

FUNDAÇÃO MUNICIPAL DE ENSINO DE PIRACICABA
ESCOLA DE ENGENHARIA PIRACICABA

DAVI ZONETE

Controle de dispositivos eletrônicos em Domótica via *smartphones*.

Piracicaba

2013

DAVI ZONETE

Controle de dispositivos eletrônicos em Domótica via *smartphones*.

Trabalho de Conclusão de Curso
apresentado à Escola de Engenharia de
Piracicaba como parte dos requisitos
exigidos para a graduação no curso de
Ciência da Computação.

Orientador:
Prof. Dr. José Martins Junior

Piracicaba

2013

Davi Zonete

Controle de dispositivos eletrônicos em Domótica via smartphones

Trabalho de Conclusão de Curso apresentado à Escola de Engenharia de Piracicaba como parte dos requisitos exigidos para a graduação no curso de Ciência da Computação.

Aprovado em: ____/____/____

Banca Examinadora

Membro: (Titulação e nome)_____

Instituição: _____ Assinatura: _____

Membro: (Titulação e nome)_____

Instituição: _____ Assinatura: _____

Membro: (Titulação e nome)_____

Instituição: _____ Assinatura: _____

“Dedico este trabalho a todas as pessoas que fizeram
parte dessa caminhada.”

AGRADECIMENTOS

Agradeço primeiramente ao Dr. José Martins Junior, orientador deste trabalho, pelo apoio e direcionamento das atividades, e principalmente pela ajuda nos momentos mais difíceis.

A todos familiares por todo apoio motivacional para a realização desse trabalho e por todo caminho trilhado durante o curso.

À minha amada Geovana por todo apoio, dicas e por toda ajuda prestada sobre esse trabalho.

A todos os amigos da Escola de Engenharia de Piracicaba por todo apoio e incentivo e encorajamento para estar realizando o trabalho da melhor forma possível.

“A verdadeira viagem de descobrimento não consiste em
procurar novas paisagens, mas em ter novos olhos.”
(Marcel Proust)

RESUMO

ZONETE, D. **Controle de dispositivos eletrônicos em Domótica via smartphones**. 2013. 49 f. Trabalho de Conclusão de Curso – Escola de Engenharia de Piracicaba, Piracicaba, 2013.

A comunicação entre aparelhos eletrônicos tem evoluído muito nos últimos anos. Em épocas passadas, a comunicação ocorria de forma limitada e exigia a utilização de cabos e de conectores específicos. Hoje em dia, é possível comunicar-se com um aparelho eletrônico através de tecnologias que não necessitam conexão mecânica. Nesse contexto, pensou-se em realizar um trabalho em Domótica que utilize de tecnologia sem fio para realizar o controle de dispositivos eletrônicos em uma residência. O objetivo deste trabalho foi criar um sistema em Domótica que permita controlar via web, através de um smartphone ou computador, o acionamento de lâmpadas e outros dispositivos, utilizando a comunicação ZigBee e uma placa controladora Arduino. Além disso, todas as informações são armazenadas no servidor para a posterior geração de relatórios sobre a rotina de uso dos equipamentos na residência e perfis de consumo de eletricidade. Também foi demonstrado o consumo da utilização do Arduino acoplado ao módulo Xbee.

Palavras-chave: Arduino; Domótica; Android; Xbee; Controle; ZigBee; MySql

ABSTRACT

ZONETE, D. **Controlling electronic devices in home automation using smartphones**. 2013. 49 f. Undergraduate dissertation – Escola de Engenharia de Piracicaba, Piracicaba, 2013.

The communication between electronic devices has evolved so much recently. In the past the communication was limited and required the use of cables and connectors specific. Nowadays, it is possible to communicate with an electronic device using technologies that require no mechanical connection. In this context, the idea was to do a project in Home Automation that uses wireless technology to make the control of electronic devices in the house. The objective was to create a domotics system that would allow control to turn on and off the lamps and others electronics using smartphones or a computer. In the project there is an Arduino that has the responsibility to control all electronics connected in it. All information is stored on the database because after there will be some reports about the use of the devices electronic.

Keywords: Arduino;Domotics;Android;Xbee;Controle;ZigBee;MySql.

LISTA DE ILUSTRAÇÕES

Figura 1 - Placa Controladora.....	15
Figura 2 - Arduino.....	17
Figura 3 - IDE Arduino	18
Figura 4 – Topologias de Redes	19
Figura 5 - Xbee e Xbee-Pro	20
Figura 6 – Alguns tipos de antenas do Xbee.....	21
Figura 7 – Visão Geral da Solução	22
Figura 8 - Modelo Cascata.....	23
Figura 9- Objetivo do Projeto.....	25
Figura 10 - Diagrama Caso de Uso.....	26
Figura 11 - Login Sistema via Internet	27
Figura 12 - Tela Login via Android	27
Figura 13 - Tela Dispositivos via Internet.....	28
Figura 14 - Tela Dispositivos via Android	28
Figura 15 - Diagrama de Atividade.....	29
Figura 16- Diagrama de Classe.....	30
Figura 17 - Indentação Eclipse	31
Figura 18 - DER.....	32
Figura 19 – Definindo variáveis de saída.....	34
Figura 20 – Principal função Arduino	35
Figura 21 – Verificação usuário	35
Figura 22 – Principal função Arduino	36
Figura 23 – Exemplo Acionamento	37
Figura 24 – Conexão ao MySQL.....	37
Figura 25 – Função de Liga()	38
Figura 26 – Envio da informações para o Arduino	39
Figura 27 – Arduino utilizado para testes	40
Gráfico 28 – Durabilidade das baterias.....	43
Figura 29 – Análise Histórico.....	44
Figura 30 – Análise Resumida.....	44
Gráfico 31 – Amostragem sem barreira.....	45
Gráfico 32 – Amostragem com barreira	46

LISTA DE TABELAS

Tabela 1 - Exemplo de Transformação Binária para Tensão(V)	15
Tabela 2- Exemplo de Classificação de Níveis	29
Tabela 3 - Exemplo de Dispositivos/Níveis	30
Tabela 4 – Entrada de Dados Arduino.....	33
Tabela 5 – Lista de Páginas JSP	39
Tabela 6 – Resultados de Segurança.....	42
Tabela 7 – Resultados de durabilidade das baterias	42
Tabela 8 – Distância de comunicação Xbee sem barreira	45
Tabela 9 – Distância de comunicação Xbee com barreira	46

LISTA DE ABREVIATURAS E SIGLAS

USB	Universal Serial Bus
Open-Source	Código Aberto
IDE	<i>Integrated Development Environment</i>

SUMÁRIO

1	INTRODUÇÃO	12
1.1.	APRESENTAÇÃO DO ORIENTADOR	12
1.2.	OBJETIVOS	13
2	DOMÓTICA	14
3	CONTROLADORES	15
4	ARDUINO	17
5	REDES DE DISPOSITIVOS	19
6	XBEE	20
7	SOLUÇÃO ADOTADA	22
7.1.	MODELO DA SOLUÇÃO	23
7.1.1.	ESTUDO DO PROBLEMA	23
7.1.2.	ANÁLISE	24
7.1.3.	PROJETO	25
7.1.4.	CODIFICAÇÃO	31
7.2.	IMPLEMENTAÇÃO	33
7.3.	TESTES	39
8	RESULTADOS	42
9	CONCLUSÕES	48
9.1.	SUGESTÕES PARA TRABALHOS FUTUROS	48
	REFERÊNCIAS	50
	ANEXO A – CERTIFICADO DE PARTICIPAÇÃO NO 21° SIICUSP	51
	ANEXO B – BANNER DO TRABALHO NO 21° SIICUSP	52

1 INTRODUÇÃO

A comunicação entre aparelhos eletrônicos tem evoluído de uma maneira fantástica. Antigamente, a comunicação era feita de forma limitada onde era necessário utilizar cabos e conectores específicos. Hoje em dia, é possível se conectar a um aparelho eletrônico através de tecnologias que não necessitam conexão mecânica. Uma dessas tecnologias é a Wi-Fi, extremamente explorada no mundo da Computação pela sua velocidade e pelo baixo custo.

Outras tecnologias que têm crescido cada dia mais são aquelas relacionadas ao envio de dados via celular. Embora os pacotes de dados enviados por um celular demorem um pouco para serem transmitidos, a vantagem é que não importa a distância que se esteja do servidor de dados. Se o sinal da rede da operadora estiver disponível, é possível ser feita a comunicação entre servidor e celular.

Outra área que tem se expandido é a Robótica aplicada em vários projetos de automação. No contexto da Robótica encontra-se uma área chamada “Domótica”, ou seja Robótica voltada para automação residencial.

Em Galera (2008), foi demonstrada a união das três tecnologias com a proposta de acionar equipamentos eletrônicos através da rede sem fio. A abordagem apresentada pelo autor, consistiu do uso da plataforma Java (J2ME) para o celular e controlador PIC para integração do módulo XBee responsável pela comunicação RF. Galera (2008) não trouxe dados muito detalhados, os dados foram divididos em 3 tipos (Bom, Médio e Ruim).

1.1. APRESENTAÇÃO DO ORIENTADOR

Orientador José Martins Jr. possui graduação em Ciência da Computação pela Escola de Engenharia de Piracicaba (1991), mestrado em Ciências da Computação e Matemática Computacional pela Universidade de São Paulo (2003) e doutorado em Engenharia Mecânica (Mecatrônica) pela Universidade de São Paulo (2010). Atualmente é professor do curso de Sistemas de Informação da Faculdade Salesiana Dom Bosco de Piracicaba e professor titular da Escola de Engenharia de Piracicaba. Tem experiência na área de Ciência da Computação, atuando

principalmente nos seguintes temas: redes de computadores, sistemas distribuídos, desenvolvimento de software, robótica e sistemas de informação.

1.2. OBJETIVOS

O objetivo do TCC foi aprimorar o trabalho apresentado em Galera (2008), utilizando tecnologias mais recentes em busca de resultados que possam trazer um bom desempenho em velocidade de transferência de dados e alcance de sinal. Também é demonstrar o consumo de energia do circuito utilizado.

Portanto, foi utilizada Java para Android, uma tecnologia mais avançada que a J2ME e que agrega opções como, por exemplo, *TouchScreen*. O controlador PIC foi substituído por uma placa Arduino que tem sido muito utilizada na área de Domótica, devido sua praticidade e qualidade.

Outro objetivo é testar e colher resultados do trabalho a respeito dos parâmetros de desempenho. A ideia é trazer valores mais significativos, como alcance de sinal com barreiras (paredes) e sem nenhuma barreira.

2 DOMÓTICA

A Domótica é uma tecnologia que está sendo utilizada cada dia mais nas residências. O termo Domótica é o resultado da junção de duas palavras “Doméstica” e “Robótica”. Analisando essas duas palavras pode-se imaginar que esta tecnologia é desenvolvida com o objetivo de facilitar o cotidiano das pessoas. Outra idéia para se utilizar a Domótica é tornar as tarefas cotidianas mais divertidas, como por exemplo, acionar lâmpadas da casa através de palmas, ou ligar um aparelho eletrônico simplesmente através de um comando de voz (ADAMIA, 2013).

A robótica é muito utilizada na indústria. Alguns benefícios que podem ser citados da robótica nas indústrias é a qualidade do serviço pois através de robôs é possível definir padrões, aumentar a qualidade dos produtos, aumentar a produtividade e, conseqüentemente, aumentar o lucro.

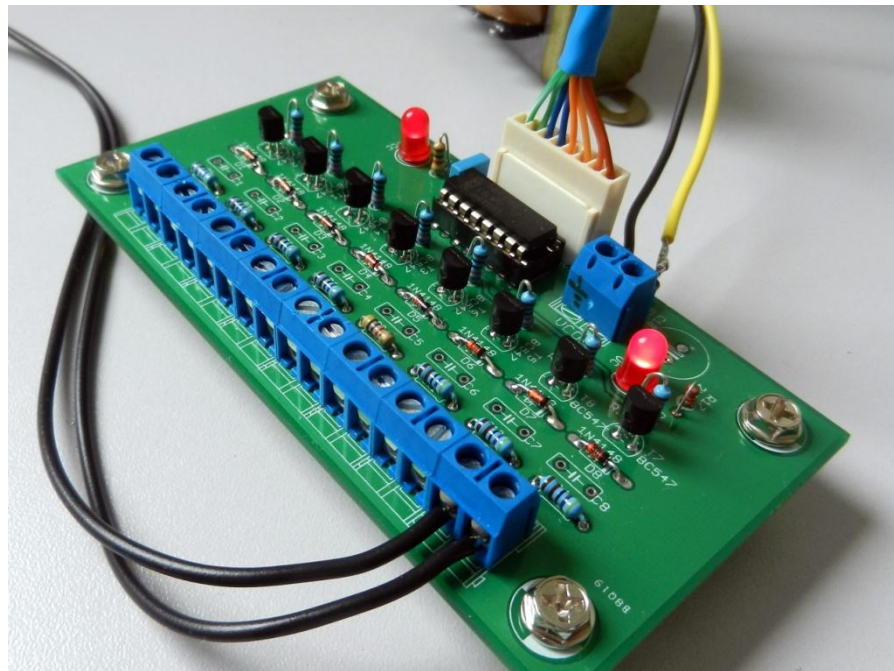
Atualmente as pessoas buscam tecnologias para melhorar seu ambiente e também facilitar tarefas rotineiras. Um grande exemplo que pode ser citado é o controle da televisão. Hoje, este instrumento é tão utilizado que nem se dá conta de como é feita a comunicação do controle para a televisão. Algumas décadas atrás, a única forma de se mudar o canal da televisão ou diminuir o volume dela era com botões acoplados nela. Quando não se tinha um controle remoto, mudar de canal era uma tarefa chata de ser executada, pois nem sempre a televisão estava tão próxima. Como a idéia da Domótica é facilitar tarefas rotineiras, em 1955 Eugene Polley teve a idéia de desenvolver um dispositivo que realizasse essa ação de mudar de canais sem precisar estar próximo da televisão. Foi então criado o primeiro controle remoto.

A perspectiva é que cada dia mais a área da robótica seja explorada e torne a vida das pessoas mais fácil.

3 CONTROLADORES

As placas controladoras (Figura 1) são componentes muito utilizados na área de tecnologia, pois através delas pode-se controlar um dispositivo eletrônico. Essas placas geralmente recebem sinais que podem ser enviados de outra placa controladora ou botões (SCHUTZER, 2013).

Figura 1 - Placa Controladora



Fonte: tobiasmugge.wordpress.com

O funcionamento dessas placas é através de sinais digitais (0 ou 1), mas na eletrônica não é possível passar valor 1 e 0. Então é definida uma tensão para o valor 1 e outra tensão para o valor 0, conforme a tabela 1.

Tabela 1 - Exemplo de Transformação Binária para Tensão(V)

Binário	Tensão(V)
0	0 até 3 V
1	3 até 12 V

Fonte: Elaborada pelo autor

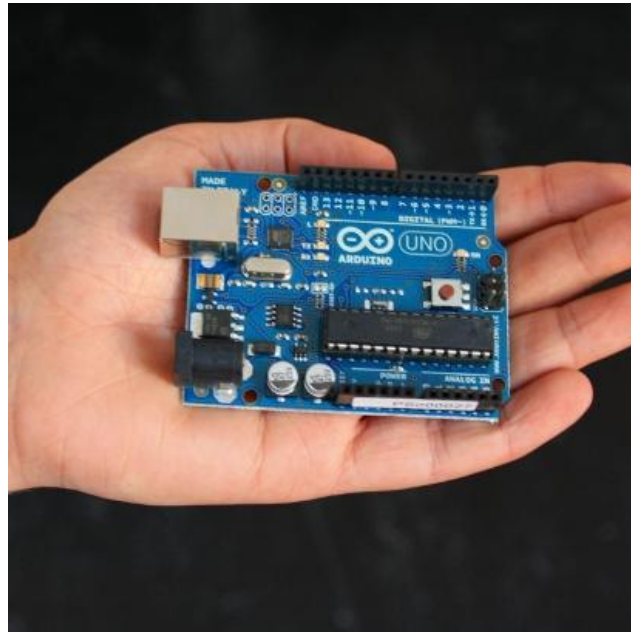
Através desses valores é possível definir ações dentro da placa controladora.

Hoje, no mercado, existem vários tipos de placas controladoras. Alguns atributos que podem variar entre uma placa e outra são a quantidade de saída para dispositivos e a forma de comunicação.

4 ARDUINO

O Arduino (Figura 2) é um componente eletrônico que tem a finalidade de facilitar a comunicação entre computador e a placa controladora. O Arduino possui uma placa controladora e uma entrada de dados via cabo USB.

Figura 2 - Arduino



Fonte: Site oficial do Arduino (<http://www.arduino.cc>)

O Arduino pode ser utilizado por pessoas que não tenham muito conhecimento em eletrônica, pois sua estrutura é simples e de fácil entendimento.

Arduino é uma plataforma de prototipagem eletrônica de código aberto, composta de hardware e software, baseada na flexibilidade e simplicidade de uso. Ele é destinado para artistas, designers, *hobbyist*, e qualquer outra pessoa interessada em criar interatividade com objetos ou ambiente (FRANCESCHI, 2012).

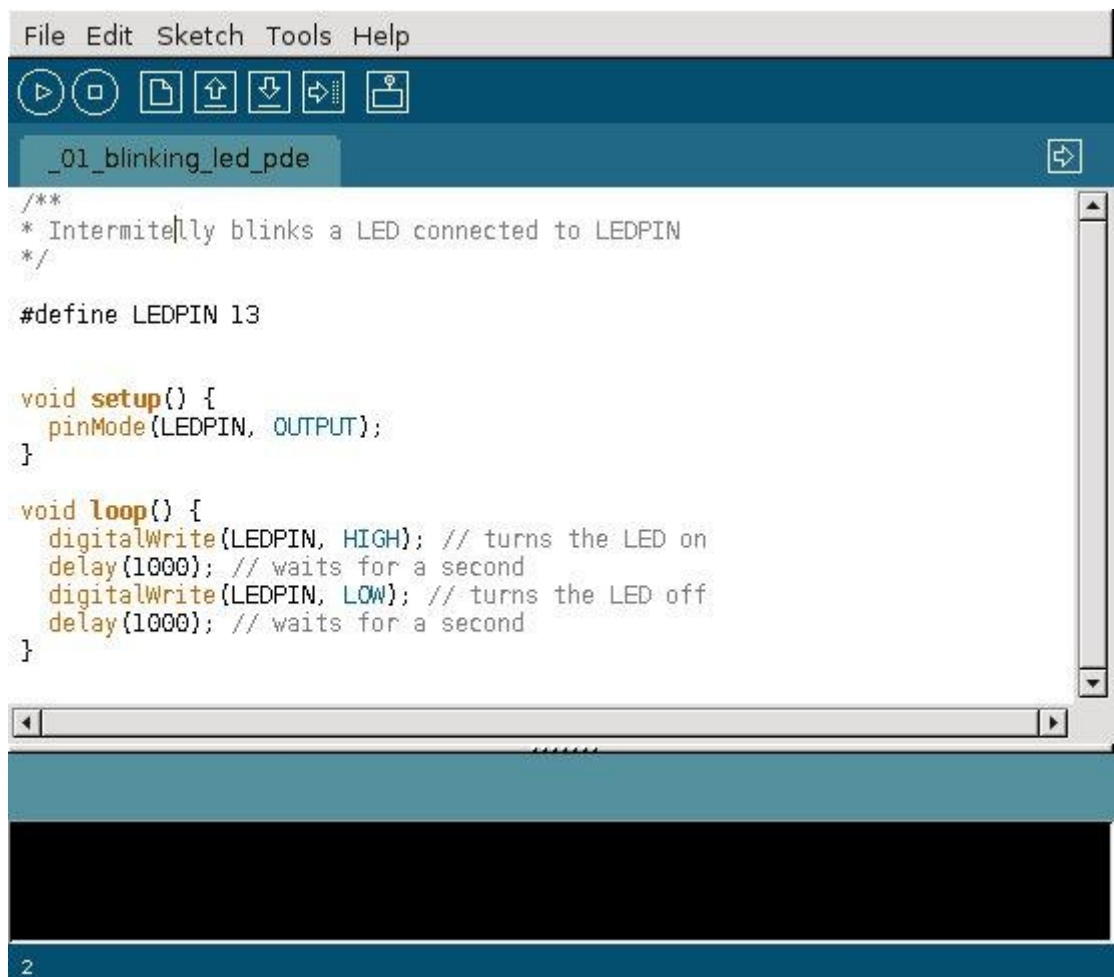
Arduino pode receber sinais de entrada de vários sensores colocados no ambiente e, através deles, controlar luzes, motores e outros atuadores. O micro controlador contido na placa é programado usando linguagem de programação Arduino (similar a C) e o ambiente de desenvolvimento baseado em processamento. Os projetos feitos no Arduino podem ser independentes ou comunicados com outro sistema existente.

Existem vários modelos de Arduino, alguns dos modelos mais famosos são Uno, Dúmílanove e Mega. As diferenças entre eles são de velocidade de processamento e também a quantidade de portas de saída.

O padrão de comunicação do Arduino é através de cabo USB, mas através de alguns acessórios é possível fazer a comunicação via ZigBee, Wi-Fi, Bluetooth, Ethernet, entre outros padrões de comunicação.

O Arduino é comercializado mundialmente e possui um grande acervo de informações e tutoriais para o seu uso. Seu preço pode variar de um modelo para outro. É possível encontrar o modelo mais básico a partir de R\$ 80,00 no Brasil.

Figura 3 - IDE Arduino



Fonte: Google Imagens

A linguagem utilizada para desenvolver programas para o Arduino é a C/C++. Através do seu site oficial (<http://www.arduino.cc>) é possível fazer o download gratuitamente da IDE (Figura 3) de desenvolvimento dos programas que serão executados nele.

5 REDES DE DISPOSITIVOS

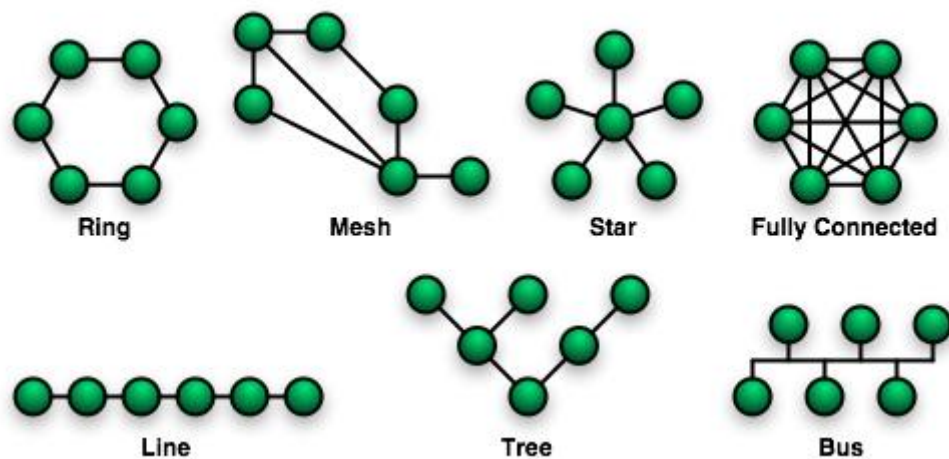
A comunicação entre dispositivos é um assunto muito abrangente. Existem vários tipos de protocolos que permitem essa comunicação entre um dispositivo e outro. Alguns dos tipos de comunicação mais conhecidos são Wi-Fi, Bluetooth, cabo serial, USB e ZigBee (MESSIAS, 2012).

Cada tipo de protocolo de comunicação tem uma aplicação diferente. Fazer a comunicação entre dois dispositivos que estejam a uma distância de 1000 metros, caso fosse possível utilizar a comunicação via cabo USB, talvez o preço ficaria um pouco alto. O mais adequado seria fazer uma comunicação Wi-Fi, podendo assim economizar gastos com fios. Por outro lado, a comunicação através de cabos USB é ótima para ligação de dispositivos com distâncias curtas.

Existem várias topologias de redes, as mais comuns são a Ponto-a-Ponto, Malha, Anel ou Circulo e estrela. As topologias de redes também são aplicadas quando se utiliza a tecnologia ZigBee.

A Figura 4 mostra algumas das topologias mais utilizadas atualmente.

Figura 4 – Topologias de Redes



Fonte: Google Imagens

6 XBEE

O Xbee (Figura 5) é um componente eletrônico que permite fazer a comunicação entre dois dispositivos através do padrão ZigBee IEEE 802.15.4. O Xbee é fabricado pela empresa MaxStream, e existem duas versões, Xbee e Xbee-Pro. A diferença entre elas é o alcance de comunicação.

Figura 5 - Xbee e Xbee-Pro



Fonte: Web Store oficial do Xbee.(www.xbeestore.com.br)

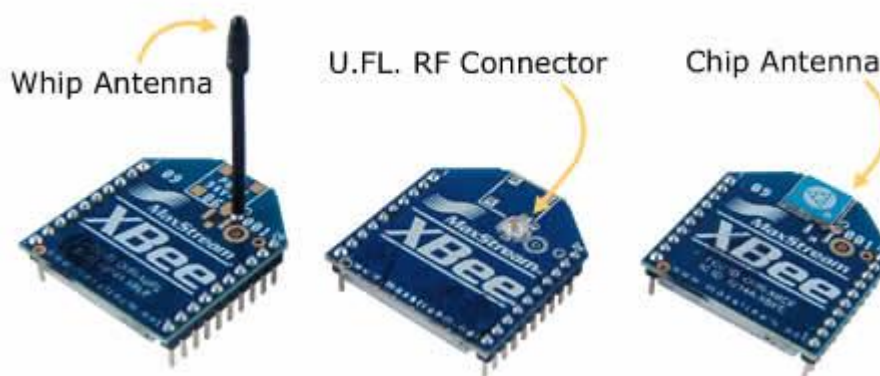
O padrão IEEE 802.15.4 proporciona um baixo consumo de energia e um custo reduzido. O ZigBee pode chegar a uma taxa de transferência de 250kbps, mas esse valor pode variar devido à distância entre dispositivos e também à qualidade de componentes utilizados pelo fabricante nas placas que realizam a comunicação.

Como o Xbee utiliza o padrão ZigBee, é possível fazer a comunicação entre Xbee e outra placa que utilize o padrão ZigBee, produzida por outro fabricante.

É possível utilizar o Xbee de forma “Ponto a Ponto”. Neste caso, só é feita a comunicação entre dois módulos Xbee. Ou utiliza-se de forma estrela, pela qual é possível comunicar vários Xbee.

Existem vários tipos de módulos Xbee, que diferem quanto ao tipo da antena acoplada. Existem Xbee com antena interna, antena externa e também com encaixe para antenas que proporcionem maior alcance (MESSIAS, 2012).

Figura 6 – Alguns tipos de antenas do Xbee



Fonte: <http://www.rogercom.com/ZigBee/ZigBee.htm>

É possível comprar os módulos Xbee através da *web store* (www.xbeestore.com.br), com valor a partir de R\$ 70,00.

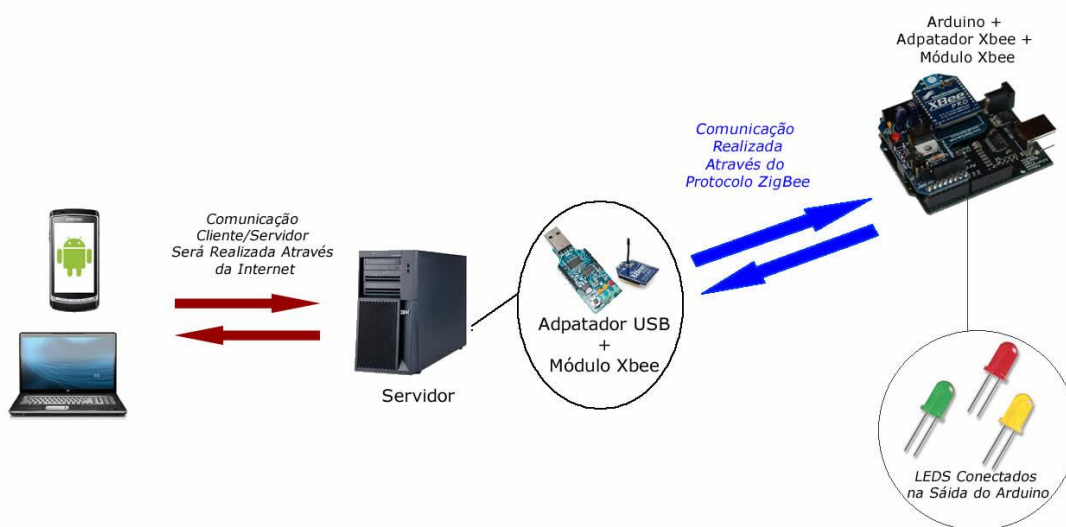
7 SOLUÇÃO ADOTADA

Para a solução do problema vislumbrou-se desenvolver um serviço Web que recebe informações como “acionar um dispositivo eletrônico”, e repassa essa informação para a placa controladora Arduino.

A placa controladora Arduino pode ser adicionada através de cabo USB ou via redes sem fio, mas para a segunda opção funcionar são necessárias duas placas Xbee. A primeira placa Xbee é conectada ao computador e pode ser definida como “Transmissora” e a outra placa pode-se chamar “Cliente”, e é conectada à placa Arduino, que recebe os comandos da placa “Transmissora”.

A placa Xbee é muito utilizada devido seu baixo consumo de energia elétrica e também por conseguir resultados considerados bons em relação à distância e à confiabilidade.

Figura 7 – Visão Geral da Solução



Fonte: Elaborada pelo Autor

A figura 7 apresenta uma visão geral da solução proposta.

O primeiro passo é fazer o serviço Web receber todas as informações de maneira segura e precisa. Concluído este passo, a segunda etapa consiste em construir a ligação entre PC e Arduino. Nessa etapa optou-se primeiro utilizar o método através de cabo USB para se definirem os códigos mais adequados. Após

isso é convertida a transmissão de dados de USB para Xbee. Após essas etapas concluídas foi desenvolvido um aplicativo para *smartphones* com sistema Android, o qual enviará dados para o serviço Web, que foi criado no primeiro passo. E por último, foram efetuados todos os testes necessários para garantir um bom funcionamento de todas as ferramentas utilizadas, e também coletar dados bem relevantes sobre o conjunto todo.

7.1. MODELO DA SOLUÇÃO

Existem vários tipos de metodologia de desenvolvimento de software, a ideia foi utilizar a metodologia chamada Cascata. A metodologia Cascata conforme mostra a Figura 8 permite o desenvolvimento de software de uma maneira sequencial e é indicada para projetos que dependem de prazos.

Figura 8 - Modelo Cascata



Fonte: Wikipédia

7.1.1. ESTUDO DO PROBLEMA

A utilização da Domótica tem aumentado cada dia mais e com isso o mercado tem desenvolvido novos dispositivos para utilização na automação residencial. Afim de facilitar a vida das pessoas, foi pensado na ideia de desenvolver um dispositivo que controlasse o acionamento de outros dispositivos eletrônicos. A

ideia inicial era fazer com que através de um aplicativo em um *smartphone*, fosse possível fazer o acionamento de lâmpadas.

Para isso, pensou-se na ideia de se utilizar um controlador PIC que recebesse informações e realizasse o acionamento das lâmpadas. Em Galera (2008), foi utilizada a ideia de Domótica acionada através de celular utilizando plataforma J2ME. A partir de um estudo sobre Galera (2008) descobriram-se algumas falhas no projeto e algumas tecnologias que atualmente são consideradas ultrapassadas. Com isso teve-se a ideia de aprimorar tal projeto realizado por Galera e coletar informações precisas sobre consumo de energia, taxa de transmissão de dados e alcance, já que essas informações não foram analisadas em Galera (2008).

Outra questão que não foi tratada em Galera (2008) é sobre ter um histórico de acionamento de dispositivos. A ideia é que a partir desses históricos seja possível gerar relatórios que mostrem para o usuário qual o consumo de energia ou qual usuário esqueceu determinado dispositivo ligado, por exemplo.

7.1.2. ANÁLISE

Foi realizada uma análise sobre quais tecnologias deveriam ser substituídas por tecnologias mais recentes e que pudessem trazer um desempenho melhor.

A primeira tecnologia apontada para sua substituição foi a J2ME. A tecnologia J2ME foi muito utilizada em telefones móveis, mas atualmente tem sido esquecida, pois não oferece muitos recursos sobre o conceito *TouchScreen*. Com a entrada dos *smartphones* no mercado o conceito *TouchScreen* tem se disseminado rapidamente e existem *smartphones* que já aboliram o teclado físico. Para atender essa demanda foi avaliada a utilização de Java para Android. Escolheu-se o Android devido ser uma tecnologia *open-source* e também gratuita. Atualmente o custo de celulares com o sistema operacional Android instalado é bem mais baixo, quando comparado com o sistema IOS da Apple, instalados nos Iphones.

Outro ponto forte na utilização de Java para Android, é que os *smartphones* têm maior capacidade de processamento que os antigos celulares.

A segunda tecnologia a ser substituída é o controlador PIC, através de algumas pesquisas foi descoberto a plataforma Arduino que já traz consigo recursos para controle de dispositivos. O Arduino utiliza o conceito código aberto e também é reconhecido mundialmente.

A utilização do Arduino tem o objetivo de analisar o consumo de energia em relação ao controlador PIC e obter um controle maior sobre os dispositivos.

Para realizar a comunicação foi criado um *WebService* que cria logs de acionamento e de controle de acesso a dispositivos.

Para o armazenamento de informações foi utilizado um banco de dados, que fica instalado no mesmo servidor que hospeda o *WebService*.

7.1.3. PROJETO

A figura 9 representa o fluxo de ações e mensagens entre os elementos envolvidos.

O usuário realiza o controle dos dispositivos eletrônicos através de aplicações que rodem no *smartphone* ou através de um computador conectado à Internet.

Figura 9- Objetivo do Projeto

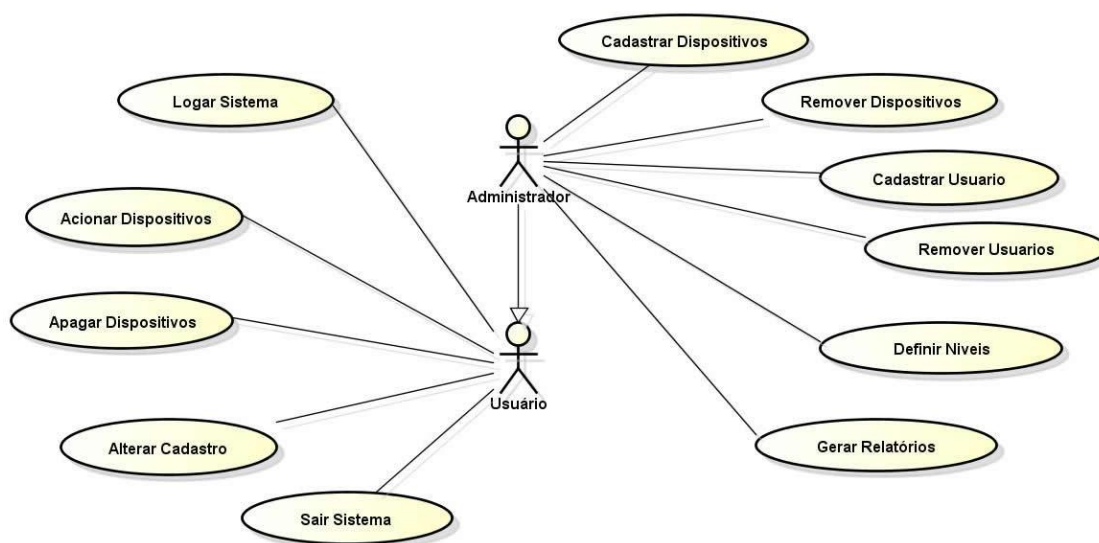


Fonte: Elaborado pelo Autor.

Para segurança do sistema, é obrigatória a autenticação do usuário. Toda vez que é aberto o sistema o usuário deve fornecer *login* e senha.

Existem dois tipos de perfis de usuário. O usuário do tipo Administrador tem permissão de acesso a todas as funções do sistema, conforme diagrama de caso de uso da Figura 10.

Figura 10 - Diagrama Caso de Uso

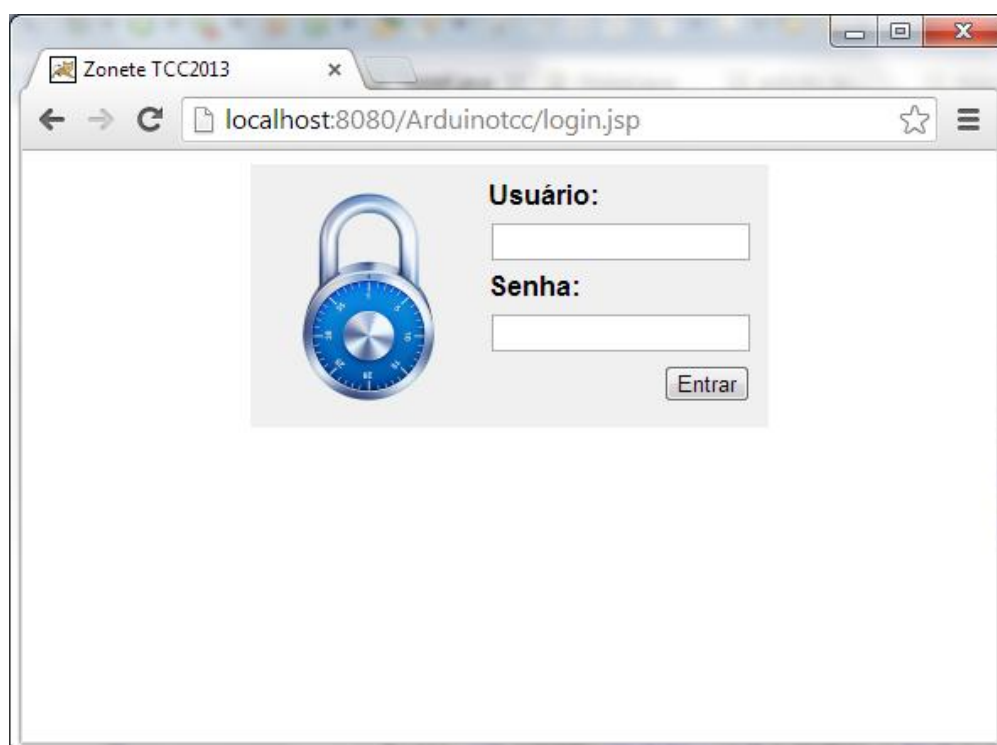


Fonte: Elaborada pelo Autor

Existe uma hierarquia nos perfis de usuário, como se pode ver no diagrama. O administrador tem acesso às ações do usuário, mas o usuário não tem acesso às ações do administrador.

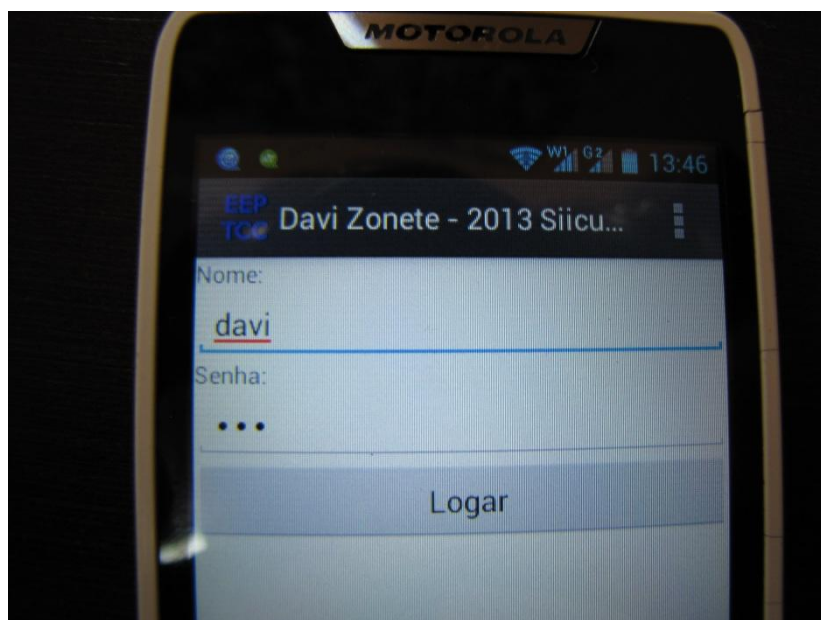
Quando o usuário acessar o sistema pelo computador através da Web, será exibida uma tela de *login*, conforme mostra a Figura 11. Caso o acesso seja feito pelo celular, também será requisitada a autenticação, conforme mostra a Figura 12. Através desse acesso ao sistema é possível liberar as opções de menu ao usuário, conforme o seu perfil.

Figura 11 - Login Sistema via Internet



Fonte: Elaborada pelo Autor

Figura 12 - Tela Login via Android

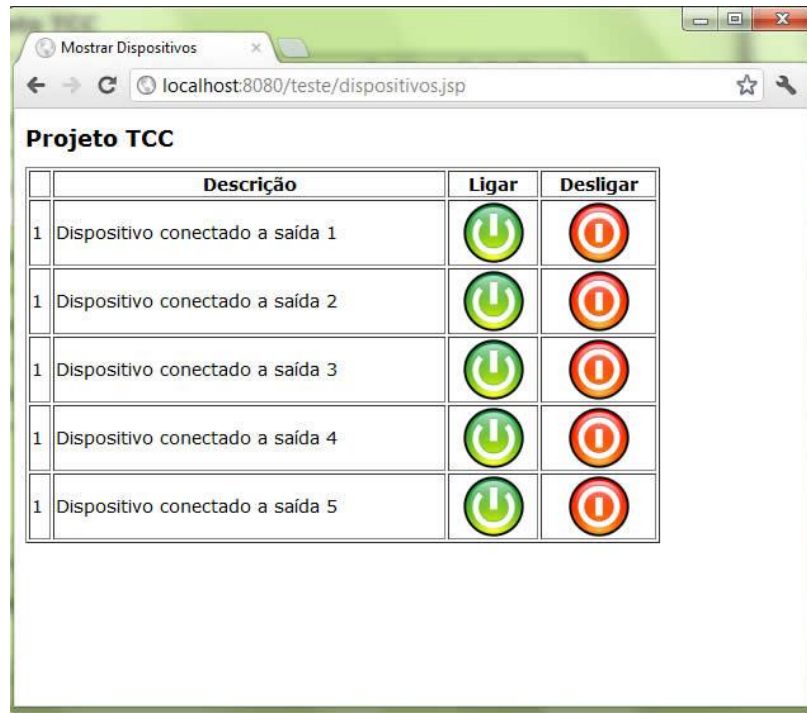


Fonte: Elaborada pelo Autor

Após o usuário ter sido validado, a próxima tela aberta é a de dispositivos, que mostra todos dispositivos conectados ao Arduino.

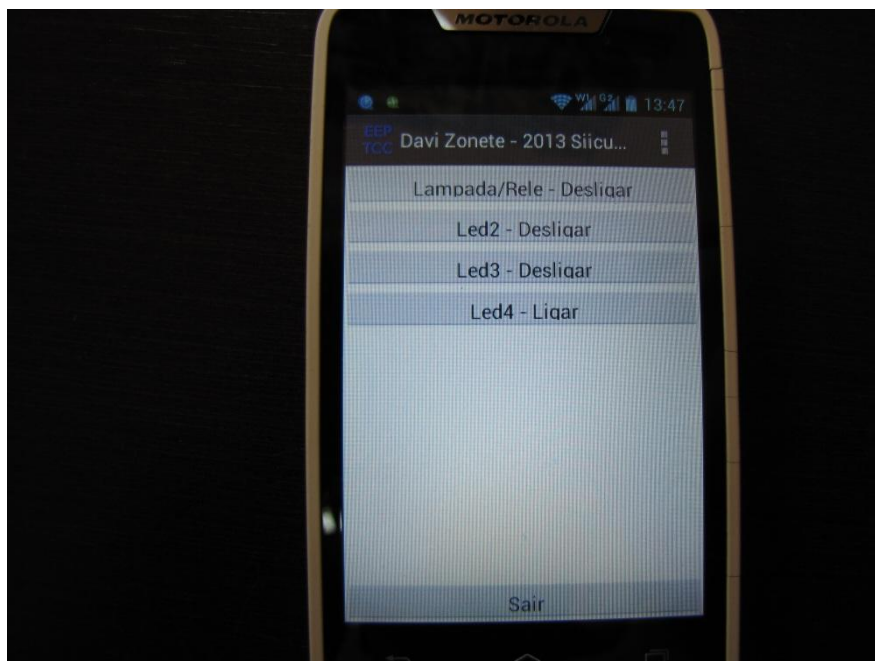
Ambas as telas, via web ou Android, são mostradas nas Figuras 13 e 14.

Figura 13 - Tela Dispositivos via Internet



Fonte: Elaborada pelo Autor

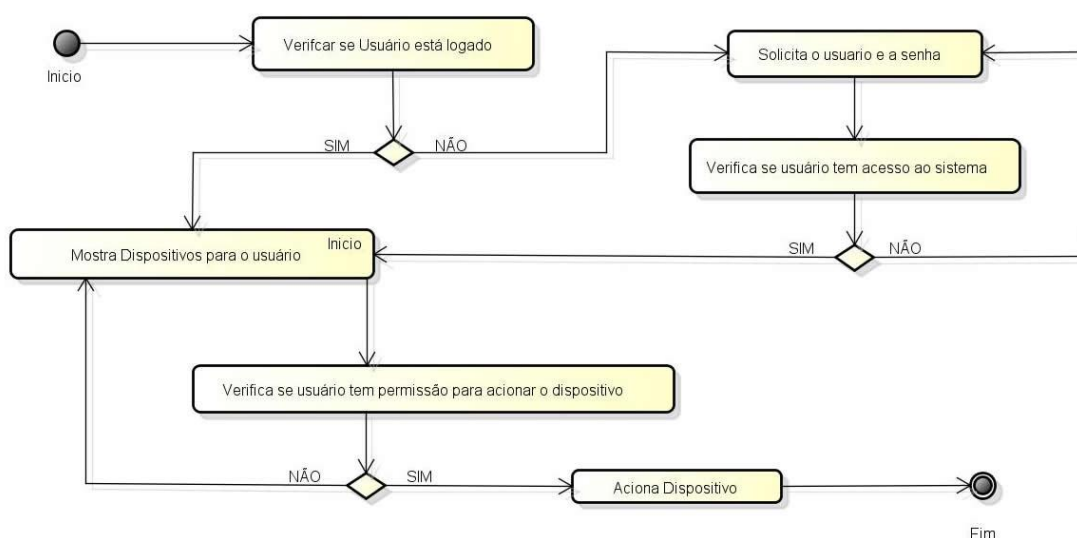
Figura 14 - Tela Dispositivos via Android



Fonte: Elaborada pelo Autor

O processo para acionamento de um dispositivo pode ser analisado no diagrama de atividade da Figura 15. A primeira verificação consiste em certificar que o usuário está logado. Caso o mesmo não esteja, será redirecionado para a tela de login (Figuras 11 e 12). Depois do usuário ter feito o *login*, serão mostrados os dispositivos ligados ao Arduino. Após o usuário escolher o dispositivo será verificado se o mesmo possui permissão de acionamento. Caso a permissão seja negada, o usuário deverá receber uma mensagem informativa e retornará à página de dispositivos (Figuras 13 e 14). Se a permissão for aceita, o sinal é enviado para o Arduino para acionar o dispositivo.

Figura 15 - Diagrama de Atividade



Fonte: Elaborada pelo Autor

A permissão é realizada sobre a tabela de níveis do usuário, onde cada usuário pertence a um nível. Cada dispositivo esta ligado a um ou vários níveis, então um dispositivo pode ser acionado por todos níveis ou apenas para um determinado nível, especificado pelo administrador do sistema. Isso permite um controle melhor dos dispositivos através da atribuição de níveis, como mostra a Tabela 2.

Tabela 2- Exemplo de Classificação de Níveis

Nível	Exemplo
1	Pai
2	Filhos menores de 10 anos
3	Filhos maiores de 10 anos

Fonte: Elaborada pelo Autor

Após definidos os níveis é possível definir quais níveis terão permissão para acionar dispositivos. A Tabela 3 relaciona os dispositivos aos códigos dos níveis da Tabela 2.

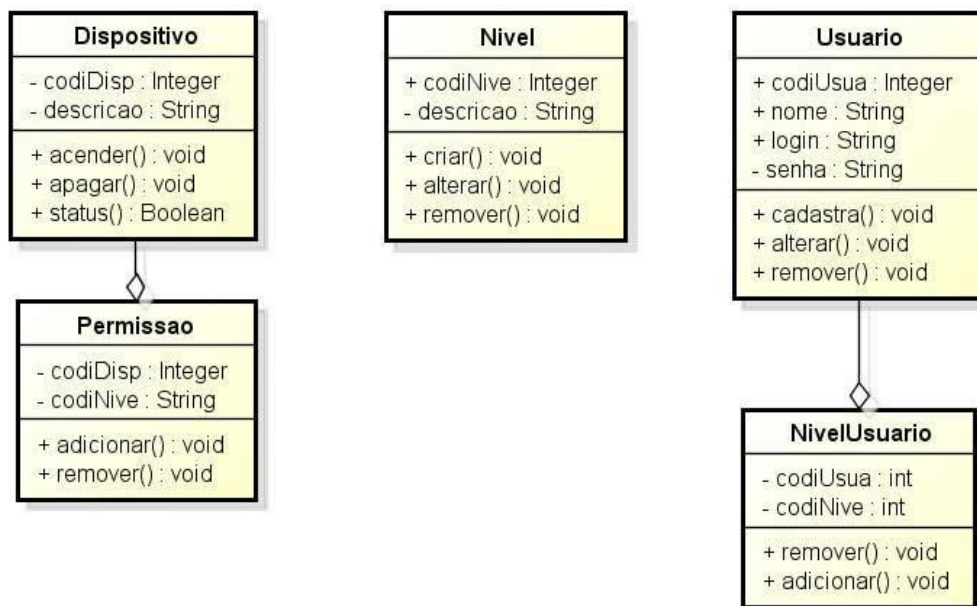
Tabela 3 - Exemplo de Dispositivos/Níveis

Dispositivos	Níveis com Permissão de Acionar
Luz da Sala de Estar	1,2,3
Luz da Cozinha	1,2,3
Portão Eletrônico	1,3
Cerca Elétrica	1

Fonte: Elaborada pelo Autor

O projeto utiliza linguagem orientada a objeto e, por isso, é necessário criar classes. A Figura 15 apresenta o diagrama de classes principais do sistema.

Figura 16- Diagrama de Classe



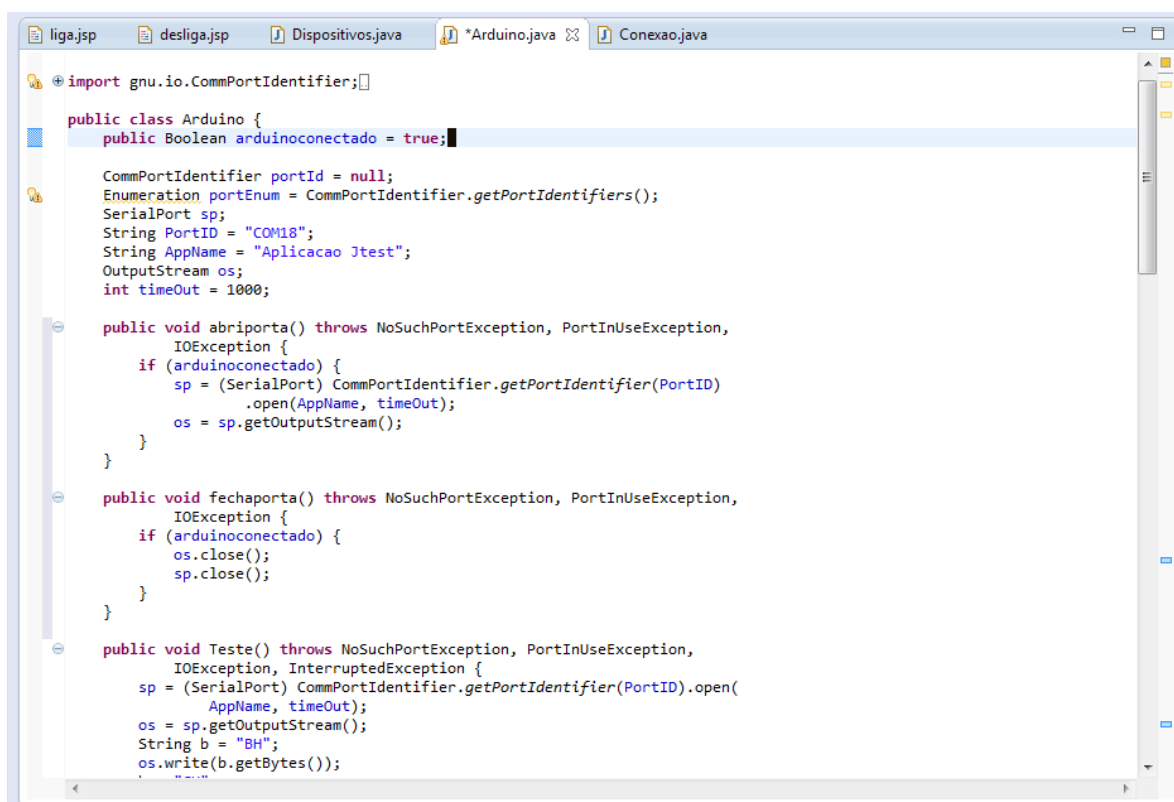
Fonte: Elaborada pelo Autor

7.1.4. CODIFICAÇÃO

Todo o processo de codificação deve utilizar alguns padrões de programação. O objetivo de se ter um padrão na codificação é facilitar a interpretação dos códigos para outros desenvolvedores.

O padrão mais importante que deve ser seguido é indentação. Todo código desenvolvido deve utilizar o padrão de indentação utilizado pelo Eclipse.

Figura 17 - Indentação Eclipse



Fonte: Elaborada pelo Autor

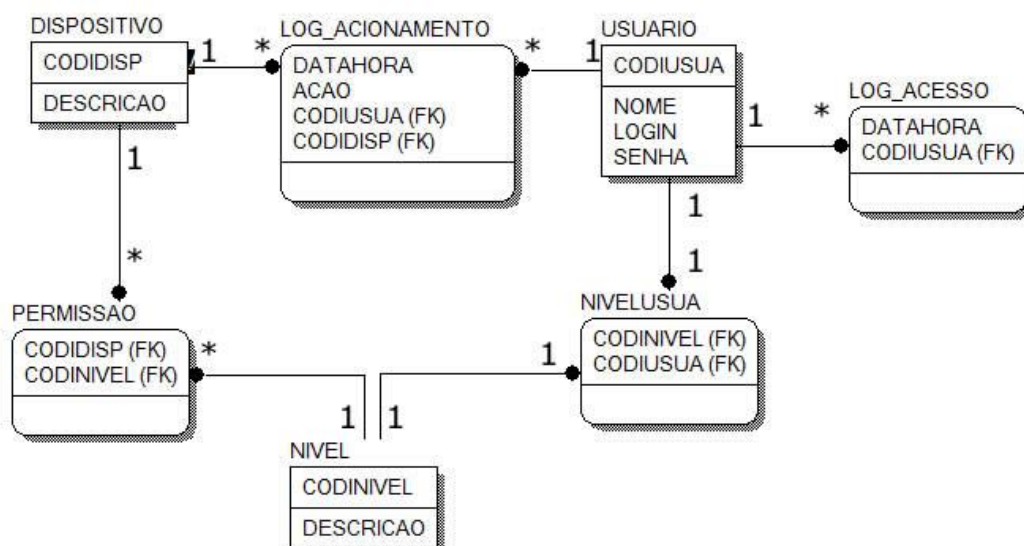
Toda classe criada deve seguir o padrão de nomes do Java, que requer que a primeira letra seja maiúscula. No exemplo da Figura 17 pode-se reparar que a classe foi criada seguindo o padrão (Arduino.java).

Os nomes das variáveis devem ter um tamanho mínimo de quatro letras e tamanho máximo de vinte letras.

A conexão do banco de dados deve ser feita através do JDBC, e toda vez que for aberta uma conexão esta deve ser encerrada após seu uso.

O banco de dados que armazena as informações tem a estrutura mostrada na Figura 18:

Figura 18 - DER



Fonte: Elaborada pelo Autor

A cada acionamento de dispositivo deve ser gerado um registro na tabela **LOG_ACIONAMENTO** e também deve ser gerado um registro quando este dispositivo for desligado.

Toda vez que o usuário fizer um acesso devem ser registradas na tabela **LOG_ACESSO** a data e hora do acesso, para controle do administrador do sistema.

As tabelas de log serão utilizadas para fazer duas análises. Uma análise deverá mostrar a quantidade de horas que ficou ligado determinado dispositivo no mês. E a outra análise deverá mostrar a quantidade de acessos por mês de cada usuário do sistema.

Essas análises deverão ser exibidas em gráficos, para que se possa visualizá-las melhor.

7.2. IMPLEMENTAÇÃO

O código desenvolvido para ser armazenado no Arduino tem o objetivo de receber a informação e, logo em seguida, executar a ação conforme a solicitação. Para executar essa tarefa foi necessário implementar um padrão de envio de dados através do protocolo Xbee.

São recebidos 2 caracteres. O primeiro caractere indica qual a saída a ser acionada, ou seja, o Arduino utilizado possui 13 portas de saída digitais. Com isso, foi montada uma tabela para que renomear cada entrada. A primeira saída foi classificada como entrada A, a segunda como saída B e assim sucessivamente. O segundo caractere indica qual a ação a ser executada. Existem dois tipos de ação, para isso foi definido H para ligar e L para desligar.

Na Tabela 4 pode-se ver a tabela verdade do Arduino.

Tabela 4 – Entrada de Dados Arduino.

1ºCaractere	2ºCaractere	Resultado
A	H	Ativar Saída 1
A	L	Desativar Saída 1
B	H	Ativar Saída 2
B	L	Desativar Saída 2
C	H	Ativar Saída 3
C	L	Desativar Saída 3
D	H	Ativar Saída 4
D	L	Desativar Saída 4
E	H	Ativar Saída 5
E	L	Desativar Saída 5
F	H	Ativar Saída 6
F	L	Desativar Saída 6
G	H	Ativar Saída 7
G	L	Desativar Saída 7
H	H	Ativar Saída 8
H	L	Desativar Saída 8
I	H	Ativar Saída 9
I	L	Desativar Saída 9
J	H	Ativar Saída 10
J	L	Desativar Saída 10
L	H	Ativar Saída 11
L	L	Desativar Saída 11
M	H	Ativar Saída 12
M	L	Desativar Saída 12
N	H	Ativar Saída 13
N	L	Desativar Saída 13

Fonte: Elaborada pelo Autor

Figura 19 – Definindo variáveis de saída

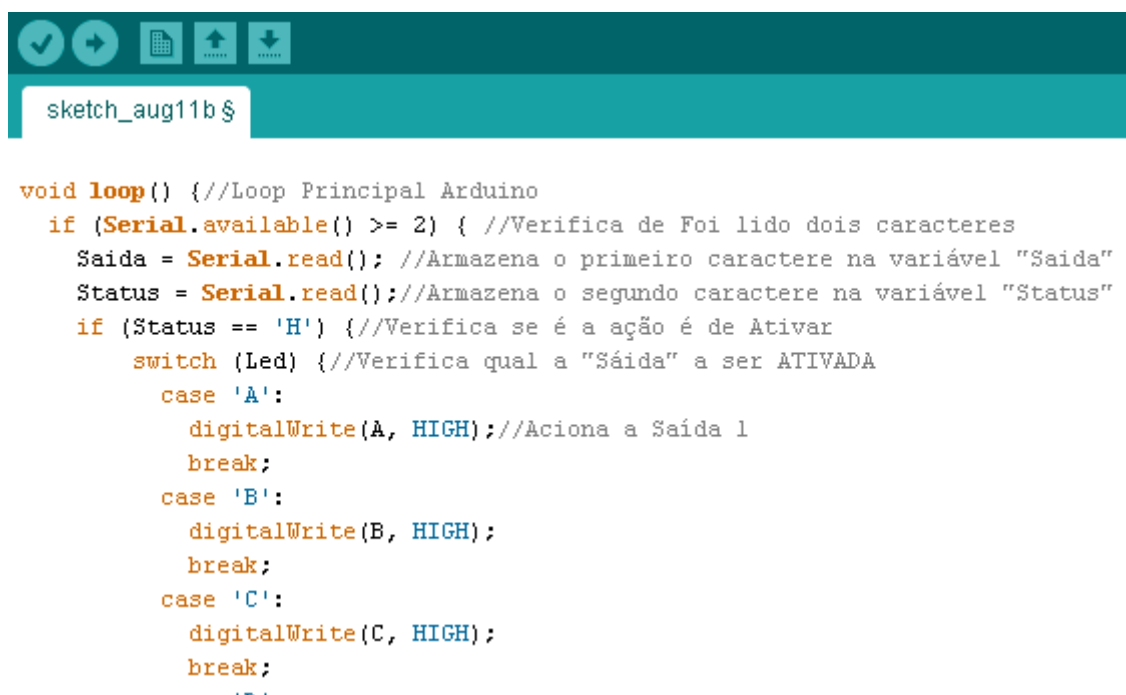
```
sketch_aug11b$  
//Primeiramente é definido os valores conforme a tabela de "saídas"  
const int A=1;  
const int B=2;  
const int C=3;  
const int D=4;  
const int E=5;  
const int F=6;  
const int G=7;  
const int H=8;  
const int I=9;  
const int J=10;  
const int L=11;  
const int M=12;  
const int N=13;
```

Fonte: Elaborada pelo Autor

Todo programa feito para o Arduino possui uma função principal, chamada `loop()`. Depois que o Arduino é ligado, essa função fica rodando infinitamente, com isso ele sempre verifica se foi passado algum parâmetro. Dessa forma adicionou-se a verificação dos caracteres definidos na Tabela 4, para que possam ser executadas as ações esperadas.

Na Figura 20 pode-se visualizar como é o funcionamento básico do programa armazenado dentro do Arduino.

Figura 20 – Principal função Arduino



```

void loop() { //Loop Principal Arduino
  if (Serial.available() >= 2) { //Verifica de Foi lido dois caracteres
    Saida = Serial.read(); //Armazena o primeiro caractere na variável "Saida"
    Status = Serial.read(); //Armazena o segundo caractere na variável "Status"
    if (Status == 'H') { //Verifica se é a ação é de Ativar
      switch (Led) { //Verifica qual a "Sáida" a ser ATIVADA
        case 'A':
          digitalWrite(A, HIGH); //Aciona a Saída 1
          break;
        case 'B':
          digitalWrite(B, HIGH);
          break;
        case 'C':
          digitalWrite(C, HIGH);
          break;
        case 'D':

```

Fonte: Elaborada pelo Autor

A implementação do serviço web, foi desenvolvida utilizando JSP, através da IDE Eclipse juntamente com o Apache Tomcat como servidor web. Para controle de usuário foi necessário estar criando algumas variáveis de sessão. Em todas as páginas criadas foi implementada uma função para verificar se a variável de sessão com o nome "codiuser" tinha sido definida. Caso essa variável estivesse com valor nulo, o usuário seria encaminhado para a página de login do sistema. Essa verificação foi criada a fim de evitar que pessoas indevidas viessem a executar ações inesperadas sobre o projeto.

Figura 21 – Verificação usuário

```

<%
//Verifica se usuário já está logado
if (session.getAttribute("codiuser") == null){
  //Mostra mensagem de Erro
  out.print("<script>alert('Acesso Restrito!');</script>");
  //Encaminha para página de login
  out.print("<meta http-equiv=\"refresh\" content=\"0;URL=login.jsp\">");
  out.close();
}

```

Fonte: Elaborada pelo Autor

Uma das principais classe do projeto desenvolvido é especialmente a classe Arduino java .Essa classe é responsável por realizar a conexão com o Xbee e também enviar as informações através do módulo Xbee. Pode-se verificar, na Figura 22, que a segunda linha é a declaração de uma variável do tipo booleana que recebe valores (True/False) e, com isso, ela sinalizará se o Adpatador Xbee está conectado ao computador.

Figura 22 – Principal função Arduino

```
public class Arduino {
    public Boolean arduinoconectado = true;

    CommPortIdentifier portId = null;
    Enumeration portEnum = CommPortIdentifier.getPortIdentifiers();
    SerialPort sp;
    String PortID = "COM18";
    String AppName = "Aplicacao Jtest";
    OutputStream os;
    int timeOut = 1000;

    public void abriporta() throws NoSuchPortException, PortInUseException,
        IOException {
        if (arduinoconectado) {
            sp = (SerialPort) CommPortIdentifier.getPortIdentifier(PortID)
                .open(AppName, timeOut);
            os = sp.getOutputStream();
        }
    }
}
```

Fonte: Elaborada pelo Autor

Após conectar o adaptador USB + módulo Xbee, esse adaptador é visualizado como uma porta COM pelo computador. Sendo assim, foi necessário ser implementada a comunicação Xbee, como se fosse uma comunicação com portas COM. Na figura 22 pode-se visualizar um trecho do código onde é possível ver a criação da conexão com a porta COM18.

A variável do tipo OutputStream pode ser considerada como o canal de comunicação pois será nela que se escreverão todas as mensagens enviadas para o Arduino, através do Xbee.

Figura 23 – Exemplo Acionamento

```

public void AcionarSaida1() throws PortInUseException, NoSuchPortException,
    IOException {
    sp = (SerialPort) CommPortIdentifier.getPortIdentifier(PortID).open(
        AppName, timeout);
    os = sp.getOutputStream();
    String AcionaSaidaUm = "AH";
    os.write(AcionaSaidaUm.getBytes());
}

```

Fonte: Elaborada pelo Autor

A comunicação com o banco de dados MySql foi desenvolvida utilizando JDBC. Na Figura 24 pode-se analisar como é realizada a conectividade com o banco de dados. Esse trecho de código pertence à classe Conexao.java.

Figura 24 – Conexão ao MySQL

```

public Statement abrirConexao() throws SQLException {
    try {
        //Atribui o driver
        String driver = "com.mysql.jdbc.Driver";
        Class.forName(driver);
        String banco = "tcc"; //Nome do DataBase
        String host = "localhost"; //Local do Banco de Dados
        String str_conn = "jdbc:mysql://" + host + ":3306/" + banco;
        String usuario = "root"; //Usuário para acesso ao BD
        String senha = ""; //Senha para acesso ao BD
        Connection conn = DriverManager.getConnection(str_conn, usuario, senha);
        statement = conn.createStatement();
        return statement;
    } catch (Exception e) {
        throw new SQLException(e.getMessage());
    }
}

```

Fonte: Elaborada pelo Autor

A gravação dos registros de acionamento é realizada toda vez que são chamadas as funções liga/desliga. Dentro de cada função é chamada a classe que realiza a conexão ao MySql e depois é inserido, através de comando SQL, um registro com os dados de quem acionou, data e hora do acionamento e qual dispositivo acionou. Na Figura 25 pode-se ver como é realizado o procedimento de gravação.

Figura 25 – Função de Liga()

```

public void liga(String porta, Integer codigouser, Integer codDisp)
    throws IOException, SQLException, NoSuchPortException,
    PortInUseException {
    try {
        if (arduinoconectado) {
            String b = porta + "H";
            os.write(b.getBytes());
            os.flush();
        }
        Conexao conn = new Conexao();
        Statement state = conn.abrirConexao();
        Date data = new Date(System.currentTimeMillis());
        SimpleDateFormat formatarDate = new SimpleDateFormat(
            "yyyy-MM-dd H:mm:ss");
        System.out.print(formatarDate.format(data));
        state.execute("insert into log_acionamento values ('"
            + formatarDate.format(data) + "','" + "1,"
            + codDisp.toString() + "','" + codigouser.toString() + "')");

        // this.fechaporta();
    } finally {
        // TODO Auto-generated catch block
        this.fechaporta();
    }
}
}

```

Fonte: Elaborada pelo Autor

A versão para Android foi desenvolvida utilizando a IDE Eclipse e a comunicação foi realizada sobre o protocolo HTTP. Com o objetivo de facilitar o desenvolvimento do aplicativo foi utilizada a classe CustomHttpClient.java a qual realiza toda a conexão através da Internet. Na Figura 26 pode-se ver o código que é responsável por chamar a página liga.jsp e passar os parâmetros corretos para que, posteriormente, esses parâmetros sejam passado para o Arduino, através do protocolo Xbee.

Figura 26 – Envio da informações para o Arduino

```

public void AcionaDisp(String Codigo, String user,String Liga) {

    ArrayList postParam = new ArrayList();
    postParam.add(new BasicNameValuePair("id", Codigo));
    postParam.add(new BasicNameValuePair("codus", user));
    String response = null;
    AlertDialog.Builder caixaalert = new AlertDialog.Builder(this);
    if(Liga.equals("0")){
        try {
            System.out.println("InfoDisp = " + Codigo + "/" + user);
            response = CustomHttpClient.executeHttpPost(
                "http://" + IP + ":8080/Arduinotcc/cel/liga.jsp",
                postParam);
            caixaalert.setMessage("Dispostivo " + Codigo + " - Ligado com sucesso");
            caixaalert.setTitle("Acao");
            caixaalert.setNeutralButton("OK",null);
            caixaalert.show();
        } catch (Exception e) {
            etNome.setText(e.toString());
        }
    }else{
        //...
    }
}

```

Fonte: Elaborada pelo Autor

7.3. TESTES

Na primeira etapa vislumbrou-se em se realizarem testes sobre a segurança na aplicação Web. O primeiro item a ser verificado é se todas as páginas jsp estavam conferindo se o usuário já havia realizado o acesso ao sistema, através da página de login. Para realização dessa etapa foram listadas todas as páginas jsp e, depois, efetuado diretamente o acesso a elas. Na Tabela 5 pode-se verificar em quais páginas foram realizados os teste de segurança.

Tabela 5 – Lista de Páginas JSP

Página JSP
analisehistorico.jsp
analiseresumida.jsp
desliga.jsp
disp.jsp
index.jsp
liga.jsp
listardisp.jsp
usuario.jsp
usuariodel.jsp
usuarioform.jsp

Fonte: Elaborada pelo autor

Também foram realizados testes sobre a gravação dos dados no banco de dados. Nessa etapa foi criada uma rotina que chamava a função liga e desliga exaustivamente, para poder se avaliar a confiabilidade de armazenamento dos dados e a velocidade de resposta do MySQL, em relação à sobrecarga do acesso.

Foi realizado teste de durabilidade das baterias ligadas ao projeto. Para a medição da durabilidade foram utilizadas duas baterias de 9V da mesma marca e fabricante, e com isso, foi verificada a quantidade de tempo que cada uma delas suportou o projeto. Para realização desse teste foi criada uma rotina em Java que chamava as funções de liga/desliga a cada 10 segundos. Com isso, era possível perceber o momento em que a bateria se esgota.

Sobre o projeto, também foi realizado teste de distância para verificar qual o alcance em campo aberto e com barreiras que o módulo Xbee conseguiria funcionar. Para realização dessa etapa foi utilizado o mesmo programa utilizado para testar a durabilidade das baterias. A cada ação recebida com sucesso era aumentada a distância em 5 metros ou 10 metros, caso fosse campo aberto. Com isso, foi possível obter resultados satisfatórios.

Os testes descritos acima foram realizados sobre um Arduino plugado com um Shield Relé com suporte a Xbee e, em uma das saídas foi adicionado um Led para verificar o acionamento (Figura 27).

Figura 27 – Arduino utilizado para testes



Fonte: Elaborada pelo autor

A unidade padrão para medir a distância suportada pela comunicação Xbee foi metros, e para realizar a medição de durabilidade da bateria foi utilizado como unidade padrão minutos.

8 RESULTADOS

Os resultados obtidos sobre os testes de segurança foram satisfatórios e também de extrema importância para realização das correções de algumas falhas. Com isso foi possível aumentar a segurança das transações de banco de dados. Na Tabela 6 pode-se verificar que em duas páginas jsp não haviam verificação de permissão de acesso e controle de acesso. Essas duas páginas foram corrigidas para evitar futuros problemas.

Tabela 6 – Resultados de Segurança

Página JSP	Segurança	Corrigido
analisehistorico.jsp	Ok	
analiseresumida.jsp	Falha	Sim
desliga.jsp	Ok	
disp.jsp	Ok	
index.jsp	Ok	
liga.jsp	Ok	
listardisp.jsp	Ok	
usuario.jsp	Ok	
usuariodel.jsp	Falha	Sim
usuarioform.jsp	OK	

Fonte: Elaborada pelo autor

Na etapa de gravação de dados os resultados obtidos foram fantásticos. Todas as transações efetuadas no banco de dados foram realizadas com sucesso. Foram realizadas ao todo 1000 transações de banco de dados em menos de 1 minuto. Vale lembrar que todas as transações eram de efeito de inserção de dados.

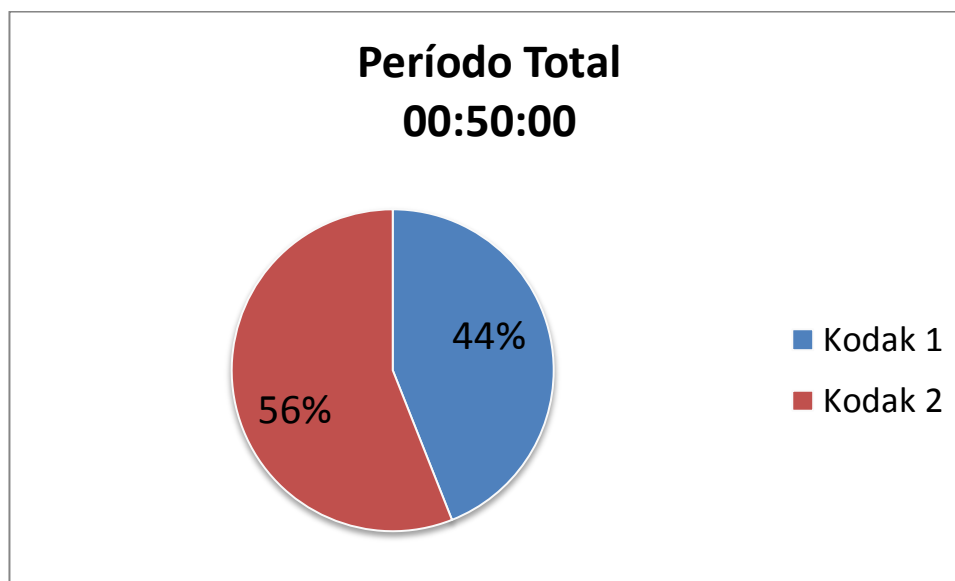
O resultado da durabilidade das baterias não foi muito agradável. Foram utilizadas duas baterias de 9V da marca Kodak. Cada uma delas conseguiu suportar o projeto no período médio de 25 minutos. Pode-se verificar esses valores na Tabela 7.

Tabela 7 – Resultados de durabilidade das baterias

Bateria	Início	Fim	Período	Média
Kodak 1	21:50	22:12	00:22	00:25:00
Kodak 2	22:30	22:58	00:28	

Fonte: Elaborada pelo autor

Gráfico 28 – Durabilidade das baterias



Fonte: Elaborada pelo autor

Foram realizados alguns testes sobre as análises existentes no projeto e foi checado se os valores eram coerentes. Os resultados obtidos foram satisfatórios e muito interessantes, pois através deles é possível descobrir qual equipamento está sendo mais utilizado e em quais momentos, e por quais usuários.

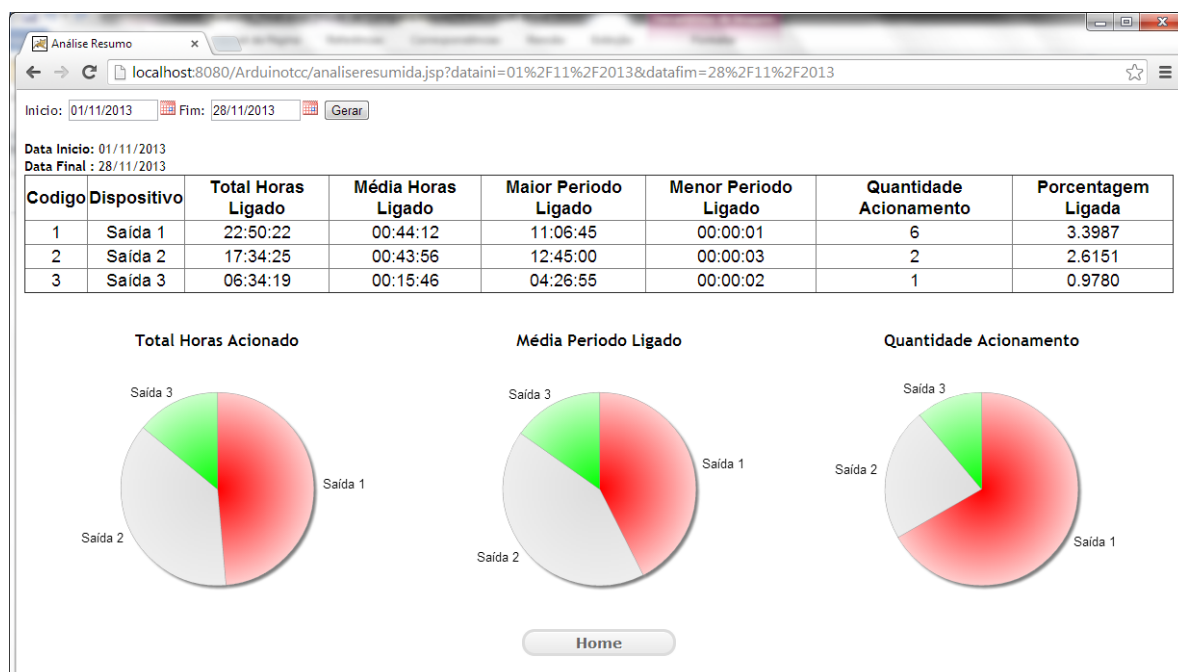
No projeto foram desenvolvida duas análises. A primeira análise é sobre o histórico de acionamento de dispositivo (Figura 29). Através dela é possível visualizar em qual horário determinado dispositivo foi acionado e quem efetuou o desligamento do mesmo. Como é armazenada a data de início e de fim, foi possível realizar a diferença entre as horas e verificar o período de utilização desse equipamento. Na análise de histórico, a última linha é referente a um totalizador de período, ou seja, a soma de todos os períodos. A segunda análise é mais resumida (Figura 30). Ela é baseada no histórico e tem o objetivo de mostrar uma informação mais precisa. Nela é possível verificar qual o maior e menor período de uso do equipamento, a média do período utilizado e também a quantidade de acionamento desse dispositivo. Também é possível visualizar a porcentagem de utilização referente ao período da análise, ou seja, a porcentagem é calculada em cima do período informado da análise.

Figura 29 – Análise Histórico

Cód.	Usuário Ligou	Início	Fim	Cód.	Usuário Desligou	Duração
1	DAVI	2013-10-05 10:15:48	2013-10-05 10:15:21	1	DAVI	00:00:27
1	DAVI	2013-10-05 10:18:04	2013-10-05 10:17:32	1	DAVI	00:00:32
1	DAVI	2013-10-05 13:40:43	2013-10-05 10:18:14	1	DAVI	03:22:29
1	DAVI	2013-10-22 12:56:41	2013-10-22 12:55:58	1	DAVI	00:00:43
1	DAVI	2013-10-22 12:56:55	2013-10-22 12:56:45	1	DAVI	00:00:10
1	DAVI	2013-10-22 12:56:55	2013-10-22 12:56:50	1	DAVI	00:00:05
1	DAVI	2013-10-22 12:57:03	2013-10-22 12:56:59	1	DAVI	00:00:04
1	DAVI	2013-10-22 12:58:13	2013-10-22 12:58:08	1	DAVI	00:00:05
1	DAVI	2013-10-22 12:59:19	2013-10-22 12:59:15	1	DAVI	00:00:04
1	DAVI	2013-10-22 13:00:10	2013-10-22 13:00:06	1	DAVI	00:00:04
1	DAVI	2013-10-22 13:00:18	2013-10-22 13:00:15	1	DAVI	00:00:03
1	DAVI	2013-10-22 13:29:04	2013-10-22 13:26:59	1	DAVI	00:02:05
1	DAVI	2013-10-22 13:39:47	2013-10-22 13:39:44	1	DAVI	00:00:03
1	DAVI	2013-10-22 13:40:02	2013-10-22 13:39:55	1	DAVI	00:00:07
1	DAVI	2013-10-22 13:58:08	2013-10-22 13:45:29	1	DAVI	00:12:39
1	DAVI	2013-11-05 19:28:06	2013-11-05 19:28:01	1	DAVI	00:00:05
1	DAVI	2013-11-05 21:53:35	2013-11-05 21:09:09	0	Ainda está ligado	00:44:26
Total:						04:24:11

Fonte: Elaborada pelo autor

Figura 30 – Análise Resumida



Fonte: Elaborada pelo autor

É importante lembrar que ambas as análises são geradas a partir de uma data de início e uma data final. O tempo de geração dos resultados não ultrapassou 10 segundos considerando um período de 3 meses.

Os resultados obtidos nos testes de distância foram satisfatórios para o projeto. O Xbee chega a enviar dados a uma distância de 100 metros em campo aberto e a 30 metros em ambiente fechado com barreiras (Tabela 8 e Tabela 9).

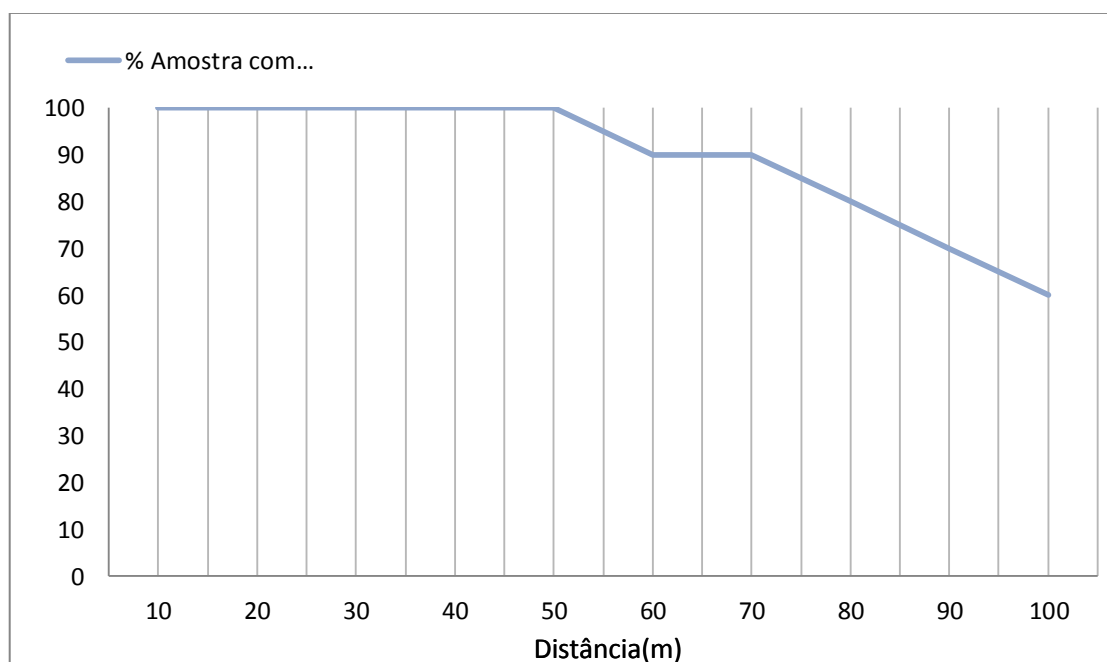
Tabela 8 – Distância de comunicação Xbee sem barreira

Distância(m)	Quantidade Amostra	Amostra com sucesso	%
10	10	10	100
20	10	10	100
30	10	10	100
40	10	10	100
50	10	10	100
60	10	9	90
70	10	9	90
80	10	8	80
90	10	7	70
100	10	6	60

Fonte: Elaborada pelo autor

A amostragem dos dados foi dividida em distâncias de 10 metros, iniciando em 10 metros até 100 metros. Em cada divisão eram disparadas 10 ações sobre o Arduino através do protocolo Xbee. Cada ação executada com sucesso foi considerada como uma amostra com sucesso. No Gráfico da Figura 31 pode-se observar que quanto maior a distância menor a confiabilidade do recebimento dos dados através do Xbee.

Gráfico 31 – Amostragem sem barreira



Fonte: Elaborada pelo autor

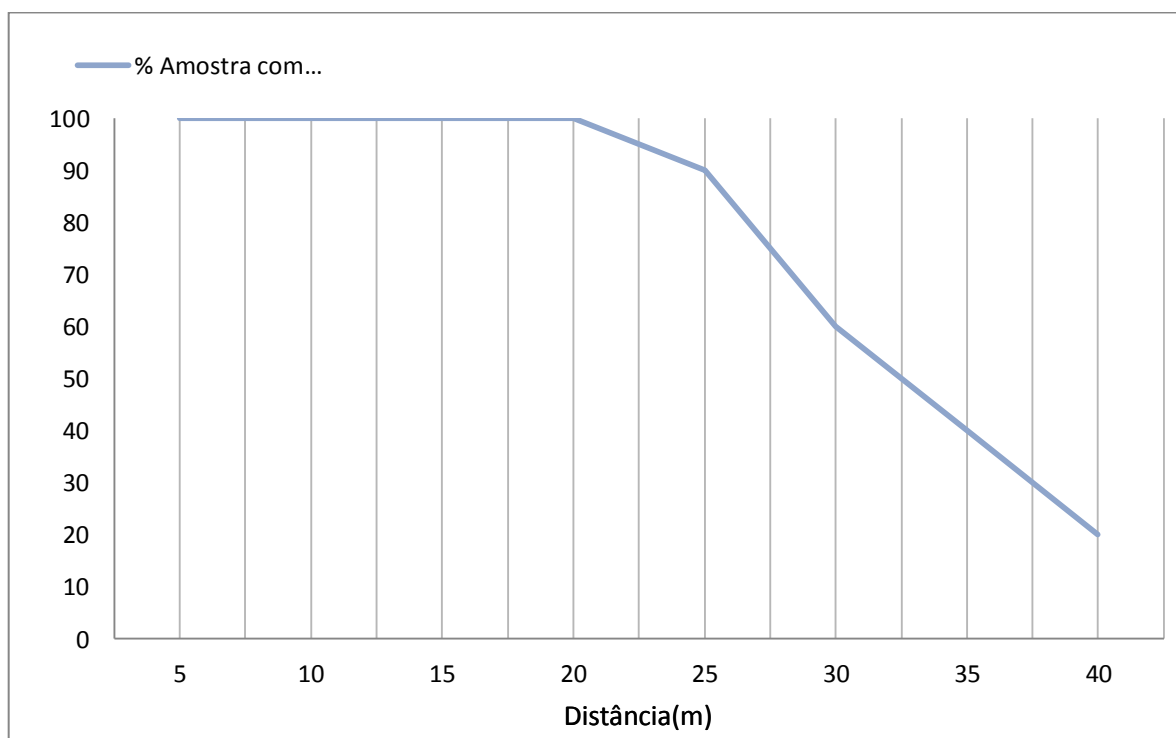
Tabela 9 – Distância de comunicação Xbee com barreira

Distância(m)	Quantidade Amostra	Quantidade Barreira	Amostra com sucesso	%
5	10	1	10	100
10	10	1	10	100
15	10	2	10	100
20	10	2	10	100
25	10	2	9	90
30	10	3	6	60
35	10	4	4	40
40	10	4	2	20

Fonte: Elaborada pelo autor

O teste de distância com barreiras foi realizado em um espaço menor com 5 metros em cada etapa. Dentro de cada etapa manteve-se a quantidade de amostragens de 10 ações. Na Tabela 9 pode-se verificar que existe uma coluna referente a quantidade de barreiras. No experimento, essas barreiras são paredes de largura aproximadamente 15 cm.

Gráfico 32 – Amostragem com barreira



Fonte: Elaborada pelo autor

Os resultados deste trabalho também foram publicados no Simpósio Internacional de Iniciação Científica da USP, que ocorreu em Outubro de 2013 (ZONETE, 2013).

9 CONCLUSÕES

Através dos resultados obtidos pode-se concluir que a utilização da placa controladora Arduino, juntamente com o módulo Xbee, obteve resultados melhores em relação à placa controladora PIC utilizada em Galera (2008). Pode-se citar, entre as vantagens, o alcance de comunicação entre os módulos de 1mW. Em Galera (2008) não foram mostrados resultados precisos sobre o alcance do Bluetooth, com isso foi realizada uma pesquisa em Alecrim (2008) que mostrou que a utilização do módulo Bluetooth com a potência de 1mW, a mesma utilizado sobre o módulos Xbee, permite um alcance de até 1 metro. Porém, o Xbee deixa a desejar em relação a sua taxa de transmissão, quando comparado ao Bluetooth a taxa de transmissão do Xbee é muito inferior. Portanto pode-se concluir que o Xbee, para a utilização da Domótica em residências, é leve, pois a quantidade de informações passada pelo protocolo Xbee é muito pequena. Com isso, a taxa de transferência não interfere de forma drástica sobre o projeto. O Xbee tem outra vantagem em relação ao Bluetooth, por ser um protocolo mais específico ele acaba sendo mais seguro que o Bluetooth.

Com relação ao aplicativo desenvolvido em Android, conclui-se que a comunicação com o serviço Web foi desenvolvida de uma maneira bem simples e fácil. Tudo isso graças a facilidades que a tecnologia Android agrega. Um ganho sobre a tecnologia J2ME é a utilização do *TouchScreen*, que permite o usuário interagir com o aplicativo com facilidade muito maior.

Com os resultados obtidos é possível concluir que, para a utilização do projeto, é aconselhável a utilização de uma fonte de energia de 9V, ao invés de utilizar baterias convencionais de 9V, devido à baixa durabilidade das baterias.

Pode-se dizer que a mesma estratégia utilizada para comunicação entre o Arduino e o serviço Web, pode ser estendida para a utilização na área industrial.

9.1. SUGESTÕES PARA TRABALHOS FUTUROS

Para um trabalho no futuro pode ser criado um web service utilizando uma tecnologia restfull, para que se possa estar verificar o ganho de performance. Assim,

também pode ser alterada a forma de envio dos dados, que hoje utiliza o método POST, os dados podem ser enviados através de um arquivo XML.

Também pode ser implementado o Hibernate para o controle de transações ocorridas sobre o banco de dados MySql, a fim de ter um controle maior e evitar problemas.

A aplicação desenvolvida em Android poderia ser desenvolvida em um ambiente *multi-thread*. Atualmente ela está desenvolvida em um ambiente *single-thread*.

REFERÊNCIAS

GALERA,F.C. Domótica Utilizando dispositivos móveis de controle sem fio. Trabalho de conclusão de curso-Fundação Municipal de Ensino de Piracicaba. Piracicaba 2008.

FRANCESCHI,M. *Arduíno, o documentário do hardware livre, leve e solto.* Disponível em <<http://baixacultura.org/2011/10/28/arduino-o-documentario-do-hardware-livre-leve-e-solto/>>
Acesso 20 de maio 2012.

MESSIAS.A. Controle remoto e aquisição de dados via XBee/ZigBee (IEEE 802.15.4)
Disponível em <<http://www.rogercom.com/ZigBee/ZigBee.htm>>
Acesso 01 de maio 2012

ALECRIM.E. Tecnologia Bluetooth: o que é e como funciona?
Disponível em < <http://www.infowester.com/bluetooth.php>>
Acesso 10 de outubro 2013

ADAMI.A. Domótica
Disponível em < <http://www.infoescola.com/tecnologia/domotica/>>
Acesso 01 de outubro 2013

SCHUTZER.W. Projeto com microcontrolador PIC
Disponível em < <http://www.dm.ufscar.br/~waldeck/pic/>>
Acesso 15 de outubro 2013

ZONETE.D Controle de dispositivos eletrônicos em Domótica via smartphones
21º Simpósio Internacional de Iniciação Científica da USP
São Carlos, 2013

ANEXO A – Certificado de Participação no 21º SIICUSP

ANEXO B – Banner do trabalho no 21º SIICUSP