Universidade Federal do Rio Grande do Norte
Centro de Ciências Exatas e da Terra
Departamento de Informática e Matemática Aplicada
Programa de Pós-Graduação em Sistemas e Computação
Mestrado Acadêmico em Sistemas e Computação

Solução para Interoperabilidade de Protocolos em Ambientes Inteligentes

Héldon José Oliveira Albuquerque

Natal-RN
Agosto/2014

Héldon José Oliveira Albuquerque

Solução para Interoperabilidade de Protocolos em Ambientes Inteligentes

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Sistemas e Computação do Departamento de Informática e Matemática Aplicada da Universidade Federal do Rio Grande do Norte como requisito parcial para a obtenção do grau de Mestre em Sistemas e Computação.

Linha de pesquisa:
Engenharia de Software

Orientador

Dr. Gibeon Soares de Aquino Junior

PPgSC – Programa de Pós-Graduação em Sistemas e Computação
DIMAp – Departamento de Informática e Matemática Aplicada
CCET – Centro de Ciências Exatas e da Terra
UFRN – Universidade Federal do Rio Grande do Norte

Natal-RN

Agosto/2014

Solução para Interoperabilidade de Protocolos em Ambientes Inteligentes

Autor: Héldon José Oliveira Albuquerque
Orientador(a): Dr. Gibeon Soares de Aquino Junior

Resumo

Nos últimos anos, foi possível observar como os dispositivos móveis vem inserido-se na vida das pessoas, tornando-se seus principais assistentes pessoais e ajudando em várias tarefas diárias. Contudo, não apenas os dispositivos móveis têm evoluído. Outros dispositivos eletrônicos também experimentaram mudanças que os tornam mais inteligente, capazes de processarem informação e tomar decisões próprias. Todos estes dispositivos interligados em uma mesma rede, fazendo uso de serviços e troca de informações com outros dispositivos, adaptando-se ao contexto ao qual o usuário permanece, caracterizam um ambiente inteligente. No entanto, devido ao rápido progresso da tecnologia e ao surgimento de um grande número de dispositivos com funções diferentes em um mesmo ambiente, uma diversidade de protocolos responsável pela intercomunicação entres os dispositivos em uma mesma rede foram criados, estabelecendo um cenário complexo para garantir interoperabilidade entre eles. Neste contexto, o principal objetivo deste trabalho é propor uma abordagem interoperável entre os dispositivos com diferentes protocolos de conectividade em ambientes inteligentes heterogêneos, para esse feito, realizamos uma analisado no estado da arte, com ênfase na interoperabilidade dos atuais dispositivos móveis com outros meios de dispositivos presentes em ambiente inteligentes. Assim, serão definidos quais protocolos de conectividade são atualmente mais utilizados para este fim, quais são os atuais projetos em andamento e quais são as limitações das soluções encontradas na pesquisa. Por fim, baseado nesta análise realizada, desenvolver uma proposta de solução para interoperabilidade em ambientes inteligentes que busca aproveitar os principais pontos fortes das propostas já existentes, enquanto solucionar as principais limitações identificadas nas mesmas.

Palavras-chave: casas inteligentes, protocolos de comunicação, conectividades, dispositivos móveis, ambientes inteligentes, interoperabilidade;

Solution to Interoperability Protocols Intelligent Environments

Author: Héldon José Oliveira Albuquerque
Supervisor: Dr. Gibeon Soares de Aquino Junior

Abstract

During the recent years, it was possible to observe how mobile devices inserted comes up in people's lives, becoming his main personal assistants and helping in various everyday tasks. However, not only mobile devices have evolved. Other electronic devices also experienced changes that make them more intelligent, able to process information and make decisions themselves. All these devices interconnected on the same network, making use of services and exchange information with other devices, adapting to the context in which the user stands, featuring an intelligent environment. However, due to the rapid advancement of technology and the emergence of a large number of devices with different functions in the same environment, a variety of protocols responsible for entres intercom devices on the same network were created, establishing a complex scenario to ensure interoperability among them. In this context, the main objective of this work is to propose an approach interoperable between devices with different connectivity protocols in heterogeneous smart environments, to that done, we conducted an analysis on the state of the art, with an emphasis on interoperability of current mobile devices with other means of devices present in intelligent environment. So, which will be defined connectivity protocols are currently most used for this purpose, which are the current projects, and what are the limitations of the solutions found in the search. Finally, based on this analysis, propose a solution for interoperability in smart environments that seeks to leverage key strengths of the existing proposals while addressing the main constraints identified in the same.

Keywords: Smarthome, systematic review, communication, connectivity, mobiles devices, smart environments, interoperability protocols;

Lista de figuras

1 Interação de diferentes tipos de ambientes inteligentes . . . . . p. 21

2 Integração das funcionalidade de dispositivos em casas inteligentes . . . p. 27

3 Camadas para interoperabilidade de protocolos com o meio de transmissão p. 30

4 Ponto de Controle UPnP, Dispositivo e Serviços . . . . . p. 33

5 Pilha de Protocolos UPnP . . . . . p. 35

6 Etapas para conectividade UPnP . . . . . p. 38

7 Descrição de um dispositivo UPnP . . . . . p. 39

8 Ambiente DLNA . . . . . p. 41

9 Pilha de Protocolos DLNA . . . . . p. 42

10 Procedimento para Revisão sistemática adaptado de (KITCHENHAM et al., 2010) . . . . . p. 46

11 Ferramenta JabRef . . . . . p. 51

12 Protocolos de Conectividade . . . . . p. 56

13 Cenários para inserção de novos protocolos . . . . . p. 63

14 Proposta para interoperabilidade entre protocolos . . . . . p. 68

15 Proxy para interoperabilidade entre protocolos UPnP e DLNA . . . . . p. 69

16 Comportamento funcional dos módulos do proxy . . . . . p. 71

17 Consumo de um serviço no dispositivo real ou virtual . . . . . p. 74

18 Mapeamento da rede multicast com a ferramenta Wireshark . . . . . p. 78

19 Descrição resumida de uma TV-DLNA . . . . . p. 79

20 Esquema XML da descrição de um TV-DLNA . . . . . p. 80

21 Esquema XML da descrição de serviço DLNA . . . . . p. 82

22 Similaridade dos esquemas XML dos dispositivos . . . . . p. 83
23 Cenário da aplicabilidade da proposta . . . . . p. 83
24 Cenário da aplicabilidade da proposta . . . . . p. 85
25 Diagrama de Classe do desenvolvimento do Proxy . . . . . p. 86
26 Listagem da Classe DescoberdaDLNA . . . . . p. 87
27 Listagem do método inicializarDescobertaDlna() da classe Descoberda-DLNA . . . . . p. 87
28 Listagem da classe DescobertaDispositivos . . . . . p. 89
29 Inicializando a descoberta dos dispositivos DLNA no ambiente . . . . . p. 90
30 Classe ExistenciaDispositivo . . . . . p. 91
31 Métodos da classe ExistenciaDispositivos . . . . . p. 91
32 Classe VirtualUPnP . . . . . p. 93
33 Continuação do método createDevice() da classe VirtualUPnP . . . . . p. 95
34 Método run() da classe VirtualUPnP . . . . . p. 95
35 Classe VirtualService . . . . . p. 96
36 Método verificarServico da Classe VirtualService . . . . . p. 97
37 Classe interna SwitchPower . . . . . p. 98
38 Método da classe SwitchPower responsável por consumir o serviço virutal local e o real conectado . . . . . p. 99
39 Método que consome um serviço do dispositivo real DLNA . . . . . p. 99

Lista de tabelas

1 Execução das Buscas . . . . . p. 51

2 Trabalhos aceitos na pesquisa . . . . . p. 52

3 Tabela de Consolidação . . . . . p. 53

4 Protocolos de Conectividade . . . . . p. 54

Lista de abreviaturas e siglas

GPS – Global Positioning System

RFID – Radio-Frequency IDentification

IoT – Internet of Things

IrDA – Infrared Data Association

OSI – Open Systems Interconnection

PPP – Point-to-Point Protocol

TCP – Transmission Control Protocol

UDP – User Datagram Protocol

FTP – File Transfer Protocol

ARP – Address Resolution Protocol

IP – Internet Protocol

SOAP – Simple Object Access Protocol

REST – Representational State Transfer

PnP – Plug and Play

PC – Personal Computer

P2P – Peer to pPeer

SSDP – Simple Service Discovery

GENA – General Event Notification Architecture

HTTP – Hypertext Transfer Protocol

SOAP – Simple Object Access Protocol

XML – Extensible Markup Language

RPC – Remote Procedure Call

DHCP – Dynamic Host Configuration Protocol

ICMP – Internet Control Message Protocol

RTP – Real -Time Transport Protocol

RTSP – Real Time Streaming Protocol

DMS – Digital Media Server

NAS – Network Attached Storage

DMP – Digital Media Player

DMR – Digital Media Renderer

DMC – Digital Media Controller

DMPr – Digital Media Printer

UPnP – Universal Plug and Play

DLNA – Digital Living Network Alliance

DPWS – Device Profile for Web Service

JVM – Java Virtual Machine

BSDP – Bluetooth Service Discovery Protocol

SDP – Service Discovery Protocol

HAVi – Home Audio Video Interoperability

IGRS – Intelligent Grouping and Resource Sharing

COAP – Constrained Application Protocol

DSB – Device Service Bus

Web4CE – Accessing Web-based Applications on Consumer Devices

Sumário

1 Introdução p. 12

1.1 Objetivos . . . . . p. 13

1.1.1 Geral . . . . . p. 14

1.1.2 Específicos . . . . . p. 14

1.2 Metodologia de Pesquisa . . . . . p. 14

1.3 Contribuições Acadêmicas . . . . . p. 15

1.4 Organização do trabalho . . . . . p. 15

2 Fundamentação Teórica p. 17

2.1 Ambiente Inteligente . . . . . p. 17

2.2 Internet das Coisas . . . . . p. 21

2.3 Casa Inteligente . . . . . p. 24

2.3.1 Dispositivos e protocolos de Conectividade . . . . . p. 28

2.4 Protocolos de Conectividade . . . . . p. 31

2.4.1 Universal Plung and Play (UPnP) . . . . . p. 32

2.4.1.1 Componentes de uma rede UPnP . . . . . p. 33

2.4.1.2 Pilha de Protocolos UPnP . . . . . p. 35

2.4.1.3 Etapas de Conectividade UPnP . . . . . p. 37

2.4.1.4 Endereçamento . . . . . p. 37

2.4.1.5 Descoberta . . . . . p. 38

2.4.1.6 Descrição . . . . . p. 39

2.4.1.7 Controle . . . . . p. 39

2.4.1.8 Eventos UPnP . . . . . p. 40
2.4.1.9 Apresentação . . . . . p. 40
2.4.2 Digital Living Network Alliance (DLNA) . . . . . p. 40
2.4.2.1 Pilha de Protocolos DLNA . . . . . p. 42
2.5 Considerações Finais do Capítulo . . . . . p. 45

3 Revisão Sistemática p. 46
3.1 Planejamento da Revisão Sistemática . . . . . p. 47
3.1.1 Protocolo da Revisão Sistemática . . . . . p. 47
3.2 Execução da Revisão Sistemática . . . . . p. 50
3.3 Análise dos Resultados . . . . . p. 53
3.3.1 Protocolos de Conectividade . . . . . p. 53
3.3.2 Interoperabilidade . . . . . p. 56
3.3.3 Principais descobertas . . . . . p. 60
3.3.4 Problema Abordado . . . . . p. 62
3.4 Considerações Finais do Capítulo . . . . . p. 64

4 Proposta p. 66
4.1 Uma solução baseada em Proxy . . . . . p. 67
4.2 Considerações Finais do Capítulo . . . . . p. 76

5 Implementação da solução baseada em Proxy p. 77
5.1 Diferenças entre os Protocolos UPnP e DLNA . . . . . p. 77
5.2 Estratégia para interoperabilidade dos protocolos DLNA e UPnP . . . p. 79
5.3 Estudo de Caso e Implementação da Solução . . . . . p. 83
5.4 Considerações Finais do Capítulo . . . . . p. 100

6 Considerações Finais p. 101

6.1 Conclusão . . . . . p. 101

6.2 Trabalhos Futuros . . . . . p. 102

6.3 Contribuições . . . . . p. 103

Referências p. 104

Apêndice A p. 109

1 Introdução

No ano de 2005, J. Krikke supôs que a computação móvel, em um futuro não tão distante, iria substituir a computação baseada em *desktops*. Que por sua vez, já era uma evolução em relação aos *mainframes* (KRIKKE, 2005). Tal suposição, segundo Kaed et al. (2011), Kim et al. (2012), Mitsugi et al. (2011), de fato se tornou verdadeira nos dias de hoje. Além disso, eles indicam que a computação ubíqua (WEISER, 1999) é a sucessora natural da computação móvel, portanto, uma realidade.

Os dispositivos móveis começaram a surgir na década de 80 (BARBARá, 2009), entretanto, foi somente no início do ano 2000 que apareceram os primeiros aparelhos com uma capacidade de processamento considerável (CORTIMIGLIA; GHEZZI; RENGA, 2011). Dentre estes, *smartphones*, *tablet*, *notebook* ou qualquer dispositivo que o usuário possa carregar consigo. Esses dispositivos ganham maior destaque em relação aos computadores pessoais devido à mobilidade que os mesmos proporcionam.

Ao longo dos últimos anos os dispositivos móveis se tornaram itens essenciais no cotidiano das pessoas. Eles se transformaram em seus principais assistentes pessoais, apresentando assim ferramentas de auxílio nas mais diversas tarefas do dia e dia. Entretanto, não só os dispositivos móveis evoluíram, outros dispositivos e eletrônicos sem mobilidade passaram por mudanças que os tornaram mais inteligentes (KAED et al., 2011), capazes de receberem informação de um outro dispositivo, processarem e realizar determinadas ações automaticamente. Eletrodomésticos como TVs, que exibem conteúdo da Internet e possuem capacidade de conexão com outros dispositivos para transmissão de áudio e vídeo de forma automática, geladeiras que fazem listas de compras ao detectar falta de determinado quantidade de alimentos, ar-condicionado e lâmpadas que são controlados pela internet e se auto configuram para se adaptar ao contexto do usuário, entre outros exemplos de dispositivos inteligentes sem mobilidade. Todos esses dispositivos interligados em um mesmo ambiente, ou em uma mesma rede, e interagindo com outros dispositivos, adaptando ao contexto dos usuários, auxiliando na vida das pessoas de forma autônoma, se caracteriza como um ambiente inteligente (CHEN; KUO; CHAO, 2009).

Um ambiente inteligente oferece uma visão do futuro para a sociedade da informação, onde são definidos conjuntos de cenários com objetivo de oferecer suporte inteligente nas atividades do dia a dia das pessoas (DUCATEL et al., 2005). Esses cenários são definidos por um conjunto de interfaces, sensores e dispositivos inteligentes que estão embutidos em todos os tipos de objetos em um ambiente, e este será capaz de reconhecer e responder à presença de diferentes usuários.

Em setembro de 2003, o projeto denominado *Housing Learning & Improvement Network* criado pelo *DTI SmatHomes Project*[1] abrange a primeira definição prática de um ambiente inteligente, em particular, uma casa inteligente. Também conhecida como, casa automatizada, Domótica ou *smarthomes* (RICQUEBOURG et al., 2006). Uma casa inteligente é um ambiente com uma diversidade de dispositivos que utilizam protocolos de comunicação heterogêneos, e a interoperabilidade entre eles é peça fundamental para a sua contribuição, como também, um dos seus principais problemas (ALLARD et al., 2004; CHEN; KUO; CHAO, 2009; MITSUGI et al., 2011). A maioria dos dispositivos é independente e oferece protocolos específicos que possam ser descobertos em uma rede doméstica.

A capacidade de um dispositivo com o uso de um protocolo de conectividade ser reconhecido em um ambiente de forma automática, segundo (KAED et al., 2011; LAI; HUANG, 2008), é peça fundamental para a concretização de um ambiente inteligente e função mais fascinante dos protocolos identificados nesse trabalho. Por outro lado, com a falta de um consenso ou norma que defina um arquitetura padrão para o uso dos protocolos de conectividade nos dispositivos presentes em tais ambientes e a diversidade destes protocolos de conectividade e tecnologias atualmente existentes, torna impeditiva a interoperabilidade transparente entre dispositivos, e consequentemente, inviabiliza a implementação plena de ambientes residenciais inteligentes. Este trabalho foca na interoperabilidade entre diferentes protocolos de conectividade, garantindo que dispositivos que utilizem diferentes protocolos possam interagirem entre si viabilizando o conceito de ambientes inteligentes.

1.1 Objetivos

Essa seção descreve o objetivo geral do trabalho, como também, os objetivos específicos que foram arbitrários para alcançar o desenvolvimento da solução proposta deste trabalho.

[1] http://www.fastuk.org/

1.1.1 Geral

- Desenvolver uma proposta de solução para a interoperabilidade entre diferentes protocolos de conectividade presentes em dispositivos heterogênicos em ambientes inteligentes.

1.1.2 Específicos

- Realizar uma análise do estado da arte com ênfase na interoperabilidade entre diferentes protocolos de conectividade em ambientes inteligentes;
- Definir quais os protocolos de conectividade mais utilizados nos dispositivos em ambientes inteligentes;
- Quais os atuais projetos em andamento, quais as limitações das soluções encontradas na pesquisa;
- Por fim, desenvolver uma proposta de uma solução alternativa para interoperabilidade entre tais protocolos;

1.2 Metodologia de Pesquisa

A metodologia de um trabalho de pesquisa é dada por um conjunto de normas, procedimentos, técnicas e ferramentas de análise que definem o padrão desejado para o desenvolvimento de projetos. Segundo Gil (2002), para garantir a investigação científica de um trabalho de pesquisa deve-se seguir um "conjunto de procedimentos intelectuais e técnicos para que seus objetivos sejam atingidos". Esses procedimentos são denominados métodos científicos. O método abordado para metodologia deste trabalho é a Revisão Sistemática (*Systematic Review*) baseada no modelo proposto por (KITCHENHAM et al., 2010). Uma revisão sistemática, assim como outros métodos científicos é uma forma de coletar informações na literatura sobre um tema em específico (GIL, 2002). Este utiliza como fonte de dados os diversos veículos de busca da literatura, seguindo um protocolo específico mediante a aplicação de métodos sistemáticos de busca, e a aplicação de critérios para escolha de trabalhos científicos sobre um determinado tema (KITCHENHAM et al., 2010).

Este método tem como principal objetivo, determinadamente, integrar as informações de um conjunto de estudos realizados separadamente sobre um tema específico, que possam manifestar resultados conflitantes e/ou coincidentes, tal como, identificar temas

que necessitam de evidência, auxiliando assim na orientação para investigações futuras (SAMPAIO; MANCINI, 2012) (KITCHENHAM et al., 2010). Além disso, após a análise dos diversos estudos sobre o tema, esta também permite ainda a elaboração de novas hipóteses ou propostas.

1.3 Contribuições Acadêmicas

Até o presente momento da apresentação desse trabalho, algumas contribuições já foram reportadas e aceitas na literatura, como mostrado a seguir:

- *ALBUQUERQUE, Héldon J.O.; SOARES, Gibeon A.J. TOWARDS A SOLUTION FOR INTEROPERABILITY OF SMARTHOMES DEVICES.First International Conference on Computer Networks & Communications (CCNET-2014). Venue: April 04-05 , 2014, Dubai.*
- *ALBUQUERQUE, Héldon J.O.; SOARES, Gibeon A.J. Solução para a Interoperabilidade de Protocolos em Smarthomes. 9ª Conferencia Ibérica de Sistemas y Tecnologías de Información (CISTI 2014), 18 y 21 de Junio de 2014, en Barcelona, España, unpublished.*
- *ALBUQUERQUE, H.J.O.; SOARES, G.A.J. A proxy-based solution for interoperability of smarthome protocols. The Eighth International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2014), Birmingham City University, July 2-4, 2014, unpublished.*

1.4 Organização do trabalho

O restante desse trabalho está organizado da seguinte forma:

- **Capítulo 2:** Define os principais conceitos teóricos necessários para o entendimento do trabalho, dando suporte teórico para os estudos, análise e reflexões sobre as informações coletadas.
- **Capítulo 3:** Esse capítulo é responsável por aplicar a revisão sistemática como metodologia de pesquisa desse trabalho, realizando uma investigação no estado da arte sobre os trabalhos acadêmicos que envolvem a área de pesquisa abordada nesse trabalho.

- **Capítulo 4:** Será definida uma proposta uma solução genérica e eficaz para o problema abordado na revisão sistemática, sobre as soluções existentes para a interoperabilidade de dispositivos em uma ambientes inteligentes.

- **Capítulo 5:** É apresentado o estudo de caso utilizando a proposta descrita nessa dissertação, como também, apresentado a arquitetura do projeto de desenvolvimento da proposta.

- **Capítulo 6:** As considerações finais do trabalho até o momento da pesquisa apresentada, resumindo as principais contribuições desta pesquisa e discussões a respeito das pesquisas futuras.

2 Fundamentação Teórica

Esta capítulo limita-se em apresentar os principais conceitos teóricos e científicos necessários ao desenvolvimento deste trabalho.

2.1 Ambiente Inteligente

Neste contexto, um ambiente inteligente oferece uma visão do futuro para a sociedade da informação, onde são definidos conjuntos de cenários com objetivo de oferecer suporte inteligente nas atividades do dia a dia das pessoas (DUCATEL et al., 2005). Essa visão, segundo Kaed et al. (2011), Kim et al. (2012), Mitsugi et al. (2011), já é uma realidade nos dias de hoje. Esses cenários são definidos por um conjunto de interfaces, sensores e dispositivos inteligentes que estão embutidos em todos os tipos de objetos em um ambiente, e esse será capaz de reconhecer e responder à presença de diferentes usuários (DUCATEL et al., 2005). Porém, em ambientes inteligentes, a abstração é peça fundamental para sua concretização, pois o usuário, necessariamente, não está interessado em saber da existência desses objetos inteligentes, ele deseja que essa interação seja de forma transparente, discreta e muitas vezes invisível (ISTAG, 2003).

Para a realização plena desse conceito, o ambiente inteligente coloca o usuário no centro do desenvolvimento. Portanto, a tecnologia deve ser projetada para adequar-se aos usuários, e não o inverso (DUCATEL et al., 2005). Essa abordagem não se aplica especificamente a ambientes inteligentes, ela foi explorada pela primeira vez na definição do termo Computação Ubíqua, definido por Mak Wieser em seu artigo, *The Computer for the 21st Century* (WEISER, 1999), onde ele aborda que, "em algum momento a tecnologia não ficará disponível somente em um ambiente, mas em todo lugar que estivermos tornando-se pervasiva nas nossas vidas". Ou seja, a tecnologia estará disponível de forma transparente nos mais triviais objetos que usamos, como etiquetas, xícaras, interruptores, luzes, canetas, etc, de certa forma, invisível para o usuário.

Um ambiente inteligente engloba qualquer ambiente que seja sensível e receptivo à presença de diferentes tipos de pessoas. Para isso, Weiser (1999), Gaggioli (2005) determinam que para um ambiente possa ser inteligente, esse precisa ser dividido em duas principais características: inteligência e incorporação. Respectivamente, a primeira refere-se ao fato de que o ambiente seja capaz de analisar o contexto, adaptar-se às pessoas e a seus objetos, analisar o comportamento delas, e eventualmente, prever determinadas ações. Já a segunda está relacionada diretamente aos dispositivos presentes no ambiente. Tais dispositivos se tornam parte do ambiente como um todo e e apresentam despercebidos para os usuários que nele estão. Portanto, de acordo com esse contexto, os usuários não vão usar a tecnologia, eles irão viver com a tecnologia (WEISER, 1999; GAGGIOLI, 2005; JIANHUA et al., 2005).

Estamos vivendo em uma época em que as tecnologias são comumente usadas, mas elas exigem uma elevada atenção dos usuários, que, em certos momentos, sentem-se desconfortáveis (SAMPAIO; MANCINI, 2012). Em um ambiente inteligente, os usuários estarão cientes da presença dessas tecnologias, entretanto, elas irão estar em segundo plano, como já acontece com outros usos comuns de tecnologias: motores, eletricidade, etc. Dispositivos presentes em ambientes inteligentes não são computadores pessoais, mas sim, um conjunto de pequenos dispositivos, independentes, interagindo entre si. Deste modo, o usuário interage de forma natural com esses dispositivos, usando a voz e os gestos de forma personalizada, utilizando o contexto e suas preferências (JIANHUA et al., 2005).

Além de todas as vantagens que o ambiente inteligente possa fornecer, como comodidade, conforto, agilidade, entre outras, ele pode levantar algumas preocupações sobre privacidade e problemas de segurança (CORONATO; TESTA, 2011). Alguns usuários também podem não querer viver ou confiar em um mundo virtual. É por isso que no ambiente inteligente deve prevalecer, mais que qualquer outra característica, a segurança (WEISER, 1999).

Para que todo o contexto que envolve um ambiente inteligente seja concretizado de forma ampla, algumas características abordadas (BAUKNECHT, 2002; ALCANIZ, 2005) devem ser levadas em consideração:

- **Interface com o usuário:** Os dispositivos que farão parte do ambiente devem ter uma boa qualidade de exibição e alta capacidade de resposta para o usuário. Ou seja, o uso desses dispositivos deve ser o mais claro e intuitivo quanto possível, e diferentes meios de entrada de dados podem ser utilizados, tais como canetas, escrita, fala ou de reconhecimento de gestos (BAUKNECHT, 2002).

- **Dispositivos de baixo custo:** Se em determinados casos vários dispositivos serão utilizados em um ambiente, esses tem que ser disponibilizado a um custo acessível para o usuário. Embora os *notebooks* e *smartsphones* de uso geral terem um custo mais elevado, os dispositivos em um ambiente inteligente não devem se encaixar nessa categoria. Como esses dispositivos são componentes específico para determinadas tarefas, sua estrutura é limitada a determinadas arquitetura, integrando-se de componentes básicos para realizar suas funções, com isso, tornando-os mais acessível para o que o usuário possa adquirir diferentes dispositivos com diferentes funcionalidades (DUCATEL et al., 2005).

- **Alta Largura de Banda:** Outro requisito fundamental é o ambiente ter largura de banda suficiente para permitir que a comunicação entre os diferentes dispositivos seja ampla, simplificada e ágil (DUCATEL et al., 2005).

- **Sistemas invisíveis:** Quando um usuário começa a usar um computador, ele tem que aprender alguns aspectos básicos sobre o sistema operacional. Em ambientes inteligentes a tecnologia deve tornar-se invisível, e serem capazes de compreender comportamentos do usuário, por exemplo, o uso de reconhecimento de voz ou outras interfaces intuitivas. Os usuários devem ser capazes de acessar os dados sem saber de arquivo específico como nomes, localização ou formato (BAUKNECHT, 2002).

- **Instalação automática:** Os sistemas devem eliminar a necessidade de instalação de programas. Nos atuais sistemas, a maioria dos mesmos precisam ser instalados e configurados e isso pode ser um problema para usuários leigos. Em qualquer um dos casos, requer a intervenção ativa do usuário. O conceito de instalação não tem sentido em ambientes inteligentes. Os programas e informações devem ser capazes de se mover a partir de um computador para outro (10).

- **Personalização:** Os sistemas no ambiente inteligente devem personalizar as informações apresentadas de forma diferente para cada utilizador, levando em consideração as características do usuário, como sua personalidade musical, cultural (BAUKNECHT, 2002; DUCATEL et al., 2005).

- **Privacidade:** Esse é um dos problemas mais importantes da utilização de sistemas em ambientes inteligentes, podendo gerar graves riscos de privacidade (WEISER, 1999; CORONATO; TESTA, 2011). Por exemplo, os sistemas podem gravar ações dos usuários, suas preferências, seus locais, seus gostos particulares, etc, e essas informações podem ser acessadas por outros usuários em um mesmo ambiente. O uso de

criptografia nesses casos deve ser constante, independendo do meio de comunicação usado para troca de informações entre sistemas e dispositivos (BAUKNECHT, 2002).

A integridade de todas essas características interligadas entre si, torna o ambiente inteligente (BAUKNECHT, 2002; DUCATEL et al., 2005). Entretanto, essas características não podem ser aplicadas no próprio ambiente, e sim, nos responsáveis por tornar um ambiente inteligente, os dispositivos. Tecnicamente, a maioria das pessoas pensam que já vivem em ambientes considerados "inteligentes", mas eles convivem com a utilização de diversos tipos de sensores, como por exemplo sensores de iluminação e de segurança. Os recentes avanços eletrônicos aumentaram a autonomia desses dispositivos tornando-os não inteligentes, e sim automatizados. E esses são conceitos bem diferentes (ALAM; REAZ; ALI, 2012).

Outra questão importante que não pode faltar para a operabilidade em um ambiente inteligente é a combinação do uso de mobilidade (KAR, 2013). Dispositivos móveis e inteligentes oferecem ao ambiente uma interpretação mais significativa e confiável para interação do usuário no contexto. Isto é, a capacidade de um dispositivo móvel capturar informações sobre a mobilidade do usuário, em diferentes ambientes, ajuda-o a criar um modelo mais rico de interação com o mesmo quando estiver em outro ambiente, como seu ambiente residencial por exemplo. Os dispositivos fixos do ambiente compartilharão das informações que o dispositivo móvel capturou durante a locomoção do usuário, permitindo assim, que o ambiente adote uma melhor adaptação às condições ambientais, tais como condições acústicas, luz, privacidade e etc (ALAM; REAZ; ALI, 2012; KAR, 2013).

A combinação desses diferentes tipos de dispositivos, como computadores pessoais, *smartphones*, GPS (*Global Positioning System*), *tablets*, sensores, etiquetas RFID (*Radio-Frequency IDentification*), sensores de movimento, todos interligados entre si, torna o ambiente mais sensível ao contexto do usuário (DUCATEL et al., 2005; ALAM; REAZ; ALI, 2012; KAR, 2013). Além disso, os recentes avanços computacionais e eletrônicos tornaram possível diferentes pesquisas trabalharem em conceitos mais específicos, tais como casas inteligentes (*smarthome*) (ALAM; REAZ; ALI, 2012), hospitais inteligentes (*smart hospitals*) (CHACZKO et al., 2010), cidades inteligentes (*smart cities*) (TOMAS et al., 2013), trânsitos inteligentes (*intelligent traffic*) (FANG, 2009), etc, e essas pesquisas vão além, como pode-se observar no exemplo da Figura 1.

A associação de diferentes ambientes inteligentes interligados torna a vida das pessoas também mais "inteligente" (ACAMPORA et al., 2013). Como ilustrado anteriormente, a interligação de diferentes ambientes inteligentes pode gratificar a vida das pessoas a

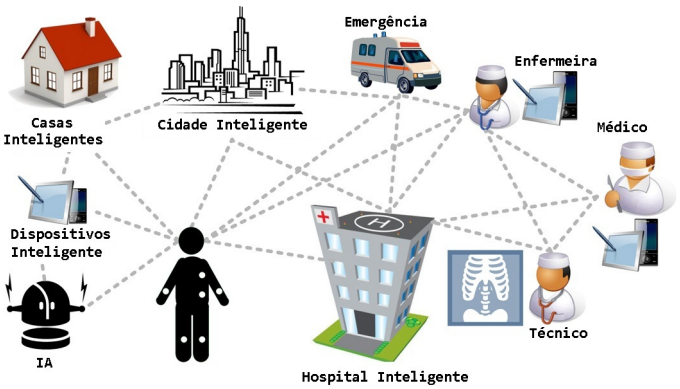


Figura 1: Interação de diferentes tipos de ambientes inteligentes
Fonte: (ACAMPORA et al., 2013)

tornar-se mais saudável e construtiva. Por exemplo, o monitoramento do estado de saúde de idosos ou pessoas com doenças crônicas, monitoramento e cuidados de assistência a pessoas com limitações físicas ou mentais em qualquer lugar que elas estejam, casa, na cidade, no transito (ACAMPORA et al., 2013). Essa intercomunicação de ambiente pode ser usada para o desenvolvimento de serviços de persuasão a fim de motivar as pessoas a levarem uma vida mais saudável. Por último, pode apoiar também os profissionais de saúde em termos de monitoramento, antecipando e preparando o ambiente de trabalho para receber aquele problema em específico. Essa combinação de recursos inteligentes irá fornecer monitoramento para uma vida mais saudável e discreta (ACAMPORA et al., 2013).

2.2 Internet das Coisas

O próximo passo para a evolução da *Internet* é com relação à integração de objetos físicos do dia a dia às redes de comunicação (GUBBI et al., 2013). Esse novo paradigma computacional integra uma grande variedade de conceitos e áreas, tais como eletrônica, automação, redes de comunicação, biotecnologia, mecânica, entre outras (GUBBI et al., 2013).

A Internet das Coisas (*Internet of Things*), como é comumente denominada, apresenta uma visão que a *Internet* se estende além do uso direto de equipamentos computacional, e sim abrange o mundo real, abraçando principalmente objetos do cotidiano das pessoas. Objetos esses que estão conectados do mundo virtual, podem ser controlados remotamente ou podem agir como pontos de acesso físico a serviços de *Internet* (MATTERN; FLOERKEMEIER, 2013).

Essa é uma visão baseada na crença que os avanços constantes da microeletrônica,

comunicações e tecnologia da informação, continuará em um futuro previsível. Na verdade, com a diminuição de tamanho, preço constantemente caindo e menor consumo de energia, processadores, módulos de comunicação e outros componentes eletrônicos estão sendo cada vez mais integrados em objetos do cotidiano (MATTERN; FLOERKEMEIER, 2013).

O principal objetivo da IoT (*Internet of Things*) consiste na presença da heterogeneidade de objetos e da interação entre si, a fim de alcançar um objetivo em comum, utilizando protocolo de comunicação comumente padronizados (ATZORI; IERA; MORABITO, 2010) da *Internet*. A integração entre esses objetos, de certa forma "inteligentes", sobre a *Internet* formam a base tecnológica para o conceito de ambientes inteligentes, nos quais a informação gerada por um objeto pode ser compartilhada entre diversas plataformas e aplicações (GUBBI et al., 2013).

Segundo Weiser (1991), "a IoT torna a computação verdadeiramente onipresente", e este desenvolvimento está abrindo grandes oportunidades. Nesse momento, considera-se que a IoT representa a próxima evolução da *Internet*, pelo o enorme salto na sua capacidade de coletar, analisar e distribuir informação, transformando objetos estáticos em coisas novas e dinâmicas, misturando inteligência ao meio e estimulando a criação de produtos inovadores (MATTERN; FLOERKEMEIER, 2013; GUBBI et al., 2013).

Esse paradigma nasceu no *MIT Auto-ID Laboratory*[1] do Instituto de Tecnologia de Massachusetts, e ganhou o nome de *Internet of Things*, partindo de um objetivo bem simples, criar um sistema global de registro de bens, utilizando um código de produto eletrônico (*Electronic Product Code*) como sistema de numeração, recorrendo ao uso do RFID e WSN (*Wireless Sensor Networks*) (GUBBI et al., 2013).

Entretanto, a IoT já pode ser considerada, por parte, uma realidade, como observador em (GUBBI et al., 2013). O mesmo destaca tecnologias e dispositivo que já fazem parte desse paradigma e que estão sendo frequentemente utilizadas ao longo de sua evolução, como por exemplo, dispositivos "vestíveis" (relógios, pulseiras, tênis) que interagem atividades cotidianas à internet, automóveis que utiliza equipamentos que sugerem melhores rotas para evitarem engarrafamento, carteira digital que realizam pagamentos usando o celular, sem necessidade de dinheiro ou cartões de crédito, entre outros. Ainda abordando (GUBBI et al., 2013), para um melhor compreensão desse paradigma algumas características deverão ser utilizadas por objetos ou dispositivos em uma rede da IoT:

- **Auto conhecimento:** Todo componente deverá ter, explicitamente ou implicita-

[1] http://www.autoidlabs.org/

mente, um identificação que a define, podendo processar informação, tomar decisões e se comportarem de maneira autônoma.

- **Existência:** Qualquer objeto que exista em um mundo real, poderá existir no mundo virtual da IoT.
- **Conectividade:** Os objetos podem e devem interagir com outros objetos e tecnologias independente da rede que esses estejam.
- **Interatividade:** Os objetos podem interoperar e colaborar com uma variedade de entidades heterogêneas, sejam essas maquinas virtuais, reais ou com os próprios usuários. Com isso, essa interatividade entre as partes produzem e consomem uma grande variedade de serviços;
- **Dinamicidade:** Objetos podem interagir entre si em qualquer hora, lugar e de qualquer maneira, sem que estejam limitadas a um único local.
- **Consumo:** Esse pode ser um dos principais problemas para essa tecnologia. Dispositivo que realizam processamento requer um consumo maior de energia, por isso, esses objetos devem ser capazes de operar por um longo tempo por conta própria, tomando e realizando estratégia para a economia de energia.
- **Segurança:** Os objetos deverão transmitir informações dos usuários de forma privada e criptografada. Esse é um dos principais motivos que muitos usuários adotam para não utilizarem desse novo conceito.

Essas características incorporadas com os objetos, de certa maneira "inteligentes", possuem a capacidade de alterar o estado do ambiente ao seu redor, ou além, podem alterar até mesmo o seu próprio estado. Isso implica em uma quantidade enorme de dados e informações que precisam ser armazenadas, processadas e apresentadas de modo eficiente e interativo (GUBBI et al., 2013). Com isso, para uma melhor adaptação e compreensão desse paradigma no mundo atual que envolva tecnologias, conceitos e padrões já habituadas, Ovidiu et al. (2011) apresenta diferentes perspectivas que a IoT abrange:

- **Oriented Internet Vision (OIV):** Essa perspectiva abrange os protocolos necessários para permitir a troca de informações entre os objetos na IoT. Nessa visão, estão as pesquisas e padrões que tratam da adaptação do protocolo IP para o ambiente de IoT;

- **Oriented Things Vision (OTV):** Essa visão considera os objetos itens simples, como etiquetas RFID (Radio-Frequency IDentification), entretanto, trata de endereçamento único e global para cada objeto tenha acesso direto a Internet (para acesso direto às coisas por meio da Internet) e da identificação unívoca das coisas.
- **Oriented View Semantics(OVS):** A quantidade diversificada de objetos conectados na rede e interconectados entre si está destinada a ser muito elevado. Neste contexto, as tecnologias semânticas desempenham um papel fundamental, pois são utilizadas para modelar objetos como também a arquitetura que irá acomodá-los.

O trabalho de Ovidiu et al. (2011) aborda com mais detalhes essas perspectivas, como também, propõem estratégias para sua aplicabilidade. Por fim, é necessário superar inúmeros desafios científicos e tecnológicos para que sejam utilizadas e difundidos os conceitos e práticas das IoT em sua forma plena.

2.3 Casa Inteligente

Casas inteligentes, residências inteligentes, domótica ou *smarthomes* são conceitos constantemente usados como sinônimos para a aplicação efetiva e específica de um ambiente inteligente residencial. Casa inteligente pode ser definida como "um ambiente que trará conforto para àqueles que o habitam, pois terão as suas tarefas básicas de rotina reproduzidas pelo ambiente e proporcionará uma maior segurança e comodidade" (ALAM; REAZ; ALI, 2012).

Para isso, alguns equipamentos domésticos e dispositivos presentes na casa permitem, relativamente, a troca de informação entre eles sobre como os utilizadores efetuam ações sobre os tais. Além disso, esses dispositivos e equipamentos podem agir de forma independente, sem intervenção humana (COOK; AUGUSTO; JAKKULA, 2009). Alguns exemplos destes equipamentos podem ser, por exemplo, o fogão, a geladeira, as torneiras, a cama, o ar condicionado, luzes, TVs, portões, fechaduras, etc.

Estes equipamentos ao possuírem "inteligência" resultam vários benefícios para o utilizador, destacando-se: redução de custos de manutenção, segurança física e patrimonial, conforto, bem estar, meios de comunicação com os equipamentos e com outras pessoas, com eficiência e praticidade em todas as funcionalidades da casa, possibilitando o gerenciamento técnico em geral (ALAM; REAZ; ALI, 2012).

Para esse êxito, diferentes pesquisas surgem a partir da ideia de alcançar estes ser-

viços de maneira mais eficiente, íntegra e flexível, através de padrões e tecnologias para criação de novos produtos e serviços de maneira mais eficientes (ALAM; REAZ; ALI, 2012). Entretanto, a maioria das funcionalidades em residências nos dias de hoje, ainda são geradas a partir da automação residencial e existem atualmente sem qualquer inteligência (ACAMPORA et al., 2013).

O usuário controla luzes, TVs e outros eletrônicos da casa com interruptores e controles existentes. Em casas inteligentes, essas ações serão controladas através de painéis de toque e, eventualmente, pela voz, gestos, expressões faciais, etc. A casa identifica o morador e ajusta suas funções de acordo com as preferências do morador. O tipo favorito de música ou canal da TV é automaticamente ligado, certo grau de iluminação e temperatura do ambiente é utilizado, etc (ACAMPORA et al., 2013). Ainda segundo Acampora et al. (2013), o grande desafio para aplicações em casas inteligentes é decidir sobre como aplicar essas funcionalidades de forma inteligente no ambiente. Isso exerce devido às diferentes abordagens que devem ser tomadas no desenvolvimento de tecnologias que envolveram esse ambiente. Essas abordagens são discutidas por diferentes autores Chaczko et al. (2010), Acampora et al. (2013), Chen et al. (2013) e resumidas a seguir:

- **Segurança física:** Em uma casa inteligente, por exemplo, o requisito de segurança abrange o monitoramento e controle do "bem estar" da casa, ou seja, qualquer problema no meio físico é relatado ao morador e são acionadas as entidades responsáveis pelo ajuste. Como também uma quantidade satisfatória de dispositivos de segurança espalhados na casa são responsáveis por manter a segurança dos que a habitam (ACAMPORA et al., 2013).

- **Controle de acesso físico:** O acesso físico nas casas de hoje é controlado principalmente com fechaduras mecânicas e eletromecânicas. O acesso inteligente deverá ser então tratado com chaves eletrônicas, tais como cartões magnéticos, RFID, como também, o acesso biométrico, sejam esse por meio de impressão digital ou leitura de íris, ou até mesmo, uma combinação entre esses (CHEN et al., 2013).

- **Gestão do bem-estar:** Saúde, bem-estar e suporte para uma vida independente parecem aplicações muito promissoras para a casa inteligente (CHACZKO et al., 2010). Esta pode ser vista como parte também da segurança física ou pode ser tratada como uma função separada (ACAMPORA et al., 2013). Antes de mais nada, há uma clara necessidade de fornecer ferramentas para apoiar a vida independente, especialmente para os idosos e crianças (monitoramento de saúde, deficiências funcionais, proteção contra acesso a objetos perigosos, etc). Assim, existe uma necessidade de vários meios

técnicos auxiliares simultaneamente, dando potenciais vantagens para a integração do usuário na mesma plataforma (CHACZKO et al., 2010; ACAMPORA et al., 2013).

- **Comunicação:** A casa é um espaço coberto por diferentes tipos de meio de comunicação espalhados em diferentes âmbitos residenciais. A casa inteligente abstrairá toda essa comunicação e deixará disponível em qualquer local da casa onde o usuário esteja, facilitando o acesso rápido e seguro por necessidades emergenciais (ACAMPORA et al., 2013).

- **Acesso à Internet:** Observa-se hoje em dia que muitos dispositivos residenciais disponibilizam de acesso à internet. Contudo, um gasto considerável para a adaptação e configuração desses dispositivos deverá ser realizado. Em casas inteligentes, essa configuração será abstraída pela própria casa. Igualmente ao acesso à comunicação, será disponibilizada no momento que for solicitada, independente do local que esteja (CHACZKO et al., 2010; CHEN et al., 2013). Como também, a partir dessa característica, o usuário poderá ter acesso a sua residência de onde ele esteja, podendo realizar ações manuais na casa (CHEN et al., 2013).

- **O trabalho doméstico:** Este abrange limpeza, trabalho de lavanderia, cozinha, preparação de refeições, etc - todas as atividades básicas de manutenção da casa como um lugar confortável para se viver. Por exemplo, a limpeza de uma casa de forma eficiente pode ser facilitada a partir da utilização de produtos de limpeza mais eficientes a vácuo, pois, eventualmente os robôs de limpeza conseguiram discernir pequenos itens no chão são objetos que devem ir para o lixo ou objetos de valor que caíram pelo chão, como um anel (CHEN et al., 2013).

- **Diversão:** Diversão é indispensável para qualquer tipo de idade, e principalmente para as crianças. Entretanto, nos dias de hoje, jogos computacionais e vídeo games estão sendo preferências entre as crianças. A proporção do aumento do intelecto das crianças com esses jogos está diretamente relacionado, com a proporção da falta de atenção e ingenuidade delas, se não houver um controle sobre sua utilização (CHEN et al., 2013). Com isso, a casa inteligente, com regras definidas pelos pais, colocará limites a essa diversão (jogos computadorizados), propondo diferentes formas de diversão e focando em jogos educacionais e de aprendizado (CHACZKO et al., 2010; CHEN et al., 2013).

- **Privacidade de dados:** Grande parte das ações tomadas em um ambiente é baseada nas informações consequentes de monitoramento do mesmo. O vazamento destas

informações acarreta risco à segurança dos moradores (ACAMPORA et al., 2013).

- **Confiança e Aceitação:** Há uma descrença quanto à aceitação de novas tecnologias em ambiente residencial. Em alguns casos, nem tudo que é tecnologicamente novo é aceito, ou, a capacidade de adquirir tecnologia tem um preço financeiro alto, ou ainda, na maioria dos casos, ter a tecnologia, não necessariamente é confiar nela (CHEN et al., 2013).

Essas funcionalidades são aplicadas nos diversos dispositivos que estarão presentes no ambiente (CHEN et al., 2013). Em outras palavras, cada dispositivo ou grupos de dispositivos, exercerão responsabilidades e funções específicas no ambiente. Contudo, o compartilhamento das informações coletadas com outros dispositivos, como também, unidos a um processamento central baseado em inteligência artificial (REAZ, 2013), põem em prática o conceito de casa inteligente, como pode ser observado na Figura 2.

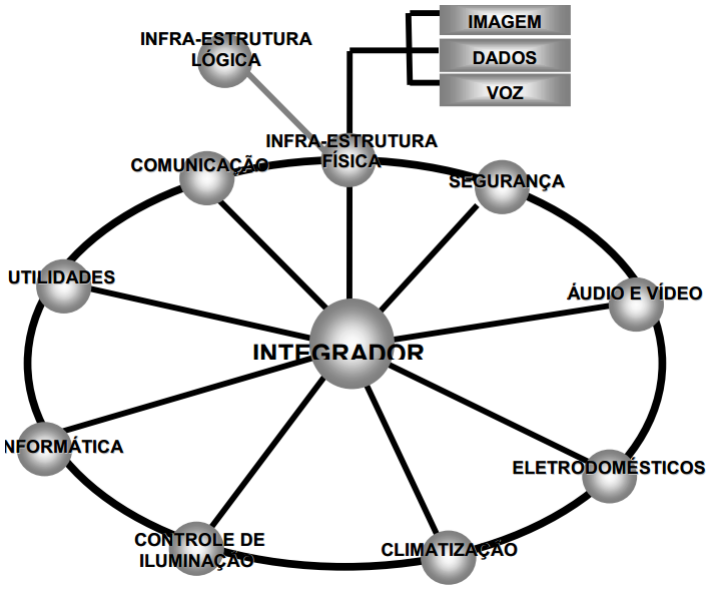


Figura 2: Integração das funcionalidade de dispositivos em casas inteligentes
Fonte: (REAZ, 2013)

Um bom exemplo dessa integração pode ser compreendido quando os sensores de um sistema de vigilância captam uma ameaça de intruso na residência e acionam, imediatamente, a iluminação de alguns ambientes. Ao mesmo tempo, o sistema de vídeo aciona a gravação, assim como, apresenta imagem do invasor, em tempo real, num canto da tela da TV. Persistindo a invasão, o sistema providencia a abertura da porta do canil para libertar os cães, trava as portas de acesso da residência e executa, ainda, ligação telefônica para a polícia (D.; P., 2010).

Por fim, o desenvolvimento de sistemas de se concentra em como o ambiente e suas tecnologias, produtos e serviços devem evoluir para atender melhor os moradores nela presentes, sendo as possibilidades desse atendimento infinitas. A seguir são exibidos mais dois cenários, adaptados de (DUCATEL et al., 2005), que exibem o potencial dessa abordagem em termos de comodidade e conforto.

- **Cenário 1:** Ao se aproximar de sua residência, a casa, reconhecendo-o a partir de uma mensagem recebida pelo seu carro inteligente, ou seu celular inteligente, começa a tomar uma sequência inteligente de determinadas ações. Por exemplo, a casa após desativar o seu sistema de alarme, começa a iluminar alguns vãos e abre o portão da garagem sozinha. Em seguida, ela ajusta o aquecimento ou o esfriamento da casa ao gosto do morador e liga o seu sistema de áudio, tocando aquele CD favorito. Ao sentar em seu sofá para assistir um determinado filme ou seriado de TV, a casa diminui a potência das luzes do ambiente automaticamente e desliga qualquer meio de comunicação se o usuário não gostar de ser interrompido naquele momento.

O sistema de controle da casas inteligentes é programado para atender às necessidades específicas do usuário, iniciando operações automáticas, sequenciais e seguras, tudo isso em resposta simplesmente a presença do usuário.

- **Cenário 2:** São 6:30 da manhã de uma segunda feira e o usuário acorda ao som suave daquelas músicas que sempre gosta de escutar todas as manhãs. As luzes em seu quarto ligam de forma lenta, permitindo ao usuário acordar no seu próprio tempo, e ao mesmo tempo o sistema de alarme interior da casa é desativado. Na cozinha, a máquina de café já está terminando de preparar o mesmo, enquanto o usuário termina de tomar. A casa abre as persianas e janelas da casa, sabendo que o usuário gosta da luz do dia enquanto toma seu café.

Este outro exemplo demonstra como a tecnologia d casa inteligente vai mudar a vida das pessoas, projetando sistemas que agrupam a automatização das tarefas diárias e simples do dia a dia, melhorando a qualidade de vida e reduzindo o estresse.

2.3.1 Dispositivos e protocolos de Conectividade

A redução no custo e tamanho dos circuitos integrados deu origem a dispositivos inteligentes cada vez menores e complexos (KIM et al., 2012). Alguns desses dispositivos

podem trocar informações entre outros dispositivos próximos, ou até mesmo pela internet (KIM et al., 2012). Esses, ainda podem dar suporte para novos tipos de serviços, como serviço de monitoramento de segurança, reconhecimento facial, biblioteca digital de áudio e vídeo, dispositivos com sensores de presença e localização, etc.

Um ambiente inteligente é um domínio físico composto de dispositivos (hardware e sofware) capazes de processarem informações, também denominado *smartdevice*. Eles são os principais responsáveis por tornar o ambiente inteligente, tendo eles, a capacidade de comunicarem-se uns com os outros (REAZ, 2013). Qualquer ambiente pode ser considerado inteligente se possuir vários dispositivos interagindo entre si. Entretanto, para determinados ambientes, como casas inteligentes, esses dispositivos devem permanecer em um ambiente "fisicamente fechado" e interagir somente com os membros nela presentes.

Além disso, dispositivos tradicionais, como uma geladeira, DVD, TV, câmera de vídeo, console de jogos, luzes, porta da garagem, fechaduras, forno micro ondas, etc, também podem ser inteligentes, se forem atualizados com os devidos componentes, ou, simplesmente substituídos por outros já adquiridos dessa tecnologia (COOK; AUGUSTO; JAKKULA, 2009). Outros dispositivos que não são fixos em uma casa, como dispositivos pessoais, não são impedidos de fazerem parte de uma ambiente inteligente, ou melhor, eles em alguns casos, são as chaves principais para que o ambiente se adapte ao contexto do usuário, no entanto, esses dispositivos devem interagir com os outros presentes no ambiente. Sendo essa interação entre os dispositivos, de forma geral, o principal problema desse contexto (COOK; AUGUSTO; JAKKULA, 2009; KIM et al., 2012; REAZ, 2013).

Nos ambientes inteligentes, diferentes camadas de operação, meio físico, redes domésticas e protocolos de comunicação permitem a intercomunicação destes dispositivos (THINAGARAN; ABDUL; CHUI, 2008). Tecnologias de rede com fio e sem fio, IrDA (*Infrared Data Association*) e fibra óptica são comumente usados nesse ambiente. Por outro lado, os responsáveis pela intercomunicação entre dispositivos no ambiente, também chamados de protocolos de conectividade, desempenham o papel principal da comunicação entre os sistemas dos dispositivos presentes no ambiente (KIM et al., 2012).

Estes meios físicos, dispositivos e serviços configurados em uma ambiente estão revolucionando o mundo em que vivemos, resultando em alguns problemas operacionais (THINAGARAN; ABDUL; CHUI, 2008). O primeiro problema é que para atingir diferentes funcionalidades de um ambiente inteligente, há uma dependência maior na heterogeneidade de dispositivos construídos com diferentes especificações (THINAGARAN; ABDUL; CHUI, 2008). O segundo é com relação a complexidade, esse o fator de intercomunicação

entre os diferentes tipos de dispositivos (THINAGARAN; ABDUL; CHUI, 2008; BREGMAN; KORMAN, 2009; KIM et al., 2012).

A Interoperabilidade para esses dispositivos envolve não apenas fornecer a conectividade entre outros dispositivos, mas também coagirem entre si para executarem tarefa e saber o que os outros dispositivos estão fazendo ou executando (REAZ, 2013). A dificuldade principal para alcançar a interoperabilidade em ambientes de casa inteligente é ocasionada pela falta de padronização de transmissão de dados dos protocolos de conectividades desenvolvidos para esse âmbito (REAZ, 2013; C. YANN B., 2013). Este fator influenciou diretamente pesquisas a desenvolverem diversos protocolos capazes de alcançar várias funcionalidades, com a interação destes com outros dispositivos, utilizando o mesmo protocolo.

Nesse contexto, a intercomunicação de dispositivos significa a capacidade dos sistemas, aplicações e serviços trabalharem em conjunto de forma confiável e previsível. Tendo como principal objetivo garantir que os dispositivos interajam sem danificar ou modificar o meio físico da conexão/comunicação que estejam utilizando (THINAGARAN; ABDUL; CHUI, 2008). Ainda segundo Thinagaran, Abdul e Chui (2008), qualquer protocolo de conectividade presente nos dispositivos em uma ambiente que deva ser inteligente, ou se comportar como um, deve seguir um modelo genérico, adaptando-se às tecnologias já desenvolvidas e utilizadas na rede física de um ambiente. Esse modelo é baseado no modelo OSI e é divido em três camadas, como ilustrado na Figura 3.

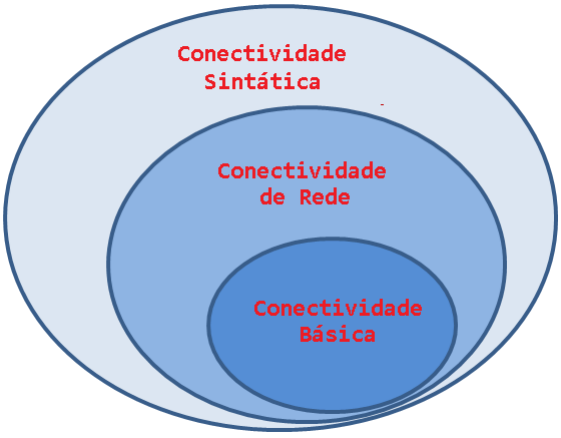


Figura 3: Camadas para interoperabilidade de protocolos com o meio de transmissão
Fonte: (THINAGARAN; ABDUL; CHUI, 2008)

- **Conectividade básica:** Nível de conectividade básica é representado pelas camadas físicas e de enlace de dados do modelo OSI (*Open Systems Interconnection*):

Ethernet; WI-Fi; e PPP (*Point-to-Point Protocol*), são exemplos de normas comuns para conectividade básica do protocolo (THINAGARAN; ABDUL; CHUI, 2008).

- **Conectividade de rede:** Essa camada permite a troca de mensagens entre sistemas através de uma variedade de tipos de redes. Essa camada é representada pelas camadas de rede, transporte, sessão e aplicação do modelo OSI. Exemplos de normas comuns de conectividade de rede são os TCP (*Transmission Control Protocol*); UDP (*User Datagram Protocol*); FTP (*File Transfer Protocol*); ARP (*Address Resolution Protocol*) e IP (*Internet Protocol*); (THINAGARAN; ABDUL; CHUI, 2008).

- **Conectividade sintática:** A camada final refere-se ao acordo e regras que controlam o formato bem como a estrutura de codificação de trocas de informações entre os dispositivos. Conectividade sintática refere-se à capacidade de dois ou mais componentes trabalharem em conjunto com os dados e mensagens transmitidos entre eles. Em outras palavras, essa camada final fornece um mecanismo de compreensão de estrutura de dados das mensagens trocadas entre duas entidades (dispositivos). Essa é representada pelas camadas de aplicação apresentação do modelo OSI. Algumas das funções fornecidas por esta camada são estruturas de codificação de mensagens, como o SOAP (*Simple Object Access Protocol*) e o REST (*Representational State Transfer*). Essa camada é fundamental para garantir uma transição suave entre mensagem dos dispositivos heterogêneos em ambientes inteligentes (THINAGARAN; ABDUL; CHUI, 2008).

Baseando neste modelo oferecido por Thinagaran, Abdul e Chui (2008), qualquer protocolo de conectividade está apto a integrar-se a um dispositivo e participar de uma infraestrutura de rede doméstica, disponibilizando assim seus serviços e suas funções. Contudo, não é somente seguindo esse modelo que teremos a interoperabilidade entre diferentes protocolos de conectividade. Como abordado anteriormente, o problema principal está na falta de uma padronização nesse tipo de protocolo de conectividade (REAZ, 2013; C. YANN B., 2013). Com isso, uma diversidade heterogênea de protocolos ainda será desenvolvida problematizando e incapacitando a interoperabilidade de diferentes dispositivos.

2.4 Protocolos de Conectividade

Um dispositivo ao ser incorporado em uma ambiente deverá ser capaz de integrar-se automaticamente ao ambiente e descobrir a presença de outros dispositivos e suas

funcionalidades. Protocolos de conectividade são os responsáveis por esse êxito (KAED et al., 2011). Essa capacidade de descoberta em tempo de execução possibilita a configuração dinâmica dos dispositivos. (KAED et al., 2011).

Nesse capítulo serão discutidas as características relevantes dos protocolos de conectividade mais abordadas nos trabalhos retornados e aceitos da revisão sistemática, e que fazem parte diretamente da proposta desse trabalho. Valendo ressaltar que, todas as informações abordadas a nas próximas seções foram coletadas das especificações de cada protocolo, disponibilizadas em sua web sites:. UPnP em `http://www.upnp.org/` e DLNA em `http://www.dlna.org/`.

2.4.1 Universal Plung and Play (UPnP)

Lançado pelo Forum UPnP, um grupo de empresas e pessoas físicas em todo o setor que pretendem desempenhar um papel de liderança na autoria de especificações para dispositivos e serviços UPnP. Criado em 1999, o UPnP é um padrão proposto pela Microsoft[2] em conjunto com outras 28 empresas, incluindo Intel[3], HP[4], Compaq[5] e Samsung [6]. Atualmente, mais de 800 empresas fazem parte do UPnP Forum, o qual objetiva definir e publicar padrões, protocolos e modelos, permitindo a interoperabilidade entre diferentes dispositivos e serviços de rede em ambientes domésticos e corporativos.

Com a adição de dispositivos *Plug and Play* (PnP), muitos recursos para o sistema operacional tornaram-se mais fáceis de gerenciar, configurar e adicionar periféricos a um PC (*Pernosal Computer*). O UPnP estende essa simplicidade para toda a rede, permitindo, assim, a descoberta e o controle de dispositivos e seus serviços. Isso inclui dispositivos e serviços de rede, como impressoras conectadas à rede, gateways da Internet e equipamentos eletrônicos de consumo.

O UPnP é mais do que apenas uma simples extensão do modelo periférico PnP. Ele é projetado para dar suporte à configuração zero na adição de um novo dispositivo na rede, ser "invisível", dar suporte a descoberta automática de dispositivos, entre outras características. Com o UPnP, um dispositivo pode dinamicamente participar de uma rede, obter um endereço IP, transmitir as suas características e descobrir sobre as características de outros dispositivos automaticamente. Os dispositivos podem ainda se comunicarem

[2]http://www.microsoft.com
[3]http://www.intel.com
[4]http://www.hp.com
[5]http://www.compaq.com
[6]http://www.samsung.com

uns com os outros diretamente sem a necessidade de um dispositivo de gerenciamento, permitindo assim redes *Peer-to-Peer* (P2P) .

As variedades de tipos de dispositivos que podem se beneficiar utilizando protocolos UPnP são grandes e incluem eletrodomésticos, diferentes tipos de sensores dispositivos sem fio (câmeras de vigilância) e outros dispositivos pessoais de todas as formas. O escopo do UPnP é grande o suficiente para abranger diferentes tipos de ambientes, bem como cenários novos e emocionantes, incluindo casas inteligentes, escritórios inteligentes, diferentes tipos de impressão e imagem, entretenimento utilizando áudio/vídeo, aparelhos de cozinha, redes para automóveis e redes de proximidade em locais públicos.

2.4.1.1 Componentes de uma rede UPnP

Existem em uma rede UPnP três componentes básicos: Dispositivo (*Device*), Serviços (*Service*) e Ponto de controle (*Control Point*). Dispositivos são as entidades físicas da rede que disponibilizam serviços no ambiente, podendo ainda conter outros dispositivos em sua estrutura. Serviços são relacionados diretamente as ações que um dispositivo está disponibilizando. Por fim, Pontos de controle são uma classe de dispositivos capazes de controlar ou usufruir dos serviços disponibilizados pelos dispositivos na rede. Como observado na Figura 4.

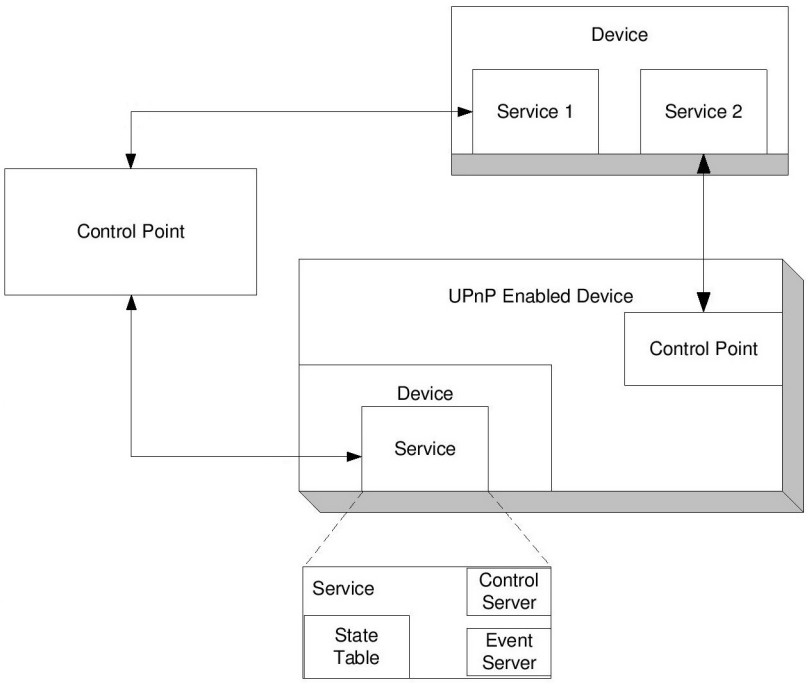


Figura 4: Ponto de Controle UPnP, Dispositivo e Serviços
Fonte: http://www.upnp.org/

- **Dispositivos UPnP:** Um dispositivo UPnP é um recipiente que disponibiliza serviços em uma rede, como também, pode compor outros dispositivos incorporados a ele (*embedded devices*). Cada dispositivo é caracterizado de acordo com a semelhança dos serviços disponibilizados, por exemplo, um dispositivo do tipo relógio pode ser composto pelos serviços *data-hora* e um *cronômetro*. Outro dispositivo denominado rádio-relógio pode ser composto pelos serviços *frequência*, *data-hora* e *cronômetro*, ou, esse mesmo dispositivo pode ser comporto somente do serviço *frequência* e um dispositivo relógio (*embedded device*). Consequentemente, diferentes fornecedores irão padronizar o conjunto de serviços que um determinado tipo de dispositivo irá proporcionar. Todo dispositivo irá fornecer essas informações na rede a partir de um formato específico contendo uma descrição detalhada do dispositivo e serviços disponibilizados.

- **Serviços UPnP:** Esta é considerada a menor unidade de controle em uma rede UPnP.Um serviço expõe ações e modela seu estado com variáveis de estado. Por exemplo, um serviço de relógio pode ser modelado como tendo uma variável de estado, *current-time*, que define o estado do relógio, e duas ações, *set-time* e *get-time*, que permitem o controle do serviço disponibilizado. Semelhante à descrição do dispositivo, estas informações fazem parte de uma descrição específica, padronizado pelo Fórum UPnP e recuperado a partir de um URL definida na descrição dos serviços disponibilizado pelo dispositivo. Um serviço em um dispositivo UPnP consiste em:

  - **State Table:** Cada serviço dispõem de uma tabela de estados que modela o estado atual das variáveis do serviço do dispositivo;
  - **Control Server:** Um servidor de controle é responsável por receber solicitações de ação (*set-time* ou *get-time*), executa-los, atualiza a tabela de estados quando necessário e retorna respostas para quem solicitou essa ação, como também, para o ambiente informando a mudança no valor da variável de estado para os interessados.
  - **Event Server:** Um Servidor de eventos é responsável por publicar eventos para dispositivos interessados sempre que um serviço é consumido e que uma mudança no valor da variável de estado do serviço é modificado. Por exemplo, o serviço de alarme de incêndio iria enviar um evento para dispositivos interessados quando seu valor da variável de estado da ação *set-fire* mudar de *false* para *true*.

- **Ponto de Controle UPnP:** Um ponto de controle em uma rede UPnP é um controlador capaz de descobrir e monitorar outros dispositivos. Após a descoberta, um ponto de controle pode:
  - Recuperar a descrição do dispositivo e obter uma lista dos serviços disponibilizados;
  - Recuperar as descrições de serviço para serviços interessantes;
  - Invocar ações para consumir o serviço.

2.4.1.2 Pilha de Protocolos UPnP

Por ser um protocolo de rede, o protocolo UPnP utiliza um conjunto de outros protocolos padronizados já existentes em uma rede, garantindo assim, a interoperabilidade entre implementações de diferentes fornecedores. Uma vez que os mesmos protocolos já estão em uso, pouco precisaria ser feito para tornar os dispositivos UPnP compatível com o ambiente. Pode-se observar na Figura 5 a pilha de protocolos utilizados pelo UPnP.

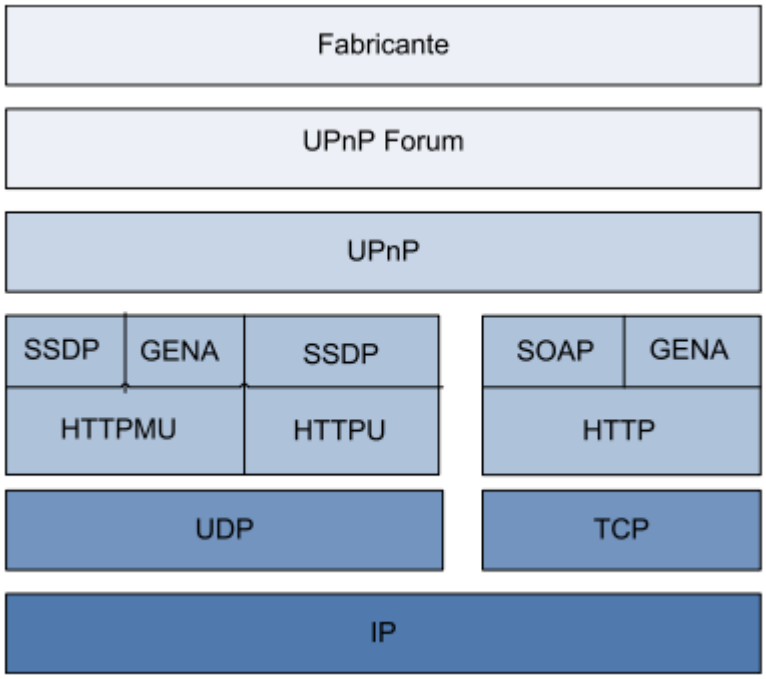


Figura 5: Pilha de Protocolos UPnP
Fonte: http://www.upnp.org/

- **Fabricante, UPnP Fórum e UPnP:** Com base na arquitetura do dispositivo, essas camadas definem a especificação do dispositivo, tais como, os tipos de TVs,

sistemas de climatização, máquinas de lavar louça e roupa, luzes, câmeras e outros aparelhos e dispositivos.

- **SSDP:** O protocolo SSDP (*Simple Service Discovery*) define como os serviços podem ser descobertos na rede. SSDP é construído sobre HTTPU e HTTPMU e define métodos tanto para um ponto de controle localizar recursos de interesse na rede quanto para os dispositivos anunciarem a sua disponibilidade na rede.

- **GENA:** *General Event Notification Architecture* (GENA) é responsável pela capacidade de enviar e receber notificações de dispositivos e Pontos de Controle utilizando HTTP (*Hypertext Transfer Protocol*) sobre TCP/IP, UDP e *multicast*. GENA também define os conceitos de assinantes e editores de notificações para ativar eventos. Formatos GENA são utilizados em UPnP para criar os anúncios de presença a serem enviados usando o protocolo SSDP e para fornecer a capacidade de sinalizar mudanças no valor das variáveis de estado do serviço.

- **SOAP:** O protocolo SOAP (*Simple Object Access Protocol*) define o uso de *Extensible Markup Language* (XML) e HTTP para executar chamadas de procedimento remoto (RPC -*Remote Procedure Call*). O UPnP usa SOAP para entregar mensagens de controle aos dispositivos do ambiente e devolver resultados ou erros aos Pontos de controle. Cada solicitação de controle é uma mensagem SOAP contendo a ação de chamada junto com um conjunto de parâmetros solicitados. A resposta dessa solicitação também é uma mensagem SOAP contendo o status, valores das variáveis e todos os parâmetros de retorno solicitados;

  - **XML:** O UPnP usa a linguagem de marcação XML por ser o formato universal para dados estruturados na Web. COm outras palavras, o XML é uma maneira de colocar qualquer tipo de dados estruturados em um arquivo de texto. XML se parece muito com HTML na utilização de *tags* e atributos. Entretanto, estas não são globalmente definidos quanto ao seu significado, mas são interpretados dentro do contexto de seu uso. Essas características do XML torná-o um bom ajuste para o desenvolvimento de esquemas para vários tipos de documentos. XML é uma parte essencial do UPnP usado em descrições de dispositivos e serviços, mensagens de controle e eventos.

- **HTTP, HTTPU, HTTPMU:** O protocolo HTTP, que é extremamente responsável pelo sucesso da Internet, também é uma parte essencial do UPnP. Todos os aspectos do UPnP é construído em cima do HTTP e suas variantes. HTTPU é uma

extensão do HTTP usando UDP como o transporte em vez dos habituais dados TCP. HTTPMU é uma variante do HTTPU que usa IP *multicast*.

- **TCP/IP:** O protocolo de rede TCP/IP serve como a base na qual o resto dos protocolos UPnP são construídos. Usando o padrão predominante conjunto de protocolos TCP/IP, UPnP aproveita a capacidade do protocolo para abranger diferentes meios físicos e garante a interoperabilidade de vários fornecedores. Dispositivos UPnP podem usar muitos dos protocolos da pilha TCP/IP, incluindo TCP, UDP, IGMP, ARP e IP, bem como serviços do TCP/IP, como DHCP e DNS.

A Arquitetura de dispositivos UPnP define um esquema ou modelo padronizado para a geração da descrições dos dispositivos e dos serviços. Posteriormente, os fabricantes de dispositivos UPnP definem somente algumas informações específicas para os seus dispositivos, tais como o nome do mesmo, número do modelo, nome do fabricante e URL para a descrição do fabricante. Estas informações são encapsuladas nos protocolos específicos de UPnP e inseridas em todas as mensagens antes de serem formatadas utilizando SSDP, GENA e SOAP e entregues via HTTP, HTTPU ou HTTPMU.

2.4.1.3 Etapas de Conectividade UPnP

O mecanismo de conectividade abordado pelo UPnP ao conectar-se em uma rede é dividido em seis etapas distintas: endereçamento, descoberta, descrição, controle, evento e apresentação, como observado na Figura 6. Todas as etapas utilizam protocolos padronizados e amplamente utilizados e difundidos em redes locais e na Internet, abordados na seção anterior.

2.4.1.4 Endereçamento

Quando dispositivo é conectado pela primeira vez à rede, se um servidor DHCP (*Dynamic Host Configuration Protocol*) estiver disponível, o dispositivo deve usar o endereço IP atribuído a ele. Se nenhum servidor DHCP estiver disponível, esse deve usar a técnica AutoIP para obter um endereço IP na rede. Em resumo, o AutoIP define como um dispositivo escolhe, de forma inteligente, um endereço IP a partir de um conjunto de endereços privados reservados, e é capaz de mover-se facilmente entre redes gerenciadas e não gerenciadas. A etapa de endereçamento independe da interação entre outros dispositivos e Pontos de Controle, essa é destinada diretamente conexão do dispositivo a rede.

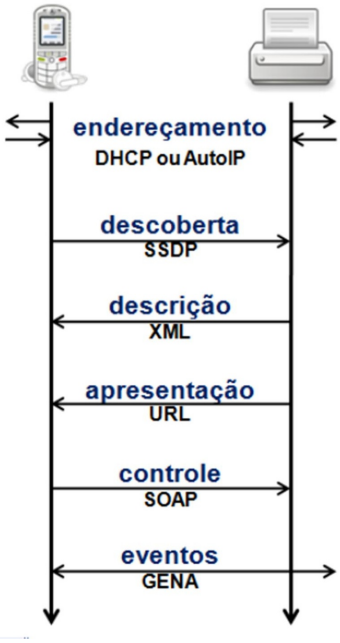


Figura 6: Etapas para conectividade UPnP
Fonte: Adaptado de http://www.upnp.org/

2.4.1.5 Descoberta

Uma vez que os dispositivos são conectados à rede, a sua descoberta estará disponível para outro dispositivo, como também, o mesmo poderá descobrir a presença do outros dispositivos. Quando um dispositivo é adicionado à rede, o SSDP permite que o dispositivo possa disponibilizar seus serviços para Pontos de controle ou para outros dispositivos interessados. Da mesma forma, quando um Ponto de controle é adicionado à rede, também é o SSDP que permite que ele possa descobrir a existência de dispositivos.

A troca fundamental para ambos os casos é uma mensagem de descoberta contendo algumas especificidades essenciais sobre o dispositivo, por exemplo, o seu tipo de dispositivo, identificador e um endereço para seu documento de descrição de dispositivo disponível em XML. Basicamente, quando um dispositivo se conecta à rede, este envia mensagens de descoberta, do tipo *multicast*, para o endereço e porta **239.255.255.250:1900** anunciando a sua presença na rede. Diferentes dos Pontos de Controle ao ser integrado na rede, esse pode esperar alguma notificação de dispositivo informando que está presente na rede, ou, esse pode enviar uma mensagens de solicitação de dispositivo, caso o dispositivo receba uma mensagem desse tipo, ele enviará para o destinatário da mensagem, uma mensagem *unicast* informando sua presença.

2.4.1.6 Descrição

Depois de um Ponto de Controle descobrir um dispositivo na rede, as informações adquiridas nesse processo não são suficientes para a utilização dos serviços disponibilizados pelo mesmo. Para que este obtenha mais informações sobre o dispositivo e as suas características, ou, para interagir com o mesmo, o Ponto de Controle deve solicitar uma descrição detalhada dispositivo a partir do URL fornecida pelo dispositivo na mensagem de descoberta. A descrição UPnP para um dispositivo é expressa em XML e inclui informações específica do fabricante, como o nome do modelo, número de série, nome do fabricante, URL do fornecedor, etc. A descrição também inclui uma lista de quaisquer dispositivos ou serviços incorporados, bem como, URLs para seu controle, eventos e apresentação do mesmo. Apresenta-se resumidamente na Figura 7 uma descrição de um dispositivo relógio UPnP.

```
<rootxmlns = urn:schemasupnporg:device10 ">
    <device>
        <deviceType>...:device:Relogio:1</deviceType>
        <friendlyName>RelógioGenérico</friendlyName>
        <manufacturer>Oriente</manufacturer>
        <manufacturerURL>...<manufacturerURL>
                <modelDescription>RelógioOrientenovomodelo.</modelDescription>
                <modelNumber>12345</modelNumber>
                <UDN>identificaçãoÚnica</UDN>
                <!—outras informações--!>
                <serviceList>
                    <service>
                        <serviceType>urn:schemasupnporg:service:Time:1</serviceType>
                        <serviceId>urn:upnporg:serviceId:11</serviceId>
                        <SCPDURL>/Clock/clocktime.xml</SCPDURL>
                        <controlURL>/Clock/clocktime/control</controlURL>
                        <eventURL>/Clock/clocktime/event</eventURL>
                    </service>
                    <service>
                </serviceList>
                <presentationURL>URL-Apresentação</presentationURL>
    </device>
</root>
```

Figura 7: Descrição de um dispositivo UPnP
Fonte: Autor

2.4.1.7 Controle

Depois que um Ponto de Controle obteve uma descrição detalhada do dispositivo, ele ainda não tem o essencial para controlá-lo. Para isso, o Ponto de Controle ainda necessita das informações sobre o serviço disponibilizado pelo mesmo. Um Ponto de Controle deve obter também, uma descrição detalhada para cada serviço disponibilizado, sendo essa descrição também expressa em XML. O Ponto de Controle faz uso dos serviços através de chamadas de métodos remotos via *Web Service* baseado no protocolo SOAP. O serviço

responde essa informação passando os parâmetros para cada ação, como também, a descrição de uma lista de variáveis onde o Ponto de Controle possa monitorar o seu estado em tempo de execução.

2.4.1.8 Eventos UPnP

O serviço UPnP publica atualizações quando alguma variável de estado de um serviço é alterado, e um Ponto de Controle pode se inscrever para receber essas informações. Por exemplo, uma chamada remota ao serviço *Play* da especificação do serviço UPnP de áudio e vídeo modificaria o estado da variável de *pause* para *reproduzindo*. Através de mensagens de evento realizadas pelos serviços, um outro Ponto de Controle ou dispositivo pode ser informado sobre estas mudanças no valor de sua variável, e com isso, atualizar as variáveis locais correlacionadas a este novo estado. Estas mensagens também são expressas em XML e formatado usando GENA.

2.4.1.9 Apresentação

Se um dispositivo tiver um URL de apresentação, então o Ponto de Controle pode recuperar uma página dessa URL, carregá-la em um navegador e, dependendo dos recursos oferecidos por essa apresentação, permitir que um usuário controle o dispositivo. O grau de realização dessa etapa depende dos recursos específicos da página de apresentação e do dispositivo.

2.4.2 Digital Living Network Alliance (DLNA)

O protocolo DLNA é uma organização constituída de diferentes empresas com a finalidade de estabelecer critérios permitindo que, empresas de dispositivo multimídias possam ser capazes de se conectarem nas redes domésticas. Outro objetivo principal do DLNA é o compartilhamento de informações digitais de imagem, áudio e vídeo, entre os dispositivos DLNA presentes na rede, como computadores pessoais, eletrônicos de consumo e dispositivos móveis. Com outras palavras, o usuário seria capaz de acessar e reproduzir seus arquivos de mídia de um computador, por exemplo, através de uma TV, *tablet*, *smartphone*, entre outros, desde que esses se encontrem conectados em uma mesma rede.

Fundada pela Sony[7] em junho de 2003, em conjunto com um grupo de empresas, necessitou em criar um protocolo, não novo, mas um baseada nos padrões já existentes e

[7] https://www.sony.com

utilizados, com o objetivo que diferentes tipos de dispositivos possam interagir em uma mesma rede, mesmo que esses dispositivos sejam desenvolvidos por diferentes fabricantes. Com o passar do tempo, o protocolo foi evoluindo e alcançando um número cada vez maior de colaborados. Tendo em vista, nos dias de hoje, mais de 240 membros unidos à associação DLNA, entre elas, estão empresas como: AT& T[8], Cisco Systems[9], Intel[10], LG[11], Samsung [12], entre outros.

A colaboração industrial da Aliança envolve os principais líderes industriais de eletrônicos de consumo. No entanto, essa colaboração não foi limitada a apenas esse tipo de fabricantes. Empresas de tecnologia PC, *notebook*, dispositivos móveis, também estavam interessadas nessa tecnologia. Com isso, essa aliança oferecem aos consumidores um amplo conjunto de produtos e serviços complementares. Como pode-se observar na descrição do seguinte cenário ilustrado pela Figura 8:

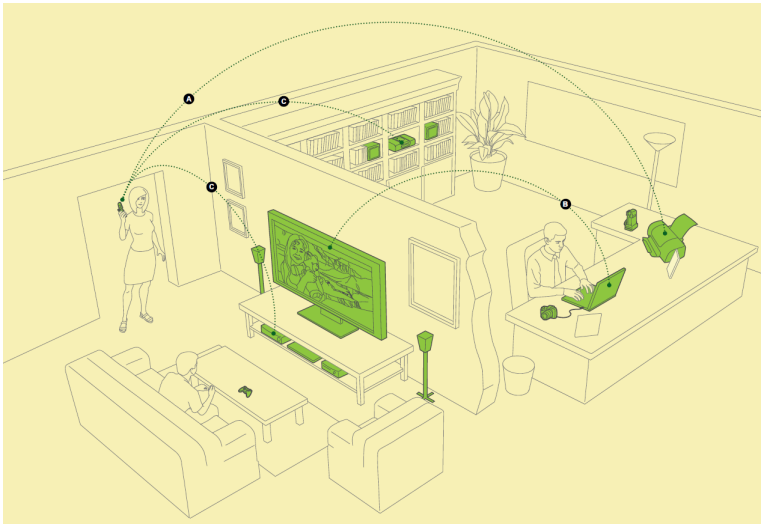


Figura 8: Ambiente DLNA
Fonte: http://www.dlna.org/

É preciso imaginar que ao término de uma viajem que foram tiradas dezenas de fotos em alta resolução, ao chegar a casa, o usuário transfira todas as fotos para seu computador tendo TV de LCD enorme na sala para exibir as fotos para a família ver. Atualmente, o usuário pegaria seu PC/Notebook, carregaria até a sala, colocaria em cima de algum objeto que deverá necessariamente está perto da TV, conectar o cabo atrás dela e exibir todas as fotos para a família. De certa forma aparenta ser um pouco cansativo,

[8]https://www.att.com
[9]https://www.cisco.com
[10]https://www.intel.com
[11]https://www.intel.com
[12]https://www.samsung.com

principalmente para leigos em tecnologia. Com a tecnologia DLNA, ao ligar uma TV, o usuário, através da rede Wi-Fi, por exemplo, acessa a pasta do seu computador, câmera digital ou celular, e terá acesso a todas as fotos para exibir para a família. É exatamente essa a proposta do DLNA. Para isso, bastam ter dispositivos certificados pela DLNA além de uma rede doméstica com ou sem fios. Podendo essa abordagem ir mais além, ou seja, o usuário ter acesso a esses dispositivo externamente de sua rede e realizar essas operações da mesma maneira que internamente.

2.4.2.1 Pilha de Protocolos DLNA

O DLNA não introduz e especifica um novo protocolo, em vez disso, são construídos sobre uma coleção de protocolos já existentes, adaptando a sintaxe de alguns deles ao seu objetivo. As diretrizes de interoperabilidade do DLNA são baseadas em uma arquitetura que define componentes interoperáveis para dispositivos e infraestrutura de software. Abrange mídia física, transportes de rede, de descoberta e controle de dispositivo, de gestão e de controle de mídia, formatos de mídia, protocolos de transporte de mídia, e interface remota. A Figura 9 ilustra os componentes funcionais abordados pela pilha DLNA.

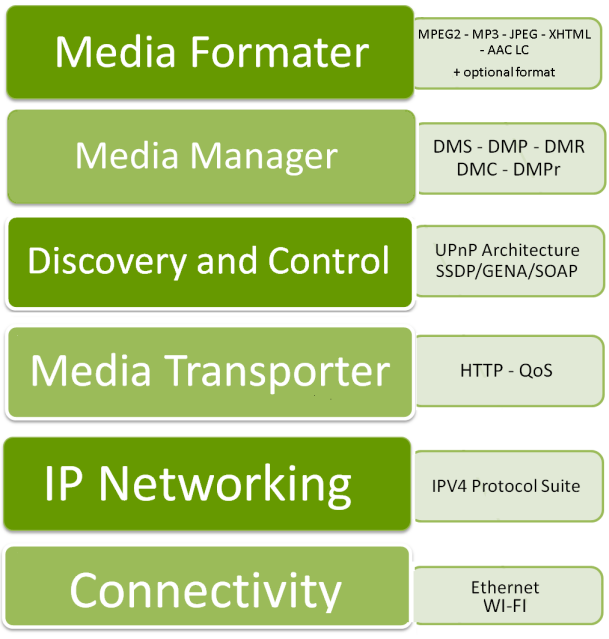


Figura 9: Pilha de Protocolos DLNA
Fonte: Adaptado de http://www.dlna.org/

- **Connectivity:** Essa camada é responsável pela conectividade de rede e aborda a

solução de conexão física. A rede física pode ser com fios ou sem fios e aborda a forma como os dispositivos se comunicam uns com os outros.

- **IP Networking:** Aborda a forma como os dispositivos se comunicam uns com os outros. O IPv4 *Protocol Suite* é aplicado na arquitetura DLNA, como também os protocolos, TCP, UDP, ICMP (*Internet Control Message Protocol*) e ARP.

- **Media Transporter:** Delimita como o conteúdo digital do dispositivo é transmitido. De acordo com a prática atual da internet, onde a maioria do tráfego digital é realizado atualmente através de HTTP sobre o TCP, a arquitetura DLNA baseasse nessas tecnologias para transferir conteúdo da origem do dispositivo para o destinatário. Além de HTTP, a arquitetura DLNA também usa RTP (*Real -Time Transport Protocol*) sobre UDP e RTSP (*Real Time Streaming Protocol*) para o controle de sessão.

- **Discovery and Control:** Para a descoberta de dispositivo e controle dos seus serviços, o DLNA utiliza em alguns protocolos específicos da arquitetura UPnP: SSDP, GENA e SOAP. Entretanto, os dispositivos DLNA reconhecem no ambiente somente outro dispositivo DLNA. Da mesma forma que o UPnP, o DLNA dispõe das mesmas etapas do processo de descoberta até o consumo dos serviços disponibilizados pelos dispositivos. Contudo, algumas adaptações foram realizadas diretamente na sintaxe do UPnP tanto para descoberta quanto para o controle desses dispositivos. Essa diferença será abordada na seção 5.1 deste trabalho.

- **Media Manager:** Essa camada define o tipo de dispositivo que oferece suporte a DLNA e os mecanismos de acesso à mídia ao longo de uma rede. As diretrizes do DLNA, em seguida, aplicam uma camada de restrições sobre os tipos de formato de arquivo de mídia, codificações e resoluções que um dispositivo deve suportar.

- **Media Formater:** Essa camada descreve como o conteúdo de mídia é formatado e codificado, e quais os formatos que ela suporta. A Arquitetura DLNA suporta formatos de mídia (imagens, áudio e áudio/video) específicos, bem como, formatos de documentos de impressão. Junto com cada formato de mídia, a aliança dispõem de um documento padronizado de formatos de mídia suportados pelo protocolo.

As Diretrizes DLNA introduzem uma série de classes de dispositivos para identificar papéis específicos para cada dispositivo, definidos na camada **Media Manager** (Gerenciador de mídia). Essas classes são responsáveis por caracterizar o tipo de dispositivo que está sendo integrado no ambiente, podendo ser dividido em diferentes categorias:

- **Digital Media Server (DMS):** Aqueles utilizados para armazenarem informações, no caso de conteúdos DLNA como vídeos, músicas e fotos. Podem ser citados nessa categorias, dispositivos como PCs, Notebooks, smartphones, tablets e também servidores do tipo NAS (*Network Attached Storage*).

- **Digital Media Player (DMP):** São dispositivos e equipamentos que reproduzem os conteúdos armazenados nos DMS. Em resumo, são aqueles equipamentos que exibem o conteúdo, como TVs, equipamentos de som, vídeos games, entre outros.

- **Digital Media Renderer (DMR):** Essa categoria tem uma função parecida com os DMP, porém, eles não se conectam aos DMS. Eles se conectam ao DMP e reproduzem o conteúdo que já está sendo reproduzido, ou seja, pode-se dizer que o DMR é uma "terceira pessoa" no conjunto. Podendo ser dispositivos como alto falantes sem fios, PCs, notebooks, TVs, gravadoras de DVDs, e outros. Um DMP é um DMR, porém um DMR nem sempre é um DMP.

- **Digital Media Controller (DMC):** São dispositivos que possibilitam encontrar conteúdos nos DMS e controlar sua reprodução nos DMR. Exemplos incluem Tablets, câmeras digitais, PDAs, PCs e Notebooks.

- **Digital Media Printer (DMPr):** É uma subdivisão do DMP, porém só podem fornecer serviços de impressão.

Existem ainda categorias exclusivas para dispositivos móveis que dão suporte ao DLNA, como *smartphones* e *tablet*. Sendo que os *smartphones* atuais tanto podem encaixar nas categorias abordadas anteriormente, como, se encaixam em todas as categorias específicas para dispositivo móveis:

- **Mobile Digital Media Uploader (M-DMU):** Esses enviam o seu conteúdo (upload) para outro aparelho que servirá de DMS. Exemplos: câmeras digitais, celulares, smartphones e tablets.

- **Mobile Digital Media Downloader (M-DMD):** Estes dispositivos encontram e baixam (download) o conteúdo de um servidor de mídia digital (DMS).

- **Mobile Digital Media Controller (M-DMC):** Estes equipamentos pegam conteúdos de servidores de mídia e enviam para processadores de mídia digital (DMR).

2.5 Considerações Finais do Capítulo

Este Capítulo contextualizou grandes áreas de pesquisa deste trabalho: Ambientes Inteligente, Internet das coisas e Protocolos de Conectividade, apresentando algumas definições disponíveis na literatura. A partir destas, a definição a ser utilizada neste trabalho foi explicitada, considerando os diferentes pontos de vistas dos demais autores.

A partir desta, pode-se notar a evidente correlação entre estas áreas de pesquisa. Este fato pode ser comprovado analisando as principais aplicações que visa otimizar alguns aspectos relacionados ao cotidiano das pessoas, tanto no ambiente doméstico como profissional (MATTERN; FLOERKEMEIER, 2013).

Por fim, também foram abordados os responsáveis com viabilizar a conectividade e troca de informações entre os dispositivos agregados no ambiente, os protocolos de conectividade, estes tem como objetivo expor os serviços disponibilizados pelo dispositivo que o agrega, como também, ao ser incorporado a um ambiente, ser capaz de integrar-se automaticamente ao mesmo e descobrir a presença de outros dispositivos e suas funcionalidades.

3 Revisão Sistemática

Uma revisão sistemática da literatura é um "meio de identificar, avaliar e interpretar toda pesquisa disponível e relevante a uma questão, ou área, ou fenômeno de interesse de uma pesquisa, aumentando a qualidade do material sobre o assunto de interesse" (KITCHENHAM et al., 2010; CRD, 2010), ou seja, compreender rigorosamente o estado da arte do assunto pesquisado e ficar atualizado com as abordagens mais recente. Para esse êxito, os autores definem uma sequência bem definida de passos metodológicos para a aplicabilidade da Revisão Sistemática em três fases principais: Planejamento da Revisão; Execução da Revisão e Análise dos Resultados; como observado na Figura 10

Figura 10: Procedimento para Revisão sistemática adaptado de (KITCHENHAM et al., 2010)
Fonte: Adaptado de (KITCHENHAM et al., 2010)

Como podem ser observadas, as fases da revisão não são sequenciais, envolvendo assim interações entre as partes, com isso, o pesquisador pode voltar às fases anteriores com o objetivo de melhorar a revisão, caso haja necessidade. Consequentemente, as funções destinadas por cada uma das fases, ainda segundo Kitchenham et al. (2010), CRD (2010), são: Planejamento da Revisão; Execução da Revisão e Análise dos Resultados.

Na primeira fase, a do Planejamento da Revisão são definidos os passos que serão usados para aplicar a revisão sistemática, elaborando assim, um protocolo para esta finalidade. Este protocolo detalha todos os elementos para ser feita a revisão: objetivos a serem alcançados; questionamentos a serem respondidos; palavras chaves que geram *string* de Busca; Critérios para procedimento de seleção, estratégias para extração das informações. Uma vez terminada a elaboração do protocolo, o pesquisador passa para a próxima

fase, a de Execução da revisão. Nesta segunda fase, serão identificados, selecionados e avaliados os dados referentes à pesquisa em questão. Esses dados retornados serão avaliados seguindo o procedimento de seleção estabelecido pelo protocolo. Por último, na fase de Análise dos resultados, os dados serão extraídos e processados de maneira eficaz, tendo como principal finalidade responder as questões de pesquisas definidas no protocolo.

Existem ainda outros modelos de Revisões sistemática na literatura (CRD, 2010). Entretanto, para esse trabalho em particular, a realização da revisão sistemática foi baseada exclusivamente no modelo disponível em (KITCHENHAM et al., 2010), por esse ser um dos modelos mais referenciados por trabalhos dessa área de pesquisa. Desta forma, as seções seguintes são destinadas somente na aplicabilidade da revisão deste trabalho.

3.1 Planejamento da Revisão Sistemática

Para elaborar o planejamento da revisão sistemática, (KITCHENHAM et al., 2010) alega que essa etapa da revisão deve ser dividida em dois estágios distintos. O primeiro define a necessidade da revisão, ou seja, o autor deve introduzir um resumo de informações existentes sobre um determinado tema onde deseja realizar a pesquisa, informações essas que já foram exploradas na introdução deste trabalho. No segundo estágio é definido o protocolo da revisão sistemática o qual orientará as outras fases da revisão, e nele serão determinadas as etapas para sua aplicabilidade.

3.1.1 Protocolo da Revisão Sistemática

Nessa seção serão definidas as etapas responsáveis por originar o protocolo da revisão sistemática.

- **Objetivo:** Essa etapa está alinhada diretamente com os objetivos deste trabalho, e devem está claros e factíveis. Vale ressaltar que essa etapa deverá ter exatidão na sua definição, uma vez que essa é a base para ser aplicada nas outras fases do protocolo. O objetivo principal desse protocolo é **realizar uma análise do estado da arte sobre a interoperabilidade dos atuais *Mobile Devices* com os dispositivos digitais presentes uma *smarthome***.

- **Questões de pesquisa**: Nessa etapa são identificadas quais questões serão respondidas com base nas características do objetivo do estudo. Nessa processo, as questões geradas fazem parte do contexto geral do trabalho, ou seja, serão essas

questões responsáveis por gerarem base para elaborar a string de busca do protocolo, como também, são elas que serão analisadas e respondidas com base na extração das informações da pesquisa. As questões de pesquisa abordadas para este trabalho são:

QP1: Quais os protocolos mais utilizados que permitem a conectividade entre os atuais *Mobile Devices* e os dispositivos digitais de uma *Smarthome*?

QP2: Qual a importância da interoperabilidade entre dispositivos em uma *Smarthomes*?

QP3: Quais são as principais soluções propostas para conectividade entre os atuais *Mobile Devices* e os dispositivos digitais de uma *Smarthome*?

QP4: Qual o principal problema encontrado nas principais soluções abordadas?

- **Estratégia de Busca:** Essa etapa descreve onde e como serão realizadas as buscas da pesquisa no estado da arte. A execução das buscas teve o suporte computacional da ferramenta JabRef[1] que auxiliou na organização e catalogação dos documentos.
  - **Palavras-chave:** Antes de gerar a *string* de busca para realizar a pesquisa, primeiramente filtram-se as palavras chaves em com relação direta as questões de pesquisa do protocolo, objetivando assim otimizar a pesquisa somente em relação ao objetivo proposto. As palavras-chave coletadas são:
    * *Mobile Devices*;
    * *Smarthome Devices(Prints, Lights, TV Digital, etc)*;
    * *Ubiquitous/Pervasive Computing*;
    * *Internet of Things*;
    * *Interoperability*;
    * *Communication*;
    * *Connectivity*;
  - **Strings de busca:** Depois de identificar as palavras chaves anteriormente, é preciso compreender as regras para criação de uma *strings* de busca otimizada. Com base nas regras definidas por (KITCHENHAM et al., 2010), e realizando teste com a combinação das palavras chaves, seus sinônimos e com os operadores lógicos da busca booleana (*AND* e*OR*), foi definida, semanticamente, a *string* de busca da seguinte forma:

[1] http://jabref.sourceforge.net

    * *(((Ubiquitous OR Pervasive) AND Computing) OR Internet of Things) AND ((Mobile Device) AND (Interoperability OR Connectivity OR Communication OR integration) AND (Smarthome))*

  – **Línguas:** Serão utilizados para a pesquisa nessa revisão sistemática trabalhos escritos somente em língua inglesa, devido esta ser o padrão para publicações internacionais adotados pelos repositórios de dados acadêmicos.

- **Fontes:** As fontes constituem-se de periódicos ou bases de dados que serão utilizados para pesquisa realizada pela revisão sistemática. Foram selecionadas como fontes para essa pesquisa bases de dados das bibliotecas digitais:

  – *ACM Digital Library*;

  – *IEEEXplore Digital Library*;

- **Critérios de Seleção:** Quando os resultados das buscas forem obtidos, estes serão analisados quanto à sua relevância de forma a qualificar se deverão ser considerados importantes para a pesquisa, levando em conta os objetivos da pesquisa. Por exemplo, se a revisão busca identificar os protocolos de conectividade entre dispositivos em um ambiente inteligente, os trabalhos retornados pela *string* de busca devem conter não somente definições que esses protocolos existam, mas também que esses protocolos façam parte também do trabalho.

  – **Critérios de Inclusão:**

    CI-1 - O resultado retornado deve conter relação com no mínimo uma das questões de pesquisa;

  – **Critérios de Exclusão;**

    CE-1 - O resultado não trata do tema e não está relacionado com os objetivos desta pesquisa.

    CE-2 - O resultado retornado já foi encontrado em pesquisa anterior;

    CE-3 - O trabalho só terá acesso de forma paga;

    CE-4 - O trabalho tenha mais de 10 anos de publicação da data dessa pesquisa;

- **Estratégia de seleção:** Essa etapa define os passos realizados pela fase de Análise de Resultado da Revisão sistemática para identificar, quantificar, avaliar e selecionar os trabalhos retornados pela *string* de busca. As etapas para realizar essa estratégia segundo (KITCHENHAM et al., 2010) são:

1. É executado a *string* de busca nos veículos de pesquisa adotados pela revisão, levando em consideração suas particularidades quanto a sintaxe mas não perdendo sua essência da semântica;
2. Os trabalhos retornados pela busca são documentados e inseridos na ferramenta JabRef, para auxiliar na sua categorizarão;
3. Os trabalhos serão analisados a partir da verificação dos critérios de inclusão e exclusão. Estes se darão pela leitura *abstract*, introdução e conclusão do artigo;
4. Os trabalhos retornados serão categorizados e inseridos em uma tabela com as seguintes informações: ID; Título; Fonte; Critério; Avaliação; Link;
5. Por último, será exibida uma tabela resumindo as informações colhidas ao longo de procedimento de seleção;

- **Estratégia de Extração:** Por último, esta etapa é destinada a fase de Análise dos Resultados da revisão sistemática, sendo projetada para coletar as informações necessárias para responder as questões de pesquisa do protocolo. Assim, cada trabalho aceito no processo de seleção, será lido por completo e terá sua qualidade analisada, considerando a relação com os assuntos abordados nas questões de pesquisa e nos critérios de inclusão. Vale destacar que essa análise será realizada somente pelo autor deste trabalho.

3.2 Execução da Revisão Sistemática

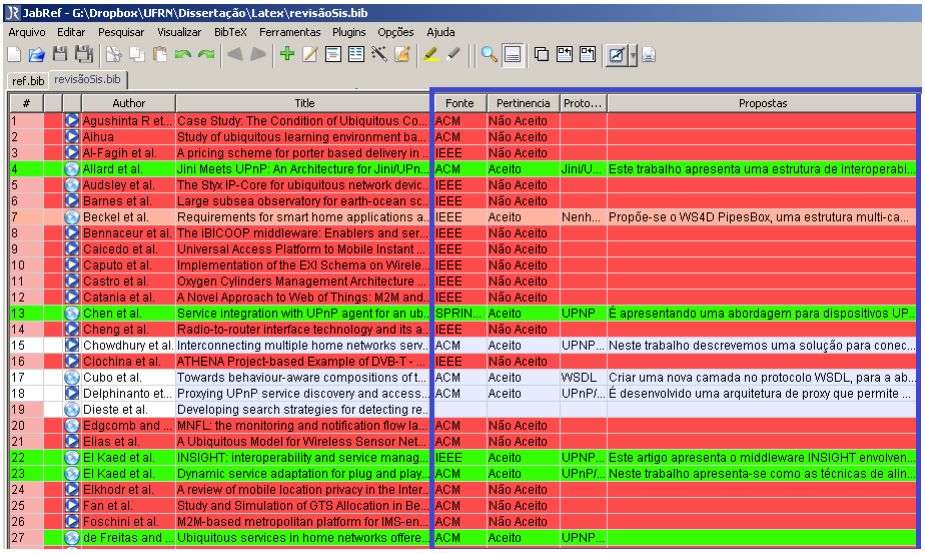
Após definido o protocolo da revisão sistemática, é realizado o procedimento de Execução da Revisão, seguindo as etapas da Estratégia de Seleção definidas também no protocolo. Entretanto, se algum item sofrer alteração durante a fase de execução, esse deve ser documentado nessa fase (KITCHENHAM et al., 2010).

- **Execução das Buscas:** Após ser definida a *string* de busca final a ser utilizada, esse tópico é responsável por executá-las nas bases de dados definidas no protocolo. A busca realizada na base de dado ACM transcorrem sem nenhuma alteração, entretanto, na base de dados IEEE, o *site* apresentou diversos problemas pela busca padrão, onde foi aplicada a *string* utilizando sua busca avançada mudando superficialmente a sintaxe da *string* mas mantendo sua essência semântica. A Tabela 1 resume a quantidade de trabalhos retornados nas bases de dados.

Base de Dados	Trabalhos retornados
ACM	46
IEEE	97

Tabela 1: Execução das Buscas

- **Documentação:** Após realizar as buscas nas bases de dados, os trabalhos retornados são então inseridos na ferramenta JabRef[2], que foi o gerenciador de referências utilizado para manipular os trabalhos retornados pelas bases de busca. A ferramenta JabRef foi útil para identificar repetições de referências. Além disso, permitiu a criação de campos customizados para registrar as informações extraídas, manter o controle dos dados de origem e acompanhar a decisão tomada em cada etapa da fase de execução da revisão. Observe uma pré-visualização do uso da ferramenta na Figura 11.



#	Author	Title	Fonte	Pertinencia	Proto...	Propostas
1	Agushinta R et...	Case Study: The Condition of Ubiquitous Co...	ACM	Não Aceito		
2	Aihua	Study of ubiquitous learning environment ba...	ACM	Não Aceito		
3	Al-Fagih et al.	A pricing scheme for porter based delivery in...	IEEE	Não Aceito		
4	Allard et al.	Jini Meets UPnP: An Architecture for Jini/UPn...	ACM	Aceito	Jini/U...	Este trabalho apresenta uma estrutura de interoperabi...
5	Audsley et al.	The Styx IP-Core for ubiquitous network devic...	IEEE	Não Aceito		
6	Barnes et al.	Large subsea observatory for earth-ocean sc...	IEEE	Não Aceito		
7	Beckel et al.	Requirements for smart home applications a...	IEEE	Aceito	Nenh...	Propõe-se o WS4D PipesBox, uma estrutura multi-ca...
8	Bennaceur et al.	The iBICOOP middleware: Enablers and ser...	IEEE	Não Aceito		
9	Caicedo et al.	Universal Access Platform to Mobile Instant ...	IEEE	Não Aceito		
10	Caputo et al.	Implementation of the EXI Schema on Wirele...	IEEE	Não Aceito		
11	Castro et al.	Oxygen Cylinders Management Architecture ...	IEEE	Não Aceito		
12	Catania et al.	A Novel Approach to Web of Things: M2M and...	IEEE	Não Aceito		
13	Chen et al.	Service integration with UPnP agent for an ub...	SPRIN...	Aceito	UPNP	É apresentando uma abordagem para dispositivos UP...
14	Cheng et al.	Radio-to-router interface technology and its a...	IEEE	Não Aceito		
15	Chowdhury et al.	Interconnecting multiple home networks serv...	ACM	Aceito	UPNP...	Neste trabalho descrevemos uma solução para conec...
16	Ciochina et al.	ATHENA Project-based Example of DVB-T - ...	IEEE	Não Aceito		
17	Cubo et al.	Towards behaviour-aware compositions of t...	ACM	Aceito	WSDL	Criar uma nova camada no protocolo WSDL, para a ab...
18	Delphinanto et...	Proxying UPnP service discovery and access...	ACM	Aceito	UPnP/...	É desenvolvido uma arquitetura de proxy que permite ...
19	Dieste et al.	Developing search strategies for detecting re...				
20	Edgcomb and ...	MNFL: the monitoring and notification flow la...	ACM	Não Aceito		
21	Elias et al.	A Ubiquitous Model for Wireless Sensor Net...	ACM	Não Aceito		
22	El Kaed et al.	INSIGHT: interoperability and service manag...	IEEE	Aceito	UPNP...	Este artigo apresenta o middleware INSIGHT envolven...
23	El Kaed et al.	Dynamic service adaptation for plug and play...	ACM	Aceito	UPnP/...	Neste trabalho apresenta-se como as técnicas de alin...
24	Elkhodr et al.	A review of mobile location privacy in the Inter...	ACM	Não Aceito		
25	Fan et al.	Study and Simulation of GTS Allocation in Be...	ACM	Não Aceito		
26	Foschini et al.	M2M-based metropolitan platform for IMS-en...	ACM	Não Aceito		
27	de Freitas and ...	Ubiquitous services in home networks offere...	ACM	Aceito	UPNP...	

Figura 11: Ferramenta JabRef
Fonte: Autor

A ferramenta também oferece a possibilidade de criar relatórios específicos para exportar todas essas informações em um formato pré-definido, facilitando o trabalho de manipulação das referências e extração das informações.

- **Seleção dos Trabalhos:** Ainda pode-se observar na Figura 11 que os campos customizados com os nomes Critérios e Pertinência criados na ferramenta JabRef, auxiliaram também no terceiro processo da Estratégia de Extração. Ou seja, a ana-

[2]http://jabref.sourceforge.net/

lise dos trabalhos com base nos critérios de inclusão e exclusão a partir da leitura do *abstract*, introdução e conclusão do artigo;

- **Categorização:** Terminada a Seleção dos Trabalhos com base nos Critérios abordados, foi exportada, também a partir JabRef, uma nova tabela resumindo as principais informações de todos os trabalhos retornados nesta pesquisa, como pode ser observado no Apêndice A. Contudo, como uma análise rápida para essa fase, a Tabela 2 exibe essas mesmas informações somente dos trabalhos que fazem parte diretamente dessa pesquisa, ou seja, aqueles que foram incluídos no processo de seleção e fazem parte da avaliação para responder as questões de pesquisa do protocolo da revisão.

Tabela 2: Trabalhos aceitos na pesquisa

ID	TÍTULO	FONTE	AVALIAÇÃO	QUESTÕES
4	UPnP Service and Jini Client	ACM	Aceito	QP-1, QP-2
7	Requirements for smart home applications and realization with WS4D-PipesBox	IEEE	Aceito	QP-1, QP-2
13	Service integration with UPnP agent for an ubiquitous home environment	IEEE	Aceito	QP-4
15	Interconnecting multiple home networks services	ACM	Aceito	QP-1
17	Towards behaviour-aware compositions of things in the future internet	ACM	Aceito	QP-2
18	Proxying UPnP service discovery and access to a non-IP Bluetooth network on a mobile phone	ACM	Aceito	QP-2
22	INSIGHT: interoperability and service management for the digital home	IEEE	Aceito	QP-1, 2, 3, 4
23	Dynamic service adaptation for plug and play device interoperability	ACM	Aceito	QP-2, QP-3
27	Ubiquitous services in home networks offered through digital TV	IEEE	Aceito	QP-1,QP-2
34	ubiHome: An Infrastructure for Ubiquitous Home Network Services	IEEE	Aceito	QP-1,2,3
38	Intelligent Traffic Management Systems for Work Zones and Incident Management	IEEE	Aceito	QP-1
43	Networking in a smart home - Providing lightweight networking services for heterogeneous devices	IEEE	Aceito	QP-2
50	Seamless Integration of Heterogeneous Devices and Access Control in Smart Homes	ACM	Aceito	QP-1, QP-2
53	Mobility support for universal plug and play (UPnP) devices using session initiation protocol (SIP)	ACM	Aceito	QP-3, QP-4
54	Context-aware multimedia streaming service for smart home	IEEE	Aceito	QP-1, QP-2
60	Domotic technologies incompatibility becomes user transparent	ACM	Aceito	QP-1, QP-2
61	Bridging UPnP and ZigBee with CoAP: protocol and its performance evaluation	ACM	Aceito	QP-1,2,3,4
69	Mobile TV content to home WLAN	IEEE	Aceito	QP-3, QP-4
82	A Middleware Platform for Application Configuration, Adaptation and Interoperability	IEEE	Aceito	QP-1, QP-4
84	Wide-Area Media Sharing with UPnP/DLNA	IEEE	Aceito	QP-3, QP-4
87	Multiagent control system with mobile ubiquitous platform for ambient intelligence	IEEE	Aceito	QP-2, QP-3
88	Heterogeneous device discovery framework for the Smart Homes	IEEE	Aceito	QP-3, QP-4
96	InfoPods: Zigbee-based remote information monitoring devices for smart-homes	IEEE	Aceito	QP-3, QP-4

- **Consolidação da pesquisa:** Terminado todo o processo de Execução da Revisão, essa etapa consolida os resultados da pesquisa. A Tabela 3 expõe a quantidade de referências recuperadas de acordo com os veículos de busca utilizados, a quantidade de artigos selecionados para leitura, avaliação e documentação, como também, a quantidade de artigos aceitos de acordo com a estratégia de seleção. Para evitar uma análise exaustiva e baseada no fato de que os resultados são ordenados por relevância, foram selecionados no máximo os 50 primeiros trabalhos retornados na busca dos veículos.

Tabela 3: Tabela de Consolidação

Base de Dados	Retornados	Selecionados	Aceitos
ACM	46	46	9
IEEE	97	50	14

3.3 Análise dos Resultados

A fase de Análise dos Resultados permite que sejam coletadas, extraídas e processadas todas as informações necessárias para responder as questões de pesquisa baseando-se na Estratégia de Extração do protocolos (KITCHENHAM et al., 2010).

3.3.1 Protocolos de Conectividade

O principal objetivo dos protocolos de conectividade em uma *smarthome* é fornecer flexibilidade no compartilhamento de informação, de forma transparente, entre os dispositivos, sejam móveis ou eletrônicos, principalmente para o usuário final (LAI; HUANG, 2008).

Os protocolos de conectividade facilitam a interação entre dispositivos eletrônicos e computadores, descartando a problemática com a configuração da entrada de um novo componente no ambiente e, consequentemente, libertando o usuário de fazê-lo. Assim, um usuário sem conhecimentos técnicos poderá monitorar e controlar equipamentos como geladeira, ar-condicionado, fogão, luzes, janelas, portas, TVs, ventiladores, impressoras e outros dispositivos computacionais, simplesmente a partir de outro dispositivo (BECKEL et al., 2011; KAED et al., 2011; KIM et al., 2012; LAI; HUANG, 2008).

Devido ao avanço acelerado das tecnologias e ao surgimento de uma grande quantidade de dispositivos heterogêneos em uma mesma *smarthome*, uma diversidade de protocolos de conectividade independentes foram criados (ALLARD et al., 2004; KAED et al., 2011; WANG et al., 2008), estabelecendo um cenário complexo de garantir a interoperabilidade entre estes. Essa seção apresenta o resultado da pesquisa do estado da arte dos protocolos de comunicação e conectividade em uma *smarthome*, respondendo à questão de pesquisa **QP1**, definida no protocolo da Revisão Sistemática.

A Tabela 4 apresenta os protocolos abordados nos 23 artigos analisados. As informações disponibilizadas são sobre os protocolos de conectividade abordados nos trabalhos e a referência onde esses foram utilizados. Essas referências são as mesmas utilizadas no campo ID no processo de categorização abordadas na etapa de Execução da Revisão

ou no Apendice A. Vale ressaltar que alguns destes trabalhos somente citavam protocolos capazes de gerenciar dispositivos presentes em uma *smarthome*, mas não chegaram a utilizá-lo em suas pesquisas. Para a análise deste trabalho, foram considerados somente os protocolos diretamente utilizados nos trabalhos de pesquisas.

Tabela. 4: Protocolos de Conectividade

PROTOCOLS	ID
UPnP	4,13,15,18,22,23,27,34,50,53,54,61,84
DLNA	13, 27, 54, 60, 69, 84
ZigBee	53, 61, 82, 87, 96
DPWS	22, 23
JINI	4
Bluetooth SDP	18

- O *Universal Plug and Play* (UPnP) , foi iniciado pela *Microsoft* no ano de 1999 como um modelo de arquitetura de rede que provê conectividade *ad-hoc* entre dispositivos de forma distribuída, transparente, independente de *driver* ou plataforma e sem a necessidade de qualquer tipo de configuração. Uma das principais características do UPnP é a facilidade em disponibilizar serviços na rede, possuindo um protocolo de descoberta de dispositivos e, conseguentemente, de serviços. Uma principal restrição do UPnP é que sua estrutura é limitada a uma única rede LAN. Empresas como Nokia, Intel e HP, colaboram com esse protocolo, surgindo com essa parceria o UPnP Forum[3]

- O *Digital Living Network Alliance* (DLNA) é um padrão estabelecido em 2003 pela *SONY* e adotado por várias empresas na indústria. É um protocolo de conectividade que tem como objetivo promover a interoperabilidade entre equipamentos e dispositivos eletrônicos, dispositivos móveis e computadores pessoais. Este utiliza o protocolo UPnP como base para descoberta de dispositivo, usufruindo do poder de sua interoperabilidade e suas características.

- O ZigBee é um protocolo de rede sem fio desenvolvido pela *Alliance ZigBee*[4] com técnicas de comunicação sem fio de baixo consumo de energia e frequência de curta distância. As principais características do ZigBee é a baixa taxa de transmissão de dados e o baixo custo dos dispositivos. Um dos projetos desenvolvidos pela *Alliance ZigBee* é o *ZigBee Home Automation*, que oferece um padrão de interoperabilidade

[3]http://www.upnp.org
[4]http://www.zigbee.org

de comunicação para equipamentos domésticos, digitais e dispositivos com plataformas de desenvolvimento, permitindo assim, a integração com dispositivos em outras redes ZigBee.

- O *Device Profile for Web Service* (DPWS) surgiu em 2004 e foi reconhecido e padronizado em 2008. Mantido pela OASIS[5], o DPWS é uma pilha de protocolos que define um conjunto mínimo de funcionalidades para que dispositivos limitados de recursos possam adotar padrões *Web Service*. Suas principais características são notificações de eventos, troca de mensagens, serviços de descoberta e descrição de serviços, desenvolvendo mecanismos de comunicação de alto nível para interoperabilidade entre dispositivos presentes em uma *smarthome*.

- O protocolo JINI[6] foi criado em 1998 e mantido pela *Sun Microsystem*. Assim como o protocolo UPnP, DPWS e DLNA, o JINI é um protocolo que possibilita que dispositivos conectem e compartilhem recursos. Os dispositivos envolvidos podem ser, desde computadores pessoais até dispositivos móveis e eletrônicos. Entretanto, esse protocolo é limitado à plataforma JAVA, podendo ser utilizado em qualquer dispositivo que contenha uma JVM (*Java Virtual Machine*) em sua plataforma.

- Por fim, o protocolo *Bluetooth Service Discovery Protocol* (BSDP) é usado para permitir que, dispositivos, através da tecnologia *Bluetooth*, possam descobrir quais serviços os outros dispositivos suportam ou disponibilizam. O BSDP tem como base o protocolo *Service Discovery Protocol* (SDP) , que é o mesmo utilizado pelo UPnP e o DLNA para descobrirem dispositivos em uma rede. O BSDP é mantido pela organização *Bluetooth SIG*[7].

Conforme pode ser observado na Figura 12, o protocolo UPnP se destaca em relação aos outros quantitativamente, ou seja, 46% dos artigos aceitos preferem a utilização do protocolo UPnP para realizarem a comunicação entre os *Mobiles Devices* com dispositivos em uma *smarthome*, ficando em segundo o protocolo DLNA com 26% e em terceiro o protocolo ZigBee.

[5] www.oasis-open.org
[6] http://www.jini.org
[7] https://www.bluetooth.org/

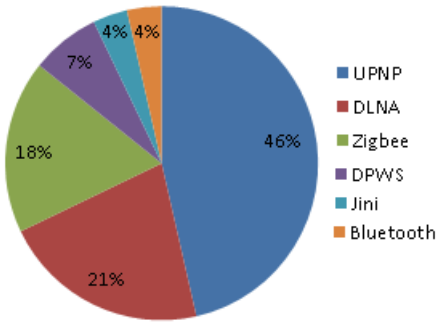


Figura 12: Protocolos de Conectividade
Fonte: Autor

O UPnP se destaca nos trabalhos selecionados por ser um padrão aberto e principalmente ser baseado no protocolo IP, tornando-se um protocolo compatível com qualquer equipamento voltado à conexão de rede (WANG et al., 2008). Ele possui uma arquitetura leve, que permite sua utilização tanto por dispositivos de alta tecnologia, como microcontroladores, podendo ser utilizado para realizar simples operações de ligar-desligar (*Light-Upnp*), ou realizar processamento para tomadas de decisões específicas (*Presence-UPnP*)(KUMAR; RAHMAN, 2006). Além disso, ele é um protocolo que opera usando a tecnologia *peer-to-peer*, não necessitando de nenhum equipamento ou tecnologia presente entre os dispositivos participantes para integrá-lo. Por fim, um dos principais motivos de sua utilização é que ele é um protocolo baseado no modelo *Plug and Play*, facilitando assim a instalação e configuração de dispositivos (CHOWDHURY et al., 2008).

Exemplos de outros protocolos, X10 (KIM et al., 2012), HAVi (*Home Audio Video Interoperability*) (URIBARREN et al., 2008), IGRS (*Intelligent Grouping and Resource Sharing*) (KAED et al., 2011), Salutation (ANAND, 2007), Z-Ware (KIM et al., 2012), Ninja (ALLARD et al., 2004), também foram citados nos artigos, apesar de não terem sido usados com foco principal.

3.3.2 Interoperabilidade

Todo ambiente inteligente, em particular *smarthome*, é definido como uma entidade capaz de adquirir e aplicar conhecimentos sobre o usuário que nele esteja a fim de cumprir as metas de conforto e eficiência. Essas metas podem ser alcançadas através da comunicação plena dos dispositivos e aplicações que nele estejam (ACAMPORA et al., 2013; KIM et al., 2012). O objetivo dessa seção é responder a **QP2** do protocolo da revisão sistemática que questiona sobre a importância da interoperabilidade entre dispositivos em uma

smarthome.

É observado nos trabalhos analisados que, ao longo desses 10 últimos anos, a maioria dos dispositivos em uma *smarthome* é denominada autossuficiente e não é capaz de comunicar-se facilmente com os outros dispositivos nela presentes (DELPHINANTO; KOONEN; PEETERS, 2007; FREITAS; TEIXEIRA, 2009).

Toda *smarthome* terá vários sistemas e infraestrutura que precisam compartilhar o mesmo meio físico. A maioria dessas tecnologias tende a ser desenvolvida de forma isolada. Desta forma, esses sistemas desenvolvidos isoladamente realmente não levam a interoperação de múltiplos sistemas ou tecnologias em um mesmo ambiente (ALLARD et al., 2004), e essa falta de intercomunicação entre essas tecnologias foi ocasionada devido ao crescimento desenfreado da tecnologia (FREITAS; TEIXEIRA, 2009; KAED et al., 2011).

Vários dispositivos em uma *smarthome* precisam compartilhar recursos e podem precisar serem de certa forma, "conscientes"sobre outros dispositivos no mesmo ambiente, bem como ter acesso também a outra fonte de informação. Com isso, esses dispositivos terão autonomia para executar alguma ação inteligente de maneira independente (KAED et al., 2011), e para fornecer um meio de comunicação entre esses dispositivos, qualquer tecnologia, sistema ou aplicação têm de ser desenvolvido seguindo um mesmo padrão tecnológico (KIM et al., 2012).

Quando um sistema é construído com base em uma norma tecnológica, a confiança é incorporada entre os diversos sistemas presentes em uma *smarthome* alcançando assim um caminho para interoperação. E isso é o principal problema quando é falado sobre interconectividade de dispositivos em ambientes inteligentes. Esses problemas podem ser resolvidos se o fator de interoperabilidade entre diferentes tecnologias forem alcançados (KAED et al., 2011; WANG et al., 2008).

Como não há um padrão tecnológico para envolver essa área de ambiente inteligente entre os dispositivos presentes nele, a interoperabilidade entre eles é o grande desafio, melhor dizendo, é o principal desafio a ser abordado nessa área de pesquisa (KIM et al., 2012). Ainda segundo Kim et al. (2012), uma das maneiras de uma *smarthome* se integrar com um usuário em seu ambiente é comunicando-se com dispositivos que estão presentes nele, que façam parte de seu dia a dia e estejam sempre consigo. Portanto, dispositivos móveis inteligentes, tais como os *smartphones*, são classe especial desses dispositivos. Para isso, esses *smartphones* devem também ser capazes de conectarem com outros dispositivos não móveis no mesmo ambiente e realizarem trocas de informações entre eles. E a interoperabilidade é um fator primordialmente para alcançar esse objetivo.

No contexto de *smarthome*, a interoperabilidade é a capacidade dos sistemas, aplicativos, aparelhos e serviços trabalharem em conjunto de forma confiável, previsível e invisível (KAED et al., 2011). Com a interoperabilidade, a heterogeneidade de dispositivos em uma *smarthome* será capaz de trocar informações entre eles, trabalhar juntos, compartilhar recursos e utilizar as informações trocadas para execução da tarefa, mesmo que sejam desenvolvidos seguindo padrões diferentes.

É desejável que as entidades em *smarthome* tenham a necessidade de interagir de forma transparente entre elas para fornecerem ao usuário presente nela uma variedade de aplicações integradas, sem se importar com a heterogeneidade de protocolos presentes. Essa tendência, segundo Mitsugi et al. (2011), Kaed, Denneulin e Ottogalli (2011), é focada na convergência que impulsiona os sistemas domésticos inteligentes ao trabalho em conjunto. Em outras palavras, essa convergência é o fator chave que impulsiona a necessidade de interoperabilidade.

Dispositivos domésticos de consumo e serviços domésticos estão gradualmente tornando parte de sistemas domésticos inteligentes. E com a falta de um padrão para essa área, as empresas estão lançando diferentes tecnologias impondo seu próprio padrão para que seus dispositivos possam trabalhar em conjunto. Com isso, essas diferentes tecnologias não se conectam entre si, ou seja, dispositivos com uma determinada tecnologia criada por uma empresa não encontrará na rede um dispositivo desenvolvido com tecnologia de outra empresa. Isto resulta na principal limitação da interoperabilidade em uma *smarthome*.

Para *smarthomes*, Mitsugi et al. (2011) determina que a interoperabilidade entre dispositivos pode ser definida em três níveis principais: Nível de Rede, que define o meio físico para a interação entre os dispositivos, com base especialmente no mecanismo de troca de mensagens; Nível sintático, define a compreensão da estrutura de dados nas mensagens trocadas entre os sistemas que envolvem os dispositivos; e por último, o principal nível para o problema de interoperabilidade entres os dispositivos presentes em uma *smarthome* com os dispositivos móveis é o nível de Protocolo ou Conectividade (MITSUGI et al., 2011; KAED; DENNEULIN; OTTOGALLI, 2011). Esse último define o padrão que o dispositivo, móvel ou não, irá utilizar para realizar a troca de informação entre eles, ou seja, esse nível fornece o mecanismo para estabelecer conexões lógicas entre os dispositivos.

Em *smarthomes*, a interoperabilidade tem como principal objetivo assegurar a troca de mensagens de forma transparente dos serviços disponibilizados entre dispositivos. Esse objetivo não muda no acréscimo de um dispositivo móvel no ambiente, pois esse nada mais é, que outro dispositivo, que irá consumir ou disponibilizar novos serviços no ambiente

(LAI; HUANG, 2008; ZUALKERNAN et al., 2009; MITSUGI et al., 2011; KAED; DENNEULIN; OTTOGALLI, 2011).

Outro ponto importante que deve ser relatado sobre a leitura dos trabalhos, definido por (KIM et al., 2012), é que para alcançar a interoperabilidade entre dispositivos, alguns requisitos devem ser tomados em consideração:

- Estrutura padrão para a descoberta do dispositivo, controle e configuração. Este requisito corresponde à capacidade dos dispositivos descobrirem a presença de outros dispositivos e identificar os serviços disponibilizados.
- Formatos de mídia padrão e protocolos para streaming. Este requisito refere-se somente aos dispositivos de áudio e vídeo.
- Construção de autenticação padrão. Esta exigência é considerada para os fornecedores de dispositivos disponibilizarem segurança para o acesso do dispositivo.
- Estrutura de QoS comuns. Apesar de que um dos objetivos principais para que um dispositivo se integre em uma *smarthome* seja a autoconfiguração, alguns usuários podem querer aplicar determinada regra de qualidade sobre o uso do seu dispositivo.

Levando em consideração o contexto expressa nessa seção e aplicabilidade dos requisitos abordados anteriormente, podemos destacar benefícios com a interoperabilidade dos dispositivos. Por exemplo, quando os dispositivos se tornarem interoperáveis, o mercado torna-se competitivo e isso acabará resultando em produtos mais baratos. A interoperabilidade irá reduzir esforços nos afazeres domésticos e proporcionará mais tempo para o usuário em atividades auxiliares. Também garantirá mais conforto e comodidade aos usuários nela presente e proporcionará uma segurança mais eficiente, com dispositivos de travas e reconhecimento do usuário, dentre muitos outros.

No entanto, pode-se concluir que o avanço exponencial tecnológico, juntamente com a falta de um padrão específico para o tema abordado neste trabalho, determinaram o surgimento de dispositivos com diferentes protocolos de conectividade no mesmo ambiente, não se adaptando a executarem ações em conjunto. E a interoperabilidade entre essas diferentes tecnologias, é a peça fundamental para a aplicabilidade não só do conceito, mas de todos os fatores e benefícios que envolvem as *smarthomes*.

3.3.3 Principais descobertas

A diversidade de funcionalidades desejadas em uma *smarthome* vem aumentando a complexidade dos sistemas envolvidos nessa área. Essa complexidade tem motivado grupos de pesquisa a descobrirem soluções neste contexto. Um dos problemas bem citados e discutidos é o de intercomunicação entre dispositivos (WARRIACH et al., 2011). Com o objetivo de responder à questão de pesquisa **QP3** abordada no protocolo desta revisão, foram avaliados os projetos de pesquisas reportados nos artigos, dentre esses:

- Em Mitsugi et al. (2011), é desenvolvido um *bridging protocol* (protocolo ponte), para comunicação entre os protocolos UPnP e o ZigBee, utilizando o protocolo COAP (*Constrained Application Protocol*)que também é um dos mais utilizado em dispositivos eletrônicos. Esta solução proporcionará flexibilidade na comunicação entre dispositivos com o protocolo UPnP e ZigBee, deixando suas particularidades transparentes entre si.

- Já Kaed et al. (2011) desenvolveu o *middleware INSIGHT*, com capacidade de fornecer a interoperabilidade e gerenciamento de serviços em uma *smarthome* sobre os protocolos UPnP e DPWS.

- Em Zualkernan et al. (2009), é apresentado o sistema *InfoPods*. Uma arquitetura aberta composta de um controlador baseado no protocolo ZigBee, permitindo assim que qualquer dispositivo que utilize esse protocolo possa controlar os dispositivos eletrônicos pertencentes em uma *smarthome*.

Além destes, outros projetos também foram identificados de maneira indireta, à medida que os artigos eram analisados e citavam trabalhos de pesquisas que não estavam presentes nos artigos resultantes da execução da *String* de busca. Entre esses projetos, podemos destacar os seguintes:

- O *Device Service Bus* (DSB) é um *middleware* que tem como função integrar dispositivos heterogêneos em um ambiente Ubíquo. Este projeto foi desenvolvido em uma plataforma portável com finalidade de ser executado em qualquer tipo de plataforma, seja móvel ou eletrônica. Tem como principal objetivo, criar um barramento de comunicação para que equipamentos com tecnologias específicas, como o RFID[8], *Bluetooth*, Wi-fi, possam disponibilizar serviços na rede. Este utiliza o protocolo DPWS para realizar a interoperabilidade entre dispositivos presentes no ambiente.

[8]http://www.rfidsystems.com.br/

- O projeto *Accessing Web-based Applications on Consumer Devices* (Web4CE) define uma arquitetura de rede que permite que dispositivos presentes em uma *smarthome* possam ter acesso diretamente a web, sem depender de outra tecnologia para isso, sendo possível acessar sua configuração individual. Esse projeto usa do protocolo DLNA para a interoperabilidade de sua rede com outros dispositivos.

- O MavHome[9] é um projeto de uma *smarthome*, desenvolvido pela Universidade do Texas. Seu objetivo é tornar os ambientes adaptativos, que percebe o estado da casa com o auxílio de sensores. Nela, dispositivos inteligentes controlam o ambiente garantindo o conformo do usuário através do reconhecimento de atividade, por meio de câmeras de vídeo e sensores em geral. Esses dispositivos processam dados para reconhecer o que os usuários estão fazendo, portanto, podem tomar alguma iniciativa para prever o que o usuário faria depois, auxiliando-o de alguma forma e minimizando o esforço por parte do usuário.

Outras propostas, como Kaed, Denneulin e Ottogalli (2011), Lai e Huang (2008), Mitsugi et al. (2011), Rus et al. (2008), tem como objetivo permitir que protocolos distintos possam interconectar-se em uma *smarthome*, evitando as particularidades de cada protocolo. A motivação principal para a maioria dos trabalhos aceitos nessa pesquisa é o desenvolvimento de plataformas de *middlewares* de integração, com o objetivo de interconectar todos os tipos de dispositivos com diversas tecnologias de comunicação em um mesmo ambiente.

Apesar dessas solução com uso de *middlewares* ser bastante aceita, trabalhos como Kumar e Rahman (2006), Patel, Anand e Kumart (2010), Zualkernan et al. (2009) defendem uma solução com a utilização de um único protocolo de conectividade em uma *smarthome*. Esta abordagem diminuiria a complexidade oferecida para a integração dos diferentes tipos de protocolos e seria certamente mais performática e econômica.

O fato é que, no estágio atual de desenvolvimento das tecnologias, uma solução integradora, baseada em *middleware*, é a resposta mais rápida e flexível para garantir a interoperabilidade entre os dispositivos, em uma visão de longo prazo, o caminho mais adequado seria que um único padrão predominasse. Esse padrão deveria ser evoluído o suficiente para incorporar as características e vantagens dos demais, além de ser aberto e mantido por um grande número de representantes da indústria e academia. Provavelmente, o UPnP é o protocolo que esteja mais próximo desta realidade, mas ainda precisa evoluir bastante do ponto de vista técnico para poder ser considerado substituto dos demais.

[9] http://ailab.wsu.edu/mavhome

3.3.4 Problema Abordado

Uma *smarthome* contém diversos dispositivos heterogêneos interligados e eles fazem uso de serviços ou trocam informações de maneira transparente e dinâmica (CHEN; KUO; CHAO, 2009). A capacidade de ser reconhecido em um ambiente de forma automática, segundo Kaed et al. (2011), Lai e Huang (2008), é peça fundamental para a concretização de um ambiente inteligente e função mais fascinante dos protocolos identificados nesse trabalho. Por outro lado, a diversidade destes protocolos e tecnologias atualmente existentes torna impeditiva a interoperabilidade transparente entre os dispositivos e consequentemente inviabiliza a implementação plena de ambientes residenciais inteligentes. Essa seção é responsável por abordar os problemas relacionados com às soluções propostas pelos trabalhos analisados, respondendo assim, a questão de pesquisa **QP4** do protocolo da revisão sistemática.

Como visão geral do funcionamento arquitetural dos protocolos integrados em uma *smarthome*, pode-se identificar duas características comuns entre esses. Primeiro, tais protocolos utilizam protocolos de comunicação baseados na pilha de protocolo TCP/IP, ou seja, qualquer equipamento com suporte a um protocolo para uma *smarthome* pode ser integrado usando qualquer meio de comunicação, seja esse ethernet, wi-fi e etc. Contudo, é necessário que o equipamento possa adquirir um endereço IP. Segundo, a classificação como componente em uma *smarthome*, segundo Kaed et al. (2011), basicamente pode ser de duas formas: dispositivos controláveis (*Devices*) - aqueles que disponibilizam os serviços que os dispositivos oferecem e Pontos de Controle (*Control Point*) - aqueles responsáveis por utilizar os serviços disponibilizados pelos dispositivos. Ainda segundo Kaed et al. (2011), outras componentes podem surgir dependendo do protocolo, mas em geral, tais protocolos são baseados na arquitetura cliente-servidor, onde o cliente é representado pelo Ponto de Controle e o Servidor o dispositivo.

Apesar desses protocolos terem metas semelhantes de alto nível, eles são compostos de arquiteturas bem diferentes (HA; SOHN; CHO, 2007). Cada equipamento, usando a descoberta de serviço, vai utilizar apenas um destes protocolos, por exemplo, o Ponto de Controle UPnP só encontrará em uma rede os dispositivos UPnP, e assim vale para os outros protocolos. O que significa que os clientes e serviços que utilizam tecnologias diferentes, não serão capazes de cooperarem. Uma vez que é provável que vários protocolos sejam amplamente utilizados no ambiente, há uma necessidade de uma estrutura de interoperabilidade que permite que os clientes e os serviços escritos usando protocolos diferentes cooperem.

Até o momento percebeu-se que para integrar um novo dispositivo com um protocolo não suportado atualmente pelo ambiente, existem duas possibilidades como ilustrado na Figura 13.

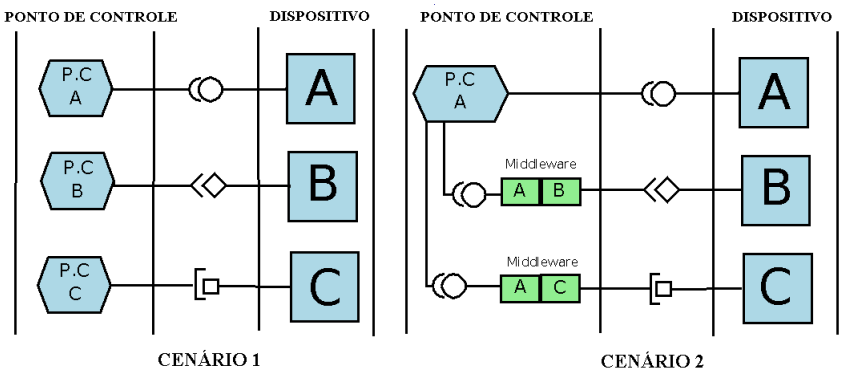


Figura 13: Cenários para inserção de novos protocolos
Fonte: Autor

Observa-se no Cenário 1 que para integração de vários dispositivos com protocolos distintos em uma *smarthome* é necessário adicionar Pontos de Controle para interagir com os serviços disponibilizados pelo dispositivo. Para cada dispositivo com protocolos A, B e C, Pontos de Controle P.C-A, P.C-B e P.C-C devem ser adicionados no ambiente. Os principais problemas relacionados a esse cenário são, a quantidade de Pontos de Controle gerenciados pelo usuário, criando uma solução não escalável, já que o principal objetivo de uma *smarthome* é diminuir o esforço do usuário em controlar os dispositivos do ambiente; e a não interação entre os dispositivos, já que cada dispositivo suporta apenas um único protocolo e esses protocolos não se reconhecem no mesmo ambiente.

Já no Cenário 2, observa-se que os *middlewares* são soluções amplamente usadas para interoperabilidade entre diferentes protocolos de comunicação em *smarthomes*. De maneira mais concreta, o problema se dá na adição de dispositivos com o protocolo de conectividade B e C em um ambiente com suporte somente ao protocolo A. *Middlewares* deverão ser desenvolvidos e integrados diretamente no ponto de controle A para interconectar os novos protocolos no ambiente. Nesse caso, observa-se como principal vantagem a minimização do overhead produzido pela conversão dos pacotes entre os protocolos, já que existe mais de uma camada de *middleware* em um único Ponto de Controle. Em contrapartida, as soluções existentes tendem a ser intrusivas nos Pontos de Controle, ou seja, para integrar qualquer camada de *middleware* em tais controladores, que seria a solução mais ideal para o reaproveitamento dos Pontos de Controle existentes, o acesso direto ao código fonte desses deverá ser liberado, o que muitas soluções já existentes não disponibilizam desse acesso ao código.

Outro ponto a ser destacado que as soluções até o presente não abordam é o fato da interação entre os próprios dispositivos, essa interação não será somente por meio de um Pontos de Controle, mas a interação direta entre eles. Dispositivo em *smarthomes* podem consumir serviços disponibilizados por outros dispositivos no mesmo ambiente. Por exemplo, uma lâmpada inteligente poderá ficar interessado no serviço ligar-desligar disponibilizado no ambiente por uma TV inteligente, para que, quando solicitação esse serviço, a lâmpada possa automaticamente desligar-se caso a TV seja ligada, ou, ligar-se caso a TV seja desligado, dependendo claro da preferência do usuário presente nela. As soluções existentes abordam somente a camada de *middleware* focado nos Pontos de Controle dos dispositivos, e nada em relação a uma camada que possa interoperabilizar a relação entre os próprios dispositivos.

No entanto, Kumar e Rahman (2006) destaca que uma diversidade de *middlewares* presentes em um componente de um ambiente inteligente proporcionará, em algum momento, incompatibilidades quando interconectarem entre si. Além disso, tais soluções produzem um aumento gradual da complexidade do ambiente e essas soluções encontradas até o momento estão ligadas diretamente a mudança no contexto somente dos Ponto de Controle e não dos dispositivos.

Outro ponto importante a ser destacado por Kaed et al. (2011) é que a heterogeneidade sintática, juntamente com a diversidade de diferente de camadas de protocolos, possam dificultar a conexão de aplicativos com o dispositivo disponível. Em outras palavras, a criação de aplicativos para suportar múltiplos protocolos é demorada, já que desenvolvedores devem implementar essa interação levando em consideração cada perfil específico do dispositivo e sua descrição própria.

3.4 Considerações Finais do Capítulo

Este capítulo apresentou algumas informações para consolidar mais a fundamentação teórica deste trabalho, como também, abordar as principais soluções para interoperabilidade entre dispositivos em um ambiente inteligente, e por fim, explorar problemas correlacionados com esse tema. Para tal, foi utilizado o método da Revisão Sistemática para isso. Primeiramente apresentou-se a metodologia utilizada durante o processo de revisão sistemática. Posteriormente, na seção 3.1 foram definidos os passos que forão utilizados para aplicar a revisão sistemática, elaborando assim, o protocolo para esta finalidade.

Depois de desenvolvido o protocolo da revisão, a Seção 3.2 é dissertado a etapa de

execução da revisão, onde foram identificados, selecionados e avaliados os dados referentes à pesquisa em questão, feito uma análise minuciosa dos trabalhos retornados do estado da arte sobre as questões definidas no protocolo.

Por fim, a seção 3.3, foram discutidos as respostas abordas no protocolo da revisão, consolidando as informações necessárias para o desenvolvimento deste trabalho, abordando problemas correlacionados com a interoperabilidade entre diferentes dispositivos em um mesmo ambiente. Logo, o próximo capítulo apresenta uma proposta para esse problema.

4 Proposta

Nesta seção, será apresentada uma proposta de solução de interoperabilidade entre diferentes protocolos de conectividade em um ambiente inteligente, com a finalidade de solucionar o problema abordado com as soluções existentes discutidas na revisão sistemática deste trabalho.

A principal dificuldade para alcançar a interoperabilidade entre diferentes protocolos de conectividade em ambientes inteligente é ocasionada pela falta de um consenso ou padronização de transmissão de dados dos protocolos de conectividades desenvolvidos para esse âmbito (REAZ, 2013; C. YANN B., 2013), ocasionando assim, a concepção de diferentes pesquisas desenvolverem diferentes protocolos de conectividade independentes umas das outras. Ou seja, dispositivos que usam um determinado protocolo de conectividade em um ambiente inteligente só poderá interagir com dispositivos que se utilizam do mesmo protocolo.

Uma vez provável que diferentes dispositivos e protocolos sejam amplamente utilizados no ambiente, sendo essa a principal razão da não cooperação e interligação desses diferentes dispositivos em um mesmo ambiente. Possuindo assim a necessidade de uma estrutura de interoperabilidade que permite que a heterogeneidade de dispositivos e protocolos coopere entre si.

Diversas propostas abordam que a solução da padronização desse tipo de protocolo como a maneira mais eficaz para resolver esse problema, sendo essa uma visão em longo prazo, pois, existem diferentes dispositivos e protocolos já sendo consumidos e utilizados nos dias de hoje.

Outras propostas abordam que a solução mais rápida e flexível para esse tipo de problema de interoperabilidade é baseada em uma camada de *middlewares* nos responsáveis por consumir os serviços disponibilizados pelos dispositivos, ou seja, nos pontos de controle. Contudo, nada é explorado ou abordado sobre os controladores já desenvolvidos, tanto nos pontos de controle que consomem os serviços dos dispositivos, como também,

nos dispositivos que também consomem serviços de outros dispositivos.

A iteração de um novo protocolo de conectividade em um ambiente inteligente limita o uso de soluções, já desenvolvidas, que trabalham somente em um determinado protocolo, ou em determinados casos, mesmo os pontos de controle que coagirem com camadas de *middlewares* que interoperabilizam diferentes protocolos de conectividade, precisaram ser reprogramados para atenderem a iteração desse novo protocolo, realizando modificações intrusivas diretamente no ponto de controle, sendo que diferentes soluções não disponibilizam a abertura do código para isso.

Ainda em determinados casos, não deparado até o momento desta pesquisa, dispositivos podem interagir com outros dispositivos de maneira inteligente para se adaptarem ao contexto do usuário, e essa iteração só poderá ser realizada se os dispositivos coagirem com o mesmo protocolo de conectividade, limitando o usuário a usar somente uma determinada tecnologia se o mesmo desejar essa iteração automática entre os dispositivos, que não é o aconselhável, pois o usuário deverá sempre ter liberdade de escolha de produtos que mais lhe possa ser útil, independente da tecnologia que o mesmo utiliza.

É esse ponto em específico que a proposta desse trabalho aborda, ou seja, desenvolver uma proposta de solução capaz de interoperabilizar os diferentes protocolos habitados em um mesmo ambiente inteligente, garantindo que pontos de controle e dispositivos já desenvolvidos, com um protocolo de conectividade específico, possam também interagir com outros protocolos de conectividade, reduzindo assim o esforço na utilização das aplicações já desenvolvidas e garantindo principalmente a interação entre dispositivos, independente do protocolo de conectividade que o mesmo utilize.

4.1 Uma solução baseada em Proxy

A proposta se baseia na introdução de um serviço *proxy* que permitirá a interoperabilidade entre protocolos de conectividade distintos sem alterar nenhum dos componentes já existentes no ambiente. Como pode ser observado na Figura 14, é adicionado um dispositivo físico na infraestrutura de comunicação que terá como objetivo principal a tradução dos pacotes de comunicação entre diferentes protocolos.

Apesar das semelhanças de alto nível entre diferentes protocolos de conectividade em um ambiente inteligente e os mesmos utilizar como base protocolos padronizados e consolidados da internet, DHCP, IP, HTTP, entre outros (FREITAS; TEIXEIRA, 2009)), a troca de informações dos serviços prestados por diferentes protocolos de conectividade e

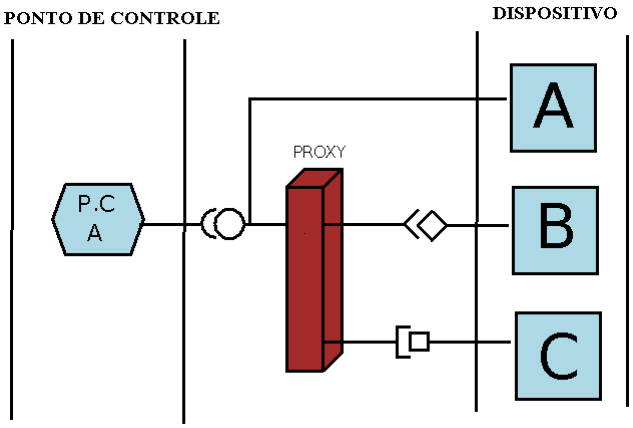


Figura 14: Proposta para interoperabilidade entre protocolos

a interoperabilidade entre eles não será simples.

Dispositivos surgem de diferentes fornecedores, com diferente interface de comunicação e protocolos de conectividade. Mesmo assim, esses precisam interoperar entre si devido à necessidade do usuário, obter em seu ambiente, diferentes tipos de dispositivos realizando funções específicas. Essa diferença de dispositivos implica diretamente na diversidade de protocolos de conectividade que surgem no ambiente e a interoperabilidade para esses diferentes protocolos envolvem, não apenas fornecer uma interconexão entre dispositivos, como também, conseguir intercomunicar e executar serviços disponibilizados pelos mesmos.

A principal dificuldade para alcançar essa interoperabilidade entre diferentes protocolos de conectividade é devido ao isolamento de sua infraestrutura pelo não consenso normalizado entre as partes envolvidas para esse tipo de tecnologia. Com isso, o desenvolvimento desses protocolos tende a não alcançar a exigência para a execução conjunta de tarefas disponibilizadas por outros dispositivos, a não ser que estejam utilizando de um mesmo protocolo de conectividade

Como forma de avaliar a aplicabilidade da solução proposta, será desenvolvido um estudo de caso considerando um cenário onde pontos de controle, já desenvolvidos e que utilizam um determinado protocolo, possam interagir com os dispositivos presentes no ambiente que utilizam um protocolo diferente do ponto de controle, obviamente, também controlar dispositivos que utilizam o mesmo protocolo.

A solução para interoperabilidade entre protocolos é, essencialmente, constituída por serviços virtuais que são instanciados automaticamente pelo *proxy*, como resposta à descoberta de um protocolo na rede. Para atingir este objetivo, é preciso "enganar" o ponto

de controle fazendo-o acreditar que o mesmo está interagindo com os serviços disponibilizados por um dispositivo que utiliza o seu mesmo protocolo de conectividade, mas que na verdade, são serviços disponibilizados por outro protocolo.

No cenário mencionado, o *proxy* criará dispositivos virtuais para cada dispositivo real descoberto no ambiente, registrando os serviços disponibilizados pelo mesmo. Por exemplo, um ponto de controle que utiliza do protocolo UPnP controlaria somente dispositivos UPnP, entretanto, o *proxy* ao descobrir no ambiente dispositivos que utilizem outro protocolo, como o DLNA por exemplo, instanciará um dispositivo virtual UPnP com as mesmas características que o DLNA, fazendo com que o ponto de controle UPnP possa interagir com esse dispositivo DLNA através do dispositivo virtual UPnP, como ilustrado na Figura 15.

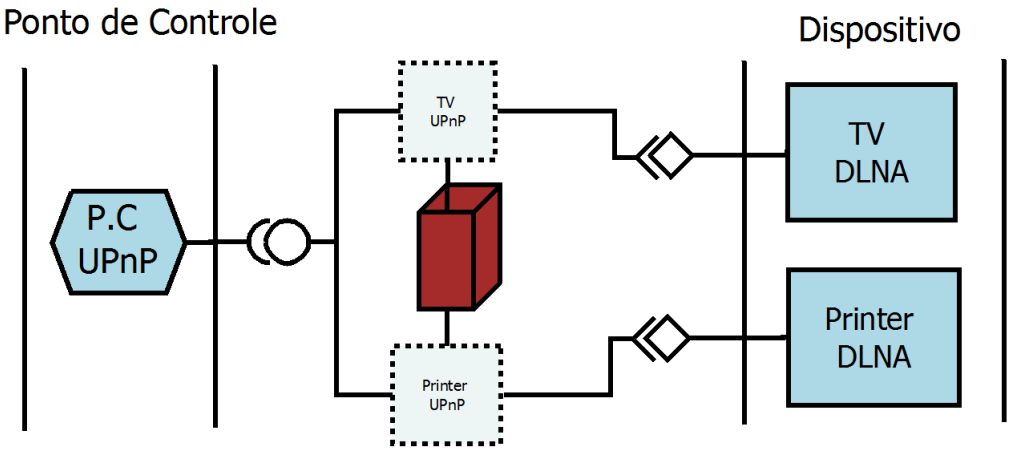


Figura 15: Proxy para interoperabilidade entre protocolos UPnP e DLNA
Fonte: Autor

Para o *proxy*, cada dispositivo com diferentes protocolos terá diretamente um único dispositivo virtual ligado e ele. Desta forma, qualquer ação realizada em qualquer um dos dispositivos, virtual ou real, será afetada diretamente no seu par, garantindo assim que as informações disponibilizadas pelos dispositivos estejam sempre atualizadas para quem estiver interessado nesse dispositivo. Para cada instância de serviço suportado, os dispositivos virtuais serão os responsáveis por concretizar a ponte entre os protocolos de conectividade e realizarem a interoperabilidade entre eles, sem a necessidade dessas informações transmitidas passarem pelo interior do *proxy*, garantindo assim a não centralização do tráfico das mensagem entre os dispositivos, diminuindo a complexidade envolvida na solução. Quando um dispositivo for removido da rede, o *proxy* destruirá automaticamente os dispositivos virtuais correspondentes. Para alcançar esse objetivo, a arquitetura da solução divide-se nos seguintes módulos:

- **Análise:** Considerado o núcleo principal do *proxy*, esse é o responsável por gerenciar os outros módulos do sistema, gerenciar os serviços disponibilizados pelos dispositivos, solicitar ao banco de dados informações sobre dispositivo adequado a ser virtualizado para realizar a tradução entre os diferentes protocolos, e por fim, gerenciar a virtualização dos mesmos;

- **Descoberta de Protocolos:** Esse módulo é responsável por descobrir na rede quais protocolos de conectividade existentes na mesma, tendo como principal objetivo, garantir que o *proxy* só inicialize a descoberta de dispositivos dos protocolos em questão, minimizando assim a sobrecarrega de tráfico de informação do *proxy* e em contrapartida, aumentando o desempenho da mesma;

- **Descoberta de Dispositivos:** Oferece funções de descoberta dos dispositivos e dos serviços disponibilizados pelos mesmos, sendo essas informações essenciais para transmitir as informações de um serviço real pelo dispositivo virtual. Esse módulo pode ser dividido em N módulos diferentes, um para cada protocolo de conectividade existente no ambiente e suportado pelo *proxy*;

- **Existência de Dispositivo:** Depois que o módulo de descoberta de dispositivo descobrir os dispositivos disponíveis no ambiente, esse módulo se encarregará em verificar a permanência do dispositivo na rede, com o único objetivo de garantir que sua virtualização só esteja disponível se o dispositivo real estiver também disponível no ambiente;

- **Virtualização de Dispositivos:** Componente usado para inicializar ou destruir os dispositivos virtuais no ambiente. Esse módulo também é responsável por garantir que cada dispositivo virtual tenha somente um dispositivo real vinculado a ele, garantindo que a concorrência criada pelas instâncias dos dispositivos virtuais possam ser independente uma das outras;

Para explanar o funcionamento de cada um dos passos gerenciados pelos componentes do *proxy*, é exemplificado o comportamento do *proxy* envolvendo quatro situações distintas. A primeira é representada desde o passo de inicialização do *proxy* no ambiente, descoberta dos protocolos de conectividade e dispositivos disponíveis no ambiente até sua virtualização. A segunda ocasião representa o comportamento do *proxy* quando um dispositivo real deixa de existir no ambiente em que o mesmo esteja conectado. A terceira ou quarta situação pode ocorrer de forma aleatória, depois que a primeira tenha acontecido, e são representadas quando algum Ponto de Controle consome um serviço disponibilizado

pelo dispositivo, seja esse, o real ou virtual. A Figura 16, ilustra o comportamento do *proxy* nas duas primeiras situações apresentadas.

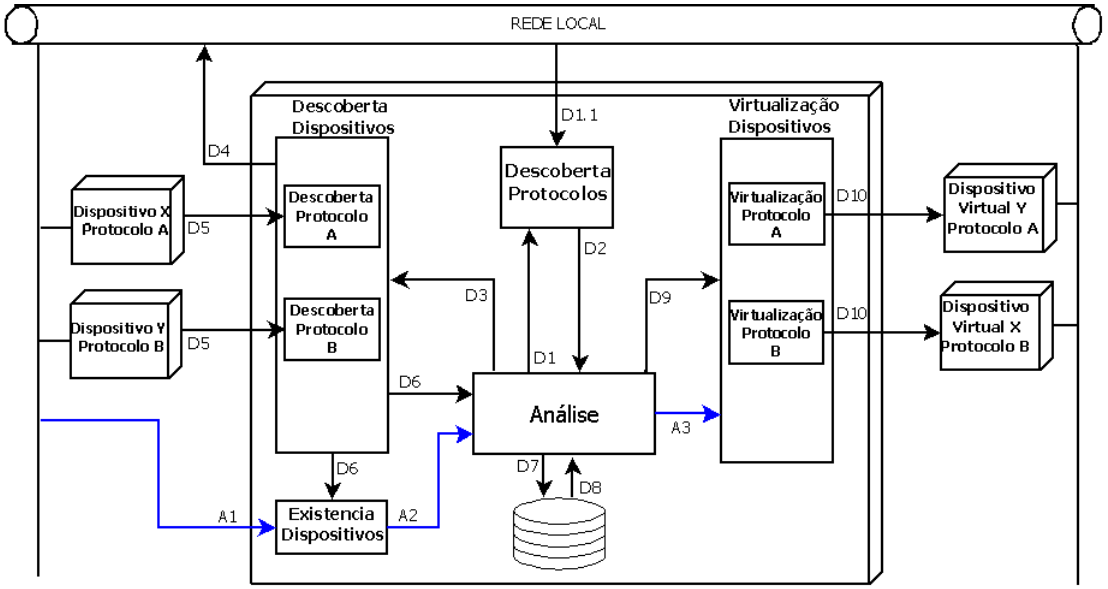


Figura 16: Comportamento funcional dos módulos do proxy
Fonte: Autor

Cada um dos passos apresentados, que estão representados por dois dígitos, uma letra e um número, representa o comportamento do *proxy* nas possíveis situações existentes. O primeiro comportamento é dado quando o *proxy* é introduzido na rede do ambiente até a virtualização dos dispositivos, dentre esse intervalo, o *proxy* irá realizar as buscas pelos protocolos de conectividade existentes no ambiente e capturar as informações necessárias dos dispositivos disponibilizados pelos mesmos, concretizando o procedimento de virtualização quando necessário. Essa primeira situação é dada pelos seguintes passos:

- **Passo D1:** Quando o *proxy* é inicializado no ambiente, o primeiro comportamento do mesmo é inicializar o módulo de Descoberta de Protocolos afim de descobrir quais protocolos de conectividade estão disponibilizados no ambiente;

- **Sub-Passo D1.1:** Depois de inicializado o módulo de Descoberta de Protocolos, esse comportamento é responsável por ficar "escutando" as informações que são transmitidas na rede e descobrir quais protocolos de conectividade coexistem no ambiente. Esse comportamento é independente dos comportamentos sequenciais apresentado nessa situação, pois, esse módulo sempre ficará verificando a transmissão das informações na rede a fim de descobrir a existência de nossos protocolos de conectividade, que são suportados pelo *proxy* e que podem ser solicitados a qualquer momento pelo módulo de Análise;

- **Passo D2:** Quando um novo protocolo de conectividade é descoberto no ambiente, o módulo de Descoberta de Protocolos notifica ao módulo de Análise qual novo protocolo foi descoberto no ambiente;

- **Passo D3:** Com a descoberta dos protocolos de conectividade disponíveis no ambiente, o módulo de Análise solicita ao módulo de Descoberta de Dispositivo, a inicialização da descoberta dos dispositivos disponíveis no ambiente, referente somente aos protocolos que foram encontrados no ambiente pelo módulo de Descoberta de Protocolos;

- **Passo D4:** O módulo de Descoberta de Dispositivo solicita a rede conectada de maneira *multicast*, à descoberta dos dispositivos disponíveis na mesma. Tal solicitação é feita de maneira única pelo próprio módulo, entretanto, essa solicitação é referente a todos os protocolos de conectividade descoberto pelo módulo responsável;

- **Passo D5:** Cada dispositivo, depois de ser solicitada sua descoberta, enviar uma resposta de forma *unicast*, ao modulo que contem o seu mesmo protocolo de conectividade. Tal responsa contém as informações necessárias para o acesso aos serviços disponibilizados por cada dispositivo. Da mesma forma que cada sub-módulo de Descoberta de Dispositivo é independente de outro sub-módulo, por esses serem processos diferentes, independentes e trabalharem com protocolos diferentes, esse mesmo passo representa as descobertas dos dispositivos disponíveis no ambiente, independente dos protocolos existentes. Contudo, somente depois da captura das informações de todos os dispositivos disponíveis, é que o módulo de Descoberta de dispositivo passará para o próximo passo;

- **Passo D6:** Depois da descoberta das informações responsáveis pelo acesso dos serviços disponibilizados pelos dispositivos, o módulo de Descoberta de Dispositivo envia essas informações para que o módulo de Análise possa solicitar quais dispositivos virtualizar. Essas mesmas informações também são passadas para o módulo de Existência de Dispositivos que será explicado posteriormente na segunda situação abordado pelo *proxy*;

- **Passos D7-D8:** Com as informações dos dispositivos reais encontrados, esses passos são responsáveis por solicitar ao banco de dados as informações sobre os tipos de dispositivos disponíveis no *proxy*, para assim, instanciar o dispositivo virtual adequado, ou aquele dispositivo que melhor se adapte ao dispositivo real encontrado. Essas informações serão referentes ao tipo de dispositivo encontrado, quantos

e quais serviços esses dispositivos estão disponibilizando no ambiente e quais tipos dos valores das variáveis de estado desses serviços;

- **Passo D9:** Depois que o módulo de Análise decidi qual o melhor dispositivo virtual a ser instanciado pelo *proxy*, essas informações serão enviadas ao módulo de Virtualização, esse por ser o responsável em instanciar e controlar os dispositivos virtuais no ambiente. Tal módulo de Virtualização dispõe de sub-módulos responsáveis pela virtualização de um protocolo específico, como pode ser observado ainda na Figura 3. O sub-módulo Virtualização de Protocolo A, será o responsável em criar um dispositivo virtual capaz de fornecer serviços do tipo de protocolo A, mas que na verdade são serviços disponibilizados por um dispositivo de protocolo B. Ou seja, o dispositivo Y que contém o protocolo B, será virtualizado pelo sub-módulo de Virtualização de Protocolo A, para um dispositivo virtual Y que agora contém também o protocolo A, e esse será o responsável por traduzir as informações que são consumidas no dispositivo virtual Y do protocolo A, para o dispositivo real Y do protocolo B;

- **Passo D10:** Cada sub-módulo de Virtualização referente ao seu determinado protocolo instancia o dispositivo virtual no ambiente;

A segunda situação aborda, ainda observado na Figura 3, é dado quando um dispositivo real, já virtualizado, é removido da rede. Desta forma, o módulo de Existência de Dispositivo tem o objetivo de observar a presença dos dispositivos na rede, informações essas sobre os dispositivos fornecidos a esse módulo pelo módulo de Descoberta de Dispositivo, referente ao **Passo D6**. Os passos referentes a esse comportamento são:

- **Passo A1:** O módulo de Existência fica sempre escutando na rede se os dispositivos que foram encontrados pelo módulo de Descoberta de Dispositivos permanecem na mesma, como também, disponibiliza essas informações de maneira independente ao módulo de Análise;

- **Passo A2:** Quando um dispositivo é removido da rede, o módulo de Existência repassa essa informação ao módulo de Análise.

- **Passo A3:** O módulo de Análise envia a solicitação ao módulo de Virtualização que destrua todos os dispositivos virtuais referentes ao dispositivo real que não existem mais no ambiente, garantindo assim, que um ponto de controle ou dispositivo não

seja enganado sobre a existência de um determinando dispositivo que não permanece mais na rede.

Depois de realizada a virtualização dos dispositivos no ambiente, a segunda e terceira situação que o *proxy* poderá apresentar é determinada quando um serviço disponibilizado por um dispositivo é consumido, ou seja, quando esse comportamento representa uma ação de mudança de valor na variável de estado dos serviços do dispositivo, seja esse real ou virtual.

Esses comportamentos são realizados de maneira similar um com o outro e o responsável direto por essa tradução de informação entre os diferentes protocolos de conectividade, é o próprio dispositivo virtual. Esse terá em seu núcleo, dois sub-módulos, um para traduzir as informações consumidas no protocolo de conectividade do dispositivo virtual para o dispositivo real, e o outro, um tradutor de informação que fica escutando quando um serviço for consumido dispositivo real e relatar essas informações para seus interessados no protocolo do dispositivo virtual, como observado na Figura 17;

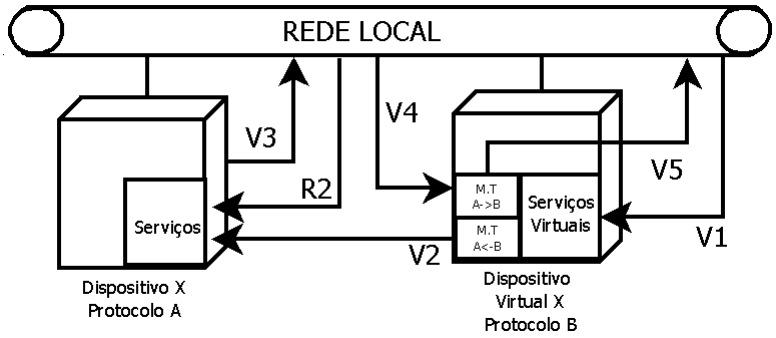


Figura 17: Consumo de um serviço no dispositivo real ou virtual
Fonte: Autor

Os passos referentes a esse comportamento são:

- **Passo V1:** Esse passo representa quando um serviço é consumido por um ponto de controle que seja compatível com o protocolo de conectividade que o dispositivo virtual esteja disponibilizando;
- **Passo V2:** O módulo de tradução (M.T A⇐B) envia ao dispositivo real a solicitação de consumo da representação do serviço solicitado ao dispositivo virtual, mudando assim suas variáveis de estado do respectivo serviço. Com isso, o dispositivo real e dispositivo virtual destinado a ele estarão sempre com os valores de suas variáveis de estado atualizadas e idênticas;

- **Passo V3:** Após o serviço ser consumido no dispositivo real, esse enviará uma mensagem, utilizando o tipo de seu protocolo, para a rede, informando o acontecido para aqueles que estão interessados nesse serviço em específico;

- **Passo V4:** O módulo de tradução (M.T A⇒B), que é o responsável de verificar qualquer alteração nos serviços disponibilizados pelo dispositivo real, ao receber essa notificação de consumo em algum dos seus serviços, traduz a informação do protocolo específico do dispositivo real, para o protocolo específico do dispositivo virtual;

- **Passo V5:** Envio à rede a notificação que o serviço, no dispositivo virtual foi consumido. Essa notificação deverá ser realizada somente depois que o módulo de tradução (M.T A⇒B) receber a notificação do dispositivo real, garantindo assim, que os valores das variáveis de estado de cada dispositivo estejam sempre atualizados e sincronizados;

Por último, quando um serviço é consumido diretamente no dispositivo real por algum ponto de controle que entenda o mesmo protocolo disponibilizado pelo mesmo, a situação é representada pelo **Passo R2**. Entretanto, o comportamento do *proxy* ainda continua idêntico a situação anterior, quando o serviço é consumido no dispositivo virtual. O **Passo V3**, notifica a rede do consumo realizado no dispositivo real, o passo **Passo V4** recebe a notificação do consumido realizado, traduz usando o sub-módulo (M.T A⇒B), e o mesmo, notifica a rede utilizando o **Passo V5**, que o serviço no dispositivo virtual, mas que na verdade foi um serviço direto no dispositivo real, desta forma os interessados por esse serviço possa realizar suas devidas ações.

Para cada dispositivo novo no ambiente não suportado pelo *proxy*, uma quantidade modesta de código deverá ser escrito para adaptá-lo para seu reconhecimento, desta forma, os dispositivos e seus Pontos de Controle não necessitam de modificação alguma. Ou ponto importante que deve ser realçado é o fato que os passos apresentados até o momento, demostra, de maneira geral, o comportamento do *proxy* nas situações existentes. E que esses comportamentos serão adaptado, criando sub-passos de acordo com os protocolos incorporados pelo *proxy*, mas não perdendo sua essência genérica.

4.2 Considerações Finais do Capítulo

Este capítulo apresentou uma proposta de solução para interoperabilidade de dispositivos em ambientes inteligentes, com o objetivo de solucionar o problema encontrado na revisão sistemática abordada neste trabalho.

Primeiramente, foi discutido o principal problema para alcançar a interoperabilidades de diferentes protocolos de conectividade em um mesmo ambiente, com base nesse problema, depois foram analisadas as propostas da literatura para solucionar esse problema. Com base nessa análise, foi relatado também o problema de consolidação nas soluções existentes, ou seja, as soluções propostas no estado da arte, ainda confinam-se em um protocolo específico para a solução de interoperabilidade.

Contudo, neste capítulo, é discutido esse problema em específico e é proposta uma solução alternativa que sobrepõem o problema encontrado nas soluções encontradas, em que, na proposta apresentada, o *proxy* não depende de nenhum protocolo em específico, pois, o mesmo irá realizar a tradução e interoperabilidade dos diferentes protocolos de conectividade no ambiente.

5 Implementação da solução baseada em Proxy

Para comprovar a proposta abordada nesse trabalho, foi implementado um modelo do *proxy* concentrado na interoperabilidade entre os protocolos UPnP e DLNA, estes por serem os protocolos com maior destaque resultante da pesquisa realizada pela revisão sistemática deste trabalho.

Nesta seção, será apresentada o desenvolvimento do *proxy* abordando tais protocolos de conectividade. Para esse feito, primeiramente é identificado as principais diferenças entres os mesmos, qual a melhor estratégia abordada para alcançar a interoperabilidade entre esses protocolos de conectividade, e por último, um estudo de caso abordando o funcionamento do comportamento do *proxy* em uma determinada situação e o processo de desenvolvimento do mesmo.

5.1 Diferenças entre os Protocolos UPnP e DLNA

O Objetivo principal do protocolo DLNA, e o compartilhamento de conteúdo digital. Entretanto, funcionalidades de descoberta, controle e configuração, também do dispositivo, fazem parte de suas características. Essas funcionalidades são atribuída pela utilização da pilha do protocolo UPnP. Contudo, a Aliança responsável pelo DLNA necessitava que os dispositivos que abordassem o seu protocolo somente interligassem com outros dispositivos com o mesmo protocolo, impondo assim que sua tecnologia deverá ser um padrão a ser abordado para esse ambiente, tanto no controle de dispositivos como no compartilhamento de mídia.

Para alcança esse objetivo e almejando exibir sua principal característica da não criação de um novo protocolo, o DLNA criou uma camada modificando duas características da pilha UPnP (VENKITARAMAN, 2008). O serviço de descoberta de um dispositivo e a sintaxe XML da descrição do dispositivo e serviços.

Diferente do protocolo UPnP que utiliza o SSDP para o serviço de descoberta do dispositivo utilizando o endereço e porta **239.255.255.250:1900** para endereçamento *multicast IP*, o DLNA utilizando o mesmo protocolo SSDP com um endereço *multicast* **228.255.255.250:1900**. Fato esse importantíssimo para descoberta dos dispositivos DLNA e não mencionado na documentação disponível no Web Site. Entretanto, foi descoberta esse endereço *multicast* com o auxílio da ferramenta Wireshark[1], como observado na Figura 18.

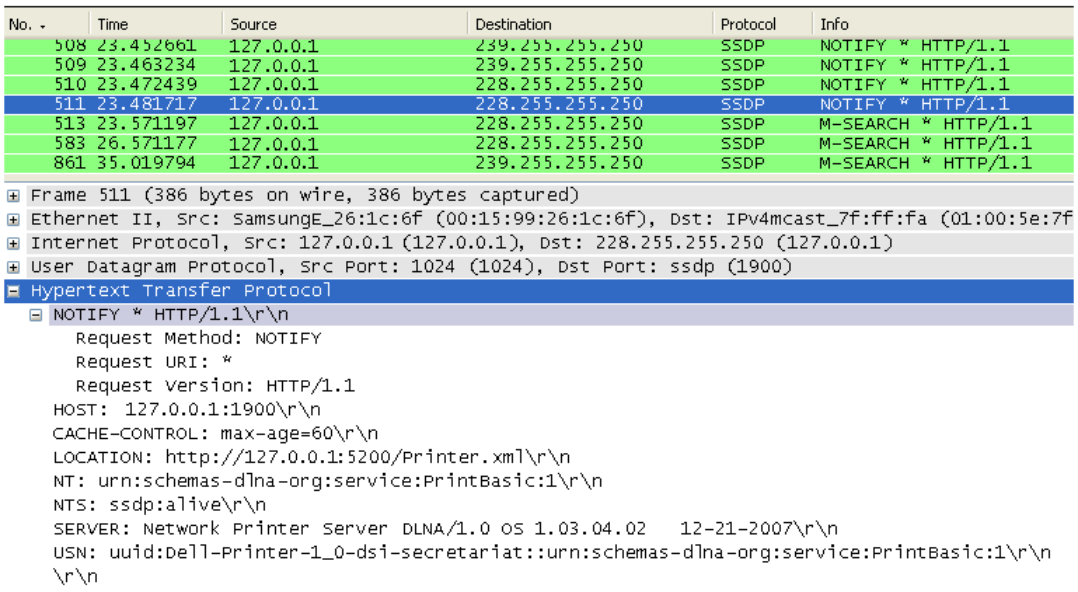


Figura 18: Mapeamento da rede multicast com a ferramenta Wireshark
Fonte: Autor

Nesse momento foram instanciados dois dispositivos virtuais, uma impressora utilizando o protocolo DLNA e uma lâmpada utilizando o protocolo UPnP. A ferramenta *Wireshark* foi limitado a pesquisar somente a faixa de IP referente ao endereçamento *multicast* que são entre as faixa de IP 224.0.0.0 a 239.255.255.255. Ela encontrou dois IP *multicast* diferentes, um que inicia com 239.255.255.250, que este é referente a a lâmpada UPnP, e outro IP 228.255.255.250, referente a impressora DLNA, como ainda observado com detalhes sua descrição da descoberta.

A segunda diferença com entre os dois protocolos, é com relação na sintaxe apresentada na etapa de descrição dos dispositivos. Além de o DLNA utilizar uma especificação própria para descrever o tipo de dispositivos como os serviços disponibilizados por ele, algumas *tags* utilizadas na linguagem de marcação XML do DLNA são diferentes das *tags* apresentadas pelo UPnP. A Figura 19 apresenta, resumidamente, uma descrição de uma

[1]http://www.wireshark.org/

TV com o protocolo DLNA.

```
<?xml version="1.0"?>
<root xmlns="urn:schemas-dlna-org:device-1-0" xmlns:dlna="urn:schemas-dlna-org:device-1-0"
xmlns:sec="http://www.dlna.org/dlna">
<dlna:device>
        <dlna:Type>urn:schemas-dlna-org:device:tv:1</dlna:Type>
        <dlna:Name>DLNA TV Device</dlna:Name>
        ...
        <dlna:servicesList>
                <dlna:service>
                        <serviceType>...device:power</serviceType>
                        <serviceId>...</serviceId>
                        <SCPDURL>.../contentDirectory.xml</SCPDURL>
                        ...
                </dlna:service>
                <dlna:service>...</dlna:service>
        </dlna:servicesList >
...
</dlna:device>
```

Figura 19: Descrição resumida de uma TV-DLNA
Fonte: Autor

As *tag* da descrição do dispositivos DLNA, as que contém as informações com base na especificação padronizada pelo DLNA, são diferentes da do UPnP. Por exemplo, a *tag* <dlna:device> que contem a informação do esquema padronizado para o dispositivo, e a *tag* <dlna:servicesList>, que contem as informações de quantos e quais os serviços que o dispositivo está disponibilizando, são descritas pelo o UPnP pela *tags* <deviceType> e <serviceList>. Como também, algumas *tags* referentes a descrição dos serviços são modificadas pelo DLNA, essas informações serão abordadas com mais detalhes na proposta deste trabalho.

5.2 Estratégia para interoperabilidade dos protocolos DLNA e UPnP

Para realizar a interoperabilidade entre diferentes protocolos, o *proxy* deverá adotar estratégias específicas para cada par de protocolos. Essas estratégias podem variar de acordo com o protocolo em questão, modificando o comportamento do *proxy* para a descoberta do protocolo, como já exemplificado na Figura **??** da seção anterior, ou como cada módulo irá se comportar internamente para realizar a ponte entre os protocolos. O Módulo de Análise é o responsável direto por realizar essa operação entre os diferentes protocolos, abordando estratégias específicas de acordo com as especificações de cada protocolo analisado.

Em relação à interoperabilidade entre os protocolos DLNA e UPnP, o Módulo de Análise realizará duas estratégias específicas. Primeiramente na descoberta do dispositivo DLNA até sua virtualização como UPnP, como também na descoberta de quantos e quais serviços o dispositivo DLNA disponibiliza no ambiente para também integrá-los no dispositivo virtual UPnP.

O primeiro comportamento interno do *proxy* é a descoberta de todas as informações referentes aos dispositivos DLNA no ambiente. Informações como a descrição do dispositivo, e quantos e quais serviços esses dispositivos estão disponibilizando no ambiente. Essas informações serão coletadas pelo Módulo de Descoberta ao encontrar no ambiente um dispositivo DLNA que ainda não o tenha encontrado. Ou seja, quando o *proxy*, depois da descoberta do dispositivo e sua interação com o mesmo, receber uma mensagem SSDP do dispositivo DLNA, ele então analisará o seu cabeçalho USN, que contém o endereço referente as informações detalhadas do dispositivo e a solicitará.

Com essas informações coletadas, que são oferecidas pelo dispositivo em um formato XML específico do protocolo DLNA, o Módulo de Descoberta enviará ao Módulo de Análise essas informações que são necessárias para que o último possa iniciar o procedimento de análise do dispositivo. Um exemplo resumido dessa descrição com as informações em destaque que serão então analisadas pelo módulo pode ser observado resumidamente no esquema XML de uma TV-DLNA na Figura 20.

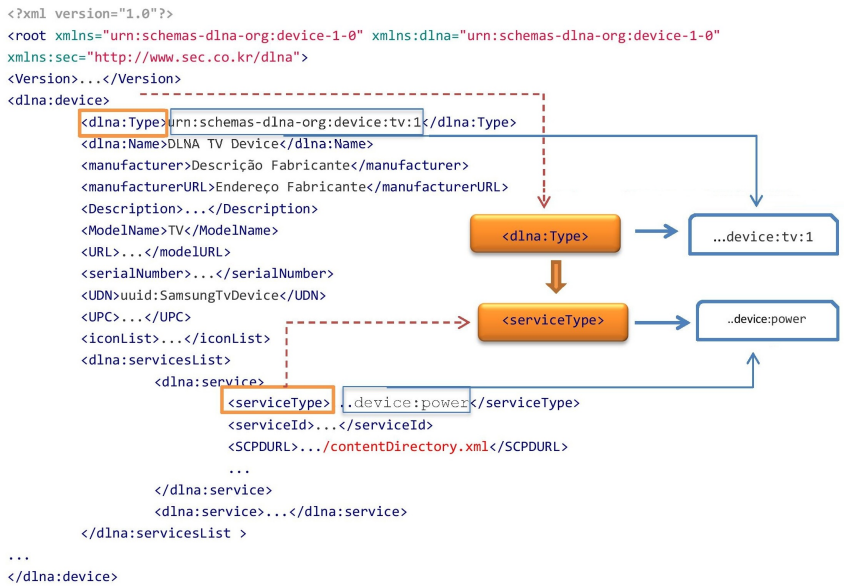


Figura 20: Esquema XML da descrição de um TV-DLNA
Fonte: Autor

Desta forma, o Módulo de Análise terá todas as informações necessárias para comparar

o dispositivo UPnP equivalente. Essas informações coletadas já podem ser consideradas suficientes para virtualização do dispositivo, isso devido à especificação obrigatória e padronizada para utilização do protocolo DLNA. Este documento especifica as diretrizes que definem um dispositivo DLNA para que seja compatível com qualquer Ponto de Controle DLNA desenvolvido, facilitando e globalizando o seu uso. Por exemplo, existem vários perfis para utilização de uma TV-DLNA, um perfil com operações básicas de uma TV: ligar/desligar; aumentar/diminuir volume; mudar canal, e outros perfis com operações mais avançadas, como de acesso às configurações da TV e compartilhamento de imagem e vídeo.

Na prática, não somente o protocolo DLNA segue obrigatoriamente uma especificação padronizada. O protocolo UPnP também obedece essa obrigatoriedade. São seguindo essas especificações que o *proxy*, em específico o Módulo de Análise, adota como base para comparar os dispositivos DLNA com o UPnP que deverá ser virtualizado.

Com essas informações coletadas pelo *proxy*, este já tem autonomia suficiente para virtualizar um dispositivo UPnP compatível, como exemplificado em um dos três cenários possíveis, a descoberta do dispositivo DLNA até sua virtualização como UPnP. Entretanto, para garantir a integridade e interconectividade também dos serviços disponibilizados pelos dispositivos, real e virtual, algumas informações ainda necessitam ser capturadas pelo Módulo de Análise. Essas informações são referentes às ações de mudança de valor na variável de estado dos serviços disponibilizados pelo dispositivo DLNA, como essas também são padronizadas pela utilização dos protocolos em formato XML, o próprio módulo tem autonomia suficiente para capturar essas informações diretamente do dispositivo. Como observado também resumidamente na Figura 21.

Este esquema XML determina a descrição do serviço disponibilizado pelo dispositivo no exemplo da TV-DLNA, recuperada do esquema XML a partir da *tag* <SCPDURL> .../contentDirectory.xml </SCPDURL>, do tipo de serviço <serviceType>...device:power </serviceType> apresentado na Figura 20.

Essa descrição determina quais ações e seus argumentos o serviço disponibiliza, define também o estado das variáveis de cada ação e os tipos de dados de cada variável. Para o esquema ilustrado, são destacado quais ações esse serviço está disponibilizando, <name> GetPower </name>; qual o estado da variável dessa ação, <relatedStateVariable> Result </relatedStateVariable>; e qual o tipo de dados de cada ação, <dataType>boolean </dataType>.

Com a coleta de todas essas informações do dispositivo DLNA, o Módulo de Análise irá

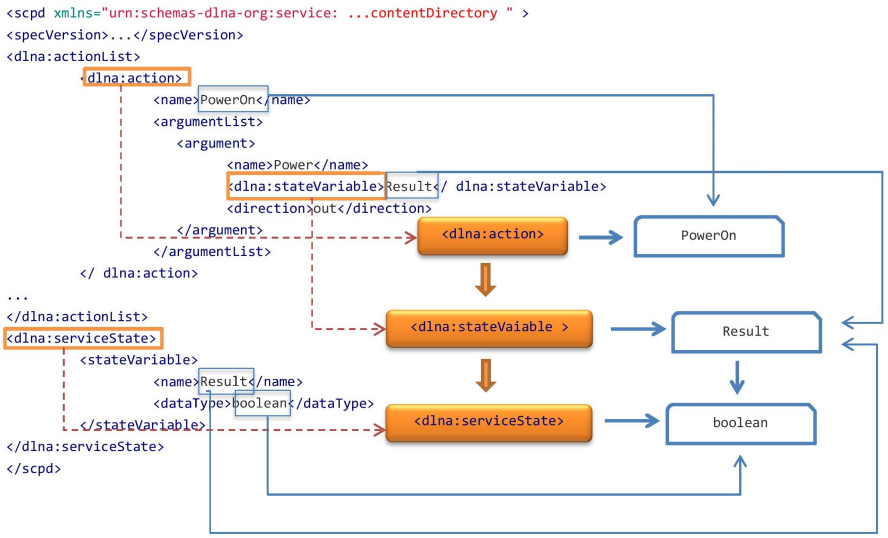


Figura 21: Esquema XML da descrição de serviço DLNA
Fonte: Autor

instanciar o dispositivo virtual UPnP adequado a partir do Módulo de Virtualização, ou seja, aquele dispositivo UPnP que poderá disponibilizar os mesmos serviços prestados pelo dispositivo DLNA, ou a maior parte deles. Isso porque em algum momento um dispositivo DLNA poderá ter mais funcionalidades ou serviços que um dispositivo UPnP existente, ou o oposto. Caso esse comportamento aconteça, será instanciado um dispositivo virtual que mais se aproxime da quantidade de serviços disponibilizados, levando em consideração o relato dessa informação ao Ponto de Controle utilizado para consumir esses serviços.

Depois de adquiridos essas informações do dispositivo DLNA, e virtualizado o dispositivo virtual UPnP, o Módulo de Análise se responsabilizará em vigiar esses dois dispositivos em particular. Essa vigilância se dará na similaridade de todas as informações adquiridas pelo processo de descoberta até a sua virtualização, com relação às *tags* correspondentes entre os protocolos, como melhor observado na Figura 22.

Juntamente com a coleta dessas informações, o Módulo de Análise irá comparar as variáveis de estado do dispositivo real com o virtual. Assim, quando um serviço for consumido, independente de qual protocolo, o módulo irá verificar qual serviço foi consumido, e se alguma variável de estado foi modificada. Caso essa variável tenha seu valor modificado, o módulo irá consumir esse mesmo serviço passando o mesmo valor da variável no dispositivo oposto, garantindo assim, a interoperabilidade entre os protocolos.

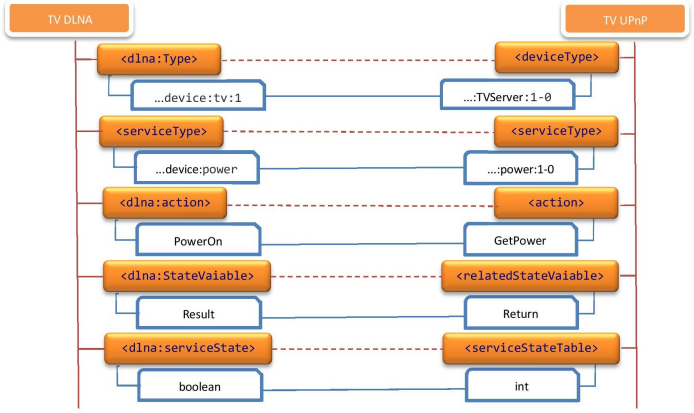


Figura 22: Similaridade dos esquemas XML dos dispositivos
Fonte: Autor

5.3 Estudo de Caso e Implementação da Solução

Nesta seção será abordado o trabalho desenvolvido, de maneira prática, inicializado pelo cenário em que o *proxy* será incorporado no ambiente, qual atividade que o mesmo irá se comportar em determinada situação e as etapas do seu desenvolvimento.

A proposta, desenvolvido utilizando a linguagem JAVA e as bibliotecas LibDLNA[2] e Cling[3], respectivamente, para conectividade dos dispositivos DLNA e UPnP, abordou a seguinte situação ilustrado na Figura 23.

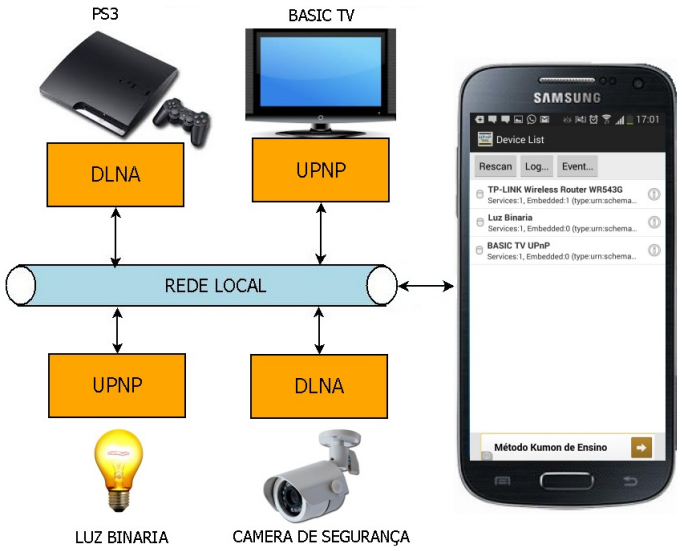


Figura 23: Cenário da aplicabilidade da proposta
Fonte: Autor

[2]http://libdlna.geexbox.org/
[3]http://4thline.org/projects

Em uma rede estão conectados quatros dispositivos, dois dispositivos com protocolo DLNA e dois dispositivo com protocolo UPnP. Respectivamente, uma câmera de segurança e um console PS3, ambos DLNA, uma televisão e uma lâmpada, ambas UPnP. Também conectado a rede, está um smartphone android com um aplicativo UPnP Tool[4] capaz de realizar a iteração entre dispositivos UPnP em uma mesma rede. O usuário deseja interagir todos os dispositivos presentes em sua casa para realizar a segurança da mesma, da seguinte forma:

- Toda imagem capturada pela câmera que representa uma detecção de movimento, deverá ser gravado em um pelo console PS3 em seu HD interno. Essa iteração poderia ser realizar se o usuário tivesse um aplicativo que interagisse somente com dispositivos DLNA localizados na rede, pois os dispositivos que ele gostaria de interagir no presado momento entendem o mesmo protocolo, mas o usuário deseja interagir com os outros dispositivos no ambiente;
- Quando a câmera detectar qualquer movimentação, o usuário também quer que a TV, onde o mesmo estivesse mais próximo, exibisse independente do que o mesmo estivesse assistindo, uma pequena projeção da movimentação detectada pela câmera e que acendesse todas as luzes localizadas próximas à movimentação detectada pela câmera. Essa interação já não seria possível, pois o aplicativo entende somente o protocolo UPnP, e o mesmo não reconhece na rede os outros dispositivo que utilizam o protocolo DLNA;

Outro cenário possível para a usabilidade do *proxy*, seria pelos próprios desenvolvedores de aplicativos. A fim de minimizar o esforço no desenvolvimento de aplicativos para entender vários protocolos, o desenvolver poderia utilizar o *proxy* para realizar a busca dos dispositivos já disponíveis no ambiente, independente dos protocolos vinculados ao mesmo, e com isso, o desenvolvedor iria focar somente a inteligência envolvida na iteração entre os dispositivos.

Desta forma, para que o usuário ou desenvolvedor possam interagir com todos os dispositivos no ambiente, ou o aplicativo já deverá ser possível interagir com os protocolos de conectividade envolvidos no ambiente, que não é o caso do cenário mencionado, pois o aplicativo interage somente com dispositivos UPnP, ou, utilizando a solução que é a proposta deste trabalho, como ilustrado na prática a execução do *proxy* nesse cenário na Figura 24.

[4]https://play.google.com/store/apps/details?id=com.tjjang.upnptool

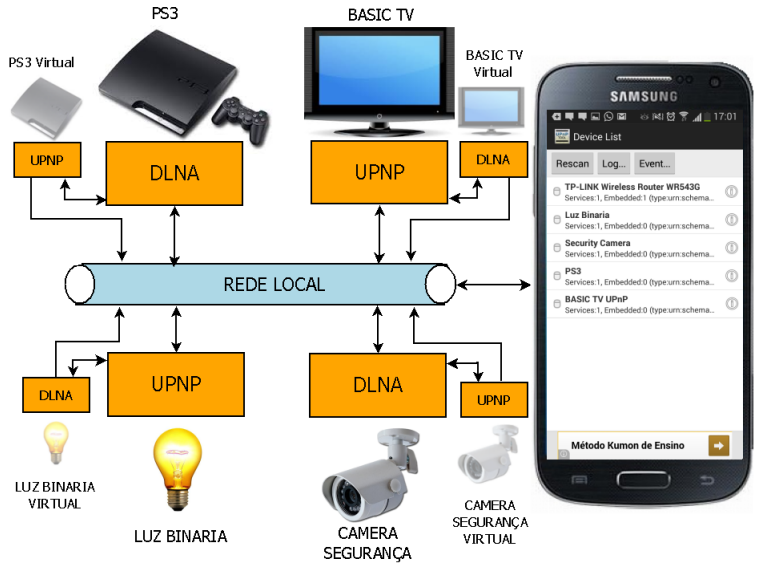


Figura 24: Cenário da aplicabilidade da proposta
Fonte: Autor

Com a ativação do *proxy*, o usuário pode realizar a interação entre todos os dispositivos conectados no ambiente, que no cenário mencionado, suportam os protocolos DLNA e UPnP. Vale ressaltar que o objetivo deste trabalho é relacionado diretamente na camada de conectividade entre diferentes protocolos de conectividade, deixando-os todos os dispositivos disponíveis para o usuário ou desenvolvedor possam interagir com os mesmos, e não em como será feita essa interação entre os mesmos.

O processo de desenvolvimento atende as especificações dos protocolos DLNA e UPnP, como já mencionado anteriormente, e aplica a estratégia abordada para a interoperabilidade dos mesmos. Para alcançar esse objetivo, o *proxy* segue a arquitetura ilustrada na Figura 25 com base na estratégia abordada na seção 5.2.

O primeiro comportamento realizado pelo *proxy* através do módulo de Análise, é a descoberta dos protocolos de conectividade envolvidos no ambiente. Esse tarefa é responsabilidade da interface DescobertaProtocolos que utiliza mais duas interfaces em comum,DescobertaDLNA e DescobertaUPnP, cada um responsável por utilizar as bibliotecas específicas para iteração com seus protocolos. A interface DLNAListener, para protocolo DLNA e a interface RegistryListener para o protocolo UPnP. Essa estratégia da criação de interfaces únicas para trabalharem com protocolos diferentes é uma maneira de padronizar um comportamento específico para trabalhar com diferentes protocolos, que é o caso abordado.

Cada interface DescobertaDLNA e DescobertaUPnP, implementa a interface Runnable do JAVA, assim, mesmo as duas interfaces sendo utilizadas por um mesmo módulo, o

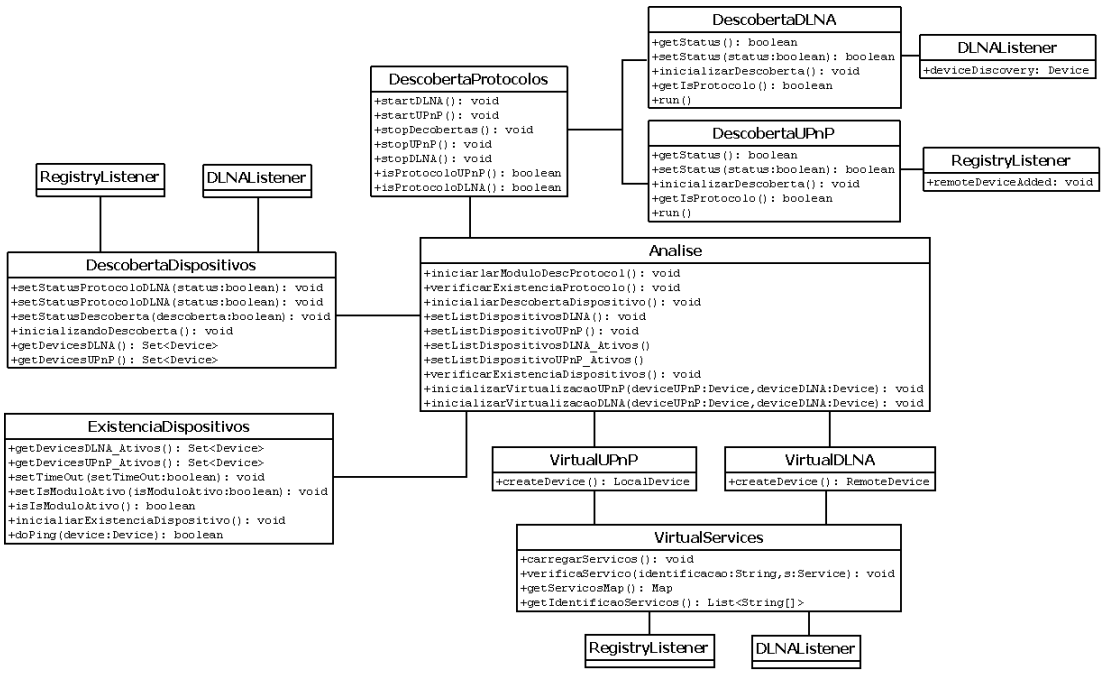


Figura 25: Diagrama de Classe do desenvolvimento do Proxy
Fonte: Autor

DescobertaProtocolos, cada uma trabalha em seus processos de maneira paralela e independe umas das outras, como pode ser observado, parcialmente, o desenvolvimento da classe DescobertaDLNA na Figura 26.

A classe, DescoberdaDLNA, implementa a interface Runnable, garantindo que o processo que a mesma trabalhe é independente do processo realizado pela classe DescobertaUPnP. As linhas, 20 e 21, são referencias aos atributos responsáveis por, respectivamente, isProtocolo, um atributo responsável por validar a descoberta do protocolo, se for do tipo true, existe no ambiente dispositivos que utilizam o protocolo DLNA, se false, ausência do mesmo, e o atributo status também do tipo boolean, que define o estado da descoberta do protocolo, se true, o processo responsável pela descoberta está ativo, se false, este processo está pausado. Esse último atributo também garante que o módulo de Análise, poderá em algum momento, cessar o processo de descoberta de um protocolo de conectividade quando achar necessário.

A linha 22, define a utilização da biblioteca DLNAListener, que é responsável por interagir com dispositivos DLNA no ambiente. Tal interface implementa vários métodos responsáveis por mantém um registro de todos os dispositivos descoberto na rede. Ele gerencia anúncios de descoberta e manipulação de eventos em segundo plano. Entretanto, a única responsabilidade nesse processo de desenvolvimento, é a descoberta do protocolo, desta forma, se alguma dispositivo for descoberto, a única ação que esse processo irá

```
18  public class DescobertaDLNA implements Runnable, IDescobertaDLNA {
19
20      private boolean isProtocolo;
21      private boolean status;
22      private final DLNAListener desobertarDLNA;
23      public DescobertaDLNA(boolean status) {
24          this.isProtocolo = false;
25          this.status = status;
26          desobertarDLNA = new DLNAListenerImpl() {
27
28              @Override
29              public void deviceDiscovery(Registry registry, RemoteDevice device) {
30                  System.out.println(
31                          "Dispositivos DLNA disponíveis: " + device.getDisplayString());
32
33                  isProtocolo = true;
34                  System.out.println("----------------------------------------");
35              }
36          };
37      }
```

Figura 26: Listagem da Classe DescoberdaDLNA
Fonte: Autor

realizar, é mudar o estado do valor do atributo isProtocolo, que por padrão é falso (linha 24), para true. Assim, o módulo de Análise a parti do módulo de DescobertaProtocolo, ao descobrir um mudança na variável de estado desse atributo, possa realizar suas devidas ações.

O método inicializardescoberta(), ainda da interface IDescoberdaDLNA é o responsável por inicializar a descoberta do protocolo no ambiente, como pode-se observar na Figura 27.

```
43      public void inicializarDescoberta() {
44          DLNAService dlnaDescoberta;
45          System.out.println("Inicializando Descoberta DLNA");
46          dlnaDescoberta = new DLNAService(desobertarDLNA);
47          System.out.println("----------------------------------------");
48          //Escuta todas as mensagem que os dispositivos disponibilizam no ambiente
49          dlnaDescoberta.getDiscovery().search(new STAllHeader());
50          System.out.println("----------------------------------------");
51          while (getStatus() == true) {
52              try {
53                  Thread.sleep(10000);
54              } catch (InterruptedException ex) {
55                  dlnaDescoberta.shutdown();
56              }
57          }
58          System.out.println("----------------------------------------");
59          //Libera todos os recursos e anunciar ByeBye a outros dispositivos DLNA
60          System.out.println("Parando Descoberta DLNA...");
61          dlnaDescoberta.shutdown();
62      }
```

Figura 27: Listagem do método inicializarDescobertaDlna() da classe DescoberdaDLNA
Fonte: Autor

Ao iniciar o método, tem-se:

- **Linhas 44 a 46:** A classe DLNAService da biblioteca DLNA, é responsável por criar os recursos de rede necessários para interagir com o protocolo DLNA, essa classe recebe como parâmetro um objeto do tipo DLNAListener, que é o responsável por interagir com os dispositivos DLNA do ambiente;

- **Linha 49:** Como já mencionado na listagem, essa linha é responsável por escutar todas as mensagens que os dispositivos, através do seu protocolo de conectividade, estão disponibilizando no ambiente, passando por parâmetro no método search(), o tipo de mensagem que o mesmo deseja captar. Nesse caso, o objeto dlnaDescoberta, deseja capturar todos os cabeçalho dos dispositivo DLNA disponíveis no ambiente.

- **Linhas 51 a 57:** Antes de finalizar o processo de descoberta de protocolo, é preciso segura-lo por tempo indeterminado, até que o mesmo encontre o protocolo específico. Depois de encontra-lo, o módulo de Análise pode pausar o módulo referente a esse protocolo para diminuir as requisições de acesso na rede, melhorando o desempenho do *proxy* para outras tarefas. Uma condição de retorno é utilizada até que o status do módulo seja igual a true, e a cada 10 segundos, é verificado se houve uma mudança nesse valor de variável. Caso ocorra alguma exceção ou o status mude seu valor de variável para false, o processo é finalizado, suando o método shutdown() da classe DLNAService, podendo esse processo ser inicializado novamente, utilizando o método run(), da interface Runnable;

- **Linha 61:** Se o valor da variável status for igual for falso, o processo de descoberta de protocolo é finalizado, podendo ser novamente inicializado pelo método run() da interface Runnable;

Uma estratégia para virtualização dos dispositivos abordada pelo módulo de Análise é iniciada somente com a descoberta de pelo menos dois protocolos de conectividade localizados em uma mesma rede. Caso exista somente um protocolo de conectividade na rede, o módulo não realizará a virtualização dos dispositivos, porquanto, com a existência de somente dispositivos com um único tipo de protocolo de conectividade, a virtualização dos mesmos é uma ação desnecessária, já que o usuário irá preferir utilizar um controlador nativo do mesmo protocolo utilizado pelos dispositivos.

Depois da descoberta dos protocolos de conectividade envolvidos no ambiente, o módulo de análise inicializa o procedimento para descoberta das informações dos dispositivos

presentes no mesmo. Procedimento esse realizado pelo módulo de Descoberta de Dispositivos referente à classe DescobertaDispositivos do desenvolvimento. O módulo de Análise considera somente os protocolos de conectividade descobertos no ambiente e inicializa a pesquisa para a descoberta das informações referentes aos dispositivos que utilizam desses protocolos. A descoberta dessas informações é ilustrada, resumidamente na Figura 28, a descoberta de dispositivos DLNA no ambiente.

```
36  public DescobertaDispositivos(boolean statusDescoberta) {
37      this.isProtocolo = false;
38      this.statusDescoberta = statusDescoberta;
39      devicesDLNA = new HashSet<>();
40      devicesUPnP = new HashSet<>();
41
42      desobertarDLNA = new DLNAListenerImpl() {
43          @Override
44          public void remoteDeviceAdded(Registry registry, RemoteDevice device) {
45              System.out.println(
46                      "Dispositivos DLNA disponíveis: " + device.getDisplayString());
47              devicesDLNA.add(device);
48              System.out.println("----------------------------------------");
49          }
50      };
```

Figura 28: Listagem da classe DescobertaDispositivos
Fonte: Autor

Pode-se observar que no construtor da classe DescobertaDispositivo, é preparado a configuração para a descoberta de todos os protocolos suportados pelo módulo. Esse processo é bem semelhante à descoberta dos protocolos no ambiente, pois esse módulo de descoberta de dispositivos, também usam as classes DLNAListener e RegistryListener para descoberta dos dispositivos DLNA e UPnP.

A classe de descoberta dos protocolos, está interessado somente na existência dos mesmos no ambiente, alterando o valor de sua variável isProtocolo para true (ver linha 32 da Figura 26), assim, quem tiver interessado nessa informação tomar suas devidas ações. Entretanto, o módulo de descoberta de dispositivo, ao verificar a existência de dispositivos no ambiente, adicionar uma referência desse dispositivo, do tipo Device, que também é disponibilizada pela API, a um Colletions do tipo HashSet, garantindo a não duplicidade de um mesmo dispositivo na lista (linha 47).

Depois de preparar o ambiente para a descoberta dos dispositivos, a Figura29, cria os recursos de rede necessários, através da interface DLNAService, para interagir com a rede no ambiente que o *proxy* esteja conectado. Essa classe recebe como parâmetro um objeto do tipo DLNAListener que é o responsável por interagir com os dispositivos DLNA

O método inicializandoDescoberta(), da classe DescobertaDispositivos, inicializada a descoberta dos dispositivo no ambiente, mas somente dos dispositivos com protocolos de

```
65      public void inicializandoDescoberta() {
66          DLNAService dlnaDescoberta = null;
67          UpnpService upnpDescoberta = null;
68          if (this.statusProtocoloDLNA == true) {
69              //Isto irá criar recursos de rede necessários para UPnP imediatamente
70              System.out.println("Inicializando Descoberta DLNA");
71              dlnaDescoberta = new DLNAService(desobertarDLNA);
72              System.out.println("----------------------------------------");
73              // Envia uma mensagem de busca para todos os dispositivos e serviços
74              dlnaDescoberta.getDiscovery().search(new STAllHeader());
75              System.out.println("----------------------------------------");
76          }
```

Figura 29: Inicializando a descoberta dos dispositivos DLNA no ambiente
Fonte: Autor

conectividade encontrados pelo módulo de descoberta de protocolos. Esse trecho de código é similar ao trecho da descoberta dos protocolos, da Figura 27, pois os mesmo criam os recursos necessários para a descoberta de determinados protocolos de conectividade. Entretanto, na descoberta dos dispositivos, uma condição de seleção garante que somente os protocolos descoberto pelo *proxy* possam ser inicializado (linha 68). Esse condição é inserida pelo módulo de Analise através do método setStatusProtocoloDLNA(boolean statusProtocoloDLNA), da classe DescobertaDispositivo, quando o módulo descobre o protocolo DLNA no ambiente, para então, realizar a chamada desse método resumido na Figura 29, para inicializar a descoberta dos dispositivos.

Depois de realizar a descoberta dos dispositivos no ambiente, o módulo de Analise inicializa o processo de virtualização dos mesmos, como também, inicializa o módulo de Existência de dispositivos. Esse último tem o objetivo de verificar a permanência dos dispositivos reais no ambiente, garantindo assim que o dispositivo virtual inicializado tenha sempre um dispositivo real vinculado a ele. Quando o dispositivo real não existir no ambiente, o módulo de Análise avalia esta informação e destrói todos os dispositivos virtuais vinculados a esse dispositivo real.

A Figura 30 resume a estratégia abordada pelo módulo de Existência para verificar a permanência dos dispositivos DLNA reais no ambiente.

A classe contem seis atributos, um inteiro de nome timeout, que tem a responsabilidade de definir a intervalo de tempo para análise da permanência do dispositivo no ambiente, um *boolean* isModuloAtivo, para ativar ou desativar o módulo de Existência para pesquisa dos dispositivos encontrado pelo módulo de Descoberta de Dispositivo, como também, quatro listas do tipo Set. Duas que são passados por parâmetros no construtor (linha 41) quando iniciado o referido módulo, que são as lista dos dispositivo encontrado pelo módulo de Descoberta de Dispositivo, devicesDLNA (linha 36) e devicesUPnP (linha37), e as outras duas listas, devicesDLNAAtivos (linha 38) e devicesUPnPAtivos (linha 39), também

```
32  public class ExistenciaDipositivos implements Runnable {
33
34      private int timeOut;
35      private boolean isModuloAtivo;
36      private Set<Device> devicesDLNA;
37      private Set<Device> devicesUPnP;
38      private Set<Device> devicesDLNAAtivos;
39      private Set<Device> devicesUPnPAtivos;
40
41      public ExistenciaDipositivos(Set<Device> devicesDLNA, Set<Device> devicesUPnP) {
42          this.devicesDLNA = devicesDLNA;
43          this.devicesUPnP = devicesUPnP;
44          isModuloAtivo = true;
45          timeOut = 5000;
46      }
```

Figura 30: Classe ExistenciaDispositivo
Fonte: Autor

do tipo Set, que garante a não duplicação de um mesmo dispositivo já adicionado na lista, e são responsáveis por criar as listas dos dispositivos ativos no ambiente, ou seja, aqueles dispositivos encontrados pelo módulo de Descoberta de Dispositivos e verificado sua existência e intervalos de tempos pelo módulo de Existência.

Portanto, o módulo de Análise, depois de virtualizar os dispositivos no ambiente, ficará verificando a existências dos mesmos, consultando a lista dos dispositivos ativos do módulo de Existência, que em um determinado intervalo de tempo, é atualizada pelo próprio módulo. Dos métodos listados do módulo de Existência, os que são responsáveis em verificar a existência dos dispositivos DLNA ativos no ambiente, são ilustrados na Figura 31.

```
47      // Verifica se determinado host esta atingivel
48      public boolean doPing(Device device) {
49          String host = device.getDetails().getBaseURL().getHost();
50          try {
51              return InetAddress.getByName(host).isReachable(this.timeOut);
52          } catch (IOException e) {
54              e.printStackTrace();
55              return false;
56          }
57      }
58      public void inicialiarExistenciaDispositivo() {
59          devicesDLNA_Ativos = new HashSet<Device>();
60          while (this.isModuloAtivo == true) {
62              if (this.devicesDLNA != null) {
63                  for (Device device : devicesDLNA) {
64                      //verifica o dispositivo está ativo
65                      if (doPing(device)) {
66                          //Nova lista com dispositivos ativos
67                          devicesDLNA_Ativos.add(device);
68                      }
                    }
                }
```

Figura 31: Métodos da classe ExistenciaDispositivos
Fonte: Autor

O método doPing(), é o responsável por de verificar as conexões dos dispositivos

remotos através do seu endereço IP obtido pela rede, esse recebe por parâmetro o objeto do tipo Device que contem todas as informações de um determinado dispositivo no ambiente, dentre elas, o endereço *host* (endereço IP) do dispositivo. Desta forma, é utilizado a classe InetAddress do pacote java.net para verificar a existência do mesmo na rede (linha 51). Nesse objeto é passado por parâmetro no método getByName() o endereço host do dispositivo que deseja verificar sua existência (linha 49), como também, o intervalo de tempo em milissegundos que o objeto irá verificar se o *host* está respondendo, caso o mesmo responda, o método retorna true, caso não, retorna false.

Para inicializar a existências dos dispositivos na rede, o método inicializarExistencia-Dispositivo() é chamado pelo objeto ExistenciaDispositivo. Primeiramente, sempre que esse método é chamado, uma nova instancia da lista onde serão armazenados todos os dispositivos ativos da rede é criado, garantindo a atualização da mesma. Depois, é verificado se o atributo isModuloAtivo é igual a true (linha 58), caso seja, o módulo verifica se existe dispositivos encontrados pelo módulo de Descoberta de dispositivos, assim, como esse método tem o objetivo de verificar a existência contínua de todos os dispositivos no ambiente, independente dos protocolos, uma verificação antecipada é obrigatória se a lista passada no construtor da classe, (linha 41, Figura 30), é diferente de null (linha 60), isso garante que o módulo de Descoberta de Dispositivos encontrou dispositivos referente a aquele determinado protocolo de conectividade vinculado a ele.

Depois dessa verificação, o módulo percorre todos os dispositivos da lista passada no construtor da classe e faz chamada ao método doPing (linha 63), caso haja resposta do dispositivo, o mesmo é adicionado em uma nova lista, contendo somente os dispositivos ativos na rede. Com isso, o módulo de Análise, depois que virtualizar os dispositivos no ambiente, fica atualizando e verificando essa nova lista a fim de garantir que o dispositivo virtual tenha sempre um virtual vinculado a ele.

Por fim, o módulo de Análise tem todas as informações necessárias para virtualizar os dispositivos reais encontrado na rede, interoperabilizando a troca de informação entre diferentes protocolos. Essa interação é feita pelo módulo de Virtualização, como pode ser observado resumidamente na Figura 32, a virtualização de um dispositivo DLNA real para um dispositivo virtual UPnP.

Primeiramente, a classe VirtualUPnP implementa a interface Runnable com o objetivo de criar uma Thread independente para cada dispositivo virtualizado, garantindo assim que cada dispositivo real tenha um dispositivo virtual único interligado a ele. Essa classe somente tem dois atributos, um do tipo Device que representa a instância do dispositivo

```
23  public class VirtualUPnP implements Runnable {
24
25      private final VirtualServices servicos;
26      private final Device deviceDLNA;
27
28      public VirtualUPnP(Device deviceDLNA) {
29          this.deviceDLNA = deviceDLNA;
30          servicos = new VirtualServices();
31      }
32
33      public LocalDevice createDevice() throws ValidationException, LocalServiceBindingException, IOException {
34          //Configuração do dispositivo Virtual
35          DeviceIdentity identity = new DeviceIdentity(
36                  UDN.uniqueSystemIdentifier(this.deviceDLNA.getIdentity().toString()));
37
38          DeviceType type = new UDADeviceType(this.deviceDLNA.getType().toString(),
39                  this.deviceDLNA.getVersion().getMajor());
40
41          DeviceDetails details = new DeviceDetails(this.deviceDLNA.getDetails().getFriendlyName(),
42                  new ManufacturerDetails(this.deviceDLNA.getDetails().getManufacturerDetails().getManufacturer()),
43                  new ModelDetails(
44                          this.deviceDLNA.getDetails().getModelDetails().getModelName(),
45                          this.deviceDLNA.getDetails().getModelDetails().getModelDescription(),
46                          this.deviceDLNA.getDetails().getModelDetails().getModelNumber()));
47
48          Icon[] icon = this.deviceDLNA.getIcons();
```

Figura 32: Classe VirtualUPnP
Fonte: Autor

DLNA encontrado no ambiente, declarada essa como final, pois nenhuma alteração na mesma poderá ser realizada. E o segundo atributo do tipo VirtualServices, que é a responsável por interligar os serviços do dispositivo real DLNA com virtual UPnP. A instância do dispositivo DLNA é passada pelo módulo de Análise no construtor dessa classe (linha 28), como também e feita a referência ao dispositivo da classe (linha 29), como também, é inicializado o objeto VirtualServices (linha 30).

Depois de obter a referência do dispositivo real DLNA a qual deseja virtualizar, o módulo de Virtualização configura o dispositivo virtual com as mesmas informações coletadas do dispositivo real, fazendo referência direta e esse dispositivo. O método createDevice() (linha 33), é o responsável por realizar a interação dessas informações. A primeira parte desse método é configurar o disposto virtual UPnP com as informações básicas do dispositivo real DLNA, que são:

- **Linhas 35 e 36:** Cada dispositivo em uma rede, não importa se seja UPnP ou DLNA, requer uma UDN (*Unique Device Name*), ou seja, uma identificação única do seu dispositivo. O chamada ao método UDN.uniqueSystemIdentifier() do objeto DeviceIdentiy oferece exatamente isso, um identificador único, que nesse caso em específico, é o identificador do dispositivo DLNA passado por referencia no Construtor da classe. Essa informação é de estrema importância, pois, como essa identificação é normalizada pela documentação do protocolo DLNA ou UPnP e ela faz referência

direta as quantos e quais serviços esse determinado dispositivo fornece, desenvolvedores podem projetar pontos de controle para consumir esses serviços fazendo referência direta a UDN de cada dispositivo.

- **Linhas 38 e 39:** É passado no construtor da classe DeviceType do dispositivo virtual, a referência do tipo do dispositivo real DLNA descoberto na rede. Essas informações são originalmente, uma referência do tipo e versão do dispositivo, que cumpre a especificação UDA do protocolo.

- **Linhas 41 a 46:** Estas linhas detalham informações sobre o nome amigável do dispositivo, bem como o modelo e as informações do fabricante do mesmo. Essas informações são refeitas referência pelo construtor da classe DeviceDetails que recebe 3 parâmetros. Primeiro uma String que faz referência ao nome amigável do dispositivo, que será o mesmo nome do dispositivo DLNA a ser virtualizado. O segundo parâmetro faz referência às informações do fabricante do dispositivo, nome, endereço URL do dispositivo e endereço URL do fabricante, entre outras informações passado por parâmetro no objeto ManufacturerDetails() (linha 42). Por último, o terceiro parâmetro faz referencia as informações detalhadas do modelo do dispositivo (linhas 44, 45, 46).

- **Linha 48:** Cada dispositivo pode ter vários ícones associados que se assemelha ao nome amigável do dispositivo para apresentações em interfaces gráficas

Depois de realizar as configurações básicas das informações específicas do dispositivo, vem à parte mais importante de um dispositivo, os serviços que os mesmo disponibilizam no ambiente. Cada serviço encapsula os metadados de um serviço em específico que são as ações e as variáveis de estado do mesmo, e como ele pode ser invocado. Essa interação dos serviços entre o dispositivo real e virtual, ainda é responsabilidade do método createDevice(), exemplificado na Figura 33.

Primeiramente, é adicionado uma condição para percorre a lista de todos os serviços que o dispositivo DLNA disponibiliza (linha 50), cada serviço do dispositivo DLNA é verificado se existe um serviço compatível, utilizando o método getIdentificacaoServicos() da classeVirtualServices (linha 51). Esse método é uma Lista de *Arrays* com duas posições, a primeira é referente às identificações dos serviços dos dispositivos DLNA e a segunda posição é a referencia a identificação dos Serviços UPnP. Tais identificações são originadas da documentação de ambos os protocolos, fazendo referência direta a UDN de cada dispositivo. Depois de percorre a lista com as identificações dos serviços, é veri-

```
50          for (Service s : deviceDLNA.getServices()) {
51              for (String[] id : servicos.getIdentificaoServicos()) {
52                  if (s.getServiceId().getId().equals(id[0])) {
53                      servicos.verificaServico(id[1], s);
54
55                  }
56              }
57          }
58          //Configuração e Informações dos Serviços
59          LocalService<VirtualServices> createService = new AnnotationLocalServiceBinder()    )
60                  .read(servicos.getClass());
61          return new LocalDevice(identity, type, details, new LocalService[]{createService}); e));
62      }
```

Figura 33: Continuação do método createDevice() da classe VirtualUPnP
Fonte: Autor

ficado uma condição se existem esse serviço (linha 52), caso essa identificação exista, é instantaneamente chamado o método verificaServico() (linha 53) do objeto servicos do tipo VirtualServicos, passando por parâmetro a referência do serviço que deverá ser instanciado no dispositivo virtual, como também, qual referência do serviço DLNA que deverá realizar a tradução das informações entre os protocolos.

Depois de vinculado o serviço DLNA com o UPnP correspondente, é criado os serviços do dispositivo virtual UPnP. A interface LocalDevice (linhas 59 e 60) representa um dispositivo UPnP local que é lançado a mesma rede em que o *proxy* está localizada, obtendo assim, um endereço host para o ambiente. Por último, o método creatDevice() (linha 33, Figura 32) retorna um novo dispositivo local com todas as configurações finalizadas, tanto das informações básicas de um dispositivos, com as informações dos serviços que o mesmo disponibiliza.

Por fim, para finalizar a criação do dispositivo virtual no ambiente, a classe VirtualUPnP, implementa o método run() da interface Runnable. Observar Figura 34.

```
76      @Override
77      public void run() {
78          try {
79              final UpnpService upnpService = new UpnpServiceImpl();
80              // Adiciona o dispositivo local ligado ao registro
81              upnpService.getRegistry().addDevice(createDevice());
82          } catch (Exception e) {
83              System.err.println("exeção ocorrida: " + e);
84              e.printStackTrace(System.err);
```

Figura 34: Método run() da classe VirtualUPnP
Fonte: Autor

Para implantar um dispositivo UPnP na rede, o mesmo utiliza a Interface UpnpService (linha 79), e lança o dispositivo na rede através do método addDevice() (linha 81), passando por parâmetro o dispositivo que foi criado na Listagem 32 e 33. Caso ocorra alguma exceção, as linhas 83 e 84 exibem no console as devidas informações. Para finalizar, será

discutido, resumidamente na Figura 35, a classe VirtualServices que é a responsável por conectar os serviços entre os diferentes protocolos, no estudo de caso desse trabalho, os protocolos DLNA e UPnP, referente linha 53 da Figura 33.

```
31  public  class VirtualServices {
32
53      private Map servicosMap = new HashMap<String, Object>();
54      private List<String[]> identificaoServicos;
45      private String URL = "identificacaoServicos.txt";
36      private SwitchPower switchPower;
37      private VolumeService volumeService;
38      private ChannelService channelService;
39      //outros Serviços
40
41      public VirtualServices() {
42          carregarServicos();
43          identificaoServicos = new ArrayList<String[]>();
44          try {
45              BufferedReader reader = new BufferedReader(new FileReader(new File(URL)));
46              while (reader.ready()) {
47                  String linha = reader.readLine();
48                  String[] info = linha.split(" ## ");
49                  identificaoServicos.add(info);
50              }
51          } catch (IOException ex) {
52              Logger.getLogger(VirtualServices.class.getName()).log(Level.SEVERE, null, ex);
53          }
54      }
```

Figura 35: Classe VirtualService
Fonte: Autor

A classe VirtualService contém os seguintes atributos:

- **servicosMap:** Um mapa chave-valor contendo todos os serviços que qualquer dispositivo virtual UPnP possa disponibilizar no ambiente. Essa mapa é preenchido através do método carregarServicos();
- **identificacaoServicos:** Esse atributo contem as referencias das identificações dos Serviços que os dispositivo DLNA e UPnP possam utilizar. Tal identificação é baseada na especificação de sua documentação e é carregada através de um arquivo (linhas 43 a 52), no construtor da classe. Esse arquivo contém Strings em cada linha, o primeiro é a identificação do serviço DNLA e o segundo é a identificação do serviço UPnP, separados pelo caracteres "##". Depois que a classe ler cada linha desse arquivo, é adicionado nessa lista que contem um *Array* de *String* de duas posições, a primeira é a referencia ao ID do serviço DLNA e a segunda ao ID do serviço UPnP;
- **URL:** Endereço de localização do arquivo onde contém as informações anteriormente abordadas, nesse caso, na raiz do projeto;

Os outros atributos são instancias dos serviços que podem ser usados pelos dispositivos

virtuais UPnP. Ainda na classe VirtualService, pode-se observar os seguintes métodos na Figura 36.

```
85      public void verificaServico(String identificacao, Service s) {
86          if (identificacao.equals("SwitchPower")) {
87              switchPower = (SwitchPower) servicosMap.get(identificacao);
88              switchPower.setService(s);
89          }
90          //outras condições dos outros serviços
91      }
```

Figura 36: Método verificarServico da Classe VirtualService
Fonte: Autor

O método verificaServico(), utilizado na listagem da Figura 33, linha 53, é o responsável por interagir os serviços do dispositivo real e o virtual. Esse método recebe por parâmetro, uma variável identificação do tipo *String* que corresponde a identificação UPnP encontrado de um determinado serviço DLNA disponibilizado pelo dispositivo, já o segundo parâmetro, recebe a instancia do serviço que o mesmo corresponde. Com o conjunto dessas informações, o método verifica se a identificação coexiste com os serviços disponibilizados pelo dispositivo virtual (linha 86), caso essa informação seja verdadeira, é recuperado a instância do serviço disponível no método get() do HashMap servicosMap (linha 87), passando por parâmetro a identificação do serviço DLNA passada no segundo parâmetro do método.

Da mesma forma que a API Cling referente ao protocolo UPnP, disponibiliza a criação de dispositivos em uma rede, a mesma, também disponibiliza a criação de serviços separadamente do dispositivo. Isso garante que um mesmo serviço possa ser criado uma única vez, e instanciado por diferentes dispositivos, garantindo assim, o reuso de código e compartilhamento de um mesmo serviço em diferentes ocasiões. A Figura 37, ilustra a criação do serviço de identificação "SwitchPower", que corresponde a ação de ligar ou desliga um dispositivo, função mais básica que praticamente todos dispositivos existentes disponibilizam em um ambiente.

A API Cling disponibiliza anotações no escopo da classe, método e atributos para identificar as informações referentes aos serviços disponibilizados no ambiente. A linha 125 faz referencia a essa classe interna informando que a mesma é um serviço UPnP, passando em seu parâmetro, a identificação do mesmo, como por exemplo, informações referente a o tipo de serviço disponibilizado e sua versão. A classe SwitchPower contem os seguintes métodos:

- **service:** Uma instancia de um serviço DLNA que será o responsável por realizar a tradução dos serviços UPnP correspondentes;

```
125     @UpnpService(serviceId = @UpnpServiceId("SwitchPower"),
126             serviceType = @UpnpServiceType(value = "SwitchPower", version = 1))
127     public class SwitchPower {
128
129         private Service service;
130         private DLNAListener desobertarDLNA;
131         private DLNAService dlnaService;
132         @UpnpStateVariable(defaultValue = "0", sendEvents = false)
133         private boolean target = false;
134         @UpnpStateVariable(defaultValue = "0")
135         private boolean status = false;
136
137         public void setService(Service service) {
138             this.service = service;
139             desobertarDLNA = new DLNAListener() {
140                 @Override
                    public void remoteDeviceAdded(Registry registry, RemoteDevice device) {
142                     String idService = service.getReference().getServiceId().toString();
143                     for (Service serv : device.getServices()) {
144                         if (serv.getAction(idService).getService().hasActions()) {
145                             setTarget(service.getAction(idService).hasArguments());
146                         }
147                     }
148                 }
149             };
150             dlnaService = new DLNAService(desobertarDLNA);
```

Figura 37: Classe interna SwitchPower
Fonte: Autor

- **descobertaDLNA:** Esse atributo é o responsável por interagir com os dispositivos DLNA do ambiente, já discutido na Figura 26;
- **dlnaService:** é responsável por criar os recursos de rede necessários para interagir com o protocolo DLNA, já discutido na Figura 28;
- **target:** Variável de estado referente ao comportamento do serviço disponibilizada, se true, o dispositivo está ligado, se false, o dispositivo está em *stand by*;
- **status:** Variável de retorno responsável por disponibilizar o status do serviço do dispositivo;

Ainda nesse método, um comportamento em específico é tratado quando um serviço é consumido em um dispositivo real e esse mesmo serviço é consumido no dispositivo virtual. Primeiramente é verificado em todos os serviços disponibilizado no dispositivo real (linha 143). A linha 144 verifica se houve esse consumo no serviço do dispositivo real, passando por parâmetro no método getAction() (linha 144) o idService (linha 142) do serviço do dispositivo DLNA, ou seja, a ação passada por parâmetro no método getAction() verifica se houve alguma mudança no se estado atual, caso essa informação seja verdadeira, o método setTarget() da classe SwitchPower consome esse mesmo serviço no dispositivo virtual passando por parâmetro o mesmo argumento modificado no serviço do dispositivo real (linha 145). Ver Figura 38 o consumo desse serviço no dispositivo virtual.

```
160     @UpnpAction
161     public void setTarget(@UpnpInputArgument(name = "newTargetValue") boolean newTargetValue) {
162         target = newTargetValue;
163         status = newTargetValue;
164         System.out.println("Mudar para: " + status);
165         executeActionDLNA(dlnaService, service, "newTargetValue");
166     }
```

Figura 38: Método da classe SwitchPower responsável por consumir o serviço virutal local e o real conectado

Fonte: Autor

O método setTarget() (linha 161) é uma ação referente a um determinado serviço virtual UPnP, nesse caso, a ação de ligar ou desligar um dispositivo. Esse recebe por parâmetro, um atributo boolean responsável de executar a ação. Entretanto, esse método além de consumido o seu serviço virtual, ele também deverá consumir um mesmo serviço no dispositivo virtual vinculado a ele, ou seja, realizar o comportamento inverso explicado anteriormente.

Quando um serviço é consumido no dispositivo virtual UPnP, esse deve consumir o mesmo serviço no dispositivo real DLNA, essa ação é realizada no método executeActionDLNA(), que passa no primeiro parâmetro o dlnaService, ou seja, o recurso necessários para interagir com o protocolo DLNA, já criado anteriormente, no segundo parâmetro, o atributo service, que é o serviço DLNA que também já está referenciado anteriormente pelo método setService() (ver linha 137 da Figura 37), e por último, a identificação da referência do serviço que é consumido no dispositivo DLNA (linha 165), referida na Figura 39.

```
138     private void executeActionDLNA(DLNAService dlnaService, Service powerService, String id, boolean target) {
139
140         ActionInvocation setTargetInvocation = new ActionInvocation(powerService.getAction(id));
141         // Executa o consumo do serviço DLNA
142         dlnaService.getControlPoint().execute(setTargetInvocation.setInput(target));
143     }
```

Figura 39: Método que consome um serviço do dispositivo real DLNA
Fonte: Autor

O método executeActionDLNA() consome no serviço DLNA powerService o mesmo valor consumido no serviço virtual UPnP referente ao quarto parâmetro com variável de nome target. O método invoca a ação do serviço do dispositivo ao qual quer consumir (linha 140), depois, essa ação é passada por parâmetro no método execute() do serviço do dispositivo DLNA dlnaService (linha 142), invocando a ação que deseja ser realizada e passando por parâmetro a variável taget referente ao valor atualizado da variável de estado do dispositivo virtual UPnP.

Com isso, todos os comportamentos abordados pelo *proxy* foram resumidamente exem-

plificados nessa seção, descrevendo parte do desenvolvimento envolvido na solução da proposta deste trabalho, garantindo assim a interoperabilidade dos protocolos de conectividade em um ambiente inteligente.

Este capítulo apresentou algumas informações para consolidar mais a fundamentação teórica deste trabalho, como também, abordar as principais soluções para interoperabilidade entre dispositivos em um ambiente inteligente, e por fim, explorar problemas correlacionados com esse tema. Para tal, foi utilizado o método da Revisão Sistemática para isso. Primeiramente apresentou-se a metodologia utilizada durante o processo de revisão sistemática. Posteriormente, na seção 3.1 foram definidos os passos que forão utilizados para aplicar a revisão sistemática, elaborando assim, o protocolo para esta finalidade.

Depois de desenvolvido o protocolo da revisão, a Seção 3.2 é dissertado a etapa de execução da revisão, onde foram identificados, selecionados e avaliados os dados referentes à pesquisa em questão, feito uma análise minuciosa dos trabalhos retornados do estado da arte sobre as questões definidas no protocolo.

Por fim, a seção 3.3, foram discutidos as respostas abordas no protocolo da revisão, consolidando as informações necessárias para o desenvolvimento deste trabalho, abordando problemas correlacionados com a interoperabilidade entre diferentes dispositivos em um mesmo ambiente. Logo, o próximo capítulo apresenta uma proposta para esse problema.

5.4 Considerações Finais do Capítulo

Por último, este capítulo detalha todo o processo de desenvolvimento da proposta abordade nesse trabalho, como também, a aplicabilidade da mesma. Primeiramente, foram identificados as principais diferenças entres os protocolos utilizados neste trabalho, qual a melhor estratégia abordada para alcançar a interoperabilidade entre esses protocolos de conectividade, e por último, um é apresentado um estudo de caso abordando o funcionamento do comportamento da proposta (*proxy*) em uma determinada situação,como também o processo de desenvolvimento do mesmo.

Vale ressaltar que, pelo tempo agregado para este trabalho, a proposta de solução ficou restrita a somente dois protocolos, UPnP e DLNA. Contudo, como se trata de uma proposta, uma abordagem escalável desse trabalho deverá ser feitas nos trabalhos futuros, agregando novos protocolos na proposta, generalizando assim o seu uso nos diferentes ambiente inteligentes.

6 Considerações Finais

Neste capítulo será exposta uma conclusão parcial do trabalho até o momento apresentado, como também, um cronograma exibindo o andamento das tarefas que deverão ser realizadas para finalização do mesmo

6.1 Conclusão

A capacidade de um dispositivo com o uso de um protocolo de conectividade ser reconhecido em um ambiente de forma automática, segundo (KAED et al., 2011; LAI; HUANG, 2008), é peça fundamental para a concretização de um ambiente inteligente e função mais fascinante dos protocolos identificados nesse trabalho. Por outro lado, com a falta de um consenso ou norma que defina um arquitetura padrão para o uso dos protocolos de conectividade nos dispositivos presentes em tais ambientes e a diversidade destes protocolos de conectividade e tecnologias atualmente existentes, torna impeditiva a interoperabilidade transparente entre dispositivos, e consequentemente, inviabiliza a implementação plena de ambientes residenciais inteligentes

Neste trabalho apresentamos uma proposta para interoperabilidade entre os protocolos de um ambiente inteligente e demonstramos a viabilidade da solução, aplicando uma contexto envolvendo os protocolos DLNA e UPnP. Esta proposta introduz serviços virtuais que permitirão que dispositivos DLNA possam interagir com Pontos de Controle UPnP já desenvolvidos, tendo como objetivo o reaproveitamento de soluções já existentes no mercado e, consequentemente, uma diminuição de esforços para criação ou adaptação das soluções de *middlewares* já existentes. Os esforços dessa solução são importantes porque, em um curto prazo, com a falta de uma iniciativa tanto acadêmica como industrial para o desenvolvimento de um padrão que envolva essa área de ambientes inteligentes, uma série de tecnologias e protocolos irão competir pelo seu espaço, sobrecarregando assim uma quantidade cada vez maior de protocolos independentes em um ambiente, tornando-o cada vez mais heterogêneo e dificultando assim a interoperabilidade entre os protocolos

presentes nele.

6.2 Trabalhos Futuros

A versão inicial da proposta abordada nesse trabalha limita somente na utilizando dos protocolos de conectividade DLNA e UPnP, sendo esses aplicado diretamente no ambiente em que os mesmo encontram-se. Como trabalhos futuros, primeiramente será desenvolvido um extensão da solução abrangendo mais protocolos de conectividade, aumentando assim o escopo na usabilidade de diferentes dispositivos que farão parte na interação e troca de informação em ambientes inteligentes, proporcionando ao usuário uma maior escolha na diversidade de dispositivos que o mesmo possa adquirir sem a necessidade de ser preocupar em qual tipo de tecnologia ou protocolo os mesmos proporcionam.

Segundo, com o avança da tecnologia e maior de processamento nos celulares inteligentes (*smartphones*), uma versão móvel da solução poderá ser desenvolvidos focando em duas vertentes de usabilidade, uma sendo novamente o usuário, pois o mesmo poderá ter uma versão móvel da solução diretamente aplicado no seu *smartphone*, proporcionando a iteração com dispositivos externos disponibilizados no contexto em que o mesmo se encontra, e por fim, outro cenário possível para a usabilidade do *proxy*, seria pelos próprios desenvolvedores de aplicativos.

Devido a questões de tempo para o desenvolvimento deste trabalho, o mesmo ficou centralizado apenas na interoperabilidade entre diferentes protocolos de conectividade e tornou-se inviável adaptar os dispositivos a uma programação sensível ao contexto, dotada de recursos de inteligência artificial capaz de compreender, registrar a rotina dos usuários e modificar dinamicamente a comunicação entre os dispositivos, sem a necessidade da modificação manual da programação dos mesmos.

Com o objetivo de minimizar o esforço no desenvolvimento de aplicativos para entender diferentes protocolos de conectividade, o desenvolver poderá utilizar o a versão móvel e extensível do *proxy* para realizar a busca dos dispositivos já disponíveis no ambiente, independente dos protocolos vinculados ao mesmo, e com isso, o desenvolvedor irá focar somente na inteligência envolvida na iteração entre os dispositivos não se preocupando com a iteração a nível de rede com os mesmo.

6.3 Contribuições

- Levantamento sobre os diferentes protocolos de conectividades existentes para desenvolvimento de ambientes inteligentes;
- Realizado uma pesquisa sistemática no estado da arte, destacando o principal problema no processo de desenvolvimento de aplicações para ambientes inteligentes, e discutido proposta para soluções temporárias e a longo prazo para o problema abordado;
- Na literatura, várias abordagens estão sendo desenvolvidas com o mesmo objetivo deste trabalho. Porém, essas abordagens ainda estão centralizadas em um dos componentes que envolvem os protocolos de conectividade dos ambientes, os pontos de controles, com isso, forçando a utilização padrão de pelo menos um dos protocolos de conectividade envolvidos na rede, ao qual está agregado com esse ponto de controle. A solução abordada por esta dissertação, aborda um componente independente de protocolos, ou seja, um terceiro componente na rede que irá realizar a tradução de qualquer protocolo de conectividade agregada a rede, sem engrandecer nenhum protocolos utilizados.
- Com a proposta desenvolvida por esta dissertação, pesquisadores podem centralizar mais esforços no processo da inteligência sensível ao contexto onde, dispositivos independentemente, possam a parti da comunicação entre outros dispositivos sem a necessidade de se preocupar com o protocolo que o mesmo esteja integrado, interagir com o contexto do usuário de forma autônoma.

Referências

ACAMPORA, G. et al. A survey on ambient intelligence in healthcare. *Proceedings of the IEEE*, v. 101, n. 12, p. 2470–2494, Dec 2013. ISSN 0018-9219.

ALAM, M.; REAZ, M. B. I.; ALI, M. A. M. A review of smart homes: Past, present, and future. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, v. 42, n. 6, p. 1190–1203, Nov 2012. ISSN 1094-6977.

ALCANIZ, B. R. M. New technologies for ambient intelligence. *In The Evolution of Technology, Communication and Cognition Towards the Future of Human-Computer Interaction*, 2005.

ALLARD, J. et al. Upnp services and jini clients. In: *Proceedings of the 2003 Symposium on Applications and the Internet*. Washington, DC, USA: IEEE Computer Society, 2004. (SAINT '03), p. 268–. ISBN 0-7695-1872-9. Disponível em: <http://dl.acm.org/citation.cfm?id=827273.829239>.

ANAND, S. A dlna framework for next gen mobile terminals connecting ims networks for human-centered digital home environment. In: *IP Multimedia Subsystem Architecture and Applications, 2007 International Conference on*. [S.l.: s.n.], 2007. p. 1–5.

ATZORI, L.; IERA, A.; MORABITO, G. The internet of things: A survey. *Comput. Netw.*, Elsevier North-Holland, Inc., New York, NY, USA, v. 54, n. 15, p. 2787–2805, out. 2010. ISSN 1389-1286. Disponível em: <http://dx.doi.org/10.1016/j.comnet.2010.05.010>.

BARBARá, D. Mobile computing and databases-a survey. *IEEE Trans. on Knowl. and Data Eng.*, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 11, n. 1, p. 108–117, jan. 2009. ISSN 1041-4347. Disponível em: <http://dx.doi.org/10.1109/69.755619>.

BAUKNECHT, K. *Ambient Intelligence: The Vision of Information Society*. [S.l.]: Seminar aus Business Intelligence, 2002.

BECKEL, C. et al. Requirements for smart home applications and realization with ws4d-pipesbox. In: *Emerging Technologies Factory Automation (ETFA), 2011 IEEE 16th Conference on*. [S.l.: s.n.], 2011. p. 1–8. ISSN 1946-0740.

BREGMAN, D.; KORMAN, A. A universal implementation model for the smart home. *International Journal of Smart Home*, 2009.

C. YANN B., B. F. N. Thoughts smart home. 2013. Disponível em: <http://twixar.me/fJT>.

CHACZKO, Z. et al. Smart hospital management system: An integration of enterprise level solutions utilising open group architecture framework (togaf). In: *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*. [S.l.: s.n.], 2010. v. 5, p. 8–15.

CHEN, C.-Y. et al. A green context-aware platform for smart living. In: *Service-Oriented Computing and Applications (SOCA), 2013 IEEE 6th International Conference on*. [S.l.: s.n.], 2013. p. 275–281.

CHEN, W.; KUO, S.-Y.; CHAO, H.-C. Service integration with upnp agent for an ubiquitous home environment. *Information Systems Frontiers*, Kluwer Academic Publishers, Hingham, MA, USA, v. 11, n. 5, p. 483–490, nov. 2009. ISSN 1387-3326. Disponível em: <http://dx.doi.org/10.1007/s10796-008-9122-3>.

CHOWDHURY, R. et al. Interconnecting multiple home networks services. In: *Telecommunications, 2008. ICT 2008. International Conference on*. [S.l.: s.n.], 2008. p. 1–7.

COOK, D. J.; AUGUSTO, J. C.; JAKKULA, V. R. Review: Ambient intelligence: Technologies, applications, and opportunities. *Pervasive Mob. Comput.*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 5, n. 4, p. 277–298, ago. 2009. ISSN 1574-1192. Disponível em: <http://dx.doi.org/10.1016/j.pmcj.2009.04.001>.

CORONATO, A.; TESTA, A. Runtime verification of location-dependent correctness and security properties in ambient intelligence applications. In: *Internet Communications (BCFIC Riga), 2011 Baltic Congress on Future*. [S.l.: s.n.], 2011. p. 153–160.

CORTIMIGLIA, M.; GHEZZI, A.; RENGA, F. Mobile applications and their delivery platforms. *IT Professional*, v. 13, n. 5, p. 51–56, 2011. ISSN 1520-9202.

CRD. *Systematic reviews: CRD's guidance for undertaking reviews in health care*. v. 10, n. 4, april 2010. 226 p. Disponível em: <http://www.thelancet.com/journals/laninf/article/PIIS1473-3099(10)70065-7/fulltext>.

D., C.; P., N. Domotica : Aplicabilidade e sistemas de automacao residencial. *VÃ©rtices*, v. 6, n. 3, 2010. ISSN 1809-2667. Disponível em: <http://essentiaeditora.iff.edu.br/index.php/vertices/article/view/1809-2667.20040015>.

DELPHINANTO, A.; KOONEN, A. M. J.; PEETERS, M. E. Proxying upnp service discovery and access to a non-ip bluetooth network on a mobile phone. In: *Communications and Vehicular Technology in the Benelux, 2007 14th IEEE Symposium on*. [S.l.: s.n.], 2007. p. 1–5.

DUCATEL, K. et al. Scenarios for ambient intelligence in 2010. In: . [S.l.]: IST Programme Advisory Group (ISTAG), 2005.

FANG, F. C. Intelligent traffic management systems for work zones and incident management. *Int. J. Intell. Syst. Technol. Appl.*, Inderscience Publishers, Inderscience Publishers, Geneva, SWITZERLAND, v. 7, n. 4, p. 370–381, set. 2009. ISSN 1740-8865. Disponível em: <http://dx.doi.org/10.1504/IJISTA.2009.028053>.

FREITAS, G. B. de; TEIXEIRA, C. A. C. Ubiquitous services in home networks offered through digital tv. In: *Proceedings of the 2009 ACM symposium on Applied Computing*. New York, NY, USA: IEEE, 2009. (SAC '09), p. 1834–1838. ISBN 978-1-60558-166-8. Disponível em: <http://doi.acm.org/10.1145/1529282.1529691>.

GAGGIOLI, A. Optimal experience in ambient intelligence. In: RIVA, G. et al. (Ed.). *Ambient Intelligence*. [S.l.]: IOS Press Amsterdam, 2005. p. 35–43.

GIL, A. *Como elaborar projetos de pesquisa*. Atlas, 2002. ISBN 9788522431694. Disponível em: <http://books.google.com.br/books?id=X4uvAAAACAAJ>.

GUBBI, J. et al. Internet of things (iot): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst.*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 29, n. 7, p. 1645–1660, set. 2013. ISSN 0167-739X. Disponível em: <http://dx.doi.org/10.1016/j.future.2013.01.010>.

HA, Y.-G.; SOHN, J.-C.; CHO, Y.-J. ubihome: An infrastructure for ubiquitous home network services. In: *Consumer Electronics, 2007. ISCE 2007. IEEE International Symposium on*. [S.l.: s.n.], 2007. p. 1–6.

ISTAG. *Ambient Intelligence: from vision to reality*. [S.l.], September 2003.

JIANHUA, M. et al. A smart ambient sound aware environment for be quiet reminding. In: *Parallel and Distributed Systems, 2005. Proceedings. 11th International Conference on*. [S.l.: s.n.], 2005. v. 2, p. 412–416. ISSN 1521-9097.

KAED, C. E. et al. Insight: interoperability and service management for the digital home. In: *Proceedings of the Middleware 2011 Industry Track Workshop*. New York, NY, USA: ACM, 2011. (Middleware '11), p. 3:1–3:6. ISBN 978-1-4503-1074-1. Disponível em: <http://doi.acm.org/10.1145/2090181.2090184>.

KAED, C. E.; DENNEULIN, Y.; OTTOGALLI, F.-G. Dynamic service adaptation for plug and play device interoperability. In: *Proceedings of the 7th International Conference on Network and Services Management*. Laxenburg, Austria, Austria: International Federation for Information Processing, 2011. (CNSM '11), p. 46–55. ISBN 978-3-901882-44-9. Disponível em: <http://dl.acm.org/citation.cfm?id=2147671.2147679>.

KAR, S. *Ambient Intelligence: Sensing a Future Mobile Revolution*. nov. 2013. Disponível em: <http://siliconangle.com/blog/2013/03/07/ambient-data-towards-a-future-mobile-revolution>.

KIM, J. E. et al. Seamless integration of heterogeneous devices and access control in smart homes. p. 206–213, 2012.

KITCHENHAM, B. et al. Systematic literature reviews in software engineering - a tertiary study. *Inf. Softw. Technol.*, Butterworth-Heinemann, Newton, MA, USA, n. 8, ago. 2010. ISSN 0950-5849. Disponível em: <http://dx.doi.org/10.1016/j.infsof.2010.03.006>.

KRIKKE, J. T-engine: Japan's ubiquitous computing architecture is ready for prime time. *Pervasive Computing, IEEE*, v. 4, n. 2, p. 4–9, 2005. ISSN 1536-1268.

KUMAR, B.; RAHMAN, M. Mobility support for universal plug and play (upnp) devices using session initiation protocol (sip). In: *Consumer Communications and Networking Conference, 2006. CCNC 2006. 3rd IEEE*. [S.l.: s.n.], 2006. v. 2, p. 788–792.

LAI, C.-F.; HUANG, Y.-M. Context-aware multimedia streaming service for smart home. In: *Proceedings of the International Conference on Mobile Technology, Applications, and Systems*. New York, NY, USA: [s.n.], 2008. (Mobility '08), p. 106:1–106:5. ISBN 978-1-60558-089-0. Disponível em: <http://doi.acm.org/10.1145/1506270.1506399>.

MATTERN, F.; FLOERKEMEIER, C. From active data management to event-based systems and more. In: SACHS, K.; PETROV, I.; GUERRERO, P. (Ed.). Berlin, Heidelberg: Springer-Verlag, 2013. cap. From the Internet of Computers to the Internet of Things, p. 242–259. ISBN 3-642-17225-3, 978-3-642-17225-0. Disponível em: <http://dl.acm.org/citation.cfm?id=1985625.1985645>.

MITSUGI, J. et al. Bridging upnp and zigbee with coap: protocol and its performance evaluation. In: *Proceedings of the workshop on Internet of Things and Service Platforms*. New York, NY, USA: ACM, 2011. (IoTSP '11), p. 1:1–1:8. ISBN 978-1-4503-1043-7. Disponível em: <http://doi.acm.org/10.1145/2079353.2079354>.

OVIDIU, V. et al. Internet of things strategic research roadmap. *Comput. Netw.*, p. 10–51, 2011. Disponível em: <http://ebookbrowsee.net/iot-cluster-strategic-research-agenda-2011-pdf-d149015693>.

PATEL, K.; ANAND, S.; KUMART, S. S. A robust qos framework on android for effective media delivery to dlna enabled home gateway in smart home environment. In: *Wireless Communications, Networking and Information Security (WCNIS), 2010 IEEE International Conference on*. [S.l.: s.n.], 2010. p. 217–222.

REAZ, M. B. I. Artificial intelligence techniques for advanced smarthome implementation. *Acta Technica Corvininesis - Bulletin of Engineering*, 2013.

RICQUEBOURG, V. et al. The smart home concept : our immediate future. In: *E-Learning in Industrial Electronics, 2006 1ST IEEE International Conference on*. [S.l.: s.n.], 2006. p. 23–28.

RUS, C. et al. Mobile tv content to home wlan. *Consumer Electronics, IEEE Transactions on*, v. 54, n. 3, p. 1038–1041, 2008. ISSN 0098-3063.

SAMPAIO, R. F.; MANCINI, M. C. Estudos de revisÃ£o sistemÃ¡tica: Um guia para sÃntese criteriosa da evidÃªncia cientÃfica. In: *Companion Proceedings of the 11th Brazilian Symposium on Human Factors in Computing Systems*. Brazilian Computer Society, 2012. p. 83–89. ISBN 978-85-7669-262-1. Disponível em: <http://dl.acm.org/citation.cfm?id=2400076.2400099>.

THINAGARAN, P.; ABDUL, R.; CHUI, Y. Interoperability for smart home environment using web services. *International Journal of Smart Home*, p. 1–16, 2008.

TOMAS, G. H. R. P. et al. Smart cities architectures - a systematic review. In: HAMMOUDI, S. et al. (Ed.). *ICEIS (2)*. SciTePress, 2013. p. 410–417. ISBN 978-989-8565-60-0. Disponível em: <http://dblp.uni-trier.de/db/conf/iceis/iceis2013-2.html>.

URIBARREN, A. et al. A middleware platform for application configuration, adaptation and interoperability. In: *Self-Adaptive and Self-Organizing Systems Workshops, 2008. SASOW 2008. Second IEEE International Conference on.* [S.l.: s.n.], 2008.

VENKITARAMAN, N. Wide-area media sharing with upnp/dlna. In: *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE.* CI1, 2, 3: [s.n.], 2008. p. 294–298. ——.

WANG, K.-K. et al. Multiagent control system with mobile ubiquitous platform for ambient intelligence. In: *Intelligent Environments, 2008 IET 4th International Conference on.* [S.l.: s.n.], 2008. p. 1–7. ISSN 0537-9989.

WARRIACH, E. et al. Heterogeneous device discovery framework for the smart homes. In: *GCC Conference and Exhibition (GCC), 2011 IEEE.* [S.l.: s.n.], 2011. p. 637–640.

WEISER, M. The computer for the twenty-first century. *Pervasive Computing, IEEE*, v. 265, n. 3, p. 94–104, 1991.

WEISER, M. The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, ACM, New York, NY, USA, v. 3, n. 3, p. 3–11, jul. 1999. ISSN 1559-1662. Disponível em: <http://doi.acm.org/10.1145/329124.329126>.

ZUALKERNAN, I. et al. Infopods: Zigbee-based remote information monitoring devices for smart-homes. *Consumer Electronics, IEEE Transactions on*, v. 55, n. 3, p. 1221–1226, 2009. ISSN 098-363.

APÊNDICE A

ID	TÍTULO	FONTE	AVALIAÇÃO	CRITÉRIOS	QUESTÕES
1	Case Study: The Condition of Ubiquitous Computing Application in Indonesia	ACM	Não Aceito	CE-1	–
2	Study of ubiquitous learning environment based on Ubiquitous computing	ACM	Não Aceito	CE-1	–
3	A pricing scheme for porter based delivery in integrated RFID-Sensor Networks	IEEE	Não Aceito	CE-1	–
4	UPnP Service and Jini Client	ACM	Aceito	CI-1	QP-1 , 2
5	The Styx IP-Core for ubiquitous network device interoperability	IEEE	Não Aceito	CE-1	–
6	Large subsea observatory for earth-ocean science: Challenges of multidisciplinary integration across hardware, software, and people networks	IEEE	Não Aceito	CE-1	–
7	Requirements for smart home applications and realization with WS4D-PipesBox	IEEE	Aceito	CI-1	QP-1 , 2
8	The iBICOOP middleware: Enablers and services for emerging pervasive computing environments	IEEE	Não Aceito	CE-1	–
9	Universal Access Platform to Mobile Instant Messaging	IEEE	Não Aceito	CE-1	–
10	Implementation of the EXI Schema on Wireless Sensor Nodes Using Contiki	IEEE	Não Aceito	CE-1	–
11	Oxygen Cylinders Management Architecture Based on Internet of Things	IEEE	Não Aceito	CE-1	–
12	A Novel Approach to Web of Things: M2M and Enhanced Javascript Technologies	IEEE	Não Aceito	CE-1	–
13	Service integration with UPnP agent for an ubiquitous home environment	IEEE	Aceito	CI-1	QP-4
14	Radio-to-router interface technology and its applicability on the tactical edge	IEEE	Não Aceito	CE-1	–
15	Interconnecting multiple home networks services	ACM	Aceito	CI-1	QP-1
16	ATHENA Project-based Example of DVB-T - WiMax Interoperability	IEEE	Não Aceito	CE-1	–
17	Towards behaviour-aware compositions of things in the future internet	ACM	Aceito	CI-1	QP-2
18	Proxying UPnP service discovery and access to a non-IP Bluetooth network on a mobile phone	ACM	Aceito	CI-1	QP-2
19	Developing search strategies for detecting relevant experiments	IEEE	Não Aceito	CE-1	–
20	MNFL: the monitoring and notification flow language for assistive monitoring	ACM	Não Aceito	CE-1	–
21	A Ubiquitous Model for Wireless Sensor Networks Monitoring	ACM	Não Aceito	CE-1	–
22	INSIGHT: interoperability and service management for the digital home	IEEE	Aceito	CI-1	QP-1, 2, 3, 4
23	Dynamic service adaptation for plug and play device interoperability	ACM	Aceito	CI-1	QP-2, QP-3
24	A review of mobile location privacy in the Internet of Things	ACM	Não Aceito	CE-1	–
25	Study and Simulation of GTS Allocation in Beacon Enabled IEEE 802.15.4	ACM	Não Aceito	CE-1	–
26	M2M-based metropolitan platform for IMS-enabled road traffic management in IoT	ACM	Não Aceito	CE-1	–
27	Ubiquitous services in home networks offered through digital TV	IEEE	Aceito	CI-1	QP-1,QP-2
28	CanCore: best practices for learning object metadata in ubiquitous computing environments	ACM	Não Aceito	CE-1	–
29	Highlights of consumer electronics show (ces) 2013	IEEE	Não Aceito	CE-1	–
30	Satellite communications, free space optics and wireless LAN combined: worldwide broadband wireless access independent of terrestrial infrastructure	ACM	Não Aceito	CE-1	–

31	CoAP over SMS: Performance evaluation for machine to machine communication	ACM	Não Aceito	CE-1	–
32	Toward autonomic pervasive computing	ACM	Não Aceito	CE-1	–
33	First IEEE international workshop on the Web of Things WoT 2010: Message from the workshop chairs	ACM	Não Aceito	CE-1	–
34	ubiHome: An Infrastructure for Ubiquitous Home Network Services	IEEE	Aceito	CI-1	QP-1,2,3
35	Multimodal appliance cooperation based on explicit goals: concepts \& potentials	ACM	Não Aceito	CE-1	–
36	UMAR: Ubiquitous Mobile Augmented Reality	IEEE	Não Aceito	CE-1	–
37	Mobile and ubiquitous malware	IEEE	Não Aceito	CE-1	–
38	Intelligent Traffic Management Systems for Work Zones and Incident Management	IEEE	Aceito	CI-1	QP-1
39	An adaptive middleware framework for context-aware applications	IEEE	Não Aceito	CE-1	–
40	eHealth service support in IPv6 vehicular networks	IEEE	Não Aceito	CE-1	–
41	An Architecture Based on Internet of Things to Support Mobility and Security in Medical Environments	IEEE	Não Aceito	CE-1	–
42	Personal Network Routing Protocol (PNRP) for Personal Ubiquitous Environments	IEEE	Não Aceito	CE-1	–
43	Networking in a smart home - Providing lightweight networking services for heterogeneous devices	IEEE	Aceito	CI-1	QP-2
44	Centaurus: an infrastructure for service management in ubiquitous computing environments	IEEE	Não Aceito	CE-3	–
45	Centaurus: an infrastructure for service management in ubiquitous computing environments	ACM	Não Aceito	CE-2	–
46	Improved decision support system (IDSS) transforming the smart network-edge	IEEE	Não Aceito	CE-1	–
47	A Network Connectivity Power-Saving Mechanism for Mobile Devices in DLNA Home Networks	ACM	Não Aceito	CE-1	–
48	Emotional LED sign board embedding political campaign or positioning information in things of Internet environment	IEEE	Não Aceito	CE-1	–
49	Attaching context-aware services to moving locations	IEEE	Não Aceito	CE-1	–
50	Seamless Integration of Heterogeneous Devices and Access Control in Smart Homes	ACM	Aceito	CI-1	QP-1, QP-2
51	Effective Variants of the Max-Sum Algorithm for Radar Coordination and Scheduling	IEEE	Não Aceito	CE-1	–
52	Automatic communication standard recognition in wireless smart home networks	IEEE	Não Aceito	CE-1	–
53	Mobility support for universal plug and play (UPnP) devices using session initiation protocol (SIP)	ACM	Aceito	CI-1	QP-3, QP-4
54	Context-aware multimedia streaming service for smart home	IEEE	Aceito	CI-1	QP-1, QP-2
55	Supporting efficient machine-to-machine communications in the future mobile internet	IEEE	Não Aceito	CE-1	–
56	Automated application-specific tuning of parameterized sensor-based embedded system building blocks	ACM	Não Aceito	CE-1	–
57	Enabling nonexpert construction of basic sensor-based systems	IEEE	Não Aceito	CE-1	–
58	Email Messaging over Heterogeneous Networks and Interfaces	ACM	Não Aceito	CE-1	–
59	Stumbling Forward into the Connected Future	ACM	Não Aceito	CE-1	–
60	Domotic technologies incompatibility becomes user transparent	ACM	Aceito	CI-1	QP-1, QP-2
61	Bridging UPnP and ZigBee with CoAP: protocol and its performance evaluation	ACM	Aceito	CI-1	QP-1,2,3,4
62	Mobile Wireless Sensor Network: Architecture and Enabling Technologies for Ubiquitous Computing	ACM	Não Aceito	CE-1	–
63	Intelligent network access and inter-system handover control in heterogeneous wireless networks for smart space environments	IEEE	Não Aceito	CE-1	–
64	UbiREAL: realistic smartspace simulator for systematic testing	IEEE	Não Aceito	CI-1	–
65	Functionality configuration for eHome systems	ACM	Não Aceito	CI-1	–
66	A fieldwork of the future with user enactments	ACM	Não Aceito	CE-1	–
67	Ecosystem analysis in the design of open platform-based in-home healthcare terminals towards the internet-of-things	ACM	Não Aceito	CI-1	–
68	Plastic: a metaphor for integrated technologies	ACM	Não Aceito	CI-1	–
69	Mobile TV content to home WLAN	IEEE	Aceito	CI-1	QP-3, QP-4
70	Software testing for mobile and ubiquitous computing	ACM	Não Aceito	CE-3	–
71	Networked surfaces: a new concept in mobile networking	ACM	Não Aceito	CE-3	–
72	WiMAX on the road to future	ACM	Não Aceito	CI-1	–
73	WiMAX on the road to future	IEEE	Não Aceito	CE-2	–

74	Distributed web-based management framework for ambient reconfigurable services in the intelligent environment	ACM	Não Aceito	CE-1	–
75	Distributed web-based management framework for ambient reconfigurable services in the intelligent environment	IEEE	Não Aceito	CE-2	–
76	Ubiquitous Semantic Space: A context-aware and coordination middleware for Ubiquitous Computing	ACM	Não Aceito	CE-1	–
77	UCSMssp: Ubiquitous computing service model based on SPKI/SDSI and P2P	IEEE	Não Aceito	CE-1	–
78	Feasibility of Wi-Fi enabled sensors for Internet of Things	IEEE	Não Aceito	CE-1	–
79	Design recommendations for a reliable body-worn patient monitoring and alarming service	IEEE	Não Aceito	CE-1	–
80	Smiling makes us happier: enhancing positive mood and communication with smile-encouraging digital appliances	ACM	Não Aceito	CE-1	–
81	A secure infrastructure for service discovery and access in pervasive computing	IEEE	Não Aceito	CE-1	–
82	A Middleware Platform for Application Configuration, Adaptation and Interoperability	IEEE	Aceito	CI-1	QP-1, QP-4
83	A Middleware Platform for Application Configuration, Adaptation and Interoperability	ACM	Não Aceito	CE-2	–
84	Wide-Area Media Sharing with UPnP/DLNA	IEEE	Aceito	CE-1	QP-3, QP-4
85	Cyber Security Concerns for Ubiquitous/Pervasive Computing Environments	ACM	Não Aceito	CE-1	–
86	Design and Implementation of Smart Equipment Management System Based on RFID	IEEE	Não Aceito	CE-1	–
87	Multiagent control system with mobile ubiquitous platform for ambient intelligence	IEEE	Aceito	CI-1	QP-2, QP-3
88	Heterogeneous device discovery framework for the Smart Homes	IEEE	Aceito	CI-1	QP-3, QP-4
89	Handy feedback: connecting smart meters with mobile phones	IEEE	Não Aceito	CE-1	–
90	A Mobile Service Language Based on Mobile Agent for Ubiquitous Computing	ACM	Não Aceito	CE-1	–
91	A Mobile Service Language Based on Mobile Agent for Ubiquitous Computing	IEEE	Não Aceito	CE-2	–
92	A secure interoperable architecture for building-automation applications	ACM	Não Aceito	CE-1	–
93	A framework for constructing semantically composable feature models from natural language requirements	ACM	Não Aceito	CE-1	–
94	An Eco-friendly Hybrid Urban Computing Network Combining Community-Based Wireless LAN Access and Wireless Sensor Networking	ACM	Não Aceito	CE-1	–
95	Service entities model for the internet of things: A bio-inspired collaborative approach	ACM	Não Aceito	CE-1	–
96	InfoPods: Zigbee-based remote information monitoring devices for smart-homes	IEEE	Aceito	CI-1	QP-3, QP-4