



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

JOSÉ GUILHERME VITORATTO

**PROTÓTIPO DE AMBIENTE INTELIGENTE BASEADO EM
RASPBERRY PI, WEB E MOBILIDADE**

Assis
2014

JOSÉ GUILHERME VITORATTO

**PROTÓTIPO DE AMBIENTE INTELIGENTE BASEADO EM
RASPBERRY PI, WEB E MOBILIDADE**

Trabalho de Conclusão de Curso apresentado ao
Instituto Municipal de Ensino Superior de Assis,
como requisito do Curso de Graduação

Orientador: Prof. Esp. Guilherme de Cleve Farto

Área de Concentração: Informática

Assis
2014

FICHA CATALOGRÁFICA

VITORATTO, José Guilherme

Protótipo de ambiente inteligente baseado em *Raspberry Pi*, *Web* e *Mobilidade* / José Guilherme Vitoratto. Fundação Educacional do Município de Assis – FEMA – Assis, 2014.

82p.

Orientador: Prof. Esp. Guilherme de Cleve Farto

Trabalho de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis – IMESA

1. Ambiente Inteligente 2. *Raspberry Pi* 3. *Web Services* 4. *Google Android*

CDD: 001.6
Biblioteca da FEMA

PROTÓTIPO DE AMBIENTE INTELIGENTE BASEADO EM RASPBERRY PI, WEB E MOBILIDADE

JOSÉ GUILHERME VITORATTO

Trabalho de Conclusão de Curso apresentado ao
Instituto Municipal de Ensino Superior de Assis,
como requisito do Curso de Graduação, analisado
pela seguinte comissão examinadora:

Orientador: Prof. Esp. Guilherme de Cleva Farto

Analisador (1): Prof. Dr. Alex Sandro Romeo de Souza Poletto

Assis
2014

DEDICATÓRIA

Dedico este trabalho à minha família, amigos, namorada e a todas as pessoas que acreditaram em meus sonhos e anseios, onde sempre estiveram presente me apoiando quando necessário para que assim pudesse realiza-los.

AGRADECIMENTOS

Primeiramente a Deus, pela sua infinita misericórdia e amor, pois sem Ele nada seria possível e não poderíamos estar reunidos aqui para desfrutar deste momento tão especial.

Aos meus familiares José Clóvis Vitoratto, Elisa Lopes Brochado Vitoratto e Rafael Vitoratto, por estarem sempre ao meu lado, me ajudando, apoiando e conduzindo-me durante minha caminhada.

A minha namorada Natália Maioli Pelegrini, pelo incansável apoio, pela ajuda, dedicação e carinho que tem por mim.

Ao meu orientador Prof. Esp. Guilherme de Cleve Farto, pelo incentivo, pelo apoio, pela paciência e pela orientação.

A todos os professores, pelo conhecimento compartilhado, e por contribuírem para o meu crescimento intelectual.

E a todos que direta ou indiretamente contribuíram para a execução deste trabalho os meus sinceros agradecimentos.

“The only place where success comes before work is in the dictionary [...]”

Albert Einstein (1879-1955)

RESUMO

Devido ao rápido aumento de novas tecnologias, o fator de comodidade e segurança em um ambiente qualquer tem sido exigido muito no mercado globalizado.

A proposta deste trabalho foi de pesquisar e compreender sobre como tornar um ambiente inteligente utilizando ferramentas simples e baratas, e como ferramenta o microcomputador *Raspberry Pi* foi escolhido, juntamente com as tecnologias *Google Android* e *Web Service* para realizar o devido trabalho.

Com a finalidade de aplicar os conceitos na fase de pesquisa, foi proposto um estudo de caso, abordando a implementação de um ambiente inteligente, onde foi desenvolvida duas aplicações, uma *Web*, utilizando as tecnologias *Java* e *JSF*, e uma *Mobile*, utilizando a plataforma *Google Android*. Essas duas aplicações foram desenvolvidas para que possam controlar o *Raspberry Pi*, que é o responsável por todo o gerenciamento do ambiente.

Junto ao *Raspberry Pi* foi conectado *LED* e uma câmera, para que esses fossem controlados pelas aplicações desenvolvidas.

Palavras-chave: Ambiente Inteligente; *LED*; Câmera; *Java*; *Web Service*; *Google Android*.

ABSTRACT

Due to advancement in technology, comfort and security in any place has been required in the global market.

The purpose of this study was to investigate and understand about making a intelligent ambience using simple and inexpensive tools, being chosen the Raspberry Pi microcomputer as tool, along with the Google Android and Web service technologies to realize the work.

In order to apply the concepts in the research phase, it was proposed a case study, addressing the implementation of an intelligent ambience, which was developed two applications, one Web, using Java and JSF technologies, and mobile using the platform Google Android.

These two applications have been developed to control the Raspberry Pi, which is responsible for the all management of the ambience.

Along the Raspberry Pi was connected LEDs and a camera, that they were controlled by the applications developed.

Keywords: Intelligent Ambience; led; camera; Java; Web Service; Google Android.

LISTA DE ILUSTRAÇÕES

Figura 1 – Sensores.	28
Figura 2 - Atuadores (CASTILLO, 2009).	29
Figura 3 - Controlador gerenciando (FERREIRA, 2008).....	30
Figura 4 - Estruturas das placas do <i>Raspberry Pi</i> (In: Hack Things, 2013).....	32
Figura 5 - Processador (In: UPTON; HALFACREE, 2013, p.27).....	34
Figura 6 - Parte de trás da placa <i>Raspberry Pi</i> (In: GEEK MUNDO).....	35
Figura 7 - Porta <i>Ethernet</i> (In: UPTON; HALFACREE, 2013, p.42).....	36
Figura 8 - Conector HDMI prateado, para saída de vídeo de alta definição (In: UPTON; HALFACREE, 2013, p.42).	37
Figura 9 - LEDs de Status (In: ADAFRUIT, 2013).....	38
Figura 10 - Cases do <i>Raspberry Pi</i>	41
Figura 11 - Desktop <i>Raspberry Pi</i>	42
Figura 12 - Protocolo de comunicação de um Web Service.....	45
Figura 13 - Especificação de mensagens em WSDL (In: RIBEIRO et al., 2004) .48	
Figura 14 - As principais estruturas de dados UDDI (In: COULOURIS et al., 2013).....	49
Figura 15 - Ciclo de Vida de um Web Service (In: LOPES; RAMALHO, 2004)....	51
Figura 16 – Empresas que participam do grupo Open Handset Alliance (In: Busy Programmer, 2013).	54
Figura 17 - Logotipo Google Android (In: Busy Programmer, 2013).	54
Figura 18 - Emulador Android SDK	56
Figura 19 - Camadas da plataforma Google Android.....	57
Figura 20 - Placa de Policarbonato.....	62
Figura 21 – <i>Raspberry Pi</i> Modelo B.	65

Figura 22 - Case.....	65
Figura 23 - Fonte de Alimentação 5 Volts.	66
Figura 24 - Cabo de Fita de 26 pinos.	67
Figura 25 - Cabo de fita conectado ao Raspberry Pi.....	68
Figura 26 – Protoboard 170 furos.	68
Figura 27 - Câmera do Raspberry Pi.....	69
Figura 28 – Maquete.....	70
Figura 29 - Interface do RPi Cam Control.....	71
Figura 30 – Overclock.	73
Figura 31 - Trecho do código JSF.....	74
Figura 32 - Método Ligar e Desligar da classe principal.....	74
Figura 33 - Interface Mobile.	75

LISTA DE TABELAS

Tabela 1 - Comparativo dos dois modelos de *Raspberry Pi* (ELEMENT14).33

SUMÁRIO

1	INTRODUÇÃO.....	16
1.1	OBJETIVOS.....	18
1.2	JUSTIFICATIVAS	19
1.3	MOTIVAÇÃO	20
1.4	ESTRUTURA DO TRABALHO.....	21
2.	AMBIENTES INTELIGENTES	23
2.1	DEFINIÇÕES DE AMBIENTES INTELIGENTES.....	23
2.2	BENEFICIOS DE UM AMBIENTE INTELIGENTE	24
2.2.1	Comodidade.....	25
2.2.1.1	Deficientes Físicos.....	25
2.2.2	Economia	25
2.2.2.1	Meio ambiente.....	26
2.2.3	Segurança	26
2.3	AMBIENTES INTELIGENTES NO BRASIL	27
2.4	DISPOSITIVOS DE UM AMBIENTE INTELIGENTE.....	28
2.4.1	Sensores	28
2.4.2	Atuadores.....	29
2.4.3	Interface	29
2.4.4	Controlador	30
3	COMPUTAÇÃO FÍSICA COM <i>RASPBERRY PI</i>.....	31
3.1	HISTÓRIA DO <i>RASPBERRY PI</i>	31
3.2	EPECIFICAÇÕES DO <i>RASPBERRY PI</i>	32

3.2.1	Periféricos do <i>Raspberry Pi</i>.....	33
3.2.1.1	Processador	34
3.2.1.2	<i>Slot</i> para cartão de memória SD.....	35
3.2.1.3	Porta USB	36
3.2.1.4	Porta <i>Ethernet</i>	36
3.2.1.5	Conector HDMI.....	37
3.2.1.6	<i>LEDs</i> de status	38
3.2.1.7	Saída de áudio analógico	39
3.2.1.8	Saída de vídeo composto	39
3.2.1.9	Entrada de energia	39
3.2.2	Pinos de energia de Entrada e Saída.....	39
3.2.3	Gabinete	40
3.3	SISTEMA OPERACIONAL	41
4	<i>WEB SERVICES</i>.....	44
4.1	INTRODUÇÃO À <i>WEB SERVICE</i>	44
4.2	PADRÕES DE <i>WEB SERVICES</i>	45
4.2.1	<i>eXtensible Markup Language (XML)</i>	46
4.2.2	<i>Simple Object Access Protocol (SOAP)</i>	46
4.2.3	<i>Hypertext Transfer Protocol (HTTP)</i>	47
4.2.4	<i>Web Services Description Language (WSDL)</i>	47
4.2.5	<i>Universal Description Discovery and Integration (UDDI)</i>	48
4.3	CICLO DE VIDA DE UM <i>WEB SERVICE</i>	50
5	TECNOLOGIA <i>GOOGLE ANDROID</i>.....	52
5.1	INTRODUÇÃO AO <i>GOOGLE ANDROID</i>	52
5.2	OPEN HANDSET ALLIANCE.....	53

5.3	ESTRUTURA GERAL DO <i>GOOGLE ANDROID</i>	55
5.3.1	<i>Android SDK</i>	55
5.3.2	Arquitetura do <i>Android</i>	57
5.3.2.1	Aplicações	57
5.3.2.2	Framework	58
5.3.2.3	Bibliotecas	58
5.3.2.4	<i>Runtime</i>	59
5.3.2.5	<i>Linux Kernel</i>	60
6	PROPOSTA DE TRABALHO	61
6.1	PROTÓTIPO DO AMBIENTE INTELIGENTE	61
6.1.1	Maquete	61
6.1.2	Dispositivos	62
6.2	CÂMERA.....	63
7	ESTUDO DE CASO	64
7.1	PARTE FÍSICA DO PROJETO.....	64
7.2	PARTE OPERACIONAL	70
7.3	DESENVOLVIDO.....	73
8	CONCLUSÃO	77
8.1	TRABALHOS FUTUROS	77
	REFERÊNCIAS	79

1 INTRODUÇÃO

Os computadores, juntamente com outras tecnologias, vêm ganhando cada vez mais espaço na vida dos seres humanos e um dos dispositivos que vem se tornando comum são os *smartphones* e *tablets* pelo fato de possuírem uma ampla variedade de funcionalidades, recursos e aplicativos que, quando acoplados com outras plataformas, como microcomputadores, como por exemplo, o *Raspberry Pi* e o *Arduino*, tornam possível a realização de atividades extraordinárias. A automatização de ambientes é uma delas, onde, por meio de comandos básicos pode-se abrir ou fechar um portão, acender e apagar luzes, verificar a temperatura ambiente e alguns outros comandos que são executados manualmente, passariam a ser controlados por um dispositivo móvel. Dessa forma, cria-se a possibilidade de controlar uma casa ou escritório a partir de um único aparelho, melhorando então a segurança e a comodidade da pessoa que mora nela.

Segundo Bolzani (2004, p.51) um ambiente inteligente pode ajudar nas atividades diárias que possuem muito tempo ou evitar algumas preocupações tais como: esquecer as janelas abertas quando a previsão de tempo avisou que ia chover; esquecer-se de apagar as luzes ao sair de casa. Ainda, segundo o autor Bolzani (2004), a automação nos permite controlar a residência indiretamente, poupando nosso tempo com tarefas repetitivas, aumentando o conforto, além de economizar energia e dinheiro.

De acordo com Alves e Mota (2003, p.9) os estabelecimentos em modo geral não são inteligentes e nunca serão, mas sim o uso que souberem fazer delas.

Alves e Mota (2003, p.9) definem que se deve chamar de inteligente apenas os ambientes que possuem características capazes de tornar a vida mais simples a quem nelas habita, separando essas características em cinco categorias principais: segurança, economia, conforto, ecologia e integração. Ainda segundo os autores Alves e Mota (2003), para cada pessoa e em cada ocasião, têm-se, certamente, uma atenção diferente para cada uma das categorias, no entanto, essas cinco propriedades quando associadas às tecnologias podem suprir as necessidades e preferências de quem as habita.

Já Bolzani (2004, p.23) define que qualquer eletroeletrônico que desenvolva uma tarefa básica, efetue troca de informações e possibilite comando remoto deva ser considerado um dispositivo inteligente.

Segundo Dias e Pizzolato (2005 apud LIPOVETSKY, 2005, p.14), o mercado entrou em outra fase da modernidade, ganhando um novo aspecto, deixando para trás a ostentação para se preocupar mais com a qualidade de vida, citando como um “luxo emocional” em que cada pessoa procura investir naquilo que lhe oferece prazer. Acredita-se que isto é um dos motivos em que todo o tipo de automação vem crescendo cada dia mais, não por razões de necessidades, mais sim por luxúria.

Além do conforto que é oferecido dentro de um ambiente inteligente, a automatização pode proporcionar também a segurança, um quesito desejável por todos. De acordo com Djehdian (2011), diretor da empresa *BioForLife*, os sistemas de segurança têm crescido e à procura que começou com fechaduras biométricas evoluiu-se para sistemas mais completos, capazes de informar quem entrou em casa e em qual horário. E para tal conforto e segurança, quando acoplados com dispositivos móveis, os custos desses privilégios são reduzidos consideravelmente.

O *Raspberry Pi* que foi considerado como o menor computador, anunciado em 2011 e idealizado pelo inglês Pete Lomas foi lançado em 29 de fevereiro de 2012, e é um computador de código aberto, portanto é uma tecnologia barata, sendo assim, de fácil acesso a todos.

No *Campus Party* de 2013 Pete Lomas, fundador do *Raspberry Pi* destacou que só em 2012 foram vendidos mais de 750 mil dispositivos e que por conta de ser uma tecnologia de custo baixo e com diversas possibilidades de uso, ele vem caindo no gosto dos desenvolvedores.

Não apenas por ser barato que ele vem ganhando espaço no gosto dos desenvolvedores, o preço é apenas um dos atrativos oferecidos pela ferramenta, que tem uma versatilidade de acoplamento imensa, facilitando seu trabalho em conjunto a outras tecnologias, como por exemplo, *Arduino*, *Google Android* e também inúmeros tipos de sensores.

Além do *Raspberry Pi*, temos o *Android*, que foi criado pela *Google* em parceria com a *Open Handset Alliance* é uma plataforma de código aberto para dispositivos

móveis. Essa plataforma funciona como um sistema operacional como os já existentes *Symbian* e *Windows Mobile*, com a diferença de ser baseado em código aberto, o que torna possível qualquer desenvolvedor criar aplicativos para a plataforma *Android*. (FARIA, 2008).

Quando juntados o *Raspberry Pi* e o *Android* para o desenvolvimento da automação, há um avanço tecnológico surpreendente. Segundo Pereira e Silva (2009, p.3) o sistema operacional *Android* foi construído com a intenção de permitir aos desenvolvedores criar aplicações móveis, de onde podem tirar total proveito que um aparelho portátil possa oferecer. Contudo, o *Android* permite criar aplicações que pode ser alinhado junto ao *Raspberry Pi* e fazer o gerenciamento de toda a automatização de uma residência ou qualquer outro tipo de mecanismo, tanto como um ventilador ou até mesmo um automóvel.

Aquino (2007) mostra que com o grande avanço de tecnologia, os dispositivos móveis estão se tornando cada vez mais poderosos em relação à capacidade de armazenamento, de processamento, de comunicação e cada vez mais acessíveis aos consumidores. Dados mostram que há o dobro de pessoas que possuem dispositivo móvel em relação ao número de televisores no mundo. No Brasil, de acordo com a GFK que realiza estudos de mercado, as vendas de smartphones saltaram de 193 mil unidades em março de 2010 para 528 mil em março de 2011.

1.1 OBJETIVOS

O objetivo geral deste trabalho é pesquisar e compreender sobre do *Raspberry Pi* onde será usado alguns sensores que ajudará no gerenciamento de toda a automatização, no qual será totalmente integrado ao *Google Android* e a *Web* com *Java EE*.

A partir deste projeto, será possível proporcionar a inovação, o conforto e a segurança das pessoas que ocupam tal ambiente, onde as funções básicas do lugar podem ser controladas e inspecionadas dentro ou fora dela, tudo isso por meio do *Raspberry Pi*, de um sistema *Web*, e também um aplicativo *Android* para ser

instalado em um smartphone caso o proprietário prefira realizar os comandos via celular.

Pretende-se, com este trabalho, adquirir os conhecimentos necessários para conceber, modelar e desenvolver um protótipo de um ambiente inteligente fazendo-se o uso do *Raspberry Pi*, *Web* e *Google Android*. De forma a tornar possível a elaboração de tal projeto, os seguintes objetivos específicos foram estabelecidos:

- Pesquisar e analisar projetos e aplicações de eletrônicas na área de automação;
- Pesquisar, analisar e descrever a plataforma de prototipagem eletrônica do *Raspberry Pi* e seus sensores, atuadores e diversos componentes a serem utilizados neste projeto;
- Pesquisar e aprender sobre a mobilidade *Google Android*.
- Pesquisar e aprender sobre *Web* com *Java EE*.
- Especificar um estudo de caso:
 - Modelar a arquitetura do projeto e desenvolver a maquete do ambiente inteligente;
 - Desenvolver a arquitetura do ambiente;
 - Desenvolver a aplicação móvel responsável por controlar todo o ambiente;
 - Desenvolver a aplicação *Web* responsável por controlar todo o ambiente;
 - Testar e validar;
 - Descrever os resultados obtidos;

1.2 JUSTIFICATIVAS

Todas as possibilidades de automação de um ambiente têm se tornado mais reais e os custos financeiros de execução também se reduzem a cada ano. Segundo a Associação Brasileira de Automação Residencial (Aureside) cerca de 300 mil residências em todo o país possuem, atualmente, algum tipo de automação e esse

número deverá crescer cinco vezes até 2015. Este é um dos motivos para tal estudo.

Para Teza (2002) um ambiente inteligente é um sistema de comunicação que visa à interconexão de dispositivos, favorecendo a comodidade, segurança, lazer, assistência, comunicação e a economia de energia. E um desses dispositivos que auxiliam na interconexão do ambiente, tem-se a *Web* e a mobilidade que são dispositivos que oferecem uma vasta gama de ferramentas.

Essa tecnologia é um tanto promissora, pois, além do conforto e segurança que lhe são dadas, um ambiente inteligente pode beneficiar quem possui algum tipo de deficiência física, fornecendo o controle total da residência em um único lugar para tornar a vida dessas pessoas mais fácil e agradável.

A automação residencial com o uso do *Raspberry Pi* é um foco de estudo muito interessante e relativamente novo, além disso é uma ferramenta muito útil para o desenvolvimento de qualquer tipo de automação.

A utilização da *Web* em um ambiente inteligente é importante, porque a partir dela pode-se desenvolver e executar sistemas que gerenciam todo o ambiente, tornando a vida do proprietário mais cômoda.

A mobilidade não foge dos padrões da *Web*, pois também se podem desenvolver sistemas para tais dispositivos, onde o proprietário não precisaria necessariamente estar em frente a um computador para realizar as operações dadas ao ambiente, tendo uma maior flexibilidade a partir da utilização de qualquer dispositivo móvel que possua um sistema operacional.

1.3 MOTIVAÇÃO

A automatização de ambientes vem crescendo cada vez mais no mercado juntamente com o uso de novos dispositivos como *smartphones* e *tablets*, destacando-se, entre os mais recentes, o *Raspberry Pi*. Por ser uma tecnologia lançada há pouco tempo, com um preço acessível e uma alta capacidade

computacional, esta plataforma tem ganhado a atenção de estudantes, professores e profissionais das áreas de computação e eletrônica digital.

O desenvolvimento deste projeto consiste no fato de que, a automatização está acontecendo em todos os ambientes em que convivemos, e para tal fato, existe uma necessidade de profissionais para a área. Um estudo realizado pela *Motorola Mobility* mostra que 78% dos brasileiros entrevistados disseram ter interesse em automação residencial. (AURESIDE, 2012). Porém, na automatização em que é usado o *Raspberry Pi* como gerenciador, existe uma escassez de materiais técnicos e científicos relacionados a *Web* e dispositivos móveis que possam controlá-lo.

Outro motivo para o desenvolvimento do projeto é que com a necessidade de profissionais, existe uma grande possibilidade de pessoas que tenham algum tipo de experiência ou conhecimento nesta área possam ingressar no mercado de trabalho. Portanto, com os conhecimentos adquiridos durante a execução deste projeto, terei consideráveis chances de atuar no mercado de trabalho, baseando-se neste tema.

1.4 ESTRUTURA DO TRABALHO

Este trabalho será estruturado nas seguintes partes:

- **Capítulo 1 – Introdução**

Capítulo constituído por uma breve especificação do tema, onde fala sobre as principais tecnologias que compõem esse trabalho.

- **Capítulo 2 – Ambientes Inteligentes**

Capítulo na qual descreve tudo sobre ambientes inteligentes e os principais conceitos que engloba esse fator.

- **Capítulo 3 – Computação Física com *Raspberry Pi***

O capítulo 3 descreve tudo sobre o *Raspberry Pi*, todas as suas funções, tecnologias utilizadas, sistemas operacionais compatíveis, conceitos físico e operacionais.

- **Capítulo 4 – *Web Services***

O capítulo 4 conta toda a evolução do *Web Service* e fala sobre as principais tecnologias que são utilizadas para a criação de um *Web Service*.

- **Capítulo 5 – Tecnologia *Google Android***

O capítulo 5 é composto por toda a história do *Google Android*, sua arquitetura e empresas que investem e fazem parte do grupo que administra toda essa tecnologia.

- **Capítulo 6 – Proposta de Trabalho**

Na proposta de trabalho é descrito todos os processos que será utilizado para a criação deste projeto.

- **Capítulo 7 – Estudo de Caso**

O estudo de caso descreve tudo que foi feito na parte final deste projeto, incluindo pesquisas e desenvolvimentos.

- **Capítulo 8 - Conclusão**

2. AMBIENTES INTELIGENTES

O escopo deste capítulo é apresentar o termo Ambiente Inteligente como um conceito, destacando os aspectos fundamentais e mostrando as circunstâncias apropriadas para seu uso.

A chegada da eletricidade nas residências foi uma inovação na época, na qual modificou significativamente a vida dos moradores, pois a iluminação proporcionou o surgimento de novas formas de entretenimento como rádio e mais tarde os televisores, telefone, entre outros, sendo de grande importância social.

A partir de então, o desenvolvimento de pesquisas e estudos relacionados a equipamentos que necessitam de energia foram evoluindo e aprimorando cada vez mais, focando em novos padrões e métricas, para que fossem criados produtos e serviços mais eficientes, auxiliando as pessoas em suas tarefas.

Para entender melhor o que é um ambiente inteligente, é necessário acompanhar as inovações tecnológicas, os novos produtos lançados pelo mercado, e se possível agregar ao cotidiano para que assim possa-se viver literalmente em um ambiente inteligente.

Neste capítulo encontram-se as definições, as vantagens e a evolução.

2.1 DEFINIÇÕES DE AMBIENTES INTELIGENTES

O termo inteligente usado para referenciar um ambiente mais evoluído requer um pouco mais de cautela. Norving e Russel definem que a Inteligência Artificial é o estudo de agentes, na qual considera um agente aquele que é capaz de perceber seu ambiente por meio de sensores e de agir sobre esse ambiente através de atuadores. Sendo assim, os agentes são aqueles que tomam decisões baseados em determinadas situações, proporcionando maior desempenho destas, agregando certo grau de autonomia e modificando o seu estado inicial (Norving 2002).

Portanto, seguindo a linha de pensamento de Norving e Russel um Ambiente Inteligente pode ser classificado como inteligente se seguir tais requisitos.

Ambiente Inteligente pode ser definido e interpretado de várias maneiras. Quando deixa-se de realizar alguma ação, por causa que, aquela ação sofreu algum tipo de automação e esta passa a realizar-se por si só, pode-se chamar de um ambiente inteligente.

De maneira mais simples, um ambiente inteligente é aquele que realiza comandos e tarefas de forma autônoma, é programado para realizar ações oferecendo ao público mais facilidade e praticidade em relação a tal ação. É aquele ambiente onde as funções são realizadas de modo prático e fácil, através de comandos e dispositivos para tornar isso possível. São ambientes que utilizam tecnologias os tornando interativo.

2.2 BENEFÍCIOS DE UM AMBIENTE INTELIGENTE

Em um ambiente inteligente onde as ações são realizadas de forma automatizada os benefícios são variados, na qual pode oferecer uma maior comodidade as pessoas, assim como gerar economia com os recursos sendo gastos de forma racional. Além disso, proporciona maior segurança que é um fator de grande relevância.

Assim como afirma QUINDERÉ (2009, apud ANGEL, 1993 e NUNES, 2002), a domótica é um sistema que prioriza a melhoria do estilo de vida das pessoas, no quesito conforto, segurança e economia da residência, através de um controle centralizado das funções desta, como água, luz, telefone e sistemas de segurança, entre outros.

Nesta seção, serão apresentados os principais benefícios que um ambiente automatizado pode proporcionar, tais como a comodidade, economia, meio ambiente e segurança.

2.2.1 Comodidade

Ambientes que são acionados automaticamente sejam por um sensor de movimento ou algum outro dispositivo, são convenientes as pessoas já que influência em aspectos da vida cotidiana trazendo maior comodidade. Segundo Bolzani (2004, p.1) pode-se utilizar as residências inteligentes para realizar as tarefas do cotidiano, aquelas que são repetitivas e mecânicas, o que possibilita um uso mais apropriado do tempo em que se perde realizando tarefas básicas e rotineiras.

Messias (2007, p.3) define que um ambiente automatizado pode oferecer quatro tipos de conforto; o conforto ambiental onde existe o controle de temperatura e umidade; conforto visual com diversas opções de iluminação; conforto acústico que tem um som ambiente; e comunicação onde há instalações de redes, telefonia interna e externa.

2.2.1.1 Deficientes Físicos

No mundo inteiro existem deficientes físicos que possuem muitas dificuldades em relação a tarefas rotineiras e sempre estão precisando de ajuda para determinadas situações. Uma casa, um apartamento ou qualquer outro espaço físico pode ser automatizado, simplificando assim a vida de muitas pessoas com necessidades físicas, pois tarefas básicas como ascender e desligar a lâmpada, ligar e desligar a TV e mais inúmeras tarefas pode ser controlado por comandos de voz. A automatização busca fornecer maior acessibilidade, gerando certa independência para pessoas com necessidades especiais.

2.2.2 Economia

A eficiência das ações de um ambiente automatizado gera economia, não havendo desperdício seja de água ou luz em uma casa, por exemplo. As ações e funções são

precisas, acionadas no momento exato não havendo gastos e excessos desnecessários.

O sistema de iluminação automatizado possibilita que a luz artificial ascenda no momento em que a incidência de luz natural estiver baixa evitando o desperdício de luz. Assim como o uso de água, ar condicionado, irrigação de jardins entre outros sistemas que podem ser automatizados. Trazem conforto e praticidade, que por sua vez, proporciona economia.

2.2.2.1 Meio ambiente

Com as preocupações que são levantadas em função do meio ambiente com os desperdícios e ações prejudicando este, muitas são as ações reversas na tentativa de protegê-lo.

Para tanto, a eficiência dos utilitários dentro de uma casa inteligente contribui para o meio ambiente, já que o consumo eficiente de eletricidade, gás e água favorecem os recursos naturais, evitando desperdícios.

2.2.3 Segurança

Nos dias de hoje, a violência nas cidades cresce cada vez mais, preocupando as pessoas e trazendo certo desconforto devido a essa sensação de não estar protegido, sendo assim, a segurança é um aspecto procurado por muitas pessoas. E um ambiente inteligente pode proporcionar essa segurança desejada.

Em seu artigo Wortmeyer, Freitas e Cardoso (2005) salientam que dentre todos os sistemas de automação residencial, o de segurança patrimonial é um dos mais procurados atualmente, substituindo o velho “olho mágico” por câmeras equipadas com sensores de presença que captam movimentos estranhos se algum objeto passar no seu raio de atuação.

Além disso, a segurança também pode ser oferecida aos pais em relação aos seus filhos. Pois o monitoramento das crianças por câmeras e sensores traz tranquilidade e facilidade aos pais, que podem monitorá-los mesmo estando no trabalho ou em algum lugar fora de casa; assim como o quarto das crianças durante a noite.

2.3 AMBIENTES INTELIGENTES NO BRASIL

Com o crescimento da área tecnológica e a quantidade de produtos tecnológicos facilmente aceitos pela sociedade, propiciou o avanço cada vez mais agressivo acerca de tecnologias pelo mundo, como é o caso de um ambiente inteligente. Portanto, no Brasil, em fevereiro de 2000 foi criada a Associação Brasileira de Automação Residencial (AURESIDE), que tem a missão de divulgar conceitos a todos os envolvidos no setor, sejam profissionais, empresas ou usuários; difundir tecnologias, homologar produtos e serviços, treinar e formar profissionais na área da domótica. (AURESIDE, 2013).

Nos últimos três anos, o mercado de automação brasileiro vem crescendo cerca de 35% ao ano em números de projetos. Além disso, o número de empresas nesse setor hoje é mais de 40, e ainda tem um grande potencial de crescimento na área tecnológica brasileira, de acordo com dados e informações do Instituto de Tecnologia do Paraná (TECPAR, 2011).

Com o aumento do uso de tecnologias, e conseqüentemente, o interesse de empresas e pessoas na área da domótica, cresce a necessidade de profissionais especializados na área. Como afirma o diretor executivo da associação Aureside, José Roberto Muratori, o aumento da procura dos brasileiros pela automação residencial e o crescimento da oferta de produtos e serviços do mesmo, e preços mais acessíveis, já se percebe a falta de profissionais qualificados para atender a demanda.

2.4 DISPOSITIVOS DE UM AMBIENTE INTELIGENTE

Para tornar um ambiente inteligente é necessário de informatização, periféricos e *hardwares*. Esses dispositivos irão auxiliar nas funcionalidades para que seja implementado a inteligência artificial, para que assim possa realizar inúmeras funções. Os dispositivos mais comuns e necessários são os sensores e os atuadores. Nesse capítulo será apresentando as suas características e funcionalidades.

2.4.1 Sensores

No mercado da automatização existe uma grande variedade de sensores. Os sensores são capazes de abstrair inúmeras mudanças em um ambiente físico, onde eles podem gerar algumas informações, tais como, temperatura; umidade; proximidade e mais uma infinidade de grandezas existente em um espaço físico.

De acordo com Bolzani (2004, p.102) um sensor é aquele que encaminha a informação de um determinado evento permitindo que um controlador saiba se uma ação enviada foi executada com sucesso ou não.

A figura 1 mostra alguns sensores.



Figura 1 – Sensores.

2.4.2 Atuadores

Assim como os sensores, os atuadores existentes também estão presentes no mercado da automação, são eles os responsáveis por realizar algum tipo de operação sonora, física, visual. Basicamente são eles os responsáveis por realizar qualquer alteração em um ambiente físico.

Segundo Bolzani (2004, p.106), os atuadores são dispositivos que tem sua característica alterada quando recebe algum tipo de impulso elétrico, onde são controlados por algum tipo *software*.

A figura 2 mostra alguns tipos de atuadores.



Figura 2 - Atuadores (CASTILLO, 2009).

Basicamente os atuadores são aqueles dispositivos em que podemos ver ou sentir qualquer alteração que ele promove.

2.4.3 Interface

A Interface é a responsável por interagir com o usuário, ela que irá mostrar a parte gráfica dos dados recebidos pelos sensores e atuadores e pela interface o usuário poderá dar os comandos necessários para realizar as operações. A interface é basicamente os monitores, *smartphones*, *tablets* entre outros.

2.4.4 Controlador

O controlador é capaz de gerenciador todos os dados de entradas e saídas, é o responsável por cuidar de todas as informações dos sensores e dos atuadores, é ele que manda os impulsos elétricos para um atuador e nele que recebe todos os dados de um sensor.

A Figura 3 demonstra a funcionalidade do controlador.

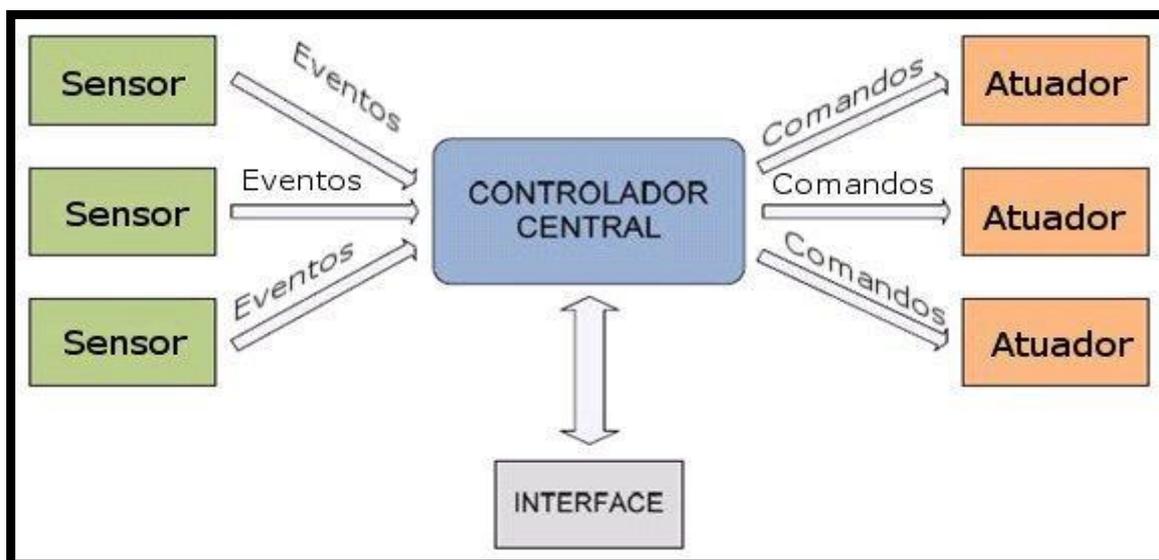


Figura 3 - Controlador gerenciando (FERREIRA, 2008).

O controlador é o responsável por receber e enviar dados para os sensores e para os atuadores, também é ele que se comunica com toda a interface, onde é possível visualizar todos os dados e realizar a interação com o dispositivo.

3 COMPUTAÇÃO FÍSICA COM *RASPBERRY PI*

O intuito deste capítulo é apresentar todos os conceitos do *Raspberry Pi*, destacando e mostrando todos os aspectos fundamentais para o uso do mesmo em diversos tipos de ambientes.

3.1 HISTÓRIA DO *RASPBERRY PI*

De acordo com Sjolgelid (2013, p.7) o *Raspberry Pi* é um computador do tamanho de um cartão de crédito, desenvolvido no Reino Unido pela Fundação *Raspberry Pi*, onde este não possui fins lucrativos. Tudo começou quando um sujeito chamado Eben Upton se reuniu com seus amigos no laboratório de informática da Universidade de Cambridge, para discutir como eles poderiam trazer algum tipo de programação mais simples onde as crianças poderiam aprender algum tipo de algoritmo, para que assim quando chegassem à faculdade, tivessem mais facilidade em programar.

Após vários anos de ajustes e estudos, a fundação surgiu com dois projetos para o *Raspberry Pi*, o Modelo A e o Modelo B. O Modelo B com o valor de 35 dólares foi o primeiro a ser lançado por volta de Fevereiro de 2012, originalmente com 256 MB de Memória RAM, a segunda revisão do Modelo B com 512 MB de memória RAM foi anunciado em outubro de 2012. Já o Modelo A no valor de 25 dólares foi lançado às vendas no primeiro trimestre de 2013.

A Figura 4 demonstra como é a estrutura da placa do *Raspberry Pi* Modelo A e Modelo B.

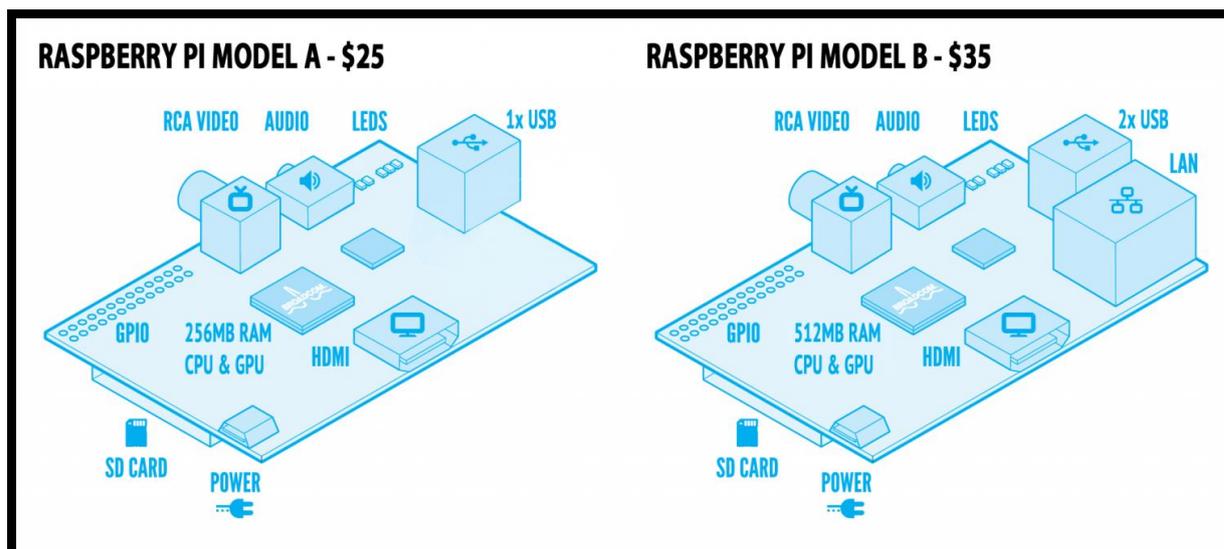


Figura 4 - Estruturas das placas do *Raspberry Pi* (In: Hack Things, 2013)

Atualmente, o modelo mais procurado, é o modelo B, devido as suas funções existentes a mais que o modelo A. O modelo B, além de possuir alguns periféricos a mais, a diferença de valor de um para o outro é muito pouca.

3.2 EPECIFICAÇÕES DO *RASPBERRY PI*

Atualmente existem dois tipos de *Raspberry Pi* o modelo A e o Modelo B. Entre esses dois modelos existem poucas diferenças: o modelo A possui 256 MB de memória RAM, possui apenas uma porta USB 2.0 e não possui porta Ethernet. Já o modelo B em sua primeira versão possui 256 MB de memória RAM, atualmente o modelo B agora possui 512 MB de memória RAM, possui duas portas USB 2.0 e uma porta *Ethernet*.

A tabela a baixo demonstra um comparativo entre os dois modelos:

Características Técnicas	Modelo A	Modelo B
Preço	A partir de 25 dólares	A partir de 35 dólares
Chip	Broadcom BCM2835 SoC Full HD processador de aplicações multimídia	Broadcom BCM2835 SoC Full HD processador de aplicações multimídia

CPU	700 MHz Low Power ARM1176JZ-F Applications Processor	700 MHz Low Power ARM1176JZ-F Applications Processor
GPU	Dual Core VideoCore IV® Multimedia Co-Processor	Dual Core VideoCore IV® Multimedia Co-Processor
Memória	256MB SDRAM	512MB SDRAM
Ethernet	Nenhum	Onboard 10/100 Ethernet RJ45 jack
USB 2.0	Conector USB Único	Conector USB duplo
Saída de Vídeo	HDMI (rev 1.3 and 1.4) Composite RCA (PAL and NTSC)	HDMI (rev 1.3 and 1.4) Composite RCA (PAL and NTSC)
Saída de Áudio	3.5mm jack, HDMI	3.5mm jack, HDMI
Onboard de armazenamento	SD, MMC, SDIO card slot	SD, MMC, SDIO card slot
Sistema Operacional	Linux	Linux
Dimensões	8.6cm x 5.4cm x 1.5cm	8.6cm x 5.4cm x 1.7cm

Tabela 1 - Comparativo dos dois modelos de *Raspberry Pi* (ELEMENT14).

Para um computador do tamanho de um cartão de créditos, o *Raspberry Pi*, tanto o modelo A, quanto o modelo B pode-se dizer que são bem completos e eficientes em relação a *Hardware* e desempenho, o que torna esse dispositivo bastante promissor.

3.2.1 Periféricos do *Raspberry Pi*

Para uma placa que é do tamanho de um cartão de credito, é um pouco tentador pensar no *Raspberry Pi* como uma placa para desenvolvimento de micro controlador como o *Arduino*, ou mesmo, podendo substituir um *notebook*. O *Raspberry Pi* é muito parecido com os componentes interno desses aparelhos, porém possui vários conectores acessíveis a criadores para várias portas e funções. (RICHARDSON; WALLACE, 2013, p.18)

3.2.1.1 Processador

No centro do *Raspberry Pi* encontra-se um processador multimídia *Broadcom BCM2835*, este é o mesmo processador que é encontrado no *iPhone 3G* e no *Kindle 2*, assim é capaz de pensar na capacidade do *Raspberry Pi* como comparáveis a esses poderosos pequenos aparelhos. Este processador é sistema em um chip de 700 MHz de 32 Bits. Isso significa que a grande maioria dos componentes no sistema, incluindo a unidade central e o processamento gráfico, assim como o hardware de áudio e comunicação, está montada em um único componente oculto abaixo do chip da memória. Ele também usa uma arquitetura de conjunto de instruções, conhecida como ARM11. (RICHARDSON; WALLACE, 2013, p.18) e (UPTON; HALFACREE, 2013, p.26).

A Figura 5, mostra como é o processador do *Raspberry Pi*



Figura 5 - Processador (In: UPTON; HALFACREE, 2013, p.27).

Devido a essa estrutura, o *Raspberry Pi* não é compatível com os sistemas tradicionais encontrado nos computadores *Desktop* e nos *notebooks*. A maior parte dos *softwares* de computadores são escritos com base na arquitetura de conjunto de instruções x86, que são encontrados em processadores *AMD*, *Intel* e *VIA*. Por tanto, eles não funcionarão no *Raspberry Pi* que é baseado na arquitetura *ARM*. (UPTON; HALFACREE, 2013, p.27).

3.2.1.2 Slot para cartão de memória SD

O *Raspberry Pi* não possui nenhum tipo de disco rígido, porém utiliza como seu principal dispositivo de armazenamento um cartão de memória SD, onde todos os arquivos são gravados dentro dele inclusive o sistema operacional. O soquete para a conexão com o cartão SD localiza-se na parte inferior da placa *Raspberry Pi*, o que torna o cuidado maior com a placa, pois se acidentalmente o cartão for dobrado, poderão ser quebrados as soldas do soquete e assim estragar o dispositivo. (RICHARDSON; WALLACE, 2013, p.19).

A Figura 6, mostra onde fica localizado o *Slot* para o cartão SD.



Figura 6 - Parte de trás da placa *Raspberry Pi* (In: GEEK MUNDO)

Para fazer a instalação do sistema operacional no *Raspberry Pi* é necessário salvar primeiro o sistema operacional dentro do cartão SD, o que não é uma tarefa tão complicada, pois os procedimentos que deve ser feito é; realizar o *download* do sistema operacional desejado e fazer o *upload* para o cartão SD. É possível instalar mais de um sistema operacional no mesmo cartão SD, o que torna o conceito bastante interessante, pois pode-se mudar o sistema operacional do dispositivo a qualquer hora.

3.2.1.3 Porta USB

As portas USB 2.0 são encontradas nos dois modelos de *Raspberry Pi*, porém o modelo A com apenas uma porta e o modelo B com duas. Algumas das primeiras placas do *Raspberry Pi* foram limitadas quanto a quantidade de corrente que elas poderiam fornecer. Geralmente os dispositivos USB podem chegar até 500mA. A placa do *Raspberry Pi* podia chegar até 100mA, mas recentemente foram feitas algumas revisões nesse quesito onde alcançam as especificações completas das portas USB 2.0. (RICHARDSON; WALLACE, 2013, p.19-20).

3.2.1.4 Porta Ethernet

Diferente do *Raspberry Pi* modelo B que possui uma porta *Ethernet* padrão RJ45, o modelo A não possui essa porta *Ethernet* para conexão com a internet, mas pode ser conectado a uma rede com fios por meio de adaptadores de rede *Ethernet* USB, existe também a conectividade *Wi-Fi* por meio de adaptadores USB externo. Devido a esses dispositivos USB, o modelo A também pode ter acesso à internet. (RICHARDSON; WALLACE, 2013, p.20).

A Figura 7, mostra a porta *Ethernet* do *Raspberry Pi*.

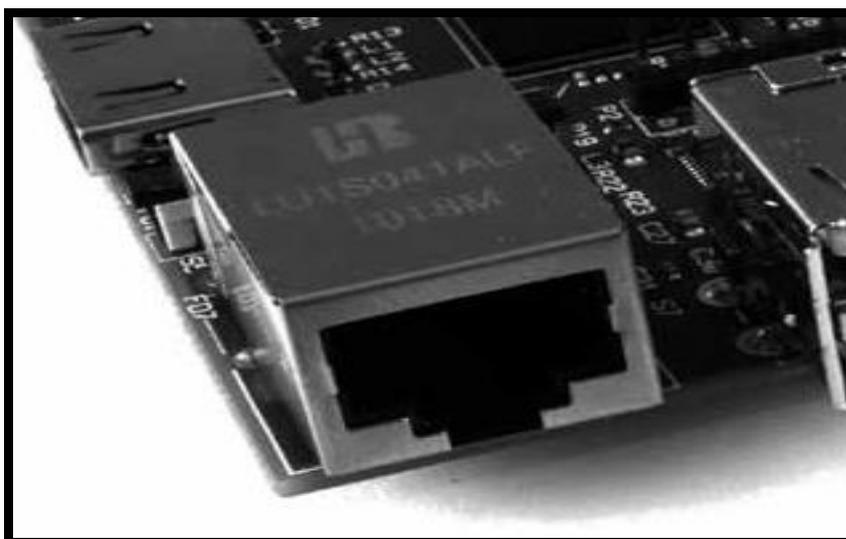


Figura 7 - Porta *Ethernet* (In: UPTON; HALFACREE, 2013, p.42)

3.2.1.5 Conector HDMI

O *Raspberry Pi* também possui em sua placa um conector HDMI. A porta HDMI oferece saída de áudio e vídeo digital, são suportadas catorze resoluções de vídeos diferentes, e o sinal HDMI pode ser convertido para DVI, vídeo composto ou SCART por meio de adaptadores externos. O DVI são cabos geralmente utilizados em monitores; o de vídeo composto são os cabos de sinal de vídeo analógico, que são normalmente transmitidos por um conector RCA amarelo; já o SCART é uma norma europeia para conexão de equipamentos audiovisuais. (RICHARDSON; WALLACE, 2013, p.20).

A Figura 8 mostra onde se localiza a porta HDMI do *Raspberry Pi*.

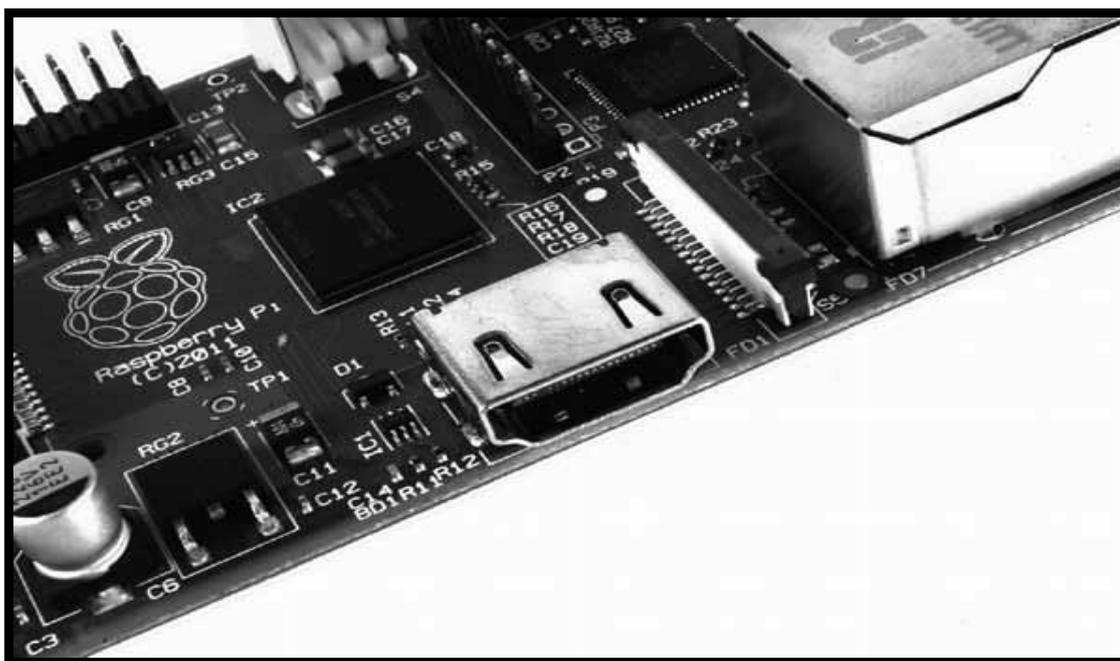


Figura 8 - Conector HDMI prateado, para saída de vídeo de alta definição (In: UPTON; HALFACREE, 2013, p.42).

No entanto o conector *HDMI*, precisa-se de um único cabo para emissão de áudio e vídeo, é o conector que possibilita uma melhor qualidade de imagem, o que torna esses dispositivos um dos mais aceitáveis entre todos os usuários.

3.2.1.6 LEDs de status

Existem cinco *LEDs* indicadores de status no *Raspberry Pi* que podem ser visualizados.

Atualmente os cinco *LEDs* que estão no *Raspberry Pi* são:

- ACT (LED verde): acende quando o cartão SD é acessado.
- PWR (LED vermelho): acende quando conectado a uma alimentação de 3.3V.
- FDX (LED verde): acende quando o adaptador de rede for *full-duplex*.
- LNK (LED verde): Indica alguma atividade de rede.
- 100 (LED amarelo): Indica quando está ligado a uma conexão de rede de 100Mbps.

A Figura 9, mostra os LEDs de status do *Raspberry Pi*.

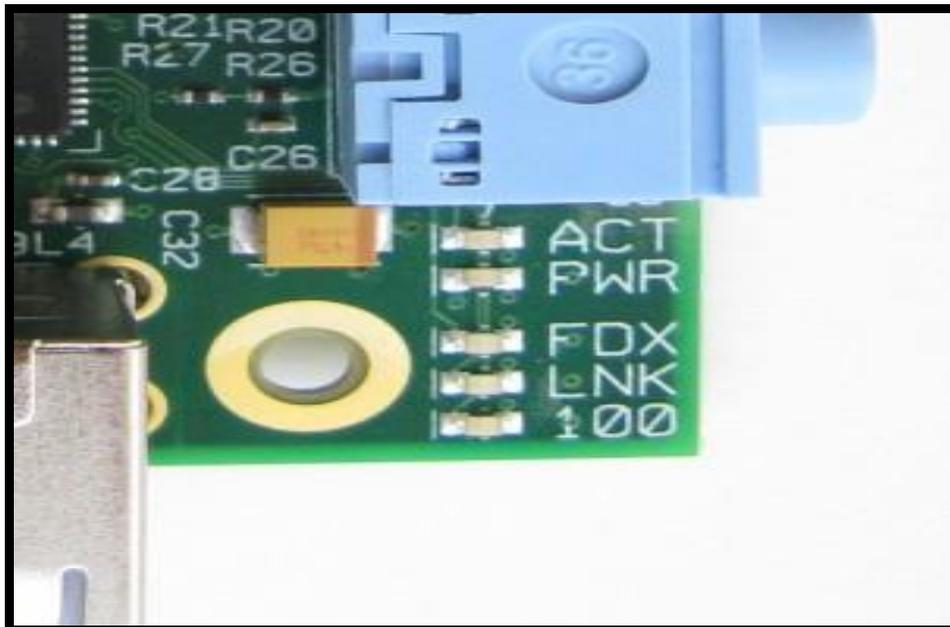


Figura 9 - LEDs de Status (In: ADAFRUIT, 2013).

Esses *LEDs* estão localizados entre as portas *USB* e a saída de áudio analógico, os *LEDs* são consideravelmente importantes, pois possibilita ao usuário a identificação do que o *Raspberry Pi* está realizando no momento.

3.2.1.7 Saída de áudio analógico

A placa também possui um conector de áudio analógico padrão de 3,5mm onde é destinado a conduzir cargas de alta impedância. Nesse caso quando o dispositivo for conectado através desse conector em função de produzir algum tipo de áudio, a qualidade do áudio quando comparado ao do HDMI é um tanto inferior. (RICHARDSON; WALLACE, 2013, p.21).

3.2.1.8 Saída de vídeo composto

A placa possui outra alternativa para a saída de vídeo, possui um conector padrão do tipo RCA que fornece sinais de vídeo composto NTSC ou PAL. Porém quando comparado à qualidade do HDMI esses formatos de vídeo tem resoluções extremamente baixas. (RICHARDSON; WALLACE, 2013, p.21).

3.2.1.9 Entrada de energia

A placa do *Raspberry Pi* não possui nenhum tipo de interruptor de alimentação de energia, para tanto existe um conector micro USB que é usado para fornecer energia. Esse conector micro USB não é uma porta USB adicional, ele serve apenas para a alimentação da placa, foi escolhido o micro USB para tal fator, pois o conector é barato e essas fontes de alimentação USB são fáceis de serem encontradas. (RICHARDSON; WALLACE, 2013, p.21).

3.2.2 Pinos de energia de Entrada e Saída

A placa do *Raspberry Pi* possui quatro pinos de energia e entrada e saída, utilizados para se comunicar com outros dispositivos. Os quatro pinos de energia de entrada e saída são:

- Pinos de Entrada e Saída de Uso Geral (GPIO): São basicamente portas programáveis de entrada e saída, usados nas comunicações entre sensores e atuadores. A placa do *Raspberry Pi* possui 26 pinos.
- Conector de Interface Serial do Display (DSI): Este conector pode receber um cabo *flet* de 15 pinos, onde pode ser usado para se comunicar com alguma tela *LCD* ou *OLED*.
- Conector de Interface Serial da Câmera (CSI): Esta porta permite que um módulo de câmera seja conectado diretamente à placa.
- Conectores P2 e P3: Estas duas linhas de conectores são os conectores *JTAG*, o P2 é usado para teste no *chip Broadcom* e o P3 é usado na rede *LAN*.

3.2.3 Gabinete

Quando compramos um *Raspberry Pi*, ele não vem com nenhum gabinete, também conhecido como *case*. Segundo Monk (2013, p.26), isso ajuda a manter o preço baixo, mas também o deixa bastante vulnerável a quebras.

Possuir um gabinete para proteger a placa do *Raspberry Pi* é muito importante, pois existem várias maneiras em que pode-se danificar a placa. Como o *slot* para o cartão SD, pode ser danificado mecanicamente, mesmo com o uso normal se a placa não for mantida em uma superfície plana. Outra forma é, caso seja colocado à placa sobre uma superfície condutora, a placa pode facilmente entrar em curto-circuito, pois, na parte inferior de placa, existem muitos componentes com uma grande quantidade de solda, material em que pode existir corrente elétrica.

A Figura 10 demonstra algumas cases disponíveis no mercado.



Figura 10 - Cases do Raspberry Pi.

Existem inúmeras formas de *cases* no mercado, porém é encontrado também *cases* feitos artesanalmente que é outra boa opção para se adquirir uma *case*.

3.3 SISTEMA OPERACIONAL

Grande parte dos *desktops* e *notebooks* disponíveis, hoje, utiliza como sistema operacional a *Microsoft Windows* ou *Apple OS X*. Esses sistemas operacionais são de código fonte fechado, criados em um ambiente privado e tem um custo para poder usá-los. Dessa forma o usuário é capaz de obter o *software* final, mas já mais podem ver como eles são feitos. (UPTON; HALFACREE, 2013, p.28).

Por outro lado o *Raspberry Pi* utiliza como sistema operacional o *Linux*, uma plataforma de código aberto, na qual o usuário é capaz de visualizar e alterar o código fonte de todo o sistema operacional. Contudo o *Raspberry Pi* é possível rodar várias distribuições do sistema operacional *Linux*.

Em função do computador *Raspberry Pi* ser baseado em um *chipset* de dispositivo móvel, existem diferenças no quesito de *softwares* em relação a um computador *desktop* ou *notebook*. O processador do *Raspberry Pi* tem características proprietárias que exigem *drivers* de dispositivos especiais e *softwares* que não são

encontrados em nenhuma distribuição *Linux* padrão. (RICHARDSON; WALLACE, 2013, p.27).

Por tanto algumas distribuições *Linux* que visam o *Raspberry Pi* têm sido desenvolvidas, algumas delas são:

- *Raspbian*: é a distribuição oficial e recomendada pela Fundação *Raspberry*, é uma distribuição baseada no *Debian*.
- *Arch Linux*: o *Arch Linux* é uma distribuição que visa especificamente computadores com base na arquitetura ARM. É mais adequada a especialistas em *Linux*.
- *Xbian*: é uma distribuição baseado na *Raspbian*, porém com um foco para usuários que desejam tornar o *Raspberry Pi* como um centro de mídia.
- *QtonPi*: é uma distribuição destinadas a pessoas que desenvolve sofisticados programas gráficos utilizando o *Framework Qt 5*.
- *Occidentalis*: é uma distribuição feita pela *Adafruit* e que tem por base o *Raspbian*, mas com melhorias destinadas a *hackers* de *hardware*.

A Figura 11, mostra como é o ambiente *Desktop* do sistema operacional *Raspbian*.

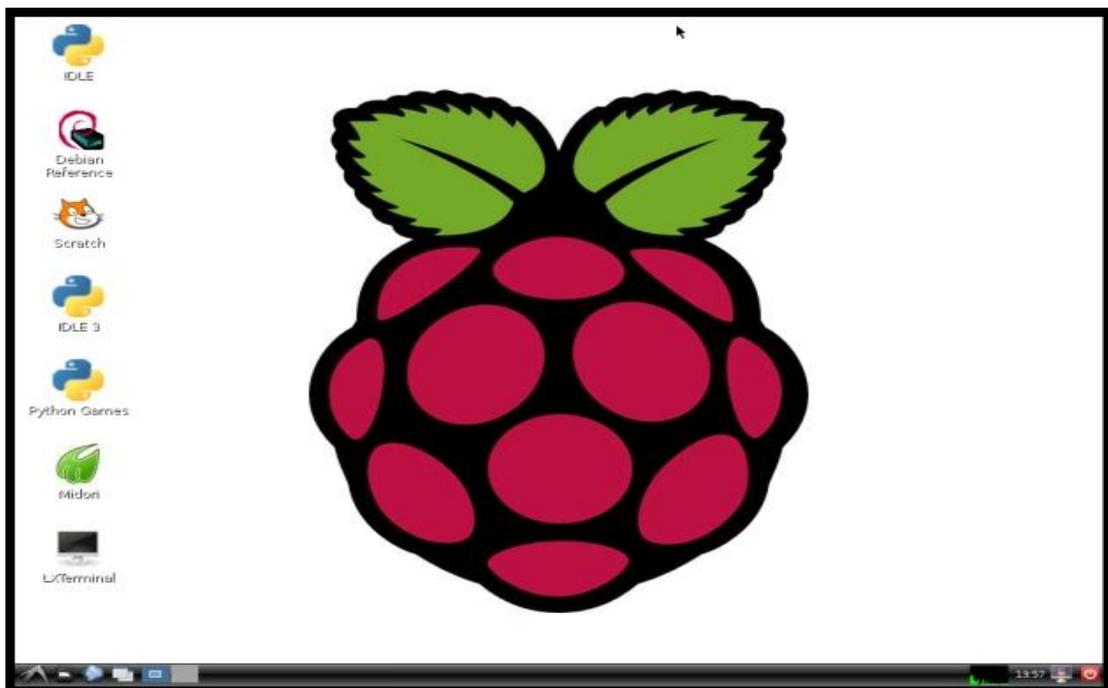


Figura 11 - Desktop Raspberry Pi.

O *Raspberry Pi* é um computador completo, é possível realizar tarefas como qualquer outro computador, e além de ser completo, o *Raspberry Pi* é mais fácil e eficiente na questão de aprendizagem em relação a computação.

4 WEB SERVICES

Um *Web Service* é uma unidade lógica de aplicações, acessível através de protocolos padrões da Internet. Esses serviços possuem uma funcionalidade que pode ser reutilizada sem a preocupação de como é implementada. A ideia central é a criação de serviços que podem ser acessados por usuários conectados à internet a partir de qualquer aplicativo ou programa que utiliza os padrões de comunicação *Web*.

Os *Web Services* surgiram como consequência da utilização da Internet, como uma solução utilizada para a integração de sistemas e na comunicação entre aplicações diferentes, com esta tecnologia é possível que novas aplicações desenvolvidas possam interagir com aquelas que já existem e que sistemas desenvolvidos em ambientes diferentes possam se comunicar entre si.

Pelo fato que existe essa possibilidade de comunicação entre dispositivos e aplicações diferentes, os *Web Services* são um tanto visados pelas empresas, pois podem trazer agilidade para os processos e segurança por não haver intervenção humana durante a comunicação.

4.1 INTRODUÇÃO À WEB SERVICE

Inicialmente a Internet era constituída de páginas estáticas, onde o conteúdo dessas páginas só era passivo de alterações quando a pessoa que mantinha a página realizava alguma alteração. Conhecida como pré-história da Internet. (ALBINADER NETO; LINS, 2006, p.10).

Com a evolução da Internet, as páginas que eram estáticas foram ficando para trás, onde as novas páginas começaram a ser capazes de interagir, acionar programas produtores de informações, provenientes de algum banco de dados. Com essas páginas mais dinâmicas, foi possível ao usuário da Internet, inserir, alterar e excluir informações.

A *Web* tornou-se o cliente universal da internet para ser usado por seres humanos, como ambiente capaz de construir interfaces com o usuário. Devido a este fato, surgiu várias soluções, no lado servidor, para a construção de aplicações que são capazes de extrair dados de várias fontes e disponibilizá-los aos clientes. (ALBINADER NETO; LINS, 2006, p.11).

Na perspectiva conceitual, o emprego do *Web Services* representa um modo para integrar tarefas e processos de negócios por meio da internet, em uma cadeia de valor na qual os procedimentos estão interligados e são interdependentes para atingir um resultado final. (ALBINADER NETO; LINS, 2006, p.11).

4.2 PADRÕES DE WEB SERVICES

Um *Web Service* corresponde a um ou mais serviços de rede da camada de aplicação, independente da linguagem de programação e plataforma utilizada. Para assegurar essa característica, são utilizados padrões abertos para estruturar as comunicações entre o cliente e servidor. Os principais padrões empregados para oferecer esse suporte são o XML, o SOAP, o HTTP, o WSDL e o UDDI. (COSTA, 2008, p. 115).

A Figura 12 mostra a estrutura dos principais padrões de *Web Service*.

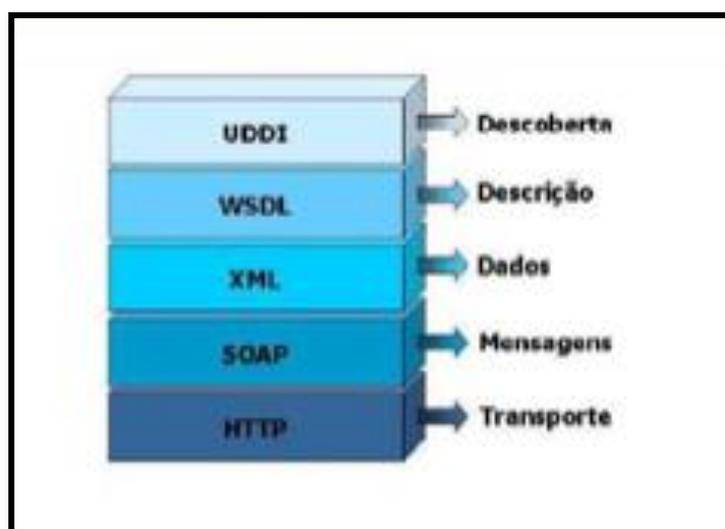


Figura 12 - Protocolo de comunicação de um Web Service

4.2.1 eXtensible Markup Language (XML)

O XML é a escolha natural para o modo de representação dos dados. Quando um usuário acessa algum *Web Service*, os dados que são trocados entre cliente-servidor estão no formato de XML (*eXtensible Markup Language*). O XML é um formato para a criação e transmissão de documentos com dados organizados de forma hierárquica. Pelo fato em que o formato de um documento XML não depender de plataformas de *hardwares* e *softwares* específicos, um banco de dados pode escrever um arquivo XML, e um outro banco diferente, pode ler estes mesmos dados.

O XML é uma linguagem universal, usada para compor dados estruturados de qualquer tipo. Com a troca de dados por meio do XML, as aplicações com estruturas diferentes e linguagens diferentes podem se comunicar de forma coerente, bastando apenas compreender a estrutura utilizada no XML. (COSTA, 2008, p.116).

4.2.2 Simple Object Access Protocol (SOAP).

O SOAP (*Simple Object Access Protocol*) é um protocolo para troca de informações estruturada em plataformas distribuídas. O formato de mensagem do Protocolo Simples de Acesso a Objetos (SOAP), é baseado na estrutura de linguagem de marcação extensível (XML). Para a transmissão de mensagens, ele baseia-se em outros protocolos, especificamente em Camada de Procedimento Remoto (RPC) e Protocolo de Transferência de Hipertexto (HTTP).

O protocolo SOAP foi especificado pelo W3C (*World Wide Web Consortium*), a principal organização de padronização da *World Wide Web*. Muitas implementações desse protocolo estão disponíveis em diferentes linguagens de programação. (COSTA, 2008, p.116).

As mensagens SOAP trafegam pela internet utilizando outros protocolos de aplicação, como o HTTP, SMTP ou FTP. Apesar dessas várias possibilidades de protocolo, o mais utilizado para transportar mensagens SOAP, é o protocolo HTTP.

Isso acontece porque o HTTP é mais adequado a Chamada de Procedimento Remoto (RPC). O SMTP, por sua vez, é mais adequado para envio de notificações SOAP, pois não necessitam de uma resposta. (CARISSIMI; ROCHOL; GRANVILLE, 2009, p.333)

4.2.3 *Hypertext Transfer Protocol* (HTTP)

O Hypertext Transfer Protocol (HTTP) surgiu da necessidade de distribuir informações por meio da internet, e para que o mesmo fosse possível, foi criada uma forma de padronizar a comunicação entre clientes e os servidores da *Web*. Com isso o protocolo HTTP passou a ser utilizado para a comunicação entre os computadores na internet.

O Protocolo de Transferência de Hipertexto (HTTP), é um protocolo de comunicação utilizado para sistemas de informação. Ele é a base para a comunicação de dados da *World Wide Web*.

O Protocolo de Transferência de Hipertexto é um protocolo por trás da World Wide Web. O HTTP está por trás de cada invocação de documento web ou gráfico exibido na página. Ele especifica como qual o tipo de solicitação do cliente e como o servidor deve responder a essas solicitações (WONG, 2000, p.1)

4.2.4 *Web Services Description Language* (WSDL)

O WSDL é mais um protocolo de comunicação *Web Service* que baseia-se na linguagem XML. O *Web Services Description Language* (WSDL) trata-se de um protocolo escrito em formato XML que além de descrever o serviço, especifica como acessá-lo e quais as operações e métodos disponíveis.

Para descrever os serviços de comunicação disponível em um *Web Service*, bem como os detalhes de acesso, como, parâmetros e dados de retorno, utiliza-se o protocolo WSDL. (COSTA, 2008, p.117).

WSDL, possibilita uma separação da descrição das funcionalidades abstratas oferecidas por um serviço dos detalhes concretos da descrição de um serviço, e também como e onde as funcionalidades são oferecidas. Com o uso de WSDL, o cliente não precisa saber qual a linguagem de programação ou plataforma de execução em que o provedor de serviços está baseado. (RIBEIRO et al., 2004, p.38).

A Figura 13 apresenta um exemplo de troca de mensagem WSDL.

```
<message name="recebeNumero">
  <part name="numero" type="xsd:integer"/>
</message>
<message name="enviaResposta">
  <part name="fatorial" type="xsd:float"/>
</message>
<portType name="fatorial_PortType">
  <operation name="fatorial">
    <input message="tns:recebeNumero"/>
    <output message="tns:enviaResposta"/>
  </operation>
</portType>
```

Figura 13 - Especificação de mensagens em WSDL (In: RIBEIRO et al., 2004)

4.2.5 *Universal Description Discovery and Integration (UDDI)*

UDDI (*Universal Description, Discovery and Integration*) é um serviço de registro *Web Service* também especificado pelo W3C que gerencia informações sobre provedores, implementações e metadados de serviços. O UDDI nada mais é do que um serviço de diretório onde empresas podem publicar e descobrir por serviços *Web*.

O UDDI fornece uma interface para publicar e atualizar informações sobre serviços *Web*. Na primeira vez que uma estrutura de dados é publicada em um servidor UDDI, ele recebe uma chave na forma de um URI e esse servidor se torna seu proprietário. (COULOURIS et al., 2013. p.406).

As estruturas de dados suportadas pelo UDDI são projetadas de forma que permita a todos os estilos de acesso anteriores e podem incorporar volume de informações legíveis para as pessoas. Os dados são organizados em termos de quatro estruturas: *businessEntity*; *businessServices*; *bindingTemplate*; *tModel*. Cada uma dessas estruturas pode ser acessada individualmente por meio de um identificador chamado de chave. A Figura 14, mostra como é a estrutura de dados UDDI. (COULOURIS et al., 2013. p.405)

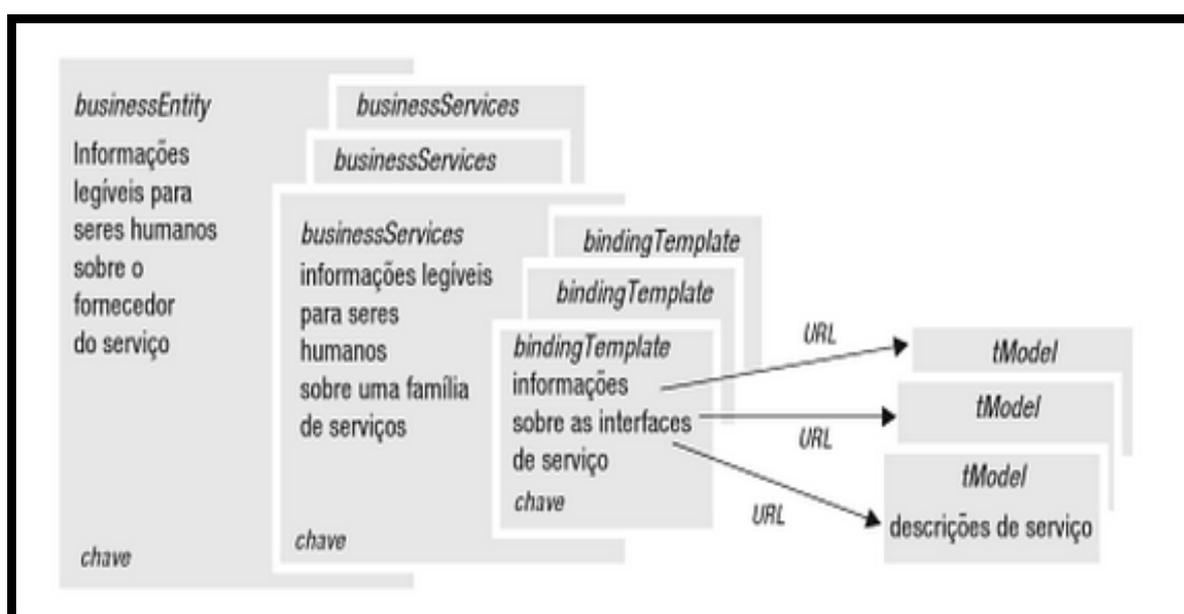


Figura 14 - As principais estruturas de dados UDDI (In: COULOURIS et al., 2013).

O *businessEntity* descreve a organização que fornece esses serviços *Web*, dando seu nome, endereço, atividade, entre outros. O *businessServices* armazena informações sobre um conjunto de instâncias de um serviço *Web*, como seu nome e uma descrição de seu propósito, como por exemplo, um agencia de viagem ou uma livraria. O *bindingTemplate* contém o endereço de uma instância de serviço *Web* e referências para descrições do serviço. E o *tModel* contém descrições de serviço, normalmente documentos WSDL, armazenadas fora do banco de dados e acessadas através de URLs. (COULOURIS et al., 2013. p.405)

4.3 CICLO DE VIDA DE UM *WEB SERVICE*

O ciclo de vida de um *Web Service* é composto por quatro estados distintos, são eles: Publicação; Descoberta; Descrição e Invocação.

O estado da Publicação, é um processo, opcional, através do qual o fornecedor do *Web Service* conhece a existência do serviço, efetuando o registro do mesmo no repositório do *Web Services* (UDDI). (LOPES; RAMALHO, 2004. p.2).

O estado de Descoberta é um processo, opcional, através do qual a aplicação possui conhecimento da existência do *Web Service* pretendido pesquisando em um repositório UDDI. (LOPES; RAMALHO, 2004. p.2)

O estado de Descrição é um processo na qual o *Web Service* expõe o documento WSDL, desta maneira a aplicação tem acesso a toda a interface do *Web Service*, onde encontra-se todas as funcionalidades. (LOPES; RAMALHO, 2004. p.2)

O estado de Invocação é o processo na qual o cliente e o servidor interagem, por meio de envio de mensagens e de eventuais recepções de mensagens. (LOPES; RAMALHO, 2004. p.2)

A junção desses quatro estados permite a construção do ciclo de vida de um *Web Service*. Na qual, o fornecedor desenvolve o serviço utilizando a linguagem de programação que quiser; em seguida especifica a interface do serviço que definiu em WSDL; após o término dos dois primeiros passos, o fornecedor registra o serviço no UDDI; a pessoa que utiliza o serviço pesquisa em um repositório UDDI e encontra o serviço; a aplicação estabelece a ligação com o *Web Service* e estabelece uma comunicação via protocolo SOAP. (LOPES; RAMALHO, 2004. p.3).

A Figura 15 ilustra como é o ciclo de vida de um *Web Service*.

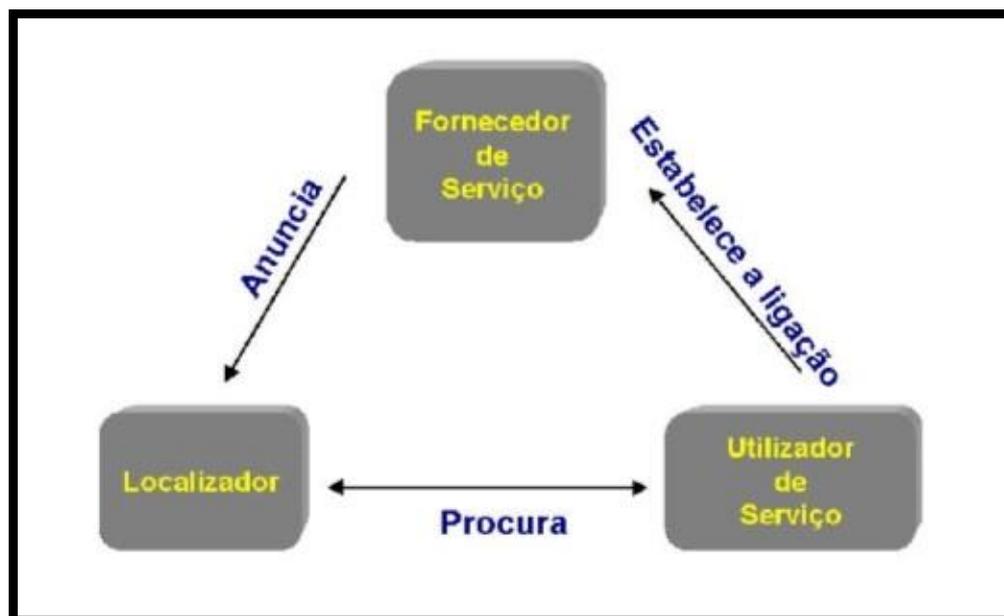


Figura 15 - Ciclo de Vida de um Web Service (In: LOPES; RAMALHO, 2004).

5 TECNOLOGIA GOOGLE ANDROID

O *Android* é uma plataforma desenvolvida pela empresa *Google*, tendo como foco os dispositivos móveis, é uma plataforma totalmente aberta e livre (*Open Source*), divulgada em 5 de novembro de 2007. Atualmente, o sistema *Android* é mantido pela *Open Handset Alliance* (OHA), um grupo de empresas de TI constituído por aproximadamente 84 empresas, que se uniram com o intuito de desenvolver aplicativos e serviços para dispositivos móveis.

Android é um sistema operacional com base no núcleo do *Linux*, possui uma rica interface gráfica, onde os dispositivos possuem algumas aplicações instaladas e também possibilita a instalação de diversos outros tipos de aplicativos.

A plataforma *Android* é uma tecnologia móvel completa, onde envolve um pacote completo de programas para celulares, já com um sistema operacional, *middleware*, aplicativos e interface do usuário. Foi construído com a intenção de permitir aos desenvolvedores criar aplicações para dispositivos móveis e que possam tirar total proveito do que um aparelho portátil possa oferecer. Foi construído para ser uma plataforma completamente aberta, para que assim, uma aplicação possa ter acesso a qualquer funcionalidade do núcleo do sistema operacional, sendo possível intervir em processos como: efetuar chamadas, enviar mensagens de textos ou até mesmo utilizar a câmera do celular. Isso ainda permite aos desenvolvedores adaptarem e evoluírem cada vez mais estas funcionalidades.

5.1 INTRODUÇÃO AO GOOGLE ANDROID

O mercado de celulares está crescendo cada vez mais. Estudos mostram que só no ano de 2013 mais de 3 bilhões de pessoas possuíam algum tipo de aparelho celular, o que corresponde a mais ou menos metade da população mundial. (LECHETA, 2013, p.21).

Nos dias de hoje, as pessoas estão procurando cada vez mais dispositivos móveis com recursos diferentes, não procuram mais celulares comuns, que fazem apenas ligações e envia SMS's, as pessoas procuram por celulares que possuam acesso à internet, que tenham câmeras, que possam acessar e-mails, que tenham GPS e possam utilizá-los para jogar. Devido a essa grande procura por essas novas tecnologias, o mercado de dispositivos móveis está cada vez mais aquecido e a procura por profissionais da área está cada vez maior.

O mercado corporativo também está crescendo muito, e um grande número de empresas estão buscando incorporar aplicações móveis no seu dia-a-dia com o intuito de agilizar seus negócios e integrar os seus aplicativos com seus sistemas de *back-end*. Essas empresas, sem dúvida, sempre buscam lucrar cada vez mais, e os celulares e *smartphones* estão ocupando um importante espaço em um mundo onde a palavra "mobilidade" está cada vez mais conhecida. (LECHETA, 2013, p.21).

O *Android* é o primeiro projeto de uma plataforma *open source* para dispositivos móveis que foi feito em conjunto com a *Open Handset Alliance* (OHA). De acordo com Silva (2012, p.17) um dos primeiros *smartphones* que ofereceu suporte ao sistema operacional *Google Android* foi o aparelho G1 fabricado pela empresa *T-Mobile*.

5.2 OPEN HANDSET ALLIANCE

A *Open Handset Alliance* é um grupo formado por grandes empresas de telefonia de celulares, onde é liderada pela empresa *Google*. A OHA consiste em todas as estruturas envolvidas no processo de telefonia móvel, como: operadoras de celular, fabricantes de aparelhos, empresas semicondutoras, empresas de software e empresas de comercialização.

A Figura 16 mostra algumas empresas que participam do grupo *Open Handset Alliance*.



Figura 16 – Empresas que participam do grupo Open Handset Alliance (In: Busy Programmer, 2013).

O objetivo do grupo *Open Handset Alliance* é definir uma plataforma única e aberta para dispositivos móveis para deixar os consumidores mais satisfeitos com o produto final. Outro objetivo desse grupo de empresas é criar uma plataforma moderna e flexível para o desenvolvimento de aplicações corporativas. E como resultado final desses objetivos proposto pelas empresas, nasceu o tão conhecido e famoso *Android*. (LECHETA, 2013, p.23).

A figura 17 mostra o logotipo escolhido para a plataforma *Android*.

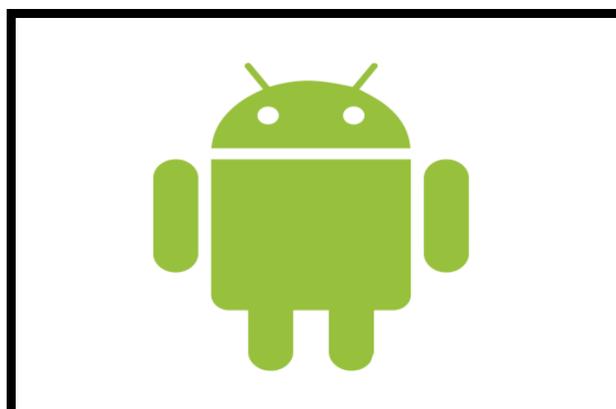


Figura 17 - Logotipo Google Android (In: Busy Programmer, 2013).

Com esta união, todos acabam se beneficiando: os fabricantes de celulares, os comércios, os usuários comuns e, é claro, os desenvolvedores de aplicações. (LECHETA, 2013, p.23).

5.3 ESTRUTURA GERAL DO GOOGLE ANDROID

O *Android* é uma ferramenta de desenvolvimento que tem disponível um conjunto de APIs necessária para o desenvolvimento de novas aplicações para a plataforma *Android*, utilizando a linguagem de programação *Java*. (SILVA, 2012, p.20).

Os recursos que são encontrados na plataforma *Android* são: *Application framework*, que permite a reutilização e substituição de componentes; *Dalvik virtual machine*, que é uma Máquina Virtual Java (JVM) voltada para dispositivos móveis; *Browser* Integrado, baseado no *webkit engine*; Gráficos Otimizados, o *Android* é constituído por bibliotecas 2D e 3D baseada na especificação *OpenGL ES 1.0*; *SQLite*, um Sistema Gerenciador de Banco de Dados (SGBD) já embutido no *Android* para o armazenamento de dados; Suporte multimídia, a plataforma já oferece para vídeo, áudio e imagens; Telefonía GSM; Bluetooth, EDGE, 3G, 4G e *WiFi*; Câmera, GPS, compasso, e acelerômetro; e um Rico ambiente de desenvolvimento, incluindo um emulador de dispositivos, ferramentas de depuração, memória, performance e um *plug-in* para o ambiente de desenvolvimento *Eclipse*. Esse conjunto de recursos encontrados na plataforma *Android* é denominado como *Android SDK*.

5.3.1 *Android SDK*

O *Android SDK* é um sistema utilizado para o desenvolvimento de aplicações no *Android*, onde o mesmo já possui um emulador para fazer a simulação de um celular, ferramentas utilitárias e uma API completa para a linguagem *Java*, com todas as classes necessárias para desenvolver as aplicações. (LECHETA, 2013, p.32).

Embora o SDK possua um emulador que possa ser executado como um aplicativo comum no computador, existe um *plug-in* para o ambiente de desenvolvimento *Eclipse* que visa justamente integrar o ambiente de desenvolvimento *Java* com o emulador. Com o *plug-in* é possível iniciar o emulador diretamente dentro do ambiente de desenvolvimento, instalando a aplicação no emulador automaticamente e, com o *debug* do *Eclipse* integrado, o que torna possível depurar o código-fonte como qualquer outra aplicação *Java*. (LECHETA, 2013, p.32).

A Figura 18 mostra como é o emulador do *Android* SDK.



Figura 18 - Emulador Android SDK

Com o SDK é possível também executar o aplicativo diretamente no celular, basta conectar um cabo USB no celular e no computador, o que acaba tornando o desenvolvimento mais produtivo, pois dessa forma é possível fazer os testes em aparelhos reais.

5.3.2 Arquitetura do *Android*

A arquitetura do sistema operacional *Android* é dividida em camadas, onde cada parte da camada é responsável por gerenciar os seus respectivos processos. A Figura 19 ilustra as camadas existentes na plataforma *Google Android*.

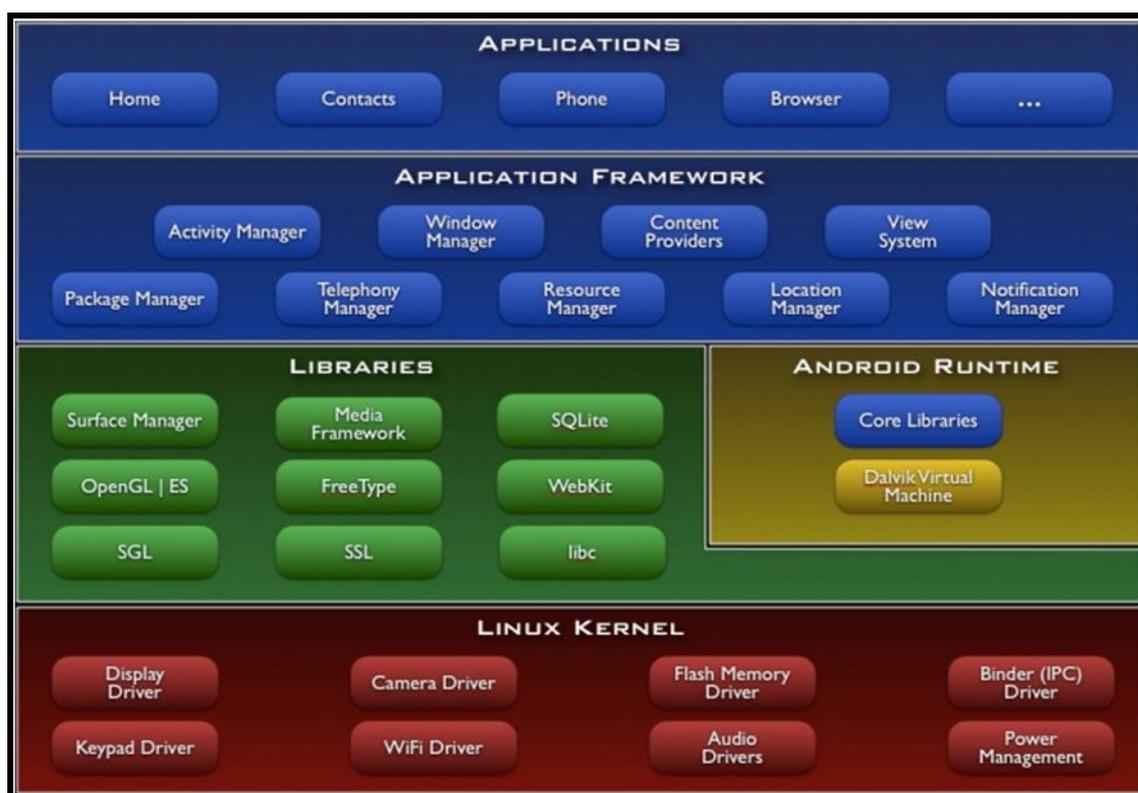


Figura 19 - Camadas da plataforma Google Android.

5.3.2.1 Aplicações

A camada de aplicação é onde localiza-se todos os aplicativos que são executados sobre o sistema operacional. A camada de aplicação está acima de todas as outras camadas, na qual se encontra todos os aplicativos fundamentais do *Android*, tais como: programas de SMS e MMS, cliente de e-mail, navegadores, mapas, calculadora, agenda, contatos, dentre outros aplicativos que executam na plataforma. (PEREIRA; SILVA, 2009, p.6).

5.3.2.2 Framework

Na camada de *framework (Application Framework)*, temos os programas que gerenciam as aplicações básicas do telefone. Nessa camada, é possível que os desenvolvedores tenham acesso total ao *framework*, como um conjunto de ferramentas básicas com a qual podem construir e implementar ferramentas mais complexas.

Na *Application Framework* encontramos todas as APIs e os recursos utilizados pelos aplicativos, como classes visuais, onde incluem listas, caixas de textos, botões, algum navegador *web* embutido que possibilita que uma aplicação possa acessar ou compartilhar informações de outras aplicações, possibilitando a troca de informações entre aplicativos e gerenciadores de recursos, gerenciador de localização, gerenciador de notificação de pacotes e de atividades. (PEREIRA; SILVA, 2009, p.6).

5.3.2.3 Bibliotecas

Camada de biblioteca é a camada que carrega consigo um conjunto de bibliotecas C/C++ que são utilizadas pelo sistema, essa camada também possui bibliotecas de multimídia, visualização de camadas 2D e 3D, funções para navegadores de internet, funções para acelerar o *hardware*, renderização 3D, funções de gráficos, fontes *bitmap* e vetorizadas e funções para ter acesso ao banco de dados *SQLite*.

O *Android* possui bibliotecas desenvolvidas em C/C++, na qual são utilizadas por muitos recursos do sistema. Estas bibliotecas são disponibilizadas para os desenvolvedores por meio de um *framework*. As principais bibliotecas que compõem essa camada são: *Freetype*, usado para renderizar fontes e *bitmaps*; *System C Library*, uma implementação derivada da biblioteca C sintonizada para dispositivos que rodam *Linux*; *Webkit*, é um renderizador de páginas para navegadores com suporte *CSS*, *javascript*, *DOOM* e *AJAX*; *SQLite*, um poderoso banco de dados

relacional; *SGL*, responsável pelos gráficos 2D subjacentes; *Surface Manager*, fornece o acesso aos subsistemas de exibição, como as camadas de aplicações 2D e 3D; *Media Libraries*, são as bibliotecas que suportam os mais populares formatos de áudio e vídeo; *LibWebCore*, um *web browser* utilizado tanto no *Android* quanto para exibições *web*; *3D Libraries*, uma implementação baseada nas *API's* do *OpenGL*. (PEREIRA; SILVA, 2009, p.8).

5.3.2.4 Runtime

Nessa camada instancia-se a máquina virtual *Dalvik*, essa máquina virtual é criada para cada aplicação executada no *Android*. Em relação a desempenho, essa máquina virtual é a melhor, possui maior integração com as novas gerações de *hardwares* e, é projetada para executar vários processos paralelamente.

Dalvik foi projetada para trabalhar em sistemas com baixa frequência de CPU, pouca memória RAM disponível e sistema operacional sem espaço de *Swap*. *Dalvik* também foi projetado para otimizar o consumo de memória, bateria e CPU. (PEREIRA; SILVA, 2009, p.8).

A máquina virtual *Dalvik* executa arquivos no formato “.DEX”, que são classes em *Java* convertidas para a máquina virtual através da ferramenta *DX*, que também é distribuída juntamente com o SDK do *Android*. O *Dalvik Virtual Machine* baseia-se no *kernel* do *Linux* para funcionalidades subjacentes como o encadeamento e a gestão de baixo nível de memória. (PEREIRA; SILVA, 2009, p.8).

Aplicações que são escritas em *Java* são compiladas e executadas utilizando a Máquina Virtual *Dalvik*, uma máquina virtual especializada e desenvolvida para o uso em dispositivos móveis, permitindo que programas sejam distribuídos em formatos binários e possam rodar em qualquer dispositivo com a plataforma *Android*, independentemente do processador utilizado. (PEREIRA; SILVA, 2009, p.9).

5.3.2.5 Linux Kernel

O núcleo do sistema operacional *Android* é baseado na versão 2.6 do *kernel* do *Linux*, herdando diversas características dessa plataforma. Na concepção do *Google Android*, parte importante utilizada do *kernel* do *Linux* é, o controle de processos, gerenciamento de memória, *threads*, protocolos de rede, modelo de drives e a segurança dos arquivos.

O *kernel* também atua como uma camada de abstração entre o *hardware* e o *software*. Um acessório interessante é o *Binder*, que é o responsável por obter e enviar para as aplicações requerente, a interface de serviço da aplicação que fez a requisição, o que possibilita a comunicação Inter processos (IPC). (PEREIRA; SILVA, 2009, p.9).

Nesta camada é encontrado também um sistema próprio de gerenciamento de energia, onde os aplicativos requisitam o gerenciamento de energia e o drive de energia do *kernel* começa a checar periodicamente todos os processos que não estão sendo utilizados por aplicações e os desliga. Tornando assim o consumo de energia menor. (PEREIRA; SILVA, 2009, p.8).

6 PROPOSTA DE TRABALHO

Este trabalho tem como proposta informar e disponibilizar tudo sobre a forma de se criar um ambiente inteligente utilizando o *Raspberry Pi*, Web e Mobilidade. Será proposto como exemplo de um ambiente inteligente, um protótipo de uma casa, onde ela irá ascender e apagar as luzes dos cômodos da casa através de um dispositivo celular, na qual utilizará como sistema operacional o *Google Android*. Além do dispositivo celular, pretende-se criar também um aplicativo *Web*, na qual será possível realizar as mesmas operações do dispositivo móvel. Será possível também visualizar a temperatura e a umidade de dentro da casa, através do dispositivo móvel e do aplicativo *Web*.

O protótipo da casa contará também com uma câmera de vigilância, onde será possível visualizar a imagem da câmera de onde se fizer necessário, sendo possível visualizar através do dispositivo móvel e do aplicativo *Web*.

6.1 PROTÓTIPO DO AMBIENTE INTELIGENTE

O protótipo do ambiente inteligente tem como base uma maquete de uma casa com quatro cômodos, onde, cada cômodo contará com um *LED*, que representará a iluminação da casa. O sensor de temperatura e umidade ficará dentro de algum cômodo da casa e a câmera na frente da casa.

6.1.1 Maquete

A maquete será construída através de placas de policarbonato. O policarbonato é um material termoplástico de alta resistência, utilizado nas mais diversas aplicações comerciais e industriais, devido à transparência, beleza e baixo peso, geralmente usado como toldo, telhas entre outros.

Este material será cortado e moldado de forma que simbolize uma casa em miniatura, as placas serão coladas através de uma pistola de cola quente.

A Figura 20 demonstra como é uma placa de policarbonato.

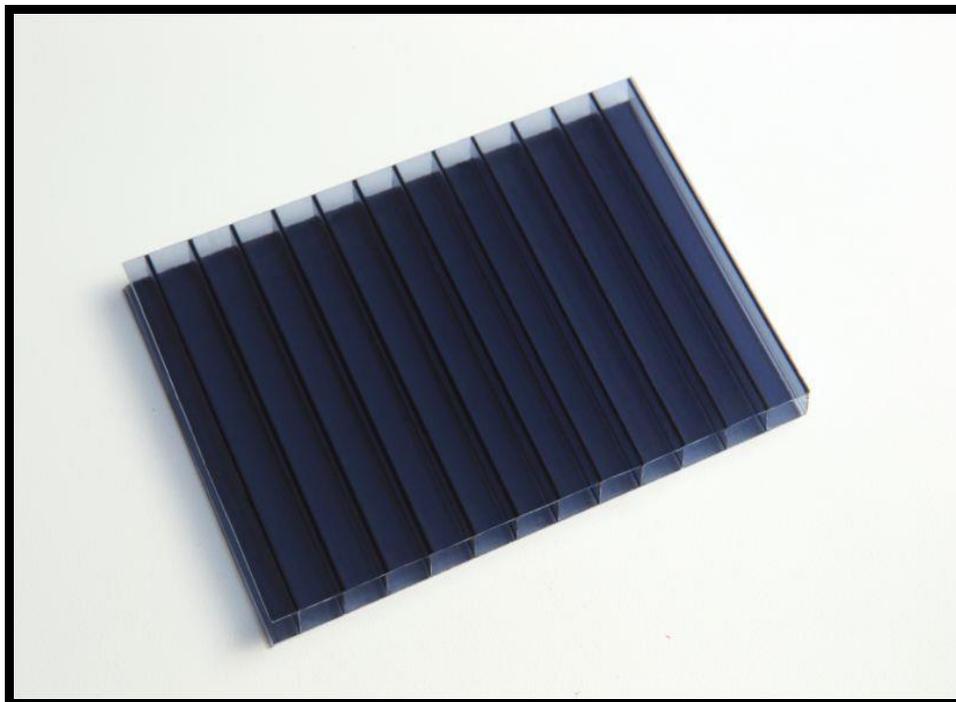


Figura 20 - Placa de Policarbonato

Este material foi escolhido devido ao peso que ele possui, é um material consideravelmente leve e rígido. Com isso tornará a mobilidade da maquete mais fácil e prático.

6.1.2 Dispositivos

O dispositivo para gerenciar todo o ambiente será o *Raspberry Pi* modelo B, na qual o sistema operacional que irá ser instalado é o *Raspbian*. O *Raspberry Pi* será ajustado e configurado para que possa conectar-se a todos os outros dispositivos

Torna-se necessário também um *Protoboard*, uma placa que será responsável por interligar todos os cabos que sairão do *Raspberry Pi* e ligarão ao sensor e aos *LEDs*. Para tal funcionalidade precisa-se também de resistores para que assim possam controlar toda a corrente elétrica que circula entre todos os dispositivos.

6.2 CÂMERA

Recentemente a Fundação *Raspberry Pi* lançou uma câmera própria para o *Raspberry Pi*, a câmera conta com uma lente de 5 *megapixels*, com possibilidade de gravação em HD.

Para tal protótipo, será utilizado a câmera do *Raspberry Pi* como câmera de vigilância. A câmera será conectada a placa do *Raspberry Pi*, onde será instalada e configurada para que assim seja possível a transmissão de vídeo através da internet.

7 ESTUDO DE CASO

Neste capítulo será apresentado a implementação do protótipo de um ambiente inteligente com base na proposta do trabalho. Será informado de como foi realizado cada passo deste trabalho, desde a parte física do até as implementações propostas.

7.1 PARTE FÍSICA DO PROJETO

O protótipo desenvolvido consiste na estrutura de um ambiente inteligente, onde o mesmo pode ser controlado e observado a partir de uma página *Web* ou algum aparelho que possua o sistema operacional *Google Android*.

Para este projeto, alguns dispositivos físicos e operacionais foram necessários. Entre os dispositivos físicos necessários foram: uma placa *Raspberry Pi* Modelo B; um *Case* (Gabinete); uma fonte de alimentação de 5 Volts; um cartão SD de 8 GB; uma Televisão com entrada HDMI ou vídeo composto; um teclado; um mouse; um *notebook*; *LED's*; Resistores; um *Protoboard*; *Jumpers*; um cabo de fita de 26 pinos; uma câmera; placas de policarbonato.

Neste projeto, a placa *Raspberry Pi* foi escolhida por ser um microcomputador, o que possibilita a locomoção de uma maneira mais fácil e ágil, em relação ao modelo da placa, foi escolhido o Modelo B pelo fato de ser uma placa onde possui um desempenho melhor. O *Raspberry Pi* é um computador na qual possui limitações em relação a sistema operacional, a placa aceita apenas sistemas operacionais específicos e desenvolvidos para a placa.

A Figura 21 mostra o *Raspberry Pi* Modelo B utilizado no projeto.



Figura 21 – Raspberry Pi Modelo B.

O Case (Gabinete) escolhido é feito de um material de plástico ABS, é um plástico mais resistente, pois, desta forma garante uma proteção melhor a placa do *Raspberry Pi*, evitando qualquer tipo de curto circuito ou até mesmo a quebra da placa pelo fato dela ser bastante sensível.

A Figura 22 mostra como é a Case escolhida.



Figura 22 - Case.

O microcomputador (*Raspberry Pi*), necessita apenas de 5 Volts para ligar, para tanto, foi necessária uma fonte de alimentação USB, onde possui uma entrada *Bivolt* (100 – 240 VAC) e uma saída de 5 Volts, para que assim, o *Raspberry Pi* pudesse ficar ligado e funcionando perfeitamente.

A Figura 23 mostra a fonte de alimentação utilizada.



Figura 23 - Fonte de Alimentação 5 Volts.

Uma fonte de alimentação é um aparelho ou dispositivo eletrônico constituído por quatro blocos de componentes elétricos: um transformador de força; um circuito retificador; um filtro capacitivo e indutivo; e um regulador de tensão. Fontes de alimentação é um aparelho muito utilizado na área de eletrônica digital.

O cartão de memória SD é utilizado para o armazenamento de arquivos. Como o *Raspberry Pi* não possui nenhum tipo de disco rígido para o armazenamento de arquivos e sistema operacionais, o cartão de memória SD faz o papel de um disco rígido. Devido ao tamanho do sistema operacional e ao trabalho proposto, foi necessário um cartão de memória SD de 8 GB, para que este fosse capaz de armazenar todos os serviços utilizados.

A televisão utilizada foi para que servisse de monitor para o *Raspberry Pi*, de início foi utilizada para a instalação do sistema operacional e alteração e configuração de alguns dispositivos. Como qualquer outro computador, foi necessário um mouse e um teclado para realizar as configurações adequadas.

Após feito todas as configurações no *Raspberry Pi*, foi utilizado o *notebook* para acessar o dispositivo remotamente e também para realizar as implementações dos aplicativos utilizados no *Raspberry Pi*.

Os *LED's* foram utilizados para simular uma lâmpada qualquer, para que isto fosse possível, foi necessário também alguns resistores, alguns *Jumpers*, um *Protoboard* e um cabo de fita para fazerem a conexão com a saída GPIO do *Raspberry Pi*. A Figura 24 mostra como é o cabo de fita utilizado para conectar junto a saída GPIO do *Raspberry Pi*.

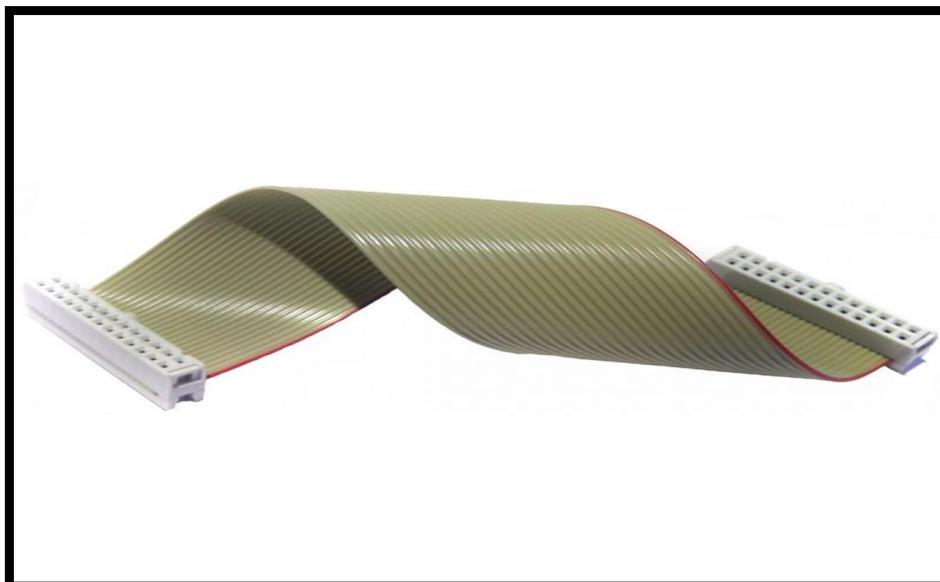


Figura 24 - Cabo de Fita de 26 pinos.

Após a conexão do cabo de fita sobre a saída GPIO do *Raspberry Pi* como mostra a Figura 25.

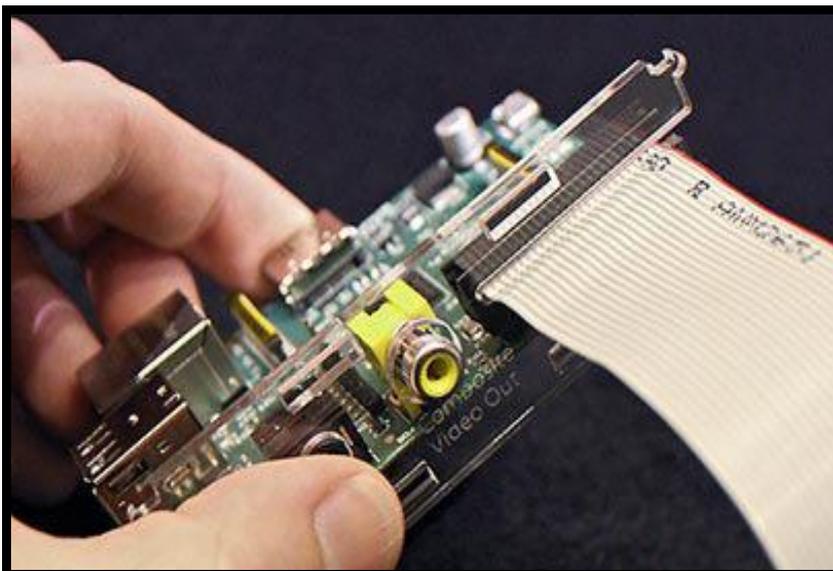


Figura 25 - Cabo de fita conectado ao Raspberry Pi.

Foi conectado os *Jumpers* na saída do cabo de fita e no *Protoboard*, para que assim, os *Jumpers* não fossem ligados diretamente aos resistores e fosse possível utilizar o mesmo pino de energia para todos os *LED's*. Como o *Protoboard* foi utilizado apenas para ligar os *LED's*, foi escolhido para realizar a conexão dos *Jumpers*, um *Protoboard* de 170 furos como mostra a Figura 26.



Figura 26 – Protoboard 170 furos.

A câmera utilizada para realizar o monitoramento do ambiente foi a própria câmera do *Raspberry Pi*, uma câmera capaz de filmar e tirar fotos em qualidade de 1080p em HD.

A câmera do *Raspberry Pi* possui um cabo flexível e deve ser colocada em um conector situado entre a porta *Ethernet* e a entrada de vídeo HDMI, com a parte prateada do cabo flexível voltado para o lado da porta HDMI deve ser encaixado junto ao conector como mostra na figura 27.



Figura 27 - Câmera do Raspberry Pi

As placas de policarbonatos foram utilizadas para a construção da maquete, com base no desenho de uma casa de quatro cômodos, as placas foram cortadas e modeladas no formato de paredes da casa. Junto a placa de policarbonato foi utilizado cola quente para a fixação das paredes da maquete, desta forma, a maquete ficou mais leve e fácil para a locomoção.

A Figura 28 mostra como ficou a maquete utilizada como protótipo da casa.

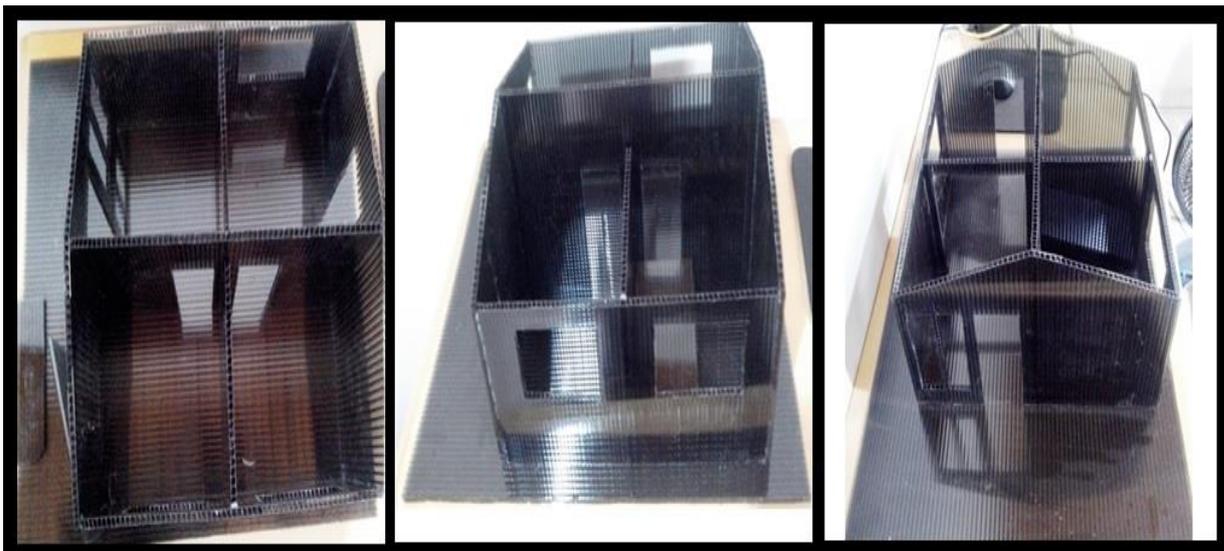


Figura 28 – Maquete.

7.2 PARTE OPERACIONAL

O *Raspberry Pi* nada mais é que um computador como qualquer outro, para tanto foi trivial a instalação de um sistema operacional, porém, devido aos *hardwares* que compõem a placa do *Raspberry Pi*, existe os sistemas operacionais devidamente desenvolvidos para o mesmo. Desta forma, foi instalado o sistema operacional *Raspbian*.

O *Raspbian* é um sistema operacional baseado no *Debian*, uma versão na qual é mais difundida e a recomendada pelo *Raspberry Pi*.

Junto ao sistema operacional *Raspbian* foi instalado e configurado diversos aplicativos, dentre eles, um aplicativo disponibilizado e criado pela comunidade do *Raspberry Pi* para a realização de *streaming* de foto e vídeo da câmera do *Raspberry Pi*. O nome do aplicativo é conhecido como “*RPi Cam Control*”, é um aplicativo que continua em desenvolvimento, ele utiliza como linguagem de programação o *PHP*.

A Figura 29 mostra a interface do aplicativo instalado para a realização do *streaming* de vídeo e foto.

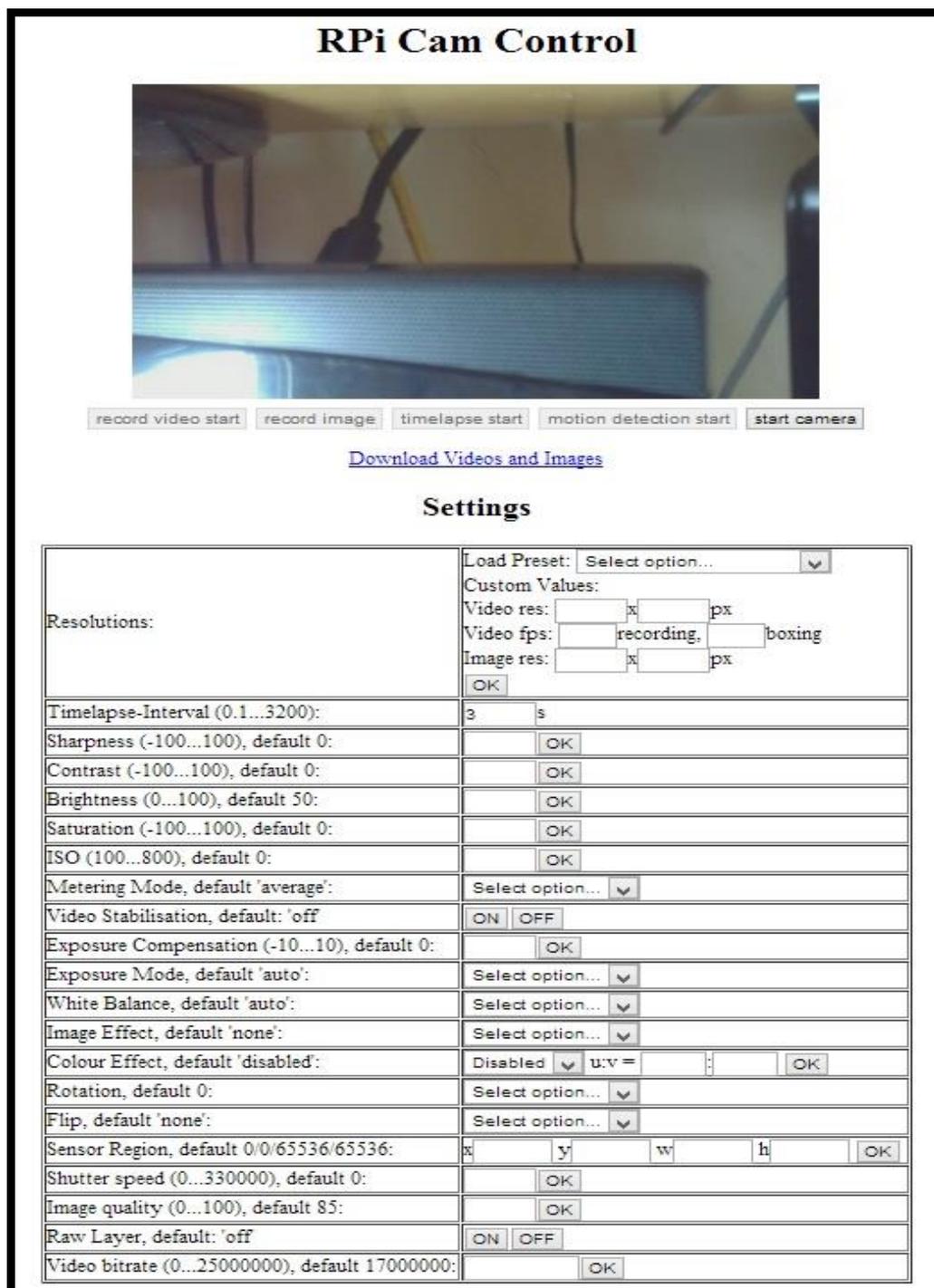


Figura 29 - Interface do RPi Cam Control.

Nesta interface é possível realizar vários procedimentos com a câmera, como por exemplo, alterar a resolução da câmera, diminuir e aumentar o brilho, tirar foto, girar a posição da câmera, tirar foto, gravar um vídeo, entre outros tipos de configurações e ações que podem ser tomadas.

Outro *software* instalado foi o *Apache Tomcat 7*. O *Apache Tomcat* é um *software* de aplicação livre, na qual tem a capacidade de atuar como um *Web Service*. Como servidor *Web*, ele provê um servidor *web HTTP* puramente em *Java*. O servidor inclui ferramentas para configuração e gerenciamento, o que também pode ser feito editando-se manualmente arquivos de configurações formatados em *XML*.

Devido ao servidor de aplicação e aos diversos aplicativos e *softwares* instalados no *Raspberry Pi*, o desempenho do mesmo diminuiu, deixando com que os aplicativos ficassem mais lento. Para melhorar o desempenho da placa, foi feito um *overclock* no *Raspberry Pi*. O *overclock* é o nome que se dá ao processo de forçar os componentes do computador a rodar em uma frequência, definida em *Hertz* mais alta do que a especificada pelo fabricante. O *overclock* está diretamente relacionado ao aumento de consumo de energia, sendo necessário ter a certeza de que a fonte de alimentação consegue debitar a potência necessária ao funcionamento do sistema em *overclock*.

No entanto no *Raspberry Pi* podemos fazer *overclock* ao processador, sem correr nenhum tipo de riscos e sem perder a garantia. O *Raspberry Pi* vem de fábrica com um processador de *700 Mhz*, no entanto é possível levar o mesmo, em segurança, até *1 Ghz*. Neste projeto, foi feito um *overclock* na placa, onde elevou o processador a *900 Mhz*.

A Figura 30 mostra a pasta onde é feito a configuração para a realização do *overclock* do *Raspberry Pi*.

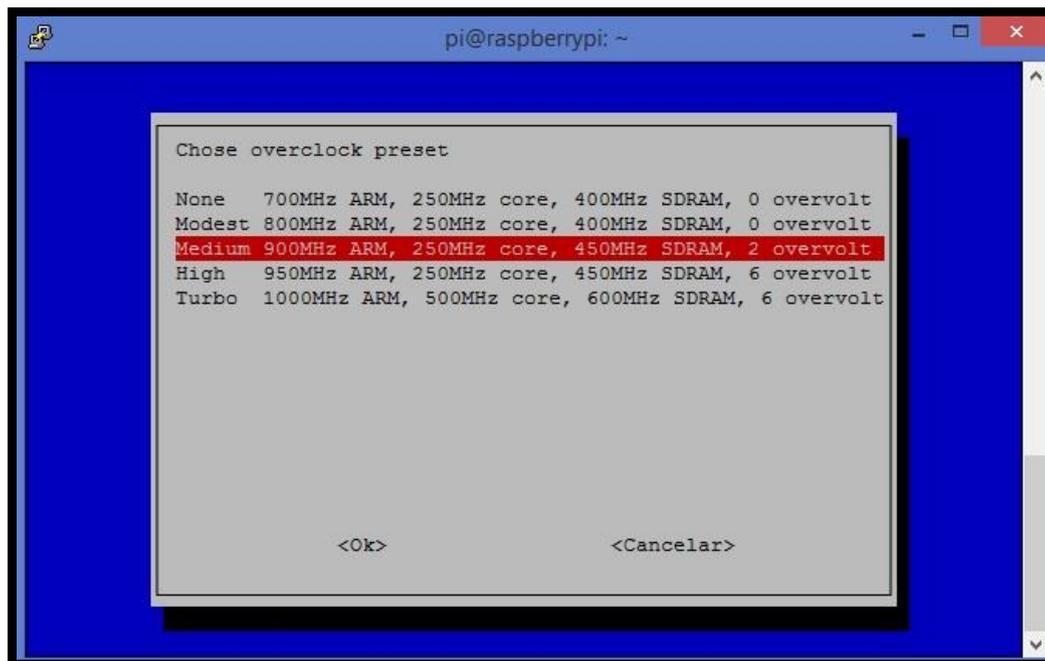


Figura 30 – Overclock.

7.3 DESENVOLVIDO

Para o controle dos *LEDs*, foi desenvolvido outra aplicação *Web*, porém utilizando *JSF*. Para o desenvolvimento da aplicação, foi utilizado uma biblioteca *Java* que ainda está em desenvolvimento, cujo o nome é *Pi4j*.

A princípio foi criado uma página *JSF* (*Java Server Faces*) com os botões de ligar e desligar os *LEDs*, e uma classe *Bean* para fazer o controle dos *LEDs*.

A Figura 31, mostra um trecho do código *JSF* referente a implementação dos botões.

```

<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:p="http://primefaces.org/ui"
      xmlns:f="http://xmlns.jcp.org/jsf/core">
  <h:head>
    <title>TCC</title>
  </h:head>
  <h:body>
    <center>
      <h1>Trabalho de Conclusão de Curso</h1>
    </center>
    <h:form>
      <h:commandButton action="#{principalBean.ligar()}" value="Ligar" />
      <h:commandButton action="#{principalBean.desligar()}" value="Desligar"/><br />
    </h:form>
  </h:body>
</html>

```

Figura 31 - Trecho do código JSF.

A Figura 31, mostra a criação de dois botões, onde ao clicar sobre o mesmo, ele cria uma ação de chamar o método ligar e desligar da classe principal (principalBean).

A Figura 32, mostra o trecho de código dos métodos ligar e desligar da classe *Bean* do projeto.

```

public void desligar() {
    final GpioController gpiocontrole = GpioFactory.getInstance();
    final GpioPinDigitalOutput pin = gpiocontrole.provisionDigitalOutputPin(
        RaspiPin.GPIO_00, "MeuLED", PinState.getInverseState(PinState.LOW));
    pin.low();
    gpiocontrole.shutdown();
    gpiocontrole.unprovisionPin(pin);
}

public void ligar() {
    final GpioController gpiocontroleliga = GpioFactory.getInstance();
    final GpioPinDigitalOutput pin = gpiocontroleliga.provisionDigitalOutputPin(
        RaspiPin.GPIO_00, "MeuLED", PinState.getInverseState(PinState.HIGH));
    pin.high();
    gpiocontroleliga.shutdown();
    gpiocontroleliga.unprovisionPin(pin);
}

```

Figura 32 - Método Ligar e Desligar da classe principal.

Neste código, o primeiro passo para controlar os *LEDs* foi a aquisição de uma instância do *Pi4J* utilizando a classe *GpioController*, essa classe envia os sinais dos pinos *GPIO* para o *Raspberry Pi*. Após a criar o controlador, é iniciado o pino desejado do *Raspberry Pi*. Na Figura 32, o pino que é iniciado, é o pino zero da placa. Depois de iniciar o pino desejado, definimos o estado dele, se é desligado ou ligado. O método *Shutdown*, serve para forçar a parada do controlador, liberando o controlador criado. Já o método *UnprovisionPin* é para finalizar o processo do pino que está iniciado.

Para desenvolver o aplicativo *mobile* foi utilizado a plataforma *Eclipse*. No aplicativo *mobile* utilizamos o *WebView*. Método na qual faz com que dentro do aplicativo é chamado qualquer página *Web*. Desta forma, foi utilizado para chamar a página *JSF* criada, e assim podemos controlar a aplicação também pelo celular, utilizando o aplicativo instalado.

A Figura 33 mostra a interface criada para o aplicativo *mobile*.



Trabalho de Conclusão de Curso



Figura 33 - Interface Mobile.

Após concluir todas as implementações, foi testado e validado todas as funcionalidades, juntamente com as partes físicas do *Raspberry Pi* e os dispositivos ligados a ele.

No entanto, o trabalho apresentou resultados condizentes com os esperados e proposto no capítulo 6.

8 CONCLUSÃO

A realização desse projeto, possibilitou a utilização de conceitos, ferramentas e técnicas aprendidas durante todo o curso de Ciência da Computação. Outras matérias estudadas durante toda a pesquisa do projeto, foi fundamental para realização de todas as implementações.

A utilização do *Raspberry Pi* para a prototipagem de um ambiente inteligente, confirmou-se como uma escolha correta, pois, considerando todas as características que ele oferece, constitui uma plataforma economicamente viável para quem gosta de sistemas embarcados e possui um alto poder computacional para todo o gerenciamento dos *softwares* e aplicativos instalados nele. Com isto, o objetivo principal do projeto, que era de modelar e construir um protótipo de ambiente inteligente com baixo custo, foi atingido.

Apesar da linguagem *Java* não ser a mais recomendada para ser utilizada no *Raspberry Pi*, pelo fato da linguagem não se preocupar com o uso de memória, o servidor não apresentou problemas de performance e seu desempenho foi suficiente para a implementação dos sistemas propostos.

Em geral, tanto a parte do servidor, quanto a parte cliente, apresentaram o funcionamento de todas as funções implementadas. Desta forma, o projeto de ambiente inteligente foi condizente com os objetivos proposto para o trabalho.

8.1 TRABALHOS FUTUROS

Trabalhos futuros para este projeto podem ser realizados expandindo o acesso ao servidor e a inserção de novas funções ao *Raspberry Pi*.

Para as novas funções, pode-se realizar a implementação do serviço de *streaming* de vídeo e imagem, a implementação de novos sensores e atuadores utilizando a linguagem *Java*.

Em relação ao servidor, é possível também transformá-lo em uma estação de multimídia, já que o *Raspberry Pi* é capaz de reproduzir vídeos em alta resolução. Outra opção é a inclusão de métodos de monitoramento, possibilitando ao usuário verificar algumas informações do servidor como: latência de comunicação, quantidade de clientes conectados e temperatura interna da placa.

Enfim, através deste projeto, pode-se realizar e acrescentar inúmeros conceitos e dispositivos, pois a placa do *Raspberry Pi* possibilita a integração com muitas outras tecnologias.

REFERÊNCIAS

ALBINADER NETO, Jorge Abilio; LINS, Rafael Dueire. **Web Services em Java**, 1. ed. Rio de Janeiro: Brasport Livros e Multimídia Ltda, 2006.

ALVES, José Augusto; MOTA, José. **Casas Inteligentes**, 1. ed. Lisboa: Centro Atlântico, 2003.

AMCHAM. **Mercado de tablets e smartphones começa a crescer no Brasil**. Disponível em: <<http://www.amcham.com.br/business-in-growth/noticias/evolucao-do-mercado-de-tablets-e-smartphones-e-lenta-no-brasil>>. Acesso em: Novembro de 2013.

AQUINO, Juliana França Santos. **Plataforma de Desenvolvimento para Dispositivos Móveis**. 2007. 14p. Monografia – Programa de Pós Graduação em Informática – Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2007.

AURESIDE. **Brasileiros têm grande interesse em Automação Residencial, diz estudo**. Disponível em: <http://www.aureside.org.br/noticias_recentes/default.asp?file=01.asp&id=332>. Acesso em: Outubro de 2013.

AURESIDE. **Casa inteligente chega à classe média**. Disponível em: <http://www.aureside.org.br/imprensa/default.asp?file=folha_130311.htm>. Acesso em: Setembro de 2013.

AURESIDE (Brasil). **MISSÃO**. Disponível em: <<http://www.aureside.org.br/quemsomos/default.asp?file=missao.asp&menu=quemsomos>>. Acesso em: Fevereiro de 2014.

BOLZANI, Caio Augustus Morais. **Residências Inteligentes**, 1. ed. São Paulo: Editora Livraria da Física, 2004.

CARISSIMI, Alexandre da Silva; ROCHOL, Juergen; GRANVILLE, Lisandro Zambenedetti. **Redes de Computadores**, 20. ed. Porto Alegre: Bookman, 2009

COSTA, Daniel Gouveia. **Java em Rede – Recursos Avançados de Programação**, 1. ed. Rio de Janeiro: Brasport, 2008.

COULOURIS, George; DOLLIMORE, Jean; KINDBERG, Tim; BLAIR, Gordon. **Sistemas Distribuídos Conceitos e Projetos**, 5. ed. Porto Alegre: Bookman, 2013.

DIAS, César Luiz de Azevedo; PIZZOLATO, Nélio Domingues. **Domótica**. 2004. 32p. Artigo – Aplicabilidade e Sistemas de Automação Residencial – CEFET. Campos dos Goytacazes, Rio de Janeiro, 2004.

FARIA, Alessandro de Oliveira. **Programa seu Andróide**. Linux Magazine, Volume 1, Número 43, p. 73-77, 2008.

FRAGA, Renê. **Queremos que as pessoas inovem juntas**. Campus Party. Disponível em: <<http://www.techtudo.com.br/noticias/noticia/2013/02/queremos-que-pessoas-inovem-juntas-diz-fundador-do-raspberry-pi.html>>. Acesso em: Agosto de 2013.

LECHETA, Ricardo R. **Google Android Aprenda a criar aplicações para dispositivos móveis com o Android SDK**. 3. ed. São Paulo: Novatec Editora Ltda, 2013.

LOPES, Carlos J. Feijó; RAMALHO, José Carlos. Web Services: Metodologia de Desenvolvimento. In: DEPARTAMENTO DE SISTEMA DE INFORMAÇÃO UNIVERSIDADE DO MINHO, 1, 2004, Porto, Portugal. **XML: Aplicações e Tecnologias Associadas**, v.1, fevereiro, 2004. 15p.

MESSIAS, Alan Fernandes. **Edifícios “Inteligentes”: A Domótica Aplicada à Realidade Brasileira**. 2007. 48p. Monografia - Engenharia de Controle e Automação – Universidade Federal de Ouro Preto, Minas Gerais, Ouro Preto, 2007.

MONK, Simon. **Programando o Raspberry Pi Primeiros Passos com Python**, 1. ed. São Paulo: Novatec Editora Ltda, 2013.

MOURÃO, Leonardo. **Automação Residencial: sem fio, controlada por celular e mais barata**. Disponível em: <<http://casa.abril.com.br/materia/automacao-residencial-sem-fio-e-barata>>. Acesso em: Setembro de 2013.

NORVING, Peter; RUSSELL Stuart. **Artificial inteligente: A modern approach**. 3. ed. New Jersey: Prentice Hall, 2002.

PEREIRA, Lúcio Camilo Oliva; SILVA, Michel Lourenço. **Android para Desenvolvedores**, 1.ed. Rio de Janeiro: Brasport, 2009.

QUINDERÉ, Patrick Romero rota. **Casa Inteligente - Um Protótipo de Sistema de Automação Residencial de Baixo Custo**. 2009. 69p. Monografia - Graduação em Ciência da Computação - Faculdade Faria Brito, Fortaleza, 2009.

RIBEIRO, Alexandre Moretto; COSTA, Celso Maciel; MENEGUETTI, Edgar; GRANVILLE, Lisandro Zambenedetti. In: FÓRUM INTERNACIONAL SOFTWARE LIVRE, 5, 2004, Porto Alegre, Brasil. **Workshop sobre Software Livre**, v.5, Junho, 2004, 214p.

RICHARDSON, Matt; WALLACE, Shawn. **Primeiros Passos com o Raspberry Pi**, 1. ed. São Paulo: Novatec Editora Ltda, 2013.

SILVA, Luciano Alves. **Aprenda Passo a Passo a Programar em Android: Guia Essencial para Desenvolvedores**. 2. ed. Rio de Janeiro: AGBOOK, 2012.

SJOGELID, Stefan. **Raspberry Pi for Secret Agents**, 1. ed. Birmingham: Editora Packt Publishing, 2013.

UPTON, Eben; HALFACREE, Gareth. **Raspberry Pi Manual do Usuário**, 1. ed. São Paulo: Novatec Editora Ltda, 2013.

TECPAR. **Mercado de automação residencial dá sinais de crescimento**. Disponível em: <<http://portal.tecpar.br/index.php/pt/noticias/1781-mercado-de-automacao-residencial-da-sinais-de-crescimento>>. Acesso em: Fevereiro de 2014.

TEZA, Vanderlei Rabelo. **Alguns aspectos sobre a automação residencial – Domótica**. 2002. 107p. Dissertação (Mestrado) – Programa de Pós Graduação em Ciência da Computação – Universidade Federal de Santa Catarina, Santa Catarina, Florianópolis, 2002.

WONG, Clinton. **HTTP Pocket Reference**, 1. ed. Sebastopol: O'Reilly e Associates, 2000.

WORTMEYER, Charles; FREITAS Fernando; CARDOSO Líuam. Automação Residencial: Busca de Tecnologias visando o Conforto, a Economia, a Praticidade e a Segurança do Usuário. In: SIMPÓSIO DE EXCELÊNCIA EM GESTÃO E TECNOLOGIA, 2, 2005. Resende, Brasil. **Resumos**. Resende: Associação Educacional Dom Bosco, 2005.