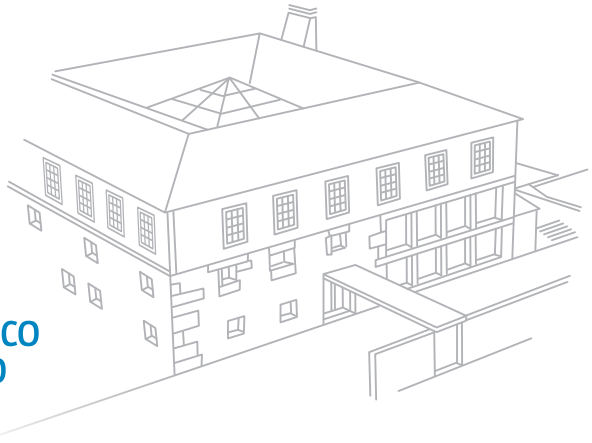

ESTGF | POLITÉCNICO DO PORTO

ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

MCRemote - Controlo de Multimédia Centers por dispositivos móveis

DESIGNAÇÃO DO MESTRADO: Mestrado em Engenharia Informática

AUTOR: Hélder José Magalhães Ribeiro

ORIENTADOR(ES): Prof. Doutor António Alberto Santos Pinto

ANO: 2011

Agradecimentos

Queria deixar o meu agradecimento à minha família, namorada e aos amigos que me apoiaram e me deram forças durante esta fase importante da minha vida académica.

Ao Professor Doutor António Pinto pela competência com que orientou esta minha tese e o tempo que generosamente me dedicou transmitindo-me os melhores e mais úteis conhecimentos, com paciência, lucidez e confiança, muito obrigado.

Agradeço também a todos os docentes da ESTGF que participaram na minha formação ao longo do mestrado, transmitindo-me todos os conhecimentos e competências para poder ter um bom desempenho ao longo da minha carreira profissional.

Antes de terminar, quero também deixar presente o meu obrigado a outras pessoas que indirectamente, com o seu trabalho, me proporcionaram um estado de espírito e dinamismo necessário para realizar esta tese de mestrado.

Resumo

O uso de equipamentos designados de *Media Centers* é já muito comum, pois permitem armazenar centralmente todos os conteúdos multimédia (fotos, vídeos e músicas), bem como reproduzi-los remotamente. Actualmente, a interacção dos utilizadores com os *Media Centers* é realizada através de periféricos dedicados e de proximidade. Tais periféricos, por serem dedicados, acarretam algumas limitações de funcionalidade e de interoperabilidade. O MCRemote visa colmatar tais limitações recorrendo, para tal, a dispositivos móveis que o utilizador tradicionalmente já possui (*smartphones*).

Palavras Chave: Interoperabilidade, *Media Centers*, *smartphones*.

Abstract

The use of equipment designed for Media Centers is already very common, because they allow centrally store all multimedia content (photos, videos and music) and play them remotely. Currently, the user interaction with the Media Centers is accomplished through dedicated devices and proximity. Such devices, because they are dedicated, result in some limitations of functionality and interoperability. The MCRemote aims to remedy these limitations using for this purpose, the mobile devices that the user already has traditionally (smartphones).

Keywords: Interoperability, *Media Centers*, *smartphones*.

Índice

Lista de Tabelas

Lista de Figuras

Lista de Blocos de Código

Lista de Acrónimos

API *Application Programming Interface*

DRM *Digital Rights Management*

EPG *Electronic Program Guides*

HD *High Definition*

HTML *HyperText Markup Language*

HTPC *Home Theater* PC

HTPCs *Home Theater* PCs

HTTP *Hypertext Transfer Protocol*

MC *Media Center*

MCs *Media Centers*

PC *Personal Computer*

PCs *Personal Computers*

TCP/IP *Transmission Control Protocol/Internet Protocol*

UPnP *Universal Plug and Play*

Capítulo 1

Introdução

Media Centers (MCs) são aplicações informáticas que armazenam centralmente conteúdos multimédia, permitindo ao seu utilizador desfrutar dos mesmos de forma extremamente prática. Este tipo de aplicações, são executadas num computador pessoal, sendo comandadas tipicamente por periféricos dedicados e de proximidade. Alguns dos MCs actuais podem ser ainda controlados remotamente.

Recentemente, têm surgido soluções cujo propósito é o controlo de MCs remotamente, utilizando para o efeito *smartphones*. As desvantagens dos periféricos dedicados e de proximidade são assim ultrapassadas. A maioria dessas aplicações apenas conseguem controlar um tipo de MC. Tal obriga o utilizador a utilizar diversas soluções, caso pretenda interagir com mais do que um tipo de MC. Estas soluções, não estão disponíveis para todos os sistemas operativos móveis actualmente presentes nos *smartphones*.

O cenário de referência adoptado para este trabalho, consiste num cenário de utilização doméstica, em que o utilizador dispõem de um *Media Center* (MC) por televisão instalados em *Home Theater* PCs (HTPCs), conseguindo assim com que todas as acções efectuadas em cada MC sejam visualizadas no respectivo televisor. Os HTPCs estão ligados à rede local sem fios, e o utilizador é portador de um *smartphone*.

O uso de periféricos dedicados ou de proximidade levou ao surgimento de aplicações de controlo remoto de MCs. No entanto, surgiram novos problemas, nomeadamente, a inexistência de um programa de controlo remoto multiplataforma, capaz de interagir com mais do que um tipo de MC, sendo assim necessário utilizar diversas soluções para controlar cada MC que se possua. A solução proposta, visa colmatar estas insuficiências adoptando uma arquitectura cliente/servidor. A adaptação desta arquitectura permite a evolução da solução proposta para interagir com um leque de MCs disponíveis livremente na internet. Na solução proposta, a aplicação cliente está embutida no dispositivo móvel, que por sua vez envia e recebe comandos para a aplicação servidor, e esta invoca os comandos junto do MC, para o mesmo executar a acção pretendida pelo utilizador. Para efectuar a comunicação entre as aplicações cliente e servidor, utiliza a rede local sem fios.

Este documento, encontra-se organizado em capítulos. O capítulo 2 explica no que consistem os MCs, quais os principais MCs de código aberto disponíveis *online*, suas funcionalidades, e em que moldes funciona a comunicação com este tipo de aplicações. O capítulo 3, descreve algumas soluções actualmente disponíveis para interagir com os MCs via *smartphones*, bem como as suas principais características e funcionalidades. O capítulo 4, descreve os objectivos gerais da proposta, um conjunto de requisitos funcionais que deverão ser assegurados pelo trabalho desenvolvido, descrição geral do MCRemote, e uma explicação detalhada da aplicação servidor e das duas aplicações cliente desenvolvidas até ao momento. No capítulo 5, são apresentados os resultados obtidos com o MCRemote, aspecto gráfico das aplicações, e demonstração da comunicação entre as aplicações que compõem o MCRemote e os MCs. No capítulo 6, encontra-se um resumo do documento.

Capítulo 2

Media Centers

MCs são aplicações informáticas que podem ser executados num computador pessoal ou em computadores específicos, denominados de HTPCs. Os HTPCs permitem a ligação à TV, tendo normalmente grande poder de processamento e memória para lidar com conteúdos em alta resolução, permitindo ao utilizador interagir com os mesmos utilizando um comando [1].

Estes equipamentos, permitem armazenar centralmente todos os conteúdos multimédia (fotos, vídeos e músicas), bem como reproduzi-los. A interacção entre utilizadores e este tipo de equipamentos pode ser efectuada de diversas formas. Em particular, por recurso a periféricos dedicados e de proximidade como comandos remotos (infravermelhos), teclados ou ratos.

Os MCs permitem também navegação web, jogos, visualização de filmes, assistir e gravar emissões televisivas, entre outras funcionalidades. Tudo isto numa só aplicação, permitindo ao utilizador desfrutar de todos os seus conteúdos multimédia de forma prática.

Existem diversos MCs de código aberto disponíveis *online*. Spagnolo, em [2], enumerou, comparou e classificou os vários MCs de código livre existentes. De seguida serão elencados os mais reconhecidos:

- XBMC
- Freevo
- MythTV
- Moovida
- Boxee
- Plex
- Entertainer
- My Media System

- MediaPortal
- Neuros OSD

O XBMC surgiu em 2002 com o nome *Xbox Media Player*, passando mais tarde a ser conhecido como XBMC. A associação à consola de jogos Xbox advém da possibilidade do XBMC poder ser instalado neste tipo de consolas. Embora o XBMC não seja suportado pela Microsoft ou qualquer outro fornecedor, é um MC muito amadurecido e é multi-plataforma (Xbox, Linux, Mac, Windows e Apple TV). O ambiente gráfico do XBMC contém um menu onde é possível adicionar conteúdos à biblioteca multimédia do programa, aceder a informações úteis (condições meteorológicas), bem como configurar o XBMC: definir o aspecto do MC, configurações de rede e definições do hardware que controla o áudio e o vídeo.

O Freevo surgiu em 2002, sendo desenvolvido em Python. Tal permite que os próprios utilizadores desenvolvam novos plug-ins. É multi-plataforma (Linux, Windows e Mac) e tem suporte multi-idioma. A nível gráfico, Freevo apresenta opções para se visionar emissões televisivas, respectiva programação e efectuar gravações, opções para acrescentar/navegar pelos conteúdos multimédia (vídeos, fotos e músicas), uma secção para visualização de notícias, e uma secção de configuração do Freevo.

O MythTV tem uma estrutura modular, de modo a que o que não pode ser encontrado em módulos padrão provavelmente está disponível em plug-ins não oficiais. A sua data de lançamento é de 2002. MythTV está disponível para Linux e tem suporte multi-idioma. O ambiente gráfico apresenta diversas opções, tais como visualização de emissões televisivas, consulta/reprodução da biblioteca de multimédia presente no programa, gestão das gravações previamente efectuadas, reprodução de CD/DVD, e uma secção para definições do MythTV e algumas utilidades, como por exemplo a consulta da meteorologia.

O Moovida surgiu em 2006 e é propriedade da empresa Fluendo. O ambiente gráfico do MC Moovida (anteriormente conhecido como Elisa) está dividido em várias secções. Estas secções permitem, entre outras funções, visualizar dispositivos removíveis, navegar pelas pastas do *Personal Computer* (PC) e pelos *Personal Computers* (PCs) que estão na mesma rede. Permite ainda a adição de conteúdos para a biblioteca de multimédia do programa. É multi-plataforma e tem suporte multi-idioma.

O Boxee foi lançado em 2008 e possibilita a visualização, classificação de conteúdos, bem como a interacção com redes sociais. O Boxee é baseado no XBMC e corre nas plataformas Mac, Linux e Windows. O seu interface está vocacionado para a interacção com as redes sociais, onde apresenta uma secção com os conteúdos multimédia mais populares entre os amigos presentes na nossa rede, e uma secção para controlo/edição dos mesmos. Apresenta ainda opções para controlo/reprodução de conteúdos multimédia, emissão e gravação de emissões televisivas, e opções para configuração do Boxee.

O Plex, tal como o Boxee, é baseado no XBMC. A primeira versão foi lançada em 2005, corre nas plataformas Mac, Windows e Linux e tem suporte multi-idioma. A última versão do Plex corre em ambiente web, onde apresenta opções para definição da localização dos conteúdos multimédia e adição dos mesmos. Uma das funcionalidades fornecidas, é a obtenção automática de metadados via internet, complementando assim com informações adicionais os conteúdos presentes no Plex.

O Entertainer foi lançado em 2007 e é baseado no *Windows Media Center*, mas corre na plataforma Linux. Uma das suas funcionalidades é o suporte multi-idioma. Foi desenvolvido em Python e para a reprodução de multimédia usa a *framework* multimédia *gstreamer*. A interface do Entertainer utiliza a biblioteca Clutter, o que permite o desenvolvimento de interfaces animadas em OpenGL.

My Media System foi lançado em 2002 e evoluiu do antigo MPeg Menu System. Corre na plataforma Linux e tem suporte multi-idioma. Em termos gráficos, apresenta opções para adição e visualização de conteúdos multimédia (músicas, vídeos e fotos), opções para adicionar e usufruir de jogos, visualização de emissões televisivas, consulta das condições meteorológicas, e permite controlar as definições do My Media System.

O MediaPortal surgiu em 2004, tendo como base a framework .NET da Microsoft. Corre em Windows, tem suporte multi-idioma e em 2006 lançou uma versão com suporte para um servidor ”TV”, o que possibilita aos utilizadores o uso de múltiplas interfaces para visualização e gravação de TV, transmitido a partir de um ou mais servidores TV. Disponibiliza ainda opção para audição de músicas e rádio, consulta da meteorologia, visualização de notícias e jogar jogos.

O Neuros OSD é propriedade da Neuros Technology. Consiste numa *set-top box*, que tem por base um firmware *Open Source* que corre em Linux, tem suporte multi-idioma e surgiu em 2003. Permite visualização de TV via internet, reprodução de conteúdos multimédia, incluindo conteúdo *High Definition* (HD), quer os mesmos estejam presentes localmente ou na internet, e visualização de notícias. No futuro, será também possível visualização de imagens.

A Tabela 2.1 compara as várias soluções existentes, focando as plataformas em que estão disponíveis e acessórios ou funcionalidades suportadas. A título de exemplo, o MC MythTV, corre em Linux, suporta controlo remoto, suporta parcialmente controlo por giroscópio remoto[1], suporta também teclado Wireless Microsoft multimédia[2]. Tem também suporte para dispositivos universais *Plug and Play*, i.e. *media streamers*, *pendrives*, memórias sd, entre outros. Não disponibiliza suporte para *Sony PSP*, i.e. não consegue ler/carregar conteúdo multimédia presente numa *Sony PSP*. Tem suporte para detectar, ler e carregar conteúdo multimédia presente num iPod, sendo necessário para tal, a instalação de um *add-on*. Não suporta *Digital*

[1] Consiste num comando remoto que contem um giroscópio, permitindo uma melhor navegação pelas opções desejadas.

[2] Consiste num teclado sem fios, que fornece algumas teclas para controlo multimédia.

Caract./Media Center	Boxee	Moovida	Entertainer	Freevo	Media Portal	My Media System	MythTV	Neuros OSD	Plex	XBMC
Corre em:										
Linux	√	√	√	√	×	√	√	×	√	√
Windows	√	√	×	√	√	×	×	×	√	√
Mac	√	×	×	√	×	×	×	×	√	√
Apple TV	×	×	×	×	×	×	×	×	×	√
set-top box	×	×	×	×	×	×	×	√	×	×
Xbox	×	×	×	×	×	×	×	×	×	√
Acessórios/Suporte:										
Sup. para controlo remoto	√	√	√	√	√	√	√	√	√	√
Sup. para giroscópio remoto	×	sup. parc.	×	×	√	×	sup. parc.	×	×	sup. parc.
Tcl wireless Ms multimédia	×	√	×	√	√	×	√	×	×	√
Plug and Play	√	√	×	√	√	×	√	√	√	√
Sony PSP	×	√	×	×	×	×	×	√	×	√
iPod	×	×	×	plugin	×	sup. parc.	add-on	√	×	×
DRM	×	×	×	×	sup. parc.	×	×	×	×	×

Tabela 2.1: Quadro comparativo dos MCs mais reconhecidos.

Rights Management (DRM), que consiste numa tecnologia que inibe o uso de conteúdo digital, quando tal não é pretendido pelo autor do conteúdo.

2.1 Comunicação com os *Media Centers*

Alguns MCs têm interfaces que possibilitam a comunicação com os mesmos por parte de aplicações externas. De seguida será demonstrado como funcionam essas interfaces.

2.1.1 XBMC

O XBMC desenvolveu uma *Application Programming Interface* (API) que opera sobre o protocolo de transferência *HyperText*, denominada *Web Interface* [3].

A API pode ser invocada via browser, e este por sua vez mostra os resultados aos comandos enviados. Para a comunicação ser possível é necessário configurar o XBMC para aceitar comunicações via servidor web, assumindo por defeito que as mesmas serão efectuadas pela porta 80.

Os comandos a enviar à API têm o seguinte formato:

```
http://xbox/xbmcCmds/xbmcHttp?command=*
```

Em * é colocada a instrução desejada. Pode também ser necessário trocar o texto `xbox` pelo IP da máquina onde o XBMC está a ser executado. Assumindo que o XBMC está a ser executado numa máquina com o IP 192.168.1.3, e que se deseja obter o volume actual, usaríamos o endereço `http://192.168.1.3/xbmcCmds/xbmcHttp?command=GetVolume()`.

De seguida estão representados alguns dos comandos reconhecidos pela API:

- Comandos que devolvem informação:
  - **GetShares(type; [option]):** Devolve a localização dos conteúdos. Em *type* pode estar: *music, video, pictures, files*.
  - **GetDirectory(directory;[mask];[option]; [lineStart; [numLines]]):** Devolve o conteúdo de uma determinada pasta. Permite também filtar por tipo de ficheiro e o número de items que se deseja receber.
  - **GetVolume:** Devolve o volume actual em forma de percentagem.
- Comandos que modificam definições:
  - **AddToPlayList(media;[playlist];[mask];[recursive]):** Adiciona ficheiros ou pastas à lista de reprodução actual.
  - **ClearPlayList([playlist]):** Limpa a lista de reprodução actual.
  - **ClearSlideshow:** Limpa a lista de reprodução das imagens.
  - **SetCurrentPlaylist(playlist):** Adiciona uma lista de reprodução à lista de reprodução.
  - **SetVolume:** Determina o volume em forma de percentagem.
- Comandos que geram acções:
  - **Action(1):** Executa a acção de ir para a esquerda.
  - **Action(2):** Executa a acção de ir para a direita.
  - **Action(3):** Executa a acção de subir.
  - **Action(4):** Executa a acção de descer.
  - **Action(7):** Executa a acção de ir para o próximo menu.
  - **Action(9):** Executa a acção de voltar para o menu anterior.
  - **Action(77):** Executa a acção de avançar na reprodução.
  - **Action(78):** Executa a acção de retroceder na reprodução.
  - **Action(79):** Executa a acção de reprodução.
  - **Action(100):** Executa a acção de OK.
  - **Pause:** Pausa o ficheiro que está a ser reproduzido.
  - **PlayFile(filename;[playlist]):** Reproduz um determinado ficheiro presente numa lista de reprodução.
  - **PlayNext:** Reproduz/mostra o próximo ficheiro na lista de reprodução.
  - **PlayPrev:** Reproduz/mostra o ficheiro anterior na lista de reprodução.
  - **PlaySlideshow([directory];[recursive]):** Reproduz a lista de reprodução de imagens.
  - **ShowPicture(filename):** Mostra uma determinada imagem.
  - **Shutdown:** Desliga o XBMC.
  - **Stop:** Pára a currente reprodução.

- Comandos diversos:
  - **WebServerStatus([status]):** Determina/devolve o estado do servidor web.

Em resposta aos comandos invocados, a API devolve quatro tipos de respostas em formato *HyperText Markup Language* (HTML), sendo elas:

- OK: significa que o comando foi executado com sucesso;
- Um valor simples: como nos casos onde é pedido o valor do volume;
- Uma lista de valores com o formato: `nome:valor`;
- Erro: mensagem de erro no seguinte formato: `Error:motivo`.

2.1.2 Boxee

O Boxee, que foi desenvolvido tendo como base o XBMC, utiliza a mesma API como interface de comunicação com aplicações externas, apresentando algumas diferenças [4]. Para a comunicação ser possível é necessário configurar o Boxee para aceitar comunicações via servidor web, assumindo por omissão que as mesmas são efectuadas pela porta 8800.

Os comandos a enviar à API têm o seguinte formato:

```
http://xbox:8800/xbmcCmds/xbmcHttp?command=*
```

Em * é colocada a instrução desejada. Pode também ser necessário trocar o texto `xbox` pelo IP da máquina onde o Boxee está a ser executado. Assumindo que o Boxee está a ser executado numa máquina com o IP 192.168.1.3, e que se deseja determinar a percentagem do volume para 80%, usaríamos o endereço `http://192.168.1.3:8800/xbmcCmds/xbmcHttp?command=SetVolume(80)`.

De seguida estão representados os comandos reconhecidos pela API do Boxee e que diferem do XBMC:

- Comandos que geram acções:
  - **SendKey(270):** Executa a acção de subir.
  - **SendKey(271):** Executa a acção de descer.
  - **SendKey(272):** Executa a acção de ir para a esquerda.
  - **SendKey(273):** Executa a acção de ir para a direita.

2.1.3 Plex

Tal como o Boxee, o Plex também foi desenvolvido tendo como base o XBMC, e também utiliza uma API remota para comunicar com aplicações externas, embora para tal ser possível é imperativo usar a versão 0.9 do Plex ou superior. A API remota do Plex ainda se encontra numa

fase de desenvolvimento/melhoramento, o que origina que em algumas situações aconteçam problemas de comunicação [5].

Os comandos a enviar à API têm o seguinte formato:

`http://server:32400/system/players/player/controller/command`.

Explicação do comando:

- ***server*:** o endereço IP ou nome do host do computador onde se encontra o servidor com os conteúdos multimédia;
- ***player*:** o nome do computador onde o Plex está a ser executado e para onde se deseja enviar o comando;
- ***controller*:** controlador que se deseja enviar;
- ***command*:** o comando que se deseja enviar.

Exemplo de um comando:

`http://nomePCservidor:32400/system/players/nomePCPlex/navigation/moveDown`.

A API do Plex disponibiliza os seguintes controladores e comandos:

- Controlador de navegação:
  - moveUp;
  - moveDown;
  - moveLeft;
  - moveRight;
  - pageUp;
  - pageDown;
  - nextLetter;
  - previousLetter;
  - select;
  - back;
  - contextMenu;
  - toggleOSD;
- Controlador de reprodução:
  - play;
  - pause;
  - stop;
  - rewind;

  - fastForward;
  - stepForward;
  - bigStepForward;
  - stepBack;
  - bigStepBack;
  - skipNext;
  - skipPrevious;

- Controlador da aplicação:
  - playFile;
  - playMedia (argumentos possíveis: path, key, userAgent (optional), httpCookies (optional), viewOffset (optional));
  - screenshot (argumentos possíveis: width=480&height=270&quality=75);
  - sendString (argumentos possíveis: text=xyz);
  - sendKey (argumentos possíveis: code=01);
  - sendVirtualKey (argumentos possíveis: code=01).

2.1.4 Resumo

As diferenças entre as API's do XBMC e Boxee centram-se nas instruções para navegar pelas opções dos MCs. No XBMC, as instruções a enviar para este efeito têm o formato `Action(*)`, onde em * é colocado o id que identifica as opções para subir(3), descer(4), ir para a esquerda(1) e ir para a direita(2). No Boxee, estas instruções passam a ter o formato `SendKey(*)`, onde em * é colocada a chave que corresponde às teclas direccionais do teclado. A resposta aos comandos invocados também apresenta algumas diferenças entre estas API's, nomeadamente nas respostas que contêm uma lista de valores. Uma das diferenças acontece quando invocamos a localização dos conteúdos (fotos, vídeos e músicas), uma vez que o Boxee possibilita a definição de vários locais de favoritos.

Na API do Plex, a estrutura dos comandos a invocar apresenta diferenças significativas comparativamente com as API's do XBMC e do Boxee. Nestas, a estrutura dos comandos é constante, apenas mudando a instrução final, ou seja, a operação que se pretende que o MC realize. No Plex, é necessário indicar o tipo de operação que se pretende (navegação, reprodução ou aplicação), e qual a operação desejada dentro de cada tipo. Para além disso, a API do Plex disponibiliza menos operações que as demais.

Capítulo 3

Trabalho Relacionado

A proliferação de dispositivos tipo *smartphones* levou ao aparecimento de várias aplicações para controlar remotamente os MCs. A generalidade das soluções disponíveis apenas interage com um MC, obrigando assim, à utilização de diversas soluções com vários tipos de configurações e formas de trabalhar, para ser possível controlar remotamente os vários tipos de MCs disponíveis.

Algumas soluções interagem directamente com o MC, enquanto outras utilizam uma aplicação intermédia que faz de ponte entre a aplicação embutida no *smarthpone* e o MC. Todas elas assumem estar em simultâneo na mesma rede local sem fios, facilitando a comunicação entre as mesmas.

As soluções encontradas não são multi-plataforma, i.e. não estão disponíveis para vários sistemas operativos (iOS, Android, Windows Phone 7, BlackBerry, Symbian), sendo que apenas um suporta mais do que um sistema operativo móvel. A maior parte das soluções disponíveis, foram desenvolvidas para o sistema operativo Android.

De seguida são elencadas algumas das soluções para *smartphones* com o intuito de controlar remotamente MCs, bem como as suas principais características e funcionalidades:

- Gmote
- Media Remote
- VLC Remote
- AndroMote Remote Control
- DirectTV Remote Free
- Remote Media Center
- My Remote
- WMC Remote

- Official XBMC Remote
- Boxee Wifi Remote

3.1 Gmote

Gmote [6] é um software gratuito constituído por duas aplicações, a cliente e a servidor, sendo necessário para correr o mesmo, satisfazer os seguintes requisitos:

- *Smartphone* com sistema operativo android e a aplicação Gmote instalada;
- Computador com conteúdo multimédia, Gmote Server e VLC media player instalados;
- Computador e *Smartphone* na mesma rede.

A aplicação servidor (Gmote Server) foi desenvolvida em Java, sendo assim multi plataforma, e é responsável pela comunicação entre a aplicação cliente e o VLC media player. Apresenta ainda algumas funcionalidades, tais como:

- definição de uma palavra passe para a comunicação entre a aplicação cliente/servidor;
- definição das pastas onde se encontram os conteúdos multimédia;
- visualização de informações úteis, como o IP do computador e os logs onde são armazenadas as definições da aplicação.

Nas Figuras 3.1 e 3.2 encontram-se exemplos destas funcionalidades.

Figura 3.1: Gmote server - Definir password.

A aplicação cliente corre em *smartphones* com sistema operativo móvel Android. Começa por tentar encontrar o aplicação Gmote server e caso não o consiga de uma forma automática, permite a inserção manual do IP onde a aplicação servidor se encontra instalada. Após ser estabelecida a comunicação com sucesso, permite navegar pelas pastas onde se encontra o conteúdo multimédia e a reprodução do mesmo. Uma das particularidades do Gmote é a possibilidade de reproduzir os conteúdos no PC ou no próprio *smartphone*, embora esta opção seja ainda beta e apresente alguns problemas, nomeadamente na reprodução de vídeo. Caso se opte pela reprodução no *smartphone*, o conteúdo multimédia é carregado para o mesmo,

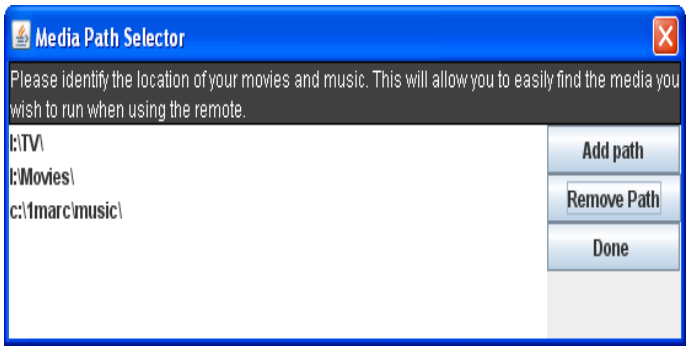

Figura 3.2: Gmote server - Definir local do conteúdo multimédia.

tendo desde logo as limitações da velocidade da rede e do espaço disponível de memória do *smartphone*.

Nas Figuras 3.3 e 3.4 encontram-se exemplos destas funcionalidades.

Figura 3.3: Gmote cliente - Indicar IP servidor.

O Gmote apresenta um interface apelativo e é bastante intuitivo, tendo como desvantagem o facto de só comunicar com o VLC media player.

3.2 Media Remote

Media Remote [7] é uma aplicação projectada para permitir controlar remotamente alguns produtos mais recentes da Sony com receptores Blu-ray/TV/AV.

Alguns dos produtos da Sony compatíveis com o Media Remote:

Figura 3.4: Gmote cliente - Reprodução de uma música.

- Blu-ray Disc Player BDP-S370, BDP-S470, BDP-S570, BDP-S770, BDP-S1700, BDP-BX37, BDP-BX57, BDP-S380, BDP-S480, BDP-S580, BDP-S780, BDP -BX38, BDP-BX58;
- Blu-ray Home Theater System BDV-IZ1000W, BDV-HZ970W, BDV-E970W, BDV-E870, BDV-E770W, BDV-E670W, BDV-E570, BDV-E470, BDV-E370, BDV-T57, BDV-F7, BDV-F700, Rede Media Player SMP-N100, N200-SMP;
- AV Receiver STR-DN1020;
- TV BRAVIA KDL-HX920series;
- Sony Internet TV NSX-24GT1, NSX-32GT1, NSX-40GT1, NSX-46GT1, NSZ-GT1.

Algumas funcionalidades do Media Remote:

- Controlo remoto simples: permite navegar através do interface gráfico do produto Sony;
- Controlo remoto completo: permite efectuar todas as operações que são possíveis de serem realizadas com um controlo remoto tradicional;
- Teclado de software: teclado QWERTY completo que permite digitar os caracteres mais facilmente;
- Visualizar informações do filme Blu-ray: permite visualizar os detalhes do filme que está a ser reproduzido, como informações dos actores, realizadores, entre outros.

Para o Media Remote funcionar é imperativo que o *smartphone* e o produto Sony estejam contidos na mesma rede Wifi. Na Figura 3.5 encontra-se um exemplo do aspecto do Media Remote.

Figura 3.5: Media Remote.

3.3 VLC Remote

VLC Remote [8] funciona como um controlo remoto para o VLC media player. Existem versões disponíveis para iPhone, iPad, Web OS e Android.

Algumas funcionalidades do VLC remote:

- Controlo da navegação pelas pastas do computador e selecção dos conteúdos para reprodução;
- Botões para controlar volume, stop, play, pausa, próxima faixa e faixa anterior;
- Controlo completo da reprodução via DVD;
- Opção para activar/desactivar *fullscreen*;
- Controlo das legendas;
- Comandos para controlar tonalidade do som;
- Visualização das *playlists* e reprodução das mesmas;
- Acesso a unidades externas;
- Controlo e personalização de *skins*;
- Procura automaticamente os VLC media players na rede local.

Na Figura 3.6 encontra-se um exemplo do aspecto do VLC Remote.

Figura 3.6: VLC Remote.

3.4 AndroMote Remote Control

AndroMote [9] é um software de controlo remoto disponível para Android. Procura por *media servers* e por *media renderers*[1] na rede Wifi que compreendem o protocolo *Universal Plug and Play* (UPnP) [10]. Tem funcionalidades para criar listas de reprodução de músicas no *smartphone* e permite reproduzir as mesmas num *media player* ou no próprio *smartphone*. Para o AndroMote funcionar, é necessário ter um *media server* na rede Wifi e é recomendado ter também um *media player* para ser controlado remotamente.

Lista de alguns *media servers* que funcionam com o AndroMote:

- Windows Media Player desenvolvido pela Microsoft;
- XBMC;
- Fritz!Musikbox desenvolvido pela AVM;
- ON2Share desenvolvido pela On2Trade;
- J. River MEDIA CENTER desenvolvido pela J. River, Inc.;
- MythTV;
- EMC LifeLine Media Server desenvolvido pela EMC Corporation.

Lista de alguns *media renderers* que funcionam com o AndroMote:

[1] Dispositivos de entretenimento conectados à rede local, com capacidade para recuperar conteúdos multimédia provenientes de um PC ou de um *media server*, conseguindo reproduzir esse conteúdo num *Home Theater* PC (HTPC) ou numa televisão.

- On2Share desenvolvido pela On2Trade;
- XBMC;
- MusicPal desenvolvido pela Freecom;
- SoundBridge desenvolvido pela Roku;
- HD TV Live desenvolvido pela WD;
- TX-NR 3007 desenvolvido pela Onkyo;
- SLA5520 desenvolvido pela Philips.

Na Figura 3.7 encontra-se um exemplo do aspecto do AndroMote.

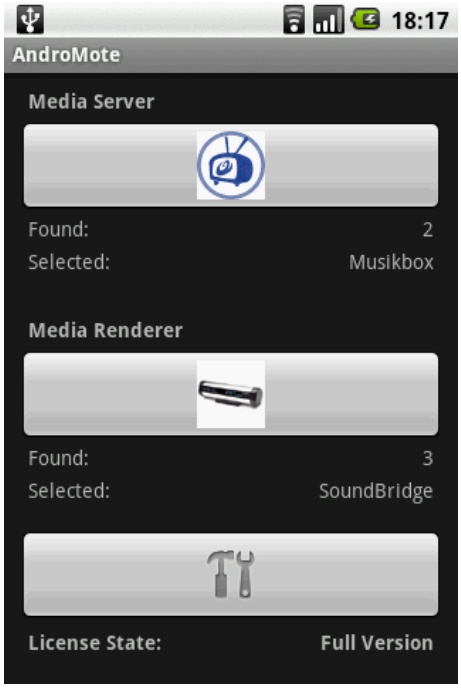

Figura 3.7: AndroMote.

3.5 DirectTV Remote Free

DirectTV Remote Free [11] é um software para controlar remotamente um receptor DIRECTV através de um *smartphone*. Para conseguir comunicar com um receptor DIRECTV é imperativo que o mesmo esteja conectado à rede Wifi. Apenas controla o receptor DIRECTTV, não conseguindo controlar todas as funções da TV.

Características do DirectTV Remote Free:

- Abrange todas as funcionalidades de um controlo remoto;
- Visualização das gravações e reprodução das mesmas;

- Adição de canais à lista de canais favoritos e visualização dos mesmos;
- Notificação na barra de *status* do que está a ser reproduzido;
- Descobre automaticamente os receptores DIRECTV.

Requisitos do DirectTV Remote Free:

- DIRECTV PLUS modelo HR20 HD, HR21, HR22, HR23, HR24 ou H21, H23, H24 conectado à rede Wifi;
- Receptor DIRECTV devidamente configurado para permitir dispositivos externos;
- Acesso Wifi para o *smartphone* com Android.

Na Figura 3.8 encontra-se um exemplo do aspecto do DirectTV Remote Free.

Figura 3.8: DirectTV Remote Free.

3.6 Remote Media Center

Remote Media Center [12] é um software cliente desenvolvido para Android, com o intuito de controlar e visualizar remotamente o Windows 7 Media Center, bem como outras versões do Windows (XP, Vista: funcionalidades limitadas), através do servidor remoto em execução no Windows. A comunicação entre a aplicação cliente e a aplicação servidor é realizada via Wifi.

Funcionalidades do Remote Media Center:

- Pesquisa na Grid de *Electronic Program Guides* (EPG);

- Procura e permite programar gravações;
- Procura por canais e programas;
- Permite reprodução de vídeos, músicas e fotos;
- Permite visualização de TV previamente gravada;
- Abrange todas as funcionalidades de um controlo remoto.

Na Figura 3.9 encontra-se um exemplo do aspecto do Remote Media Center.

Figura 3.9: Remote Media Center.

3.7 My Remote

My Remote [13] permite controlar remotamente o MC *Windows Media Center*, tendo como funcionalidades todas as operações possíveis de serem realizadas com um comando remoto, bem como entrada de voz para controlar o MC.

Na Figura 3.10 encontra-se um exemplo do aspecto do My Remote.

3.8 WMC Remote

WMC Remote [14] é um software de controlo remoto para controlar o *Windows Media Center*, sendo necessário a instalação do *Windows Remote Service*.

Figura 3.10: My Remote.

Funcionalidades do WMC Remote:

- Navegação pelo *Windows Media Center*;
- Comandos que permitem controlar a reprodução dos conteúdos;
- Opção para activar/desactivar *fullscreen*;
- Comandos que permitem navegar pelas pastas onde se encontram os conteúdos multimédia;
- Menus para consulta/visualização das opções Cinema, Música, Imagens, Rádio, TV, *TV-Guide* e Gravações.

Na Figura 3.11 encontra-se um exemplo do aspecto do WMC Remote.

3.9 Official XBMC Remote

Official XBMC Remote [15] fornece todos os recursos de controlo remoto para XBMC *Media Center*.

Funcionalidades gerais do Official XBMC Remote:

- Controlo do volume através dos botões de volume do *smartphone*;
- Acesso rápido aos items na lista via teclado;
- Consegue gerir várias instâncias do XBMC;

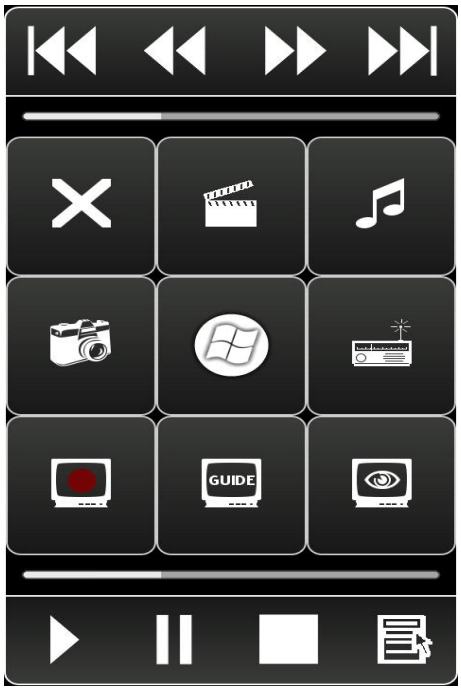

Figura 3.11: WMC Remote.

- Visualização e reprodução das *playlists*;
- Barra de notificação que mostra o que está a ser reproduzido no momento;
- Ao receber uma chamada, mostra quem está a ligar, e faz pausa na reprodução até a chamada terminar;
- Configuração que impede o bloqueio do *smartphone*;
- Permite download de *skins* nas versões mais recentes do XBMC;
- Nas livrarias de músicas:
  - Mostra a capa do álbum, caso a mesma esteja disponível;
  - Navegação por álbum, artista e género;
  - Lista todos os álbuns e músicas de um determinado artista;
  - Listagem da lista actual com as opções para ir para outra música ou remover uma música.
- Nas livrarias de vídeos:
  - Procura por título, actores e género;
  - Mostra todos os filmes de um determinado género ou de um determinado actor;
  - Mostra o cartaz do filme caso o mesmo esteja disponível.
- Nas livrarias de *TV Shows*:
  - Procurar por *show*, temporada, episódio ou actores e género;

– Reproduz episódios e mostra os respectivos detalhes.

Nas Figuras 3.12 e 3.13 encontram-se um exemplos do aspecto do Official XBMC Remote.

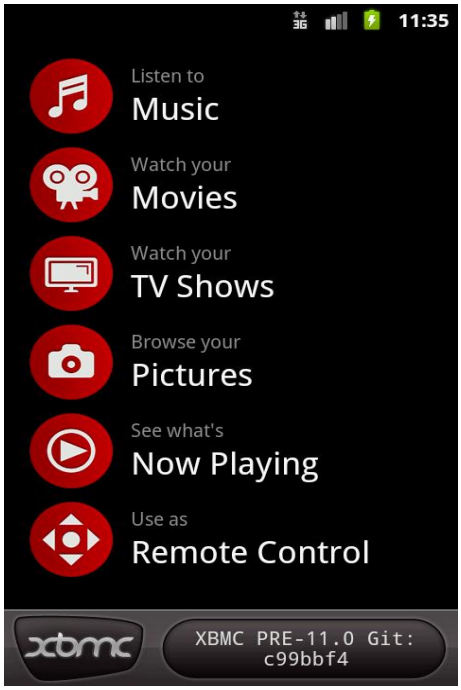

Figura 3.12: Official XBMC Remote - Menu principal.

Figura 3.13: Official XBMC Remote - Botões de reprodução e navegação.

3.10 Boxee Wifi Remote

Boxee Wifi Remote [16] foi desenvolvido para Android e permite controlar o MC Boxee. Esta aplicação fornece uma navegação pelas funcionalidades do Boxee através da rede Wifi. Para que a comunicação seja possível, é necessário activar a opção *webserver*.

Funcionalidades do Boxee Wifi Remote:

- Conexão automática com o Boxee;
- Reprodução dos conteúdos multimédia;
- Navegação pelas *play lists*;
- Teclado QUERTY para efectuar pesquisas.

Na Figura 3.14 encontra-se um exemplo do aspecto do Boxee Wifi Remote.

Figura 3.14: Boxee Wifi Remote.

3.11 Comparação das soluções disponíveis

Na Tabela 3.1 são comparadas as várias soluções existentes para controlar remotamente MCs via *smartphones*. A título de exemplo, o Remote Media Center suporta o MC Windows 7 Media Center, consegue detectar automaticamente o MC na rede, disponibiliza opções de navegação, sendo assim possível navegar pelas diversas opções/menus do Windows 7 Media Center. Contem opções para controlar a reprodução dos conteúdos multimédia, disponibiliza listagens com todas as músicas, videos e fotos presentes no MC, permitindo escolher qual o conteúdo que se deseja reproduzir. Oferece opções para controlar o volume, e não consegue

gerir mais do que um Windows 7 Media Center em simultâneo.

Software	Gmote	Media Remote	VLC Remote	AndroMote Remote Control	DirectTV Remote Free	Remote Media Center	My Remote	WMC Remote	Official XBMC Remote	Boxee Wifi Remote
MCs suportados	VLC media player	Produtos Sony com receptores Blu-ray/TV/AV	VLC media player	MythTV, XBMC, Windows Media Player	DIRECTTV	Windows 7 Media Center	Windows Media Center	Windows Media Center	XBMC	Boxee
Detecção automática do MC	√	√	√	√	√	√	√	√	√	√
Opções navegação	√	√	√	√	√	√	√	√	√	√
Opções Reprodução	√	√	√	√	√	√	√	√	√	√
Listagem de músicas	√	×	√	√	√	√	√	√	√	√
Listagem de vídeos	√	×	√	√	√	√	√	√	√	√
Listagem de imagens	√	×	√	√	√	√	√	√	√	√
Controlo de volume	√	√	√	√	√	√	√	√	√	√
Gere várias instâncias	×	×	×	×	×	×	×	×	√	×

Tabela 3.1: Comparação das soluções disponíveis para controlar remotamente MCs.

Capítulo 4

MC Remote

A interacção entre os utilizadores e os MCs pode ser efectuada de diversas formas. Em particular, por recurso a periféricos dedicados e de proximidade como comandos remotos (infravermelhos), teclados ou ratos. Tais periféricos, por serem dedicados, acarretam algumas limitações de funcionalidade e de interoperabilidade. A título de exemplo, dispositivos como ratos não permitem a execução de pesquisas por texto. Já os teclados, devido às suas dimensões, não são de transporte prático. Em regra, não são bidireccionais, i.e. só de entrada ou só de saída de dados. A presente proposta, de nome MCRemote, visa colmatar tais limitações recorrendo a dispositivos móveis, que o utilizador tradicionalmente já possui (*smartphones*), como mecanismos de interacção. O desafio encontrado consiste em permitir que a proposta de solução interaja com o grande leque de MCs disponíveis livremente na Internet. Tal implica, que a aplicação a executar nos dispositivos móveis, reconheça as instruções a serem enviadas ao MC.

O MCRemote foi pensado para ser posto em prática num ambiente familiar, nomeadamente numa sala de estar, contendo a mesma os seguintes elementos:

- Televisão/Plasma/LCD/LED;
- HTPC/PC provido com placa TV e com um MC instalado;
- Router wireless;
- *Smartphone*.

O HTPC tem de obrigatoriamente ter um MC *open source* e a aplicação servidor instaladas, bem como necessita de estar conectado com o televisor, através de uma placa TV contida no mesmo. Assim, todas as acções efectuadas no MC vão ser visualizadas em alta definição no televisor. O HTPC tem também de estar ”conectado”com o router wireless, utilizando para isso um cabo de rede ou uma ligação wireless, tornando possível a comunicação de outros dispositivos presentes na mesma rede com o mesmo. Para a interacção com o MC será utilizado um *smartphone*. Na Figura 4.1 encontra-se representado o cenário de referência adoptado neste trabalho.

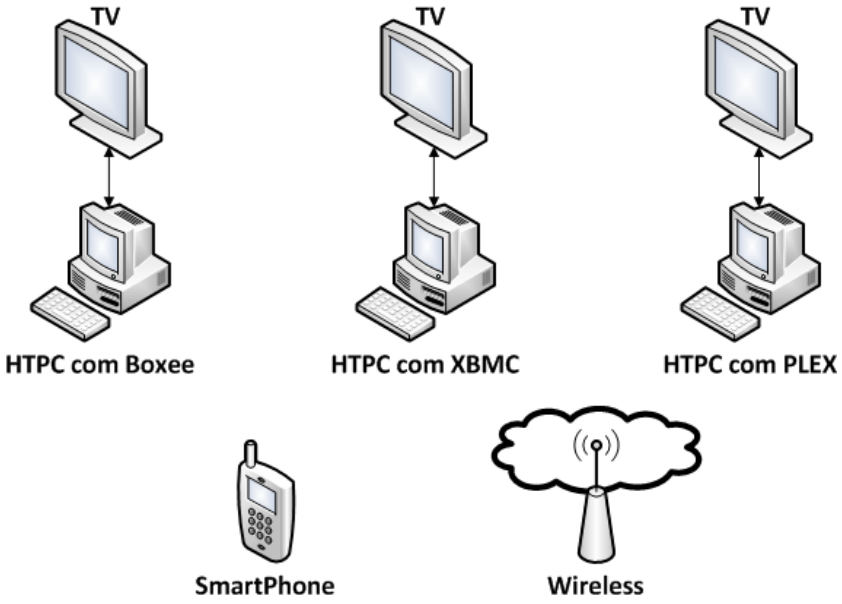

Figura 4.1: Cenário de referência.

4.1 Objectivos

O principal objectivo deste projecto, consiste em permitir a interacção entre um dispositivo móvel (*smartphones*) e um MC *Open Source*, atingindo assim, uma maior comodidade na utilização deste tipo de equipamento. Como segundo objectivo, pretende-se ultrapassar o problema da interoperabilidade, contornando assim o facto de os vários MCs analisados não utilizarem um interface de controlo padrão. Finalmente, pretende-se que a solução proposta não recorra a periféricos dedicados e de proximidade para interagir com o MC, ultrapassando as limitações associadas aos actuais comandos.

4.2 Requisitos

O projecto tem os seguintes requisitos funcionais:

- **Comunicação entre dispositivo móvel e MC:** a aplicação deverá ser capaz de estabelecer a comunicação entre o dispositivo móvel e o MC;
- **Navegação pelas diversas opções do MC:** a aplicação deverá ser capaz de navegar pelas diversas opções/menus disponíveis no MC;
- **Permitir que o MC consiga reproduzir os conteúdos nele incluídos:** a aplicação deverá ser capaz de enviar os comandos necessários para o MC reproduzir os conteúdos nele incluídos;
- **Aplicação de fácil utilização:** a aplicação deverá ser prática, funcional e ter um aspecto amigável para o utilizador, tornando-se assim acessível a todos os utilizadores independentemente dos seus conhecimentos a nível informático.

4.3 Descrição

Este projecto consiste numa aplicação do tipo cliente/servidor, onde a aplicação cliente está embutida no dispositivo móvel, que por sua vez, envia e recebe comandos para a aplicação servidor que está alojada no computador ou na set-top box. A aplicação servidor, é responsável por enviar as instruções ao MC. Na Figura 4.2 encontra-se um esquema da arquitectura da solução proposta.

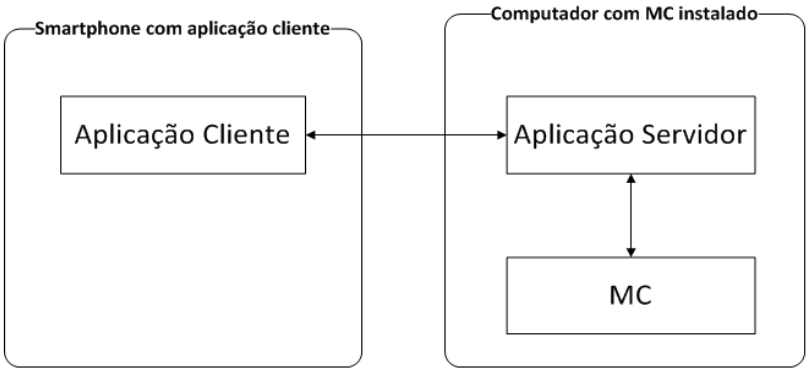

Figura 4.2: Arquitectura da Solução.

A comunicação entre a aplicação cliente e a aplicação servidor, é estabelecida através de uma ligação *Transmission Control Protocol/Internet Protocol* (TCP/IP). As aplicações cliente/servidor são capazes não só de estabelecer a comunicação entre o dispositivo móvel e o computador/set-top box, bem como ter conhecimento de quais as instruções a enviar ao MC para o mesmo reagir. Para tal ser possível, a aplicação servidor tem de saber exactamente os comandos necessários para por exemplo, navegar entre as diversas opções do MC, conseguir reproduzir vídeos/músicas, aumentar/reduzir o volume, etc.. Em suma, tem de ser capaz de realizar todas as opções possíveis de serem realizadas usando o teclado, rato ou o comando, para que o utilizador não sinta qualquer limitação com o uso de um dispositivo móvel.

Na Figura 4.3 encontra-se um diagrama de sequência de mensagens, que exemplifica a comunicação entre a aplicação cliente, servidor e MC para instruir o cursor do MC a subir, ou seja, o comando UP. Na Figura 4.4 encontra-se um diagrama de sequência de mensagens, entre a aplicação cliente, servidor e MC, para o MC devolver a localização das músicas. Após a recepção da localização, por parte da aplicação cliente, a mesma pede o conteúdo dessa localização, a fim de o mesmo ser visível ao utilizador no *smartphone*. Na Figura 4.5 encontra-se um diagrama de sequência de mensagens que representa essa comunicação.

Com o MCRemote obtêm-se a vantagem inerente à não utilização de periféricos dedicados ou de proximidade. Deixa também de existir a necessidade de linha de vista associada aos actuais comandos por infravermelho. A sua estrutura em modelo cliente/servidor, permite a qualquer altura, adicionar suporte para novos MCs do lado do servidor sem requerer a modificação da aplicação cliente.

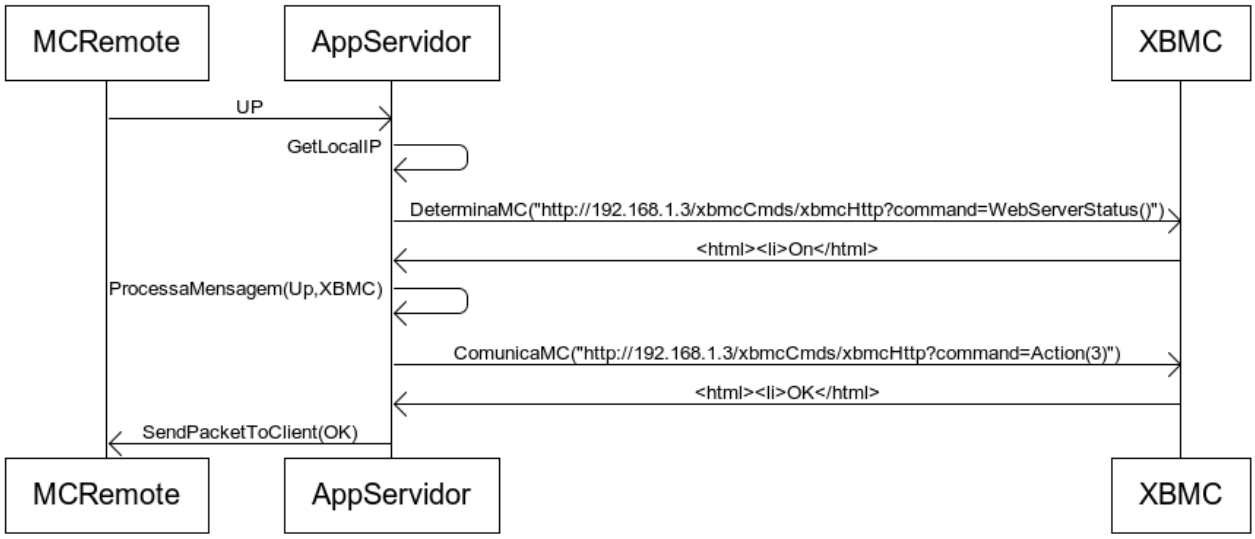

Figura 4.3: Diagrama de sequência de mensagens para instruir o MC a executar o comando UP.

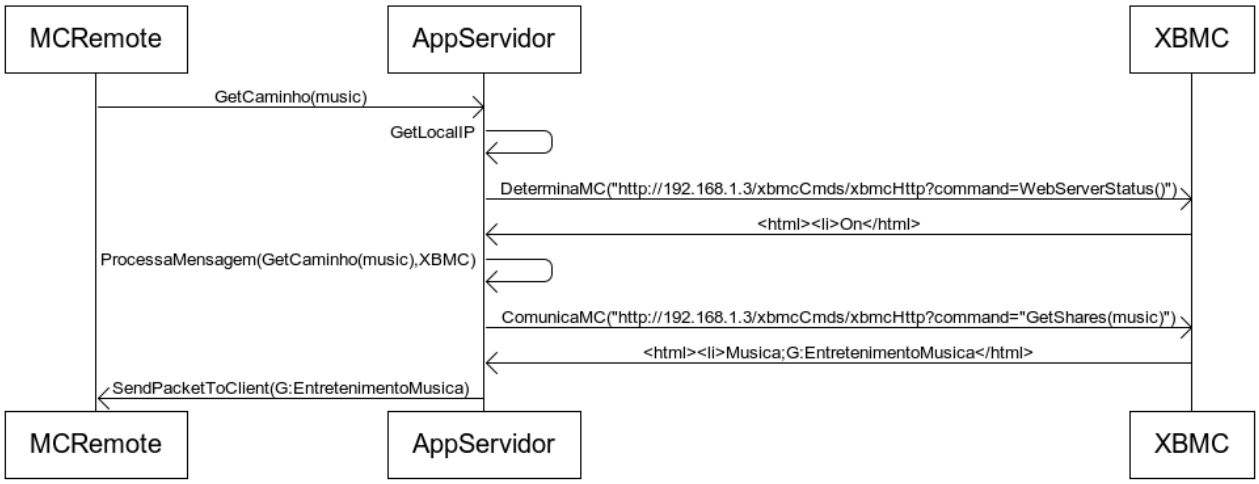

Figura 4.4: Diagrama de sequência de mensagens para determinar o caminho das músicas.

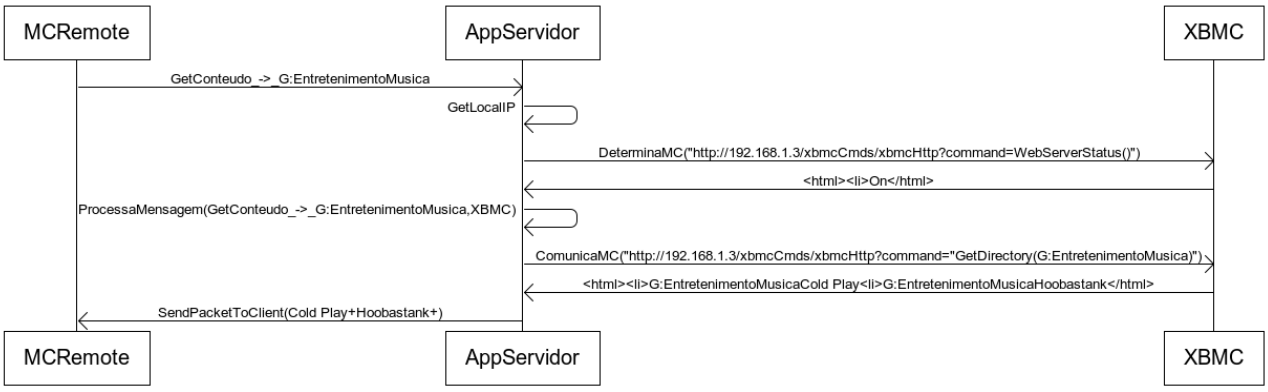

Figura 4.5: Diagrama de sequência de mensagens para obter o conteúdo de um determinado caminho.

4.3.1 MCRemote Server

A aplicação servidor tem de estar contida no computador onde se encontra o MC em funcionamento. Até à data, a aplicação servidor suporta dois MC's, sendo eles o XBMC e o Boxee. As várias etapas desde a recepção das mensagens enviadas pela aplicação cliente, comunicação com o MC, e resposta da operação à aplicação cliente, podem ser enumeradas da seguinte forma:

- Criação de Sockets para envio/recepção de mensagens;
- Aguarda a recepção de mensagens por parte da aplicação cliente;
- Processa a mensagem;
- Envia a resposta para a aplicação cliente.

Criação de Sockets para envio/recepção de mensagens

A aplicação servidor, no seu arranque começa por declarar dois DatagramPacket com o intuito de armazenar as mensagens recebidas e a enviar, entre as aplicações cliente e servidor (linhas 1 e 2). De seguida é criado um DatagramSocket na porta 5000, que é a porta onde é estabelecido o canal por onde vão ser transmitidas as mensagens entre as aplicações (linha 14). O bloco de código 4.1 representa este passo.

```
1  private DatagramPacket sendPacket, receivePacket;
2  private DatagramSocket socket;
3
4  public Server1() {
5          super("MCRemote - Server");
6
7          displayArea = new JTextArea();
8          getContentPane().add(new JScrollPane(displayArea), BorderLayout.CENTER);
9          setSize(1370, 730);
10         setVisible(true);
11
12         //cria DatagramSocket para enviar e receber pacotes
13         try {
14             socket = new DatagramSocket(5000);
15         } catch (SocketException socketException) {
16             System.exit(1);
17         }
18 }
```

Código 4.1: MCRemote - Server - Criação de DatagramSocket para envio e recepção de pacotes.

De seguida, é iniciada a aplicação servidor que começa por chamar o método waitForPackets(), que fica à espera de receber pacotes por parte da aplicação cliente.

Aguarda a recepção de mensagens por parte da aplicação cliente

No bloco de código 4.2 está representado o método onde a aplicação servidor está à escuta de receber pacotes por parte da aplicação cliente. Assim que detecta a recepção de um pacote, armazena o conteúdo do mesmo para posteriormente o poder processar (linha 6), enviando de seguida a resposta da operação à aplicação cliente.

```
1  public void waitForPackets() {
2          while (true) {
3              //recebe o pacote, processa o conteudo e responde para o cliente
4              try {
5                  byte[] data = new byte[256];
6                  receivePacket = new DatagramPacket(data, data.length);
7
8                  //espera o pacote
9                  socket.receive(receivePacket);
10
11                 //processa o pacote
12                 String resposta = process_packet();
13
14                 //ecoa as informacoes do pacote de volta para o cliente
15                 sendPacketToClient(resposta);
16
17             } catch (IOException ioException) {
18                 displayArea.append(ioException.toString() + "\n");
19             }
20         }
21     }
```

Código 4.2: MCRemote - Server - Aguarda recepção de pacotes enviados pela aplicação cliente.

Processa o pacote

Após a recepção do pacote, a aplicação servidor começa por filtrar o conteúdo da mensagem recebida. No próximo passo é determinado o IP da máquina onde a aplicação está a correr e de seguida determina qual o MC que está a ser executado no momento. A determinação do IP é importante, uma vez que essa informação é necessária para invocar os comandos junto das API's dos MCs. Tendo conhecimento do MC que está a ser executado, a aplicação determina qual a mensagem que necessita de enviar à API, e envia a mesma ao MC, aguardando o resultado da comunicação.

Para determinar qual o MC que está a ser executado, a aplicação servidor envia a cada um dos MCs a instrução `WebServerStatus([status])` (linha 6), uma vez que a mesma informa se o servidor web dos MC's está activo, que é uma condição necessária para a comunicação com os MCs ser possível. No bloco de código 4.3 representa a verificação se o XBMC está em execução.

```
1          endereco = "http://" + IP + "/xbmcCmds/xbmcHttp?command=WebServerStatus()";
```

```
2
3           try {
4               URL url = new URL(endereco);
5               //Verifica se recebe alguma resposta
6               BufferedReader reader = new BufferedReader(new InputStreamReader(url.
                    openStream()));
7               String line;
8               while ((line = reader.readLine()) != null) {
9                   displayArea.append("\n" + GetData() + ": Response from MC -
                        DeterminaMC - XBMC: " + line);
10                  MC="XBMC";
11                  flag=1;
12              }
13              displayArea.append("\n");
14          } catch (MalformedURLException e) {
15          } catch (IOException e) {
16          }
```

Código 4.3: MCRemote - Server - Determina qual MC está a ser executado.

Tendo conhecimento do MC que está a ser executado, torna-se possível determinar qual o comando a enviar ao mesmo e que corresponde à instrução enviada pela aplicação cliente. O bloco de código 4.4 representa a determinação dos comandos de navegação para controlar o XBMC.

```
1  if (MC.equals("XBMC")) {
2      if (mensagem.equals("Up")) {
3           mensagemFinal = "Action(3)";
4      } else if (mensagem.equals("Down")) {
5          mensagemFinal = "Action(4)";
6      } else if (mensagem.equals("Left")) {
7          mensagemFinal = "Action(1)";
8      } else if (mensagem.equals("Right")) {
9          mensagemFinal = "Action(2)";
10     } else if (mensagem.equals("Ok")) {
11         mensagemFinal = "Action(100)";
12     } else if (mensagem.equals("Previous_Menu")) {
13         mensagemFinal = "Action(9)";
14     } else if (mensagem.equals("Next_Menu")) {
15         mensagemFinal = "Action(7)";
16     } else if (mensagem.equals("Off")) {
17         mensagemFinal = "Shutdown()";
18     }
19 }
```

Código 4.4: MCRemote - Server - Processa mensagem a enviar ao MC.

No próximo passo, a aplicação servidor envia a instrução ao MC e guarda a resposta do mesmo ao comando invocado. Nesta fase, o MC executa a instrução e o utilizador visualiza essa acção no seu televisor. O bloco de código 4.5 representa o envio do comando ao MC e a recepção da resposta por parte de mesmo.

```
URL url = new URL(mensagem);

displayArea.append("\n" + GetData() +  ": Command sent to the MC: " + mensagem);

//Verifica se recebe alguma resposta
BufferedReader reader = new BufferedReader(new InputStreamReader(url.openStream()));
String line;

Integer flag = 0;
while ((line = reader.readLine()) != null) {
       displayArea.append("\n" + GetData() + ": Response from MC: " + line);

       if (line.equals("<html>")) {
       } else if (line.equals("</html>")) {
       } else if (line.equals("")) {
       } else if (line.contains("Error")) {
              RespostaMC += "Error+";
       } else {
             if (line.contains("OK")){
                       RespostaMC = RespostaMC +  "OK " ;
             } else {
             (...)
             (...)
             }
        }
}
```

Código 4.5: MCRemote - Server - Comunica com o MC.

Envia a resposta para a aplicação cliente

No fim do processo, a aplicação servidor, envia a resposta da execução do comando recebido, para a aplicação cliente. O bloco de código 4.6 representa o envio da mensagem para a aplicação cliente.

```
 private void sendPacketToClient(String resposta) throws IOException {
       displayArea.append("\n" + GetData() + ": Echo data to client -> " + resposta
           );

       byte data[] = resposta.getBytes();
       sendPacket = new DatagramPacket(data, data.length, receivePacket.getAddress
           (), receivePacket.getPort());
       socket.send(sendPacket);

       displayArea.append("\n" + GetData() + ": Packet sent\n
           ========================================\n");
       displayArea.setCaretPosition(displayArea.getText().length());
 }
```

Código 4.6: MCRemote - Server - Envia resposta para a aplicação cliente.

4.3.2 MCRemote Java

A aplicação cliente Java começa por pedir para o utilizador inserir o IP do PC onde a aplicação servidor está a ser executada. Seguidamente, cria uma conexão do tipo Datagrama (linha 4), e é através dessa conexão que serão enviadas as mensagens para a aplicação servidor. O bloco de código 4.7 representa este passo.

```
1  public void run() {
2
3          try {
4              DatagramConnection dc = (DatagramConnection)  Connector.open("datagram
                   ://" + servidor + ":" + porta);
5
6              sender = new Sender(dc);
7
8              while (true) {
9                  Datagram dg = dc.newDatagram(100);
10                 dc.receive(dg);
11             }
12         } catch (ConnectionNotFoundException cnfe) {
13             Alert a = new Alert("Media Center Remote Control", "The server is not
                   running", null, AlertType.ERROR);
14             a.setTimeout(Alert.FOREVER);
15             display.setCurrent(a);
16         } catch (IOException ioe) {
17             ioe.printStackTrace();
18         }
19 }
```

Código 4.7: MCRemote - Java - Estabelece canal de comunicação.

Após este passo, a aplicação cliente Java revela o menu principal, onde o utilizador pode escolher o tipo de acção que pretende que o MC execute. O bloco de código 4.8, mostra as possibilidades que o utilizador dispõe para conseguir navegar pelas opções do MC, no bloco de código 4.9, estão representadas as opções de reprodução dos conteúdos via MC, e por fim, no bloco de código 4.10, mostra as opções para controlar o volume do MC.

```
1  public void navigation() {
2          choose = new List("Navigation: ", Choice.EXCLUSIVE);
3          choose.addCommand(backCommand);
4          choose.addCommand(sendCommand);
5          choose.setCommandListener(this);
6
7          choose.append("Up", null);
8          choose.append("Down", null);
9          choose.append("Left", null);
10         choose.append("Right", null);
11         choose.append("Ok", null);
12         choose.append("Previous_Menu", null);
13         choose.append("Next_Menu", null);
```

```
14          choose.append("Off", null);
15          choose.setTicker(new Ticker("Media Center Remote Control"));
16          display.setCurrent(choose);
17          currentMenu = "navegation";
18  }
```

Código 4.8: MCRemote - Java - Opções do menu Navegação.

```
1   public void reproduction() {
2           choose = new List("Reproduction: ", Choice.EXCLUSIVE);
3           choose.addCommand(backCommand);
4           choose.addCommand(sendCommand);
5           choose.setCommandListener(this);
6
7           choose.append("Play", null);
8           choose.append("Pause", null);
9           choose.append("Stop", null);
10          choose.append("Next", null);
11          choose.append("Previous", null);
12          choose.append("Forward", null);
13          choose.append("Backward", null);
14          choose.setTicker(new Ticker("Media Center Remote Control"));
15          display.setCurrent(choose);
16          currentMenu = "reproduction";
17  }
```

Código 4.9: MCRemote - Java - Opções do menu Reprodução.

```
1   public void volume() {
2           choose = new List("Volume: ", Choice.EXCLUSIVE);
3           choose.addCommand(backCommand);
4           choose.addCommand(sendCommand);
5           choose.setCommandListener(this);
6           choose.setTicker(new Ticker("Media Center Remote Control"));
7
8           int i = 0;
9           while (i < 101) {
10              choose.append(Integer.toString(i), null);
11              i = i + 5;
12          }
13          display.setCurrent(choose);
14          currentMenu = "volume";
15  }
```

Código 4.10: MCRemote - Java - Opções do menu Volume.

Após o utilizador escolher alguma das opções disponíveis, a aplicação cliente utiliza a classe sender para enviar a instrução desejada para a aplicação servidor. No bloco de código 4.11 está representada essa classe.

```
1   public class Sender extends Thread {
2       private DatagramConnection dc;
```

```
3       private String address;
4       private String message;
5
6       public Sender(DatagramConnection dc) {
7           this.dc = dc;
8           start();
9       }
10
11      public synchronized void send(String addr, String msg) {
12          address = addr;
13          message = msg;
14          notify();
15      }
16
17      public synchronized void run() {
18          while (true) {
19              if (message == null) {
20                  try {
21                      wait();
22                  } catch (InterruptedException e) {
23                  }
24              }
25              try {
26                  byte[] bytes = message.getBytes();
27                  Datagram dg = null;
28
29                  if (address == null) {
30                      dg = dc.newDatagram(bytes, bytes.length);
31                  } else {
32                      dg = dc.newDatagram(bytes, bytes.length, address);
33                  }
34                  dc.send(dg);
35              } catch (Exception ioe) {
36                  ioe.printStackTrace();
37              }
38              message = null;
39          }
40      }
41  }
```

Código 4.11: MCRemote - Java - Comunicação com o servidor.

4.3.3 MCRemote Android

A aplicação cliente Android, começa por verificar se o *smartphone* tem Wifi activo, e caso essa condição não seja verdadeira, informa o utilizador. Esta verificação é importante, uma vez que sem o Wifi activo, não é possível comunicar com a aplicação servidor. No bloco de código 4.12 está representada a classe que faz a verificação do estado do Wifi.

```
1   import android.content.Context;
```

```
2  import android.net.wifi.WifiInfo;
3  import android.net.wifi.WifiManager;
4
5  public class ConectivityManager {
6
7          Context myContext;
8
9          public ConectivityManager(Context cxt){
10                 myContext = cxt;
11         }
12
13         public void startWifi(){
14            //start-wifi
15         }
16
17         public boolean WifiState () {
18
19                 boolean estado = false;
20
21                 WifiManager wifi;
22                 wifi = (WifiManager) myContext.getSystemService(Context.WIFI_SERVICE
                       );
23
24                 if (wifi.isWifiEnabled()) {
25                         if (wifi.getWifiState() == WifiManager.WIFI_STATE_ENABLED)
                               {
26                            estado =  true;
27                         } else {
28                                 estado =  false;
29                         }
30                 }
31                 return estado;
32         }
33
34     public String GetWifiIP() {
35                 String  IP = "";
36                 int ipAddress = 0;
37
38                 WifiManager wifi;
39                 wifi = (WifiManager) myContext.getSystemService(Context.WIFI_SERVICE
                       );
40
41                 WifiInfo info = wifi.getConnectionInfo();
42                 if (info.getBSSID() != null) {
43                         ipAddress = info.getIpAddress();
44                 }
45
46                 IP = intToIp(ipAddress);
47                 return IP;
48         }
49
50     private String intToIp(int i) {
```

```
51          return (( i & 0xFF) + "." +
52                      ((i >> 8 ) & 0xFF) + "." +
53                      ((i >> 16 ) & 0xFF) + "." +
54                      ((i >> 24 ) & 0xFF ));
55      }
56  }
```

Código 4.12: MCRemote - Android - Verificação do estado da conectividade Wifi no *smartphone*.

Após esta verificação, é apresentado ao utilizador a opção de inserir o IP do PC onde a aplicação servidor está a ser executada. Em alternativa, pode optar por ser a aplicação a executar uma pesquisa pela rede, a fim de descobrir automaticamente o(s) IP(s) em questão. Para esta descoberta ser possível, a aplicação começa por determinar qual o IP do *smartphone* na rede (linha 43 do bloco de código 4.12). De seguida, envia a todos os IP's da mesma gama, uma mensagem de teste e fica a aguardar resposta. Caso a resposta seja positiva, significa que no IP em causa está instalada a aplicação servidor e que a mesma respondeu positivamente à comunicação teste. No fim desta verificação, é apresentada ao utilizador uma lista com todos os PCs que têm a aplicação servidor em execução.

Uma vez definido o IP da localização da aplicação servidor, o mesmo é guardado nas definições gerais da aplicação. Estas definições, são usadas ao longo de toda a aplicação para armazenar e devolver informações como:

- IP do *smartphone* na rede;
- IP onde se encontra a aplicação servidor;
- A localização reconhecida pelo MC onde as fotos estão armazenadas;
- A localização reconhecida pelo MC onde as músicas estão armazenadas;
- A localização reconhecida pelo MC onde os vídeos estão armazenados.

O bloco de código 4.13 representa a classe onde são guardadas as definições gerais.

```
1   public class Settings extends Application {
2
3           public static String ServerIp = "";
4           public static Integer ServerPort = 5000;
5           public static String CaminhoMusicas = "";
6           public static String CaminhoVideos = "";
7           public static String CaminhoImagens = "";
8           public static String WifiIP = "";
9
10          public static Integer getServerPort(){
11                  return ServerPort;
12          }
13          public static void setServerIp(String s){
14                  ServerIp = s;
15          }
```

```
16          public static String getServerIp(){
17                  return ServerIp;
18          }
19          public static void setCaminhoMusicas(String s){
20                  CaminhoMusicas = s;
21          }
22          public static String getCaminhoMusicas(){
23                  return CaminhoMusicas;
24          }
25          public static void setCaminhoVideos(String s){
26                  CaminhoVideos = s;
27          }
28          public static String getCaminhoVideos(){
29                  return CaminhoVideos;
30          }
31          public static void setCaminhoImagens(String s){
32                  CaminhoImagens = s;
33          }
34          public static String getCaminhoImagens(){
35                  return CaminhoImagens;
36          }
37          public static void setWifiIP(String s){
38                  WifiIP = s;
39          }
40          public static String getWifiIP(){
41                  return WifiIP;
42          }
43  }
```

Código 4.13: MCRemote - Android - Regista as definições gerais.

Seguidamente, é apresentado o menu principal, onde é possível escolher entre as diversas opções, desde navegar pelo MC, controlar a reprodução dos conteúdos, mostrar listagens com todas as fotos, músicas e vídeos e reproduzir estes conteúdos. Após a escolha da opção desejada, é enviada a instrução à aplicação servidor para a mesma ser executada pelo MC. No bloco de código 4.14 está representado o envio da instrução *"Play"* para o servidor (linha 7).

```
1  final ConectivityManager conManager = new ConectivityManager(this);
2
3  btnPlay.setOnClickListener(new View.OnClickListener() {
4      public void onClick(View v) {
5          try {
6              if (conManager.WifiState() == true) {
7                  String Resposta = Sender.ComunicaServer("Play",2000);
8                  if (Resposta.equals("Unable to communicate")) {
9                      ProcessaErro("Unable to communicate with the Media Center.
                           The Media Center or the MCServer application may be off."
                           );
10                 }
11             } else {
12                 Settings.setWifiIP("");
```

```
13                  ProcessaErro("The wifi is not connected.");
14                  }
15          } catch (IOException e) {
16                  ProcessaErro(e.toString());
17          }
18      }
19  });
```

Código 4.14: MCRemote - Android - Envio da instrução *Play* para o servidor.

Todas os envios de mensagens e recepção dos resultados, entre a aplicação cliente e servidor, são efectuados através de uma classe. Esta, obtêm as informações necessárias presentes nas definições gerais da aplicação, tais como o IP do servidor e a porta, e posteriormente envia a instrução desejada à aplicação servidor através de um sockect criado para esse efeito (linha 13). Seguidamente, aguarda pela resposta por parte da aplicação servidor à instrução invocada (linha 20), e posteriormente, a função que invocou esta classe, trata o resultado da instrução desejada, de forma conveniente. No bloco de código 4.15 está representada a classe para envio/recepção de mensagens.

```
1  public class Sender extends Thread {
2
3      public static String ComunicaServer (String message, int timeout) throws
           IOException {
4
5        try {
6          int ServerPort = Settings.getServerPort();
7          String ServerIP = Settings.getServerIp();
8          DatagramSocket s = new DatagramSocket();
9          InetAddress local = InetAddress.getByName(ServerIP);
10         int msg_length = message.length();
11         byte[] message1 = message.getBytes();
12         DatagramPacket p = new DatagramPacket(message1, msg_length,local,ServerPort)
               ;
13         s.send(p);
14         s.setSoTimeout(timeout);
15
16         //RECEBE A RESPOSTA DO SERVIDOR
17         DatagramPacket receivePacket;
18         byte[] data = new byte[2560];
19         receivePacket = new DatagramPacket(data, data.length);
20         s.receive(receivePacket);
21         String respostaServer = new String(receivePacket.getData(), 0, receivePacket
               .getLength());
22
23         return respostaServer;
24
25           } catch (IOException e) {
26                 throw e;
27           }
28     }
29 }
```

Código 4.15: MCRemote - Android - Comunicação com o servidor.

4.4 Resumo

Este projecto foi pensado para ser executado num ambiente familiar, nomeadamente numa sala de estar, composta por uma Televisão/Plasma/LCD/LED, pelo menos um HTPC/PC ligado à TV com um MC e a aplicação servidor instaladas, um router wireless e um *smartphone*. O HTPC tem de estar conectado com o router wireless, para tornar possível a comunicação de outros dispositivos presentes na rede local com o mesmo. Será usado um *smartphone* para interagir com o MC.

O MCRemote encontra-se estruturado num conjunto de aplicações do tipo cliente/servidor. A aplicação cliente, está embutida no dispositivo móvel e tem como função enviar as instruções desejadas para a aplicação servidor e aguardar a resposta da operação. A aplicação servidor, encontra-se alojada no HTPC/PC e é responsável por converter as instruções recebidas e enviar as mesmas para o MC, para que o mesmo as possa executar. A comunicação entre a aplicação cliente e a aplicação servidor, é efectuada através de uma ligação TCP/IP. O facto de esta solução estar estruturada no modelo cliente/servidor, permite a qualquer altura, adicionar suporte para novos MCs do lado da aplicação servidor sem requerer a modificações das aplicações cliente.

A aplicação servidor começa por criar sockets para o envio/recepção das mensagens. De seguida, aguarda pela recepção das mesmas por parte da aplicação cliente, e assim que regista a recepção efectua o seu processamento. Neste, verifica qual o MC que se encontra em execução no momento, converte a mensagem recebida para uma instrução reconhecida pela API do MC e envia a mesma ao MC. No fim, envia o resultado da operação para a aplicação cliente.

Foram criadas duas aplicações cliente (Java e Android). A aplicação cliente Java, foi desenvolvida com o intuito de ser compatível com todos os sistemas operativos móveis. A aplicação cliente Android, foi desenvolvida com o intuito de ser disponibilizada uma solução com um interface e uma usabilidade superiores em relação à aplicação cliente Java. Estas aplicações, necessitam de saber o IP onde a aplicação servidor está a ser executada, para conseguirem comunicar com a mesma. As aplicações cliente, disponibilizam todas as opções/menus para permitir ao utilizador realizar todas as operações possíveis de serem realizadas num MC, utilizando periféricos dedicados ou de proximidade. Após o utilizador escolher a acção que pretende, é estabelecida a comunicação com a aplicação servidor e é-lhe enviada a instrução desejada, ficando a aguardar pelo resultado da operação.

Capítulo 5

Resultados

Neste capítulo são apresentados os resultados das três aplicações desenvolvidas. É também demonstrado, que a comunicação entre as aplicações cliente e servidor é realizada com sucesso, e que por sua vez, a aplicação servidor, consegue instruir os MCs a realizarem as operações desejadas.

A aplicação servidor actualmente suporta dois MCs, o XBMC e o Boxee. Em termos gráficos, apresenta apenas uma janela , não sendo assim necessário ao utilizador ter de efectuar alguma operação, como por exemplo, definir os locais onde se encontram os conteúdos multimédia, uma vez que esta gestão é da responsabilidade dos MCs. Todas as operações são realizadas em *background*, sendo apresentado ao utilizador um histórico das mesmas, para que o utilizador possa verificar o seu bom funcionamento. A Figura 5.1, mostra o resultado de uma solicitação por parte da aplicação cliente Android para testar a comunicação com o servidor.

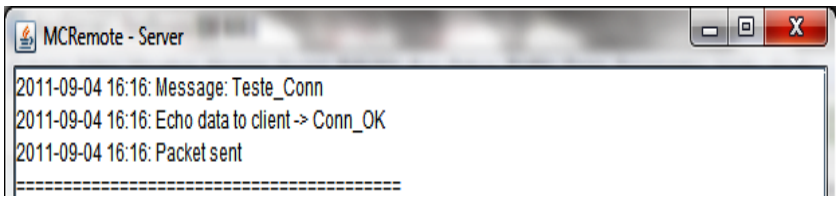

Figura 5.1: MCRemote - Server - Teste conecção.

A aplicação cliente Java foi desenvolvida com o intuito de ser compatível com todos os sistemas operativos móveis. Esta versão, consome poucos recursos, podendo assim ser executada na generalidade dos *smartphones*, mesmo quando estes dispõem de pouca capacidade de processamento e memória.

A desvantagem da aplicação cliente Java é a sua interface. Tendo em vista o aumento da portabilidade desta versão, a sua interface foi construída recorrendo aos elementos gráficos mais simples, logo mais disponíveis. Tal opção, teve impacto na usabilidade da aplicação final. A Figura 5.2 apresenta o ecrã de arranque da aplicação cliente Java.

Figura 5.2: MCRemote - Java - Intro.

A aplicação cliente Java fornece as seguintes funcionalidades:

- Inserção do IP da aplicação servidor (Figura 5.3);
- Apresenta um menu com as opções Navegação, Reprodução e Volume (Figura 5.4);
- Navegar pelas diversas opções do MC (Figura 5.5);
- Controlar a reprodução de conteúdos multimédia (Figura 5.6);
- Controlar o volume (Figura 5.7).

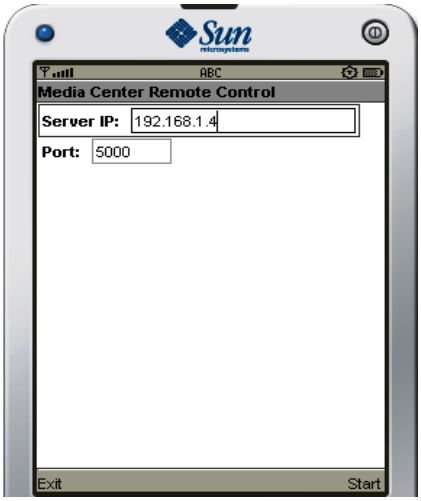

Figura 5.3: MCRemote - Java - Definições de conexão.

A Figura 5.8 apresenta o output do servidor quando este processa um pedido para o cursor do MC subir enviado pelo cliente.

A aplicação cliente Android, apresenta uma interface gráfica mais simples de se utilizar que a versão Java. Tal deriva directamente do facto de a plataforma Android ser mais recente. A escolha do sistema operativo móvel Android, prendeu-se com o facto de o mesmo ser o sistema operativo móvel mais acessível financeiramente do momento, de entre os três sistemas

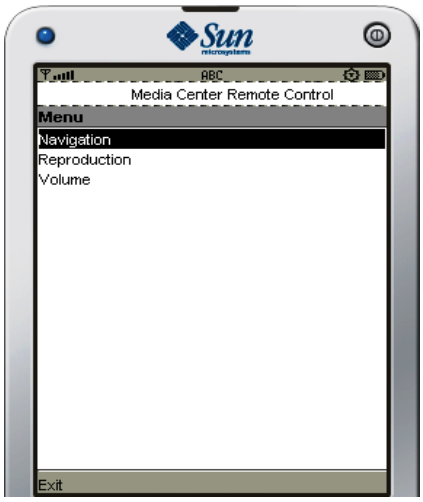

Figura 5.4: MCRemote - Java - Menu.

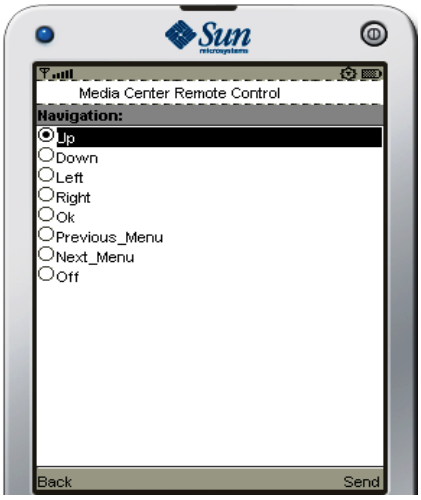

Figura 5.5: MCRemote - Java - Menu navegação.

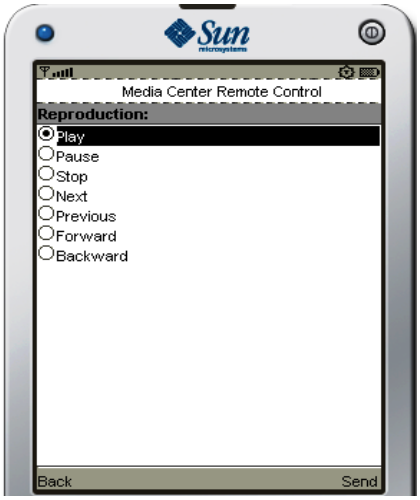

Figura 5.6: MCRemote - Java - Menu reprodução.

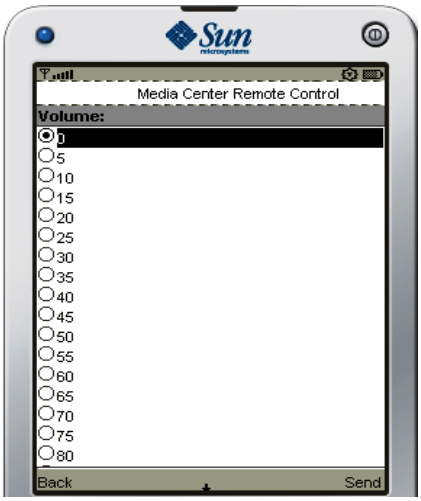

Figura 5.7: MCRemote - Java - Menu volume.

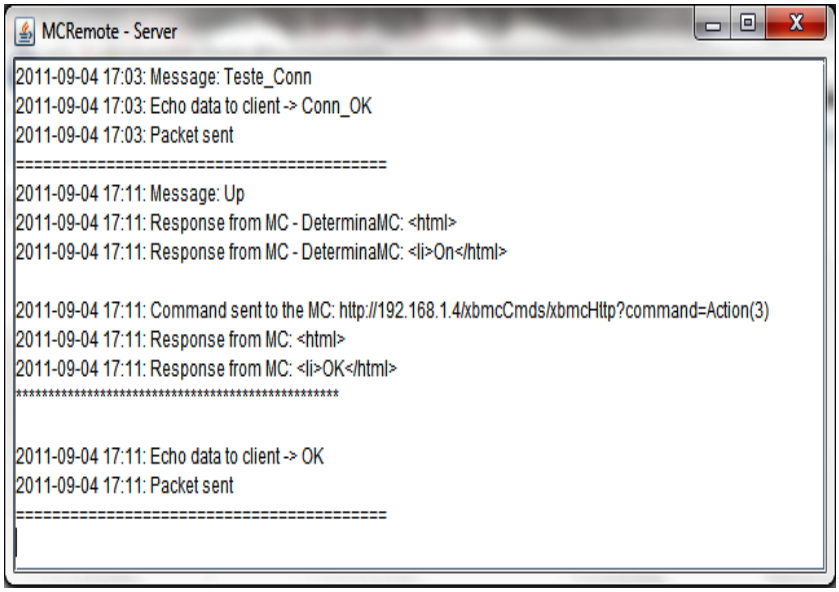

Figura 5.8: MCRemote - Server - Comando UP.

operativos móveis mais vendidos ou reconhecidos (iOS, Android e Windows Phone 7).

A aplicação cliente Android fornece as seguintes funcionalidades:

- Inserção do IP onde a aplicação servidor está a ser executada (Figura 5.9);
- Pesquisa na rede por aplicações servidor em execução (Figura 5.10);
- Apresenta um menu com as opções: Imagens, Musicas, Videos, Navegação, Reprodução e *Shutdown* (Figura 5.11);
- Navegar pelas opções do MC (Figura 5.12);
- Controlar a reprodução dos conteúdos multimédia através do MC (Figura 5.13).

A opção de pesquisar na rede por aplicações servidor, sofreu evoluções ao longo do tempo com o intuito de tornar a mesma mais rápida. A aplicação cliente Android segue os seguintes

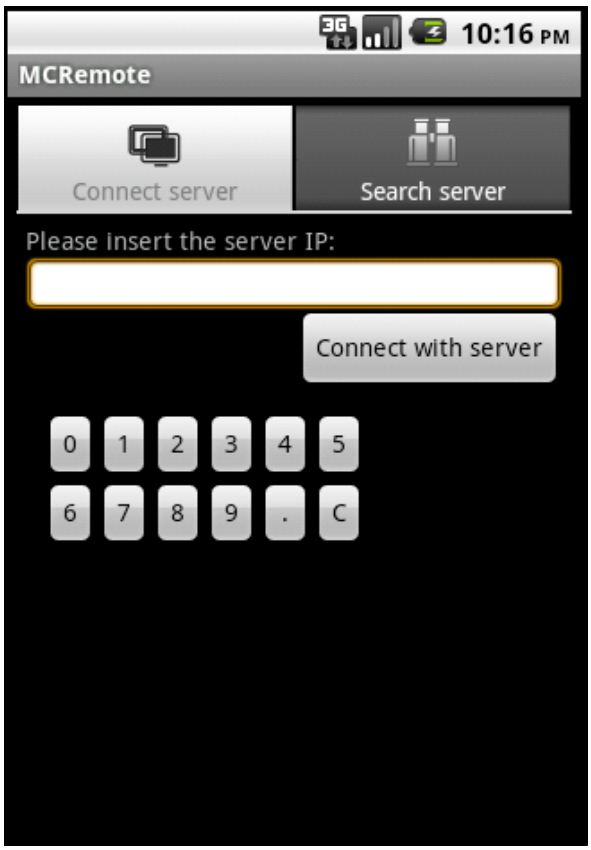

Figura 5.9: MCRemote - Android - Opção para inserir IP.

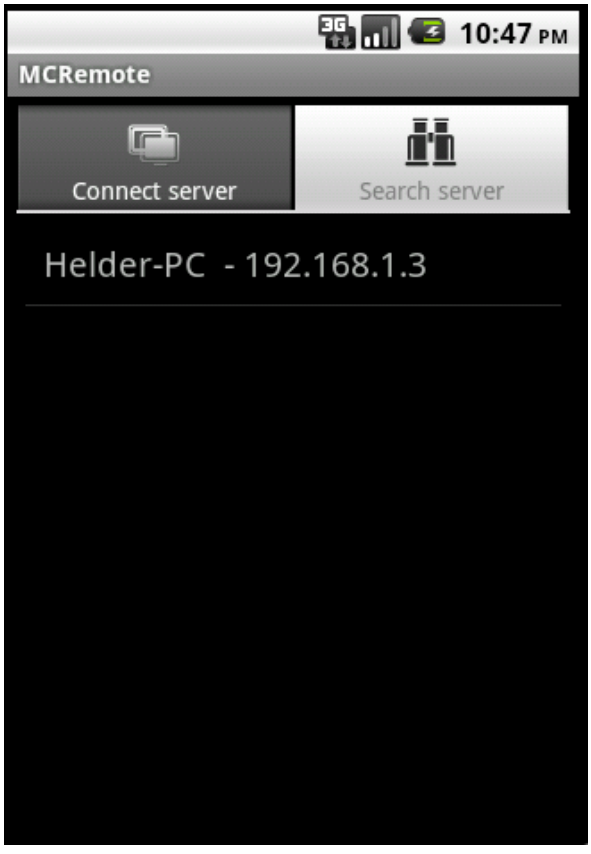

Figura 5.10: MCRemote - Android - Pesquisa automática da aplicação servidor.

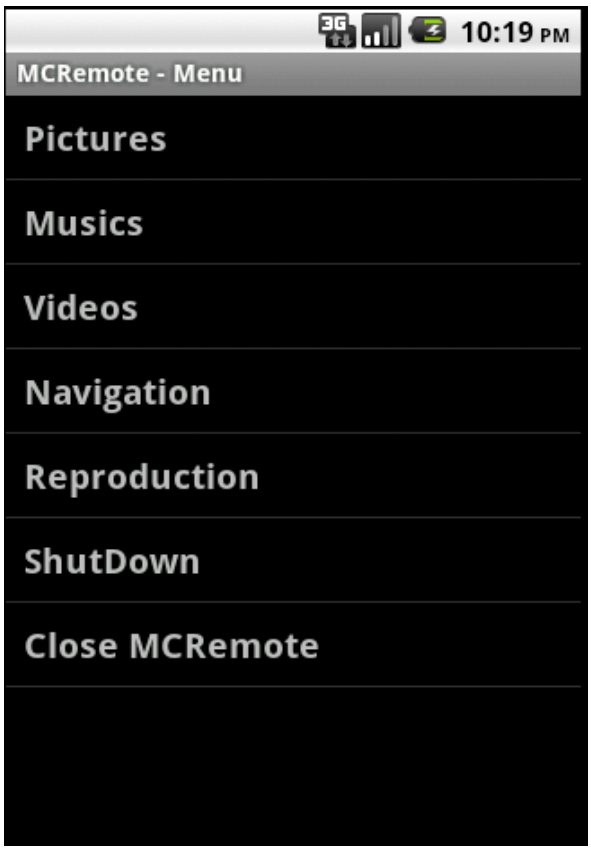

Figura 5.11: MCRemote - Android - Menu principal.

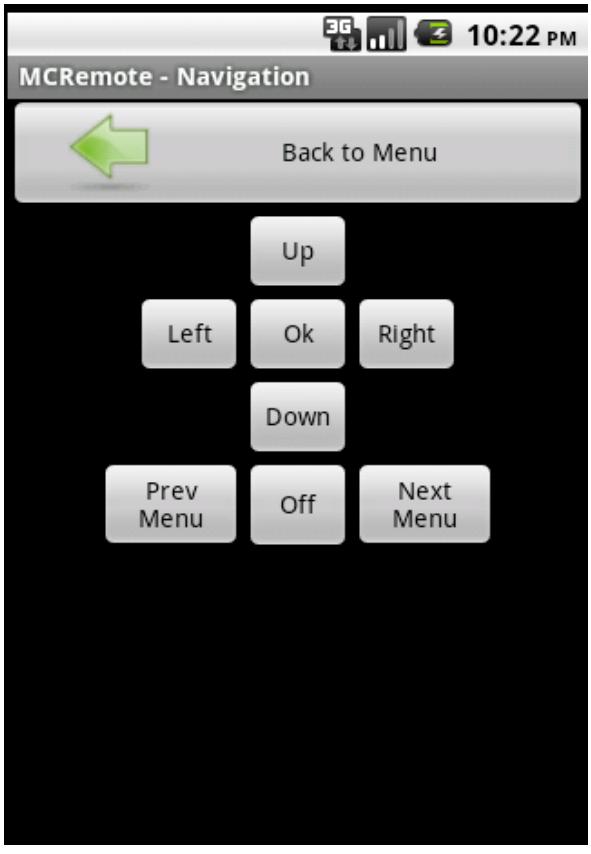

Figura 5.12: MCRemote - Android - Opções de navegação.

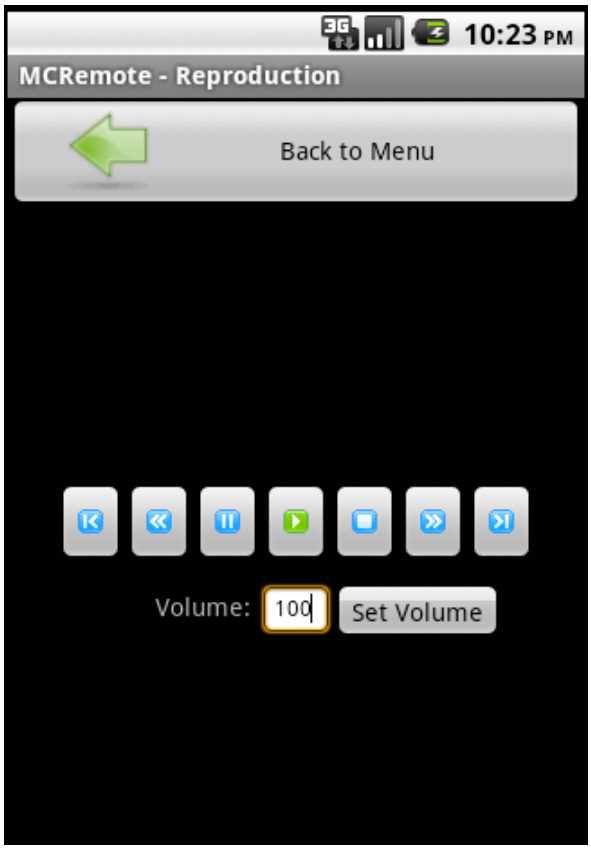

Figura 5.13: MCRemote - Android - Opções para controlar a reprodução.

passos:

- Verifica se o *smartphone* tem a opção Wifi activa;
- Caso a condição anterior seja verdadeira, verifica qual o IP do *smartphone* na rede local, para de seguida efectuar uma pesquisa pelos restantes IPs.

Na primeira versão, após a obtenção do IP, a aplicação cliente Android enviava um comando ping para os restantes IPs da rede local. Nos casos onde a resposta ao ping enviado era positiva, a aplicação enviava uma segunda instrução para esses IPs direccionada para a aplicação servidor, e aguardava a resposta por parte da mesma. Se a resposta fosse positiva, adicionava esse IP à lista de computadores na rede com a aplicação servidor em execução no momento.

Uma vez que esta pesquisa era muito demorada, pois era necessário aguardar o tempo de resposta dos pings enviados para cada IP, foram efectuadas alterações com o intuito de tornar a mesma mais rápida. Na versão actual, a aplicação Android assim que obtém o IP do *smartphone*, envia uma instrução para cada IP da rede em questão, assumindo que lá existe uma aplicação servidor. Em caso de resposta positiva, adiciona esse computador à lista de aplicações servidor. Esta nova pesquisa, demora no máximo trinta segundos. Na Figura 5.14, encontra-se representado um pedido de informação por parte da aplicação cliente Android, sobre o nome do computador na rede, para mostrar o mesmo na lista de computadores na

rede com a aplicação servidor em execução.

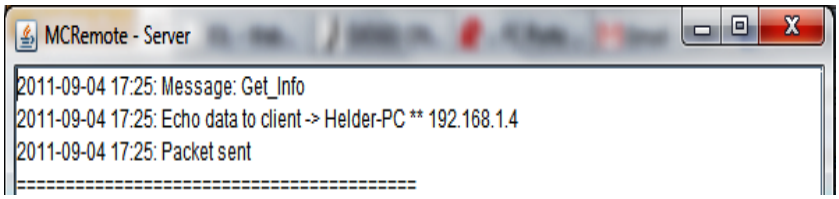

Figura 5.14: MCRemote - Server - Get PC Info.

De seguida, nas Figuras 5.15 e 5.16, encontra-se um exemplo do comportamento da aplicação servidor, a quando da recepção da mensagem por parte da aplicação cliente Android, para instruir o MC a devolver a lista dos artistas musicais.

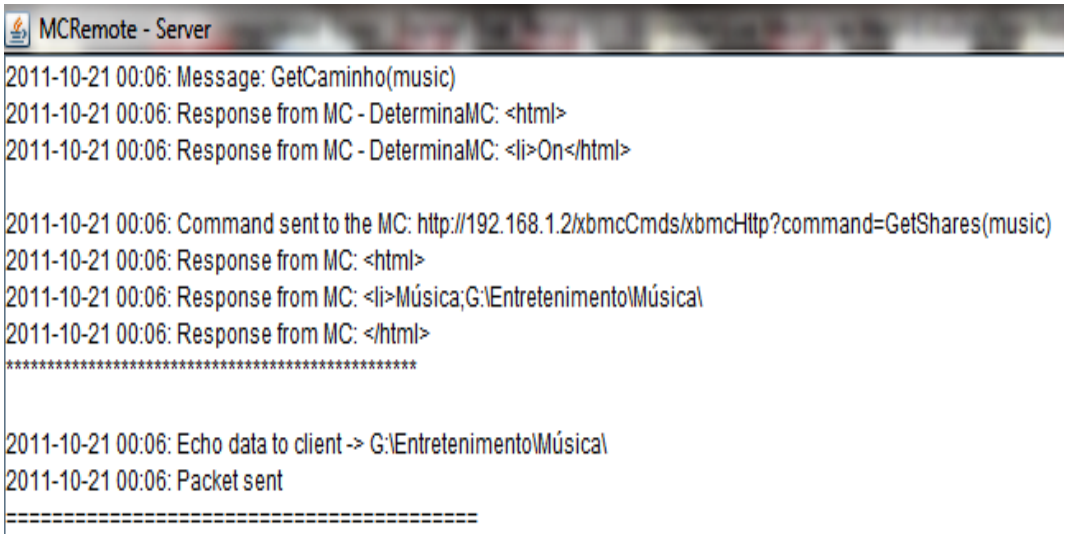

Figura 5.15: MCRemote - Server - Localização das músicas.

Na Figura 5.15, a aplicação servidor recebe a instrução para devolver a localização dos artistas musicais (`GetCaminho(music)`). Após invocar o comando junto do MC e a recepção da resposta, devolve a mesma à aplicação cliente (`G:\Entretenimento\Música`). Na Figura 5.16, a aplicação servidor recebe a instrução para devolver o conteúdo da localização determinada anteriormente, que no caso é a pasta onde se encontram os artistas musicais (`GetConteudo_->_G:\Entretenimento\Música`). No fim, devolve os mesmos à aplicação cliente.

O resultado desta sequência de mensagens na aplicação cliente Android, encontra-se na Figura 5.17.

Caso o utilizador seleccione alguma pasta, a aplicação cliente Android, envia uma instrução à aplicação servidor, para obter o conteúdo dessa pasta. Nas Figuras 5.18 e 5.19, encontram-se representados os passos desta comunicação, onde na Figura 5.18, foi pedido o conteúdo da pasta `G:\Entretenimento\Música\Cold Play`, e na Figura 5.19, foi pedido o conteúdo da pasta `G:\Entretenimento\Música\Cold Play\LeftRightLeftRiightLeft`.

```
2011-10-21 00:06: Message: GetConteudo_->_G:\Entretenimento\Música
2011-10-21 00:06: Response from MC - DeterminaMC: <html>
2011-10-21 00:06: Response from MC - DeterminaMC: <li>On</html>

2011-10-21 00:06: Command sent to the MC: http://192.168.1.2/xbmcCmds/xbmcHttp?command=GetDirectory(G:\Entretenimento\Música)
2011-10-21 00:06: Response from MC: <html>
2011-10-21 00:06: Response from MC:
2011-10-21 00:06: Response from MC: <li>G:\Entretenimento\Música\AC_DC Live_ Special Collector's Edition\
2011-10-21 00:06: Response from MC: <li>G:\Entretenimento\Música\Cold Play\
2011-10-21 00:06: Response from MC: <li>G:\Entretenimento\Música\Hoobastank\
2011-10-21 00:06: Response from MC: <li>G:\Entretenimento\Música\Iron Maiden\
2011-10-21 00:06: Response from MC: <li>G:\Entretenimento\Música\Linkin Park - Minutes To Midnight [2007][CD_SkidVid_Cov]\
2011-10-21 00:06: Response from MC: <li>G:\Entretenimento\Música\Metallica - 1999 - SM\</html>
**************************************************

2011-10-21 00:06: Echo data to client -> AC_DC Live_ Special Collector's Edition+Cold Play+Hoobastank+Iron Maiden+Linkin Park -
Minutes To Midnight [2007][CD_SkidVid_Cov]+Metallica - 1999 - SM+
2011-10-21 00:06: Packet sent
========================================
```

Figura 5.16: MCRemote - Server - Lista de artistas musicais.

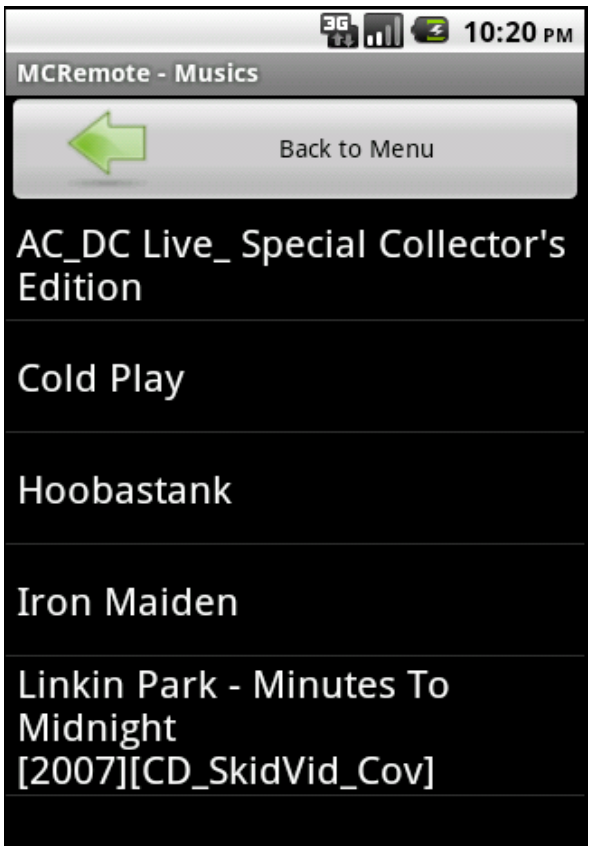

Figura 5.17: MCRemote - Android - Lista com músicas.

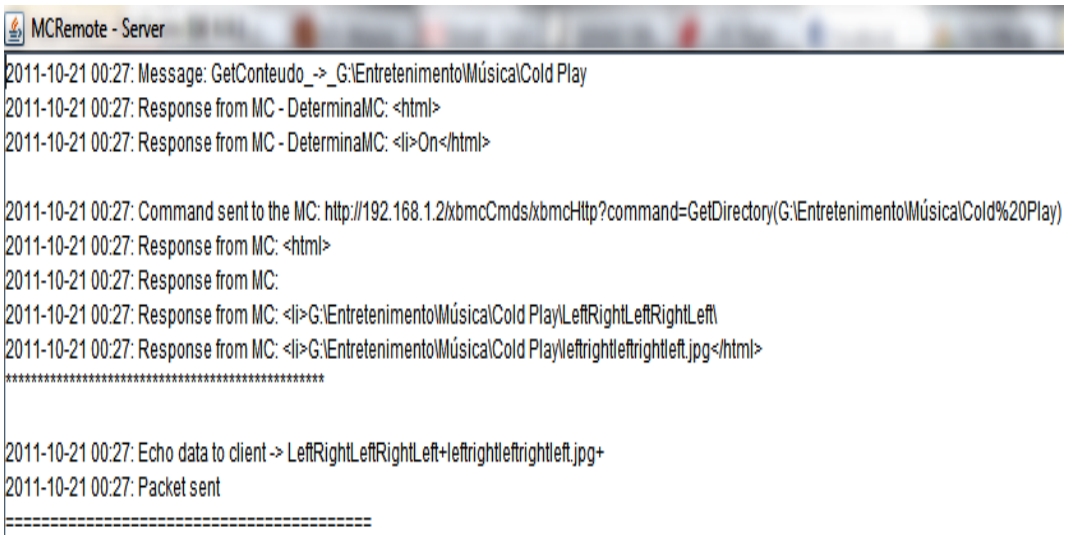

Figura 5.18: MCRemote - Server - Conteúdo de uma determinada pasta.

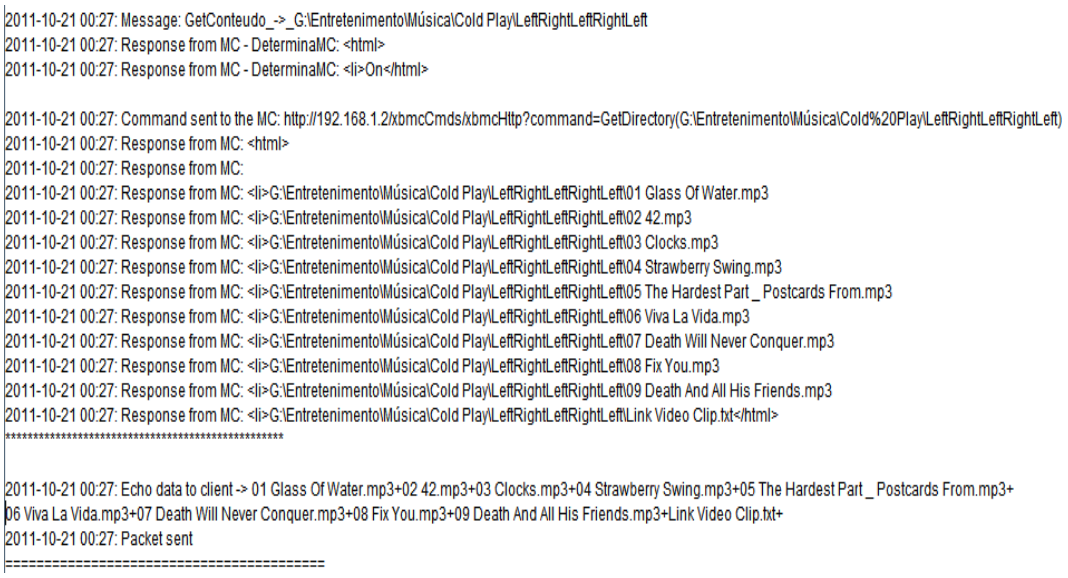

Figura 5.19: MCRemote - Server - Lista de músicas de um determinado artista.

Se o utilizador optar por reproduzir o conteúdo de uma determinada localização (Figura 5.20), a aplicação cliente Android, envia as instruções necessárias à aplicação servidor, para efectuar a reprodução. Nas Figuras 5.21 e 5.22, encontra-se representado todas as instruções por parte das aplicações para realizar este passo.

Caso o utilizador deseje obter a lista com todos os vídeos ou fotos, o procedimento das aplicações é similar ao anteriormente exemplificado, sendo que no caso dos vídeos, a instrução enviada pela aplicação cliente para obter a localização dos mesmos é `GetCaminho(video)`, e no caso das fotos a mesma é `GetCaminho(pictures)`. Após a obtenção das listagens e no caso de existirem sub pastas na localização dos vídeos e fotos, o utilizador pode optar por visualizar o conteúdo das mesmas, para posteriormente instruir o MC a reproduzir o seu conteúdo. Para instruir o MC a reproduzir uma série de vídeos, o comando enviado pela aplicação cliente à aplicação servidor é `PlayPlayList_Video`, enquanto que para reproduzir um *slideshow* de

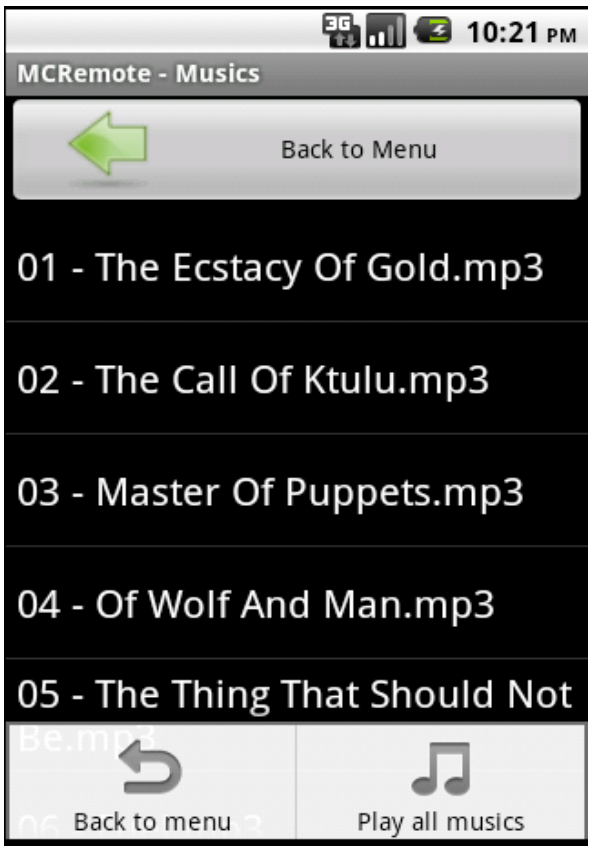

Figura 5.20: MCRemote - Android - Reprodução de músicas.

```
2011-10-21 00:30: Message: ClearPlayList_Music
2011-10-21 00:30: Response from MC - DeterminaMC: <html>
2011-10-21 00:30: Response from MC - DeterminaMC: <li>On</html>

2011-10-21 00:30: Command sent to the MC: http://192.168.1.2/xbmcCmds/xbmcHttp?command=ClearPlayList(0)
2011-10-21 00:30: Response from MC: <html>
2011-10-21 00:30: Response from MC: <li>OK</html>
**************************************************

2011-10-21 00:30: Echo data to client -> OK
2011-10-21 00:30: Packet sent
==========================================
2011-10-21 00:30: Message: AddPlayList_Music_-> _G:\Entretenimento\Música\Cold Play\LeftRightLeftRightLeft\
2011-10-21 00:30: Response from MC - DeterminaMC: <html>
2011-10-21 00:30: Response from MC - DeterminaMC: <li>On</html>

2011-10-21 00:30: Command sent to the MC: http://192.168.1.2/xbmcCmds/xbmcHttp?command=AddToPlayList(G:\\Entretenimento\\Música\\Cold%20Play\\LeftRightLeftRightLeft,[music])
2011-10-21 00:30: Response from MC: <html>
2011-10-21 00:30: Response from MC: <li>OK</html>
**************************************************

2011-10-21 00:30: Echo data to client -> OK
2011-10-21 00:30: Packet sent
==========================================
```

Figura 5.21: MCRemote - Server - Reprodução de uma lista de músicas (definição da *playlist*).

```
2011-10-21 00:30: Message: PlayPlayList_Music
2011-10-21 00:30: Response from MC - DeterminaMC: <html>
2011-10-21 00:30: Response from MC - DeterminaMC: <li>On</html>

2011-10-21 00:30: Command sent to the MC: http://192.168.1.2/xbmcCmds/xbmcHttp?command=SetCurrentPlaylist(0)
2011-10-21 00:30: Response from MC: <html>
2011-10-21 00:30: Response from MC: <li>OK</html>
**************************************************

2011-10-21 00:30: Echo data to client -> OK
2011-10-21 00:30: Packet sent
==========================================
2011-10-21 00:30: Message: PlayNextPlayList
2011-10-21 00:30: Response from MC - DeterminaMC: <html>
2011-10-21 00:30: Response from MC - DeterminaMC: <li>On</html>

2011-10-21 00:30: Command sent to the MC: http://192.168.1.2/xbmcCmds/xbmcHttp?command=PlayNext
2011-10-21 00:30: Response from MC: <html>
2011-10-21 00:30: Response from MC: <li>OK</html>
**************************************************

2011-10-21 00:30: Echo data to client -> OK
2011-10-21 00:30: Packet sent
==========================================
```

Figura 5.22: MCRemote - Server - Reprodução de uma lista de músicas (reprodução da *playlist*).

imagens é `PlaySlideShow`.

Capítulo 6

Conclusão

MCs permitem o armazenamento de conteúdos multimédia, para posterior reprodução. A interacção com este tipo de equipamentos, usualmente, é efectuada com recurso a periféricos dedicados e de proximidade. Tal implica algumas limitações de funcionalidade, comunicação e interoperabilidade. Alguns dos MCs de código aberto actualmente disponíveis *online*, oferecem interfaces de controlo acessíveis via rede (*Hypertext Transfer Protocol* (HTTP)), nomeadamente o XBMC, Boxee e o Plex. Tais interfaces, operam via web e executam as acções pretendidas em resposta a comandos invocados por parte das aplicações externas. Estes comandos, têm formatos específicos, e para cada acção existe uma instrução. Para que esta interacção seja possível, os MCs têm de ser configurados para aceitarem comunicações remotas, e têm de estar instalados em computadores ligados a uma rede.

Nos tempos mais recentes surgiram soluções com o propósito de controlar remotamente MCs, recorrendo para tal a *smartphones*, tais como o Gmote, AndroMote Remote Control, ou o Remote Media Center. Estas aplicações têm como vantagem a não necessidade do uso de periféricos dedicados e de proximidade para interagir com os MCs. O principal problema destas aplicações, reside no facto de a grande parte das mesmas só controlar um MC, não resolvendo assim o problema da interoperabilidade. O utilizador é obrigado a recorrer a aplicações diferentes para controlar MCs diferentes.

A solução MCRemote que recorre a *smartphones* que o utilizador tradicionalmente já possui, surge como forma de colmatar as limitações do uso de periféricos dedicados e de proximidade. Visa ainda, contornar o problema da interoperabilidade, uma vez que os MCs analisados não utilizam uma interface de controlo padrão. Esta solução encontra-se estruturada num modelo cliente/servidor, conseguindo assim suportar a interacção com diversos MCs em simultâneo, uma vez que a aplicação servidor é a responsável por efectuar a ponte entre as aplicações cliente e os MCs. Esta estruturação, também permite a qualquer altura adicionar suporte para novos MCs do lado da aplicação servidor, sem requerer a modificação das aplicações cliente. Actualmente, esta solução suporta a interacção com os MCs XBMC e Boxee, e disponibiliza duas aplicações clientes, a versão Java e a versão Android.

O MCRemote tem a capacidade de estabelecer a comunicação entre o dispositivo móvel

e o MC. Para tal, utiliza a rede local (sem fios), e faz uso de ums ligação TCP/IP para comandar o MC e permitir a navegação pelas diversas opções dos MCs. Permite-se assim que o utilizador consiga realizar todas as opções possíveis de serem realizadas aquando do uso de periféricos dedicados e de proximidade. Outras das características do MCRemote, é a sua usabilidade, sendo acessível a todos os utilizadores independentemente dos seus conhecimentos a nível informático. Neste aspecto, a aplicação cliente Android leva clara vantagem quando comparada com a aplicação cliente Java. O MCRemote consegue cumprir com todos os requisitos funcionais previamente identificados.

No futuro, a aplicação servidor irá suportar a interacção com mais tipos de MCs. Para além disso, a comunicação entre a aplicação servidor e o XBMC terá de ser adaptada para suportar o novo interface (JSON-RPC). A aplicação cliente Android, será melhorada ao nível visual tornando a aplicação mais atractiva. O procedimento de pesquisa na rede local por aplicações servidor será também melhorado. Aqui, após a primeira pesquisa, será guardada a lista de IPs encontrados. Em futuras pesquisas, verifica se a aplicação servidor ainda está disponível, diminuindo assim o tempo de pesquisa. Irá também ser desenvolvido a aplicação cliente para *smartphones* equipados com Windows Phone 7 e iOS.

Bibliografia

[1] J. Fitzpatrick and K. Purdy, “Which media center is right for you: Boxee, xbmc, and windows media center compared,” (2010, Fevereiro 2), [On-Line] Disponivel em: http://lifehacker.com/5462275/ [Ago. 10, 2009].

[2] G. Spagnolo, “Floss media centers state of the art comparison chart,” (2008, Setembro 18), [On-Line] Disponivel em: http://www.telematicsfreedom.org/en/project/14/floss-media-center-state-art [Jul. 21, 2009].

[3] XBMC, “Xbmc remote api,” [Jul. 10, 2011], internet: http://wiki.xbmc.org/index.php?title=Web_interface.

[4] Boxee, “Boxee remote api,” [Jul. 28, 2011], internet: http://developer.boxee.tv/Remote_Control_Interface.

[5] Plex, “Plex remote api,” [Set. 12, 2011], internet: http://forums.plexapp.com/index.php/topic/15850-plex-9-remote-api/.

[6] Gmote, “Gmote,” [Ago. 13, 2011], internet: http://www.gmote.org.

[7] AndroidZoom, “Media remote,” [Ago. 15, 2011], internet: http://pt.androidzoom.com/android_applications/media_and_video/media-remote-for-android_imez.html.

[8] H. Software, “Vlc remote,” [Ago. 15, 2011], internet: http://hobbyistsoftware.com/vlc-more.

[9] AndroMote, “Andromote remote control,” [Ago. 19, 2011], internet: http://www.andromote.de/.

[10] U. Forum, “Upnp,” [Ago. 20, 2011], internet: http://www.upnp.org/.

[11] A. Market, “Directtv remote free,” [Ago. 22, 2011], internet https://market.android.com/details?id=com.WiredDFW.DIRECTV.unWiredRemoteLite.

[12] R. M. Center, “Remote media center,” [Ago. 22, 2011], internet: http://myfrem.nl/.

[13] M. Remote, “My remote,” [Ago. 25, 2011], internet: https://sites.google.com/site/myremoteandroid/.

[14] Banamalon, “Wmc remote,” [Ago. 28, 2011], internet: http://www.banamalon.net/android/?p=1037.

[15] Android-xbmcremote, “Official xbmc remote,” [Ago. 29, 2011], internet: http://code.google.com/p/android-xbmcremote/.

[16] Boxee, “Boxee wifi remote,” [Ago. 31, 2011], internet: http://www.boxee.sunilsadasivan.com/.