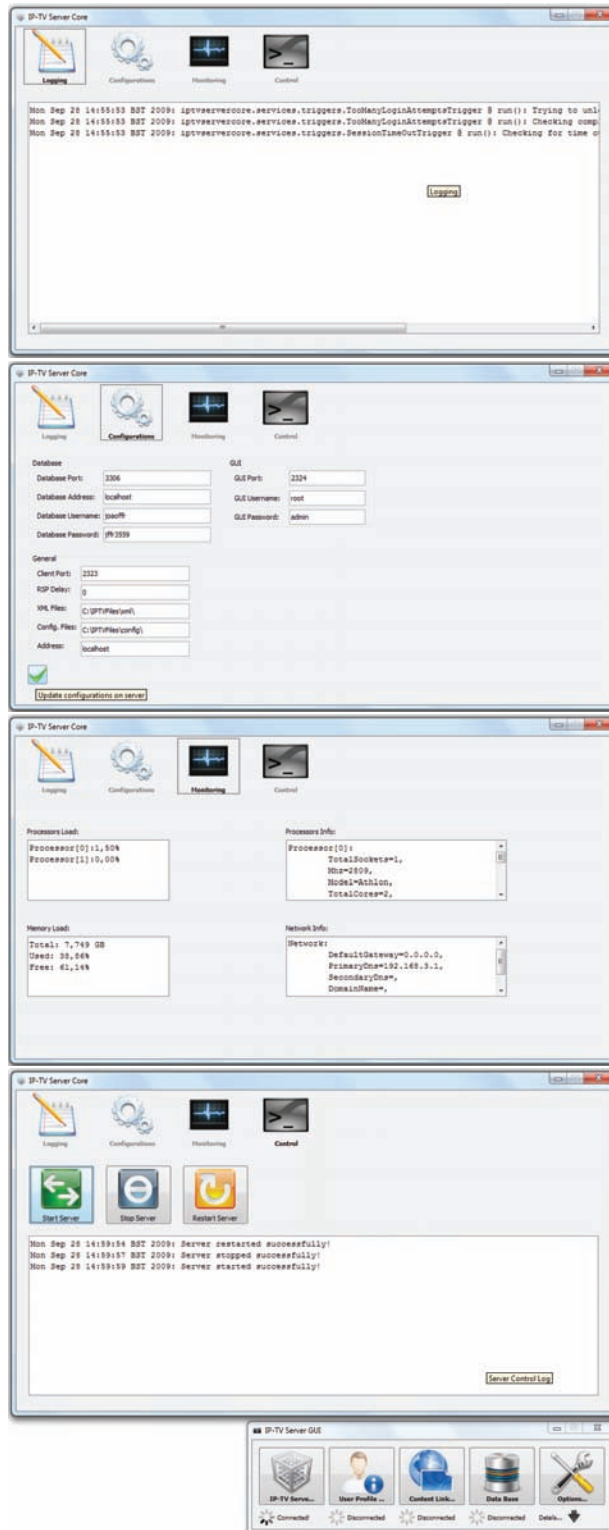


Figura 4.13: Janelas da ferramenta *IPTV Server GUI* relativas à gestão e monitorização do módulo *IPTV Server Core*.

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <GUISettings xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
3   <configFilePath value="C:\IPTVFiles\config\" />
4   <pathToImg value="C:\IPTVFiles\img\" />
5   <coreserverGuiPort value="2324" />
6   <coreserverAddress value="localhost" />
7   <coreserverUser value="root" />
8   <coreserverPass value="admin" />
9   <uplearningGuiPort value="2325" />
10  <uplearningAddress value="localhost" />
11  <uplearningUser value="root" />
12  <uplearningrPass value="admin" />
13  <clinksourcesGuiPort value="2326" />
14  <clinksourcesAddress value="localhost" />
15  <clinksourcesUser value="root" />
16  <clinksourcesPass value="admin" />
17  <databasePort value="3306" />
18  <databaseAddress value="localhost" />
19  <databaseUser value="joaoffr" />
20  <databasePass value="jffr3559" />
21 </GUISettings>
```

É de notar que as *tags* XML utilizadas no ficheiro de configuração são auto-explicativas no que respeita à sua semântica. Qualquer questão relativa às mesmas, é favor consultar a documentação *Javadoc* deste módulo.

Algumas funcionalidades ainda não implementadas e cuja a introdução está prevista são:

- Manipulação de todas as tabelas da base de dados (criação, edição e eliminação de entradas nas tabelas) de acordo com as dependências entre elas, utilizando um sistema de completção automática de campos baseado nas relações entre as tabelas;
- Visualização das dependências entre as tabelas bem como de estatísticas relativas à BD;
- Apresentação de gráficos em “tempo real” relativos a várias métricas do sistema (número de utilizadores autenticados, número de mensagens trocadas por unidade de tempo, atraso médio da resposta a cada tipo de mensagem, etc.);
- Visualização e manipulação de todo o ficheiro XML de configuração de cada módulo;

- Comunicação com os vários módulos do sistema utilizando túneis ou outros mecanismos de forma a contornar as *firewalls*;

Devido a esta ferramenta utilizar uma política de acesso baseada em RMI, foram definidas restrições de segurança e acesso num ficheiro *RMI policy*. Este encontra-se na directoria do código *Java* do servidor, acessível pelo caminho “/IPTVFiles/config/rmi.policy”.

É de notar que devido ao sistema funcionar sobre portos TCP não padrão, o acesso aos módulos via *IPTV Server GUI* pode ser comprometido devido às *firewalls* dos *routers* da rede e do próprio terminal de acesso. Para facilitar o acesso, basta configurar a ferramenta de gestão e respectivos módulos funcionais para utilizarem portos TCP padrão.

4.6.1 Arquitectura Geral do Funcionamento Interno

Esta secção pretende dar uma ideia geral do funcionamento e arquitectura interna da ferramenta de gestão *IPTV Server GUI*. Para isso, recorreu-se à apresentação de alguns dos diagramas UML criados na fase de modelação desta ferramenta e que foram sofrendo algumas pequenas alterações até ao momento em que se deu esta fase do desenvolvimento do sistema por terminada. A figura 4.14 mostra o diagrama UML do encapsulamento interno dos pacotes *Java* criados, bem como algumas das relações de dependência entre estes. Os pacotes que se encontram fora do pacote com fundo cinza não pertencem a este módulo, sendo partilhados com todo o sistema.

De modo a permitir um conhecimento mais pormenorizado sobre o funcionamento interno desta ferramenta, a figura 4.15 mostra o diagrama UML relativo às classes *Java* implementadas, representando também algumas das associações e dependências entre estas.

É de notar que nos diagramas apresentados nesta secção não se encontram representados todos os pacotes/classes relativos à ferramenta de gestão desenvolvida, nem todas as dependências entre os mesmos. Somente os aspectos mais importantes estão evidenciados.

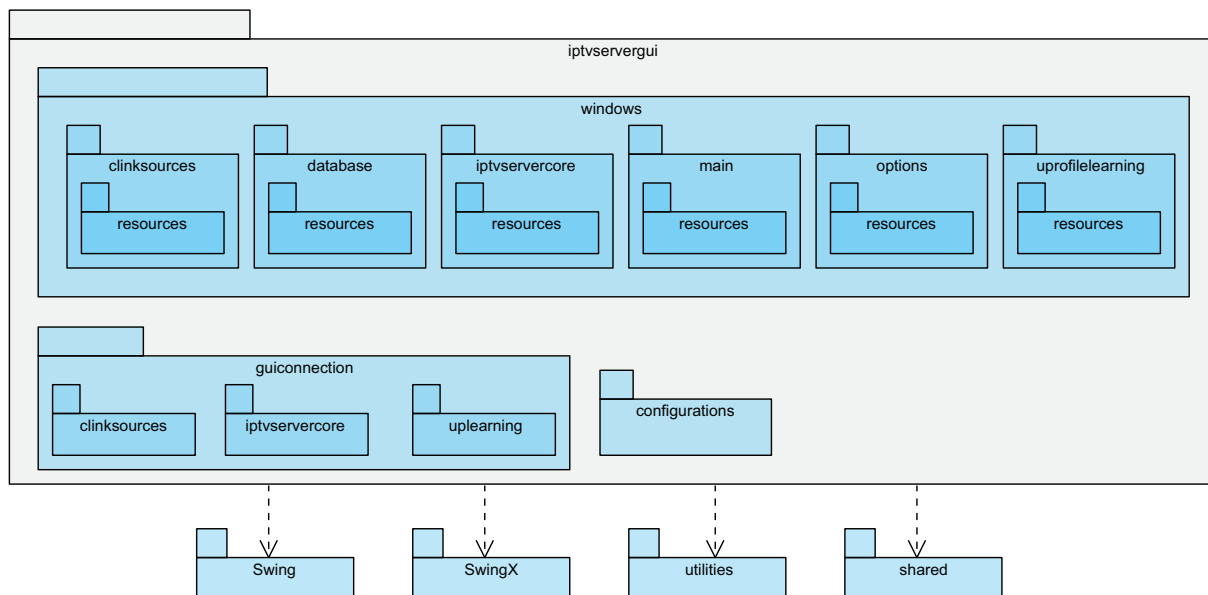


Figura 4.14: Diagrama de pacotes do módulo *IPTV Server GUI*.

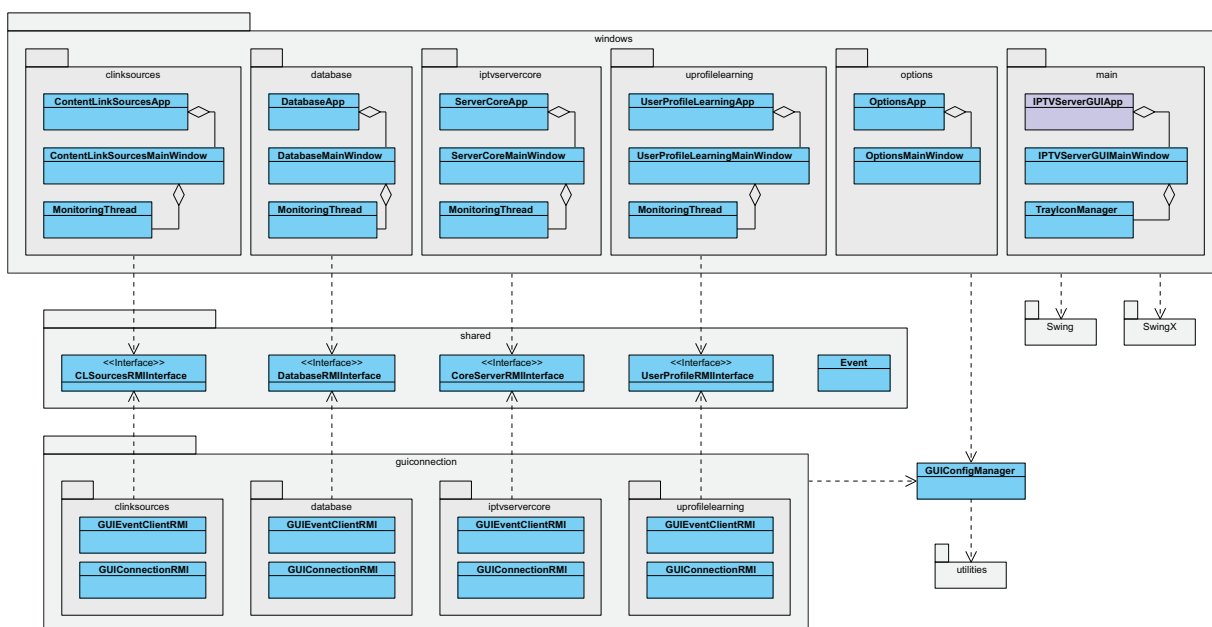


Figura 4.15: Diagrama das classes principais do módulo *IPTV Server GUI* e seus relacionamentos.

Capítulo 5

API em *Java* para Cliente IPTV

5.1 Introdução

Neste capítulo é apresentada a API construída para comunicação com o servidor IPTV a partir de um cliente genérico cujo núcleo de comunicação seja desenvolvido em *Java*. São apresentadas também alguns sugestões funcionais e tecnológicas para o desenvolvimento do cliente.

5.2 Sugestões para o Desenvolvimento do Cliente IPTV

Sugestões de linguagem/sistemas multi-plataforma para o desenvolvimento de um cliente IPTV independente do sistema operativo:

- *Adobe Flash* e/ou *Flex*
- *JavaFX*
- *Zk*
- *HTML + Javascript(jQuery) + PHP*
- *WTVML*

Deverá ser construído um cliente para cada classe de dispositivo: STB, Televisor IPTV, PC e dispositivo móvel. Cada cliente deverá, dentro da sua classe, adaptar-se o mais possível às características específicas do equipamento no qual se encontra a funcionar, tornando assim a sua utilização muito escalável.

Actualmente as consolas de jogos possuem muito poder gráfico e de processamento. Para além disso, normalmente estas possuem também interfaces de rede *wired* e/ou *wireless*. Deste modo, estes dispositivos poderão ser excelentes para funcionarem como STB de acesso ao sistema IPTV a partir de aparelhos de televisão convencionais. Algumas das consolas existentes no mercado que poderiam suportar de forma fácil uma aplicação deste tipo seriam, por exemplo, a *Sony PlayStation 3* ou a *Microsoft Xbox 360*.

Sendo que o acesso ao serviço IPTV poderá necessitar de pagamento quando incluídos conteúdos pagos, o cliente IPTV poderá ser distribuído gratuitamente de modo a incentivar o público alvo a experimentarem o sistema de forma completamente gratuita. Deste modo, deverão estar acessíveis as versões para cada tipo de dispositivo suportado.

Ao longo desta Dissertação de Mestrado foram referidos alguns serviços e funcionalidades que o sistema IPTV poderia suportar. Uma grande parte deles são de suporte inerente ao cliente IPTV, como é o caso de serviços de *Messaging*, acesso a mapas interactivos, acesso a redes sociais, clientes de *e-mail*, entre outros. Assim sendo, cabe ao cliente IPTV implementar este tipo de funcionalidades, se for o caso disso.

O cliente deve permitir também a procura genérica de conteúdos por critérios múltiplos. Actualmente o servidor suporta procura por:

- Área de interesse;
- Nome do conteúdo;
- Autores do conteúdo;
- Provedor do conteúdo;
- Mais vistos;

- Mais votados;
- Mais populares;
- Data do conteúdo;
- Tipo de conteúdo;
- Etc.

Para fomentar a interação e também melhor o próprio aspecto visual do cliente, é de privilegiar a colocação de *thumbnails* relativos aos conteúdos (parâmetro que o servidor prevê). Caso o utilizador aceda ao cliente mas não esteja autenticado, os conteúdos evidenciados através dos *thumbnails* poderão ser os “mais vistos”, “mais votados”, “mais populares”, entre outros. Caso o utilizador esteja autenticado no sistema, os conteúdos salientados poderão ser os sugeridos ou relacionados com conteúdos vistos anteriormente, conteúdos vistos recentemente pelos elementos dos seus “grupos virtuais de amigos”, os sugeridos a outros utilizadores, etc.

Deve ser usado, sempre que possível, as potencialidades do servidor IPTV no que respeita à personalização de conteúdos. Não deverão ser enviadas mensagens desnecessárias para o servidor, sob pena de exercer uma carga excessiva e, deste modo, o mesmo demorar mais tempo a responder às solicitações dos diversos clientes.

5.3 A API

A necessidade de construção de uma API de acesso ao servidor surgiu devido à exigência de execução de testes em *batch* ao servidor implementado. Deste modo, foi criada uma API genérica em *Java* que poderá ser usada por um futuro cliente IPTV deste sistema.

A figura 3.6 enquadra a utilização desta API num cliente IPTV genérico, com núcleo de comunicação em *Java*.

É de salientar que o módulo *IPTV Server Core* possui mecanismos de verificação sintática e semântica de alguns dos dados recebidos por intermédio das mensagens XML.

Deste modo, a fim de evitar a geração desnecessária de erros relacionados com a introdução de dados pelo utilizador (por exemplo, tipos de dados incorrectos, endereços *web* ou *e-mail* com caracteres inválidos/sintaxe não permitida, entre outros), o cliente IPTV deverá efectuar a verificação dos dados inseridos antes destes serem enviados para o servidor IPTV, em consonância com a sua semântica.

De modo a melhor projectar o sistema, foram criados *a priori* múltiplos diagramas UML que descrevem os aspectos mais importantes deste sobre diferentes pontos de vista, uns com mais pormenor técnico, outros mais conceptuais, abrangentes ou gerais. A modelação desta API não constituiu uma excepção, sendo que também foi alvo de uma modelação prévia.

Encontra-se assim representado na figura 5.1 o diagrama UML do encapsulamento interno dos pacotes criados, bem como algumas das relações de dependência entre estes. Os pacotes que se encontram fora do pacote com fundo cinza não pertencem directamente ao código da API.

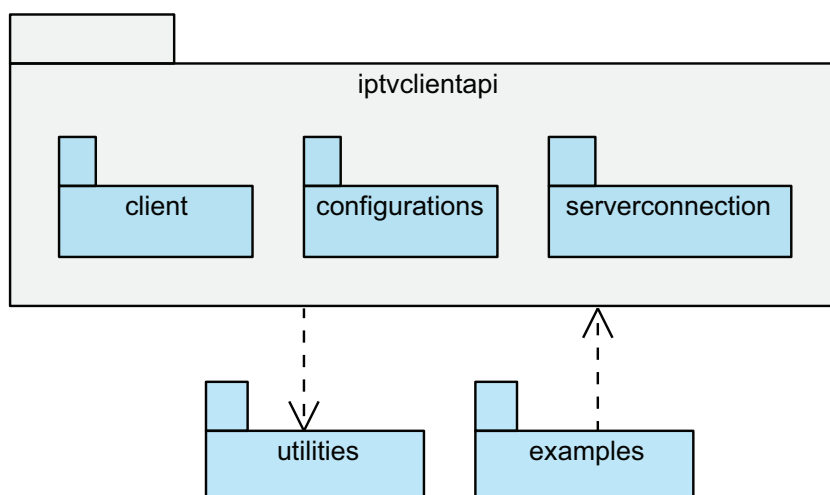


Figura 5.1: Diagrama de pacotes da API de comunicação cliente–servidor.

De modo a permitir um conhecimento mais pormenorizado sobre o funcionamento interno desta API, a figura 5.2 representa o diagrama UML das classes *Java* criadas, representando também algumas das associações e dependências entre elas.

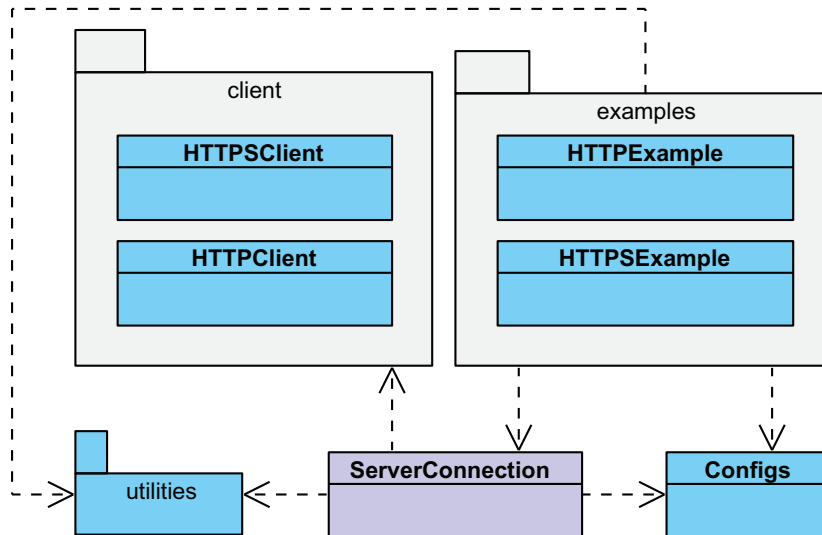


Figura 5.2: Diagrama das classes principais da API de comunicação cliente–servidor.

É de notar que este documento não possui a listagem dos valores introduzidos na base de dados pelo sistema de inicialização da mesma (secção 4.5.5), devido ao facto destes serem voláteis e poderem ser facilmente alterados/adicionados. De qualquer forma, está disponível a consulta da listagem do conteúdo da BD com recurso à ferramenta *phpMyAdmin*, que permite facilmente esclarecer esta questão.

Para mais detalhes sobre a API implementada ou esclarecimento de alguma dúvida, por favor consultar a documentação *Javadoc* fornecida com a mesma, que se encontra na directoria da API junto do código fonte do servidor IPTV.

Capítulo 6

Requisitos e Modelos de Instalação do Servidor

6.1 Introdução

Neste capítulo serão apresentados tanto os requisitos de *software* como de *hardware* necessários para o bom funcionamento do sistema implementado. Para além disso, serão também mencionados alguns possíveis cenários de instalação utilizando diferentes distribuições dos módulos funcionais, suas vantagens e desvantagens.

6.2 Requisitos

No que toca aos requisitos de *hardware*, o servidor exige uma arquitectura que seja suportada pela máquina virtual de *Java* e que possua pelo menos uma interface de rede com acesso à Internet. Para além disso, o sistema computacional deverá possuir pelo menos 700MB livres de espaço em disco e 2GB de RAM.

Em termos de *software* instalado, as exigências já são mais e constam da seguinte lista:

- Todo o código do servidor. É de notar que os binários (*.jar) se encontram dentro da pasta “dist” de cada módulo e devem ser executados com o comando “java -jar <jar file name>”;

- *Java JDK 1.6.0.14* ou superior instalado e funcional;
- *MySQL 5.1.33* ou superior instalado e funcional;
- *PHP 5.2.9.2* ou superior (opcional);
- *PHP MyAdmin 3.1.3.1* ou superior (opcional);
- Pasta “IPTVFiles” colocada na raiz do disco ou então configurar nos módulos para a detectar em outra directoria;
- Portos necessários à comunicação abertos na *firewall* do sistema computacional;
- Ligação à rede local e à Internet;
- Nome DNS atribuído (opcional);

6.3 Instalação

A instalação do servidor pode ser efectuada segundo 3 modelos gerais distintos:

- Instalação Centralizada (figura 6.1);
- Instalação Distribuída sem Replicação de Módulos (figura 6.2);
- Instalação Distribuída com Replicação de Módulos.

A figura 6.1 ilustra a instalação dos servidor IPTV de forma centralizada enquanto que a figura 6.2 ilustra a instalação do mesmo de forma distribuída sem replicação de qualquer módulo.

A instalação do servidor tendo em conta uma arquitectura distribuída com replicação de um ou mais módulos é em tudo semelhante a uma instalação distribuída. As únicas diferenças a ter em consideração é que cada módulo pode ser instanciado múltiplas vezes na mesma ou em diferentes máquinas ligadas em rede. É somente necessário ter em atenção que cada módulo deve ser configurado de forma a funcionar em um diferente porto TCP.

Para efectuar a configuração de cada módulo, basta recorrer à edição do ficheiro XML de configuração de cada um. A existência deste ficheiro de configuração permite a alteração dos parâmetros de instalação de cada módulo do servidor de forma muito simples e rápida, sem necessidade de edição do código fonte do mesmo. É de notar que os parâmetros presentes no ficheiro de configuração só serão aplicados depois do sistema ser reiniciado.

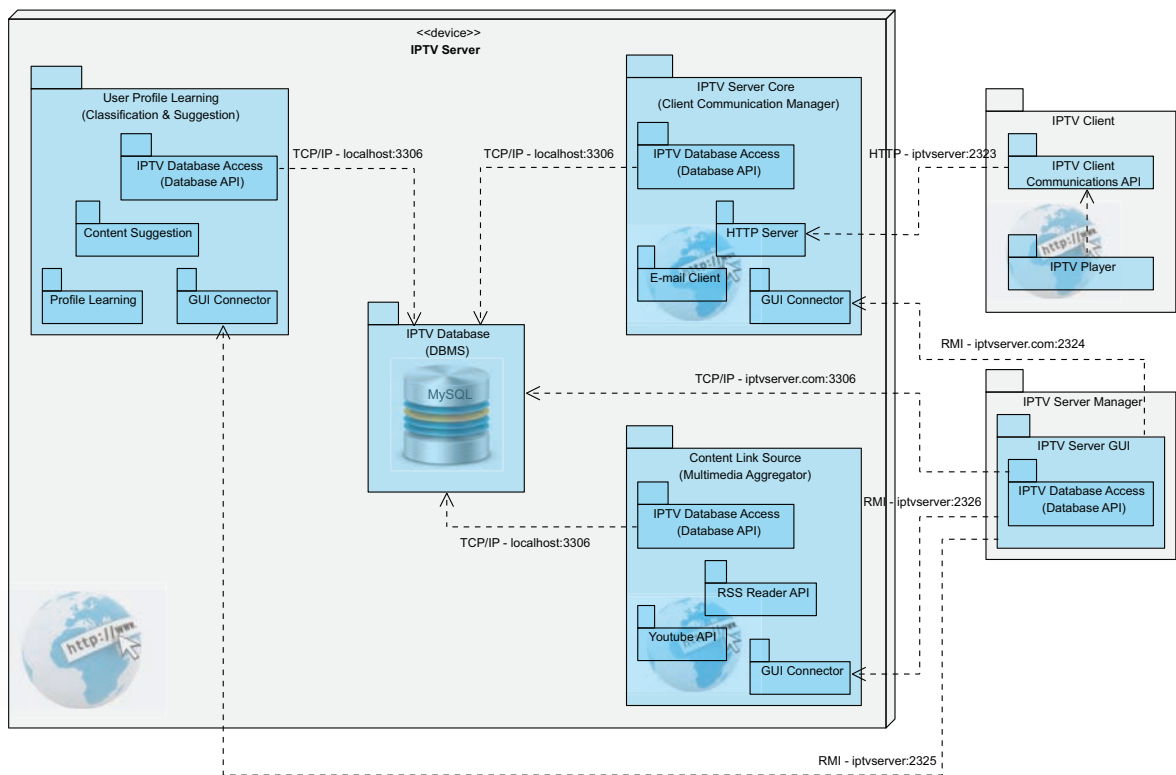


Figura 6.1: Diagrama de instalação do servidor IPTV com todos os módulos a funcionar na mesma máquina.

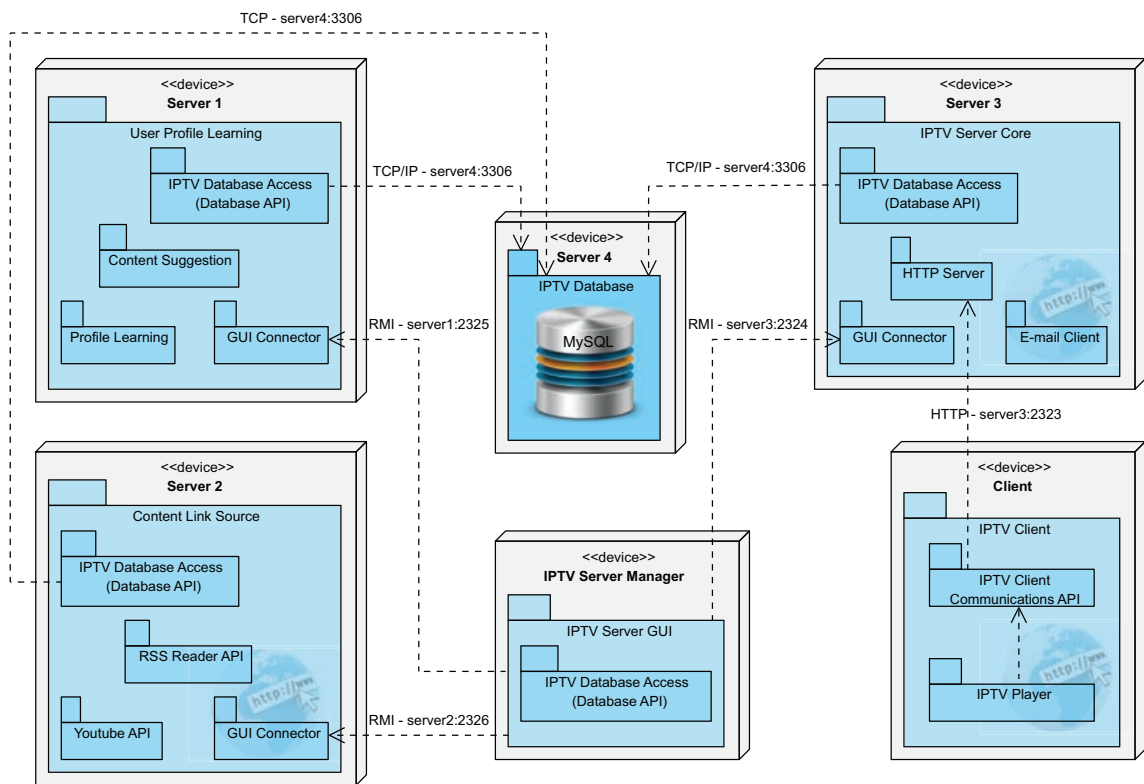


Figura 6.2: Diagrama de instalação do servidor IPTV com os módulos distribuídos por várias máquinas sem replicação de nenhum módulo.

Capítulo 7

Testes Gerais de Desempenho e Comportamento

7.1 Introdução

Neste capítulo serão apresentados e descritos os testes gerais efectuados ao sistema conjuntamente com os resultados dos mesmos e a sua análise. Os testes efectuados pretendem realizar medições de métricas básicas como o atraso na resposta a uma solicitação e a carga no servidor em termos de percentagem de recursos utilizados.

É de salientar que os testes efectuados exerceram uma carga muito elevada no servidor IPTV que, para os valores mais elevados, poderá não corresponder à realidade dada a sua índole de utilização. Posto isto, estes testes podem ser considerados como testes de *stress* do servidor (*stress-tests*).

Dado que foi considerado que nesta fase os módulos *Content Link Sources* e *User Profile Learning* ainda não possuem funcionalidades que levem a um nível de carga no servidor considerado relevante, foi somente colocado em execução, na mesma máquina, os módulos *IPTV Server Core* e *IPTV Database*.

Todos os testes foram realizados com a mesma base de dados, isto é, com a base de dados contendo exactamente os mesmos registos. Esta foi preenchida com um número bastante significativo de registos, de modo a melhor emular a realidade do sistema.

Ao nível dos parâmetros subjacentes, todos os testes foram realizados utilizando os mesmos sistemas computacionais e os mesmos parâmetros não variáveis.

7.2 Descrição dos Procedimentos

Os testes ao sistema foram baseados nos seguintes procedimentos e parâmetros:

- Execução de uma instância do módulo *IPTV Server Core* conjuntamente com a API de acesso à base de dados e a própria BD do sistema, utilizando uma única máquina (instalação centralizada - secção 6.3, figura 6.1). O sistema computacional servidor possuía as seguintes características gerais:

SO *Ubuntu 8.04 Server edition*;

Processador Dois processadores independentes *Intel Xeon Dual Core 3.0, 6MB cache* cada um;

RAM 4 GBytes;

Disco(s) Dois discos de 500 GBytes SATA;

Interface de Rede *Atheros 10/100/1000 Mbps*;

Fornecedor *Asus*;

Local Instituto de Telecomunicações - Aveiro;

- Número máximo de utilizadores simultaneamente autenticados no servidor a requererem a execução de tarefas: $C_{max} = 10000$. Deste modo, foram criados 10000 diferentes contas de utilizador no sistema com *user name* e *password* iguais ao padrão “*user_id*”, em que “id” é um número sequencial. É de notar que os clientes IPTV não foram executados no mesmo sistema computacional do servidor, tendo sido utilizados 4 computadores distintos, ligados por uma rede *Ethernet 10/100 Mbps*, para simular os múltiplos clientes;

- Número mínimo de utilizadores simultaneamente autenticados no servidor a requerem a execução de tarefas: $C_{min} = 100$;
- Dado o número de clientes estar compreendido entre 100 e 10000, o sistema foi analisado com um número de clientes igual a 100, 200, 500, 1000 e 10000 de forma independente, de modo a simplificar o processo de análise dos resultados;
- Foi criado um modelo de cliente de teste genérico, utilizando a API de comunicação com o servidor desenvolvido (secção 5). Este cliente foi modelado sob a forma de um *thread* que efectua um conjunto de tarefas e termina depois disso. As tarefas são executadas aleatoriamente segundo as seguintes regras:
 - A primeira mensagem é sempre de *User Authentication Request*, com uma probabilidade da autenticação ser com sucesso igual a $A = 75\%$ segundo uma distribuição uniforme;
 - Sempre que uma tentativa de autenticação falha, o cliente IPTV tenta novamente N vezes, sendo este valor calculado aleatoriamente com distribuição uniforme no intervalo $[0, 6]$;
 - A última mensagem pode ser de *User Deauthentication Request* com uma probabilidade de existir igual a $D = 50\%$, segundo uma distribuição uniforme;
 - Todas as restantes mensagens podem ocorrer com igual probabilidade segundo uma distribuição uniforme;
 - O número de mensagens enviadas é M , sendo este valor calculado aleatoriamente com distribuição uniforme, compreendida entre 7 e 350;
 - O tempo de espera entre o instante de recepção da resposta a uma mensagem enviada e o envio da mensagem seguinte é T , sendo este valor calculado aleatoriamente, com uma distribuição uniforme, no intervalo $[10, 100]$ ms;
- Para gerir todos os parâmetros dos testes e os *threads* cliente lançados, foi criado um *thread* controlador. Este *thread* controlador efectua também a escrita em ficheiro

(*Comma-Separated Values* (CSV)) de todos os dados relevantes que os vários clientes lançados devolvem depois de acabarem a sua simulação;

Os dados retirados da experiência constam de vários ficheiros CSV gerados (consoante o número de clientes simulado e os parâmetros utilizados) e foram tratados com recurso à utilização do *software Matlab R2009b*.

Os testes realizados visaram:

- Medir o atraso médio do servidor na resposta a cada tipo de mensagens ao longo do tempo, associado ao número de utilizadores/clientes no sistema;
- Comparar entre a utilização do protocolo HTTP e HTTPS;
- Comparar a utilização de validação XSD das mensagens recebidas pelo servidor com a sua não utilização;
- Medir a percentagem do CPU do servidor utilizado pelo sistema;

Na secção 7.3 encontram-se os resultados dos testes efectuados. Na secção 7.4 efectua-se a conclusão sobre os mesmos.

7.3 Resultados

Foram efectuados dois grandes grupos de testes: servidor com validação XSD das mensagens recebidas activa e inactiva. Para cada um destes cenários, foi testado o uso do protocolo HTTP e HTTPS, segundo o procedimento descrito na secção 7.2.

Para o caso de execução do servidor utilizando HTTP com validação XSD activa, os resultados dos testes, no que respeita ao atraso médio medido entre o envio de uma mensagem pelo cliente e a recepção da sua resposta pelo mesmo, podem ser vistos de forma gráfica na figura 7.1.

Relativamente à carga média no servidor durante os testes, a mesma pode ser observada na tabela 7.1. É de notar que os valores referem-se à carga global nos processadores em percentagem dos seus recursos totais.

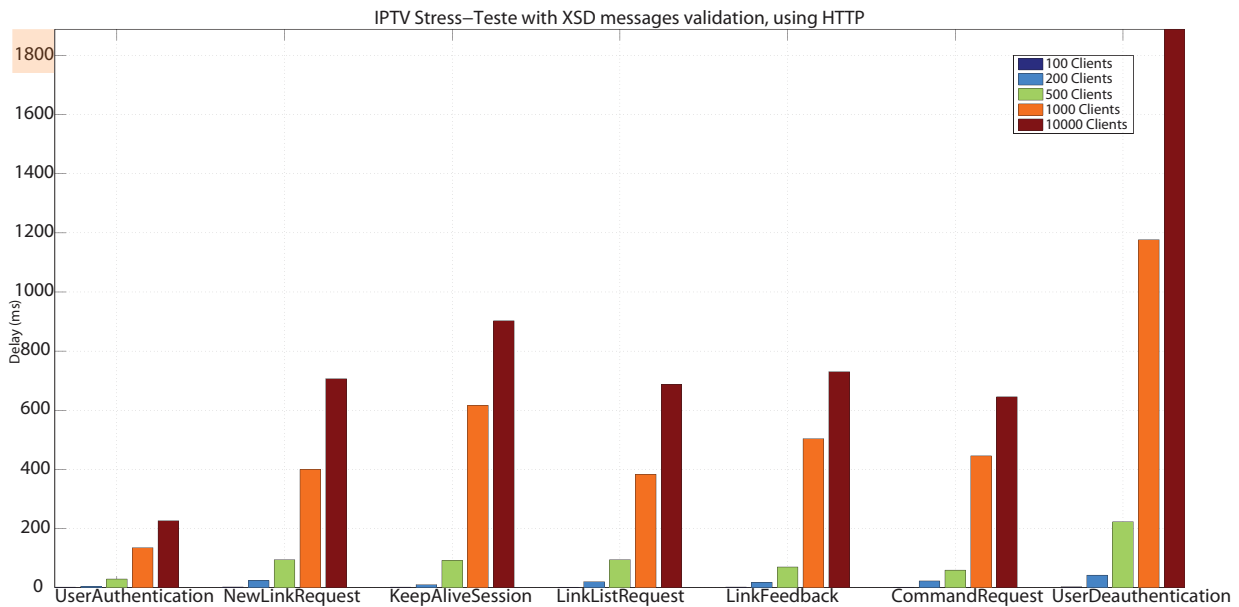


Figura 7.1: Gráfico referente aos atrasos produzidos utilizando HTTP e validação XSD das mensagens recebidas pelo servidor.

N.º de Clientes	Carga devido ao <i>Java</i>	Carga devido ao <i>MySQL</i>
100	78%	9%
200	80%	9%
500	82%	10%
1000	85%	13%
10000	86%	13%

Tabela 7.1: Tabela da carga média nos processadores do servidor IPTV para testes utilizando HTTP e validação XSD.

A tabela 7.2 possui os valores médios aproximados do atraso de resposta do servidor a qualquer mensagem dependente do número de clientes considerado, relativo ao cenário “HTTP com validação XSD”. Nesta tabela é também considerado o caso do desprezo da mensagem com maior atraso gerado neste cenário (*User Deauthentication*).

No que respeita ao caso de execução do servidor utilizando o protocolo HTTP mas sem validação XSD activa, os resultados dos testes, utilizando a mesma abordagem dos anteriores, podem ser vistos de forma gráfica na figura 7.2.

A carga média observada no servidor encontra-se descrita na tabela 7.3. Pela sua ob-

N.º de Clientes	Atraso médio	Atraso médio sem <i>User Deauthentication</i>
100	1.5ms	1.2ms
200	20ms	16ms
500	94ms	73ms
1000	523ms	414ms
10000	827ms	650ms

Tabela 7.2: Tabela do atraso médio para cada número de clientes considerado, utilizando HTTP com validação XSD das mensagens XML recebidas pelo servidor.

servação pode-se concluir que esta foi semelhante ao relatado na tabela 7.1 para o teste anterior. Mais uma vez, é de notar que os valores referem-se à carga global nos processadores em percentagem dos seus recursos totais.

N.º de Clientes	Carga devido ao <i>Java</i>	Carga devido ao <i>MySQL</i>
100	75%	9%
200	80%	10%
500	81%	10%
1000	84%	13%
10000	85%	14%

Tabela 7.3: Tabela da carga média nos processadores do servidor IPTV para testes utilizando HTTP sem validação XSD.

A tabela 7.4 possui os valores médios aproximados do atraso de resposta a qualquer mensagem dependente do número de clientes considerado, relativo ao cenário “HTTP sem validação XSD”. Nesta tabela é também considerado o caso do desprezo da mensagem com maior atraso gerado neste cenário (*User Deauthentication*).

Todos os resultados obtidos relativos aos cenários simulados, encontram-se devidamente comentados na secção 7.4.

Tal como já foi referido, foram também testados cenários utilizando HTTPS. No decorrer da análise dos resultados dos testes efectuados a esses cenários, verificou-se que o atraso médio causado pela utilização do protocolo HTTPS era desprezável, visto os atrasos médios medidos serem muito semelhantes aos atrasos médios medidos nos testes utilizando HTTP. Do mesmo modo, a percentagem de utilização do processador do sistema computacional

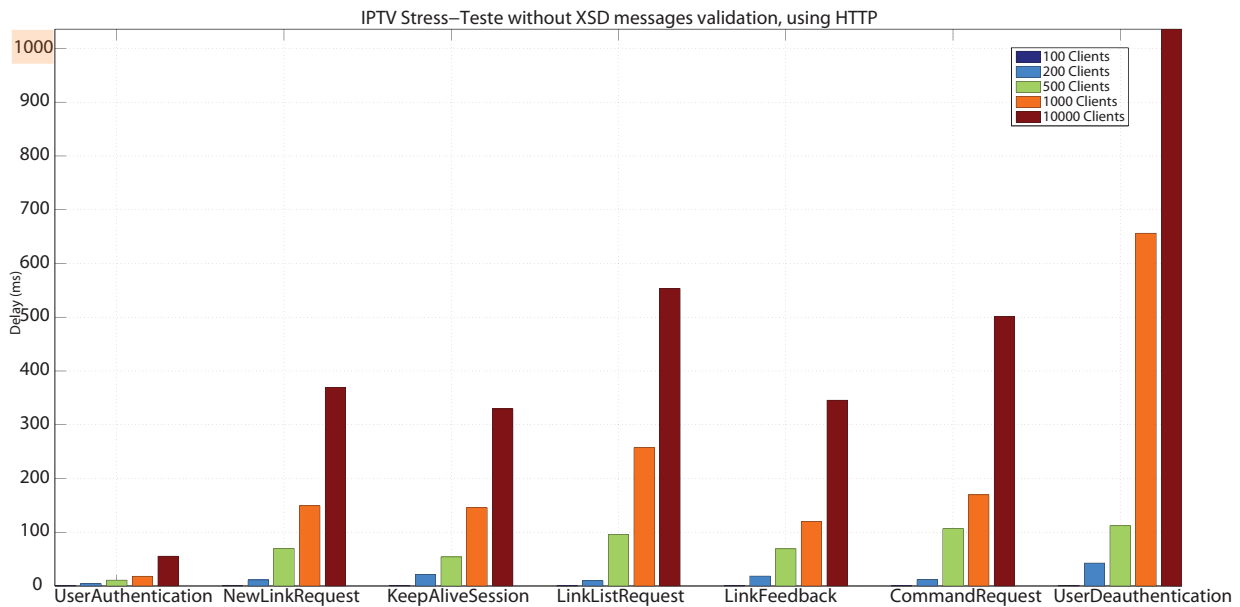


Figura 7.2: Gráfico referente aos atrasos produzidos utilizando HTTP sem validação XSD das mensagens recebidas pelo servidor.

servidor foi também muito semelhante à utilizada nos testes com HTTP, demonstrando que a carga no sistema reside fundamentalmente no processamento das mensagens e acesso à base de dados do servidor. Deste modo, não foi considerado necessário colocar neste documento a análise dos testes utilizando HTTPS.

N.º de Clientes	Atraso médio	Atraso médio sem <i>User Deauthentication</i>
100	0.4ms	0.2ms
200	17ms	13ms
500	74ms	68ms
1000	217ms	144ms
10000	456ms	359ms

Tabela 7.4: Tabela do atraso médio para cada número de clientes considerado, utilizando HTTP sem validação XSD das mensagens XML recebidas pelo servidor.

7.4 Conclusão

Como é possível observar pelos resultados obtidos, a tarefa/mensagem *User Deauthentication* foi a que mais atraso desenvolveu no servidor ao longo de todos os testes em todos os cenários. Isto deve-se, possivelmente, ao facto do sistema realizar a procura da actual sessão do utilizador na tabela *UserSessionData* que, depois de algum tempo de utilização, possui um número muito elevado de registos. Para resolver este problema, a criação de índices nos campos utilizados nas pesquisas nessa mesma tabela, poderá ser a solução. Para além disso, é também através desta operação que alguns dados da tabela de sistema (*SystemValue*) são actualizados bem como alguns recursos do sistema libertados.

No que diz respeito aos restantes tipos de mensagens testados, o atraso desenvolvidos por estas corresponde ao esperado e encontra-se dentro dos padrões admissíveis. De qualquer modo, é ainda possível efectuar muitas optimizações no sistema que permitam diminuir o tempo de atraso médio do mesmo, principalmente ao nível da utilização da base de dados.

Comparando a utilização do sistema com validação XSD activa das mensagens recebidas com a utilização do mesmo sistema mas sem validação XSD, é facilmente verificável que com validação activa, o sistema demora significativamente mais tempo a processar uma mensagem XML. Isto deve-se ao facto do servidor ter de abrir o ficheiro XSD, fazer o seu *parsing* e finalmente verificar se o XML é válido comparando-o com a sua definição XSD, fazendo assim uso de muito tempo extra e esforço computacional neste processo.

Quer utilizando HTTP quer utilizando HTTPS, o sistema comporta-se sensivelmente da mesma forma, não tendo sido detectado um acréscimo substancial no atraso do servidor quando utilizado HTTPS. Por este motivo, não foi considerado necessário mostrar aqui os resultados obtidos dos testes a este cenário.

Em qualquer uma das simulações efectuadas, a utilização do processador do servidor rondou sempre os 90%, nunca tendo ultrapassado os 29% de memória ocupada no caso do processo da máquina virtual de *Java* e 2% no caso do processo do *MySQLd* referente ao SGBD.

Do ponto de vista geral, o servidor IPTV comportou-se de uma forma bastante estável e rápida ao longo de todos os testes, tendo somente gerado algumas exceções no caso dos testes com dez mil clientes, devido a limitações impostas pelo *Java* ao nível do número máximo de ligações de rede activas simultaneamente. Os valores obtidos do atraso médio para qualquer mensagem (tabelas 7.2 e 7.4), foram muito satisfatórios tendo em conta o actual estado de desenvolvimento do servidor implementado.

Não foram testadas exaustivamente outras mensagens passíveis de serem enviadas ao servidor, como é o caso de *New User Request* ou *New User Validation Request*, dado que estas, comparativamente com as restantes, são utilizadas muito esporadicamente pelos clientes IPTV. Considerando o caso padrão, cada utilizador só irá requerer uma única vez, em toda a “vida útil” da sua conta de utilizador, a evocação de cada um destes dois tipos de mensagens.

Testes de diferente índole ou de índole semelhante mas mais aprofundados e específicos, poderão ser úteis para testar e avaliar de forma mais profunda o desempenho de cada módulo do servidor, numa fase de desenvolvimento subsequente.

Capítulo 8

Conclusões e Trabalho Futuro

Este capítulo apresenta e discute as conclusões retiradas após pesquisa e desenvolvimento deste trabalho. Aqui também serão apresentados alguns tópicos de trabalho futuro, sugestões e opiniões no âmbito de sistemas IPTV.

Esta Dissertação de Mestrado não se centrou exclusivamente numa só temática ou problema, tendo focado diversos tópicos relativos a sistemas IPTV no geral. Devido a isso, a elaboração deste trabalho possibilitou a aprendizagem de múltiplos novos conceitos e tecnologias no âmbito de sistemas IPTV, bem como a aquisição de uma maior noção do que é a actualidade e importância dessa tecnologia.

O sistema implementado contou com uma pesquisa e modelação prévia (UML) dos requisitos necessários e posterior implementação destes utilizando as tecnologias especificadas.

O servidor foi projectado tendo sempre em vista a sua real utilização, sendo que são conhecidas as suas actuais limitações e pontos onde é possível melhorar. Desta forma, este trabalho é visto como um processo em desenvolvimentos que não termina com a elaboração deste documento, podendo ser continuado explorando tanto as ideias aqui descritas como outras que surjam e colmatando as suas actuais deficiências.

Decorrente da análise deste documento ficou bem patente que o sistema ainda não contempla, por exemplo, os mecanismos de segurança considerados necessários para garantir políticas de AAA correctas, sendo este aspecto um dos pontos a melhorar futuramente no sistema. A implementação de mais tipos de mensagens é também um aspecto a considerar

caso se necessite de adicionar funcionalidades ao servidor. As mensagens XML actualmente implementadas garantem as funcionalidades básicas do sistema.

Como principio básico desta abordagem, o sistema IPTV não deverá actuar isoladamente quando a sua integração com outras tecnologias, sistemas ou serviços for uma mais-valia. Desta forma, alguns exemplos de sistemas/serviços cuja a integração no sistema seria interessante são:

Metadados Busca automática de múltiplas informações sobre os conteúdos multimédia utilizando serviços como o *The Internet Movie Database* (IMDB), *LastFM*, etc.;

Legendas Utilização de um dos muitos serviços disponíveis *online* que facultam legendas para filmes;

Tradução Utilização de serviços de tradução automática de legendas de conteúdos multimédia;

Redes sociais Integração e disponibilidade de acesso a redes sociais como o *Facebook*, o *Twitter*, o *hi5* e etc.;

Messaging Integração de um cliente de *Messaging* “universal” que suporte protocolos como AIM, ICQ, IRC, MSN, QQ, SILC, Yahoo, etc. (como é o caso do *Pidgin*);

Vídeo-chamada Integração com sistemas de vídeo-chamada e vídeo-conferência, bem como as redes de comunicação por voz, de modo a permitir efectuar e receber chamadas, SMSs e *Multimedia Messaging Services* (MMSs) no equipamento de TV;

Google Integração com outros serviços da *Google* como é o caso do *Google Wave*, *Google Maps*, *Google Talk*, etc.;

E-mail Integração de um cliente gestor de contas de *e-mail* (como é o caso do *Thunderbird*);

Geolocalização Integração de sistemas de geolocalização que tornassem possível, por exemplo, localizar outros utilizadores do sistema (caso estes permitissem), bens ou

serviços. O *Google Latitude* é um exemplo de um serviço gratuito actualmente em funcionamento, e que possibilita a partilha da localização entre utilizadores do mesmo;

Álbuns de Fotos Possibilidade de visualização e construção de apresentações/vídeos com fotos armazenadas localmente ou em serviços como o *Flickr*;

Guias turísticos Introdução de guias turísticos interactivos sobre múltiplas localizações no mundo, possuindo estes não só um carácter comercial e publicitário mas também lúdico e pedagógico. Este guias turísticos poderão possibilitar um grau de interactividade muito elevado, semelhante a um jogo de computador ou *software* análogo;

É de notar que o local de implementação (cliente IPTV ou servidor IPTV) de cada item da lista anterior varia consoante a semântica de cada um.

Utilizando este conceito de sistema IPTV universal, uma grande quantidade de serviços estaria disponível de uma forma natural no sistema, podendo ser acedidos de forma fácil e padrão através de qualquer dispositivo que suporte IPTV.

A personalização da área de ecrã pode não só ter como objectivo tornar o *layout* visualmente mais atractivo, mas também possibilitar a definição de “máscaras” contendo múltiplos conteúdos de vários tipos, provenientes de diferentes fontes, como adaptar os conteúdos a mostrar à área gráfica do dispositivo utilizado. A passagem de notícias personalizadas no roda-pé do sistema de TV, condições meteorológicas, a visualização de múltiplos conteúdos simultaneamente, o estabelecimento de comunicações de vídeo simultaneamente com vários utilizadores, entre outras funcionalidades, são exemplos da utilização de “mascaras”, caso o dispositivo de visualização as suporte.

Essas mesmas notícias e conteúdos extra poderiam também focar as áreas de interesses do utilizador. Um exemplo disso seria o anúncio de um concerto do musico preferido do utilizador (que vai acontecer para a semana), a evolução da bolsa de valores ou a informação de um novo atentado terrorista. Como será de esperar, as informações automáticas devem ter em consideração parâmetros como o perfil do utilizador, o local onde este se encontra no momento ou a hora do dia e mês do ano.

Para tornar a informação ainda mais acessível, a configuração de alertas via telemóvel (SMS, MMS ou mensagem de voz), *e-mail* ou outra via, de eventos ou acontecimentos importantes relativos ao perfil e/ou posição actual do utilizador, também poderá ser considerado.

Cada vez mais se assiste à criação de novos conceitos publicitários, utilizando todas as vias comunicativas disponíveis. Deste modo, o sistema IPTV projectado possui um carácter muito interessante neste campo, visto possibilitar a construção e aprendizagem do perfil do utilizador. Com esta informação tão específica sobre o utilizador, é possível oferecer a este bens e serviços pelo quais ele poderá estar verdadeiramente interessado, dado estarem de acordo com o seu perfil. A utilização de informações como a posição GPS do utilizador, hora do dia, época do ano entre outras, permite a apresentação de publicidade útil, interactiva e completamente personalizável.

Os sistemas IPTV constituem um avanço tecnológico muito significativo de carácter útil não só ao utilizador comum mas também às empresas. Já foi aqui mencionada a possibilidade de criação de publicidade personalizada e interactiva. Uma outra funcionalidade interessantes deste sistema é a criação fácil, versátil e poderosa de canais de televisão locais com conteúdos específicos e adaptados às necessidades das entidades, como por exemplo ginásios, lojas, restaurantes, monumentos, bares, postos de abastecimento, postos de turismo, aviões, transportes públicos (comboios, autocarros, etc.), locais de trabalho, museus, publicidades institucionais, promoção de eventos, imagens e notícias de transito, etc.

Para além do sistema permitir a criação de canais de TV acessíveis a qualquer utilizador, o próprio utilizador pode aceder ao seu próprio sistema de TV em qualquer local, quer seja através de um dispositivo móvel, computador portátil ou de um sistema de TV com STB de outra entidade. Deste modo é possível que, por exemplo, um utilizador continue a ver o programa que estava a assistir em casa, mesmo que tenha de ir ao ginásio ou fazer um viagem de comboio demorada. O Utilizador tem assim a possibilidade de levar sempre “a sua televisão” consigo para qualquer lado, mesmo que não possua um dispositivo móvel.

A criação de um cartão de identificação do utilizador no sistema (com tecnologia *Radio-Frequency IDentification* (RFID) ou *chip*, por exemplo) para fácil autenticação, autorização

e contabilização da utilização do sistema, pode substituir a autenticação normal (*login* e *password*) no sistema em dispositivos públicos, tornando o serviço ainda mais acessível e disponível.

Dado que os utilizadores podem possuir *a priori* dispositivos como consolas de jogos, computadores *desktop*, computadores portáteis, leitor de *Blu-Ray* e/ou telemóveis e *smart-phones*, deverá ser permitida a interacção e partilha de recursos e conteúdos entre os sistema IPTV e estes dispositivos. Esta funcionalidade requer a adição de mais algumas tarefas no servidor mas depende em muito da implementação específica do cliente IPTV, sendo este também um grande responsável pela agregação de múltiplos serviços e tecnologias.

Tendo em conta as tecnologias existentes, no caso de um sistema IPTV ser implementado sobre uma rede sem garantias de QoS como é o a Internet, os conteúdos multimédia deverão ser partilhados segundo uma política P2PTV. Com esta abordagem consegue-se um serviço com melhor qualidade de serviço e logo de experiência também, onde todos os dispositivos clientes com mais capacidade de armazenamento actuam como nós consumidores e fornecedores de conteúdos. Mesmo com esta tecnologia, é sempre necessário a existência de “servidores” centrais que introduzem inicialmente e actualizam os conteúdos multimédia nos *peers* da rede.

Ao nível do servidor de acesso ao sistema IPTV, uma abordagem de instalação (não testada) possível, poderia ser baseada numa arquitectura distribuída com replicação (secção 6.3) utilizando virtualização de recursos. Deste modo, cada utilizador possui uma conta no servidor que lhe permite iniciar, terminar e efectuar tarefas na sua máquina virtual, onde teria virtualmente só para si uma instância de cada módulo do servidor e um conjunto de recursos disponíveis. Seria assim criado um nível de isolamento e segurança nas operações e tarefas, permitindo também uma melhor gestão dos recursos computacionais disponíveis no servidor.

No que respeita aos testes exaustivos de desempenho do sistema (secção 7), estes revelaram resultados bastante satisfatórios tendo em conta o actual estado de desenvolvimento e optimização do servidor. O sistema manteve-se sempre operacional e estável durante todas as operações, acontecendo simplesmente algumas excepções nos clientes de teste de-

vido a *connection time out*. Estas exceções são devidas a limitações impostas pelo *Java* ao nível do número máximo de ligações de rede activas simultaneamente.

Desta forma, dá-se por terminada esta etapa do desenvolvimento do Servidor IPTV com Personalização Automática de Canais, esperando a continuidade deste trabalho e/ou do conceito subjacente.

Anexo A

Diagramas de Base de Dados

Neste anexo encontram-se os diagramas de base de dados que se revestem de maior importância tanto na construção como análise e *debug* da mesma. Os diagramas são auto-explicativos no que se refere à função e ao tipo de dados de cada campo de cada entidade. A descrição da funcionalidade de cada entidade encontra-se na tabela 4.2. Quaisquer dúvidas no mapeamento entre os tipos de dados personalizados e os reais utilizados no SGBD, podem ser esclarecidas através da consulta da tabela 4.3.

Para consulta dos valores colocados por defeito pelo sistema de inicialização automática, é favor consulta o código *Java* fornecido, mais especificamente, a directoria da API

DatabaseAccess e, dentro desta, o ficheiro *PutInitialValues.java* pertencente ao pacote *databaseaccess.initialvalues*.

Tal como mencionado no capítulo 4.2, pode existir a necessidade de implementar no servidor uma política de restrição de conteúdos dependente do utilizador final. Esta restrição e controlo podem ser conseguidos através da utilização do conceito de “pacote de conteúdos multimédia”, em que só é permitido ao utilizador aceder aos conteúdos que pertençam aos pacotes que este detém e aos conteúdos que não possuam qualquer pacote associado. A figura A.1 exemplifica as possíveis alterações na BD de modo a implementar uma política de controlo de acesso aos conteúdos com base na criação de “pacotes de conteúdos multimédia”.

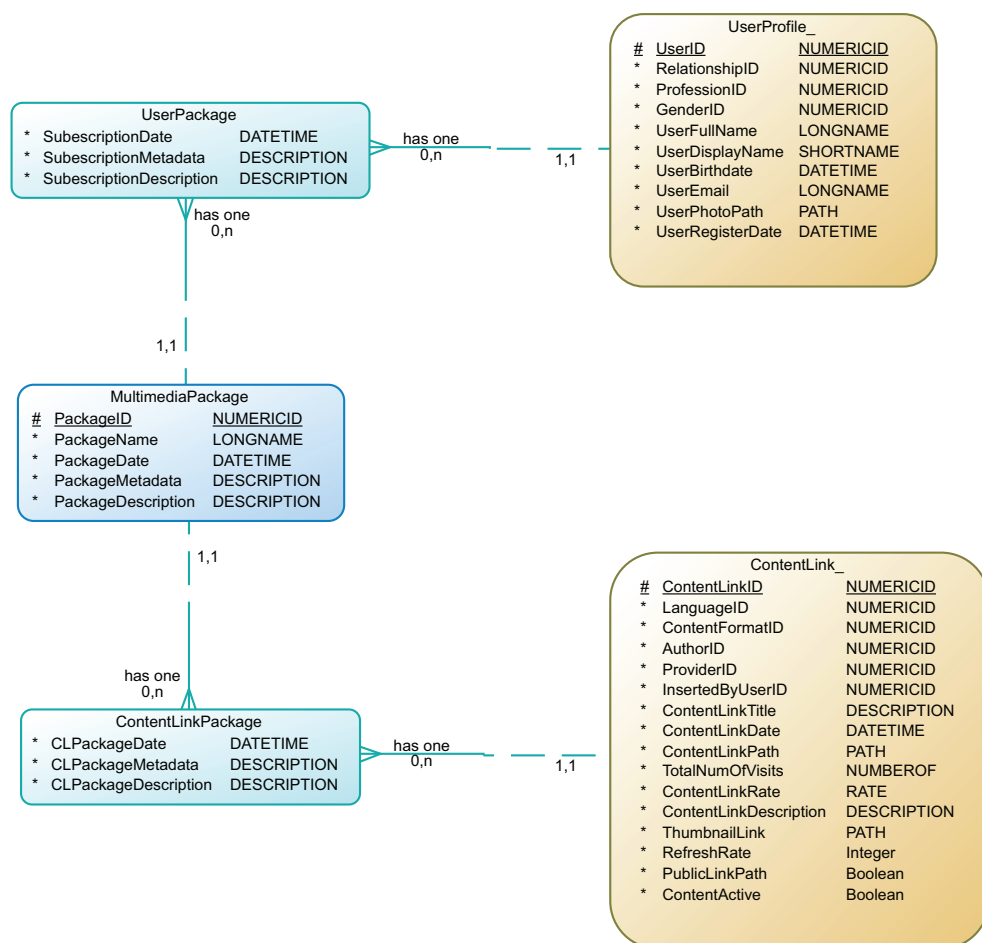


Figura A.1: Diagrama físico das novas entidades a introduzir na BD para permitir a criação de “pacotes de conteúdos multimédia”.

Seguem-se então os três diagramas da base de dados em formato completo. Estes são o diagrama conceptual, o diagrama lógico e o diagrama físico, respectivamente.

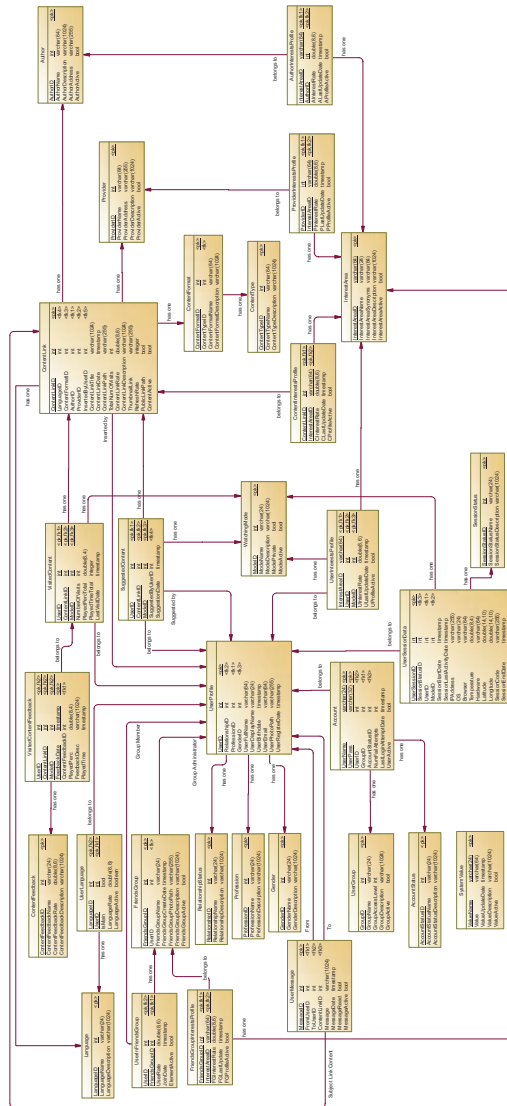


Figura A.4: Diagrama do modelo físico da base de dados implementada em notação relacional.

Anexo B

Descrição das Mensagens XML e Respectiva Sintaxe XSD

Este anexo é referente às mensagens definidas para a comunicação entre o servidor IPTV (mais especificamente, o módulo *IPTV Server Core* responsável pela comunicação cliente-servidor) e um cliente IPTV genérico. Para cada tipo de mensagem implementada é possível encontrar neste anexo a sua especificação XSD e o diagrama que exprime as relações entre os elementos e atributos que compõem a mensagem. Para cada tipo de mensagem é também mostrado os nomes dos atributos pertencentes a cada elemento XML que a constitui bem como os seus tipos de dados, valores por defeito, tipo de utilização (obrigatória/opcional) e funcionalidade. Por último, é exemplificada uma instância específica do tipo de mensagem em questão para uma melhor compreensão da estrutura da mesma.

É de salientar que as mensagens especificadas neste anexo encontram-se totalmente implementadas no servidor. Tendo todo o servidor sido criado numa perspectiva de programação modular, clara e bem organizada, para a alteração ou adição de novas mensagens é somente necessário alterar ou adicionar três entidades: a classe *Java* que efectua a validação sintática e conversão dos dados do formato XML para os atributos da classe; o método que executa as tarefas associadas à semântica da mensagem; a árvore de decisão lógica que permite, dada uma qualquer mensagem, executar o método específico que contém as tarefas necessárias à sua execução.

Para uma fácil e rápida expansão das funcionalidades de comunicação do servidor IPTV com o cliente, é possível utilizar as mensagens do tipo *Command Request* (secção B.11) e *Command Response* (secção B.12) para requerer e responder à requisição de qualquer comando ou operação.

Como a implementação do cliente IPTV não pode ser totalmente controlada e restringida pelas regras de implementação do servidor, o significado do conteúdo de alguns campos pode variar. Exemplo disso é um campo com conteúdo vazio: em alguns casos o servidor permite campos vazios (vazio é diferente de *null!*), sendo que cabe ao cliente a sua posterior correcta interpretação.

B.1 Mensagem *User Authentication*

Esta mensagem tem como funcionalidade permitir que um utilizador se autentique e inicie uma nova sessão no servidor IPTV através de uma opção para o efeito disponibilizada pelo cliente IPTV. O servidor ao receber uma mensagem deste tipo irá responder sempre com uma mensagem *Command Response* (secção B.12) cujo conteúdo indica o sucesso ou não da operação e possui a descrição das possíveis falhas ou problemas. No caso de uma autenticação com sucesso, são ainda enviados na mensagem *Command Response* todos os dados relativos ao utilizador que acabou de iniciar sessão. Estes dados encontram-se dentro do elemento XML *nested*, que se encontra dentro do elemento *response*, e a sua sintaxe é exactamente a mesma utilizada na mensagem *New User* no envio dos dados do utilizador.

A definição formal em XSD para este tipo de mensagem pode ser encontrada de seguida.

```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!--
3 XML schema for the messages 'User Authentication Request' sent by the IPTV Client to
  the IPTV Server when a user want to login in the system.
4 -->
5 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
6   <xs:element name="communication">
7     <xs:complexType>
8       <xs:sequence>
9         <xs:element ref="command" />
10        <xs:element ref="user" />
11      </xs:sequence>

```

```

12     </xs:complexType>
13 </xs:element>
14 <xs:element name="command">
15     <xs:complexType>
16         <xs:attribute name="name" use="required">
17             <xs:simpleType>
18                 <xs:restriction base="xs:string">
19                     <xs:enumeration value="UserAuthenticationRequest" />
20                 </xs:restriction>
21             </xs:simpleType>
22         </xs:attribute>
23     </xs:complexType>
24 </xs:element>
25 <xs:element name="user">
26     <xs:complexType>
27         <xs:attribute name="password" type="xs:string" use="required" />
28         <xs:attribute name="name" type="xs:string" use="required" />
29         <xs:attribute name="ipaddress" type="xs:string" use="required" />
30         <xs:attribute name="os" type="xs:string" use="required" />
31         <xs:attribute name="browser" type="xs:string" use="required" />
32         <xs:attribute name="temperature" type="xs:decimal" use="optional" />
33         <xs:attribute name="hardware" type="xs:string" use="required" />
34         <xs:attribute name="latitude" type="xs:decimal" use="optional" />
35         <xs:attribute name="longitude" type="xs:decimal" use="optional" />
36         <xs:attribute name="watchingmode" type="xs:string" use="optional" />
37     </xs:complexType>
38 </xs:element>
39 </xs:schema>

```

Esta definição pode ainda ser visualizada de uma forma mais intuitiva e compacta através do diagrama da figura B.1.

A tabela B.1 descreve, para cada elemento da mensagem XML, o nome dos seus atributos bem como o tipo de dados permitido, tipo de utilização (obrigatória/opcional) e valores por defeito.

Descrita toda a sintaxe para este tipo de mensagem enviada pelo cliente para o servidor IPTV com um pedido de autenticação de um utilizador, segue-se agora a descrição semântica e funcional de cada elemento XML e seus atributos (quando estes existem).

Elemento *communication* Este elemento XML é obrigatório e é a “raíz” da árvore XML definida pela mensagem. A sua funcionalidade restringe-se a ser o elemento base que contém todos os outros semanticamente mais importantes e não possui atributos específicos.

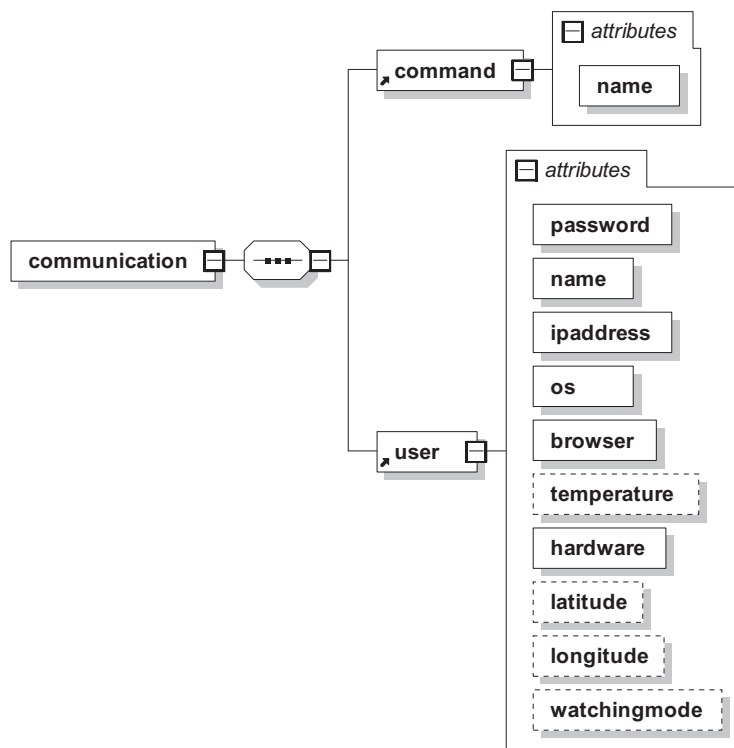


Figura B.1: Diagrama das relações entre os elementos e atributos para mensagens do tipo *User Authentication*.

Elemento	Nome do Atributo	Tipo do Atributo	Uso do Atributo	Default do Atributo
communication	–	–	–	–
command	<i>name</i>	string	obrigatório	–
user	<i>password</i>	string	obrigatório	–
	<i>name</i>	string	obrigatório	–
	<i>ipaddress</i>	string	obrigatório	–
	<i>os</i>	string	obrigatório	–
	<i>browser</i>	string	obrigatório	–
	<i>temperature</i>	decimal	opcional	–
	<i>hardware</i>	string	obrigatório	–
	<i>latitude</i>	decimal	opcional	–
	<i>longitude</i>	decimal	opcional	–
<i>watchingmode</i>	string	opcional	–	

Tabela B.1: Tabela dos elementos e atributos para mensagens do tipo *User Authentication*

Elemento *command* Este elemento obrigatório define e identifica o tipo de mensagem de que se trata. Para o caso de mensagens *User Authentication* este elemento possui sempre o seu único atributo (que também é obrigatório) “name” com o valor igual a “UserAuthenticationRequest”, que é o identificador deste tipo de mensagens.

Elemento *user* Este elemento XML obrigatório contém nos seus atributos a informação útil relativa ao pedido de autenticação de um utilizador. Os atributos deste elemento e sua semântica são:

password Atributo obrigatório que contém a *password* do utilizador previamente encriptada através de um algoritmo de encriptação que no caso poderá ser o MD5. Tem como funcionalidade permitir alguma privacidade e segurança no acesso ao sistema bem como a identificação do utilizador em conjunto com o *user name*;

name Atributo obrigatório que contém o nome do utilizador relativo à sua conta no sistema (*user name*). Tem como funcionalidade permitir alguma privacidade e segurança no acesso ao sistema bem como a identificação do utilizador em conjunto com a *password*;

ipaddress Atributo obrigatório que contém o endereço IP da máquina que o utilizador está a utilizar para tentar aceder ao sistema. Permite a identificação da proveniência da tentativa de acesso ao serviço bem como possíveis políticas de bloqueio e segurança;

os Atributo obrigatório que contém o nome do Sistema Operativo (Windows XP, Ubuntu 9.04, Mac OS X, etc.) que o utilizador está a operar aquando a tentativa de acesso ao servidor IPTV. Tem como funcionalidade permitir uma maior parametrização do serviço disponibilizado, adaptando-se a possíveis especificidades do Sistema Operativo em uso.

browser Atributo obrigatório que, caso o acesso esteja a ser efectuado por um cliente *web*, contém o nome do *browser* (programa de acesso à Internet como por ex-

emplo o Internet Explorer, Mozilla Firefox, etc.) que está a ser utilizado e o URI base do serviço, separados por “_” – *underscore* (exemplo: “Mozilla Firefox_http://www.omeuclienteIPTV.com”). Caso o acesso seja efectuado por um cliente não *web* (por exemplo, um programa *desktop* ou *mobile*), este atributo contém o nome do programa que está a ser utilizado (exemplo: “oMeu-ClienteIPTV”). Este campo serve essencialmente para parametrizar o serviço disponibilizado relativamente ao tipo de cliente IPTV em uso;

temperature Atributo opcional que, caso o sistema computacional possua *hardware* que meça a temperatura ambiente sentida pelo utilizador, contém a mesma em graus *Celsius*. Caso contrário, este atributo não existe no elemento *user* da mensagem XML. Este campo tem como funcionalidade permitir que o sistema obtenha informações mais específicas sobre o ambiente em que o utilizador se encontra de forma a melhor parametrizar o serviço disponibilizado. Caso o dispositivo do utilizador não possua as funcionalidades de termómetro nem de GPS, o cliente IPTV deverá possibilitar a introdução textual da morada, cidade ou, no limite, o país onde o utilizador se encontra, e posteriormente determinar a temperatura local recorrendo a serviços gratuitos como o *AccuWeather*;

hardware Atributo obrigatório que identifica o *hardware* do sistema computacional utilizado na tentativa de acesso ao sistema. Este atributo serve para distinguir os vários tipos de dispositivos que podem ser utilizados no acesso ao sistema e assim disponibilizar um serviço mais escalável e próximo das reais capacidades do equipamento em uso;

latitude Atributo opcional que, caso o sistema computacional possua hardware de geolocalização (GPS), contém o valor medido para a latitude da actual posição do utilizador. Este atributo, conjuntamente com o atributo *longitude*, tem como funcionalidade permitir que o sistema de IPTV obtenha informações mais específicas sobre o ambiente em que o utilizador se encontra de forma a melhorar a parametrização do serviço disponibilizado;

longitude Atributo opcional que, caso o sistema computacional possua hardware de geolocalização (GPS), contém o valor medido para a longitude da actual posição do utilizador. Este atributo, conjuntamente com o atributo *latitude*, tem como funcionalidade permitir que o sistema obtenha informações mais específicas sobre o ambiente em que o utilizador se encontra de forma a melhor parametrizar o serviço disponibilizado. Caso o dispositivo do utilizador não possua a funcionalidade de GPS, o cliente IPTV deverá possibilitar a introdução textual da morada, cidade ou, no limite, o país onde o utilizador se encontra, e posteriormente calcular as coordenadas GPS aproximadas dessa mesma localização, recorrendo a serviços como o *Google Maps*;

watchingmode Atributo opcional que contém o modo de visualização que o utilizador deseja usar quando inicia a sessão no sistema IPTV. Sendo este atributo opcional, quando o mesmo não é especificado é assumido o modo de visualização denominado por *alone*. É de notar que este atributo, com ou sem valor inicial, define um parâmetro do servidor IPTV que pode ser alterado a qualquer momento durante a utilização do sistema sem necessidade de terminar e voltar a iniciar sessão (no caso da utilização de modos não limitadores da utilização do serviço - controlo parental). Esta funcionalidade pode ser comparada aos estados “online”, “ausente”, “ocupado”, entre outros, disponibilizados geralmente pelos programas de *Instant messaging*. A funcionalidade deste atributo tem a ver com a parametrização dinâmica do sistema, permitindo permutar de forma fácil entre diferentes modos que reflectem o ambiente de visualização dos conteúdos bem como os próprios estados de espírito do utilizador ou um possível controlo parental. Neste último caso (controlo parental), a utilização de um modo que limite a utilização do sistema pode obrigar à introdução das credenciais de acesso para permitir efectuar algumas tarefas bloqueadas, inclusive mudar o próprio modo de visualização.

Um exemplo de instanciação deste modelo de mensagens numa mensagem XML (com

todos os elementos e atributos possíveis) que requer uma operação de autenticação de um utilizador, pode ser visto seguidamente.

```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!--
3 XML to be sent by the IPTV Client to the IPTV Server when a user want to login in the
  system.
4 -->
5 <communication xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
6   <command name="UserAuthenticationRequest"/>
7   <user name="joaoffr" password="6d6b9d0cfe9f35b4d8ba5874931d914d"
  ipaddress="212.452.76.23" os="Ubuntu 9.04" browser="Mozilla Firefox 3.5"
  temperature="15.7" hardware="AMD Athlon X3" latitude="0.0" longitude="0.0"
  watchingmode="allone"/>
8 </communication>

```

B.2 Mensagem *User Deauthentication*

Esta mensagem tem como funcionalidade permitir que um utilizador termine a sua sessão (previamente iniciada) de forma regular no servidor IPTV através de uma opção para o efeito disponibilizada pelo cliente IPTV. É de notar que o servidor ao receber uma mensagem deste tipo irá responder sempre com uma mensagem *Command Response* (secção B.12) cujo conteúdo indica o sucesso ou não da operação. Caso a operação falhe, apresenta a descrição das possíveis falhas ou problemas detectados.

De seguida encontra-se a definição formal em XSD para este tipo de mensagem.

```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!--
3 XML schema for the messages 'User Deauthentication Request' sent by the IPTV Client to
  the IPTV Server when a user want to logout from the system.
4 -->
5 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
6   <xs:element name="communication">
7     <xs:complexType>
8       <xs:sequence>
9         <xs:element ref="command"/>
10        <xs:element ref="user"/>
11      </xs:sequence>
12    </xs:complexType>
13  </xs:element>
14  <xs:element name="command">
15    <xs:complexType>

```

```

16     <xs:attribute name="name" use="required">
17         <xs:simpleType>
18             <xs:restriction base="xs:string">
19                 <xs:enumeration value="UserDeauthenticationRequest" />
20             </xs:restriction>
21         </xs:simpleType>
22     </xs:attribute>
23 </xs:complexType>
24 </xs:element>
25 <xs:element name="user">
26     <xs:complexType>
27         <xs:attribute name="sessionid" type="xs:string" use="required" />
28         <xs:attribute name="sessioncode" type="xs:string" use="required" />
29     </xs:complexType>
30 </xs:element>
31 </xs:schema>

```

A estrutura deste tipo de mensagem pode ainda ser visualizada de uma forma mais intuitiva e compacta através do diagrama da figura B.2.

A tabela B.2 descreve, para cada elemento da mensagem XML, o nome dos seus atributos bem como o tipo de dados permitido, tipo de utilização (obrigatória/opcional) e valores definidos por defeito.

Descrita toda a sintaxe para este tipo de mensagem enviada pelo cliente para o servidor IPTV com um pedido de fim de sessão previamente iniciada pelo utilizador, segue-se agora a descrição semântica e funcional de cada elemento XML e seus atributos (quando estes existem).

Elemento *communication* Este elemento XML é obrigatório e é a “raíz” da árvore XML

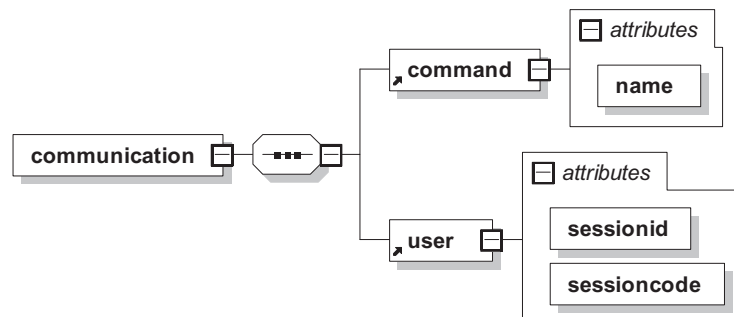


Figura B.2: Diagrama das relações entre os elementos e atributos para mensagens do tipo *User Deauthentication*.

Elemento	Nome do Atributo	Tipo do Atributo	Uso do Atributo	Default do Atributo
communication	–	–	–	–
command	<i>name</i>	string	obrigatório	–
user	<i>sessionid</i>	long	obrigatório	–
	<i>sessioncode</i>	string	obrigatório	–

Tabela B.2: Tabela dos elementos e atributos para mensagens do tipo *User Deauthentication*

definida pela mensagem. A sua funcionalidade restringe-se a ser o elemento base que contém todos os outros semanticamente mais importantes e não possui quaisquer atributos específicos.

Elemento *command* Este elemento obrigatório define e identifica o tipo de mensagem de que se trata. Para o caso de mensagens *User Deauthentication* este elemento possui sempre o seu único atributo (que também é obrigatório) “name” com o valor igual a “UserDeauthenticationRequest”, que é o identificador deste tipo de mensagens.

Elemento *user* Este elemento XML obrigatório contém nos seus atributos a informação útil que será usada para efectuar o fecho da corrente sessão do utilizador no sistema. Os atributos deste elemento e sua semântica são:

sessionid Atributo obrigatório que contém um identificador numérico que permite relacionar o utilizador com a sua actual sessão activa no sistema. Tal como no caso deste tipo de mensagem (que identifica uma operação), sempre que uma operação (sobre a forma de uma mensagem XML) que necessite de ser requirida por um utilizador autenticado para se realizar seja enviada ao servidor IPTV, este atributo conjuntamente com o *sessioncode* terá de ser enviado num elemento *user* possuindo valores válidos. Isto deve-se não só à necessidade de identificação do utilizador que requer a operação mas também ao estabelecimento de políticas de segurança e autenticação;

sessioncode Atributo obrigatório que contém uma cadeia de caracteres que permite ao sistema verificar a autenticidade da operação requisitada pela mensagem

XML. Este campo em conjunto com o *sessionid* permite a identificação da sessão do utilizador no sistema (se esta existir) e permite também o estabelecimento de políticas de segurança na realização das operações.

Um exemplo de uma instância XML deste modelo de mensagens que requer uma operação de fim de sessão de um utilizador, pode ser visto seguidamente.

```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!--
3 XML to be sent by the IPTV Client to the IPTV Server when a user want to logout from
  the system.
4 NOTE: 'sessioncode' should not have the 'sessionid' at the end. This should have
  exactly 32 characters.
5 -->
6 <communication xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
7   <command name="UserDeauthenticationRequest" />
8   <user sessionid="10" sessioncode="1fea143b66b02696ead45ffacb792347" />
9 </communication>

```

B.3 Mensagem *New User*

Este tipo de mensagem permite que sejam adicionados novos utilizadores ao sistema de uma forma interactiva. Assim sendo, o cliente IPTV só necessita recolher os dados necessários relativos ao novo utilizador e enviar uma mensagem XML deste tipo para o servidor IPTV. Este irá responder sempre com uma mensagem *Command Response* (secção B.12), cujo conteúdo varia de acordo com a correcção sintáctica e semântica dos dados enviados, podendo a criação do novo utilizador suceder ou falhar. No caso da existência de alguma falha, é apresentada a descrição da mesma bem como uma possível causa.

Devido ao facto da definição formal em XSD para este tipo de mensagem ser muito extensa, esta não consta deste documento, podendo ser encontrada na directoria

“IPTVFiles/xml/NewUserRequest-Communication.xsd” presente na pasta que contém o código do servidor.

Esta definição pode ainda ser visualizada de uma forma mais intuitiva e compacta através do diagrama da figura B.3.

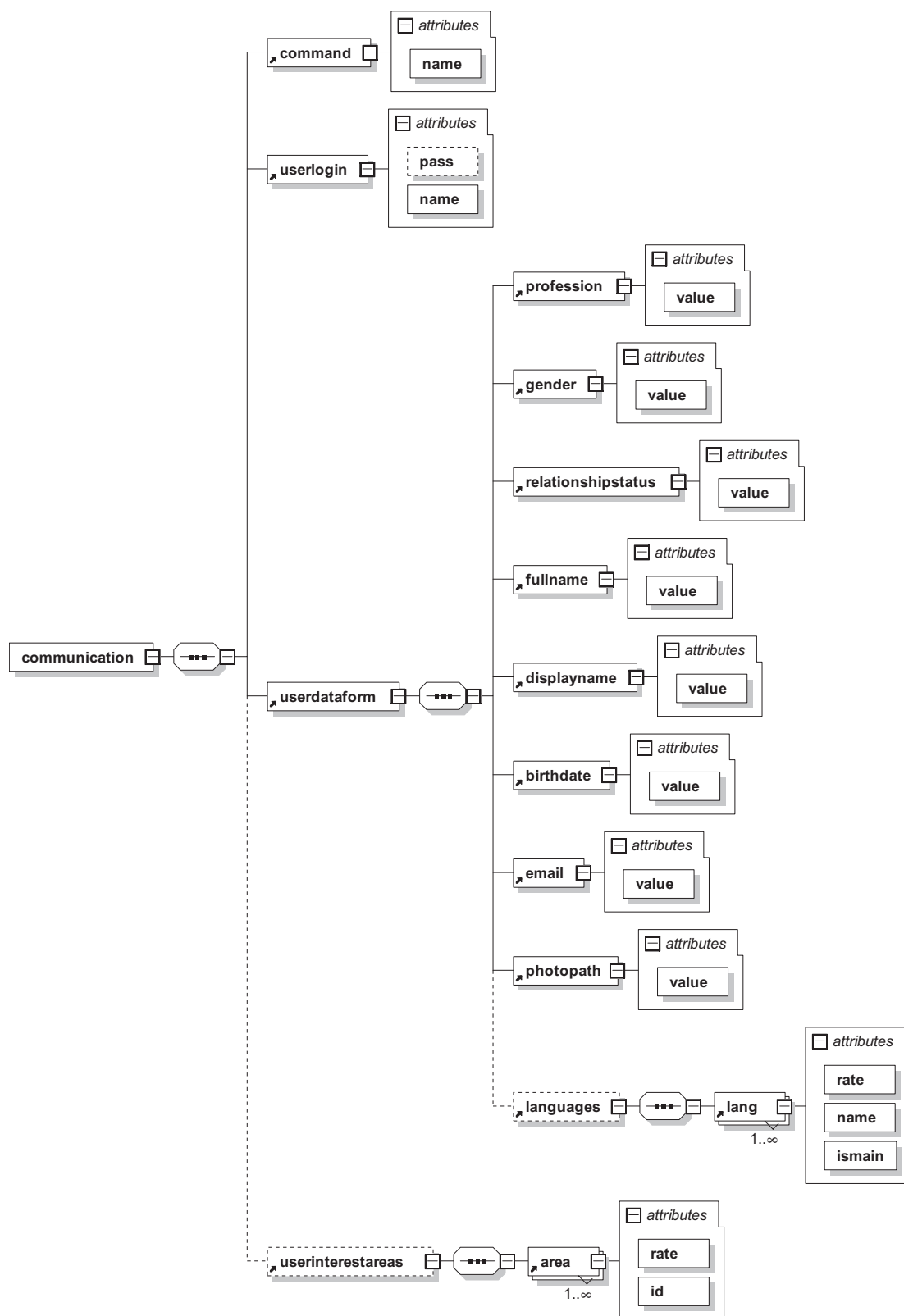


Figura B.3: Diagrama das relações entre os elementos e atributos para mensagens do tipo *New User*.

A tabela B.3 descreve, para cada elemento da mensagem XML, o nome dos seus atributos bem como o tipo de dados permitido, tipo de utilização (obrigatória/opcional) e valores por defeito.

Descrita toda a sintaxe para este tipo de mensagem enviada pelo cliente para o servidor IPTV com um pedido de registo de um novo utilizador do sistema, segue-se agora a descrição semântica e funcional de cada elemento XML e seus atributos (quando estes existem).

Elemento *communication* Este elemento XML é obrigatório e é a “raíz” da árvore XML definida pela mensagem. A sua funcionalidade restringe-se a ser o elemento base que contém todos os outros semanticamente mais importantes e não possui atributos específicos.

Elemento *command* Este elemento obrigatório define e identifica o tipo de mensagem de que se trata. Para o caso de mensagens *New User* este elemento possui sempre o seu único atributo (que também é obrigatório) “name” com o valor igual a “NewUserRequest”, que é o identificador deste tipo de mensagens.

Elemento *userlogin* Este elemento XML obrigatório contém nos seus atributos a informação que será utilizada para futura autenticação no sistema. É de notar que um utilizador não se poderá autenticar pela primeira vez no sistema sem efectuar a validação da sua nova conta. Os atributos deste elemento e sua semântica são:

name Atributo obrigatório que contém o nome do utilizador relativo à sua conta no sistema (*user name*). Tem como funcionalidade permitir alguma privacidade e segurança no acesso ao sistema bem como a identificação do utilizador em conjunto com a sua *password*;

pass Atributo opcional que contém uma *password* do utilizador previamente encriptada através de um algoritmo de encriptação que no caso poderá ser o MD5. Este atributo é opcional devido a ser sempre necessária a sua alteração durante o processo de validação da nova conta do utilizador. Caso não seja especificada

Elemento	Nome do Atributo	Tipo do Atributo	Uso do Atributo	Default do Atributo
communication	–	–	–	–
command	<i>name</i>	string	obrigatório	–
userlogin	<i>name</i> <i>pass</i>	string string	obrigatório opcional	– –
userdataform	–	–	–	–
profession	<i>value</i>	string	obrigatório	–
gender	<i>value</i>	string	obrigatório	–
relationshipstatus	<i>value</i>	string	obrigatório	–
fullname	<i>value</i>	string	obrigatório	–
displayname	<i>value</i>	string	obrigatório	–
birthdate	<i>value</i>	string	obrigatório	–
email	<i>value</i>	string	obrigatório	–
photopath	<i>value</i>	string	obrigatório	–
languages	–	–	–	–
lang	<i>rate</i>	decimal	obrigatório	–
	<i>name</i>	string	obrigatório	–
	<i>ismain</i>	byte	obrigatório	–
userinterestareas	–	–	–	–
area	<i>rate</i>	decimal	obrigatório	–
	<i>id</i>	string	obrigatório	–

Tabela B.3: Tabela dos elementos e atributos para mensagens do tipo *New User*

um valor para este atributo, será gerada uma *password* aleatória que será alterada aquando da validação da conta. Este atributo tem como funcionalidade permitir a identificação do utilizador em conjunto com o *user name* bem como conferir segurança no processo de validação da conta do utilizador no sistema.

Elemento *userdataform* Este elemento XML obrigatório é o elemento aglutinador dos dados relativos ao perfil geral do utilizador. A sua funcionalidade restringe-se a ser o elemento base que contém todos os outros elementos que com informação sobre o nome, data de nascimento, sexo, profissão, entre outros, relativos ao novo utilizador, não possuindo qualquer atributo específico.

Elemento *profession* Este elemento obrigatório identifica, através do seu único atributo (também obrigatório) “value”, a profissão ou ocupação do novo utilizador. Este elemento e respectivo atributo tem como funcionalidade fornecer informação para a construção inicial do perfil do utilizador.

Elemento *gender* Este elemento obrigatório identifica, através do seu único atributo (também obrigatório) “value”, o sexo do novo utilizador. Este elemento e respectivo atributo tem como funcionalidade fornecer informação para a construção inicial do perfil do utilizador.

Elemento *relationshipstatus* Este elemento obrigatório identifica, através do seu único atributo (também obrigatório) “value”, o actual estado civil do novo utilizador (casado, solteiro, divorciado, etc.). Este elemento e respectivo atributo tem como funcionalidade fornecer informação para a construção inicial do perfil do utilizador.

Elemento *fullname* Este elemento obrigatório identifica, através do seu único atributo (também obrigatório) “value”, o nome completo do novo utilizador. Este elemento e respectivo atributo tem como funcionalidade fornecer informação para a construção inicial do perfil do utilizador.

Elemento *displayname* Este elemento obrigatório identifica, através do seu único atrib-

uto (também obrigatório) “value”, o nome que o novo utilizador deseja que seja mostrado na aplicação, que poderá ser igual ou diferente do *user name*, e deverá ser mais curto que o seu nome completo. Este elemento e respectivo atributo tem como funcionalidade fornecer informação para a construção inicial do perfil do utilizador.

Elemento *birthdate* Este elemento obrigatório identifica, através do seu único atributo (também obrigatório) “value”, a data de nascimento do novo utilizador. Este elemento e respectivo atributo tem como funcionalidade fornecer informação para a construção inicial do perfil do utilizador e personalização do mesmo.

Elemento *email* Este elemento obrigatório identifica, através do seu atributo obrigatório único “value”, o endereço de e-mail do novo utilizador. Devido ao sistema poder fazer uso desta forma de comunicação para múltiplos fins, a presença e correcção deste atributo é muito importante. Este elemento e respectivo atributo tem como funcionalidade fornecer informação para a construção inicial do perfil do utilizador.

Elemento *photopath* Este elemento obrigatório identifica, através do seu único atributo (também obrigatório) “value”, a fotografia ou *avatar* do novo utilizador. Cada utilizador só poderá exibir uma imagem associada à sua conta. Esta pode estar localizada no servidor IPTV ou em qualquer outro recurso acessível através da Internet, e por isso a utilização deste atributo. Caso o valor deste campo seja vazio ou inválido, cabe ao cliente IPTV definir uma política para contornar este problema, como por exemplo considerar uma imagem por defeito. Este elemento e respectivo atributo tem como funcionalidade permitir uma maior personalização do sistema.

Elemento *languages* Este elemento XML opcional é o elemento aglutinador da declaração das múltiplas línguas que o utilizador compreende. A sua funcionalidade restringe-se a ser o elemento base que contém a especificação dos idiomas compreendidos pelo novo utilizador não possuindo qualquer atributo específico. Este é opcional para que a recolha da informação inicial do registo não seja demasiado exaustiva e extensa, sendo que caso não seja especificado será assumida uma língua principal por defeito

que dependente da implementação do sistema. No caso da implementação actual, o idioma por defeito é o Inglês.

Elemento *lang* Este elemento XML obrigatório (no caso do seu elemento agregador *languages* existir) especifica um idioma compreendido segundo um determinado nível pelo novo utilizador. Permite a definição de múltiplos idiomas e a marcação de uma ou mais destes como línguas principais. É também possível a definição de um nível de compreensão associado a cada idioma. As principais razões da existência deste tipo de elementos deve-se à possibilidade de personalização linguística do cliente IPTV e também possibilitar que conteúdos multimédia nos idiomas compreendidos possuam uma maior relevância aquando os processos de sugestão e/ou apresentação dos mesmos. Os atributos deste elemento e sua semântica são:

name Atributo obrigatório que contém o nome do idioma definido através dos códigos alfabéticos internacionais (*pt-Portuguese*, *en-English*, etc. – ver página [http://msdn.microsoft.com/en-us/library/ms533052\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms533052(VS.85).aspx)). Permite a distinção de uma forma padrão e inequívoca dos múltiplos idiomas suportados;

rate Atributo obrigatório que define um nível de domínio/compreensão da linguagem no intervalo [0.0, 1.0] (valor normalizado de uma percentagem irracional de valor mínimo 0.0% e valor máximo 100.0%), em que o valor mínimo define uma compreensão mínima/nula, e o valor máximo define uma compreensão total/máxima. Este atributo permite estabelecer uma hierarquia entre os idiomas compreendidos pelo utilizador, determinada pelo grau de domínio deste;

ismain Atributo obrigatório que indica se o idioma em causa é considerado uma “língua materna” para o utilizador. Quando uma língua é assinalada como “materna”, é considerada mais preponderante que uma outra não “materna” com um nível de compreensão semelhante. É de notar que múltiplos idiomas podem ser considerados maternos.

Elemento *userinterestareas* Este elemento XML opcional é o elemento aglutinador da

declaração das múltiplas áreas de interesse que o utilizador pode assinalar aquando o seu registo no sistema. A sua funcionalidade restringe-se a ser o elemento base que contém a especificação das múltiplas áreas de interesse preferidas pelo novo utilizador. É de notar que estas áreas de interesse são aplicadas por defeito ao modo de visualização (*watching mode*) *alone*. As áreas de interesse referentes os restantes modos de visualização bem como ao modo *alone* podem ser configuradas manualmente após a criação da conta no sistema. Este elemento é opcional para que a recolha da informação inicial aquando o registo não seja demasiado exaustiva e extensa. Caso este elemento não seja especificado, será assumido inicialmente que o utilizador tem igual preferência por qualquer área de interesse existente no sistema, considerando assim que a “raíz” 0 do grafo possui preferência 0.5.

Elemento *area* Este elemento XML obrigatório (no caso do seu elemento agregador *user-interestareas* existir) determina a definição inicial da preferência relativa a uma área de interesse específica. Permite a definição do interesse relativo a múltiplas áreas de interesse, sejam elas preferidas ou detestadas. Estes elementos existem com o intuito de possibilitar ao utilizador definir uma base inicial mais restrita para a construção dinâmica do seu perfil. Os atributos deste elemento e sua semântica são:

id Atributo obrigatório que contém o identificador único da área de interesse seguindo uma metodologia organizada em árvore com uma sintaxe inspirada na sintaxe das MIBs (protocolo SNMP) dos equipamentos de rede (ver exemplo na página <http://www.geocities.com/SiliconValley/Vista/5635/cap6.html>). Este identificador por si só traduz uma área de interesse específica e também determina quais as super-áreas das quais deriva;

rate Atributo obrigatório que define um nível de preferência que o utilizador tem pela área de interesse em questão. Este valor situa-se no intervalo [0.0, 1.0] (valor normalizado de uma percentagem irracional de valor mínimo 0.0% e valor máximo 100.0%), em que o valor mínimo define uma preferência mínima/nula, e o valor máximo define uma preferência máxima/total. Este atributo permite

definir as preferências iniciais do utilizador de uma forma numérica ponderada, situando o utilizador numa sub-região do universo de todas as áreas de interesse disponíveis.

Um exemplo de instanciação deste modelo de mensagens numa mensagem XML (com todos os elementos e atributos possíveis) que requer uma operação de registo de um novo utilizador, pode ser visto seguidamente.

```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!--
3 XML sent by the IPTV Client to the IPTV Server with a request for a new user register.
4 This message contain all the data for all the fields needed for the new user creation.
5 -->
6 <communication xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
7   <command name="NewUserRequest" />
8   <userlogin name="pmfm" />
9
10  <userdataform>
11    <profession value="engineer" />
12    <gender value="male" />
13    <relationshipstatus value="single" />
14    <fullname value="Paulo Martins Ferreira Marcelo" />
15    <displayname value="Paulo Martins" />
16    <birthdate value="1976-10-22 01:00:00" />
17    <email value="paulo.marcelo@gmail.com" />
18    <photopath value="http://www.pmf.com/myPhoto.jpg" />
19    <languages>
20      <lang name="portuguese" ismain="1" rate="0.95" />
21      <lang name="english" ismain="0" rate="0.8" />
22      <lang name="french" ismain="0" rate="0.3" />
23      <lang name="spanish" ismain="0" rate="0.6" />
24    </languages>
25  </userdataform>
26
27  <userinterestareas>
28    <area id="0.1" rate="1.0" />
29    <area id="0.2" rate="0.2323" />
30    <area id="0.2.1" rate="0.3" />
31    <area id="0.2.3" rate="0.5" />
32    <area id="0.3" rate="0.123" />
33    <area id="0.3.1" rate="0.22" />
34    <area id="0.4" rate="0.0" />
35  </userinterestareas>
36 </communication>

```

B.4 Mensagem *New User Validation*

Este tipo de mensagem permite que um utilizador previamente registado efectue a validação e activação da sua nova conta. Somente deste modo é possível iniciar uma sessão no servidor IPTV pela primeira vez com essa mesma conta. O processo que permite ao utilizador validar a sua conta depende da implementação do cliente IPTV, sendo que passa necessariamente pela mudança ou verificação da *password* atribuída ou especificada aquando o registo do utilizador. Esta pequena operação de validação da conta e possível mudança da *password* subjacente, confere um pouco mais de segurança na criação de contas no sistema. O servidor, ao receber uma mensagem deste tipo, irá responder sempre com uma mensagem *Command Response* (secção B.12) cujo conteúdo indica o sucesso ou falha da operação, possuindo uma descrição dos possíveis erros ou problemas caso a operação não seja bem sucedida. A definição formal em XSD para este tipo de mensagem encontra-se seguidamente.

```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!--
3 XML Schema for the 'New User Validation Request' messages sent by the IPTV Client to
4 the IPTV Server.
5 -->
6 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
7   <xs:element name="uservalidation">
8     <xs:complexType>
9       <xs:attribute name="useroldpass" type="xs:string" use="required" />
10      <xs:attribute name="usernewpass" type="xs:string" use="required" />
11      <xs:attribute name="username" type="xs:string" use="required" />
12    </xs:complexType>
13  </xs:element>
14  <xs:element name="communication">
15    <xs:complexType>
16      <xs:sequence>
17        <xs:element ref="command" />
18        <xs:element ref="uservalidation" />
19      </xs:sequence>
20    </xs:complexType>
21  </xs:element>
22  <xs:element name="command">
23    <xs:complexType>
24      <xs:attribute name="name" use="required">
25        <xs:simpleType>
26          <xs:restriction base="xs:string">
27            <xs:enumeration value="NewUserValidationRequest" />

```

```

27         </xs:restriction>
28     </xs:simpleType>
29 </xs:attribute>
30 </xs:complexType>
31 </xs:element>
32 </xs:schema>

```

Esta definição pode ainda ser visualizada de uma forma mais intuitiva e compacta através do diagrama da figura B.4.

A tabela B.4 descreve, para cada elemento da mensagem XML, o nome dos seus atributos bem como o tipo de dados permitido, tipo de utilização (obrigatória/opcional) e valores por defeito.

Descrita toda a sintaxe deste tipo de mensagem enviada pelo cliente para o servidor IPTV com um pedido de validação da conta criada previamente pelo utilizador no sistema, segue-se agora a descrição semântica e funcional de cada elemento XML e dos seus atributos (quando estes existem).

Elemento *communication* Este elemento XML é obrigatório, sendo mesmo a “raíz” da árvore XML definida pela mensagem. A sua funcionalidade restringe-se ao facto de ser o elemento base que contém todos os outros semanticamente mais importantes, não possuindo atributos específicos.

Elemento *command* Este elemento obrigatório define e identifica o tipo de mensagem de que se trata. Para o caso de mensagens do tipo *New User Validation*, este elemento

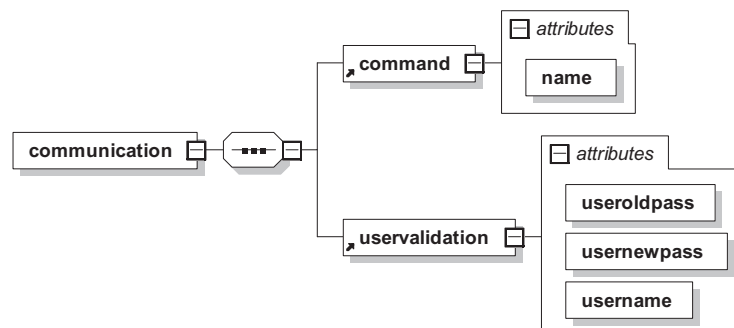


Figura B.4: Diagrama das relações entre os elementos e atributos para mensagens do tipo *New User Validation*.

Elemento	Nome do Atributo	Tipo do Atributo	Uso do Atributo	Default do Atributo
communication	–	–	–	–
command	<i>name</i>	string	obrigatório	–
uservalidation	<i>username</i>	string	obrigatório	–
	<i>useroldpass</i>	string	obrigatório	–
	<i>usernewpass</i>	string	obrigatório	–

Tabela B.4: Tabela dos elementos e atributos para mensagens do tipo *New User Validation*

possui sempre o seu único atributo (que também é obrigatório) “name” com o valor igual a “NewUserValidationRequest”, que funciona como identificador deste tipo de mensagens.

Elemento *uservalidation* Este elemento XML obrigatório contém nos seus atributos a informação útil relativa ao pedido de validação e activação da conta do utilizador. Os atributos deste elemento e sua semântica são:

username Atributo obrigatório que contém o nome do utilizador relativo à sua conta no sistema (*user name*). Este atributo tem como funcionalidade permitir a identificação e autenticação do utilizador em conjunto com o atributo *useroldpass*;

useroldpass Atributo obrigatório que contém a *password* do utilizador especificada antes da validação codificada no formato MD5. Esta corresponde à *password* definida durante o processo de registo do novo utilizador, seja ela uma *password* especificada pelo mesmo ou auto-gerada pelo sistema. Este atributo tem como funcionalidade permitir a identificação e autenticação do utilizador em conjunto com o atributo *username*;

usernewpass Atributo obrigatório que contém a nova *password* (em MD5) que o utilizador deseja definir como sendo a sua nova palavra-chave de acesso ao sistema após a validação da conta. Esta poderá ser igual ou diferente da *password* definida anteriormente. A mudança de *password* possui mais sentido quando a anterior foi definida com recurso à geração automática durante o registo do

utilizador. Independentemente disso, este atributo terá de existir sempre no elemento *uservalidation* e deverá possuir um valor válido.

Um exemplo de instanciação deste modelo de mensagens numa mensagem XML que requer uma operação de validação da conta de um utilizador, pode ser visto de seguida.

```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!--
3 XML message to be sent by the IPTV Client to the IPTV Server with a request for a new
  user validation previously created and registered specifically with the 'username'
  and 'useroldpass'.
4 When a new account is created, it is set to default state 'wating' for activation
  which does not allow its use.
5 Once validated, the user account in question will have as password the new password
  'usernewpass' and the account status 'active '.
6 -->
7 <communication xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
8   <command name="NewUserValidationRequest"/>
9   <uservalidation username="joaoffr" useroldpass="512d495f887d7d138f2ef132f59cf27c"
    usernewpass="6d6b9d0cfe9f35b4d8ba5874931d914d"/>
10 </communication>

```

B.5 Mensagem *Session Keep Alive Request*

Este tipo de mensagem permite que um utilizador se mantenha ligado ao sistema mesmo quando está aparentemente inactivo no mesmo devido a estar a assistir a um conteúdo. Neste caso, é o cliente IPTV que terá de deliberar quando esta mensagem será utilizada. O servidor irá responder sempre com uma mensagem *Command Response* (secção B.12) cujo conteúdo indica o sucesso ou falha da operação. A definição formal em XSD para este tipo de mensagem pode encontrar-se seguidamente.

```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!-- edited with XMLSpy v2008 rel. 2 sp2 (http://www.altova.com) by SHOCK (SHOCK) -->
3 <!--
4 XML schema to messages sent by the IPTV Client to the IPTV Server when the client want
  to keep the user session alive.
5 -->
6 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
7   <xs:element name="user">
8     <xs:complexType>
9       <xs:attribute name="sessionid" type="xs:long" use="required"/>

```

```

10     <xs:attribute name="sessioncode" use="required">
11         <xs:simpleType>
12             <xs:restriction base="xs:string" />
13         </xs:simpleType>
14     </xs:attribute>
15 </xs:complexType>
16 </xs:element>
17 <xs:element name="communication">
18     <xs:complexType>
19         <xs:sequence>
20             <xs:element ref="command" />
21             <xs:element ref="user" />
22         </xs:sequence>
23     </xs:complexType>
24 </xs:element>
25 <xs:element name="command">
26     <xs:complexType>
27         <xs:attribute name="name" use="required">
28             <xs:simpleType>
29                 <xs:restriction base="xs:string">
30                     <xs:enumeration value="KeepAliveSessionRequest" />
31                 </xs:restriction>
32             </xs:simpleType>
33         </xs:attribute>
34     </xs:complexType>
35 </xs:element>
36 </xs:schema>

```

Esta definição pode ainda ser visualizada de uma forma mais intuitiva e compacta através do diagrama da figura B.5.

A tabela B.5 descreve, para cada elemento da mensagem XML, o nome dos seus atributos bem como o tipo de dados permitido, tipo de utilização (obrigatória/opcional) e valores

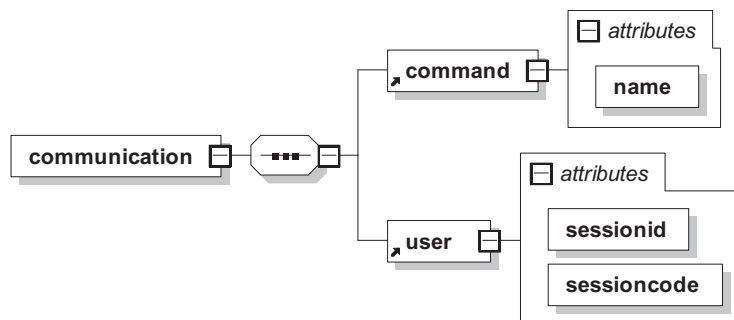


Figura B.5: Diagrama das relações entre os elementos e atributos para mensagens do tipo *Session Keep Alive*.

por defeito.

Descrita toda a sintaxe deste tipo de mensagem enviada pelo cliente para o servidor IPTV, segue-se a descrição semântica e funcional de cada elemento XML e dos seus atributos.

Elemento *communication* Este elemento XML é obrigatório e é a “raíz” da árvore XML definida pela mensagem. A sua funcionalidade restringe-se a ser o elemento base que contém todos os outros semanticamente mais importantes e não possui quaisquer atributos específicos.

Elemento *command* Este elemento obrigatório define e identifica o tipo de mensagem de que se trata. Para o caso de mensagens *Session Keep Alive Request* este elemento possui sempre o seu único atributo (que também é obrigatório) “name” com o valor igual a “KeepAliveSessionRequest”, que é o identificador deste tipo de mensagens.

Elemento *user* Este elemento XML obrigatório contém nos seus atributos a informação útil que será utilizada para manter a sessão aberta. Os atributos deste elemento e sua semântica são:

sessionid Atributo obrigatório que contém um identificador numérico que permite relacionar o utilizador com a sua actual sessão activa no sistema. A utilização deste atributo deve-se não só à necessidade de identificação do utilizador que requer a operação mas também ao estabelecimento de políticas de segurança e autenticação;

Elemento	Nome do Atributo	Tipo do Atributo	Uso do Atributo	Default do Atributo
communication	–	–	–	–
command	<i>name</i>	string	obrigatório	–
user	<i>sessionid</i>	long	obrigatório	–
	<i>sessioncode</i>	string	obrigatório	–

Tabela B.5: Tabela dos elementos e atributos para mensagens do tipo *Session Keep Alive*

sessioncode Atributo obrigatório que contém uma cadeia de caracteres que permite ao sistema verificar a autenticidade da operação requisitada pela mensagem XML.

Um exemplo de instanciação deste modelo de mensagens numa mensagem XML pode ser visto de seguida.

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!--
3 XML to be sent by the IPTV Client to the IPTV Server when the client want to keep the
4 user session alive
5 (for exemple, if user is seeing a one hour movie).
6 NOTE: 'sessioncode' should not have the 'sessionid' at the end. This should have
7 exactly 32 characters.
8 -->
9 <communication xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
10 <command name="KeepAliveSessionRequest" />
11 <user sessionid="147" sessioncode="1fea143b66b02696ead45ffacb7923ab" />
12 </communication>
```

B.6 Mensagem *New Link Request*

Esta mensagem permite que um cliente IPTV peça o próximo *link* para um conteúdo sugerido pelo próprio sistema ou por outro utilizado, relativo ao utilizador autenticado actualmente no cliente IPTV. Este tipo de mensagens implementa, de certo modo, a função de *zapping* de conteúdo sobre os conteúdos sugeridos, controlados pelo sistema de sugestão automática. O servidor ao receber uma mensagem deste tipo irá responder com uma mensagem *New Link Response* (secção B.7) no caso do pedido ter sido efectuado com sucesso. Caso o pedido recebido pelo servidor IPTV esteja mal formulado, não exista na base de dados nenhum conteúdo actualmente sugerido para o utilizador em questão nas suas actuais condições ou exista qualquer outro problema, é enviada uma mensagem *Command Response* (secção B.12). O conteúdo desta mensagem enviada em situações anómalas é a descrição das possíveis falhas ou erros ocorridos.

Seguidamente encontra-se a definição formal em XSD para este tipo de mensagem.

```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!--
3 XML Schema for the 'New Link Request' messages sent by the IPTV Client to the IPTV
  Server.
4 -->
5 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
6   <xs:element name="user">
7     <xs:complexType>
8       <xs:attribute name="sessionid" type="xs:long" use="required" />
9       <xs:attribute name="sessioncode" type="xs:string" use="required" />
10    </xs:complexType>
11  </xs:element>
12  <xs:element name="previouslink">
13    <xs:complexType>
14      <xs:attribute name="playedtime" type="xs:long" use="required" />
15      <xs:attribute name="playedperc" type="xs:double" use="required" />
16      <xs:attribute name="modename" type="xs:string" use="required" />
17      <xs:attribute name="id" type="xs:long" use="required" />
18      <xs:attribute name="feedbackname" type="xs:string" use="required" />
19    </xs:complexType>
20  </xs:element>
21  <xs:element name="communication">
22    <xs:complexType>
23      <xs:sequence>
24        <xs:element ref="command" />
25        <xs:element ref="user" />
26        <xs:element ref="previouslink" minOccurs="0" />
27      </xs:sequence>
28    </xs:complexType>
29  </xs:element>
30  <xs:element name="command">
31    <xs:complexType>
32      <xs:attribute name="name" use="required">
33        <xs:simpleType>
34          <xs:restriction base="xs:string">
35            <xs:enumeration value="NewLinkRequest" />
36          </xs:restriction>
37        </xs:simpleType>
38      </xs:attribute>
39    </xs:complexType>
40  </xs:element>
41 </xs:schema>

```

Esta definição pode ainda ser visualizada de uma forma mais intuitiva e compacta através do diagrama da figura B.6.

A tabela B.6 descreve, para cada elemento da mensagem XML, o nome dos seus atributos bem como o tipo de dados permitido, tipo de utilização (obrigatória/opcional) e valores

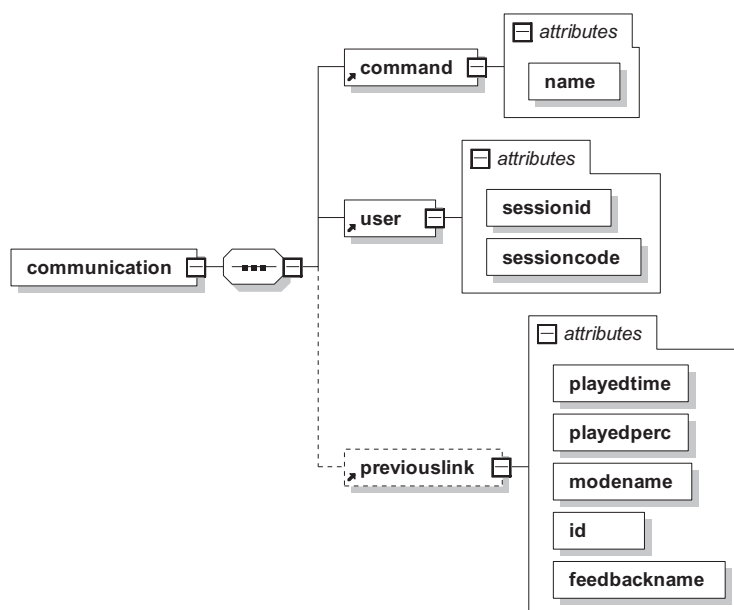


Figura B.6: Diagrama das relações entre os elementos e atributos para mensagens do tipo *New Link Request*.

por defeito.

Descrita toda a sintaxe deste tipo de mensagem enviada pelo cliente para o servidor IPTV com um pedido de um novo conteúdo sugerido, segue-se agora a descrição semântica e funcional de cada elemento XML e seus atributos (quando estes existem).

Elemento *communication* Este elemento XML é obrigatório e é a “raíz” da árvore XML

Elemento	Nome do Atributo	Tipo do Atributo	Uso do Atributo	Default do Atributo
communication	–	–	–	–
command	<i>name</i>	string	obrigatório	–
user	<i>sessionid</i> <i>sessioncode</i>	long string	obrigatório obrigatório	– –
previouslink	<i>id</i> <i>modename</i> <i>feedbackname</i> <i>playedperc</i> <i>playedtime</i>	long string double double long	obrigatório obrigatório obrigatório obrigatório obrigatório	– – – – –

Tabela B.6: Tabela dos elementos e atributos para mensagens do tipo *New Link Request*

definida pela mensagem. A sua funcionalidade restringe-se a ser o elemento base que contém todos os outros semanticamente mais importantes e não possui atributos específicos.

Elemento *command* Este elemento obrigatório define e identifica o tipo de mensagem de que se trata. Para o caso de mensagens *New Link Request* este elemento possui sempre o seu único atributo (que também é obrigatório) “name” com o valor igual a “NewLinkRequest”, que é o identificador deste tipo de mensagens.

Elemento *user* Este elemento XML obrigatório contém nos seus atributos a informação que será usada para autenticar este pedido (operação) e associar o mesmo a um utilizador com sessão previamente iniciada no sistema. Os atributos deste elemento e sua semântica são:

sessionid Atributo obrigatório que contém um identificador numérico que permite relacionar o utilizador com a sua actual sessão activa no sistema. A necessidade deste atributo tem como objectivo não só a identificação do utilizador que requer a operação mas também o estabelecimento de políticas de segurança e autenticação;

sessioncode Atributo obrigatório que contém uma cadeia de caracteres que permite ao sistema verificar a autenticidade da operação requisitada pela mensagem XML. Este campo em conjunto com o *sessionid* permite a identificação da sessão do utilizador no sistema (se esta existir) e permite também o estabelecimento de políticas de segurança na realização das operações.

Elemento *previouslink* Este elemento XML opcional permite que o cliente IPTV envie nesta mensagem não só um pedido de um novo conteúdo multimédia sugerido mas também a classificação *feedback* relativa a um conteúdo acedido anteriormente. Os atributos deste elemento e sua semântica são:

id Atributo obrigatório que contém um identificador numérico que permite identificar o conteúdo multimédia inequivocamente na base de dados. Este atributo

permite efectuar o relacionamento entre a classificação do utilizador (*feedback*) e o conteúdo multimédia respectivo;

modename Atributo obrigatório que contém o identificador do modo de visualização que o utilizador usou para assistir ao conteúdo em questão. Este atributo permite estabelecer a relação entre o *feedback* do utilizador, o conteúdo assistido e o modo de visualização utilizado, contribuindo para a construção do perfil do utilizador;

feedbackname Atributo obrigatório que contém o identificador relativo à classificação (*feedback*) dada pelo utilizador ao conteúdo multimédia assistido anteriormente. A classificação é efectuada segundo níveis específicos que traduzem as preferências do utilizador pelos conteúdos assistidos;

playedperc Atributo obrigatório que identifica a percentagem (no intervalo [0.0, 1.0]) do conteúdo multimédia em questão reproduzida pelo utilizador. Este atributo é utilizado pelo sistema para recolha de informação mais específica sobre a experiência de utilização do serviço de IPTV e construção do perfil do utilizador;

playedtime Atributo obrigatório que possui a quantidade de tempo (em milisegundos) reproduzido pelo utilizador relativo ao conteúdo em questão. Este atributo é utilizado pelo sistema para recolha de informação mais específica sobre a experiência de utilização do serviço de IPTV e construção do perfil do utilizador.

Um exemplo de instanciação deste modelo de mensagens numa mensagem XML (com todos os elementos e atributos possíveis), pode ser visto seguidamente.


```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!--
3 XML sent by the IPTV Client to the IPTV Server with a request for a new content link .
4 Alternatively , it is also possible to send a feedback from a previous content link
   inside this message .
5 NOTE: 'Time' in the format 'hh:mm:ss [.f...]' ;
6       'sessioncode' should not have the 'sessionid' at the end . This should have exactly
       32 characters ;
7       'playedperc' in the interval [0.0 , 1.0] ;
8       'playedtime' in milliseconds .
9 -->
10 <communication xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
11   <command name="NewLinkRequest" />
12   <user sessionid="10" sessioncode="1fea143b66b02696ead45ffacb792347" />
13   <previouslink id="3" modename="allone" feedbackname="I like it" playedperc="0.35"
       playedtime="21475" />
14 </communication>

```

B.7 Mensagem *New Link Response*

Esta mensagem é a resposta enviada pelo servidor IPTV aquando a recepção de uma mensagem *New Link Request* (secção B.6) bem estruturada e caso exista na base de dados pelo menos um conteúdo sugerido ao utilizador em questão. Em vez desta mensagem, o servidor poderá responder a uma operação *New Link Request* (secção B.6) com uma mensagem *Command Response* (secção B.12) caso o pedido recebido pelo servidor IPTV esteja mal formulado, não exista na base de dados nenhum conteúdo actualmente sugerido para o utilizador em questão ou exista qualquer outro problema.

A definição formal em XSD para mensagem do tipo *New Link Response* pode ser encontrada de seguida.

```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!--
3 XML Schema for the 'New Link Response' messages sent by the IPTV Server to the IPTV
   Client .
4 -->
5 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
6   <xs:element name="provider">
7     <xs:complexType>
8       <xs:attribute name="name" type="xs:string" use="required" />
9       <xs:attribute name="id" type="xs:long" use="required" />
10      <xs:attribute name="description" type="xs:string" use="required" />

```

```

11     <xs:attribute name="address" type="xs:anyURI" use="required" />
12     <xs:attribute name="active" type="xs:boolean" use="required" />
13 </xs:complexType>
14 </xs:element>
15 <xs:element name="newlink">
16   <xs:complexType>
17     <xs:sequence>
18       <xs:element ref="provider" />
19       <xs:element ref="author" />
20       <xs:element ref="contentformat" />
21     </xs:sequence>
22     <xs:attribute name="title" type="xs:string" use="required" />
23     <xs:attribute name="path" type="xs:anyURI" use="required" />
24     <xs:attribute name="id" type="xs:long" use="required" />
25     <xs:attribute name="description" type="xs:string" use="required" />
26     <xs:attribute name="date" type="xs:string" use="required" />
27     <xs:attribute name="active" type="xs:boolean" use="required" />
28   </xs:complexType>
29 </xs:element>
30 <xs:element name="contentformat">
31   <xs:complexType>
32     <xs:attribute name="name" type="xs:string" use="required" />
33     <xs:attribute name="id" type="xs:long" use="required" />
34     <xs:attribute name="description" type="xs:string" use="required" />
35     <xs:attribute name="typeid" type="xs:long" use="required" />
36     <xs:attribute name="typename" type="xs:string" use="required" />
37     <xs:attribute name="typedescription" type="xs:string" use="required" />
38   </xs:complexType>
39 </xs:element>
40 <xs:element name="communication">
41   <xs:complexType>
42     <xs:sequence>
43       <xs:element ref="command" />
44       <xs:element ref="newlink" />
45     </xs:sequence>
46   </xs:complexType>
47 </xs:element>
48 <xs:element name="command">
49   <xs:complexType>
50     <xs:attribute name="name" use="required">
51       <xs:simpleType>
52         <xs:restriction base="xs:string">
53           <xs:enumeration value="NewLinkResponse" />
54         </xs:restriction>
55       </xs:simpleType>
56     </xs:attribute>
57   </xs:complexType>
58 </xs:element>
59 <xs:element name="author">
60   <xs:complexType>
61     <xs:attribute name="name" type="xs:string" use="required" />

```

```

62     <xs:attribute name="id" type="xs:long" use="required" />
63     <xs:attribute name="description" type="xs:string" use="required" />
64     <xs:attribute name="address" type="xs:anyURI" use="required" />
65     <xs:attribute name="active" type="xs:boolean" use="required" />
66   </xs:complexType>
67 </xs:element>
68 </xs:schema>

```

Esta definição pode ainda ser visualizada de uma forma mais intuitiva e compacta através do diagrama da figura B.7.

A tabela B.7 descreve, para cada elemento da mensagem XML, o nome dos seus atributos bem como o tipo de dados permitido, tipo de utilização (obrigatória/opcional) e valores por defeito.

Descrita toda a sintaxe deste tipo de mensagem enviada pelo servidor para um cliente IPTV com a resposta a um pedido de um novo *link* referente a um conteúdo sugerido, segue-se a descrição semântica e funcional de cada elemento XML e seus atributos (quando estes existem).

Elemento *communication* Este elemento XML é obrigatório e é a “raíz” da árvore XML definida pela mensagem. A sua funcionalidade restringe-se a ser o elemento base que contém todos os outros semanticamente mais importantes e não possui atributos específicos.

Elemento *command* Este elemento obrigatório define e identifica o tipo de mensagem de que se trata. Para o caso de mensagens *New Link Response* este elemento possui sempre o seu único atributo (que também é obrigatório) “name” com o valor igual a “NewLinkResponse”, que é o identificador deste tipo de mensagens.

Elemento *newlink* Este elemento XML obrigatório contém nos seus atributos a informação referente ao novo *link* para o respectivo conteúdo sugerido. Os seus atributos e sua semântica são:

id Atributo obrigatório que contém um identificador numérico que permite referenciar o conteúdo inequivocamente na base de dados;

Elemento	Nome do Atributo	Tipo do Atributo	Uso do Atributo	Default do Atributo
communication	–	–	–	–
command	<i>name</i>	string	obrigatório	–
newlink	<i>id</i>	long	obrigatório	–
	<i>title</i>	string	obrigatório	–
	<i>date</i>	string	obrigatório	–
	<i>path</i>	string	obrigatório	–
	<i>description</i>	string	obrigatório	–
provider	<i>active</i>	byte	obrigatório	–
	<i>id</i>	long	obrigatório	–
	<i>name</i>	string	obrigatório	–
	<i>address</i>	string	obrigatório	–
author	<i>description</i>	string	obrigatório	–
	<i>active</i>	byte	obrigatório	–
	<i>name</i>	string	obrigatório	–
	<i>address</i>	string	obrigatório	–
contentformat	<i>id</i>	long	obrigatório	–
	<i>name</i>	string	obrigatório	–
	<i>description</i>	string	obrigatório	–
	<i>typeid</i>	long	obrigatório	–
	<i>typename</i>	string	obrigatório	–
	<i>typedescription</i>	string	obrigatório	–

Tabela B.7: Tabela dos elementos e atributos para mensagens do tipo *New Link Response*

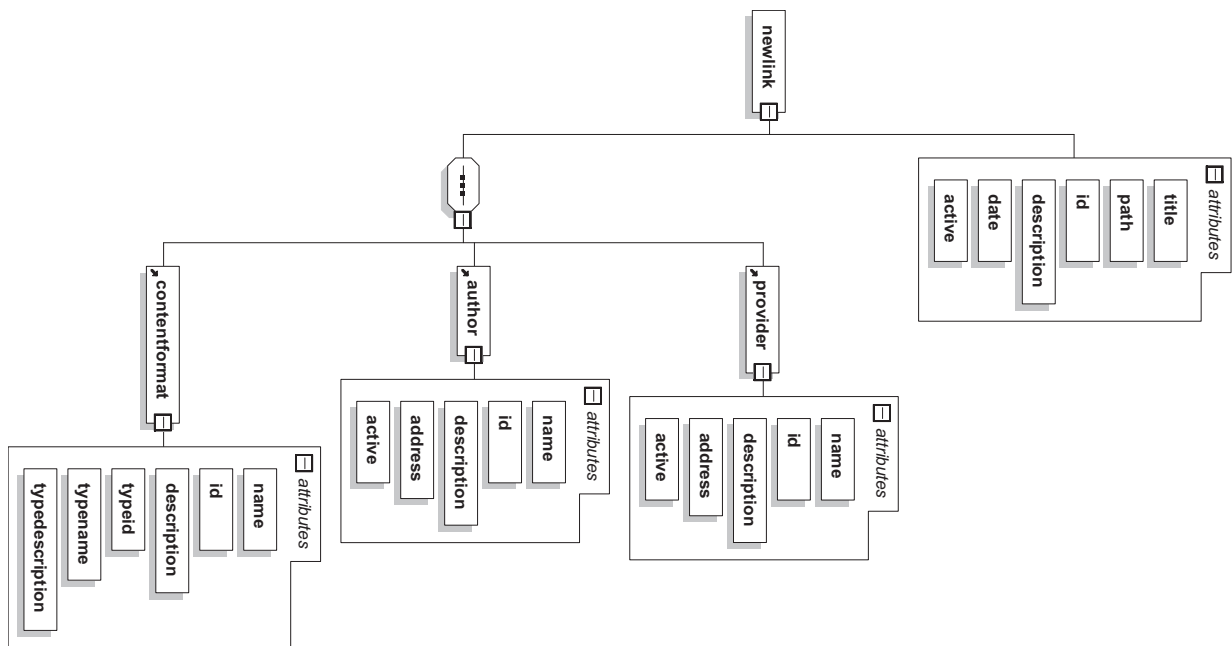


Figura B.7: Diagrama das relações entre os elementos e atributos para mensagens do tipo *New Link Response*.

title Atributo obrigatório que contém o título do respectivo conteúdo proveniente da fonte que disponibiliza o mesmo;

date Atributo obrigatório que possui a data de criação do conteúdo proveniente da fonte que disponibiliza o mesmo;

path Atributo obrigatório que contém o URI (o elemento muitas vezes referido nesta dissertação como *link*) de acesso ao conteúdo multimédia disponibilizado pelo provedor do conteúdo;

description Atributo obrigatório que possui uma pequena descrição do respectivo conteúdo multimédia proveniente da fonte que disponibiliza o mesmo;

active Atributo obrigatório que indica se o respectivo conteúdo se encontra activo (disponível) – valor igual a “1” – ou inactivo (indisponível) – valor igual a “0”. Como será de esperar, o servidor só irá enviar mensagens deste tipo com este atributo do elemento *newlink* com valor igual a “1”, pois o sistema não oferece aos seus utilizadores *links* para conteúdos indisponíveis à partida.

Elemento *provider* Este elemento XML obrigatório possui nos seus atributos informações acerca da entidade que disponibiliza o acesso ao respectivo conteúdo. Os atributos deste elemento e sua semântica são:

id Atributo obrigatório que contém um identificador numérico que permite referenciar o provedor do conteúdo multimédia inequivocamente na base de dados do sistema. Este atributo permite o relacionamento entre o conteúdo e a entidade que o disponibiliza;

name Atributo obrigatório que possui o nome da entidade que disponibiliza o acesso ao conteúdo;

address Atributo obrigatório que contém o URI da entidade que disponibiliza o conteúdo;

description Atributo obrigatório que contém uma pequena descrição relativa ao provedor do conteúdo;

active Atributo obrigatório que informa se a entidade que disponibiliza o conteúdo ainda existe e/ou está acessível e disponível (valor igual a “1”) ou, pelo contrário, já não existe ou não se encontra acessível/disponível (valor igual a “0”). Como será de esperar, o servidor só irá enviar mensagens deste tipo com este atributo com valor igual a “1”, pois o sistema não oferece aos seus utilizadores *links* para conteúdos cuja entidade que os disponibiliza não exista ou não esteja acessível/disponível à partida.

Elemento *author* Este elemento XML obrigatório possui nos seus atributos informações acerca do autor do respectivo conteúdo. Os atributos deste elemento e sua semântica são:

id Atributo obrigatório que contém um identificador numérico que permite referenciar o autor do conteúdo multimédia inequivocamente na base de dados do sistema. Este atributo permite o relacionamento entre o conteúdo e o seu autor (que poderá ser diferente da entidade que disponibiliza o conteúdo);

name Atributo obrigatório que possui o nome do autor do conteúdo;

address Atributo obrigatório que contém o URI do autor do conteúdo. Caso o autor não possua uma *web page* ou algo do género que possa ser referenciado neste atributo, este ficará em branco;

description Atributo obrigatório que possui uma pequena descrição das características do autor do conteúdo multimédia respectivo;

active Atributo obrigatório que informa se o autor do conteúdo ainda está activo no sistema. Caso o autor ainda esteja activo, o valor deste campo será “1” (verdadeiro). Caso contrário, será “0” (falso). O servidor irá enviar mensagens deste tipo com este atributo possuindo um valor que possivelmente será irrelevante para o cliente IPTV, possuindo somente carácter informativo.

Elemento *contentformat* Este elemento XML obrigatório contém nos seus atributos a informação referente ao formato multimédia do conteúdo respectivo no que diz respeito à sua forma de codificação áudio-visual. Para além disso, possui ainda informação referente ao tipo de fluxo, ou seja, áudio “gravado”, vídeo com áudio “gravado” ou conteúdo em “tempo real”. Os seus atributos e sua semântica são:

id Atributo obrigatório que contém um identificador numérico que permite referenciar o formato (codificação) do conteúdo inequivocamente na base de dados do servidor IPTV;

name Atributo obrigatório que contém o nome do formato dado pela extensão com a sintaxe “.<<*nome da extensão*>>”;

description Atributo obrigatório que contém uma pequena descrição referente ao respectivo formato do conteúdo multimédia;

typeid Atributo obrigatório que contém um identificador numérico que permite referenciar de forma inequívoca o tipo de fluxo a que este se refere;

typename Atributo obrigatório que possui o nome do tipo de fluxo multimédia (áudio “gravado”, vídeo com áudio “gravado”, conteúdo em “tempo real”, etc.);

typedescription Atributo obrigatório que possui uma breve descrição sobre o tipo de fluxo multimédia em questão.

Um exemplo de instanciação deste modelo de mensagens numa mensagem XML que permite o envio para um cliente IPTV de um novo *link* de um conteúdo sugerido, pode ser visto seguidamente.

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!--
3 XML sent by the IPTV Server for the IPTV Client with a new content link and all data
4 related.
5 NOTE: 'date' in the format 'yyyy-mm-dd hh:mm:ss [.f...]' (sql timestamp);
6 Time in the format 'hh:mm:ss [.f...]' .
7 -->
8 <communication xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
9 <command name="NewLinkResponse"/>
10 <newlink id="114" title="Benfica vs Sporting" date="2009-02-09 23:59:36"
11 path="http://www.youtube.com/aksdlkajsd" description="Um jogo muito interessante!"
12 active="1">
13 <provider id="2" name="Youtube" address="http://www.youtube.com" description="Google
14 Youtube" active="true"/>
15 <author id="2" name="joaoffr" address="http://joaoffr.servehttp.com" description="O
16 autor desta tese de mestrado!" active="true"/>
17 <contentformat id="3" name=".flv" description="Flash Video File." typeid="3"
18 typename="StaticVideo" typedescription="Non-Real Time Video Content."/>
19 </newlink>
20 </communication>
```

B.8 Mensagem *Link List Request*

Esta mensagem possibilita que um cliente IPTV peça uma lista de um ou mais *links* para conteúdos, segundo vários critérios de procura e selecção. O servidor ao receber uma mensagem deste tipo irá responder com uma mensagem *Link List Response* (secção B.9) no caso do pedido ter sido efectuado com sucesso. Caso o pedido recebido pelo servidor IPTV esteja mal formulado ou exista qualquer outro problema, é enviada uma mensagem *Command Response* (secção B.12) cujo conteúdo possui a descrição das possíveis falhas ou problemas.

A definição formal em XSD para este tipo de mensagem pode ser encontrada de seguida.


```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!--
3 XML Schema for the 'Link List Request' messages sent by the IPTV Client to the IPTV
  Server.
4 -->
5 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
6   <xs:element name="user">
7     <xs:complexType>
8       <xs:attribute name="sessionid" type="xs:integer" use="required" />
9       <xs:attribute name="sessioncode" type="xs:string" use="required" />
10    </xs:complexType>
11  </xs:element>
12  <xs:element name="linktlistrequest">
13    <xs:complexType>
14      <xs:attribute name="typename" type="xs:string" use="required" />
15      <xs:attribute name="limit" type="xs:integer" use="required" />
16      <xs:attribute name="page" type="xs:integer" use="required" />
17      <xs:attribute name="order" type="xs:string" use="optional" />
18      <xs:attribute name="keywords" type="xs:string" use="optional" />
19    </xs:complexType>
20  </xs:element>
21  <xs:element name="communication">
22    <xs:complexType>
23      <xs:sequence>
24        <xs:element ref="command" />
25        <xs:element ref="user" />
26        <xs:element ref="linktlistrequest" />
27      </xs:sequence>
28    </xs:complexType>
29  </xs:element>
30  <xs:element name="command">
31    <xs:complexType>
32      <xs:attribute name="name" use="required">
33        <xs:simpleType>
34          <xs:restriction base="xs:string">
35            <xs:enumeration value="LinkListRequest" />
36          </xs:restriction>
37        </xs:simpleType>
38      </xs:attribute>
39    </xs:complexType>
40  </xs:element>
41 </xs:schema>

```

Esta definição pode ainda ser visualizada de uma forma mais intuitiva e compacta através do diagrama da figura B.8.

A tabela B.8 descreve, para cada elemento da mensagem XML, o nome dos seus atributos bem como o tipo de dados permitido, tipo de utilização (obrigatória/opcional) e valores

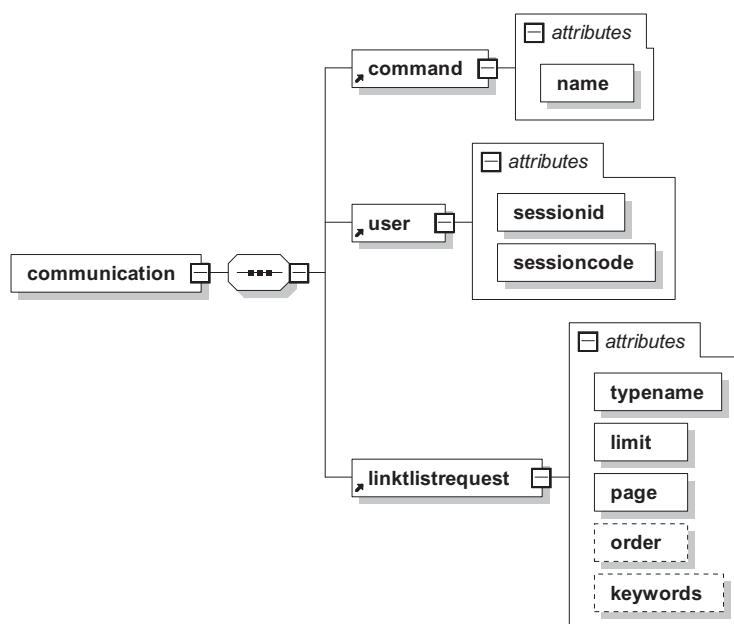


Figura B.8: Diagrama das relações entre os elementos e atributos para mensagens do tipo *Link List Request*.

por defeito.

Descrita toda a sintaxe deste tipo de mensagem enviada pelo cliente IPTV para o servidor IPTV com um pedido de uma lista de *links* referentes a conteúdos multimédia, segue-se a descrição semântica e funcional de cada elemento XML e seus atributos (quando estes existem).

Elemento *communication* Este elemento XML é obrigatório e é a “raíz” da árvore XML

Elemento	Nome do Atributo	Tipo do Atributo	Uso do Atributo	Default do Atributo
communication	–	–	–	–
command	<i>name</i>	string	obrigatório	–
linktlistrequest	<i>typename</i>	string	obrigatório	–
	<i>limit</i>	integer	obrigatório	–
	<i>page</i>	integer	obrigatório	–
	<i>order</i>	string	opcional	–
	<i>keywords</i>	string	opcional	–

Tabela B.8: Tabela dos elementos e atributos para mensagens do tipo *Link List Request*

definida pela mensagem. A sua funcionalidade restringe-se a ser o elemento base que contém todos os outros semanticamente mais importantes e não possui atributos específicos.

Elemento *command* Este elemento obrigatório define e identifica o tipo de mensagem de que se trata. Para o caso de mensagens *Link List Request* este elemento possui sempre o seu único atributo (que também é obrigatório) “name” com o valor igual a “LinkListRequest”, que é o identificador deste tipo de mensagens.

Elemento *linklistrequest* Este elemento XML obrigatório contém nos seus atributos a informação relativa aos critérios de procura e selecção dos conteúdos pretendidos. Os seus atributos e sua semântica são:

typename Atributo obrigatório que identifica a categoria geral dos conteúdos pretendidos no que se refere ao utilizador em questão. Este atributo poderá possuir um dos seguintes valores:

SuggestedContent Servidor só irá procurar resultados de entre os conteúdos sugeridos ao utilizador em questão e que este ainda não tenha assistido (conteúdos pelos quais o utilizador ainda não tenha manifestado qualquer *feedback* no sistema);

VisitedContent Servidor só irá procurar resultados de entre os conteúdos anteriormente assistidos pelo utilizador em questão (conteúdos pelos quais o utilizador já tenha manifestado anteriormente um qualquer *feedback* no sistema);

AnyContent O servidor irá procurar qualquer resultado de entre todos os conteúdos multimédia disponíveis na base de dados do sistema.

limit Atributo obrigatório que determina o número máximo de elementos da lista de conteúdos pedidos;

page Atributo obrigatório que indica qual a página a que este pedido diz respeito, ou seja, caso um primeiro pedido de uma lista de conteúdos tenha retornado o

número máximo *limit* definido e possua a informação que existe mais resultados que respeitam os critérios de procura, o atributo *page* permite pedir os seguintes resultados. Matematicamente tem-se: com $limit = l, page = p, itemID = i$ pode-se admitir que $\forall i \in [0, max_l * max_p - 1] \rightarrow \exists p \in [1, max_p] \wedge \exists l \in [1, max_l] : l * p - 1 = i;$

order Atributo opcional que determina a ordem pela qual os conteúdos resultantes da procura são colocados na lista que será enviada ao cliente IPTV. A sintaxe deste atributo é `<< campo >> _ << ordem >>`. Um exemplo poderá ser “date_desc” que informa que os resultados deverão ser devolvidos ordenados por data de forma descendente. É de notar que caso este campo não exista ou possua uma sintaxe/semântica incorrecta, os resultados serão ordenados por defeito por data em ordem descendente (do mais actual para o menos recente). Os tipos de ordenação possíveis são somente dois: “desc” – descendente e “asc” – ascendente. Relativamente aos campos de ordenação, é possível utilizar:

- “author” Autor do conteúdo;
- “date” Data do conteúdo;
- “insertedby” Conteúdo inserido por;
- “language” Idioma do conteúdo;
- “provider” Provedor do conteúdo;
- “rate” Popularidade do conteúdo;
- “refreshrate” Taxa de actualização do conteúdo;
- “title” Título do conteúdo;
- “visits” Número de visitas do conteúdo.

keywords Atributo opcional que possui uma lista de palavras-chave separadas por um e só um de dois tipos de delimitadores com funções distintas: ponto-e-vírgula (;) ou vírgula (,). Caso todas as palavras-chave sejam separadas por vírgulas, isto indica ao sistema que somente deverão ser incluídos conteúdos na

lista de resultados que possuam todas as palavras-chave algures nas informações descritivas mantidas para cada conteúdo. Caso todas as palavras-chave sejam separadas por ponto-e-vírgula ou exista um misto dos dois tipos de separador, isto indica ao sistema que pelo menos uma das palavras-chave deve ser encontrada num dos campos de dados relativos às informações mantidas sobre cada conteúdo, para que este seja incluído na lista resultado.

Um exemplo de instanciação deste modelo de mensagens numa mensagem XML (com todos os elementos e atributos possíveis) pode ser visto seguidamente.

```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!--
3 XML to be sent by the IPTV Client to the IPTV Server to request a list of N content
4 links of a specific category of content links (visited, suggested or any).
5 NOTE: 'sessioncode' should not have the 'sessionid' at the end. This should have
6 exactly 32 characters;
7 'typename' can be: 'SuggestedContent', 'VisitedContent' or 'AnyContent';
8 'keywords': multiple keywords separated by a semicolon (;) or comma(,) (optional
9 field) ;
10 'Order': FIELD_ORDER (eg: date_asc) (optional).
11 -->
12 <communication xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
13 <command name="LinkListRequest"/>
14 <user sessionid="115" sessioncode="1fea143b66b02696ead45ffacb792347"/>
15 <linktlistrequest typename="SuggestedContent" limit="16" page="3" order="date_desc"
16 keywords="opcional keywords filter; key1; key2"/>
17 </communication>

```

B.9 Mensagem *Link List Response*

Esta mensagem possibilita que o servidor IPTV responda a uma mensagem do tipo *Link List Request* (secção B.8) previamente enviada por um cliente IPTV. Esta mensagem contém, para além de algumas informações extra, a lista de *links* para os conteúdos que resultaram da pesquisa efectuada com os parâmetros submetidos. É possível que a lista de conteúdos não exista (estando presente somente a informação extra na mensagem) caso a pesquisa efectuada não coincida com qualquer conteúdo. Caso o pedido recebido pelo servidor IPTV que despoleta esta resposta esteja mal formulado ou exista qualquer outro

problema, não será enviada uma mensagem deste tipo mas sim uma mensagem *Command Response* (secção B.12) cujo conteúdo possui a descrição das possíveis falhas ou problemas.

Devido ao facto da definição formal em XSD para este tipo de mensagem ser muito extensa, esta não consta deste documento, podendo ser encontrada na directoria

“IPTVFiles/xml/LinkListResponse-Communication.xsd” presente na pasta com o código do servidor.

Esta definição pode ainda ser visualizada de uma forma mais intuitiva e compacta através do diagrama da figura B.9.

A tabela B.9 descreve, para cada elemento da mensagem XML, o nome dos seus atributos bem como o tipo de dados permitido, tipo de utilização (obrigatória/opcional) e valores por defeito.

Descrita toda a sintaxe deste tipo de mensagem enviada pelo servidor IPTV para o cliente IPTV com uma lista de *links* para conteúdos, segue-se a descrição semântica e funcional de cada elemento XML e seus atributos (quando estes existem).

Elemento *communication* Este elemento XML é obrigatório e é a “raíz” da árvore XML definida pela mensagem. A sua funcionalidade restringe-se a ser o elemento base que contém todos os outros semanticamente mais importantes e não possui atributos específicos.

Elemento *command* Este elemento obrigatório define e identifica o tipo de mensagem de que se trata. Para o caso de mensagens *Link List Response* este elemento possui sempre o seu único atributo (que também é obrigatório) “name” com o valor igual a “LinkListResponse”, que é o identificador deste tipo de mensagens.

Elemento *linktlistmetadata* Este elemento XML obrigatório contém nos seus atributos a informação relativa aos critérios de procura e selecção dos conteúdos pretendidos. Estes correspondem aos dados previamente enviados numa mensagem do tipo *Link List Request*. Para além disso, contém também alguma outra informação. Os seus atributos e sua semântica são:

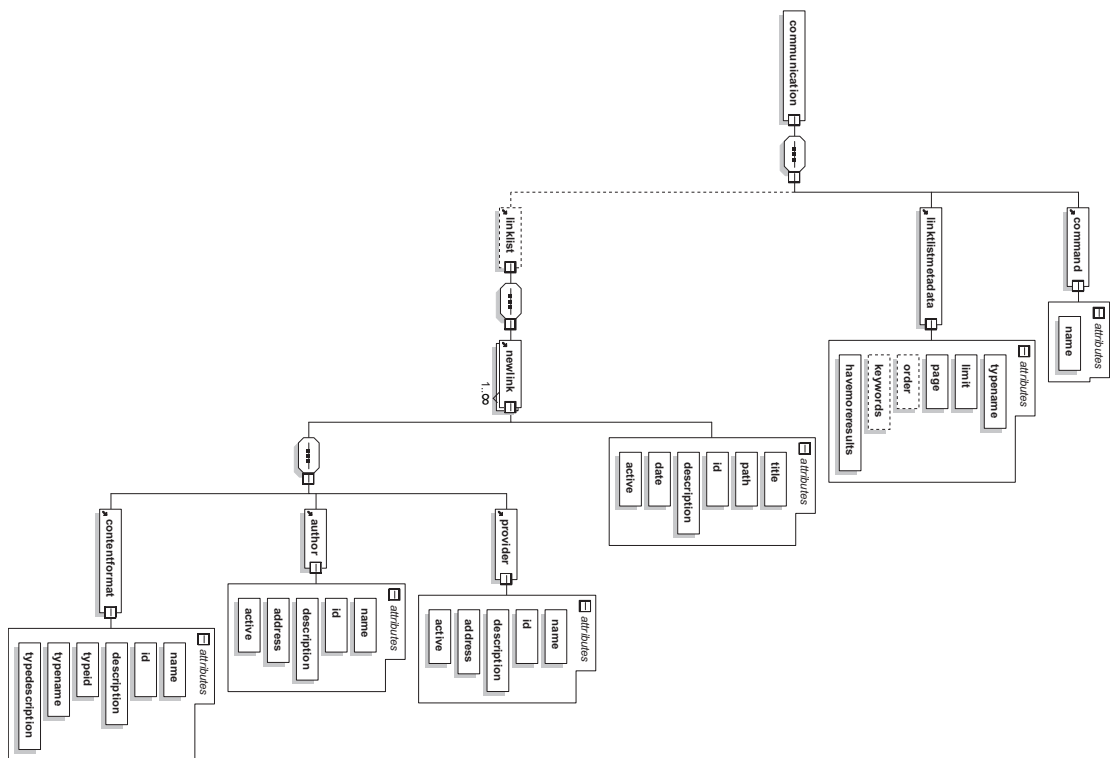


Figura B.9: Diagrama das relações entre os elementos e atributos para mensagens do tipo *Link List Response*.

Elemento	Nome do Atributo	Tipo do Atributo	Uso do Atributo	Default do Atributo
communication	–	–	–	–
command	<i>name</i>	string	obrigatório	–
linktlistmetadata	<i>typename</i>	string	obrigatório	–
	<i>limit</i>	integer	obrigatório	–
	<i>page</i>	integer	obrigatório	–
	<i>order</i>	string	opcional	–
	<i>keywords</i>	string	opcional	–
	<i>havemorerresults</i>	byte	obrigatório	–
newlink	<i>id</i>	long	obrigatório	–
	<i>title</i>	string	obrigatório	–
	<i>date</i>	string	obrigatório	–
	<i>path</i>	string	obrigatório	–
	<i>description</i>	string	obrigatório	–
	<i>active</i>	byte	obrigatório	–
provider	<i>id</i>	long	obrigatório	–
	<i>name</i>	string	obrigatório	–
	<i>address</i>	string	obrigatório	–
	<i>description</i>	string	obrigatório	–
	<i>active</i>	byte	obrigatório	–
author	<i>id</i>	long	obrigatório	–
	<i>name</i>	string	obrigatório	–
	<i>address</i>	string	obrigatório	–
	<i>description</i>	string	obrigatório	–
	<i>active</i>	byte	obrigatório	–
contentformat	<i>id</i>	long	obrigatório	–
	<i>name</i>	string	obrigatório	–
	<i>description</i>	string	obrigatório	–
	<i>typeid</i>	long	obrigatório	–
	<i>typename</i>	string	obrigatório	–
	<i>typedescription</i>	string	obrigatório	–

Tabela B.9: Tabela dos elementos e atributos para mensagens do tipo *Link List Response*

typename Atributo obrigatório que identifica a categoria geral dos conteúdos pretendidos no que se refere ao utilizador em questão. Este atributo poderá possuir um dos seguintes valores:

SuggestedContent Servidor só procurou resultados de entre os conteúdos sugeridos ao utilizador em questão e que este ainda não tenha assistido (conteúdos pelos quais o utilizador ainda não tenha manifestado qualquer *feedback* no sistema);

VisitedContent Servidor só procurou resultados de entre os conteúdos anteriormente assistidos pelo utilizador em questão (conteúdos pelos quais o utilizador já tenha manifestado anteriormente um qualquer *feedback* no sistema);

AnyContent Servidor procurou qualquer resultados de entre todos os conteúdos disponíveis na base de dados do sistema.

limit Atributo obrigatório que informa o número máximo de elementos da lista de conteúdos pedidos;

page Atributo obrigatório que indica qual a página a que esta resposta diz respeito;

order Atributo opcional que identifica a ordem pela qual os conteúdos resultantes da procura foram colocados na lista que é parte integrante desta mensagem. A sintaxe deste atributo é << *campo* >> - << *ordem* >>;

keywords Atributo opcional que possui uma lista de palavras-chave separadas por um e só um de dois tipos de delimitadores com funções distintas: ponto-e-vírgula (;) ou vírgula (,);

havemoreresults Atributo obrigatório que caso possua um “0” (falso) indica que não existem mais páginas de resultados com número superior à actual. Caso o seu valor seja “1” (verdadeiro), indica que ainda existem mais páginas com os próximos resultados.

Elemento *newlink* Este elemento XML obrigatório contém nos seus atributos a in-

formação referente a um *link* para o respectivo conteúdo. Os seus atributos e sua semântica são:

id Atributo obrigatório que contém um identificador numérico que permite referenciar o conteúdo inequivocamente na base de dados;

title Atributo obrigatório que contém o título do respectivo conteúdo proveniente da fonte que disponibiliza o mesmo;

date Atributo obrigatório que possui a data de criação do conteúdo proveniente da fonte que disponibiliza este;

path Atributo obrigatório que contém o URI (o elemento muitas vezes referido nesta dissertação como *link*) de acesso ao conteúdo multimédia disponibilizado pelo provedor do conteúdo;

description Atributo obrigatório que possui uma pequena descrição do respectivo conteúdo multimédia proveniente da fonte que disponibiliza o mesmo;

active Atributo obrigatório que indica se o respectivo conteúdo se encontra activo (disponível) – valor igual a “1” – ou inactivo (indisponível) – valor igual a “0”. Como será de esperar, o servidor só irá enviar *links* para conteúdos com o este campo de valor igual a “1”, pois o sistema não oferece aos seus utilizadores *links* para conteúdos indisponíveis à partida.

Elemento *provider* Este elemento XML obrigatório possui nos seus atributos informações acerca da entidade que disponibiliza o acesso ao respectivo conteúdo. Os atributos deste elemento e sua semântica são:

id Atributo obrigatório que contém um identificador numérico que permite referenciar o provedor do conteúdo multimédia inequivocamente na base de dados do sistema. Este atributo permite o relacionamento entre o conteúdo e a entidade que o disponibiliza;

name Atributo obrigatório que possui o nome da entidade que disponibiliza o acesso ao conteúdo;

address Atributo obrigatório que contém o URI da entidade que disponibiliza o conteúdo;

description Atributo obrigatório que contém uma pequena descrição relativa ao provedor do conteúdo;

active Atributo obrigatório que informa se a entidade que disponibiliza o conteúdo ainda existe e/ou está acessível e disponível (valor igual a “1”) ou, pelo contrário, já não existe ou não se encontra acessível/disponível (valor igual a “0”). Como será de esperar, o servidor só irá enviar mensagens deste tipo com este atributo com valor igual a “1”, pois o sistema não oferece aos seus utilizadores *links* para conteúdos cuja entidade que os disponibiliza não exista ou não esteja acessível/disponível à partida.

Elemento *author* Este elemento XML obrigatório possui nos seus atributos informações acerca do autor do respectivo conteúdo. Os atributos deste elemento e sua semântica são:

id Atributo obrigatório que contém um identificador numérico que permite referenciar o autor do conteúdo multimédia inequivocamente na base de dados do sistema. Este atributo permite o relacionamento entre o conteúdo e o seu autor. A noção de autor é distinta da noção de provedor de acesso;

name Atributo obrigatório que possui o nome do autor do conteúdo;

address Atributo obrigatório que contém o URI do autor do conteúdo. Caso o autor não possua uma *web page* ou algo do género que possa ser referenciado neste atributo, este ficará em branco;

description Atributo obrigatório que possui uma pequena descrição das características do autor do conteúdo multimédia respectivo;

active Atributo obrigatório que informa se o autor do conteúdo ainda está activo no sistema. Caso o autor ainda esteja activo, o valor deste campo será “1” (verdadeiro). Caso contrário, será “0” (falso). O servidor irá enviar mensagens deste

tipo com este atributo possuindo um valor que possivelmente será irrelevante para o cliente IPTV, possuindo somente carácter informativo.

Elemento *contentformat* Este elemento XML obrigatório contém nos seus atributos a informação referente ao formato multimédia do conteúdo respectivo no que diz respeito à sua forma de codificação áudio-visual. Para além disso, possui ainda informação referente ao tipo de fluxo, ou seja, áudio “gravado”, vídeo com áudio “gravado” ou conteúdo em “tempo real”. Os seus atributos e sua semântica são:

id Atributo obrigatório que contém um identificador numérico que permite referenciar o formato (codificação) do conteúdo inequivocamente na base de dados do servidor IPTV;

name Atributo obrigatório que contém o nome do formato dado pela extensão com a sintaxe “.<<*nome da extensão*>>”;

description Atributo obrigatório que contém uma pequena descrição referente ao respectivo formato do conteúdo multimédia;

typeid Atributo obrigatório que contém um identificador numérico que permite referenciar de forma inequívoca o tipo de fluxo a que este se refere;

typename Atributo obrigatório que possui o nome do tipo de fluxo multimédia (áudio “gravado”, vídeo com áudio “gravado”, conteúdo em “tempo real”, etc.);

typedescription Atributo obrigatório que possui uma breve descrição sobre o tipo de fluxo multimédia em questão.

Um exemplo de instanciação deste modelo de mensagens numa mensagem XML (com todos os elementos e atributos possíveis) pode ser visto seguidamente.

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!--
3 XML to be sent by the IPTV Server to the IPTV Client with a list of N content links in
   response to a previous 'LinkListRequest' command.
4 -->
5 <communication xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
6 <command name="LinkListResponse" />
```

```

7 <linktlistmetadata typename="AnyContent" limit="5" page="1" order="date_desc"
  keywords="futebol; benfica; sporting" havemorerresults="0"/>
8 <linklist>
9 <newlink id="114" title="Benfica vs Sporting" date="2008-02-09 21:59:36"
  path="http://www.youtube.com/aksdlkajsd" description="O Benfica perde mais um
  jogo..." active="1">
10 <provider id="2" name="Youtube" address="http://www.youtube.com"
  description="Google Youtube" active="1"/>
11 <author id="2" name="joaoffr" address="http://joaoffr.servehttp.com"
  description="O autor desta tese de mestrado!" active="1"/>
12 <contentformat id="3" name=".flv" description="Flash Video File." typeid="3"
  typename="StaticVideo" typedescription="Non-Real Time Video Content."/>
13 </newlink>
14 <newlink id="155" title="Benfica vs Porto" date="2008-07-15 17:00:38"
  path="http://www.youtube.com/dgfdgdfg" description="Um jogo de nervos!"
  active="1">
15 <provider id="2" name="Youtube" address="http://www.youtube.com"
  description="Google Youtube" active="1"/>
16 <author id="2" name="joaoffr" address="http://joaoffr.servehttp.com"
  description="O autor desta tese de mestrado!" active="1"/>
17 <contentformat id="3" name=".flv" description="Flash Video File." typeid="3"
  typename="StaticVideo" typedescription="Non-Real Time Video Content."/>
18 </newlink>
19 <newlink id="224" title="Porto vs Sporting" date="2008-09-09 20:59:11"
  path="http://www.youtube.com/derteaawwr" description="Jogo FCP contra SCP
  (campeonato PortugÃAs)." active="1">
20 <provider id="2" name="Youtube" address="http://www.youtube.com"
  description="Google Youtube" active="1"/>
21 <author id="2" name="joaoffr" address="http://joaoffr.servehttp.com"
  description="O autor desta tese de mestrado!" active="1"/>
22 <contentformat id="3" name=".flv" description="Flash Video File." typeid="3"
  typename="StaticVideo" typedescription="Non-Real Time Video Content."/>
23 </newlink>
24 <newlink id="251" title="Estoril vs Belenenses" date="2009-02-09 23:59:36"
  path="http://www.youtube.com/zasdasa" description="Jogo de futebol do campeonato
  portuguÃAs." active="1">
25 <provider id="2" name="Youtube" address="http://www.youtube.com"
  description="Google Youtube" active="1"/>
26 <author id="2" name="joaoffr" address="http://joaoffr.servehttp.com"
  description="O autor desta tese de mestrado!" active="1"/>
27 <contentformat id="3" name=".flv" description="Flash Video File." typeid="3"
  typename="StaticVideo" typedescription="Non-Real Time Video Content."/>
28 </newlink>
29 </linklist>
30 </communication>

```

B.10 Mensagem *Link Feedback Request*

Esta mensagem permite que um cliente IPTV envie um *feedback* relativo a um conteúdo específico para o servidor IPTV. O servidor ao receber uma mensagem deste tipo irá responder sempre com uma mensagem *Command Response* (secção B.12), cujo conteúdo pode variar entre a confirmação da recepção do *feedback* com sucesso ou um qualquer erro detectado.

A definição formal em XSD para este tipo de mensagem pode ser encontrada de seguida.

```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!--
3 XML Schema for the 'Link Feedback Request' messages sent by the IPTV Client to the
4 IPTV Server.
5 -->
6 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
7   <xs:element name="user">
8     <xs:complexType>
9       <xs:attribute name="sessionid" type="xs:long" use="required" />
10      <xs:attribute name="sessioncode" type="xs:string" use="required" />
11    </xs:complexType>
12  </xs:element>
13  <xs:element name="previouslink">
14    <xs:complexType>
15      <xs:attribute name="playedtime" type="xs:long" use="required" />
16      <xs:attribute name="playedperc" type="xs:double" use="required" />
17      <xs:attribute name="modename" type="xs:string" use="required" />
18      <xs:attribute name="id" type="xs:long" use="required" />
19      <xs:attribute name="feedbackname" type="xs:string" use="required" />
20    </xs:complexType>
21  </xs:element>
22  <xs:element name="communication">
23    <xs:complexType>
24      <xs:sequence>
25        <xs:element ref="command" />
26        <xs:element ref="user" />
27        <xs:element ref="previouslink" />
28      </xs:sequence>
29    </xs:complexType>
30  </xs:element>
31  <xs:element name="command">
32    <xs:complexType>
33      <xs:attribute name="name" use="required">
34        <xs:simpleType>
35          <xs:restriction base="xs:string">
36            <xs:enumeration value="LinkFeedbackRequest" />
37            <xs:enumeration value="NewLinkRequest" />
38          </xs:restriction>

```

```

38         </xs:simpleType>
39     </xs:attribute>
40 </xs:complexType>
41 </xs:element>
42 </xs:schema>

```

Esta definição pode ainda ser visualizada de uma forma mais intuitiva e compacta através do diagrama da figura B.10.

A tabela B.10 descreve, para cada elemento da mensagem XML, o nome dos seus atributos bem como o tipo de dados permitido, tipo de utilização (obrigatória/opcional) e valores por defeito.

Descrita toda a sintaxe deste tipo de mensagem enviada pelo cliente IPTV para o servidor IPTV com um *feedback* relativo a um conteúdo assistido, segue-se agora a descrição semântica e funcional de cada elemento XML e seus atributos (quando estes existem).

Elemento *communication* Este elemento XML é obrigatório e é a “raíz” da árvore XML definida pela mensagem. A sua funcionalidade restringe-se a ser o elemento base que contém todos os outros semanticamente mais importantes e não possui atributos específicos.

Elemento *command* Este elemento obrigatório define e identifica o tipo de mensagem de que se trata. Para o caso de mensagens *Link Feedback Request* este elemento possui sempre o seu único atributo (que também é obrigatório) “name” com o valor igual a “LinkFeedbackRequest”, que é o identificador deste tipo de mensagens.

Elemento *user* Este elemento XML obrigatório contém nos seus atributos a informação que será usada para autenticar este pedido (operação) e associar o mesmo a um utilizador com sessão previamente iniciada no sistema. Os atributos deste elemento e sua semântica são:

sessionid Atributo obrigatório que contém um identificador numérico que permite relacionar o utilizador com a sua actual sessão activa no sistema. A necessidade deste atributo tem como objectivo não só a identificação do utilizador que

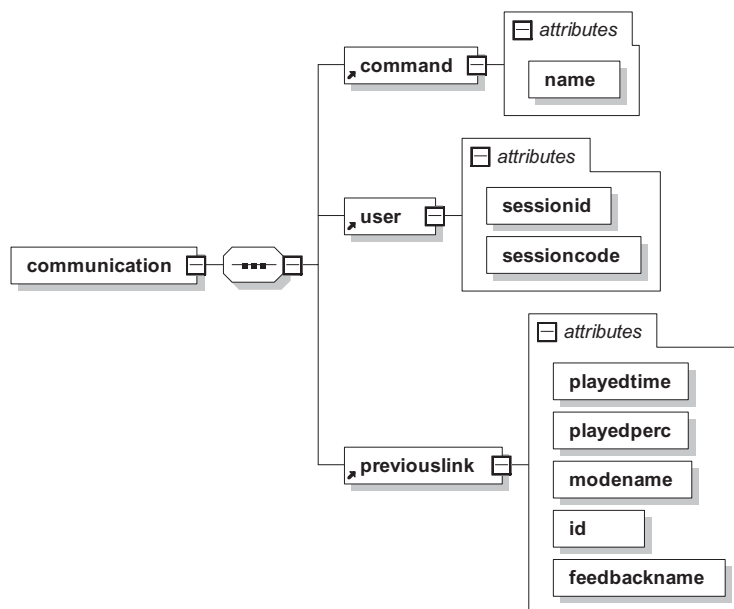


Figura B.10: Diagrama das relações entre os elementos e atributos para mensagens do tipo *Link Feedback Request*.

Elemento	Nome do Atributo	Tipo do Atributo	Uso do Atributo	Default do Atributo
communication	–	–	–	–
command	<i>name</i>	string	obrigatório	–
user	<i>sessionid</i> <i>sessioncode</i>	long string	obrigatório obrigatório	– –
previouslink	<i>id</i> <i>modename</i> <i>feedbackname</i> <i>playedperc</i> <i>playedtime</i>	long string double double long	obrigatório obrigatório obrigatório obrigatório obrigatório	– – – – –

Tabela B.10: Tabela dos elementos e atributos para mensagens do tipo *Link Feedback Request*

requer a operação mas também o estabelecimento de políticas de segurança e autenticação;

sessioncode Atributo obrigatório que contém uma cadeia de caracteres que permite ao sistema verificar a autenticidade da operação requisitada pela mensagem XML. Este campo em conjunto com o *sessionid* permite a identificação da sessão do utilizador no sistema (se esta existir) e permite também o estabelecimento de políticas de segurança na realização das operações.

Elemento *previouslink* Este elemento XML obrigatório permite que o cliente IPTV envie a classificação (*feedback*) relativa a um conteúdo acedido anteriormente. Os atributos deste elemento e sua semântica são:

id Atributo obrigatório que contém um identificador numérico que permite identificar o conteúdo multimédia inequivocamente na base de dados. Este atributo permite efectuar o relacionamento entre a classificação do utilizador (*feedback*) e o conteúdo multimédia respectivo;

modename Atributo obrigatório que contém o identificador do modo de visualização que o utilizador usou para assistir ao conteúdo em questão (ambiente de visualização). Este atributo permite estabelecer a relação entre o *feedback* do utilizador, o conteúdo assistido e o modo de visualização utilizado, contribuindo para a construção do perfil do utilizador;

feedbackname Atributo obrigatório que contém o identificador (nome) relativo à classificação (*feedback*) dada pelo utilizador ao conteúdo multimédia assistido anteriormente. A classificação é efectuada segundo níveis específicos que traduzem as preferências do utilizador pelos conteúdos assistidos (*"I Like it"*, *"Don't like it"*, etc.);

playedperc Atributo obrigatório que identifica a percentagem (no intervalo [0.0, 1.0]) do conteúdo multimédia em questão reproduzida pelo cliente IPTV. Este atributo é utilizado pelo sistema para recolha de informação mais específica

sobre a experiência de utilização do serviço de IPTV e construção do perfil do utilizador;

playedtime Atributo obrigatório que possui a quantidade de tempo (em milisegundos) reproduzido pelo cliente IPTV relativo ao conteúdo em questão. Este atributo é utilizado pelo sistema para recolha de informação mais específica sobre a experiência de utilização do serviço de IPTV e construção do perfil do utilizador.

Um exemplo de instanciação deste modelo de mensagens numa mensagem XML, pode ser visto seguidamente.

```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!--
3 XML sent by the IPTV Client to the IPTV Server with the feedback related to a specific
4 content.
5 NOTE: 'Time' in 'hh:mm:ss [.f...]' format;
6       'sessioncode' without 'sessionid' at the end. 'sessioncode' must have exactly 32
7       characters.
8       'playedperc' in the set [0.0, 1.0];
9       'playertime' in milliseconds.
10 -->
11 <communication xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
12   <command name="LinkFeedbackRequest" />
13   <user sessionid="10" sessioncode="1fea143b66b02696ead45ffacb792347" />
14   <previouslink id="1" modename="friends" feedbackname="I like it" playedperc="0.545"
15     playedtime="8447" />
16 </communication>

```

B.11 Mensagem *Command Request*

Esta mensagem permite que um cliente IPTV envie um pedido com um comando ou operação que não exija a autenticação de qualquer utilizador para ser executada. O servidor ao receber uma mensagem deste tipo irá responder sempre com uma mensagem *Command Response* (secção B.12), cujo conteúdo pode variar entre a confirmação da execução do comando com sucesso, a resposta à operação requerida ou a informação de um problema detectado.

A definição formal em XSD para este tipo de mensagem pode ser encontrada de seguida.

```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!--
3     XML Schema of the message sent by the IPTV Client to the IPTV Server with an
4     configuration/information request.
5 -->
6 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
7     <xs:element name="communication">
8         <xs:complexType>
9             <xs:sequence>
10                <xs:element ref="command" />
11                <xs:element ref="request" />
12            </xs:sequence>
13        </xs:complexType>
14    </xs:element>
15    <xs:element name="command">
16        <xs:complexType>
17            <xs:attribute name="name" use="required">
18                <xs:simpleType>
19                    <xs:restriction base="xs:string">
20                        <xs:enumeration value="CommandRequest" />
21                    </xs:restriction>
22                </xs:simpleType>
23            </xs:attribute>
24        </xs:complexType>
25    </xs:element>
26    <xs:element name="request">
27        <xs:complexType>
28            <xs:attribute name="name" use="required">
29                <xs:simpleType>
30                    <xs:restriction base="xs:string">
31                        <xs:enumeration value="Configuration" />
32                        <xs:enumeration value="Ping" />
33                    </xs:restriction>
34                </xs:simpleType>
35            </xs:attribute>
36            <xs:attribute name="data" type="xs:string" use="optional" />
37            <xs:attribute name="description" type="xs:string" use="optional" />
38        </xs:complexType>
39    </xs:element>
40 </xs:schema>

```

Esta definição pode ainda ser visualizada de uma forma mais intuitiva e compacta através do diagrama da figura B.11.

A tabela B.11 descreve, para cada elemento da mensagem XML, o nome dos seus atributos bem como o tipo de dados permitido, tipo de utilização (obrigatória/opcional) e valores por defeito.

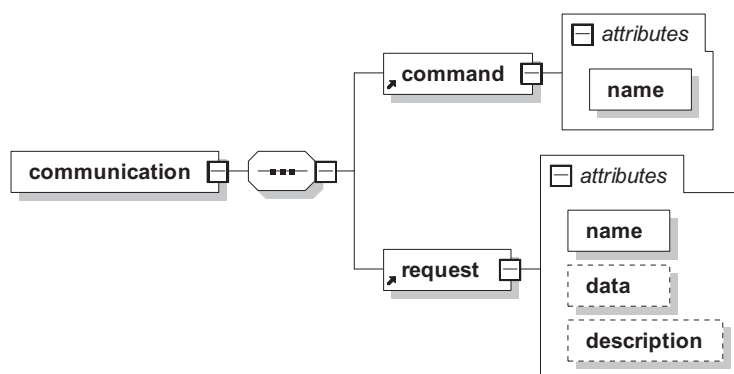


Figura B.11: Diagrama das relações entre os elementos e atributos para mensagens do tipo *Command Request*.

Elemento	Nome do Atributo	Tipo do Atributo	Uso do Atributo	Default do Atributo
communication	–	–	–	–
command	<i>name</i>	string	obrigatório	–
request	<i>name</i>	string	obrigatório	–
	<i>data</i>	string	opcional	–
	<i>description</i>	string	opcional	–

Tabela B.11: Tabela dos elementos e atributos para mensagens do tipo *Command Request*

Descrita toda a sintaxe deste tipo de mensagem enviada pelo cliente IPTV para o servidor IPTV, segue-se agora a descrição semântica e funcional de cada elemento XML e seus atributos (quando estes existem).

Elemento *communication* Este elemento XML é obrigatório e é a “raíz” da árvore XML definida pela mensagem. A sua funcionalidade restringe-se a ser o elemento base que contém todos os outros semanticamente mais importantes e não possui atributos específicos.

Elemento *command* Este elemento obrigatório define e identifica o tipo de mensagem de que se trata. Para o caso de mensagens *Command Request* este elemento possui sempre o seu único atributo (que também é obrigatório) “name” com o valor igual a “CommandRequest”, que é o identificador deste tipo de mensagens.

Elemento *request* Este elemento XML obrigatório contém nos seus atributos os dados referentes ao comando/operação pretendida. Os atributos deste elemento e sua semântica são:

name Atributo obrigatório que contém o identificador textual do comando/operação a realizar;

data Atributo opcional que contém dados em forma de texto utilizados na realização da operação/comando identificada pelo atributo “name”;

description Atributo opcional que contém (no formato de texto) dados suplementares à realização da operação/comando pretendido.

Seguem-se dois exemplos de instanciação deste modelo de mensagem com todos os elementos e atributos possíveis. O primeiro refere-se a um comando “Ping” na camada protocolar da aplicação definida pelo servidor de IPTV e o segundo refere-se ao pedido de uma estrutura XML contendo informações específicas de configuração do cliente. Ambos podem ser vistos seguidamente.

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!--
3 XML sent by the IPTV Client to the IPTV Server with a request for a 'ping' message in
  the layer defined by the IPTV Server.
4 If online , the server will respond with a 'CommandResponse - Pong' message.
5 -->
6 <communication xmlns:xsi=" http://www.w3.org/2001/XMLSchema-instance">
7   <command name="CommandRequest" />
8   <request name="Ping" data="Data to be returned by the server (opcional)"
     description="description (opcional)" />
9 </communication>
```

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!--
3 XML sent by the IPTV Client to the IPTV Server with a request for a
  configuration/information described in the field 'data'.
4 The server will respond with the requested data in a 'CommandResponse' XML message.
5 -->
6 <communication xmlns:xsi=" http://www.w3.org/2001/XMLSchema-instance">
7   <command name="CommandRequest" />
8   <request name="Configuration" data="InterestAreaTree" description="Basic Interest Area
     Tree (opcional field)" />
9 </communication>
```

B.12 Mensagem *Command Response*

Esta mensagem permite ao servidor IPTV responder aos mais variados pedidos de uma forma padrão. Desde a uma mensagem *Command Request* (secção B.11) com um comando ou operação que não exija a autenticação de qualquer utilizador para ser executada, até a uma qualquer mensagem recebida caso exista algum erro de sintaxe/semântica na recepção desta ou durante a execução das tarefas associadas à mesma, esta mensagem permite dar um *feedback* linear ao cliente IPTV. Caso não exista um tipo de resposta mais específico e especializado para uma determinada mensagem recebida, no limite será usado este tipo de mensagem como resposta, reflectindo o estado de recepção, estado do *parsing* e retorno da execução das tarefas subjacentes.

Para além destes dados, esta mensagem contém opcionalmente um elemento de nome *nested* (dentro do elemento *response*) que suporta quaisquer elementos XML dentro do

mesmo. Esta característica reveste-se de extrema utilidade, permitindo o envio de qualquer tipo de informação de uma forma organizada. Um exemplo desta funcionalidade é a resposta enviada a uma mensagem *User Authentication* (secção B.1) que, no caso de sucesso, será uma mensagem *Command Response* em que no elemento *nested* são enviados os dados do respectivo utilizador para tratamento pelo cliente IPTV.

A definição formal em XSD para este tipo de mensagem pode ser encontrada de seguida.

```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!-- edited with XMLSpy v2008 rel. 2 sp2 (http://www.altova.com) by SHOCK (SHOCK) -->
3 <!--
4 XML Schema of the message sent by the IPTV Server to the IPTV Client with the response
   to one command made previously by the IPTV Client.
5 -->
6 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
7   <xs:element name="communication">
8     <xs:complexType>
9       <xs:sequence>
10        <xs:element ref="command" />
11        <xs:element ref="response" />
12      </xs:sequence>
13    </xs:complexType>
14  </xs:element>
15  <xs:element name="command">
16    <xs:complexType>
17      <xs:attribute name="name" use="required">
18        <xs:simpleType>
19          <xs:restriction base="xs:string">
20            <xs:enumeration value="CommandResponse" />
21          </xs:restriction>
22        </xs:simpleType>
23      </xs:attribute>
24    </xs:complexType>
25  </xs:element>
26  <xs:element name="response">
27    <xs:complexType>
28      <xs:sequence>
29        <xs:element name="nested" minOccurs="0">
30          <xs:complexType>
31            <xs:sequence>
32              <xs:any />
33            </xs:sequence>
34          </xs:complexType>
35        </xs:element>
36      </xs:sequence>
37      <xs:attribute name="name" use="required">
38        <xs:simpleType>
39          <xs:restriction base="xs:string">

```

```

40         <xs:enumeration value="OK" />
41         <xs:enumeration value="CANCEL" />
42         <xs:enumeration value="FAIL" />
43         <xs:enumeration value="ERROR" />
44         <xs:enumeration value="PONG" />
45     </xs:restriction>
46 </xs:simpleType>
47 </xs:attribute>
48 <xs:attribute name="data" type="xs:string" use="optional" />
49 <xs:attribute name="description" type="xs:string" use="optional" />
50 </xs:complexType>
51 </xs:element>
52 </xs:schema>

```

Esta definição pode ainda ser visualizada de uma forma mais intuitiva e compacta através do diagrama da figura B.12.

A tabela B.12 descreve, para cada elemento da mensagem XML, o nome dos seus atributos bem como o tipo de dados permitido, tipo de utilização (obrigatória/opcional) e valores por defeito.

Descrita toda a sintaxe deste tipo de mensagem enviada pelo servidor IPTV para o cliente IPTV, segue-se agora a descrição semântica e funcional de cada elemento XML e seus atributos (quando estes existem).

Elemento *communication* Este elemento XML é obrigatório e é a “raíz” da árvore XML definida pela mensagem. A sua funcionalidade restringe-se a ser o elemento base que contém todos os outros semanticamente mais importantes e não possui atributos específicos.

Elemento *command* Este elemento obrigatório define e identifica o tipo de mensagem

Elemento	Nome do Atributo	Tipo do Atributo	Uso do Atributo	Default do Atributo
communication	–	–	–	–
command	<i>name</i>	string	obrigatório	–
response	<i>name</i>	string	obrigatório	–
	<i>data</i>	string	opcional	–
	<i>description</i>	string	opcional	–
nested	–	–	opcional	–

Tabela B.12: Tabela dos elementos e atributos para mensagens do tipo *Command Response*

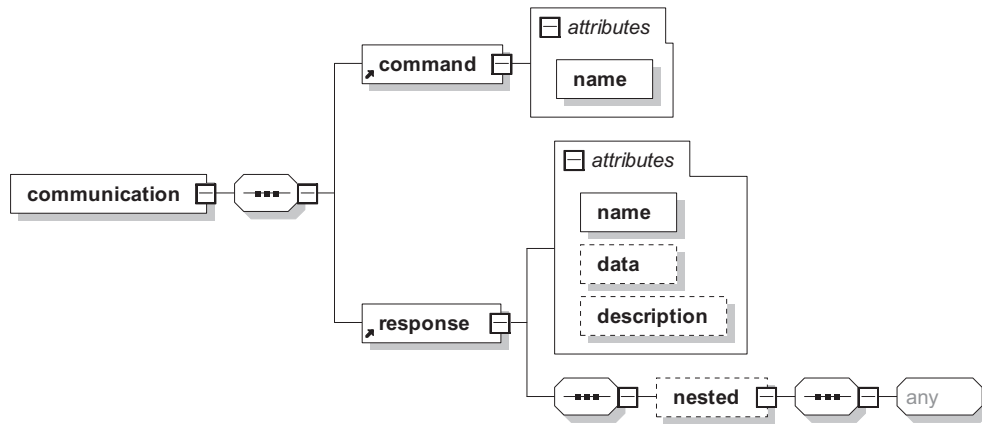


Figura B.12: Diagrama das relações entre os elementos e atributos para mensagens do tipo *Command Response*.

de que se trata. Para o caso de mensagens *Command Response* este elemento possui sempre o seu único atributo (que também é obrigatório) “name” com o valor igual a “CommandResponse”, que é o identificador deste tipo de mensagens.

Elemento *response* Este elemento XML obrigatório contém nos seus atributos os dados referentes ao comando/operação realizada e seu estado final de execução. Para além destes atributos, este elemento possui ainda um elemento embebido de nome *nested* que é opcional. Caso seja usado, é possível colocar dentro deste qualquer estrutura XML sintacticamente correcta, conferindo uma grande flexibilidade a este tipo de mensagem. Os atributos do elemento *response* e sua semântica são:

name Atributo obrigatório que contém o identificador textual do comando/operação realizado ou o seu estado final;

data Atributo opcional que contém dados em forma de texto retornados após a execução da operação/comando previamente requerido ou então um identificador de um sub-comando;

description Atributo opcional que contém (no formato de texto) dados ou descrições suplementares retornadas após a execução da operação/comando pretendido.

Seguem-se um exemplo genérico de instanciação deste modelo de mensagem com todos os atributos possíveis mas sem a utilização do elemento *nested*. Este modelo é muito utilizado na resposta a comandos/operações quer quando estas falham quer quando ocorrem com sucesso e não devolvem qualquer valor.

```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!--
3 XML sent by the IPTV Server to the IPTV Client with the response for a command made
4 previously by the IPTV Client.
5 This response has an required field named 'name', a optional data block 'data' and a
6 description field named 'description' also optional.
7 -->
8 <communication xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
9   <command name="CommandResponse" />
10  <response name="OK" data="any data block (opcional)" description="description
11    (opcional)" />
12 </communication>

```

Considerando agora o caso particular de um pedido efectuado com recurso a uma mensagem *Command Request - Ping*, que procede à verificação da ligação na camada protocolar definida pelo servidor IPTV, a sua resposta é construída com base numa mensagem *Command Response* com dados específicos. Esta mensagem possui no atributo *name* e *data* do elemento *response* o identificador textual constante *Pong* e a cópia dos dados enviados em *data*, respectivamente. Segue-se um exemplo da mesma.

```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!--
3 XML sent by the IPTV Server to the IPTV Client for the ping response.
4 This response has an required field named 'name', a optional data block 'data' and a
5 description field
6 named 'description' also optional.
7 -->
8 <communication xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
9   <command name="CommandResponse" />
10  <response name="pong" data="114887f7a454cc5d7e7a" description="Server echo reply">
11 </response>
12 </communication>

```

No caso de um pedido de dados de configuração, mais especificamente, o pedido da árvore de áreas de interesse, os dados serão enviados dentro do elemento *nested*, pertencente ao elemento *response*, como é possível ver seguidamente. Os atributos *name* e *data* do elemento *response* possuem respectivamente o estado e o nome da configuração pedida.

```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!--
3 XML sent by the IPTV Server to the IPTV Client with the interest area tree.
4 This response has an required field named 'name', a optional data block 'data' and a
   description field
5 named 'description' also optional.
6 -->
7 <communication xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
8   <command name="CommandResponse" />
9   <response name="OK" data="InterestAreaTree">
10     <nested>
11       <interestareatable minrate="0.0" maxrate="1.0" defaultrate="0.5">
12         <area name="all" synonyms="*" description="all interest areas">
13           <area name="sport" synonyms="desporto" description="All sports in the world!">
14             <area name="football" synonyms="futebol" description="A game with a ball and
15               11 players.">
16             </area>
17             <area name="volleyball" synonyms="volei" description="A game that is played
18               with the hands.">
19             </area>
20             <area name="running" synonyms="corrida">
21             </area>
22             <area name="tennis" synonyms="tenis" description="Play it with rackets and a
23               yellow ball!">
24             </area>
25             <area name="politics" synonyms="politica">
26             </area>
27             <area name="education">
28             </area>
29             <area name="art" synonyms="arte" description="artistic things">
30             </area>
31           </interestareatable>
32         </nested>
33       </response>
34     </communication>

```

Qualquer configuração pode ser enviada ao cliente IPTV através de uma mensagem deste tipo. Isto tem como objectivo controlar e padronizar as configurações e dados presentes nos múltiplos (tipos de) clientes IPTV.

B.13 Estrutura geral das mensagens XML

É de notar que existe um ficheiro XML *schema* com a estrutura e sintaxe global que qualquer mensagem enviada ou recebida deverá respeitar. Esta sintaxe não é mais que a composição da sintaxe de todas as mensagens permitidas num único ficheiro *.xsd*.

Devido ao facto deste ficheiro conter bastantes linhas de código XSD que são redundantes relativamente ao que foi mencionado anteriormente, o mesmo não foi incluído aqui. Para o consultar, basta aceder à pasta do código do servidor, especificamente ao directório “IPTVFiles/xml/” e, dentro da mesma, ao ficheiro “GeneralSchema-Communication.xsd”.

Este ficheiro é utilizado pelo sistema sempre que chega uma nova mensagem, proveniente de um qualquer cliente IPTV. Antes da nova mensagem recebida ser processada individualmente, é sujeita a uma comparação com as regras presentes em “GeneralSchema-Communication.xsd”, e só depois é validada pelas suas regras específicas e processadas as tarefas associadas.

Este passo evita que mensagens incorrectas entrem na árvore de decisão das tarefas a executar, criando uma maior protecção contra erros no sistema. Obviamente que esta verificação/validação extra leva a uma perda de eficiência computacional na execução do servidor, podendo ser desactivada caso seja necessário. Tal como esta, qualquer uma das verificações individuais de sintaxe das mensagens recebidas através dos seus respectivos ficheiros XSD pode ser desactivada, aumentando a eficiência computacional do sistema mas perdendo-se em escalabilidade, robustez e tolerância a erros.

Tal como é verificável pela anterior descrição das mensagens XML implementadas, o elemento *command* está sempre presente em todas elas, possuindo no seu atributo único obrigatório *name* o nome (sem espaços) da respectiva mensagem. Isto permite reconhecer o formato da mensagem de forma simples e rápida.

Anexo C

Exemplo de Utilização da API Cliente

Neste anexo será apresentado um exemplo simples de utilização da API de comunicação com o servidor IPTV utilizando HTTP. Este exemplo visa somente ilustrar a metodologia de funcionamento desta API.

Para mais exemplos e/ou informações sobre o código desenvolvido, é favor consultar a documentação *Javadoc* fornecida com o código da API, bem como outros exemplos que se encontram no pacote “teste/examples”.

```
1 /*
2  * Example how to use Client API – HTTP.
3  */
4 package examples;
5 import iptvclienteapi.configurations.Configs;
6 import iptvclienteapi.serverconnection.ServerConnection;
7 import utilities.xml.XML;
8 import utilities.xmlhandler.XMLCommandResponse;
9 /**
10  * @author Joaoffr
11  */
12 public class HTTPExample {
13     /**
14      * HTTP Client Communication API example.
15      * @param args / main arguments
16      */
17     public static void main(String[] args) {
18         String resp = null;
19         String userSessionCode = null;
20         int userSessionId = -1;
```

```

21      /* Create ServerConnection object and teste HTTP connection */
22      ServerConnection httpserver =
23          new ServerConnection("iptvserver.servehttp.com", 2323,
24                               "c35bb76076877a4dcacc06d1dec30300");
25      if (httpserver.isConnectionOK()) {
26          System.out.println("HTTP connection established!");
27      } else {
28          System.out.println("IPTV server seems to be inactive (HTTP)...");
29      }
30      /****** User Authentication *****/
31      resp = httpserver.SendUserAuthentication("joaoffr",
32                                              "512d495f887d7d138f2ef132f59cf27c",
33                                              "93.108.179.170", "Window 7", "Firefox",
34                                              null, "AMD", null, null, "alone");
35      // Parse XML String - extract 'usersessionid' and 'usersessioncode' to use next
36      String comResp = (new XMLCommandResponse(XML.parseXmlString(resp),
37                                               Configs.XML.FILES_PATH)).getData();
38      userSessionCode = comResp.substring(0, Configs.SESSION.CODELENGTH);
39      userSessionId = Integer.parseInt(comResp.substring(Configs.SESSION.CODELENGTH));
40      System.out.println(resp);
41      /****** *****/
42      /** Send Link Request without feedback **/
43      resp = httpserver.SendNewLinkRequest(userSessionId, userSessionCode,
44                                          -1, null, null, 0.0, 0);
45      System.out.println(resp);
46      /****** *****/
47      /******* Send Keep Alive Session *****/
48      resp = httpserver.SendKeepAliveSession(userSessionId, userSessionCode);
49      System.out.println(resp);
50      /****** *****/
51      /******* Send Link List Request *****/
52      resp = httpserver.SendLinkListRequest(userSessionId, userSessionCode,
53                                          "AnyContent", 50, 1, null, "golf");
54      System.out.println(resp);
55      /****** *****/
56      /******* Send Content Feedback *****/
57      resp = httpserver.SendLinkFeedback(userSessionId, userSessionCode,
58                                          5, "alone", "I like it!", 0.86, 319800);
59      System.out.println(resp);
60      /****** *****/
61      /******* User Deauthentication *****/
62      resp = httpserver.SendUserDeauthentication(userSessionId, userSessionCode);
63      System.out.println(resp);
64      /****** *****/
65      }
66  }

```

Referências

- [1] Akamai website, September 2009. [online] <http://www.akamai.com>.
- [2] Atis iptv interoperability forum, September 2009. [online] www.atis.org/iif.
- [3] Bittorrent website, September 2009. [online] <http://www.bittorrent.org/>.
- [4] Castup website, September 2009. [online] <http://www.castup.net>.
- [5] Cisco end-to-end solution for iptv, September 2009. [online] http://www.cisco.com/en/US/solutions/collateral/ns341/ns524/ns610/net_brochure0900aecd80613e32.pdf.
- [6] Digital living network alliance, September 2009. [online] www.dlna.org.
- [7] Dvb fact sheets, August 2009. [online] http://www.dvb.org/technology/fact_sheets/index.xml.
- [8] Espial iptv solution, September 2009. [online] http://www.espial.com/solutions/large_telco/.
- [9] Estados-membros prestes a “desligar” a televisão analógica. IP/09/266, February 2009. [online] <http://europa.eu/rapid/pressReleasesAction.do?reference=IP/09/266&format=HTML&aged=0&language=PT&guiLanguage=fr>.
- [10] Etsi tispan, September 2009. [online] www.etsi.org/tispan/.
- [11] Hp iptv, September 2009. [online] www.hp.com/go/iptv.

- [12] Internet streaming media alliance, September 2009. [online] <http://www.isma.tv/>.
- [13] Iptv explained - part 1 in a bsf series, September 2009. [online] <http://www.broadbandservicesforum.org/images/Pages/IPTV%20Explained.pdf>.
- [14] Iptv focus group, September 2009. [online] <http://www.itu.int/ITU-T/IPTV/>.
- [15] Iptv impact on fulfillment, assurance, and billing, September 2009. [online] http://www.wipro.com/microsoft/pdf_files/IPTV.pdf.
- [16] Iptv news, the 1st source for iptv news, September 2009. [online] <http://www.iptvnews.net/>.
- [17] The open iptv forum – website, September 2009. [online] www.openiptvforum.org.
- [18] Pdtp website, September 2009. [online] <http://www.pdtp.org>.
- [19] Traffic management mechanisms for the support of iptv services, September 2009. [online] <http://www.itu.int/md/T05-FG.IPTV-DOC-0152/en>.
- [20] Working document: Iptv architecture. Focus Group. FG IPTV-DOC-0115, August 2009. [online] http://www.itu.int/dms_pub/itu-t/oth/19/04/T19040000020001MSWE.doc. 51
- [21] Internet Streaming Media Alliance. Planning the future of iptv with isma. *ISMA*, September 2006. [online] http://www.isma.tv/technology/white-papers/ISMA-IPTV_whitepaper_V11_2006-09-14.pdf.
- [22] Christian Bauer and Gavin King. *Java Persistence with Hibernate*. Manning, 2006. 11
- [23] Hervé Benoit. *Digital Television: Satellite, Cable, Terrestrial, IPTV, Mobile TV in the DVB Framework*. Focal Press, 3 edition, February 2008.
- [24] William Cooper and Graham Lovelace. *IPTV Guide – Delivering audio and video over broadband*. Informativ, 1 edition, December 2006.

- [25] Charles Cranor, Matthew Green, and Chuck Kalmanek. *Enhanced Streaming Service in a Content Distribution Network*, volume 5. IEEE INTERNET COMPUTING, 2 edition, July 2001.
- [26] Dvb group – about portugal. DVB Group, July 2009. [online] http://www.dvb.org/about_dvb/dvb_worldwide/portugal/index.xml. 21, 24
- [27] Dvb group fact sheets. DVB Group, July 2009. [online] http://www.dvb.org/technology/fact_sheets. 23, 25, 26
- [28] Dvb-h anacom. Anacom, July 2009. [online] <http://www.anacom.pt/render.jsp?contentId=565874>. 26
- [29] Johan Hjelm (Ericsson). *Why IPTV? Interactivity, Technologies and Services*. Wiley, 1 edition, 2008.
- [30] Tv analógica acaba em abril de 2012. Jornal de Notícias, Feb 2009. [online] <http://ultimahora.publico.clx.pt/noticia.aspx?id=1366825&idCanal=61>. 20
- [31] Xiaojun Hei, Yong Liu, and Keith W. Ross. Iptv over p2p streaming networks: the mesh-pull approach. *IEEE Communications Magazine*, 2008. [online] <http://cis.poly.edu/~heixj/publication/mesh-pull-p2p-streaming.pdf>. 50
- [32] Java hibernate library. Sun Microsystems, July 2009. [online] <https://www.hibernate.org/>. 11
- [33] Merlin Hughes, Michael Shoffner, Maria Winslow, and Conrad Hughes. *Java Network Programming*. O’Reilly, second edition edition, 1999. 11
- [34] Iptv. Wikipedia, August 2009. [online] <http://en.wikipedia.org/wiki/IPTV>. 40
- [35] Ken Kerpez and Dave Waring. Iptv service assurance. *IEEE Communications Magazine*, pages 166–172, September 2006.

- [36] Jay Kishigami. The role of qoe on iptv services. *Ninth IEEE International Symposium on Multimedia*, pages 11–13, December 2007.
- [37] Mysql web page. MySQL, July 2009. [online] <http://www.mysql.com/>. 11
- [38] Gerard O’Driscoll. *Next Generation IPTV Services and Technologies*. Wiley, 2 edition, January 2008.
- [39] H. J. Park and J. H Yang. Qos negotiation for iptv service using sip. *IEEE - Advanced Communication Technology, The 9th International Conference on*, 2:945–948, February 2007.
- [40] Rss 2.0 specification. RSS Advisory Board, July 2009. [online] <http://cyber.law.harvard.edu/rss/rss.html>. xxvi
- [41] Thomas Silverston and Olivier Fourmaux. Measuring p2p iptv systems. *IEEE*, 2007.
- [42] Stations turn off analog signals as digital tv deadline arrives. The New York Times, June 2009. [online] <http://mediadecoder.blogs.nytimes.com/2009/06/12/stations-turn-off-analog-signals-as-digital-tv-deadline-arrives/>. 20
- [43] Sun java. Sun Microsystems, July 2009. [online] <http://java.sun.com/>. 11
- [44] C. M. Sperberg-McQueen Tim Bray, Jean Paoli and Eve Maler. Extensible markup language (xml) 1.0 (fifth edition), November 2008. [online] <http://www.w3.org/TR/REC-xml/>.
- [45] Benjamin Trott. Why we need echo. *Six Apart – News and Events*, 2003-06-29. xx
- [46] Marcio Zapater and Graça Bressan. A proposed approach for quality of experience assurance for iptv. *IEEE ICDS’07*, January 2007.
- [47] Uwe Zdun, Markus Voelter, and Michael Kircher. Design and implementation of an asynchronous invocation framework for web services. *Cover Pages*, 2003. [online] <http://xml.coverpages.org/Kircher-ICWS03.pdf>.

- [48] X. Zhang, J. Liu, B. Li, and T. S. P. Yum. Donet/coolstreaming: A data-driven overlay network for peer-to-peer live media streaming. *IEEE INFOCOM*, 3:2102–2111, Mar. 2005. 50