Instituto Politécnico de Leiria

Departamento de Engenharia Informática

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA INFORMÁTICA COMPUTAÇÃO MÓVEL

Arquiteturas IPTV em Ambientes IMS

IM**SIP**TV

Cláudio Eurico Simões Freire

Setembro 2013

"A pessimist sees the difficulty in every opportunity;

an optimist sees the opportunity in every difficulty."

Winston Churchill

Dissertação de Mestrado realizada sob a orientação do Professor Doutor António Manuel de Jesus Pereira, Professor Coordenador da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria.

Agradecimentos

Em primeiro lugar, gostaria de agradecer ao meu orientador, o Doutor António Manuel de Jesus Pereira, pela disponibilidade, compreensão e auxílio. Por ter tornado possível a realização desta dissertação, e pela orientação que me prestou nos momentos mais complexos.

Gostaria ainda de agradecer às várias pessoas que de algum modo contribuíram para a realização desta dissertação, nomeadamente ao Rui Sábio, Luís Correia e Paulo Jorge, por toda a ajuda prestada e por terem sido a minha companhia nas longas horas de trabalho.

Agradeço ainda ao Instituto Politécnico de Leiria (IPL), pelos meios e condições que colocou ao meu dispor para a realização deste trabalho. Ao Centro de Investigação de Informática e Comunicações e aos seus elementos por me proporcionarem um excelente ambiente de trabalho.

Gostaria ainda de expressar os meus sinceros agradecimentos ao Instituto de Novas Tecnologias (INOV - INESC Inovação), por me ter facilitado a composição desta dissertação por vezes colidindo com o horário de trabalho.

Por fim, gostaria de agradecer à minha mãe, irmão e irmãs, sem os quais nada disto teria sido possível, por todos os sacrifícios que fizeram por mim ao longo desta etapa da minha vida.

Ainda um especial obrigado a todos os meus amigos e à minha namorada Susana Paula Costa que suportaram a minha ausência durante a execução desta dissertação.

O meu muito obrigado a todos.

Nota Prévia

A presente dissertação foi realizada no Centro de Investigação de Informática e Comunicações do Instituto Politécnico de Leiria (CIIC-IPL), em parceria com o Instituto de Novas Tecnologias (INOV - INESC Inovação) delegação na ESTG de Leiria. Do trabalho realizado resultaram as seguintes publicações:

- Cláudio Freire, Luís Correia, Luís Marcelino, Catarina Silva, Carlos Rabadão e António Pereira, "Menu and Context Based Interfaces Evaluation for Mobile TV", in 4th Conference of ENTERprise Information Systems – aligning technology, organizations and people (CENTERIS 2012), Portugal, Outubro de 2012. Este artigo foi convidado para publicação, em versão estendida, na revista International Journal of Web Portals (IJWP).

- Cláudio Freire, Luís Correia, Luís Marcelino, Catarina Silva, Carlos Rabadão e António Pereira, "Defining Interfaces for Mobile TV", publicação submetida à International Journal of Web Portals (IJWP).

- Cláudio Freire e António Pereira, "Arquiteturas IPTV em Ambientes IMS", publicação submetida à 13ª Conferência sobre Redes de Computadores - A Internet do Futuro, CRC'2013, Leiria, Portugal, Novembro de 2013.

Resumo

Com a alteração da difusão dos canais abertos televisivos de analógico para digital, Televisão Digital Terrestre (TDT)[1], inúmeras zonas rurais do país ficaram desprovidas do acesso ao sinal TDT disponibilizado. Estas zonas rurais, normalmente isoladas geograficamente e, por consequência, tecnologicamente, são alvo de descriminação em relação à população concentrada em áreas urbanas. O facto de se encontrarem geograficamente isoladas, e geralmente em zonas de difícil acesso, resulta numa falta de investimento do estado e de empresas tecnológicas, nomeadamente os *Internet Service Providers* (IPS's) [2],[3],[4].

Estas zonas rurais, habitualmente de população reduzida e envelhecida[5], [6], mantêm um contacto limitado com o mundo "exterior", permanecendo limitadas a nível social e pessoal.

Atualmente, várias razões levam os membros ativos do agregado familiar a terem um défice de tempo para cuidar e acompanhar os seus familiares mais velhos. Motivos profissionais ou a necessidade que a classe trabalhadora tem de se deslocar geograficamente na procura de uma melhor qualidade de vida (migração e imigração de populações de faixas etárias mais jovens, por exemplo), resultam, muitas vezes, no isolamento das populações mais envelhecidas que ficam com contacto limitado ou inexistente com a sociedade circundante e com os seus familiares[4].

Assim, e essencialmente devido ao isolamento, surgem soluções que tentam, de alguma forma, resolver ou menorizar o problema. Com o crescimento exponencial de novas tecnologias, estas soluções, na sua maioria projetos-piloto ou apenas teóricos, focam-se na agregação de serviços tecnológicos, suportados por redes *wireless* de âmbito alargado.

Neste âmbito, é proposto o estudo de uma arquitetura *Internet Protocol Television* (IPTV) em ambientes *IP Multimedia Subsystem* (IMS) e o desenvolvimento de um protótipo de acesso a conteúdos multimédia, bem como, a associação de novos

serviços. A agregação de novos serviços providenciará não só multimédia, mas também formas de comunicação bidirecional com o exterior. A solução é intitulada de $^{IM}SIP^{TV}$.

A arquitetura proposta encontra-se dividida em três módulos: servidor, infraestrutura de rede e cliente.

O servidor é interno à rede de âmbito alargado, fazendo a gestão de utilizadores, transmissão de conteúdos multimédia e disponibilização de serviços. Este módulo captura, encripta e retransmite os canais TDT.

A infraestrutura de rede é o elo de ligação e comunicação entre o módulo do servidor e a aplicação cliente.

O módulo do cliente pretende simular uma *set-top box*, reproduzindo conteúdos multimédia disponibilizados e possibilitando o acesso a serviços de comunicação. Sucintamente é uma aplicação que permite aos utilizadores finais interagirem com todos os serviços fornecidos.

A arquitetura proposta foi concebida para uma rede de âmbito alargado, no entanto, pode ser estudada e aplicada noutro tipo de redes e.g.: redes cabladas.

Abstract

With the change in the diffusion of open channels from analog to digital television, Digital terrestrial television (DTT) [1], many rural areas in the country were devoid of access to DTT available signal. These rural areas usually geographically isolated and, therefore, technologically, are subjected to discrimination in relation to the population concentrated in urban areas. The fact that they are geographically isolated, and usually in hard-to-reach areas, results in a lack of investment from the State and from technology companies, including Internet Service Providers (IPS) [2], [3], [4].

These rural areas, usually with a reduced and aging population[5], [6], maintain a limited contact with the "outside" world, remaining limited to social and personal level.

Currently, several reasons lead active members of the household to have a deficit of time to care for and keep up with their older relatives. Professional reasons or the need of the working class to move geographically in the search for a better quality of life (migration and immigration of populations of younger age groups, for example), often result in the aged population isolation that get limited or have no contact with the surrounding society and family members [4].

So, due the isolation, there are solutions that attempt to somehow to overcome the problem. With the exponential growth of new technologies, these solutions, mostly pilot projects or just theoretical focus in technological services aggregation, supported by wide-ranging wireless networks.

In this context, it is proposed the study of a suitable architecture for Internet Protocol Television (IPTV) in IP Multimedia Subsystem (IMS), development of an application prototype to provide access to multimedia content, as well the addition of new services. The aggregation of new services will provide not only multimedia, but also forms of two-way communication with the outside world. The solution is entitled IMSIPTV.

The proposed architecture is divided into three modules: network infrastructure, server and client.

The server is internal to the extended network, managing of users, transmission of multimedia content and available services. This module captures, encrypts and retransmits the DTT channels.

The network infrastructure is the liaison and communication between the server module and the client application.

The client module aims to simulate a customer's set- top box, playing multimedia content and providing access to communication services. Succinctly is an application that allows end users to interact with all the services provided.

The proposed architecture was designed for extended network, however, it can be studied and applied in other types of networks eg: wired networks.

Índice

Lista de Figuras

Lista de Tabelas

Lista de Acrónimos

3GPP ..... 3rd Generation Partnership Project
AAA ..... Autenticação Autorização Accounting
AAC ..... Advanced Audio Coding
AC-3 ..... Digital Audio Compression
ADTS ..... Audio Data Transport Stream
AMR ..... Adaptive Multi-Rate
API ..... Application programming interface
CSS ..... Content Scrambling Systems
DRM ..... Digital Rights Management
DSS ..... Darwin Streaming Sever
DV ..... Digital Video
DVB-T ..... Digital Video Broadcasting-Terrestrial
DVD ..... Digital Versatile Disc
GCC ..... GNU Compiler Collection
IGMP ..... Internet Group Management Protocol
IMS ..... IP Multimedia Subsystem
IP ..... Internet Protocol
IPTV ..... Internet Protocol Television
ISP's ..... Internet Service Providers
JDK ..... Java Development Kit
MP3 ..... MPEG-1/2 Audio Layer 3
MPEG ..... Moving Picture Experts Group
PHP ..... PHP Hypertext Preprocessor
PIM ..... Protocol-Independent Multicast
QoE ..... Quality of Experience
QOS ..... Quality Of Service
QTSS ..... QuickTime Streaming Server
RTP ..... Real-time Transport Protocol
RTP ..... Real-time Transport Protocol

RTSP ..... Real Time Streaming Protocol
SIP ..... Session Initiation Protocol
SOAP ..... Simple Object Access Protocol,
TCP ..... Transmission Control Protocol
TDT ..... Televisão Digital Terrestre
TIC ..... Tecnologias de Informação e Comunicação
UDP ..... User Datagram Protocol
URL ..... Uniform Resource Locator
VOB ..... Video Object
WAV ..... Waveform Audio File Format
WebRTC ..... Web Browser Real time Communication

Capítulo 1 - Introdução

O contínuo desenvolvimento de novas tecnologias potencia novos serviços que seriam de todo inviáveis sem a evolução das Tecnologias de Informação e Comunicação (TIC). Estes novos serviços, maioritariamente de cariz inovador, e aplicados em diferentes contextos, tentam facilitar o dia-a-dia dos seus utilizadores, recorrendo a novas abordagens, minimizando assim o incómodo gerado pelas rotinas diárias.

Tendo em conta o supracitado e aliado ao facto de existir um esforço crescente para a otimização da utilização destas tecnologias, diminuindo entre outras a marginalização o que leva ao surgimento de várias iniciativas de combate à exclusão tecnológica.

As zonas geograficamente remotas são as que apresentam um maior défice de tecnologias de acesso a informação, por existir um desinteresse das entidades responsáveis em investir nessas regiões. Muitas vezes, o investimento não compensa o retorno financeiro que será obtido.

Em Portugal existe o mesmo desinteresse de investimento em locais de difícil acesso, mesmo por tecnologias recém-implementadas. Um caso bem conhecido é a cobertura do sinal TDT que é demasiado fraco ou inexistente, isto é, não cobre todo o território nacional, perdendo o objetivo principal de fornecer televisão gratuita a toda a população.

Tendo o supracitado em mente, os ISP's, aproveitaram a falta de cobertura da TDT para "impingirem" pacotes de serviços televisivos às populações afetadas, normalmente via satélite e apenas com a disponibilização de canais já por si de emissão gratuita.

Uma vez que o país se encontra numa conjetura económica e social extremamente debilitada, é do interesse destas populações, maioritariamente com baixos rendimentos que se encontre uma solução para a emissão de conteúdos televisivos. Para estas populações, muitas vezes a sua única companhia era/é a televisão sendo o único elo de ligação ao mundo que as rodeia.

Obviamente, por si só, o acesso a canais televisivos não resolve a situação de isolamento sentida, pelo que também necessitam do contato com familiares que se

encontram fora e com quem normalmente pouco ou nada comunicam por não existirem formas simples de estabelecimento de contacto.

Uma forma de disponibilização de canais gratuitos é através de uma rede IP que transporte, não apenas os conteúdos multimédia, mas também o fornecimento de mais serviços à população, contribuindo para o combate ao isolamento existente.

Esta solução permite à população o acesso a mais e melhores serviços, os quais são desagregados das operadoras, e ter de facto acesso gratuito aos conteúdos multimédia. Para além da eliminação de custos é clara a mais-valia de ter acesso gratuito à internet e usufruir de serviços internos à rede.

Imediatamente associado ao fornecimento de internet e serviços relacionados vem a diminuição do isolamento das populações já referidas, tendo em conta que podem comunicar mais facilmente o que torna possível uma maior aproximação aos seus respetivos familiares, com quem até então não tinham forma de comunicar assiduamente.

Perante este panorama, seria de grande importância encontrar uma solução que permitisse facultar serviços e conteúdos multimédia gratuitamente, melhorando a qualidade de vida e diminuindo os encargos monetários das populações afetadas.

1.1 Motivação e objetivos

Pretende-se, com esta dissertação, contribuir para melhorar ou resolver esta problemática, através de sistemas que agreguem vários serviços. Para o efeito, propõem-se a definição e avaliação de uma arquitetura de disponibilização de conteúdos multimédia e serviços de comunicação. Esta arquitetura deverá de ser capaz de efetuar *streaming* de vídeo e fornecer serviços de comunicação entre indivíduos bilateralmente. O *streaming* será avaliado apenas em redes locais de âmbito alargado e os serviços de comunicação em redes públicas e privadas. Por fim, desenvolver um protótipo de uma aplicação cliente que receba o *streaming* de dados e permita, em simultâneo, o uso de serviços de comunicação, para o incentivo de uso de tecnologias e aumento de contato com o exterior. Esta aplicação terá de ser de configuração transparente para os utilizadores finais, para que se torne de fácil utilização e não requeira conhecimentos especiais para o efeito.

1.1 Estrutura da Dissertação

Na sequência dos objetivos anteriormente definidos, é apresentado no segundo capítulo o estado da arte, onde são descritas e analisadas as soluções relacionadas á existentes. É efetuado o levantamento de servidores aplicacionais, e tecnologias existentes que possam fazer parte constituinte da solução. Por fim, são apresentadas as tecnologias e servidores aplicacionais selecionados.

No terceiro capítulo é apresentada a arquitetura proposta, ou seja, é definida e apresentada a arquitetura ideal com base nos requisitos obtidos através do estudo realizado no segundo capítulo. É apresentada a arquitetura, em que são expostas as funcionalidades de cada módulo que constituem a mesma.

No quarto capítulo são abordados os protótipos e implementação em que será explicada a implementação de cada um dos módulos detalhadamente e a implementação de um protótipo de uma aplicação cliente.

De forma a apurar se os objetivos desta dissertação foram cumpridos, no quinto capítulo são expostos os testes efetuados à implementação. Estes vão desde a realização de testes a nível funcional da arquitetura implementada, à usabilidade da aplicação cliente desenvolvida.

No sexto capítulo são apresentadas as conclusões relativas ao trabalho realizado, referindo os pontos que podem ser melhorados, bem como, as limitações e dificuldades encontradas.

Capítulo 2 - Estado da Arte

Neste capítulo é apresentado, em primeira instância, um conjunto de soluções relacionadas com o âmbito da dissertação e tecnologias usadas no desenho da arquitetura ideal que propomos no terceiro capítulo. De seguida, é aprofundado o leque de tecnologias candidatas que se relacionam com a resolução da problemática apresentada. Por último, são justificadas as tecnologias selecionadas para a definição da arquitetura final, no terceiro capítulo, e implementação da solução que será apresentado mais à frente neste documento.

Desta forma, pretende-se dar uma visão global das decisões mais importantes tomadas durante o desenvolvimento da solução, assim como, demostrar algumas necessidades às quais o presente trabalho pode dar resposta.

2.1 Soluções Relacionadas

Existem diversos projetos focados no desenvolvimento e definição de tecnologias e *frameworks* para suportarem IPTV e integrar vários serviços para, dessa forma, facultarem uma experiência totalmente nova para o público-alvo. Tendo isto em mente, são abordadas algumas das soluções existentes relacionadas com esta temática, de modo a que a esta proposta seja enquadrada no âmbito destas tecnologias.

Neste seguimento, são apresentados alguns projetos focados na definição e criação de novas formas de utilização de IPTV, as quais serão analisadas como poderão ser interpretadas pelo utilizador final.

Nestes projetos são propostas arquiteturas baseadas em IMS, para facultarem Autenticação, Autorização e *Accounting* (AAA), de forma a controlarem acessos a serviços disponibilizados. Estes serviços que são integrados na mesma aplicação de acesso, têm o objetivo de facultarem mais informação e comunicação entre os utilizadores.

Os serviços abordados nestes projetos são serviços de comunicação bilateral, sendo o caso mais comum a videoconferência, isto é, é permitido aos utilizadores efetuarem chamadas de vídeo através do seu aparelho televisivo para outros clientes que também suportem este mesmo serviço. Deste modo, esta integração possibilita

um uso mais amplo e apelativo dos aparelhos televisivos, para além daquele para que foram inicialmente projetados.

2.1.1 *Towards a New User Experience in IPTV: Convergence Services and Simpler E-commerce on IMS-based IPTV*

O projeto *Towards a New User Experience in IPTV: Convergence Services and Simpler E-commerce on IMS-based IPTV*[7] que defende uma solução IPTV baseada em ambientes IMS, a qual irá proporcionar uma maior versatilidade aos utilizadores. Os autores deste projeto, sugerem a introdução de serviços de videoconferência, *messaging* e *voice-mai* e publicidade. Estes aspetos fornecem uma diferenciação em relação aos serviços comummente proporcionados pelos distribuidores de conteúdos televisivos. Os autores deste projeto apresentam ainda um protótipo da aplicação do lado do cliente, definindo a forma como os serviços seriam apresentados ao utilizador, como se pode observar na Figura 1.

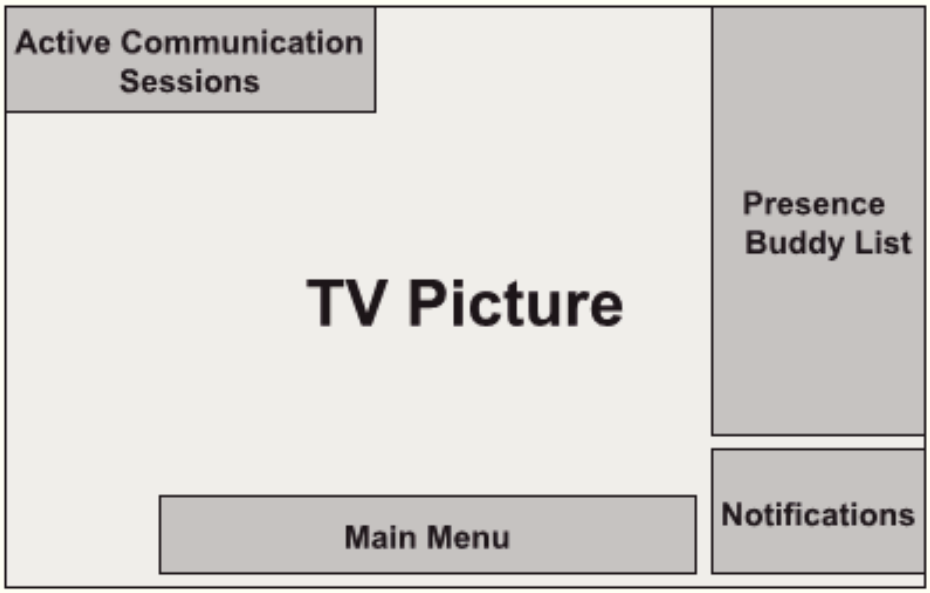


Figura 1 - Interface gráfica da TV[7]

Na Figura 1, podemos observar o protótipo que os autores definiram, em que se distingue as sessões de comunicação ativas e a lista de contatos.

2.1.2 *The Evolvin IPTV Service Architecture*

O projeto *The Evolvin IPTV Service Architecture*[8], da CISCO, apresenta uma solução também baseada em ambientes IMS, apostando no fornecimento de IPTV

com serviços diferenciadores, como o *instant messaging* e videoconferência e gravação remota de conteúdos televisivos. A Figura 2 apresenta um ecrã de uma possível interface gráfica para essa solução.

Figura 2 - Chamada telefónica no ecrã da TV[8]

Observa-se na Figura 2 a visualização de conteúdos televisivos e, em simultâneo, uma chamada telefónica em curso.

Ambos os projetos anteriormente referidos são de extrema semelhança, focando-se nos pontos-chave para a definição de uma nova visão de serviços IPTV para se criar uma experiência mais rica aquando da sua utilização.

Não obstante, as soluções apresentadas de seguida focam-se mais na distribuição dos conteúdos televisivos. No entanto, é facultado um serviço comum a ambos, simples e de utilidade já conhecida: o guia de programação televisiva.

2.1.3 uLikeTV

O projeto uLikeTV[9] tem como principal foco a distribuição de IPTV numa rede *wireless*, em que utiliza como arquitetura base o VLC *Media Player*[10], [11],[12].

No entanto, os conteúdos são acedidos diretamente no *browser* e não numa TV. No entanto, é facultado para além do IPTV, o acesso ao guia de programação televisivo que se pode considerar com um serviço colocado sobre o IPTV. No core da solução, o servidor uLikeTV integra no *browser* o serviço de guia de TV.

A Figura 3 apresenta a interface gráfica (*frontend*) do projeto.

Figura 3 - Interface gráfica *frontend* do projeto uLikeTV[9]

Como se pode verificar na interface, é facultado o serviço de guia de TV integrado na solução de IPTV no qual os utilizadores podem aceder aos seus detalhes. São ainda facultados os controlos básicos para a manipulação da emissão.

2.1.4 myMobileTV

O projeto myMobileTV[13] tem como intuito primário a avaliação de interfaces direcionadas para IPTV por parte de utilizadores em dispositivos móveis. No entanto, a interface oferece um conjunto de serviços aos utilizadores. Os serviços oferecidos, de simples conceção, vão desde o guia de programação, à partilha de informação nas redes socias dos conteúdos *live* que estão a visualizar. Mais concretamente, é oferecido aos utilizadores a possibilidade de: atribuírem uma pontuação aos programas existentes, visualizar a pontuação geral, partilharem informação sobre os conteúdos nas redes sociais, a definição de programas favoritos e, obviamente, a consulta do guia de programação.

Os serviços são acedidos pela aplicação através de *web services*, em que são facultados os canais existentes, guia de programação e outras informações relevantes, mediante a uma autenticação prévia.

A importância deste projeto é o fato de se ter avaliado a forma como os utilizadores se comportam com um superior número de serviços integrados numa solução de IPTV, e ser possível avaliar a sua aceitação perante o número de serviços facultados.

O myMobileTV, utiliza o uLikeIPTV *Streaming Server*, já referenciado anteriormente, como fornecedor de conteúdos televisivos.

Figura 4 - Interface gráfica da myMobileTV

Na Figura 4, podem ser observadas algumas das opções fornecidas aos utilizadores para acesso aos serviços, com os quais podem interagir.

2.2 Tecnologias Candidatas

Neste subcapítulo são abordadas um conjunto de tecnologias candidatas, suscetíveis de seleção para o desenvolvimento da solução, referindo em pormenor o seu modo de funcionamento e possíveis utilizações.

2.2.1 Ambientes baseados em IMS

O IMS é uma arquitetura desenvolvida e definida pela 3gpp[14], [15], [16] , com o objetivo de disponibilizar serviços multimédia através de *Internet Protocol* (IP), ao mesmo tempo que faz o controlo de acesso e gestão dos mesmos, com recurso ao protocolo *Session Initiation Protocol* (SIP)[17][18][19].

O IMS segue um conjunto de *standards* definidos pelo 3gpp. Em que é definida a forma como todos os componentes existentes na arquitetura devem comunicar entre si[15], [16].

Esta arquitetura serve de base a diversas implementações, três das quais apresentaremos com maior pormenor nos subcapítulos seguintes.

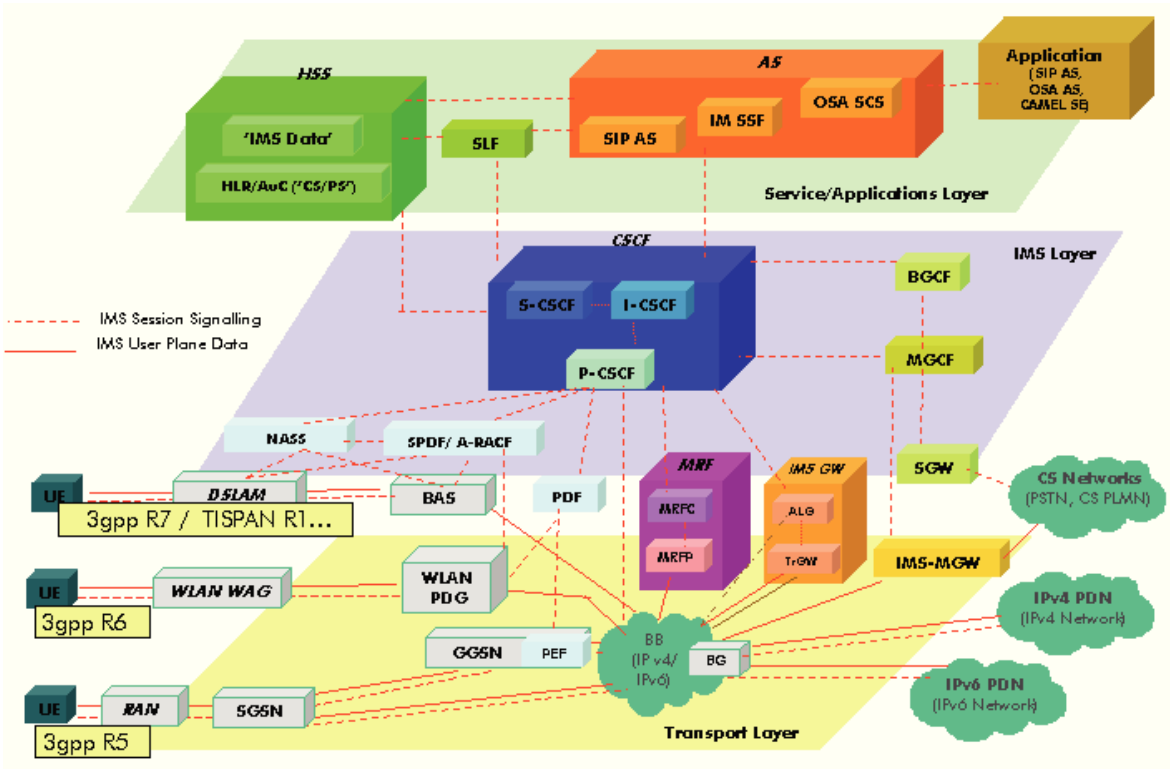


Figura 5 - Arquitetura IMS[20]

Conforme consta na Figura 5, são observados os diversos módulos constituintes de uma arquitetura IMS.

2.2.1.1. Open-Ims Core by Fraunhofer Fokus

O *Open-Ims Core*[21] desenvolvido pela *Fraunhofer Fokus*[22] é um dos exemplos de uma implementação baseada em IMS. Neste sentido, o *Open-Ims Core* utiliza

Call Session Control Functions (CSCFs) para implementar os *standards* definidos pela 3gpp no que diz respeito ao IMS.

De seguida é apresentada a sua arquitetura geral em que é explicado cada um dos componentes principais em maior pormenor.

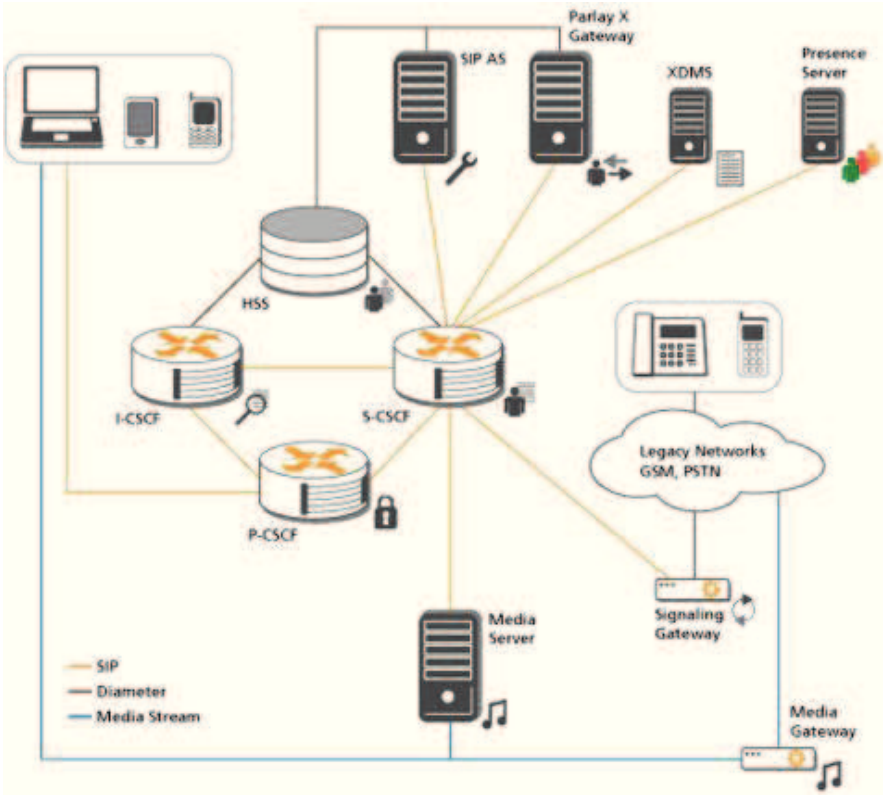


Figura 6 - Arquitetura Open-Ims Core by Fraunhofer Fokus[22]

Analisando a Figura 6, verifica-se que além dos serviços integrantes do núcleo da *framework,* os serão descritos de seguida, existem outros serviços integrados que funcionam como *Application Servers* (AS), ou seja, aplicações externas à plataforma, mas disponíveis para uso dos clientes e serviços integrantes.

De salientar que novos AS podem ser adicionamos à *framework,* estendendo assim o leque de serviços disponibilizados através da solução, o que a torna extremamente modular. É importante realçar que não existe qualquer tipo de limitação (número ou tipo) de serviços que se possam integrar/adicionar na *framework*, o que a torna escalável.

- *Fokus Home Subscriber Server* (FHoSS)

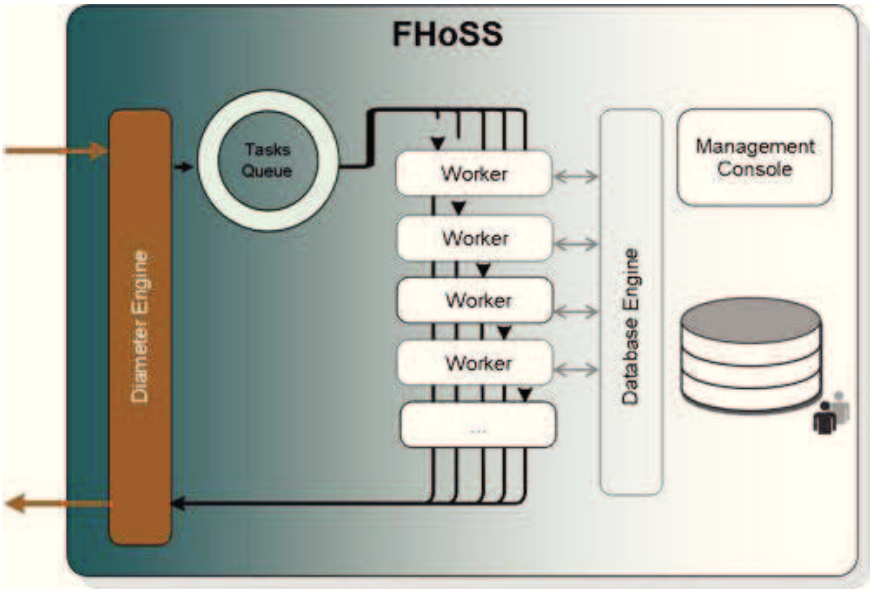


Figura 7 - Esquema FHoSS[23]

Na Figura 7 está representado o FHoSS[23], um dos serviços integrados no core da solução Open-Ims Core, do qual fazem parte uma base de dados MySQL com a informação dos utilizadores e uma interface gráfica web que permite a gestão da informação dos mesmos. É o FHoSS, como consta na figura que recebe os pedidos de autenticação depois de passarem pelos CSCFs, validando se o cliente que se está a ligar tem permissões para tal.

- *Proxy Call Session Control Function* (P-CSCF)

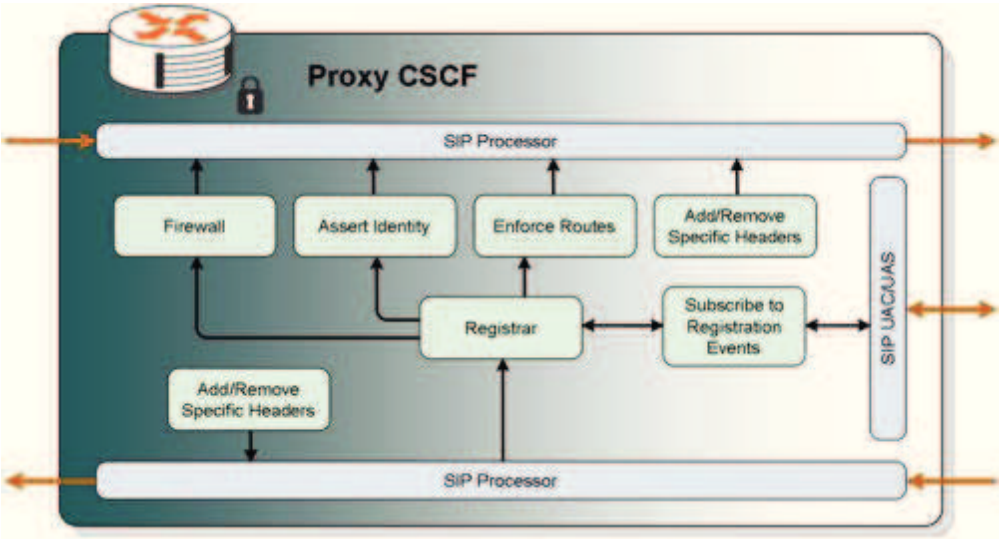


Figura 8 - Esquema P-CSCF[24]

O P-CSCF[24], Figura 8, age como uma *firewall* isolando os restantes componentes core de modo a que só utilizadores registados tenham acesso aos mesmos.

Deste modo é o P-CSCF que recebe a primeira comunicação feita por parte do cliente, processa a mensagem SIP, adiciona-lhe as informações necessárias e reencaminha o pedido para o seguinte serviço.

- *Interrogating Call Session Control Function* (I-CSCF)

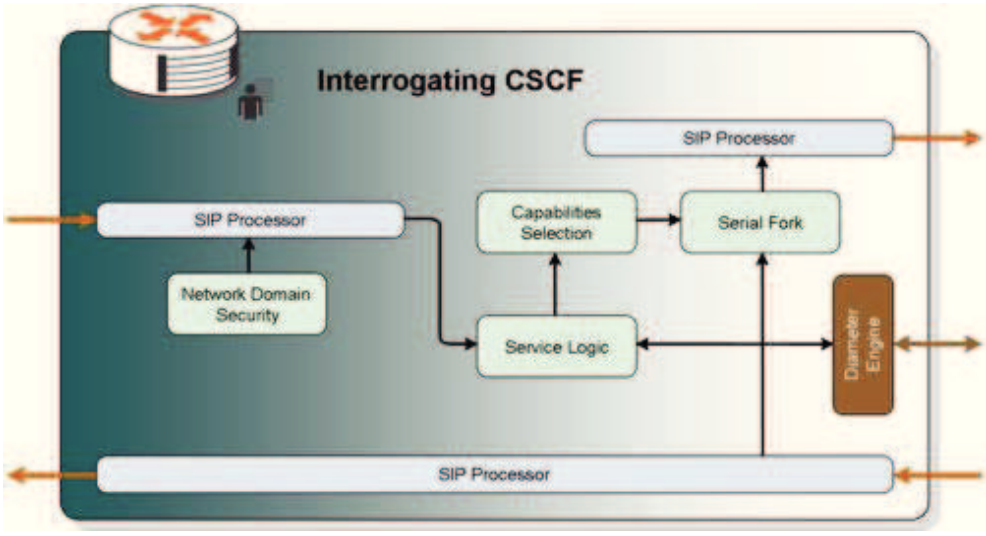


Figura 9 - Esquema I-CSCF[25]

O I-CSCF[25], tal como verificado na Figura 9, recebe o pedido após processamento por parte do P-CSCF e reencaminha o mesmo para o FHoSS para que seja verificada a informação do cliente. Após essa validação, reencaminha o pedido para o *Serving Call Session Control Function* (S-CSCF).

Este módulo é também responsável por reencaminhar as mensagens trocadas internamente entre clientes previamente registados, reencaminhando esses pedidos diretamente para os clientes autenticados a que se destinam.

- *Serving Call Session Control Function* (S-CSCF)

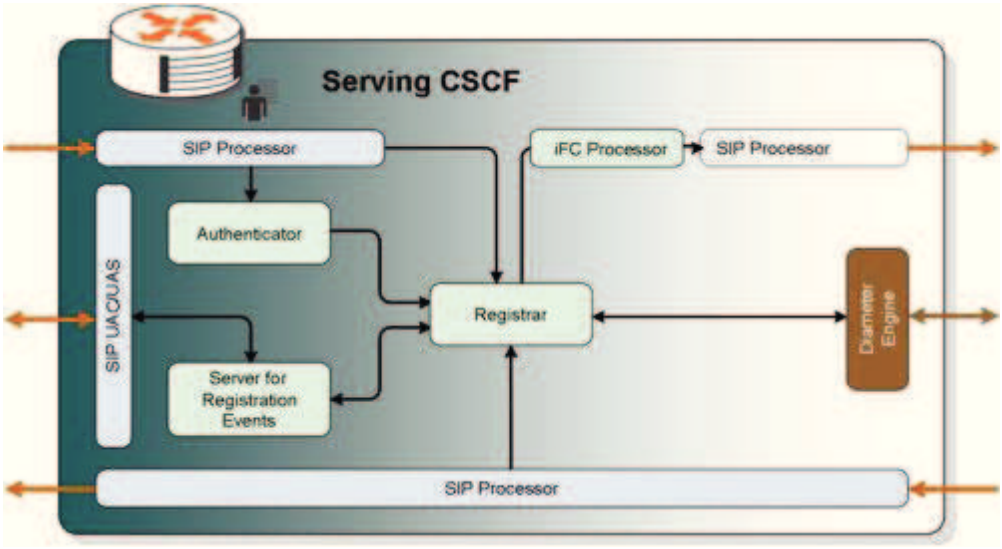


Figura 10 - Esquema S-CSCF [26]

Como atestado na Figura 10, o S-CSCF[26], após a receção do pedido reencaminhado pelo I-CSCF, comunica com o FHoSS com o intuito de receber a informação referente ao utilizador que se está a autenticar e a que serviços o cliente tem acesso, finalizando deste modo o processo de registo de um cliente. É também o S-CSCF que trata dos diversos pedidos aos AS integrados/registados na plataforma.

2.2.1.2. OSIMS

O projeto OSIMS[27] é outra das soluções existentes integrada no âmbito das arquiteturas baseadas em IMS, desenvolvido pela Universidade de Patras – Grécia e tem como objetivo criar uma plataforma de testes baseada na arquitetura Open-Ims

Core, descrita no subcapítulo "*Open-Ims Core by Fraunhofer Fokus*". De seguida, será apresentada a arquitetura da solução OSIMS.

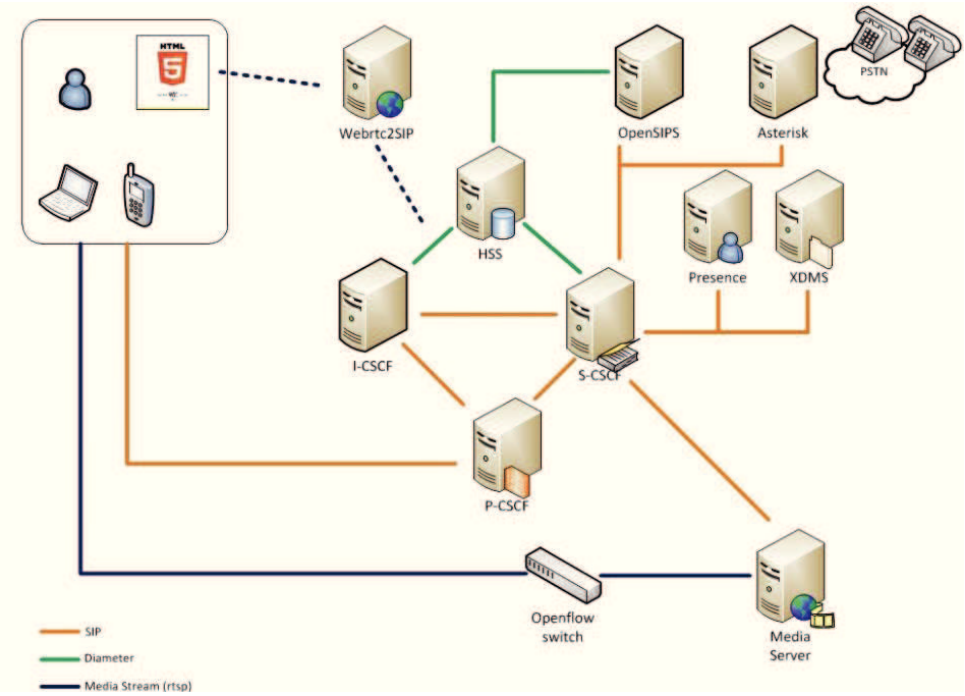


Figura 11 - Arquitetura OSIMS[27]

Como demonstrado na Figura 11, a solução OSIMS, diferencia-se da implementação anteriormente citada, na medida em que acrescenta à arquitetura o conceito de *Web Browser Real time Communication* (WebRTC), no módulo WebRTC2SIP. No entanto, este projeto não se encontra disponível para o público, ou seja, não é *open-source*.

O módulo WebRTC2SIP *gateway* age como uma interface tradutora das mensagens SIP enviadas pelos clientes HTML5/WebRTC, tornando assim possível a comunicação entre estes e a *framework*, já que o P-CSCF não as saberia interpretar.

2.3 *Digital Rights Management* (DRM)

Digital Rights Management (DRM)[30], [31] ou em português, Gestão de Direitos Digitais, é uma tecnologia incorporada em produtos ou serviços eletrónicos com o objetivo de limitar o seu leque de utilização após a aquisição dos mesmos. As DRM foram definidas para que os consumidores finais de produtos tecnológicos não possam utilizá-los de uma forma diferente da estipulada pelos seus fornecedores ou fabricantes.

Esta tecnologia utiliza formas de encriptação, os *Content Scrambling Systems* (CSS) que têm como principal objetivo prevenir a cópia *byte-for-byte* de *streams* de vídeo, áudio, etc., de forma a não ser possível a visualização sem a chave certa.

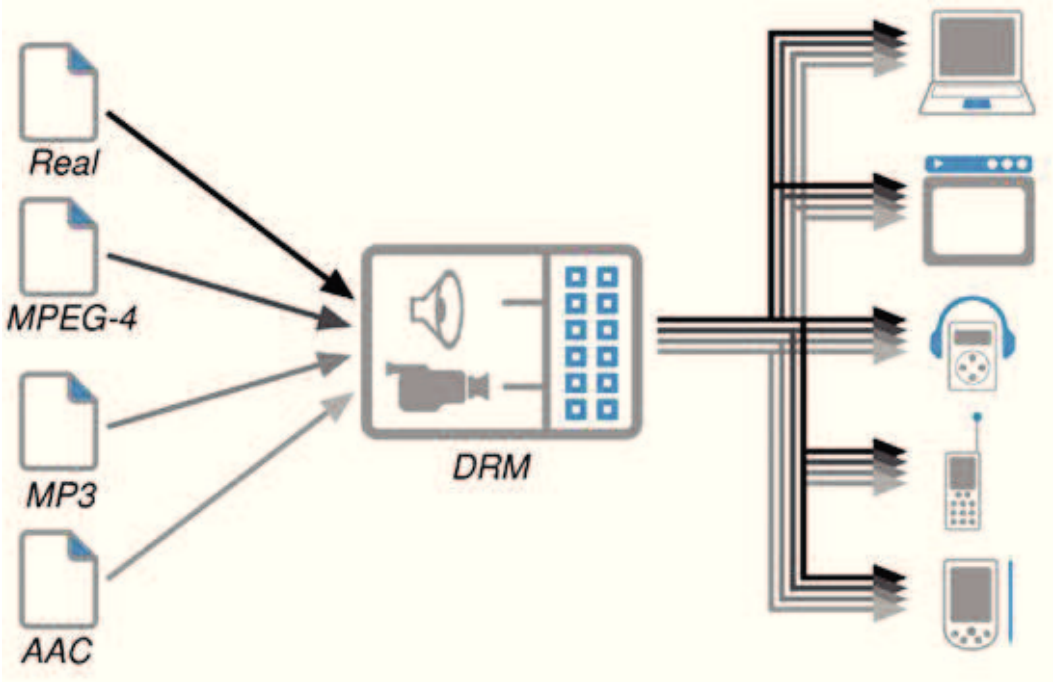


Figura 12 - Arquitetura DRM[32]

Na Figura 12 é apresentada uma arquitetura genérica das DRM, em que os dados são recebidos e encriptados pelo conjunto de funcionalidades representadas por "DRM" na figura. A encriptação é feita internamente pelos CSS, após a qual, os dados são enviados e desencriptados por dispositivos ou aplicações terminais próprias para o efeito.

2.4 Servidores Aplicacionais IPTV

Nesta seção são abordados alguns dos servidores aplicacionais passíveis de serem utilizados nesta dissertação.

2.4.1 uLikeTV *Streaming Server*

O servidor aplicacional uLikeTV[9] *Streaming Server* foi desenvolvido no âmbito do projeto informático de final de curso da licenciatura em Engenharia Informática. Este servidor tem como base o VLC[10],

O servidor captura e descodifica o sinal de TDT através de um dispositivo DVB-T[33], posteriormente, os canais televisivos disponibilizados pela TDT são então, após a captura, capturados e enviados pela rede, com recurso ao protocolo *Multicast*, para diminuir a sobrecarga na rede. Para além da funcionalidade descrita, servidor ainda possui uma interface de gestão web que permite controlar todo o serviço, consultar dados estatísticos, obter informações de acesso e constrói, através de dados do sinal TDT capturado o Guia de Programação Eletrónico que armazena numa BD apenas para esse propósito.

O uLikeTV é de instalação e configuração automática apresenta um módulo de gestão web, em que são facultadas ferramentas para os processos mais comuns a serviços ou servidores aplicacionais.

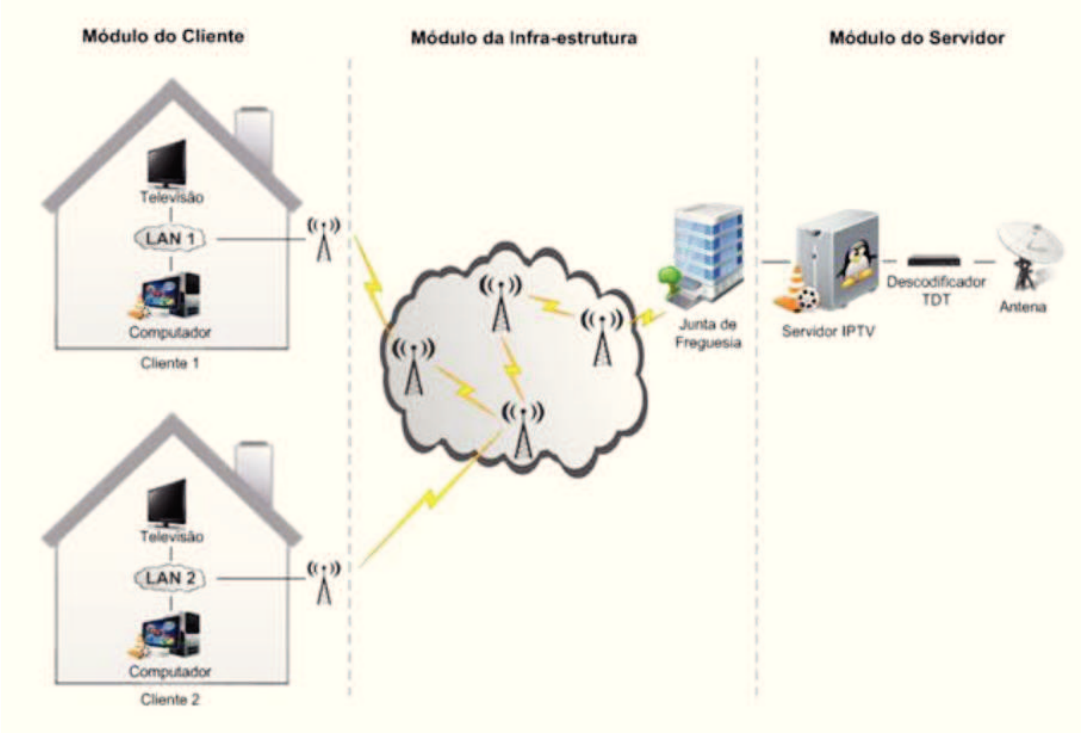


Figura 13 - Arquitetura do Servidor Aplicacional uLikeTV[9]

Na Figura 13 podemos observar a arquitetura geral do servidor e as componentes mais importantes do módulo, no qual a captura do sinal TDT é feita através de um descodificador e posteriormente os *streams* são enviados pela rede para todos os clientes.

2.4.2 *QuickTime Streaming Server* (QTSS) e *Darwin Streaming Server* (DSS)

O QuickTime *Streaming Server* (QTSS)[34] e o Darwin *Streaming Server* (DSS)[35][34], fazem parte da QuickTime *Suite*. Estes, permitem o envio de *streams* de multimédia em tempo real pela internet. Os utilizadores podem aceder a *live* broadcasts ou a conteúdos pré-gravados, ou ainda a *video-demand*. O envio destes *streams* pode ser efetuado via *Multicast* ou *Unicast*.

As funcionalidades presentes no QTSS e pelo DSS são:

- *Streaming* nativo de MPEG-4;
- *Streaming* de áudio no formato MP3;
- Administração baseada em Web;
- Qualidade de Serviço (QOS);
- *Streaming* de *playlists*, ou seja, envio de um conjunto de *streams*.
- Autenticação, para controlo de acessos e proteção dos *streams* de multimédia;
- Suporte virtualmente ilimitado de clientes, devido ao uso de *Relay*.

O DSS, multiplataforma, atualmente já não se encontra em desenvolvimento, tendo sido a sua última versão lançada em maio de 2007.

O QTSS difere do DSS uma vez que não é multiplataforma e esteve em uso e em desenvolvimento até julho de 2011, altura em que foi descontinuado pela Apple.

2.4.3 Live555 *Streaming* Media

O Live555 *Streaming Media*[36] agrega um conjunto de bibliotecas em C++ para *streaming* de multimédia em que são usados os *standards* RTP, RTSP e SIP. Este é constituído por uma gama de soluções definidas através das bibliotecas referidas.

As aplicações constituintes do *Live555 Streaming Media* são o *Live555 Media Server*[37], *Live555 Proxy Server*[38], *liveCaster*[39], *vobStreamer*[40]. De seguida, abordamos individualmente cada uma delas.

- **Live555 *Media Server***

O Live555 *Media Server* é um servidor aplicacional completo que suporta um vasto tipo de ficheiros multimédia e *codec's*, nos quais se incluem o *Moving Picture*

Experts Group (MPEG), *Matroska*, WebM, MPEG, MPEG-1, MPEG-2, MPEG-4, H.264 (MPEG-4 Parte 10), *Video Object* (VOB), *Digital Vídeo* (DV), MPEG-1 e 2 (layer III, MP3), WAV, AMR, *Advanced Audio Coding* (AAC) e *Digital Audio Compression* (AC-3).

- **Live555 *Proxy Server***

O Live555 *Proxy Server* é um servidor RTP *unicast* que atua como um *proxy* para um ou mais *streams unicast* ou *multicast* disponibilizados por outros servidores aplicacionais de *back-end*. A principal vantagem deste servidor é a interpretação de cada *stream* individual de *back-end* uma única vez, independentemente do número de clientes que serve, como demonstrado na Figura 14.

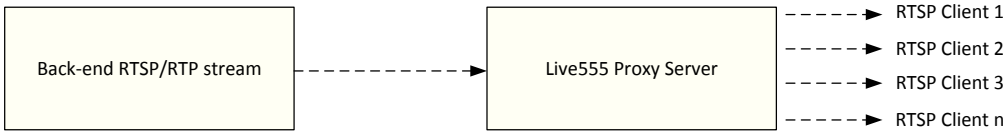


Figura 14 - Arquitetura do Live555 *Proxy Server*

- **liveCaster**

O liveCaster é uma aplicação que permite o *streaming* de mp3 via *multicast*, para uma "audiência" quase ilimitada, mesmo para quem possua uma ligação à internet de baixa largura de banda. Esta aplicação está disponível para Unix e Windows, com ou sem ambiente gráfico.

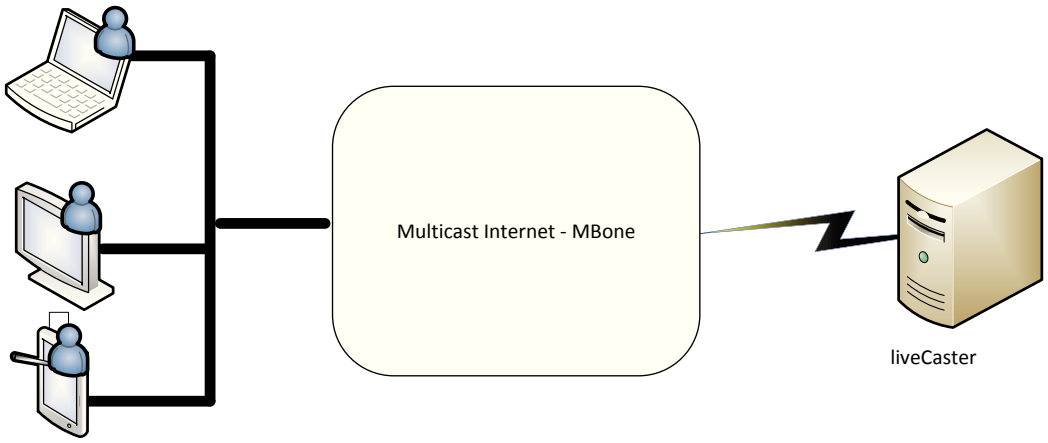


Figura 15 - Arquitetura liveCaster

Como se pode observar na Figura 15, o liveCaster envia o *stream* mp3 via *multicast* para o MBone[41], através do qual os clientes acedem ao *stream* disponibilizado.

O acesso ao *stream pode* ser feito através de um *player* com capacidades para leitura de mp3, via *multicast*, tal como o Audioactive, Winamp, entre outros.

- ***vobStreamer***

O *vobStreamer* é um programa sem interface gráfica que interpreta o áudio e vídeo de ficheiros vob, por exemplo, de um DVD, e os envia, via *multicast,* para uma rede local, sem fios ou cablada, usando os *standards* RTSP e RTP, como exemplificado na Figura 16.

Uma nota importante sobre esta aplicação é o fato de esta não ter o intuito de permitir o *stream* de DVD's comerciais já que para além destes estarem encriptados (DRM's), esta aplicação não possui a capacidade de desencriptação de dados multimédia.

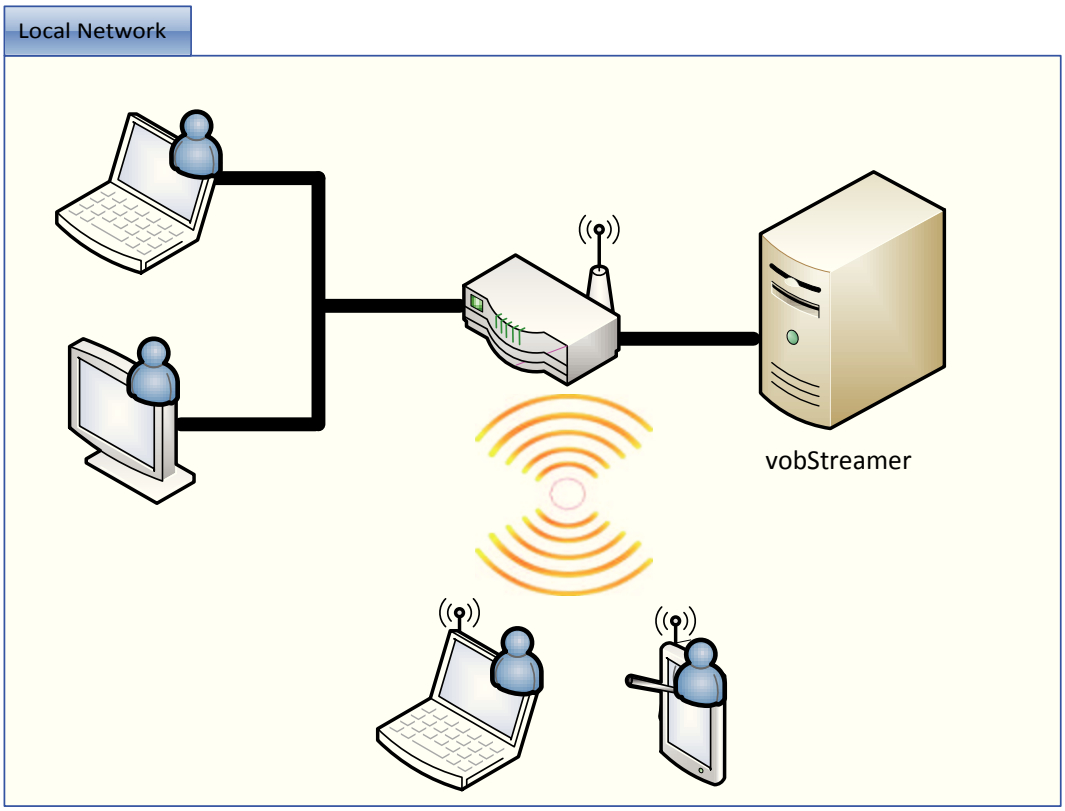


Figura 16 - Arquitetura vobStreamer

2.4.4 VLC *Media Player*

O VLC *Media Player*[10], [11],[12] é um reprodutor multimédia gratuito, *open-source* e multiplataforma que suporta a maioria de ficheiros multimédia, vários dispositivos de entrada e vários tipos de protocolos de *streaming*.

Este reprodutor possui características de uma aplicação final (lado do utilizador) que permite visualizar multimédia. No entanto, uma das fortes vantagens deste reprodutor é a sua capacidade de atuar como um servidor aplicacional de *streaming* que permite a captura/leitura de multimédia de vários tipos de dispositivos de entrada (DVB-T, Ficheiros, DVD's, etc[1]), assim como, a sua retransmissão/output em vários formatos[2].

O VLC é uma solução cliente-servidor extremamente completa, já que para além do supracitado, permite a transmissão e receção de multimédia transversalmente aos sistemas operativos suportados, como se pode verificar na arquitetura representada na Figura 17.

Este leitor possui ainda uma vasta gama de opções, nomeadamente:

- Encriptação e desencriptação[42] pelo uso de CSS (*Content Scrambling System*);
- Escolha entre várias ou nenhuma interface gráfica;
- Suporte para envio e receção de multimédia, via IPV6;
- *Streaming* de conteúdos de qualquer *input* suportado diretamente para a rede, através de *multicast* ou *unicast*, usando protocolos *standards*, como o RTSP, RTP.

[1] Consultar Anexo II
[2] Consultar Anexo II

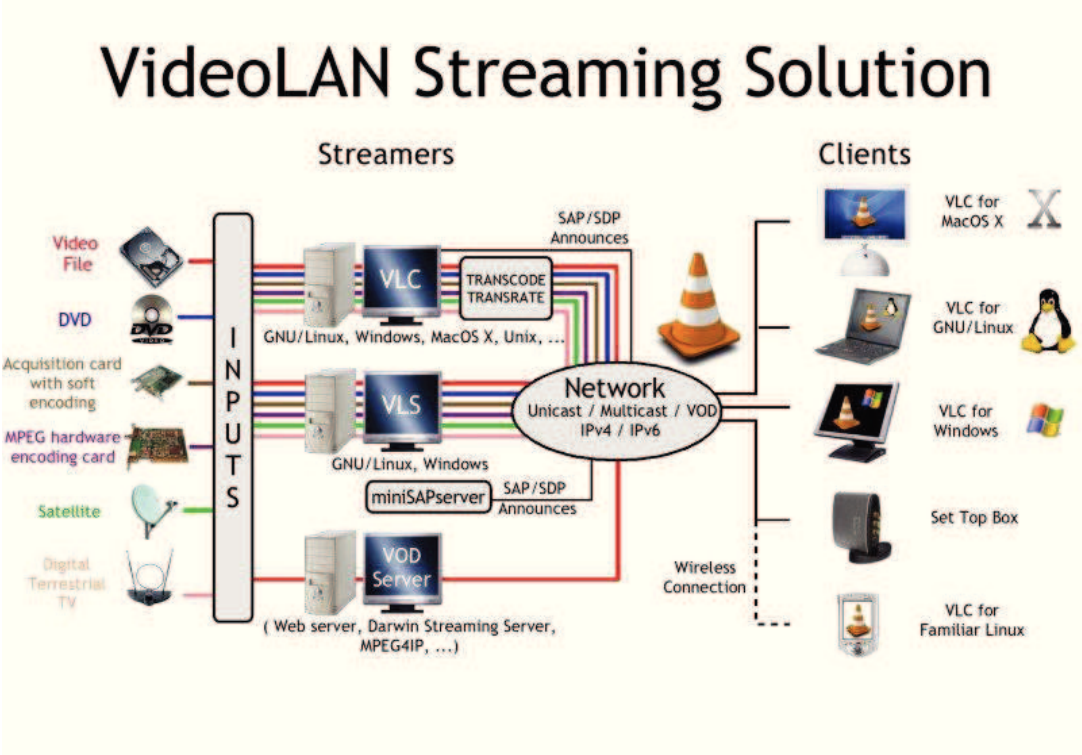


Figura 17 - VideoLan *streaming solution*[11]

Na Figura 17 pode verificar-se que o VideoLAN/VLC *Media Player* é uma aplicação cliente-servidor extremamente versátil, multiplataforma e que suporta uma vasta gama de *inputs* e *outputs*.

2.5 Protocolos

Nesta secção são abordados os protocolos estudados no âmbito desta dissertação para a sua definição e implementação.

2.5.1 *Real Time Streaming Protocol* (RTSP)

O RTSP[43] é um protocolo ao nível da camada de aplicação que fornece controlo sobre dados entregues em tempo real e permite a seleção de canais para entrega de conteúdos, UDP, *Multicast* UDP e TCP, oferecendo ainda mecanismos de entrega baseados em RTP.

2.5.2 *Real-time Transport Protocol* (RTP)

O RTP[44] é um protocolo que permite a entrega de dados de serviços em tempo real, como áudio e vídeo. Estes serviços incluem o tipo/identificação, numeração da sequência, *timestamp* e monotorização da entrega de dados. As aplicações utilizam

tipicamente o RTP sobre UDP para fazer uso da sua multiplexagem e serviços de *checksum*. O RTP suporta o envio de dados para múltiplos destinos quando usado com *multicast* e se a rede em questão o facultar.

2.5.3 *IP MULTICAST*

O IP *Multicast*[45],[46],[47] ajuda a cumprir a exigência de aplicações ou serviços baseados em comunicações de um para muitos ou, de muitos para muitos, e que necessitam de uma alta largura de banda, como, por exemplo, a transmissão de vídeo.

Com o IP *Multicast* é possível enviar dados de um servidor para vários clientes, sem que se verifique um aumento exponencial de tráfego na infraestrutura de rede. Isto permite o não congestionamento e sobrecarga na rede, diminuindo desta forma o impacto causado. A Figura 18 representa o caso referido, em que o conteúdo disponibilizado pelo *Server* está a ser acedido por quatro utilizadores sendo, no entanto, utilizado apenas um canal para a disponibilização dos conteúdos.

O IP *Multicast* em si, é composto por um conjunto de protocolos de *routing*, o *Internet Group Management Protocol* (IGMP) e o *Protocol-Independent Multicast* (PIM) que pode operar em três modos distintos, como iremos abordar de seguida.

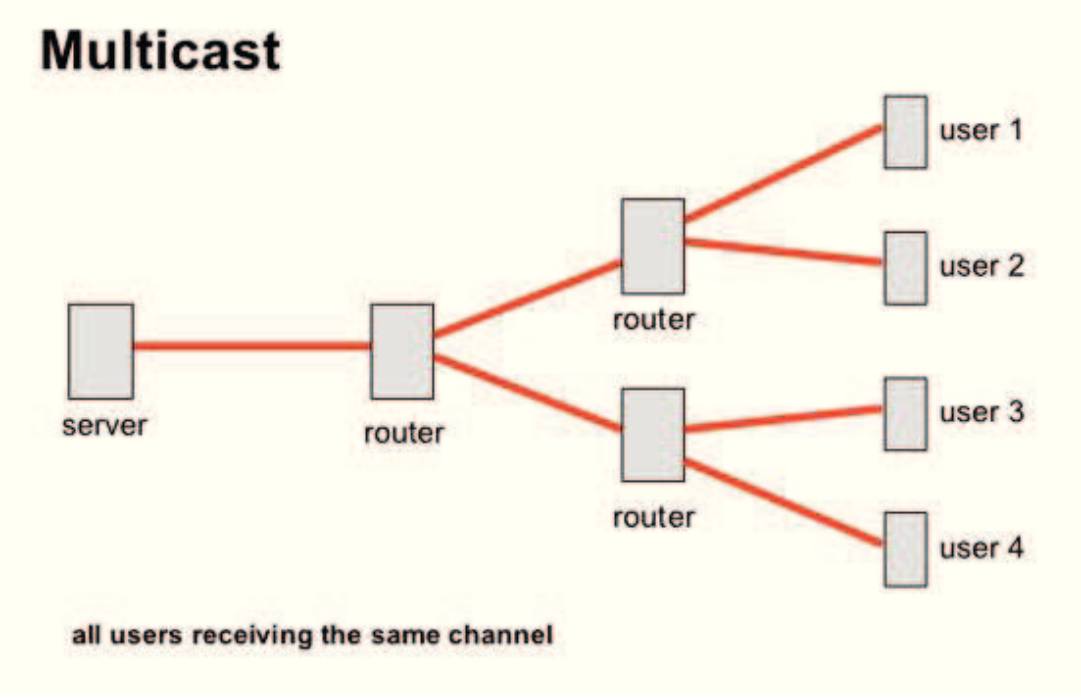


Figura 18 - Funcionamento do *Multicast*[48]

2.5.4 *Multicast Routing Protocols*

Como referido anteriormente, o IP *Multicast* utiliza protocolos de encaminhamento *multicast*[46], o IGMP e o PIM que passamos a descrever:

- O IGMP é utilizado pelos *hosts* para reportarem aos *routers* adjacentes um grupo *multicast.*
- O PIM[46],[49] permite o uso de *multicast* em redes existentes, nas quais pode operar em três modos distintos: dense, *sparse* ou o *sparse dense mode*.

O PIM *dense mode*[50] é orientado aos dados em que todos os pacotes de dados são enviados para todas as interfaces de saída. O custo de utilizar este modo prende-se com o facto de o seu comportamento natural ser o envio maciço de pacotes e provocar um excedente de dados.

O PIM *sparse mode*[51] tenta limitar a distribuição de dados de forma a minimizar o número de *routers* que os recebem. Os pacotes apenas são enviados quando pedidos por um *router*, designado de "*rendezvous point*" (RP). Por seu lado, os recetores estão distribuídos pela rede para que não exista a necessidade de duplicação de *streams* pela rede. O custo de utilizar este modo está relacionado com o facto de ser necessária uma atualização periódica pelo envio de mensagens *join*. No entanto, este é mais escalável que o *dense mode*

O PIM *sparce dense mode* é um modo híbrido de operação que oferece flexibilidade para fornecer vários canais para o envio de dados, em qualquer um dos modos. Este modo é o recomendado para utilização devido à sua versatilidade.

2.5.5 UNICAST

O *Unicast* é uma ligação de um para um (cliente e servidor), ou seja, para cada pedido ao servidor, é criado um canal de transmissão de dados novos. Este método é ineficiente para a transmissão de dados coletiva, isto devido ao fato da infraestrutura de rede ficar congestionada se o número de ligações ou pedidos ao servidor for elevado. A Figura 19 apresenta o modo de funcionamento do *Unicast*, pelo que se pode verificar que para cada pedido de um utilizador é criado um novo canal entre o servidor e o cliente respetivo.

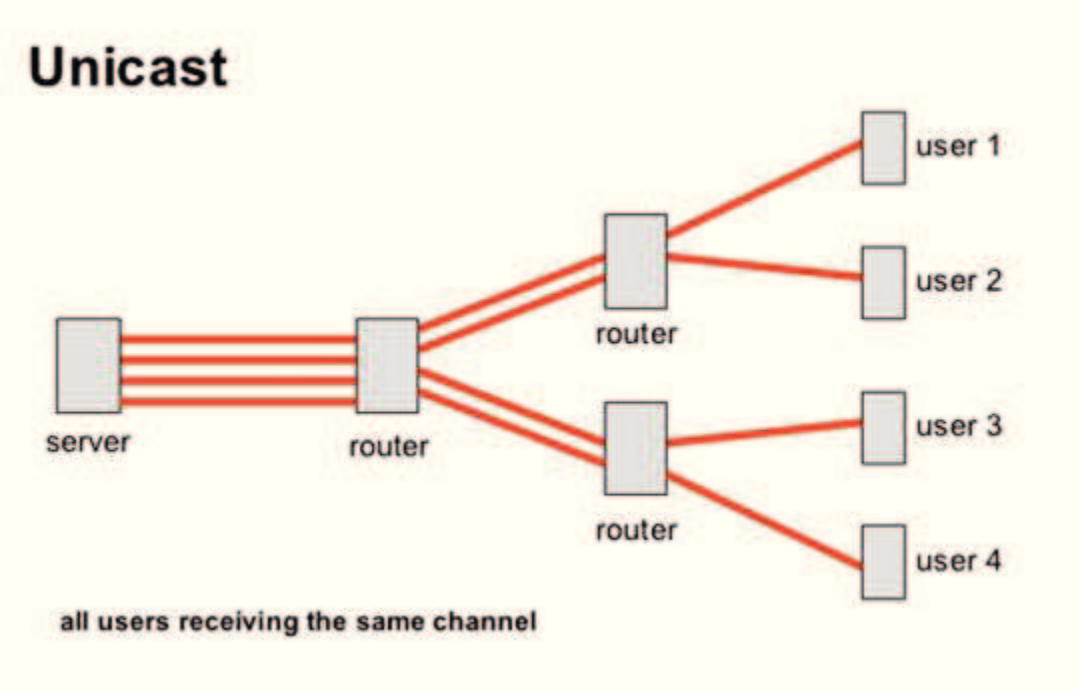


Figura 19 - Funcionamento do *Unicast*[48]

2.5.6 *Session Initiation Protocol* (SIP)

O SIP[17], [18], [19][52] afirma-se como um protocolo de sinalização que utiliza mensagens de controlo para mediar comunicações interativas entre utilizadores, como, por exemplo, videochamadas efetuadas através da rede IP (ver Anexo I).

2.6 *Application Programming Interfaces* (API's)

Nesta secção são abordadas as API's estudadas no âmbito da dissertação. A seleção e estudo de determinadas API's focou-se essencialmente na sua capacidade de fornecer uma estrutura de desenvolvimento rápida e fléxivel. Nesse sentido, serão abordadas API's de captura de *stream* de vídeo e de SIP.

2.6.1 *VlcDotNet*

A VlcDotNet API ou biblioteca, em .net, fornece controlo sobre áudio e vídeo através de diferentes implementações, para *Windows Forms e Windows Presentation Foundation* (WPF), possuindo para ambas o mesmo suporte de formatos que o VLC.

Com a API para *Windows Forms*, é possível criar aplicações *Desktop*, em que o seu *design* ou apresentação é bastante familiar ao utilizador comum.

Com a API para *WPF*, é possível criar aplicações *Desktop*, com um *design* bastante atrativo e interativo. O WPF é a nova geração em aplicações *Desktop* para *Windows*, introduzindo novos efeitos visuais para o utilizador e uma experiência de utilização totalmente nova.

2.6.2 LumiSoft

A *Lumisoft* é uma API que fornece uma implementação SIP para *Windows*, baseada em .net. Esta implementa métodos de autenticação SIP, estabelecimento de videochamada e envio de mensagens instantâneas. Os autores desta API, criaram uma pequena aplicação para testes e facilitar a compreensão sobre ela. A aplicação, desenvolvida em *Windows Forms,* permite o estabelecimento de uma videochamada entre dois clientes, via autenticação SIP.

2.7 Tecnologias Selecionadas

Após a análise das várias tecnologias candidatas, apresentamos de seguida o conjunto de tecnologias selecionadas para o desenvolvimento do protótipo.

- Open-Ims Core by Fokus Fraunhofer

  Como explicado anteriormente, o Open-Ims é uma implementação baseada em IMS que permite controlo de utilizadores e disponibilização de serviços. Esse fato agregado às funcionalidades de segurança, asseguradas pelo uso do protocolo SIP, partes integrantes da implementação, oferecem uma comunicação segura garantindo a proteção dos dados dos utilizadores. De salientar que pesou também o fato de ser *open-source*.

- RTP

  O RTP suporta o envio de dados para múltiplos destinos quando usado com *multicast*, permitindo um uso mais controlado dos recursos de rede, utilizando os mesmos canais para a distribuição de dados para destinos diferentes.

- Multicast

  Como mencionado, o *multicast* permite um uso mais controlado dos recursos de uma rede, pelo que é protocolo selecionada para uso na solução.

- uLikeIPTV *Streaming Server*

  Como foi referido, este servidor aplicacional tem como base o VLC que possui funções importantes para o desenvolvimento da solução pretendida, como as DRM's e um suporte extenso de protocolos de *streaming*. O uLikeIPTV já efetua *streaming* via *multicast*, constrói o guia de programação eletrónica para disponibilizar posteriormente e faculta ainda uma interface de gestão web para controlar todo o servidor aplicacional.

- Linguagens de programação de API's para aplicação cliente, C#

  A seleção da linguagem de programação deveu-se sobretudo a uma escolha pessoal e na capacidade de desenvolvimento rápido com esta linguagem de programação

  - VlcDotNet – WPF

  Foi selecionado a versão do WPF do VlcDotNet pelo fato de disponibilizar uma interface mais interativa e fluída para os utilizadores finais e por suportar todas as funções de controlo existentes no VLC.

- LumiSoft

  Como foi referido anteriormente, esta API disponibiliza uma aplicação de testes que permite um rápido entendimento do seu funcionamento interno, fator que pesou na sua escolha.

2.8 Síntese

Em suma, neste capítulo apresentámos um conjunto de trabalhos relacionados com o âmbito da dissertação e que têm como objetivo facultar IPTV e serviços integrados (vídeo chamada, publicidade, *chat*), permitindo ainda a adição de um conjunto não quantificável de serviços.

Foram igualmente estudadas diversas tecnologias, com o intuito de escolhermos as mais indicadas para o desenvolvimento da solução proposta e dar resposta às nossas necessidades.

Capítulo 3 - Arquitetura

Neste capítulo são abordados de forma mais pormenorizada as várias arquiteturas pensadas para o funcionamento da solução e a sua implementação. Apresentando com maior pormenor todos os módulos integrantes de cada uma delas, explicando em cada ponto o conceito por detrás de cada arquitetura.

3.1 Arquitetura Geral

Tendo em consideração a problemática supracitada, foi definida uma arquitetura ideal como base para o desenvolvimento de uma solução que responda às necessidades identificadas no capítulo 1 - Introdução.

A Figura 20 mostra que a arquitetura idealizada constituída por dois módulos distintos, o Módulo Servidor e o Módulo Cliente.

O Módulo Servidor é constituído por três módulos distintos: Gestão Open-Ims e Gestão IPTV e Gestão Web Comum. Estes módulos internos têm como função captura e *streaming* de dados, autenticação, gestão de conteúdos e controlo.

O Módulo Cliente permite a receção de dados multimédia e comunicação entre clientes, em que é definida uma aplicação cliente, transversal a sistemas operativos e *hardware*, de forma a simular uma *set-top box.*

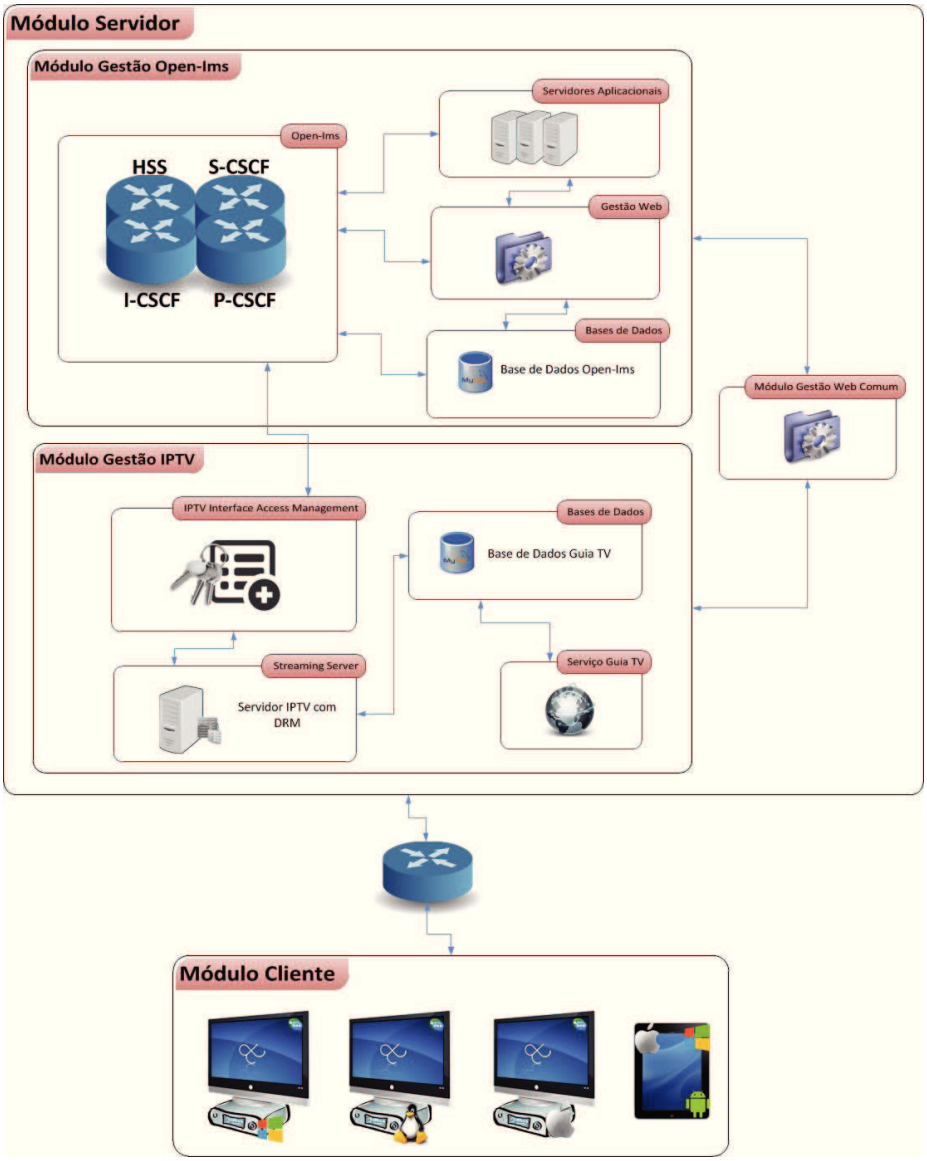


Figura 20 - Arquitetura Geral

No Módulo Cliente estão apresentados todos os sistemas operativos suportados e tipos de dispositivos. É apresentada ainda a estrutura interna de rede na qual existe um ponto de distribuição interno (LAN), existindo uma ligação direta ao equipamento terminal (*set-top box*). O módulo servidor encontra-se subdividido em três módulos distintos: Gestão Open-Ims e Gestão IPTV e Gestão Web Comum que iremos descrever de seguida.

3.1.1 Módulo Gestão Open-Ims

O Módulo de Gestão Open-Ims, baseado numa arquitetura Linux, incorpora vários servidores aplicacionais, serviços disponibilizados pela solução, bases de dados associadas e um sistema de gestão para a gestão do módulo em si.

Na Figura 21, é apresentada a arquitetura do Módulo Gestão *Open-Ims*.

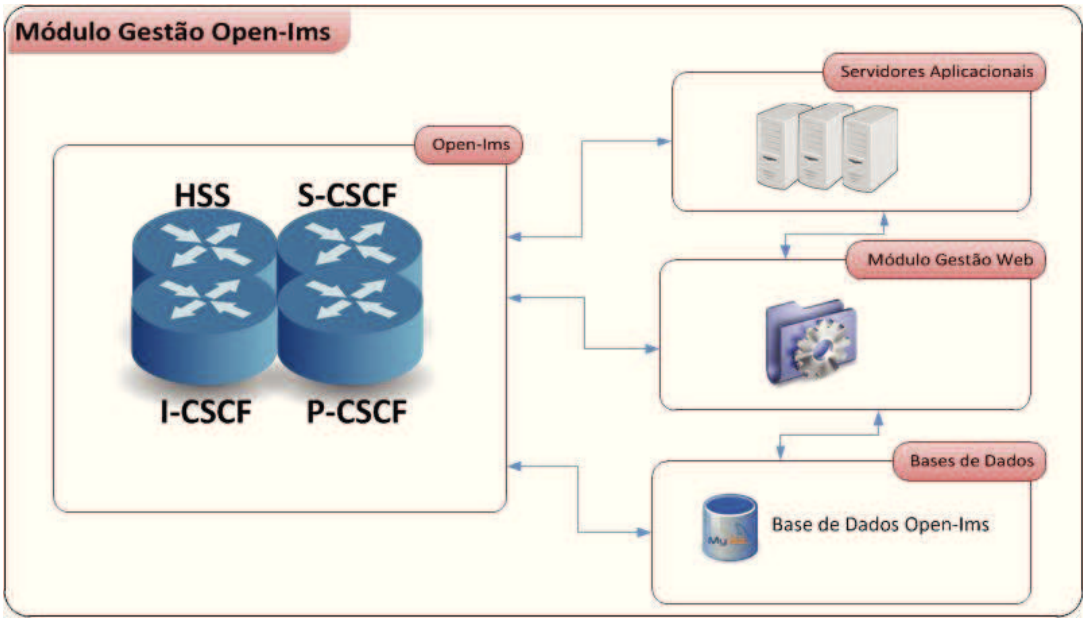


Figura 21 – Arquitetura Módulo Gestão Open-Ims (Arquitetura Geral)

Este módulo incorpora diversas funcionalidades, tais como, o serviço Open-Ims descrito no capítulo 2. Incorpora também as bases de dados necessárias à solução, os servidores aplicacionais cuja função é disponibilizar o conjunto de serviços pré-definidos e o módulo de gestão *web* que centraliza o controlo do serviço.

3.1.2 Módulo Gestão IPTV

O Módulo Gestão IPTV disponibiliza os *streams* de vídeo para emissão, uma base de dados associada ao à disponibilização do Guia TV, e uma interface de gestão de acessos aos *streams*.

No esquema da Figura 22, está representada a arquitetura do Módulo Gestão IPTV.

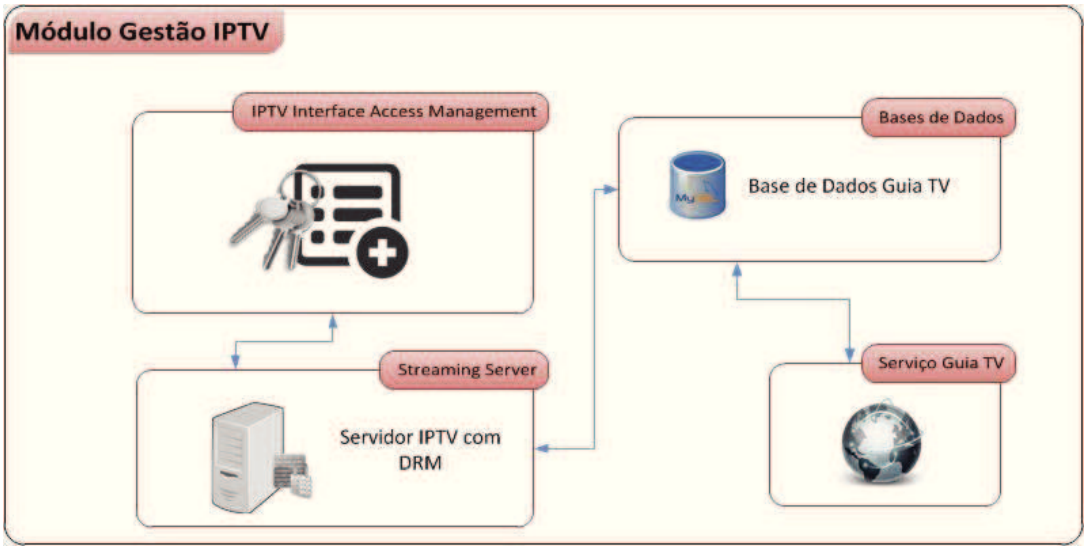


Figura 22 - Arquitetura Módulo Gestão IPTV (Arquitetura Geral)

Este módulo define todas as funcionalidades necessárias para a solução, este é responsável pelo envio dos *streams* de vídeo, sua encriptação via DRM's, criação, armazenamento e disponibilização do Guia TV e é onde é processada a gestão de acessos aos canais disponíveis e acesso à chave de desencriptação dos mesmos.

3.1.3 Módulo de Gestão Web Comum

O Módulo de Gestão Web Comum agrega todas as funcionalidades e controlos sobre os módulos de Gestão Open-Ims e IPTV.

3.1.4 Módulo Cliente

No Módulo Cliente são representadas várias plataformas físicas sobre as quais a plataforma seja funcional, assim como os diversos sistemas operativos onde a aplicação cliente permita aos utilizadores aceder ao leque de serviços oferecidos.

3.2 Arquitetura Implementada

De forma a provar o conceito em tempo útil, foi especificada uma arquitetura com o intuito de implementar um protótipo funcional que comprove o conceito idealizado e apresentado no capítulo Arquitetura Geral. As diferenças entre a arquitetura anteriormente definida e a implementada é a inexistência do Módulo de Gestão Web comum, no Módulo Servidor, e na redefinição do módulo cliente em que foi necessário, criar uma aplicação para um *Andoid tablet* de forma a disponibilizar o

controlo total sobre a aplicação cliente desenvolvida e a definição de rede local no contexto do cliente.

Na Figura 23, é apresentada a arquitetura implementada.

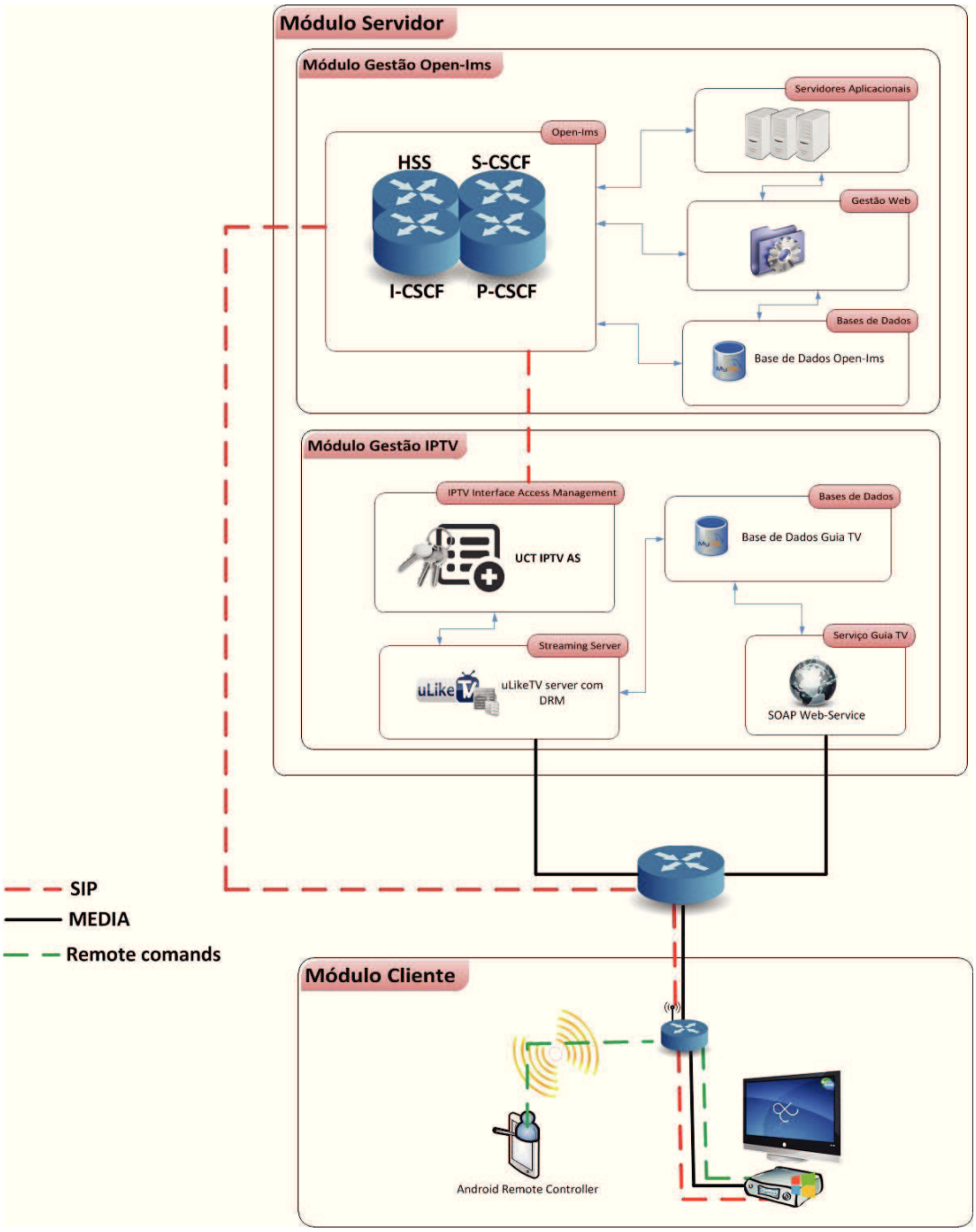


Figura 23 - Arquitetura Implementada

Analisando a Figura 23, verificamos que a arquitetura implementada integra as funcionalidades de autenticação SIP, captura, encriptação e envio de *streams*, bases

de dados de todo o sistema e uma aplicação *desktop* em *Windows* a simular uma *set-top box*.

3.2.1 Módulo Gestão Open-Ims

Na Figura 24, apresenta-se a arquitetura do módulo Gestão Open-Ims.

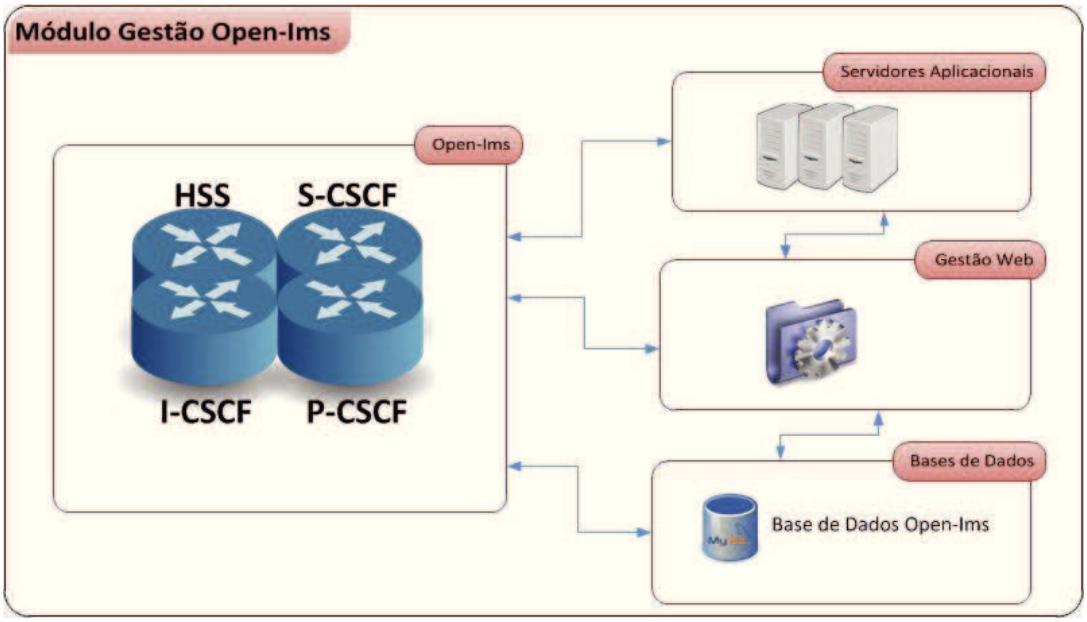


Figura 24 - Arquitetura Módulo Gestão Open-Ims (Arquitetura Implementada)

O Módulo Gestão Open-Ims pertencente à arquitetura implementada, incorpora as funcionalidades do serviço Open-Ims descrito no capítulo 2, os servidores aplicacionais cuja função é disponibilizar um conjunto de serviços pré-definidos. Para a definição deste nódulo foram utilizadas várias linguagens de programação: perl, bash, java e PHP.

3.2.1.1. Base de Dados Open-Ims

Na Figura 25 é apresentada a estrutura de tabelas da base de dados do Open-Ims que o serviço utiliza para manter registo da informação necessária ao bom funcionamento do mesmo, como por exemplo a informação dos servidores aplicacionais agregados ao sistema. De salientar que esta base de dados é transversal às duas arquiteturas propostas.

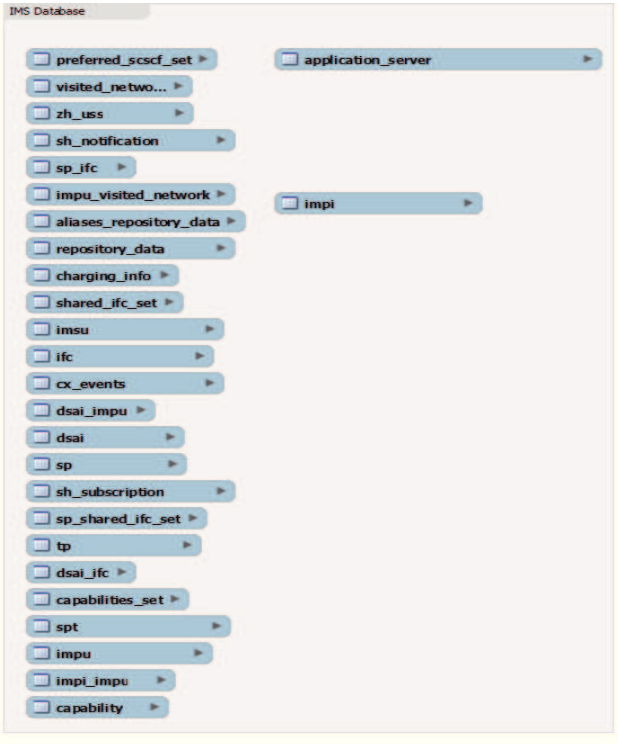


Figura 25 - Esquema Base Dados Open-Ims

3.2.2 Módulo Gestão IPTV

Na Figura 26 é apresentada a arquitetura do módulo Gestão IPTV.

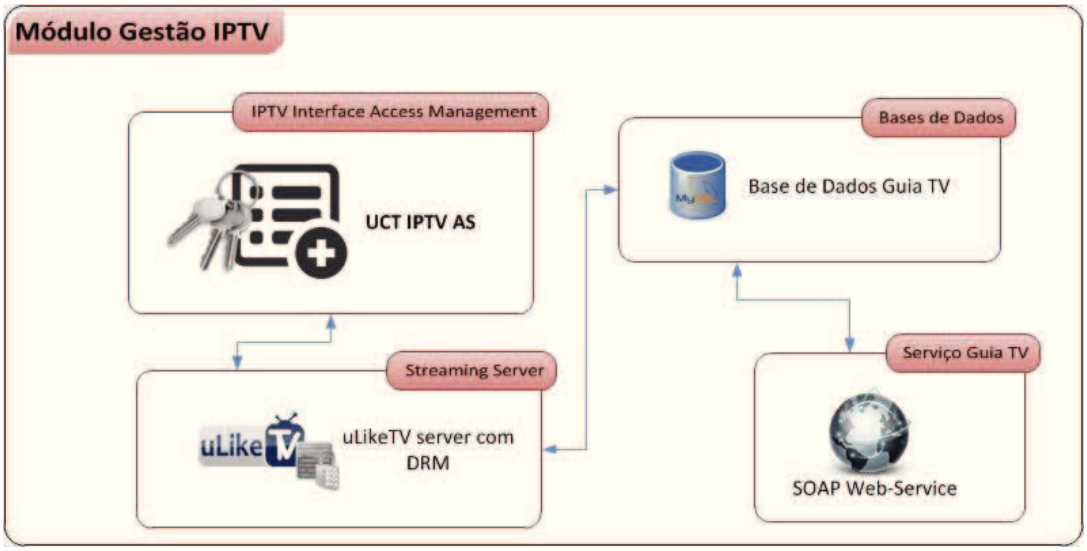


Figura 26 - Arquitetura Módulo Gestão IPTV (Arquitetura Implementada)

O Módulo Gestão IPTV é composto pelo *streaming server,* composto pelo servidor aplicacional uLikeTV e pelo DRM integrado no mesmo, pela IPTV *Interface Access Management (*UCT IPTV AS*)*, pela base de dados responsável pelo armazenamento do dados do Guia TV e por um *Simple Object Access Protocol* (SOAP) *Web-Service* que disponibiliza a Guia TV.

Em primeira instância o servidor uLikeTV efetua a captura dos *streams*, os quais são encriptados com recurso às DRM's, o servidor disponibiliza os canais em transmissão e chave de acesso ao IPTV *Interface Management*. Por último, é inserida na BD a informação relativa ao Guia TV que são disponibilizados através de *Web-Services*.

3.2.2.1. Streaming Server

O *streamimg server*, como já foi referido, este captura os *streams* (TDT) e encripta-os recorrendo às DRM's. Desta forma só é possível aceder aos mesmos com uma chave específica de desencriptação. Este tem ainda a função de enviar os dados relativos ao Guia TV para a base de dados.

Figura 27 - Interface Web do uLikeTV *Streaming Server* (Arquitetura Implementada)

A Figura 27 apresenta a interface Web do servidor uLikeTV, em que são apresentadas algumas opções dadas por este[9].

3.2.2.2. IPTV *Interface Access Management*

O IPTV *Interface Access Management* tem com objetivo enviar os canais disponíveis, endereços e chave de desencriptação para os clientes autenticados no *Open-Ims*, efetuando chamada SIP a este servidor aplicacional, o qual, após troca de mensagens envia os dados referentes aos canais ao cliente. Internamente é utilizado a aplicação ou servidor aplicacional UCT IPTV AS.

3.2.2.3. Serviço Guia TV

Este módulo é um SOAP *Web-Service* desenvolvido em Java que envia para os clientes o guia de programação de todos os canais existentes em *streaming*. Este *Web-Service* obtém os dados de uma base de dados criada pelo uLikeTV *streaming server*, onde é armazenado e atualizado pertinentemente o guia de programação.

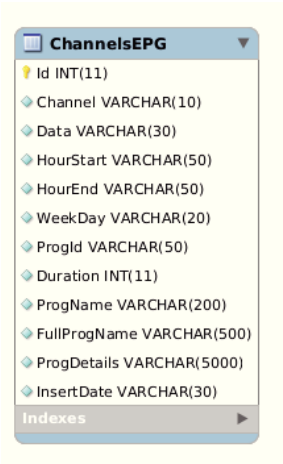


Figura 28 - Tabela Base de Dados do Guia TV (Arquitetura Implementada)

A Figura 28 representa unicamente a tabela da base de dados que é relevante para esta arquitetura, pelo que a base de dados em si é constituída por outras tabelas não relevantes.

3.2.3 Módulo Cliente

O módulo cliente é constituído por um computador com o sistema operativo *Windows* com o propósito de simular uma *set-top box,* como podemos constatar na Figura 29.

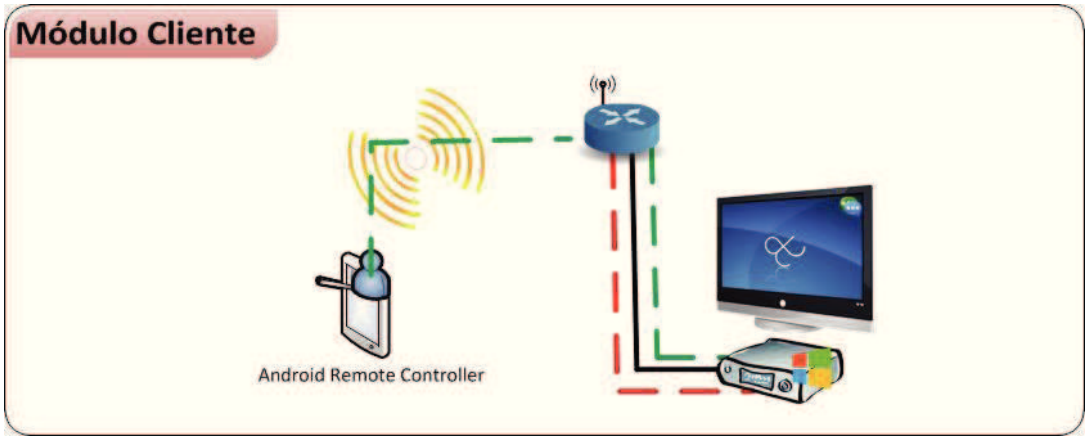


Figura 29 - Módulo Cliente (Arquitetura Implementada)

A diferença face á arquitetura geral que foi proposta, deve-se ao facto de ter sido necessário criar uma aplicação para um *Andoid tablet* de forma a disponibilizar o controlo total sobre a aplicação cliente desenvolvida, devido ao facto da aplicação cliente simular uma *set-top box*. Como tal, numa *set-top box* não é possível ligar um teclado com sucesso, e mesmo sendo possível, a quantidade e opções e *inputs* do

teclado associados tornariam o controlo penoso. A de rede local do cliente foi apresentada para dar foco à comunicação desta com a aplicação de controlo e o Módulo Servidor.

O suporte a vários sistemas operativos deve-se ao facto de não se ter optado uma linguagem de desenvolvimento transversal às diversas plataformas existentes. Este fato originou a reestruturação do módulo cliente, o qual mantém as funcionalidades de autenticação SIP, receção de *streams* e acesso a serviços à semelhança do módulo cliente da arquitetura geral.

3.3 Síntese

Neste capítulo foi dada a conhecer a arquitetura que considerada ideal como resposta para o problema apresentado posteriormente, foi definida tendo em conta as tecnologias estudadas no capítulo 2 com todos os prós e contras associados. As vantagens desta solução prendem-se com o fato de permitir a integração de serviços *out of the box* numa plataforma IPTV permitindo que esta se diferencie, tenha uma maior aceitação um uso superior e alternativo. Diminuindo o isolamento e exclusão social a que o público-alvo está sujeita.

Após a apresentação da arquitetura geral, foi definida a arquitetura a implementar. As características fundamentais dos módulos a implementar e que serão descritas no capítulo seguinte.

Capítulo 4 - Implementação

Neste capítulo são descritas as diversas etapas de desenvolvimento do protótipo, descrevendo em pormenor a implementação do Módulo Gestão Open-Ims, Módulo IPTV e Módulo Cliente.

No Módulo Gestão Open-Ims e Módulo IPTV são abordadas as dependências e os automatismos criados para a sua configuração e instalação.

No Módulo Cliente são abordadas soluções intermédias desenvolvidas aquando da conceção desta dissertação, as quais representam um estudo inicial, e crescente perceção, das verdadeiras necessidades intrínsecas à aplicação cliente. Após a apresentação destas é descrita a solução final da aplicação cliente.

4.1 Módulo Servidor

No intuito da dissertação, foi configurado um servidor físico que permitisse a implementação e lançamento deste módulo, com as seguintes características:

Caraterísticas	
Sistema Operativo	Ubuntu Server 12.04 LTS
Processador	2 Cores
Memória Ram	3 GB
Disco Rígido	10 GB
Placa de Rede	1 Gigabit

Tabela 1 – Tabela de Características do Ubuntu Server

4.1.1 Módulo Gestão *Open-Ims*

Este módulo trata da gestão e armazenamento dos dados dos utilizadores, e faz o controlo da autenticação que é feita pelos mesmos através da plataforma. De referir que se encontra dividido em quatro partes, Open-Ims Core, Base de Dados, módulo de gestão e Servidores aplicacionais. O Open-Ims Core engloba os serviços que gerem a autenticação SIP e controlam o acesso aos servidores aplicacionais. A plataforma de gestão fornece um controlo básico sobre o Open-Ims *Core.* Por outro lado o módulo Base de Dados, engloba as base de dados associadas tanto ao *Open-Ims* como à plataforma.

4.1.1.1. Open-Ims

Antes de iniciarmos do Open-Ims, temos que ter em consideração as suas dependências em termos de *software* previamente instalado para possibilitar a compilação do mesmo que são:

- Ant;
- Apache;
- Bind9;
- Bison;
- Curl;
- Flex;
- *GNU Compiler Collection* (GCC);
- Ipsec-tools;
- *Java Development Kit* (JDK);
- Libmysql-dev;
- Libxml2-dev;
- Make;
- MySql Server;
- PHP5.

4.1.1.2. Servidores Aplicacionais

Esta componente define os servidores aplicacionais necessários para o funcionamento correto do Open-ims. Os servidores são:

- Tomcat;
- MySql Server;
- Apache2.

4.1.1.3. Gestão Web

O módulo de gestão Web que disponibiliza várias opções de controlo sobre o *Open-Ims* tem as seguintes dependências:

- PHP5;
- *Java Development Kit* (JDK);

- MySql Server;
- Tomcat;
- Apache2.

4.1.2 Módulo Gestão IPTV

O módulo de gestão IPTV é responsável pela captura e envio de *streams* de vídeo, efetua a codificação/encriptação dos *streams*, gestão de acessos, envio da chave de desencriptação e disponibilização do Guia TV. Este módulo encontra-se dividido em quatro partes, O IPTV *Interface Access Management*, a Base de Dados, O serviço Guia TV e o *Streaming Server*.

4.1.2.1. Streaming Server

O uLikeTV *streaming Server* é o servidor aplicacional que é responsável pela captura dos canais TDT, processamento do guia de programação e envio dos *streams* via *multicast* para os equipamentos terminais mediante *Join* no grupo *multicast*. Este servidor foi alterado de forma a encriptar os *streams* enviados com uma chave. Desta forma, apenas os utilizadores autorizados é que têm acesso aos canais e os podem visualizar.

Este possui as seguintes dependências:

- Apache2;
- VLC;
- GCC;
- Make;
- DvbSnoop;
- PHP5.

4.1.2.2. IPTV Interface Access Management

O UCT IPTV AS, é um servidor aplicacional SIP/IPTV que gere os pedidos dos utilizadores para facultar os canais disponíveis (endereços *multicast*). Disponibiliza ainda em simultâneo a chave de desencriptação dos respetivos canais.

4.1.2.3. Serviço Guia TV

O Serviço Guia TV fornece os dados, a pedido, da programação televisiva dos canais em emissão. Este serviço possui as seguintes dependências:

- *Java Development Kit* (JDK);
- MySql Server;
- Tomcat.

4.1.3 Instalação e configuração

Tendo em conta as dependências e as modificações necessárias ao serviço para que o módulo fique funcional, foi desenvolvido um script de instalação que de forma automatizada instala todas as dependências que não se encontrem no sistema operativo, e que procede à alteração de todos os ficheiros de configuração.

Este *script* é fundamental pois procede a alterações essenciais aos serviços e servidores aplicações aquando da sua execução. Para além disso, efetua uma série de testes elementares aos serviços após a sua configuração. Essencialmente, o *script* configura todo o ambiente automaticamente, tornado a primeira instalação deste módulo mais simples de concretizar.

4.1.3.1. Módulo Gestão Open-Ims

Nesta secção são identificadas as alterações e configurações feitas ao módulo de gestão Open-Ims.

- add-imscore-user_newdb.sh

Este *script* tem como função gerar de forma automática um script sql que após execução permite adicionar um novo utilizador à base de dados. No entanto, devido ao fato de o Open-Ims ser uma plataforma de desenvolvimento em constante mudança, este script não se encontrava funcional e foi corrigido.

- configurator.sh

O configurator.sh é um *script* que permite a modificação do domínio e IP associados ao serviço uma única vez. Contudo, este funcionamento não é o mais eficiente em termos de gestão, nesse sentido foi também alterado para receber o IP e

domínio por parâmetro, alterando de uma forma dinâmica os dados anteriores nos respetivos ficheiros de configuração do sistema.

- hss_db_mod.sql

Este *script* trata da criação da base de dados necessária ao funcionamento do FHoSS, no entanto fá-lo utilizando o motor MyISAM que não permite o uso de transações. Fato que limita a gestão da base de dados e pode levar a corrupção de dados aquando da inserção de dados em várias tabelas. Por estas razões o script foi modificado para que a base de dados seja criada com um motor InnoDB o qual já permite o uso de transações.

- userdata_mod.sql

Este *script* tem como função a inserção de dados relativos ao AS interno que fornece as funcionalidades SIP, assim como de dois utilizadores padrão para testes. Contudo este funcionamento não correspondia ao pretendido, como tal, foi alterado para simplesmente adicionar os parâmetros relativos ao AS.

- pcscf.cfg

Ficheiro de configuração do serviço p-cscf que originalmente vem configurado de forma a bloquear os pedidos de redes externas. Como tal, foi alterado para permitir que o servidor aceite pedidos externos, permitindo assim a comunicação proveniente da internet.

- startup.sh

Este *script* inicia o serviço FHoSS, mas o caminho pré-definido para a instalação do JDK necessita de adaptação consoante o sistema operativo escolhido. Como tal é alterado pelo script de instalação de modo a que o caminho seja o correto mediante o sistema operativo utilizado.

- named.conf.local

Este é um dos ficheiros de configuração do serviço *Domain Name System* (DNS) que permite a adição de zonas de resolução de domínios. Como tal, é alterado durante o processo de instalação, de modo a conter o domínio sobre o qual o *Open-Ims* irá funcionar.

- dhclient.conf

Este é um dos ficheiros de configuração do *Dynamic Host Configuration Protocol* (DHCP) server que teve de ser alterado para que os pedidos de resolução do domínio do serviço, sejam resolvidos para o IP associado ao mesmo.

- scscf.cfg

Neste ficheiro de configuração do serviço s-cscf, foi alterado o algoritmo usado na encriptação dos dados, para corresponder à solução proposta.

Após todas as ações supracitadas, obtemos o módulo de gestão *Open-Ims* totalmente funcional.

4.1.3.2. Módulo Gestão IPTV

Nesta secção são identificadas as alterações e configurações feitas ao módulo de gestão IPTV.

- capture_all_channels_vlm.conf

Este ficheiro de configuração é onde existem todos os dados referentes à captura dos *streams* e envio dos mesmos. Procedeu-se à alteração/criação deste ficheiro para incluir a encriptação dos *streams*, para que apenas possam ser visualizados com a chave correta.

- startLiveTV.pl

Este é um dos principais *scripts* existentes no uLikeTV *Server*, em que é o responsável por iniciar o serviço e iniciar a captura e envio de *streams.* É neste é referenciado o ficheiro de configuração dos canais a fazer *streaming*. Este ficheiro, foi alterado de forma a efetuar a leitura do novo ficheiro de configuração criado.

- Iptvas.xml

Este ficheiro é parte integrante do IPTV *Interface access Management*, define os canais disponíveis para os clientes e é necessário de modo a conter a listagem de canais disponíveis e a chave de desencriptação dos mesmos.

4.2 Módulo Cliente

Para definir o módulo do cliente foram utilizadas várias API's de suporte. No entanto, numa fase inicial, foi necessário iniciar o desenvolvimento de uma API de SIP de raiz, já que até a esse momento não tinham sido encontradas nenhumas API's em .Net que fossem gratuitas ou que fornecessem suporte para o Open-Ims.

Neste módulo foram desenvolvidos dois protótipos da aplicação cliente, após a integração da SIP API, tendo os mesmos requerido a utilização/integração do WPF com a VlcDotNet API para *Windows Forms* e com a VlcDotNet API para WPF.

Neste sentido é apresentado nesta secção as soluções intermédias desenvolvidas, a solução final implementada para o módulo do cliente e a aplicação *Android* para controlo da aplicação cliente.

4.2.1 Soluções Intermédias

Durante a implementação do módulo cliente, foi necessário o desenvolvimento de soluções intermédias de modo a ser possível encontrar a melhor opção. Esta secção apresenta as soluções desenvolvidas até ter sido encontrada a solução final.

Foram também desenvolvidas e analisadas soluções intermédias referente à API de SIP e à integração da VlcDotNet API, nomeadamente a direcionada para *Windows Forms* e para WPF.

4.2.1.1. SPI API

A SIP API é uma biblioteca que implementa alguns dos métodos/mensagens mais comuns do SIP2.0, segundo as especificações da sua RFC. As mensagens incluem, a autenticação (toda a troca de mensagens, incluindo a resposta após o *challenge* para concluir a fase de autenticação), e o início do estabelecimento de uma chamada de voz/vídeo para outro cliente previamente autenticado.

Na Figura 30 é apresentado o diagrama de classes implementadas para a API desenvolvida.

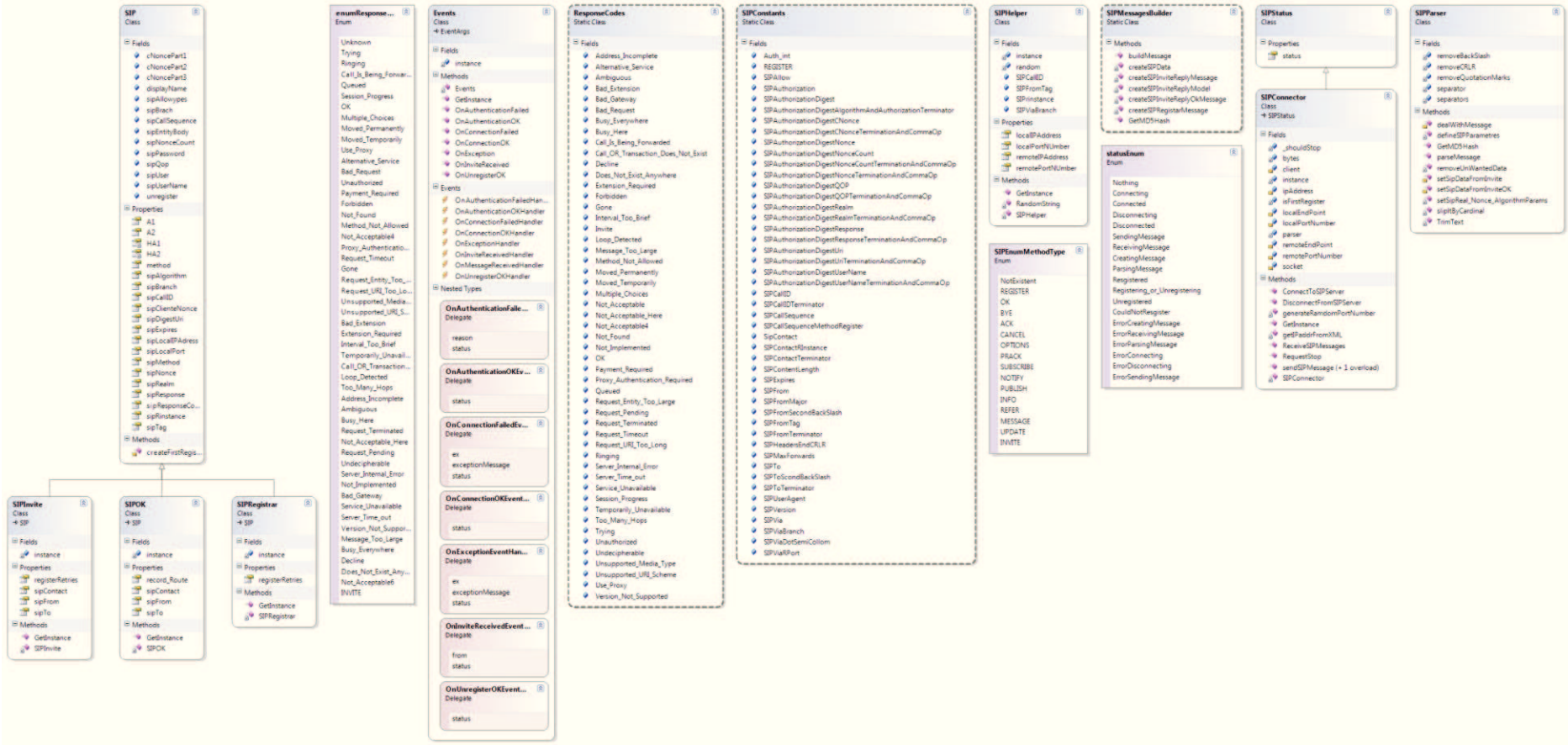


Figura 30 - SIP API desenvolvida (Solução Intermédia)

4.2.1.2. Aplicação Cliente híbrida (WPF e *Windows Forms*)

Esta aplicação híbrida foi desenvolvida inicialmente, em que se definiu o uso de WPF para interface gráfica e o uso do VLC *Plugin* para *Windows Forms*, visto que a sua manipulação era mais simplificada do que a VLC API para WPF.

No entanto, durante o desenvolvimento inicial, aquando da receção se *streams* e pré-visualizações de *streams*, deparámo-nos com o problema denominado por *AIRSPACE* que ocorre aquando do uso de componentes desenhados para *Windows Forms* em WPF e vice-versa que os componentes ficam desacoplados da interface ou janela em que supostamente estariam embutidos, dificultando o controlo sobre os mesmos e em certos casos a total ausência de controlo. Constatou-se ainda que este problema ocorria em todas a versões do .Net

Na Figura 31 pode-se verificar a interface desenvolvida, e o estado de desenvolvimento no momento.

Figura 31 - Interface da aplicação hibrida desenvolvida

A Figura 32 representa as classes desenvolvidas referentes à interface.

Figura 32 - Diagrama de classes da aplicação cliente híbrida

4.2.2 Solução Final

Nesta secção é descrita a implementação final da aplicação cliente, enumerando as API'S utilizadas, alterações feitas a estas e aspetos mais importantes.

É descrita ainda a implementação, opções, definição de menagens de controlo e interface do *Android Remote Controller*.

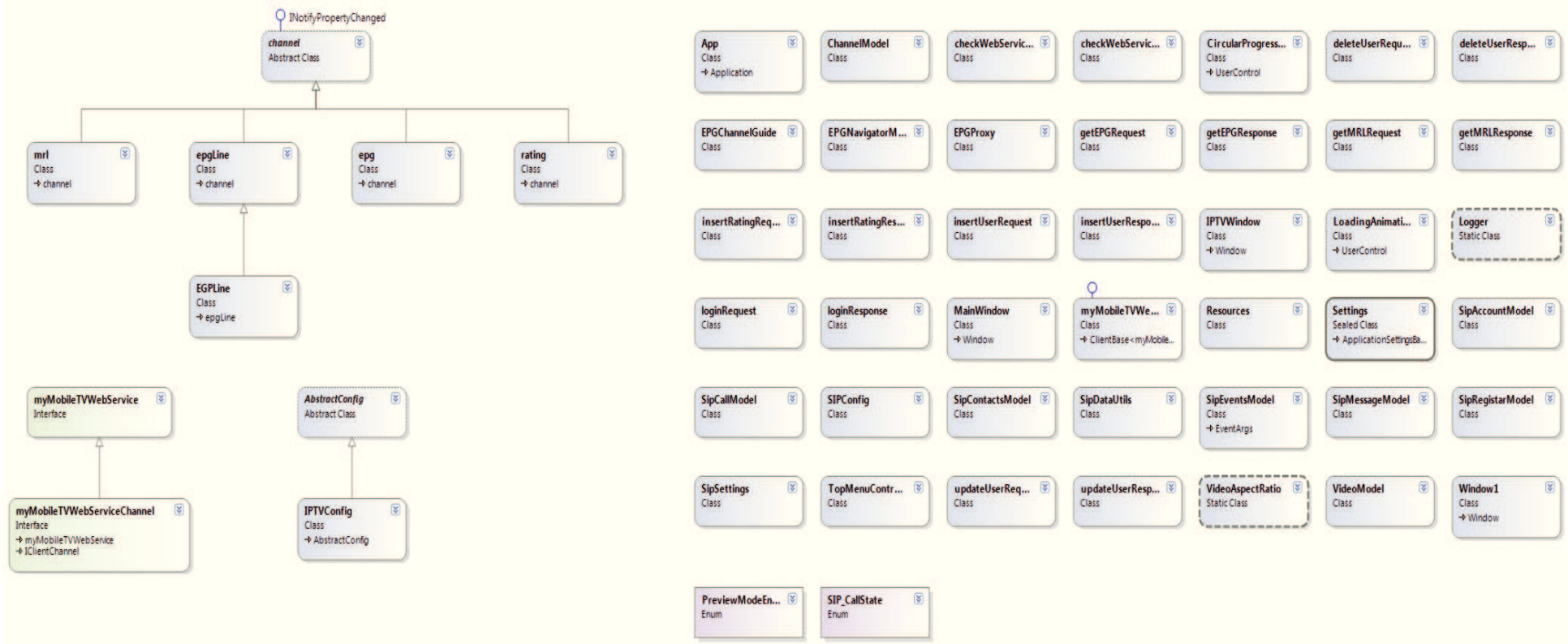


Figura 33 - Diagrama de classes da aplicação cliente final

Na Figura 33 é apresentado o diagrama de classes que representa todas a iterações realizadas pela aplicação cliente, a nível de *design* e operabilidade.

4.2.2.1. LumiSoft

A Lumisoft, API, como indicado anteriormente é uma biblioteca SIP gratuita que fornece uma base sólida de implementação do SIP2.0 que já integra uma aplicação cliente nativa para testes. No entanto, esta API, não suporta a autenticação no Open-Ims, pelo que se teve de localizar os pontos chaves que falhavam, de forma efetuar alterações cruciais para esta suportar a autenticação, em contrapartida, todos os outros processos e interpretação de mensagens SIP já se encontravam nativamente suportadas e sem problemas.

A Figura 34 demonstra e identifica algumas das alterações efetuadas à API em causa para esta passasse a suportar a autenticação definida pelo Open-Ims.

```
/// <summary>
/// Made by Cláudio
/// </summary>
private void setAuthQopOpenIMS()
{
    string[] authParams;
    if (this.m_Qop.Contains(","))
    {
        authParams = this.m_Qop.Split(',');
        foreach (string item in authParams)
        {
            if (item.ToLower() == "auth-int")
                this.m_Qop = item;
            else
                this.m_Qop = item;
        }
    }
}
```

Figura 34 - Alterações feitas à API LumiSoft

Em concreto, a LumiSoft não conseguia efetuar o *parsing* do *challenge* enviado pelo Open-Ims, em que no instante de receção da mensagem esta não era reconhecida como válida e consequentemente descartada. Nesta sequência, procedeu-se à correção e reimplementação de métodos definidos para o que *parsing* fosse bem-sucedido e a mensagem de *challenge* reconhecida pela API.

4.2.2.2. Prototipagem

De seguida é apresentado o protótipo realizado na pré-implementação, e que serviu como base para a implementação da aplicação cliente desenvolvida posteriormente.

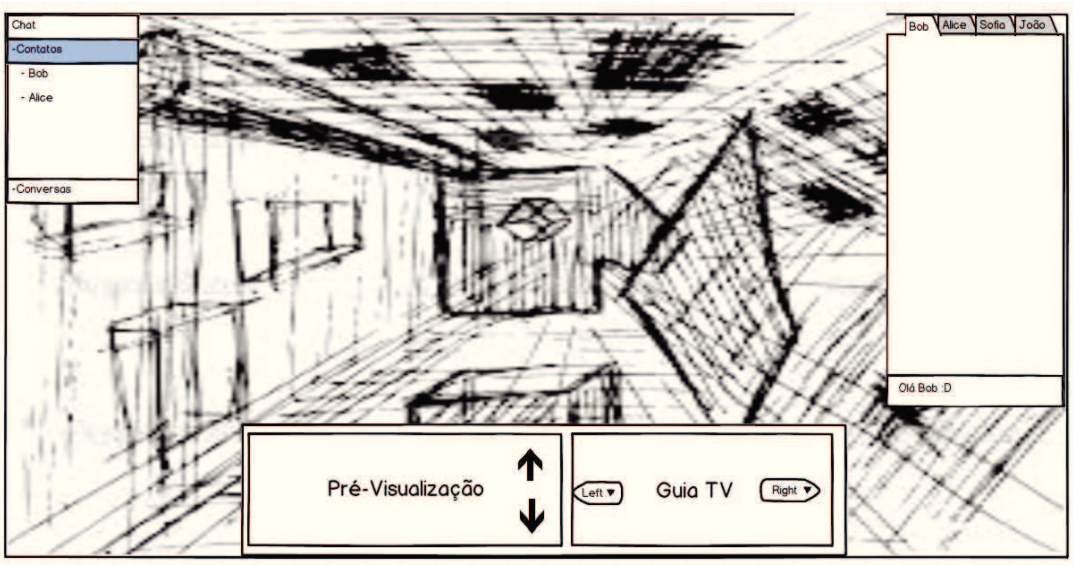


Figura 35 – Protótipo interface da aplicação cliente

Como se pode observar na Figura 35, optou-se por ter apenas uma "janela " na aplicação, em que todas as opções são representadas sobre esta.

4.2.2.3. Android Remote Controller

O *Android Remote Controller* é uma aplicação desenvolvida para um *Android Tablet* com o intuito de fornecer e facilitar o controlo total sobre a aplicação cliente.

Esta aplicação possibilita a navegação entre menus, escrita de mensagens e controlo sobre a visualização dos canais.

A comunicação entre o *Android Remote Controller* e a aplicação cliente é efetuada via de *sockets* UDP. Foram definidos comandos específicos a serem enviados desta aplicação para o simulador de *set-top box* que são interpretados de maneira a que cada comando corresponda a uma determinada ação no contexto atual.

Ecrã Android remote controller

Figura 36 - Ecrã do Android Remote Controller

Na Figura 36 temos a representação das opções facultadas por esta aplicação. As opções, associadas a um comando específico têm significado consoante o contexto da aplicação cliente que simula uma *set-top box*. As opções suportadas são:

- **Botão 1**

Intercalar entre o modo standby e o modo ativo;

- **Botões 2 e 3**

Controlo do volume, incremento ou decremento;

- **Botões 4 e 5**

Alterna entre canais, incrementando ou decrementando os canais;

- **Botões 6, 7, 8 e 9**

Estes botões têm funcionalidades diferentes dependendo do contexto. Se o contexto for o menu de serviços (Chat), permitem a navegação pelas opções, se o contexto for a apenas visualização de um canal, as setas esquerda e direita, permitem a previsualização de canais, iterando cada um e ainda neste contexto, de pré-visualização, a seta esquerda e direita possibilitam a consulta do Guia TV;

- **Botão 10**

Este botão tem várias funcionalidades, isto é, depende do contexto em que a aplicação se encontra. Se se estiver no menu de serviços, permite escolher a opção pretendida, se o contexto for na escrita de uma mensagem, a mensagem é enviada, e se for no contexto de previsualização de canais, permite a escolha do canal em pré-visualização;

- **Botão 11**

Mostra ou esconde o Menu de serviços (Chat), consoante o estado atual do menu;

- **Botão 12**

Fecha qualquer menu ou pré-visualização que esteja ativa;

- **Botão 13**

Este botão permite intercalar entre *play* (*resume*) e *pause*;

- **Botão 14**

Este botão permite intercalar entre *mute* e *unmute* (com e sem som);

- **Botão 15**

Este botão permite inicializar a escrita de uma mensagem. Esta escrita é efetuada no teclado do dispositivo *android* e cada carater escrito é enviado e interpretado pela aplicação destino de forma a construir a mensagem sem que seja necessário retirar a atenção dos conteúdos na aplicação cliente.

Escrita de uma Mensagem

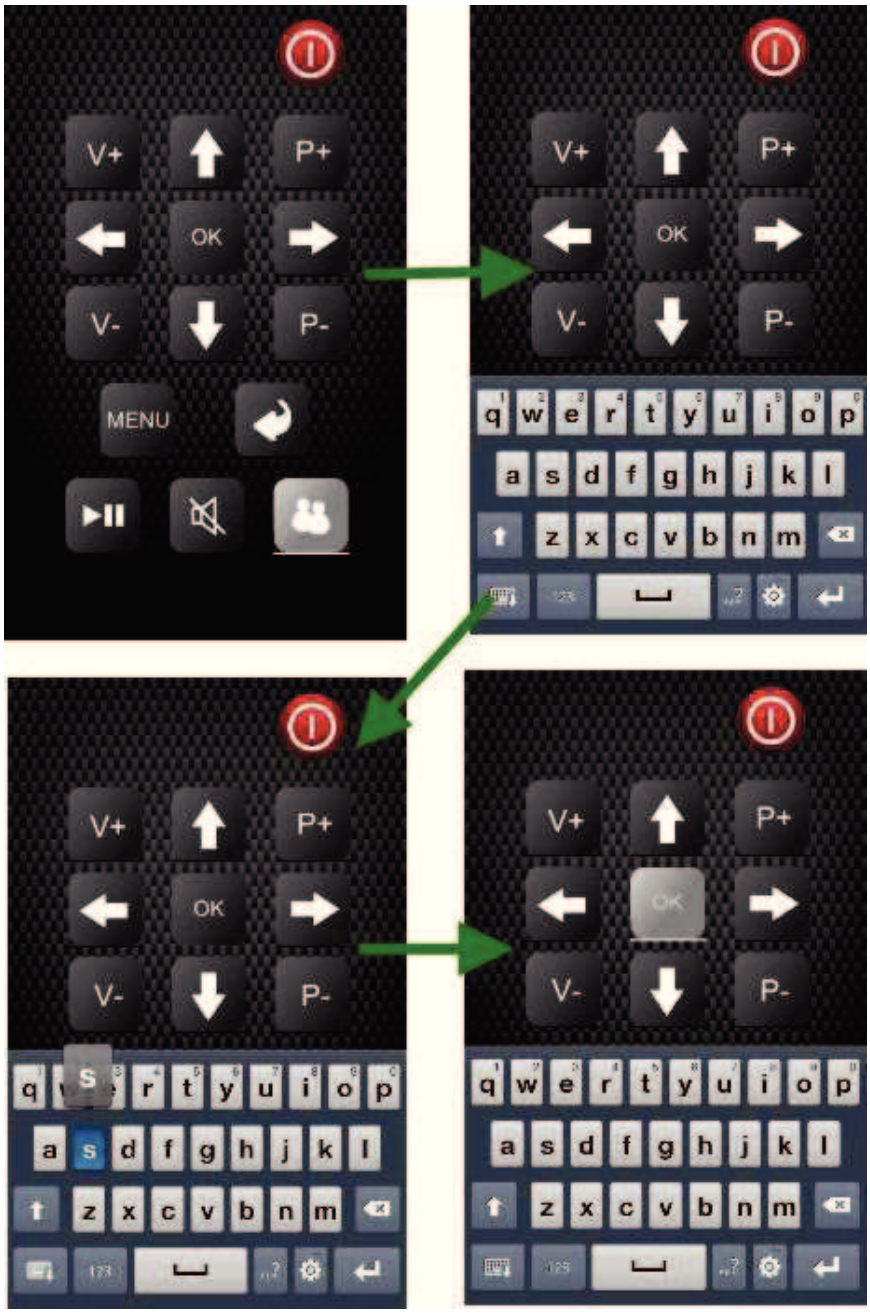


Figura 37 - Escrita de uma mensagem

Na Figura 37 é demonstrada a escrita de uma mensagem, em que é pressionado o botão de *chat* que abre o teclado virtual do dispositivo. Após a abertura do teclado virtual, pode-se começar a digitar carateres, cujos são enviado ao cliente durante a sua introdução. Concluída a escrita da mensagem é pressionado o botão "*OK*" e a mensagem é enviado de um cliente para outro.

Comandos definidos

De seguida, na Tabela 2 são apresentados os comandos definidos para a comunicação entre a aplicação Android (emissora) e a aplicação cliente (recetora).

Comando	Significado
v_audioup	Aumentar o volume
v_audiodown	Diminuir o volume
channel_up	Incrementar um canal na visualização
channel_down	Decrementar um canal na visualização
m_up	Navegação sentido ascendente
m_down	Navegação sentido descendente
m_left	Navegação com deslocação lateral para a esquerda
m_right	Navegação com deslocação lateral para a direita
m_enter	Seleção de uma opção ou o envio de uma mensagem
m_menu	Mostra ou esconde o menu de serviços
l_back	Esconde todos os menus abertos
v_tooglePlay	Alterna entre *pause* e *play/resume*
message:	Este comando é diferenciado visto que indica que é uma mensagem a ser escrita. Este campo é composto pelo indentificador ***message:*** mais o carater introduzido na aplicação e.g.: ***message:O*** ***message:l*** ***message:á*** A mensagem na aplicação cliente fica: ***Olá***

Tabela 2 - Comandos definidos e seu significado

4.2.2.4. Interface

Como o público-alvo não se restringe apenas a uma faixa etária em específico, pelo que aplicação tem de ser intuitiva, fácil de usar e agradável.

Com o supracitado em mente, foram definidas zonas distintas para a localização de cada componente da aplicação, no entanto, o maior desafio é colocar e definir a localização mais adequada para os serviços de comunicação fornecidos,

nomeadamente o de Chat. Esta dificuldade prende-se com o facto de que um utilizador pode estar em *chat* com mais do que um só indivíduo.

Um aspeto importante (no serviço de *Chat*) que se tem de ter em consideração é a distinção visual de quem foi o autor da mensagem para cada um dos intervenientes. Nesse seguimento, foram usadas formatações de texto diferentes com o intuito de serem distinguidos sem esforço os emissores dos recetores. Outro aspeto tido em conta foi o estado de entrega das mensagens, isto é, se uma mensagem não for entregue o emissor recebe uma notificação com o texto a amarelo para sobressair e alertar para o facto ocorrido.

Os *previews* e consulta do Guia TV foram agrupados numa área específica, em que funcionam em conjunto utilizando ”instruções naturais“ (frente, atrás, cima e baixo), para que sejam familiares, de fácil uso e compreensão por parte dos utilizadores.

Para além do descrito anteriormente, também possui as funções mais elementares de uma televisão, controlo de som, ligar/desligar e mudar de canal.

De Seguida apresentamos as várias componentes da interface desenvolvida e enumeramos as suas funcionalidades com maior detalhe.

Janela de *Loading*

Figura 38 - Janela de Loading

Na Figura 38 é apresentado o ecrã de *Loading* da aplicação, é neste momento que a aplicação carrega todos os dados do utilizador e dos serviços a ligar-se, especificados previamente e guardados no sistema em ficheiros xml.

Após o carregamento de todos os dados necessários para a ligação ao servidor e autenticação, é efetuada uma autenticação SIP no sentido da aplicação para o servidor. Nesta fase, após autenticação bem-sucedida, a aplicação efetua uma chamada para o IPTV *Interface Access Management*, o qual responde, após uma troca de mensagens, com uma mensagem SIP OK que possui um campo *Uniform Resource Locator (url)* em que devolve os endereços *multicast* dos canais e a chave de desencriptação dos mesmos.

Posteriormente à receção dos canais disponíveis e chave respetiva de desencriptação, é inicializada a visualização do último canal escolhido pelo utilizador, assim como o seu respetivo som.

Visualização e pré-visualização de um canal

Figura 39 - Visualização e pré-visualização de um canal

Na Figura 39 é apresentada a visualização e pré-visualização de um canal. Nesta interface gráfica, é possível navegar ou fazer *zapping* entre canais sem deixar de ver o canal em ecrã inteiro. Esta funcionalidade é útil devido ao fato de se poder escolher um canal a visualizar após a sua previsualização.

É ainda possível navegar pelo Guia TV de cada canal, esteja em visualização ou em pré-visualização. O Guia TV apresenta informação sobre, o canal correspondente, o programa, a descrição do programa e a data/hora de início e fim do programa. O mesmo ser aplica quando se consulta programas que ainda não começaram, isto é programas que ainda irão dar, numa determinada hora e dia.

Navegação no Guia TV

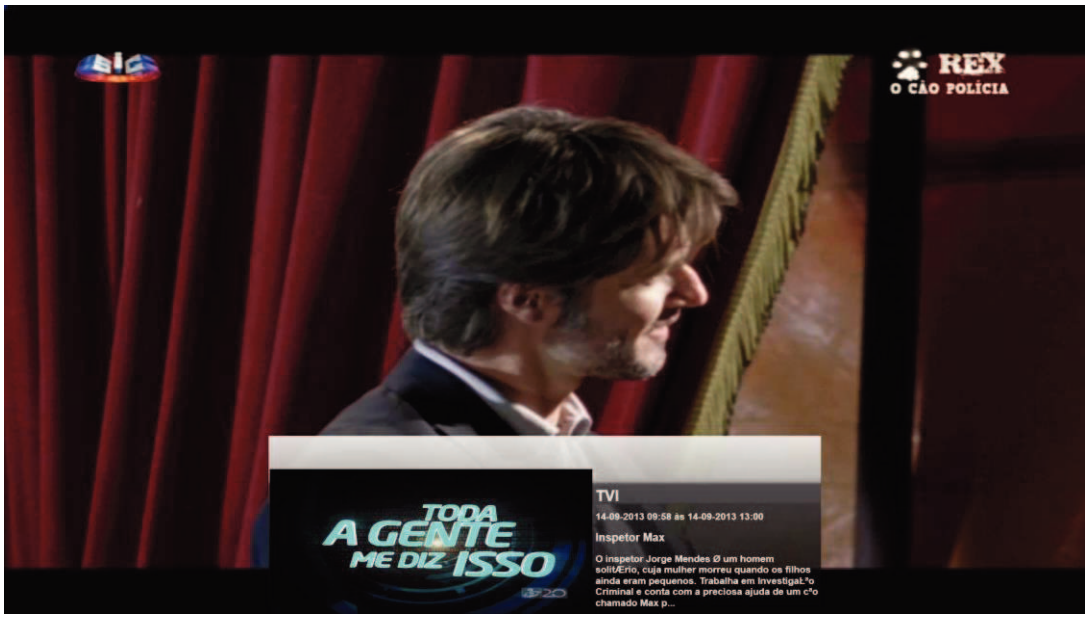


Figura 40 - Navegação no Guia TV

Na Figura 40 é observável na parte inferior da imagem o Guia TV, cujo se está a consultar em simultâneo à visualização do canal "SIC" e cujo Guia corresponde ao canal "TVI".

O Guia TV permite a consulta de informação relativa a programas que decorrem no momento e programas que irão ser emitidos e que já se encontram definidos na programação.

Seleção de um canal em pré-visualização

Figura 41 - Seleção de um canal em pré-visualização

A Figura 41 apresenta a seleção de um canal em pré-visualização, em que após o *zapping,* é escolhido o canal que se pretende. O canal selecionado é colocado em modo ecrã inteiro e a pré-visualização fechada.

Também é possível alternar entre canais sem pré-visualização, isto é incrementa-se ou decrementa-se o número do canal. Por exemplo, ao estar no canal 1, se se decrementar o canal, este vai para o canal 5 (último canal na lista), se for uma incrementação, do canal 1 passa para o canal 2, e assim respetivamente.

Aumento e diminuição do volume

Figura 42 - Aumento e diminuição do volume

Na Figura 42 é apresentado o controlo de volume que indica o estado do mesmo à medida que é alterado. Assume valores de 0 a 100.

O valor atual do volume é guardado entre execuções da aplicação. Assim, em cada execução o volume já se encontra definido de acordo com o último incremento/decremento do volume.

Mute/Unmute

Figura 43 - Som em Mute/Unmute

A Figura 43 apresenta a seleção de colocar o som em *mute* e *unmute*. Para alternar entre os dois estados basta escolher a opção de mute que se estive em mute, passa para *unmute* (assume o valor do som anterior) e se estiver com som (*unmuted*), fica sem som.

Menu Serviços

Figura 44 - Menu de Serviços (Chat)

Na Figura 44 é apresentado o menu de serviços que contempla o neste momento apenas contempla o *Chat*. Neste menu, é possível consultar contatos e selecionar um para inicializar uma conversa. O submenu Conversas nesta instância não foi desenvolvido. No entanto, o seu objetivo é consultar conversas armazenadas automaticamente

Interface de escrita de mensagens - Escrita

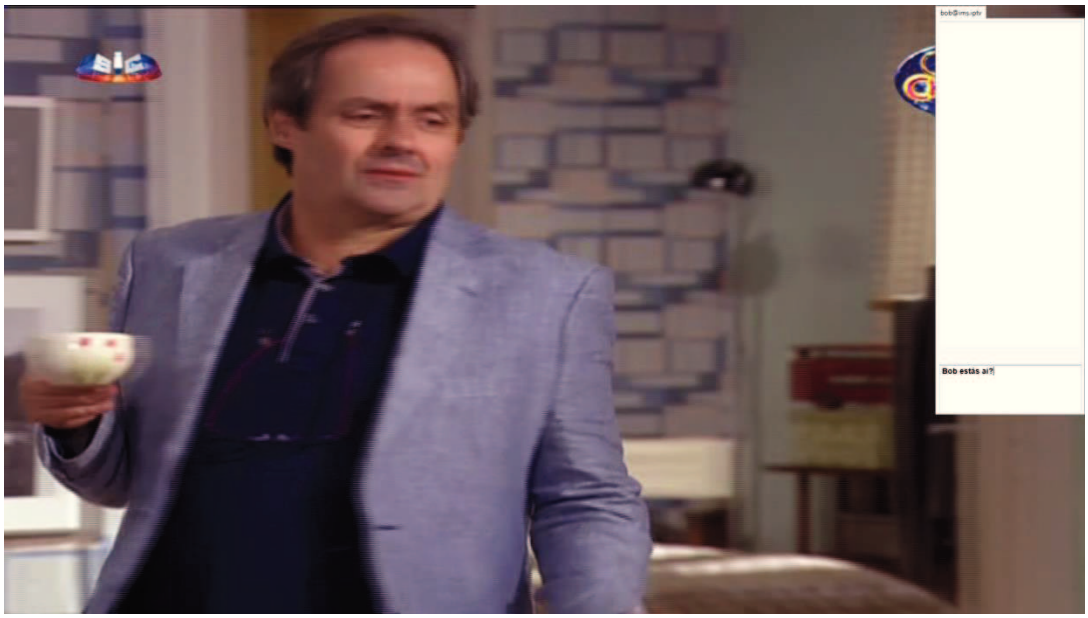


Figura 45 - Escrita de mensagem para um contato

Na Figura 45 é possível observar o componente que apresenta a escrita de mensagens para um utilizador. Esta Figura é o seguimento da escolha de um dos contatos referida e demonstrada na Figura 44.

Envio de uma mensagem

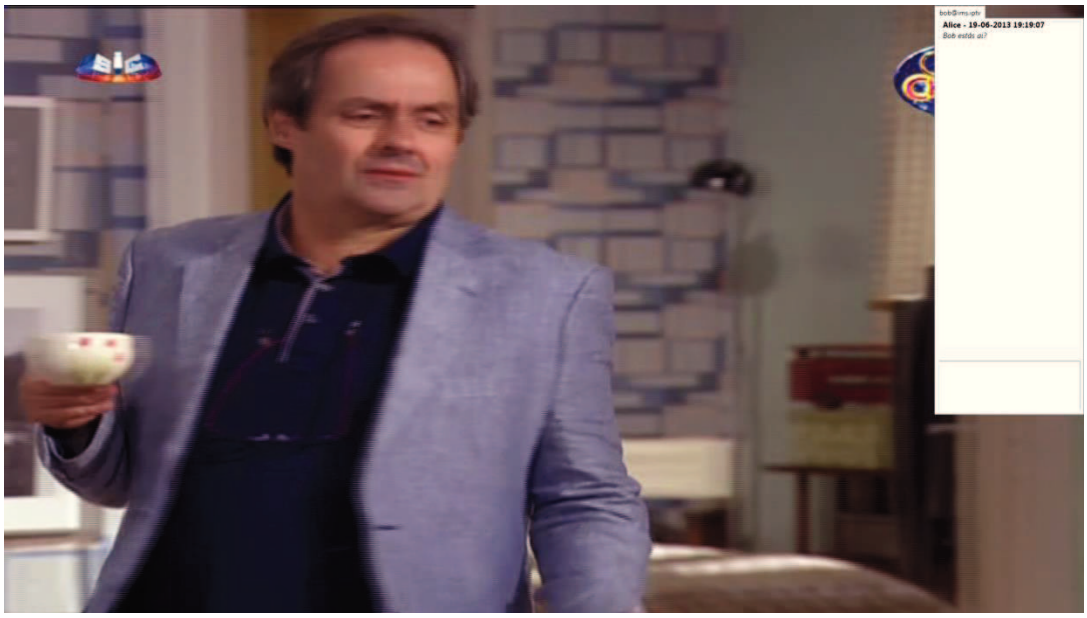


Figura 46 - Envio de uma mensagem para um contato

A Figura 46 apresenta a finalização da escrita e envio de uma mensagem para um contato. Após o envio, são apresentados detalhes do envio, como o emissor, a data e hora de envio e a mensagem propriamente dita.

Receção de uma mensagem

Figura 47 - Receção de uma mensagem de um contato

Na Figura 47 é visível o envio da mensagem apresentada na Figura 46 para um cliente, observado no canto inferior direito. Este cliente SIP, é o X-Lite 4.

No cliente X-Lite 4, o Bob, está a escrever uma mensagem para a Alice em resposta à sua mensagem. À medida que o Bob escreve a mensagem para a Alice, é possível verificar que no cliente desenvolvido, é dado o *feedback* que o Bob está a escrever uma mensagem.

Após o envio da mensagem, do Bob para a Alice, é visível a receção da mensagem, com os detalhes descritos anteriormente mas com uma tonalidade e tipo de letra diferente para se distinguir mais facilmente os emissores.

Falha no envio de uma mensagem

Figura 48 - Falha no envio de uma mensagem

A Figura 48 pode-se observar o comportamento da aplicação ao ser enviada uma mensagem para um destinatário e esta falha.

De notar que quando ocorre a falha da mensagem, o emissor é avisado com uma mensagem "*Mensagem não entregue!!*" e com uma coloração diferente da normal, para chamar a atenção ao utilizador

Modo standby

Figura 49 - Aplicação em Standby

Na Figura 49 é apresentada a aplicação em modo *standby* ou em modo *offline*.

Para colocar a aplicação em *standby*/*offline* basta "desligar" a visualização. Neste modo não é possível utilizar os serviços associados à aplicação. A troca entre modos é feita de forma semelhante à do *mute*/*unmute*, isto é, a aplicação sabe em que estado está e responde de acordo.

4.3 Síntese

Este capítulo abordou todas as fases de implementação necessárias ao desenvolvimento de cada um dos módulos, os quais em conjunto fornecem as peças necessárias para que a nossa solução funcione como pretendido, como por exemplo a autenticação SIP, encriptação e desencriptação de *streams* de multimédia e *Chat*.

Por outro lado, a solução projetada visa ser uma infraestrutura de baixo custo, tanto em termos de manutenção como em termos de custos para os utilizadores, o que a torna numa ferramenta sem necessidade de grandes recursos ou investimentos financeiros, fato importante na sociedade atual.

Capítulo 5 - Testes

Neste capítulo apresentamos os testes efetuados à solução proposta e os respetivos resultados. Os testes focaram-se em três âmbitos distintos, funcionamento dos módulos do servidor, módulo do cliente e *Quality of Experience* (QoE).

Os testes efetuados aos módulos do servidor e do cliente, serviram para a aferição do correto funcionamento de ambos como um conjunto, por outro lado, os testes de QoE foram realizados com o intuito de aferir o grau de usabilidade da interface desenvolvida para a plataforma

5.1 Testes Efetuados ao Módulo Servidor

Como referido anteriormente, é o servidor que processa e fornece todos os dados que a aplicação cliente necessita, pelo que se a autenticação não for efetuada corretamente ou a listagem de canais e chave respetiva não forem disponibilizados, não será possível proceder à utilização de nenhuma das funcionalidades da aplicação.

Deste modo, é importante testar a autenticação de clientes, o envio da lista de canais e chave de desencriptação para que se possa aferir o correto funcionamento do servidor e da aplicação cliente.

5.1.1 Autenticação SIP

Objetivos: Com este teste pretende-se constatar o funcionamento da autenticação e seus passos para ser bem-sucedida.

Descrição: Para efetuar este teste é necessário ter o servidor operacional e executar a aplicação cliente.

A aplicação cliente, ao iniciar, irá ler os ficheiros de configuração e com os dados recolhidos irá proceder à autenticação.

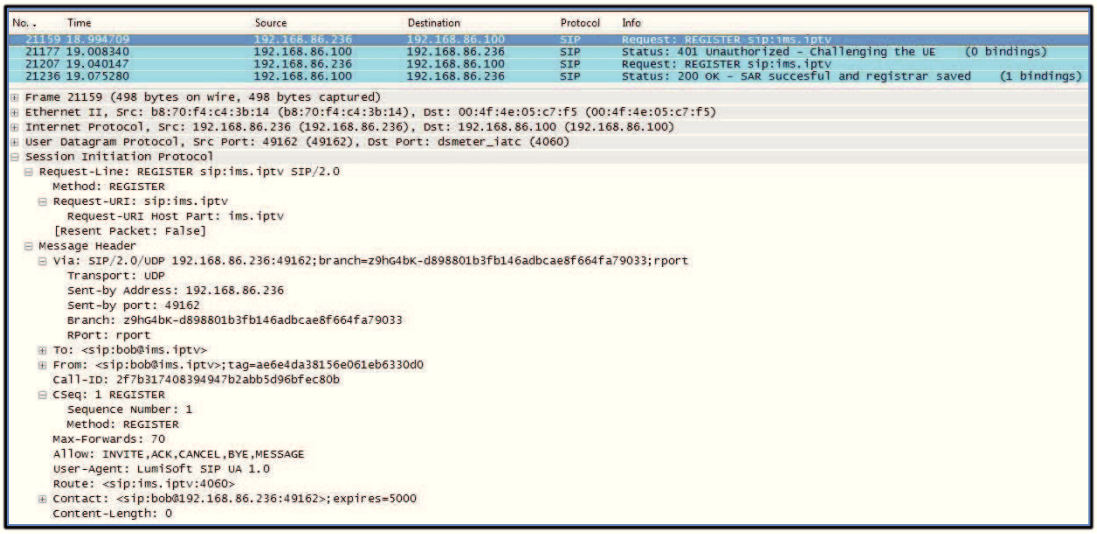


Figura 50 - Autenticação da aplicação (Wireshark)

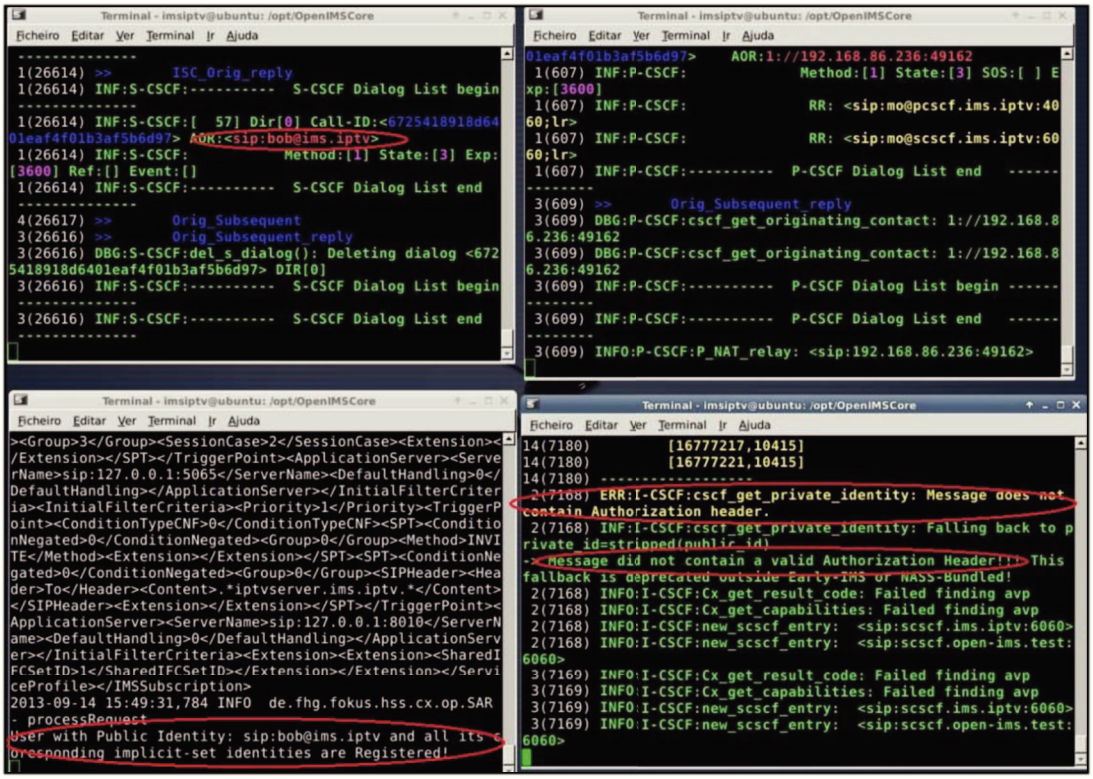


Figura 51 - Processamento de autenticação (lado do servidor)

Conclusão: Na Figura 50 e Figura 51, verifica-se que a aplicação autenticou-se com sucesso no servidor.

A Figura 50 demostra os passos necessários para proceder à autenticação, o envio da mensagem REGISTER do cliente para o servidor, o envio do *401 Unauthorized – Challenging the UE* para o cliente, a segunda mensagem REGISTER com

parâmetros específicos recebidos e criados, para o servidor e finalmente a receção do OK no cliente.

Enquanto a Figura 51 demostra os *outputs* no lado do servidor em que existe um aviso que não existe cabeçalho de autenticação, é nesse momento que o servidor envia uma mensagem SIP com o *challenge*. Nesse momento a cabeçalho de autenticação é construído com dados enviados pelo servidor e recriados de acordo com especificações na RFC. Após esse processo, a autenticação é bem-sucedida, observável no canto inferior esquerdo desta mesma figura.

5.1.2 Pedidos de canais e chave de desencriptação

Objetivos: Este teste pretende demostrar o pedido dos canais disponíveis e a chave de desencriptação por parte do cliente. Que após a receção de ambos, visualizará os canais disponíveis

Descrição: Para efetuar este teste é necessário executar a aplicação cliente que efetuará a autenticação. Após o passo anterior, é efetuada uma chamada, mensagem INVITE, para a IPTV *Interface Access Management*, ao qual a interface responde, após um conjunto mensagem trocadas, responde com um OK, a qual possui os canais e a chave de desencriptação. Após a receção dos canais e a chave, a aplicação procederá à receção e desencriptação dos conteúdos televisivos.

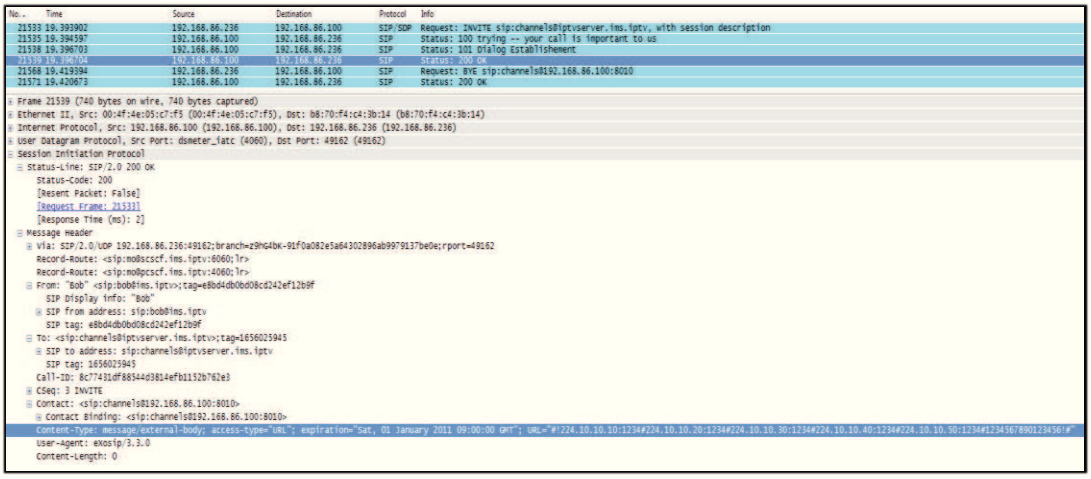


Figura 52 - Pedido de canais disponíveis e chave de desencriptação

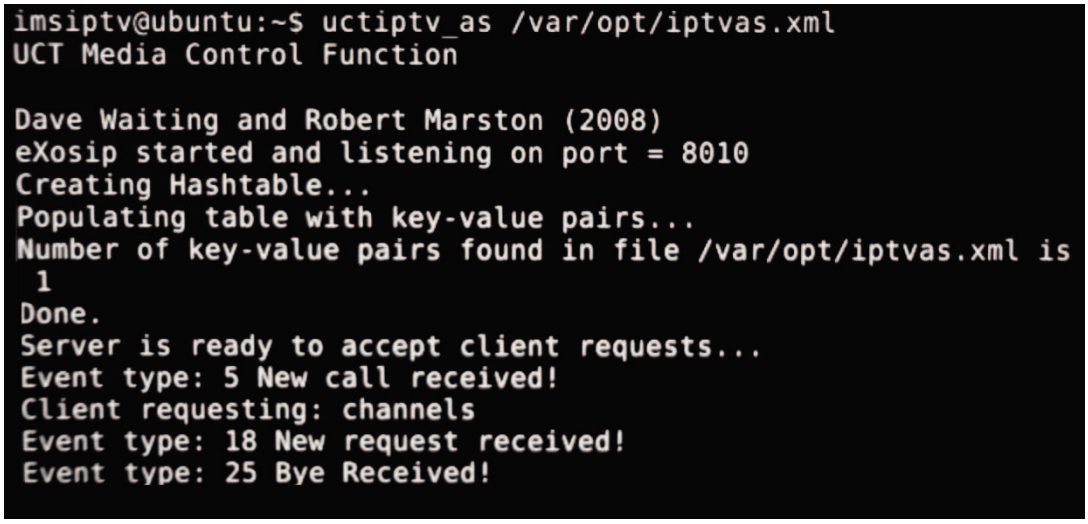


Figura 53 - uctiptv_as à espera de pedidos e resposta a um pedido

Conclusão: A Figura 52 e Figura 53 representam os pedidos e respostas do cliente para a *IPTV Interface Access Management* (UCTIPTV_AS) em que se verifica que o cliente requer a lista de canais e esta lhe é devolvida.

5.2 Testes Efetuados à Aplicação Cliente

Os testes efetuados à aplicação cliente têm como objetivo de comprovar o correto funcionamento da aplicação cliente em determinadas situações, podendo-se ainda encontrar situações de em que os testes não obtiveram o resultado pretendido e chegar ao cerne do problema para o podermos resolver.

5.2.1 Visualizar canais sem a chave correta

Objetivos: É pretendido com este teste averiguar o comportamento da aplicação desenvolvida e de outros *players* de vídeo, quando se tenta obter o *stream* de vídeo e não se possui a chave de desencriptação correta, cujo comportamento deverá ser a não visualização dos conteúdos.

Descrição: Para efetuar este teste é necessário ter em execução o servidor de *streaming* que envia os *streams* codificados para os clientes sem disponibilizar a chave de desencriptação.

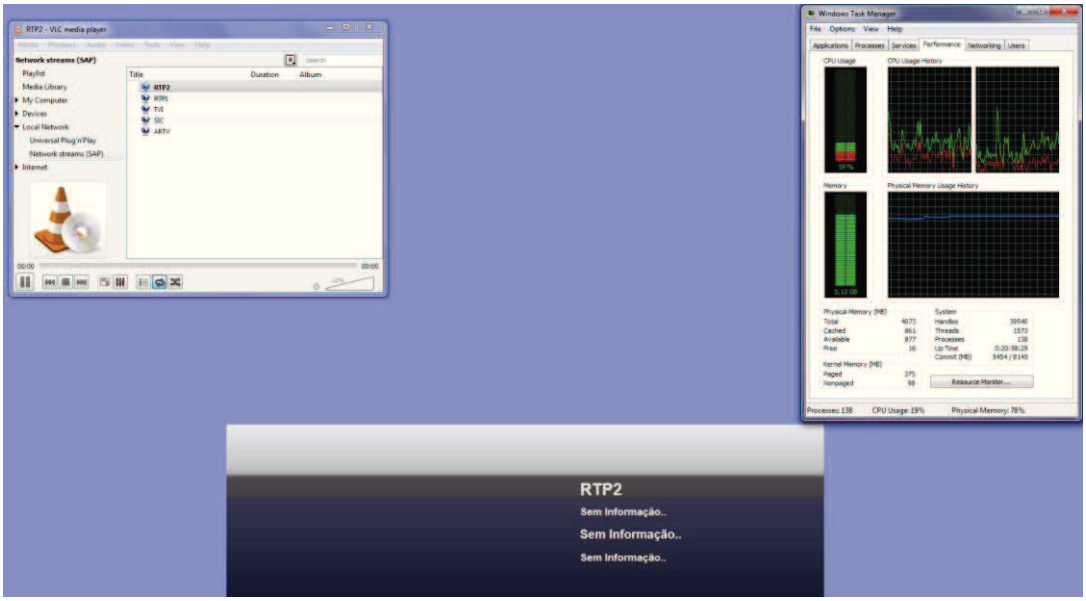


Figura 54 - Visualizar canais sem a chave correta

Conclusão: Ao analisar a Figura 54 pode verificar-se que se está a tentar visualizar três canais em simultâneo através de dois *players* distintos. Na aplicação desenvolvida, tenta-se visualizar o canal 1, supostamente em ecrã inteiro e o canal 2, em pré-visualização.

Foi ainda usado o VLC para garantir que os *streams* não eram desencriptados sem a chave correta para o efeito, como se pode comprovar na figura.

5.2.2 Visualizar canais com a chave correta

Objetivos: É pretendido com este teste visualizar os *streams* de vídeo, quando é fornecida a chave de desencriptação correta.

Descrição: Para efetuar este teste é necessário ter em execução o servidor de *streaming* que envia os *streams* codificados para os clientes e é necessário iniciar a aplicação cliente e ter acesso à chave de desencriptação correta.

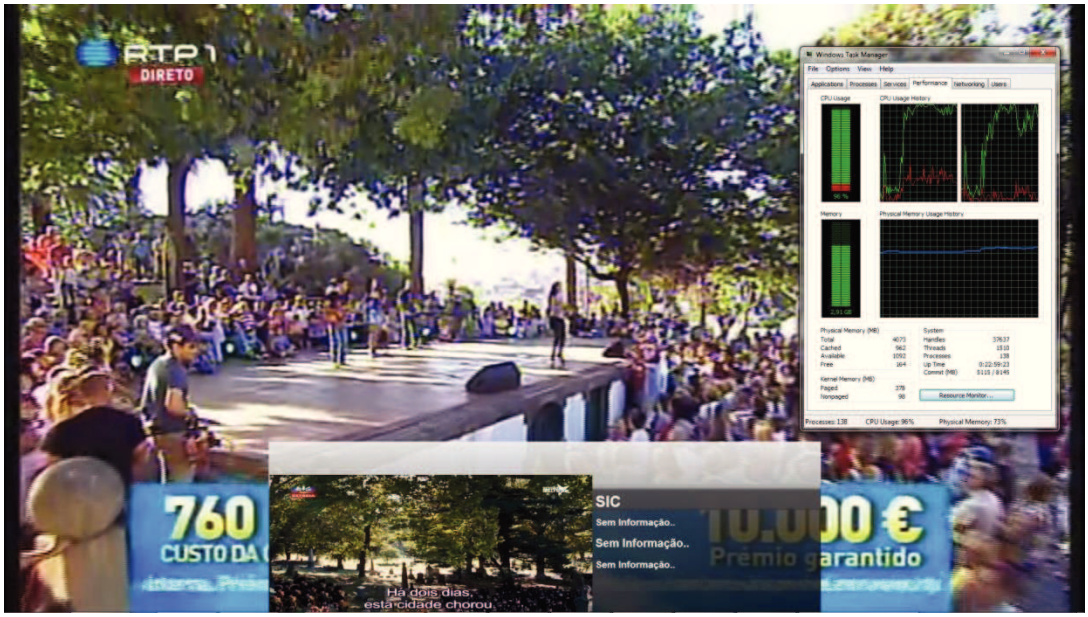


Figura 55 - Visualizar canais com a chave correta

Conclusão: Ao analisar a Figura 55 pode verificar-se que se está a tentar visualizar três canais em simultâneo através de dois *players* distintos. Na aplicação desenvolvida, tenta-se visualizar o canal 1, supostamente em ecrã inteiro e o canal 2, em pré-visualização.

Foi ainda usado o VLC para garantir que os *streams* não eram desencriptados sem a chave correta para o efeito, como se pode comprovar na figura.

5.2.3 Envio de mensagens instantâneas para utilizador autenticado

Objetivos: O objetivo deste teste é averiguar o funcionamento do envio de mensagens instantâneas (via *Chat*) entre clientes registados no servidor.

Descrição: Para efetuarmos o teste é necessário que dois clientes/utilizadores se autentiquem perante o servidor. Após a autenticação ambos podem comunicar um com o outro. Para este teste foi usada a aplicação desenvolvida (utilizador Bob) e o X-Lite 4 (utilizador Sofia).

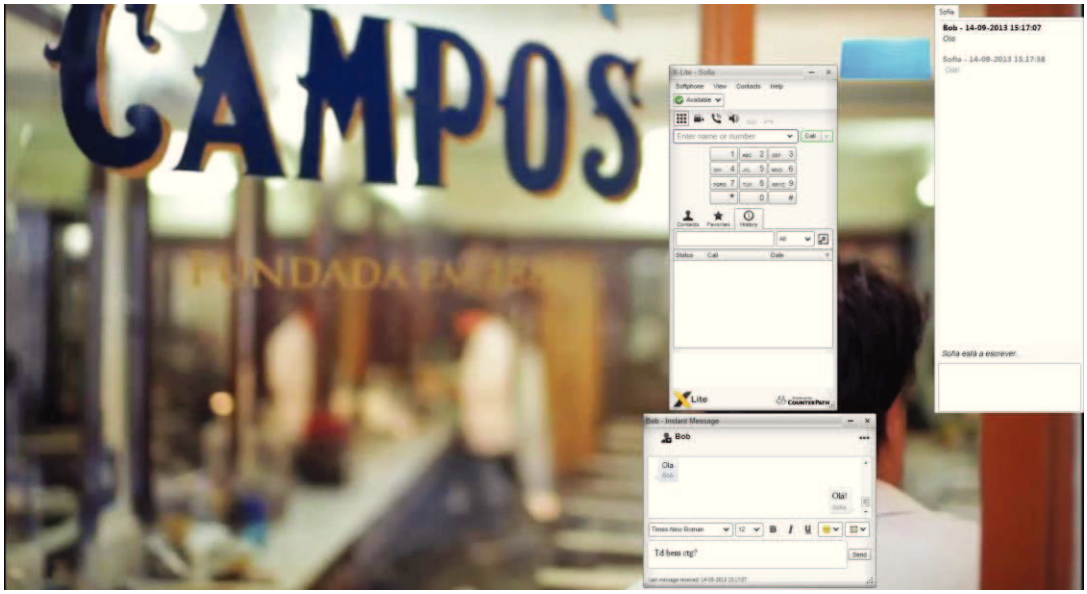


Figura 56 - Envio de mensagens instantâneas

Conclusão: Na Figura 56 averigua-se que o envio de mensagens instantâneas funciona corretamente, e que é possível manter uma conversação entre utilizadores e com o uso aplicações direcionadas para diferentes âmbitos.

5.2.4 Envio de mensagens instantâneas para utilizador não autenticado

Objetivos: O objetivo deste teste é constatar que uma mensagem instantânea ao não ser entregue ao destinatário o utilizador é informado para o sucedido.

Descrição: Para efetuarmos o teste é necessário termos um cliente autenticado perante o servidor. Após a autenticação o cliente acede aos seus contatos e seleciona um deles (a Sofia) e redige uma mensagem e procede ao seu envio.

Figura 57 - Falha no envio de mensagens instantâneas

Conclusão: Na Figura 57 averigua-se que ao falhar o envio de mensagens instantâneas, o utilizador é informado desse mesmo facto.

5.2.5 Usabilidade

Os testes efetuados, via Android *tablet*, têm como objetivo aferir o nível de usabilidade do *interface* desenvolvido para a plataforma. Para isso, a interface foi utilizada por um conjunto de 15 utilizadores que seguiram um guião de tarefas [Anexo III], ao mesmo tempo que íamos registando o desempenho dos utilizadores na execução das mesmas [Anexo IV].

De salientar que a faixa etária dos utilizadores que executaram o teste, se situa entre os 18 e os 35 anos, pelo que não se pode aferir o desempenho da interface, face à abrangência da faixa etária que serve como alvo desta proposta. Os mesmos teriam total efeito caso realizados com utilizadores de diversificadas faixas etárias e diversos níveis de literacia tecnológica.

Tarefas
Mudar de Canal
Pré-visualizar um canal
Navegar no Guia de Programação
Aceder ao contatos
Escolher um contato e escrever uma mensagem
Enviar uma mensagem
Colocar a aplicação sem som
Desligar a aplicação

Tabela 3 - Tarefas a Realizar

De seguida apresentamos os parâmetros usados para a realização dos mesmos:

- Cada utilizador realizou o teste individualmente e de forma cronometrada;
- O teste foi realizado num computador com recurso à aplicação *Android* para *tablet* (*Android Remote Control*);
- Os resultados foram registados à medida que cada teste foi executado.

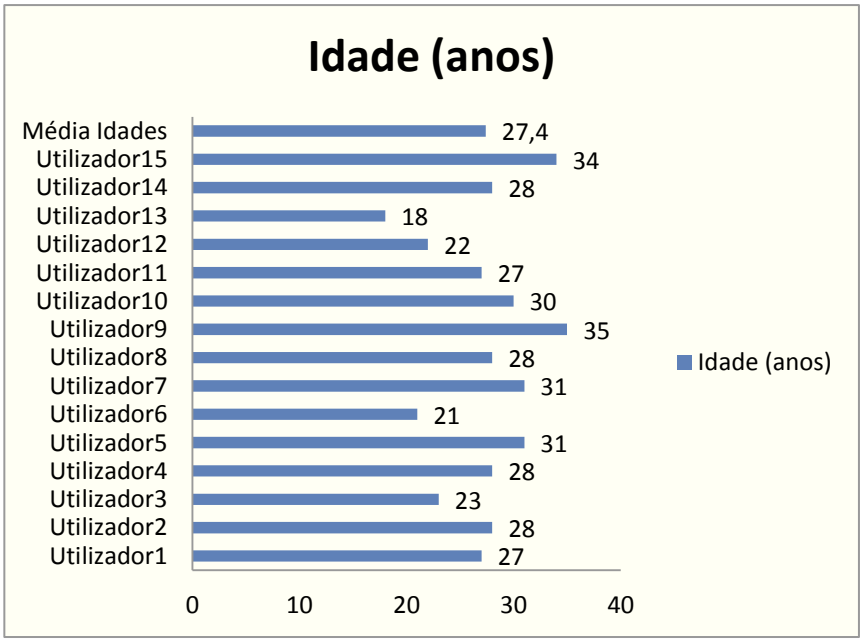


Figura 58 - Idades dos Utilizadores

A Figura 58 apresenta os valores representativo das idades por utilizador e a média das mesmas.

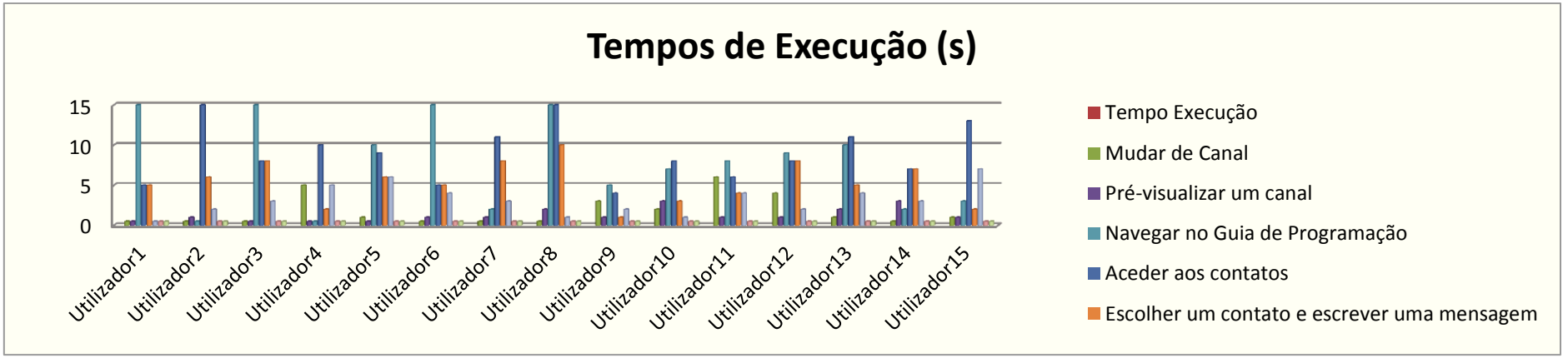


Figura 59 - Tempos de Execução

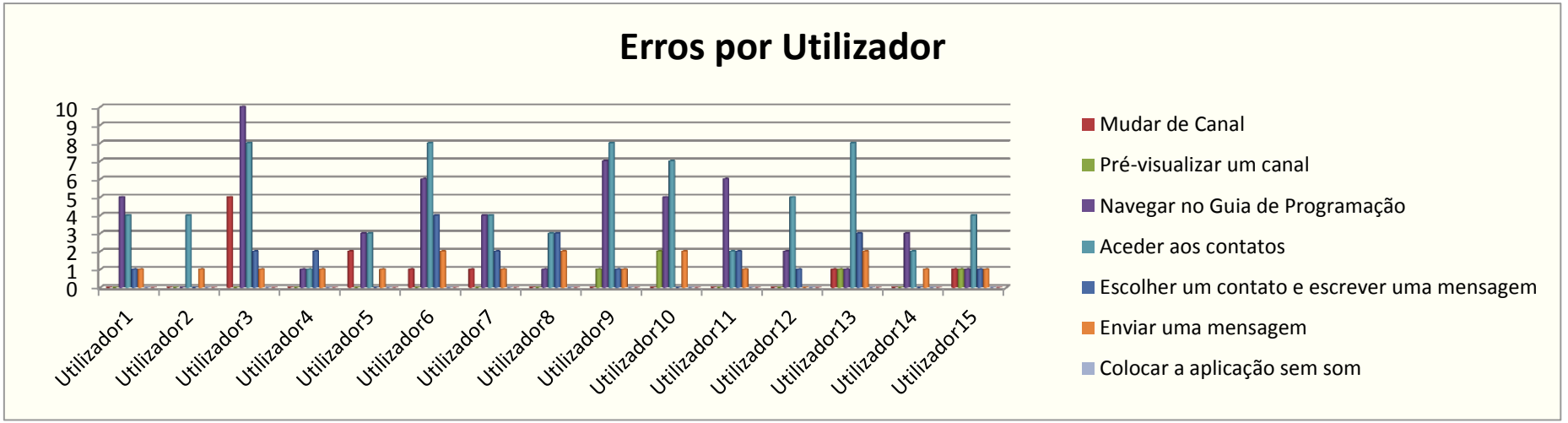


Figura 60 - Erros por Utilizador

Na Figura 59 é apresentado o tempo, em segundos que cada utilizador levou a executar cada tarefa. Analisando os resultados, conclui-se que as tarefas que em média demoraram mais tempo a serem executadas, foi "Navegar no Guia de Programação" e "Aceder aos Contatos".

A média de tempo consumido numa tarefa que se revelou a mais baixa foram, "Colocar a Aplicação sem som" e "Desligar a aplicação". Estes resultados prendem-se com a complexidade das tarefas em si e do tempo necessário a cada utilizador para completar a sua curva de aprendizagem no que diz respeito à utilização de uma solução deste tipo. Não obstante aos fatores mencionados anteriormente, verifica-se que os resultados estão dentro dos valores esperados, o que valida a interface como uma opção válida no âmbito do trabalho exposto nesta dissertação.

A Figura 60 apresenta os erros cometidos por cada utilizador na realização das diversas tarefas. Ao analisarmos os resultados obtidos, apercebemo-nos que as tarefas que deram origem a mais erros, foram "Aceder aos Contatos" e "Navegar no Guia de Programação".

O fato de ambas as tarefas terem tido resultados elevados em termos de erros, na nossa opinião, deve-se ao interface do *Android Remote Controller* não replicar na íntegra a interface da aplicação cliente. Este fato leva a que a curva de aprendizagem seja prolongada, levando a um maior número tentativas e consequentemente erros por parte do utilizador e consequentemente a um maior número de erros observados durante a execução dos testes.

Em síntese, os resultados obtidos são indicadores de que a interface da aplicação cliente é válida, não descartando a necessidade de um tempo de aprendizagem adequado a tarefas com este tipo de complexidade.

5.3 Síntese

Os testes e resultados apresentados neste capítulo, forneceram informação importante para a avaliação geral da nossa solução. Através deles, constatámos que o nosso módulo do servidor está de acordo com os requisitos da arquitetura implementada, permitindo a escalabilidade e adaptabilidade da mesma a diversas infraestruturas e necessidades.

De salientar também que os testes efetuados à comunicação entre os diversos módulos constituintes da solução revelam não só a concordância com os requisitos definidos anteriormente, como a eficácia dos protocolos utilizados.

No seguimento da análise, ficou claro que a aplicação cliente desenvolvida, é de fácil utilização e vai de acordo com o âmbito proposto pela nossa solução, o de fornecer uma plataforma com acesso a multimédia e serviços de comunicação. As vantagens inerentes de uma solução deste tipo são claras, no sentido em que a componente SIP permite não só o uso de um conjunto diversificado de serviços, sejam eles os intrínsecos ao protocolo ou externos ao mesmo.

Por último, é notório que a aplicação *Android Remote Controller* carece ainda de algumas afinações a nível de integração com a aplicação cliente, conclusão que é fruto do resultado de alguns dos testes efetuados, os quais revelam que o interface não se encontra completamente em consonância com a usabilidade pretendida para a solução cliente.

Capítulo 6 - Conclusão e Trabalho Futuro

Neste capítulo é efetuado um balanço sobre todo o trabalho desenvolvido no âmbito da presente dissertação. No final do capítulo, são indicados alguns aspetos que podem ser melhorados num trabalho futuro.

6.1 Conclusão

Tendo em conta os mecanismos que a evolução constante das tecnologias da comunicação e informação fornecem atualmente, esta dissertação tinha como principal objetivo apresentar uma solução de IPTV integrada com o protocolo SIP, tudo isto com recurso ao IMS. Esta abordagem tinha como foco otimizar o uso dos serviços mencionados anteriormente e ao mesmo tempo usufruir das vantagens inerentes ao uso do SIP. Deste modo, foi estruturada uma arquitetura com diferentes módulos, os quais possibilitassem não só o fornecimento de um serviço IPTV *standard,* como também a agregação de serviços *out of the box* ao mesmo, permitindo deste modo aos utilizadores um conjunto diversificado de ferramentas de comunicação entre eles.

Com o intuito de cumprir os objetivos propostos, foi elaborada uma pesquisa às soluções existentes no âmbito e às tecnologias intrínsecas da solução. Ambas apresentadas no capítulo 2 do presente documento.

Após a análise das tecnologias, foi especificada uma arquitetura geral para suprimir a problemática estudada e uma arquitetura a ser implementada, como prova de conceito, ambas descritas no capítulo 3.

A arquitetura implementada é constituída por um conjunto de módulos que servem de suporte à aplicação cliente, a qual permite, visualizar e pré-visualizar conteúdos televisivos, consultar o Guia TV, controlar a emissão e a comunicação entre utilizadores através do serviço de *Chat* incorporado.

As vantagens desta solução prendem-se com o facto de permitir aos utilizadores da mesma, a comunicação em tempo real enquanto realizam a visualização dos conteúdos televisivos, ponto que combate o isolamento e exclusão social. Por outro lado, toda a infraestrutura de suporte desenvolvida permite a integração de uma forma facilitada, de serviços normalmente só disponíveis através de outro tipo de

terminais e mecanismos de suporte, serviços socialmente relevantes tal como a videochamada, agregando-os numa interface comum.

Tendo em conta não só os fatores supracitados, como também os fatores económicos relacionados com a instalação e manutenção de um serviço deste tipo, a solução proposta visa ser uma infraestrutura de baixo custo. Esta preocupação com os custos é refletida no custo final para o consumidor, tornando este serviço uma ferramenta de âmbito social que permite aos utilizadores usufruírem de uma interação social com uma maior facilidade.

Em suma as vantagens inerentes desta solução são claras e viabilizam a mesmo como um serviço de IPTV fiável e inovador. Não obstante, é importante que seja desenvolvido uma bateria de testes de maior dimensão, assim como um estudo mais acentuado sobre a interface de controlo e os mecanismos de suporte à mesma.

6.2 Trabalho Futuro

A solução desenvolvida satisfaz os requisitos básicos da arquitetura proposta, contudo foram identificados alguns aspetos que futuramente poderão vir a ser melhorados. Assim sendo, de seguida são apresentadas algumas propostas a implementar num trabalho futuro.

De forma a criar uma aplicação transversal a sistemas operativos é proposta a migração da aplicação cliente para java, potenciando assim a sua utilização nos diversos sistemas operativos existentes.

Seria útil ainda redefinir a aplicação cliente e a aplicação *Android Remote Controller* ao nível de *design* gráfico para que seja de fácil utilização para as distintas faixas etárias, combatendo desse modo a exclusão social e infoexclusão existente nos meios rurais.

Por último, considerando que a aplicação cliente implementa as funcionalidades base da arquitetura, é útil desenvolver novas funcionalidades, tais como: videochamada, videoconferência, adição/remoção de contatos, estado atual de cada contacto (ausente, ocupado, *offline*) e permitir a definição de estados no próprio cliente

Bibliografia

[1] "Televisão Digital Terrestre -TDT." [Online]. Available: http://tdt.telecom.pt/. [Accessed: 10-May-2013].

[2] DECO, "TDT: fracasso da PT e ICP-ANACOM confirmado," 2013. [Online]. Available: http://www.deco.proteste.pt/tecnologia/televisores/comunicado-de-imprensa/fracasso-na-tdt-confirmado-por-62-por-cento. [Accessed: 12-Jul-2013].

[3] Jornal de Notícias, "Moradores de Alferce revoltados por ficarem sem televisão," 2012. [Online]. Available: http://www.jn.pt/PaginaInicial/Sociedade/Media/Interior.aspx?content_id=2258099&page=-1. [Accessed: 15-Aug-2013].

[4] E. Portuguesa, "Faculdade de Ciências Humanas O Envelhecimento da População : Dependência , Ativação e Qualidade," 2012.

[5] M. J. V. Rosa, "ENVELHECIMENTO: ENCARGO ou OPORTUNIDADE ECONÓMICA? Envelhecimento da população de Portugal: enquadramento demográfico e social," 2012.

[6] INE, *20 1 1*. 2011.

[7] O. Neuwirt, J. Da Silva, D. Abbadessa, and F. Winkler, "Towards a New User Experience in IPTV : Convergence Services and Simpler E-commerce on IMS-based IPTV," vol. 3, no. 4, pp. 103–106, 2008.

[8] CISCO, "The Evolving IPTV Service Architecture," pp. 1–12, 2007.

[9] L. Correia and C. Freire, "Redes Wireless do Futuro IPTV," 2011.

[10] VideoLan, "VideoLAN." [Online]. Available: http://www.videolan.org/vlc/. [Accessed: 16-Jul-2013].

[11] VideoLan, "VideoLan Streaming Solution Arq." [Online]. Available: http://www.videolan.org/vlc/streaming.html. [Accessed: 15-Jul-2013].

[12] VideoLan, "VLC-VLM." [Online]. Available: http://www.videolan.org/doc/streaming-howto/en/ch05.html. [Accessed: 05-Aug-2013].

[13] C. Freire, L. Correia, L. Marcelino, C. Silva, C. Rabadão, and A. Pereira, "Menu and Context Based Interfaces Evaluation for Mobile TV," vol. 00, 2012.

[14] "3gpp - IMS." [Online]. Available: http://www.3gpp.org/Technologies/Keywords-Acronyms/article/ims. [Accessed: 10-Jul-2013].

[15] T. Specification and G. Services, "3gpp ts 23.228," vol. 0, no. Release 12, pp. 1–293, 2013.

[16] T. Specification, “IP Multimedia Subsystem (IMS);,” vol. 0, 2013.

[17] “SIP - RFC 3680,” 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3680.txt?number=3680. [Accessed: 06-Feb-2013].

[18] “Session Initiation Protocol (SIP) Extension for Instant Messaging,” 2002. [Online]. Available: http://www.ietf.org/rfc/rfc3428.txt. [Accessed: 05-May-2013].

[19] “SIP - RFC 3665,” 2003. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3665.txt. [Accessed: 06-Jun-2013].

[20] “Wiki - IP Multimedia Subsystem.” [Online]. Available: https://en.wikipedia.org/wiki/IP_Multimedia_Subsystem. [Accessed: 11-Jul-2013].

[21] “Open-Ims Core.” [Online]. Available: http://www.openimscore.org/. [Accessed: 13-Nov-2012].

[22] “Fokus Fraunhofer.” [Online]. Available: http://www.fokus.fraunhofer.de/en/fokus_testbeds/open_ims_playground/components/index.html. [Accessed: 10-Jul-2013].

[23] “FHoSS.” [Online]. Available: http://www.fokus.fraunhofer.de/en/fokus_testbeds/open_ims_playground/components/osims/fhoss/index.html. [Accessed: 11-Jul-2013].

[24] “P-CSCF.” [Online]. Available: http://www.fokus.fraunhofer.de/en/fokus_testbeds/open_ims_playground/components/osims/cscfs/p_cscf/index.html. [Accessed: 11-Jul-2013].

[25] “I-CSCF.” [Online]. Available: http://www.fokus.fraunhofer.de/en/fokus_testbeds/open_ims_playground/components/osims/cscfs/i_cscf/index.html.

[26] “S-CSCF.” [Online]. Available: http://www.fokus.fraunhofer.de/en/fokus_testbeds/open_ims_playground/components/osims/cscfs/s_cscf/index.html. [Accessed: 11-Jul-2013].

[27] “OSIMS.” [Online]. Available: http://nam.ece.upatras.gr/ppe/?q=node/2. [Accessed: 11-Jul-2013].

[28] “IMS in the Cloud - Clearwater.” [Online]. Available: http://www.projectclearwater.org/. [Accessed: 11-Jul-2013].

[29] “IMS in the Cloud - Clearwater.” [Online]. Available: http://www.projectclearwater.org/technical/clearwater-architecture/. [Accessed: 11-Jul-2013].

[30] “DRM Info,” *30-09-2003*, 2003. [Online]. Available: http://drm.info/pt/what-is-drm. [Accessed: 15-Jul-2013].

[31] C. E. N. Final, “Digital Rights Management Final Report,” no. September, 2003.

[32] "DRM Arquitecture." [Online]. Available: http://media.soundonsound.com/sos/aug03/images/drmhelixdiagram.l.jpg. [Accessed: 16-Jul-2013].

[33] "DVB - Digital Video Broadcasting." [Online]. Available: http://www.dvb.org/technology/standards/. [Accessed: 16-Jul-2013].

[34] K. A. Computer, "QuickTime Suite - Administrator ' s Guide," 2002.

[35] "Darwin Streaming Server." [Online]. Available: http://dss.macosforge.org/. [Accessed: 15-Jul-2013].

[36] "LIVE555." [Online]. Available: http://www.live555.com/liveMedia/. [Accessed: 15-Jul-2013].

[37] "Live555 Media Server." [Online]. Available: http://www.live555.com/mediaServer/. [Accessed: 17-Jul-2013].

[38] "LIVE555 proxyServer." [Online]. Available: http://www.live555.com/proxyServer/. [Accessed: 18-Jul-2013].

[39] "LIVE555 liveCaster." [Online]. Available: http://www.live555.com/liveCaster/. [Accessed: 17-Jul-2013].

[40] "LIVE555 vobStreamer." [Online]. Available: http://www.live555.com/vobStreamer/. [Accessed: 18-Jul-2013].

[41] IETF, "MBone." [Online]. Available: http://datatracker.ietf.org/wg/mboned/charter/. [Accessed: 15-Jul-2013].

[42] "VLC - CSS." [Online]. Available: http://www.pcworld.com/article/168622/VLC_Video_Player_New_DVD_Copying_Feature_Could_Run_Afoul_of_the_MPAA.html. [Accessed: 05-Aug-2013].

[43] IETF, "RTSP." [Online]. Available: http://www.ietf.org/rfc/rfc2326.txt. [Accessed: 15-Aug-2013].

[44] IETF, "RTP." [Online]. Available: http://www.ietf.org/rfc/rfc3550.txt. [Accessed: 15-Aug-2013].

[45] CISCO, "IP Multicast - CISCO." [Online]. Available: http://www.cisco.com/en/US/products/ps6552/products_ios_technology_home.html. [Accessed: 15-Aug-2013].

[46] CISCO, "Delivering Multicast Video Over ADSL."

[47] C. Schluting, "Networking 101 - Multicast." [Online]. Available: http://www.enterprisenetworkingplanet.com/netsp/article.php/3623181/Networking-101--Understanding-Multicast-Routing.htm. [Accessed: 19-Aug-2013].

[48] InternetStreams, “Multicast-Unicast.” [Online]. Available: http://www.slideshare.net/internetstreams/multicast-vs-unicast-diagram-presentation. [Accessed: 16-Aug-2013].

[49] CISCO, “PIM.” [Online]. Available: http://www.cisco.com/en/US/docs/ios/solutions_docs/ip_multicast/White_papers/mcst_ovr.html#wp1009068. [Accessed: 20-Aug-2013].

[50] IETF, “PIM-DM.”

[51] IETF, “PIM-SM.” [Online]. Available: http://tools.ietf.org/html/rfc4601. [Accessed: 19-Aug-2013].

[52] IETF, “SIP 3261.” [Online]. Available: http://www.ietf.org/rfc/rfc3261.txt. [Accessed: 15-Jul-2013].

[53] VideoLan, “VLC Features.” [Online]. Available: http://www.videolan.org/vlc/features.html. [Accessed: 15-Jul-2013].

Capítulo 7 - Anexos

Anexo I – Mensagens SIP

Nos diagramas de sequência seguintes são expostas as mensagens SIP mais importantes estudadas no âmbito desta dissertação. Estes diagramas revelam a forma de interação entre clientes e o servidor SIP.

É retratada a autenticação SIP (REGISTER), o INVITE, para o estabelecimento de uma chamada entre clientes e a MESSAGE, para o chat entre dois clientes.

1. *REGISTER*

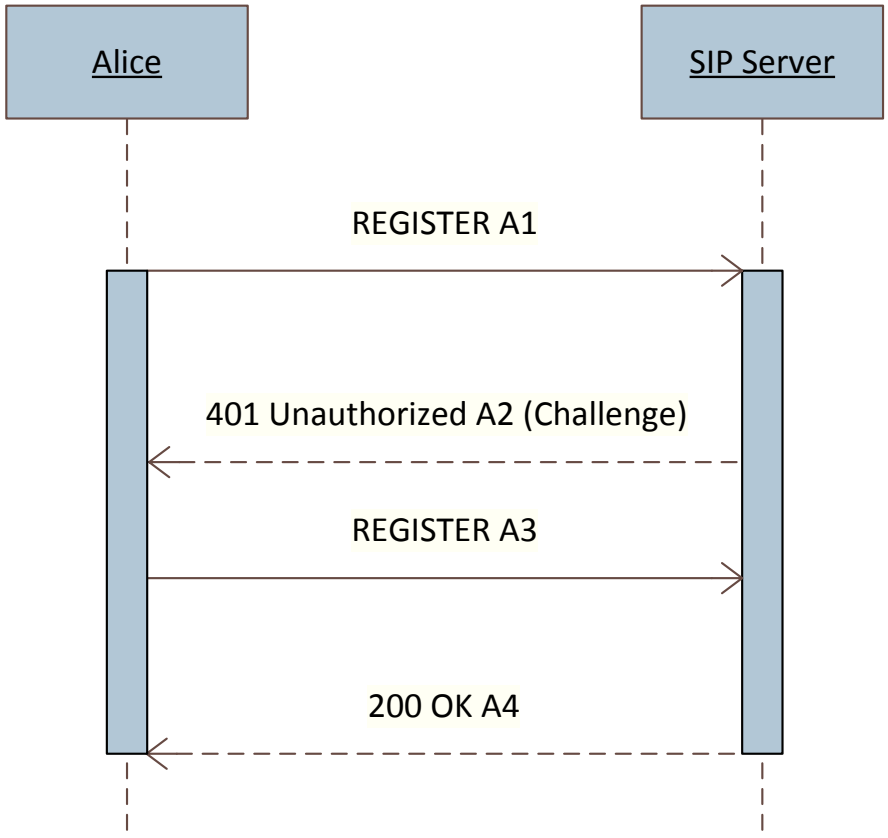


Diagrama de Sequência 1 - SIP REGISTAR

Ao ser analisado o diagrama de sequência anterior, podemos observar que para um cliente (Alice) proceder a uma autenticação ou registo num servidor SIP, terá que

efetuar uma série de passos bem delineados. Esta, terá que remeter uma mensagem REGISTER para o SIP Server, o qual retorquirá também com uma mensagem REGISTER, mas com o código 401 que não é mais que um challange para a Alice. A Alice ao interpretar a mensagem recebida, expede novamente uma mensagem REGISTER erigida com campos específicos obtidos da mensagem anteriormente recebida. Após o servidor rececionar e autenticar esta mensagem (A3), este envia ao cliente um 200 OK a confirmar a autenticação ou registo bem-sucedido.

2. INVITE

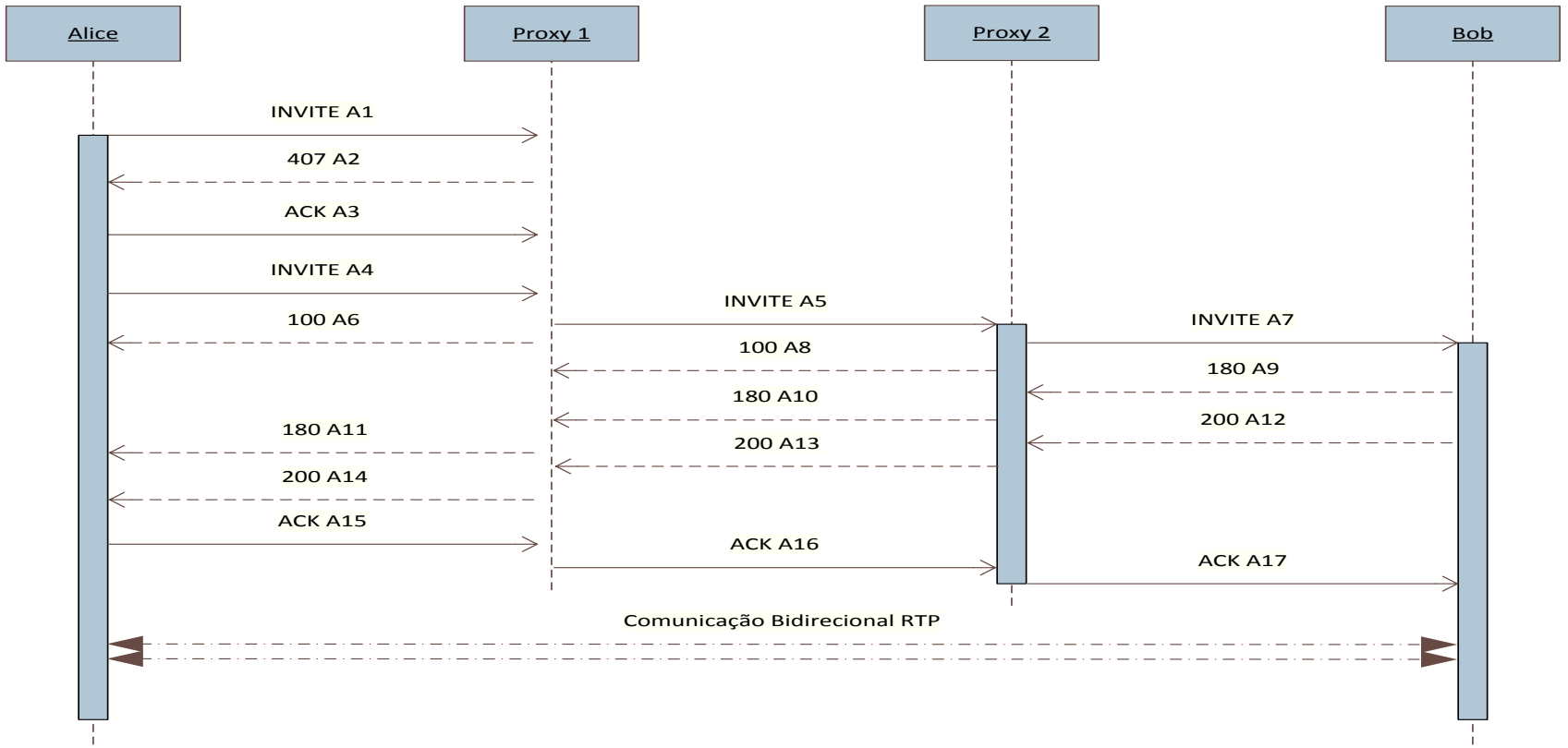


Diagrama de Sequência 2 – INVITE

Este diagrama de sequência apresenta o cenário existente na solução desenvolvida, em que existem dois proxies intermédios entre clientes.

Nos passos seguintes descritos, omite-se a sequência de mensagens entre os próprios proxies e os seus intervenientes.

Tendo em conta o fato supracitado, no esquema acima, constatamos que a Alice, envia para o Bob um INVITE (A1), o qual responde com um RINGING 180 (A9). A Alice, após a receção do RINGING, responde com um ACK (A15) para o Bob. Após estes passos os dois estabelecem uma ligação de RTP multimédia bidirecional.

3. MESSAGE

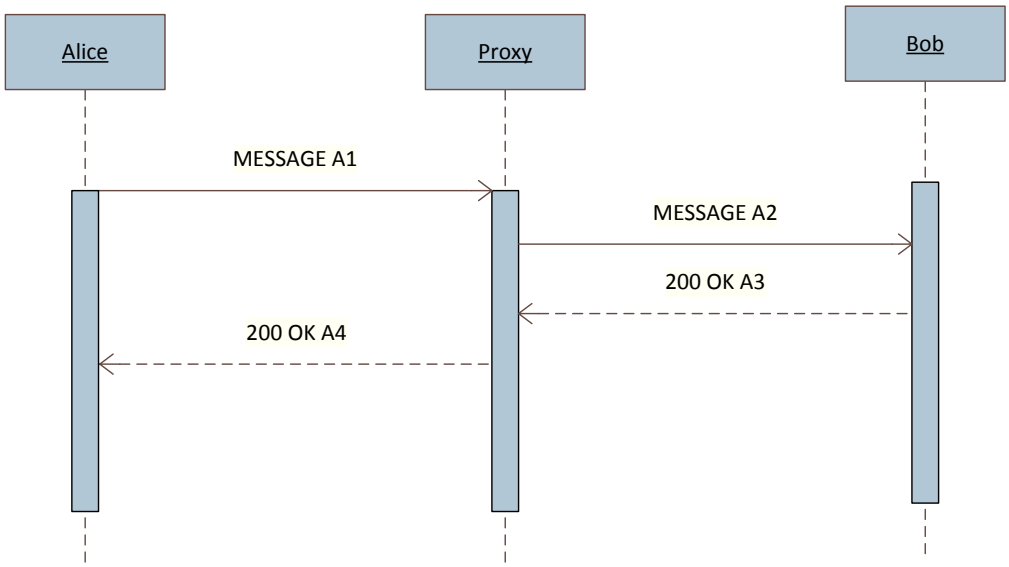


Diagrama de Sequência 3 - MESSAGE

O diagrama anterior revela como se processa a sequência de envio de mensagens instantâneas entre clientes.

A Alice envia a MENSAGEM (A1) para o Proxy, o qual encaminha essa mesma mensagem (A2) para o Bob. O Bob, após a receção envia para o Proxy o 200 OK (A3) para o Proxy. O Proxy de seguida encaminha o 200 OK (A4) para a Alice, de forma a ser informada que a mensagem foi entregue com sucesso.

Anexo II – Características *VLC Media Player*

Formatos de Entrada

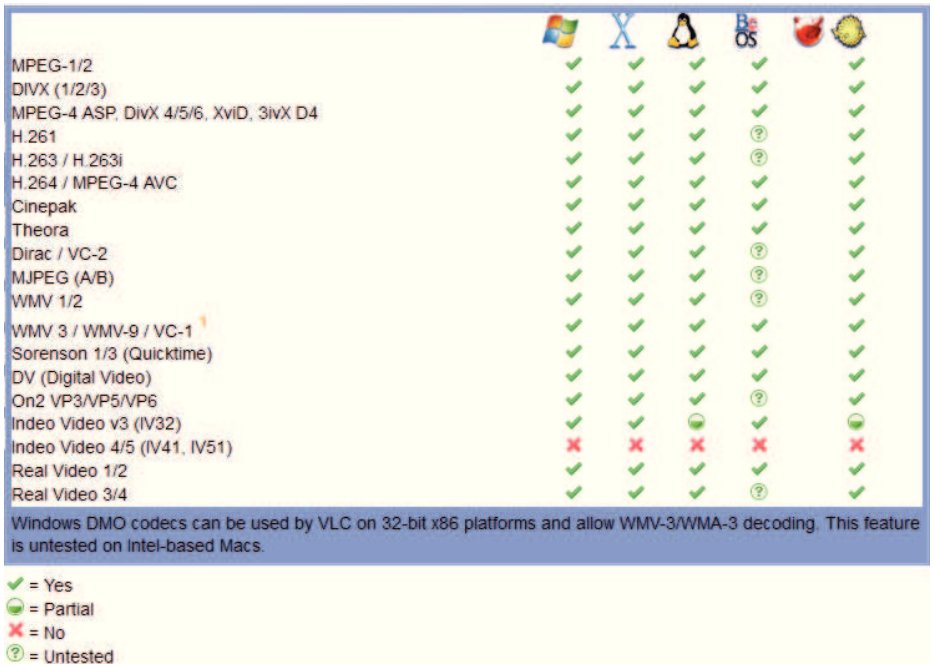
Input media	UDP/RTP Unicast	✔	✔	✔	✔	✔
	UDP/RTP Multicast	✔	✔	✔	✘	✔
	HTTP / FTP	✔	✔	✔	✔	✔
	MMS	✔	✔	✔	✔	✔
	TCP/RTP Unicast	✔	✔	✔	✔	✔
	DCCP/RTP Unicast	✘	✘	✔	✘	✘
	File	✔	✔	✔	✔	✔
	DVD Video [1]	✔	✔	✔	✔	✔
	Video CD / VCD	✔	✔	✔	✘	✔
	SVCD [2]	◒	◒	◒	✘	◒
	Audio CD (no DTS-CD)	✔	✔	✔	✘	✔
	DVB (Satellite, Digital TV, Cable TV)	✔	◒ EyeTV [3]	✔	✔	✘
	MPEG encoder [4]	✔	✘	✔	✘	✘
	Video acquisition	✔ Direct Show	✔ QTKit [5]	✔ V4L, V4L2	✘	✘
Input formats	MPEG (ES,PS,TS,PVA,MP3)	✔	✔	✔	✔	✔
	AVI	✔	✔	✔	✔	✔
	ASF / WMV / WMA	✔	✔	✔	✔	✔
	MP4 / MOV / 3GP	✔	✔	✔	✔	✔
	OGG / OGM / Annodex	✔	✔	✔	✔	✔
	Matroska (MKV)	✔	✔	✔	✔	✔
	Real	◒	◒	◒	◒	◒
	WAV (including DTS)	✔	✔	✔	✔	✔
	Raw Audio: DTS, AAC, AC3/A52	✔	✔	✔	✔	✔
	Raw DV	✔	✔	✔	✔	✔
	FLAC	✔	✔	✔	✔	✔
	FLV (Flash)	✔	✔	✔	?	✔
	MXF	✔	✔	✔	?	✔
	Nut	✔	✔	✔	?	✔
	Standard MIDI / SMF	✔	✔	✔	✔	✔
	Creative™ Voice	✔	✔	✔	✔	✔

DVD decryption is done through the libdvdcss library.
VLC on GNU/Linux, Solaris, and Microsoft Windows has playback control support via libcdio and libvcdinfo. On other platforms, SVCD support varies depending on the availability of these libraries. (Volunteers for adding support are always welcome.). Handling still frames (often used in menus) and switching between different video formats is problematic.
On Mac OS X 10.4 or later, VLC is able to grab video and audio from EyeTV applications and therefore all EyeTV-compatible capture devices. The user needs to install a plugin to EyeTV.app in order to use this feature. Guidance is provided in the User Interface. Requires VLC 0.9.0 or later.
VLC for GNU/Linux supports V4L2 compatible encoding cards as well as two kinds of MPEG-2 encoding cards: Hauppauge WinTV-PVR-250/350 and Visiontech Kfir.
VLC can capture video from internal iSights on Mac OS X 10.5 or later (video only) since version 0.9.0. VLC 1.2 adds capturing from all devices supported by QTKit. It also enables audio capturing support on Mac OS X 10.6 and later.

✔ = Yes
◒ = Partial
✘ = No
? = Untested

Figura 61 - VLC, formatos de entrada[53]

Formatos de Vídeo



MPEG-1/2	✔	✔	✔	✔		✔
DIVX (1/2/3)	✔	✔	✔	✔		✔
MPEG-4 ASP, DivX 4/5/6, XviD, 3ivX D4	✔	✔	✔	✔		✔
H.261	✔	✔	✔	?		✔
H.263 / H.263i	✔	✔	✔	?		✔
H.264 / MPEG-4 AVC	✔	✔	✔	✔		✔
Cinepak	✔	✔	✔	✔		✔
Theora	✔	✔	✔	✔		✔
Dirac / VC-2	✔	✔	✔	?		✔
MJPEG (A/B)	✔	✔	✔	?		✔
WMV 1/2	✔	✔	✔	?		✔
WMV 3 / WMV-9 / VC-1 [1]	✔	✔	✔	✔		✔
Sorenson 1/3 (Quicktime)	✔	✔	✔	✔		✔
DV (Digital Video)	✔	✔	✔	✔		✔
On2 VP3/VP5/VP6	✔	✔	✔	?		✔
Indeo Video v3 (IV32)	✔	✔	◒	✔		◒
Indeo Video 4/5 (IV41, IV51)	✘	✘	✘	✘		✘
Real Video 1/2	✔	✔	✔	✔		✔
Real Video 3/4	✔	✔	✔	?		✔

Windows DMO codecs can be used by VLC on 32-bit x86 platforms and allow WMV-3/WMA-3 decoding. This feature is untested on Intel-based Macs.

✔ = Yes
◒ = Partial
✘ = No
? = Untested

Figura 62 - VLC, formatos de vídeo[53]

Formatos de Áudio

MPEG Layer 1/2	✔	✔	✔	✔		✔
MP3 - MPEG Layer 3	✔	✔	✔	✔		✔
AAC - MPEG-4 part3	✔	✔	✔	✔		✔
Vorbis	✔	✔	✔	✔		✔
AC3 - A/52 (Dolby Digital)	✔	✔	✔	✔		✔
E-AC-3 (Dolby Digital Plus)	✔	✔	✔	✔		✔
MLP / TrueHD">3	✔	✔	✔	✔		✔
DTS	✔	✔	✔	✔		✔
WMA 1/2	✔	✔	✔	✔		✔
WMA 3 [1]	✔	✔	✔	✘		✘
FLAC	✔	✔	✔	✔		✔
ALAC	✔	✔	✔	✔		✔
Speex	✔	✔	✔	?		✔
Musepack / MPC	✔	✔	✔	?		✔
ATRAC 3	✔	✔	✔	?		✔
Wavpack	✔	✔	✔	✔		✔
Mod (.s3m, .it, .mod)	✔	✔	✔	?		✔
TrueAudio (TTA)	✔	✔	✔	✔		✔
APE (Monkey Audio)	✔	✔	✔	✔		✔
Real Audio [2]	◒	◒	◒	?		◒
Alaw/µlaw	✔	✔	✔	✔		✔
AMR (3GPP)	✔	✔	✔	✔		✔
MIDI [3]	✔	?	✔	✘		✔
LPCM	✔	✔	✔	✔		✔
ADPCM	✔	✔	✔	✔		✔
QCELP	✔	✔	✔	✔		✔
DV Audio	✔	✔	✔	✔		✔
QDM2/QDMC (QuickTime)	✔	✔	✔	?		✔
MACE	✔	✔	✔	✔		✔

Native playback supported by VLC 1.0.3 and later. Previous versions could use Windows DMO codecs on 32-bit x86 platforms and allow WMV-3/WMA-3 decoding. This feature was never tested on Intel-based Macs.
Sipr codec playback is not supported.
Requires a .sf2 soundfont, see our wiki.

✔ = Yes
◒ = Partial
✘ = No
? = Untested

Figura 63 - VLC, formatos de áudio[53]

Audio/Vídeo Outputs

Video Outputs	Native	Direct3D DirectX GDI	OpenGL	✔	✔	✔
	X11	-	?	✔	-	✔
	XVideo	-	-	✔	-	✔
	SDL	✔	✘	✔	?	✔
	FrameBuffer	-	-	✔	-	-
	ASCII Art	✔	✔	✔	?	✔
Audio Outputs	Native	DirectX WaveOut	✔	OSS ALSA	✔	OSS
	S/PDIF	DirectX WaveOut	✔	OSS ALSA	✘	?
	Multi-channel	DirectX WaveOut	✔	OSS ALSA	✘	?
	PulseAudio	-	-	✔	-	?
	PortAudio	✔	?	?	-	?
	JACK	-	?	✔	-	?

✔ = Yes
◒ = Partial
✘ = No
? = Untested

Figura 64 - VLC, A/V Outpus[53]

Diversos

SAP/SDP announces	✔	✔	✔	✘		✔
Bonjour protocol	✘	✔	✔	?		?
Mozilla/Firefox plugin	✔	✔	✔	✘		✔
ActiveX plugin	✔	-	-	-		-
SVCD Menus	◒	✘	◒	✘		◒
Localization	✔	✔	✔	✔		✔
CD-Text [1]	✔	✘	✔	✘		◒
CDDB CD info	✔	✔	✔	✘		◒
IGMPv3 [2]	✔	✘	✔	✘		✔
IPv6 [2]	✔	✔	✔	✘		✔
MLDv2 [2]	✔	✘	✔	✘		✔
CPU acceleration [3]	✔	✔	✔	✔		✔

CD-Text information provided via libcdio. This service is available on all platforms supported by the library. Depending on the operating system's support.
Supported CPU extensions are MMX, MMXEXT, SSE, SSE2 and 3D Now! on x86 processors, and AltiVec on G4/G5 processors.

✔ = Yes
◒ = Partial
✘ = No
? = Untested

Figura 65 - VLC, diversos[53]

Anexo III – Guião de Tarefas

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA
Mestrado em Engenharia Informática – Computação Móvel
Dissertação
Ano letivo 2012/2013

Guião de tarefas - *MSSIPTV*

Contexto

A dissertação IMSIPTV tem como objetivo proporcionar às populações isoladas, uma aplicação onde possam ter as vantagens de aceder a conteúdos televisivos, serviços de comunicação coletivos.

Este teste tem como função avaliar a interface da aplicação desenvolvida e as suas funcionalidades.

Tarefas

1. Mude de canal
2. Pré-visualize um canal à sua escolha
3. Navegue no Guia de Programação
4. Aceda aos seus contatos
5. Escolha um contato e escreva uma mensagem
6. Envie a mensagem
7. Tire o som do vídeo
8. Desligue a aplicação

Anexo IV – Questionário de Testes

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA
Mestrado em Engenharia Informática – Computação Móvel
Dissertação
Ano letivo 2012/2013

Registo de Avaliação de Utilização
Data e Hora:
Local:

1. **Dados do utilizador**
   Sexo: **Idade:**
   Utilizador de Tecnologia: Não ☐ Raramente ☐ Frequente ☐
2. **Tarefas**

#	Tarefa	Tempo	Erros	Sucesso	Observações
1	Mude de canal				
2	Pré-visualize um canal à sua escolha				
3	Navegue no Guia de Programação				
4	Aceda aos seus contatos				
5	Escolha um contato e escreva uma mensagem				
6	Envie a mensagem				
7	Tire o som do vídeo				
8	Desligue a aplicação				