**FACULDADE DE TECNOLOGIA DE SÃO JOSÉ DOS CAMPOS  
FATEC PROFESSOR Jessen Vidal**

# OSMAR SALLES DE CARVALHO

## TÍTULO DA TESE TÍTULO DA TESE TÍTULO DA TESE TÍTULO DA TESE TÍTULO DA TESE TÍTULO DA TESE

São José dos Campos  
2013

# OSMAR SALLES DE CARVALHO

## TÍTULO DA TESE TÍTULO DA TESE TÍTULO DA TESE TÍTULO DA TESE TÍTULO DA TESE TÍTULO DA TESE

Trabalho de Graduação apresentado à Faculdade de Tecnologia São José dos Campos, como parte dos requisitos necessários para a obtenção do título de Tecnólogo em Informática com Ênfase em Banco de Dados.

**Orientador: Prof. Me. Aníbal Evaristo Fernandes**

São José dos Campos  
2013

**Dados Internacionais de Catalogação-na-Publicação (CIP)  
Divisão de Informação e Documentação**

SOBRENOME, Nome do Aluno  
Título do Trabalho de Graduação.   
São José dos Campos, 200X.  
999f. (número total de folhas do TG)

Trabalho de Graduação – Curso de Tecnologia em (Logística, Informática ou Aeronáutica) com  
Ênfase em Xxxxxxx, FATEC de São José dos Campos: Professor Jessen Vidal, 2013. Orientador: Titulação Nome do Orientador.

1. Áreas de conhecimento. I. Faculdade de Tecnologia. FATEC de São José dos Campos: Professor Jessen Vidal. Divisão de Informação e Documentação. II. Título

**REFERÊNCIA BIBLIOGRÁFICA –**

SOBRENOME, Nome do Aluno. **Título do Trabalho de Graduação.** 2013. 999f. Trabalho de Graduação - FATEC de São José dos Campos: Professor Jessen Vidal.

**CESSÃO DE DIREITOS –**

NOME DO AUTOR: Osmar Salles de Carvalho

TÍTULO DO TRABALHO: Título do Trabalho de Graduação

TIPO DO TRABALHO/ANO: Trabalho de Graduação / 2013.

É concedida à FATEC de São José dos Campos: Professor Jessen Vidal permissão para reproduzir cópias deste Trabalho e para emprestar ou vender cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte deste Trabalho pode ser reproduzida sem a autorização do autor.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Osmar Salles de Carvalho  
Rua Procópio Ferreira 164, Jd. Nova Detroit  
CEP 12224-570 – São José dos Campos – São Paulo

**OSMAR SALLES DE CARVALHO**

TÍTULO DA TESE TÍTULO DA TESE TÍTULO DA TESE TÍTULO DA TESE TÍTULO DA TESE TÍTULO DA TESE

Trabalho de Graduação apresentado à Faculdade de Tecnologia São José dos Campos, como parte dos requisitos necessários para a obtenção do título de Tecnólogo em Informática com Ênfase em Banco de Dados.

**Composição da Banca**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Nome do Componente da Banca, titulação e Instituição**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Nome do Componente da Banca, titulação e Instituição**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Prof. Me. Aníbal Evaristo Fernandes, Fatec São José dos Campos**

**\_\_\_\_\_/\_\_\_\_\_/\_\_\_\_\_  
DATA DA APROVAÇÃO**

O autor oferece a obra (elemento sem titulo e sem indicativo numérico), ou presta homenagem a alguém, de forma clara e breve em folha única.

**Resumo**

Apresentação concisa dos pontos relevantes do documento deve ser exposta no resumo. No presente caso o resumo será informativo, assim deverá ressaltar o objetivo, a metodologia, os resultados e as conclusões do documento. A ordem desses itens depende do tratamento que cada item recebe no documento original. O resumo deve ser composto por uma seqüência de frases concisas, afirmativas e não em enumeração de tópicos. Deve ser escrita em parágrafo único e espacejamento de 1,5. A primeira frase deve ser significativa, explicando o tema principal do documento. Deve-se usar o verbo na voz ativa e na terceira pessoa do singular. Quanto a sua extensão, o resumo deve possuir de 150 a 500 palavras.

**Palavras-Chave**: Com máximo 10 palavras, separadas entre si por ponto e vírgula “;” e finalizadas por ponto. As palavras-chave sãopalavras representativas do conteúdo do documento.

**ABSTRACT**

O abstract é o resumo da obra em língua estrangeira, que basicamente segue o mesmo conceito e as mesmas regras que o texto em português. Recomenda-se que para o texto do abstract o autor traduza a versão do resumo em português e faça, se necessário, os ajustes referentes à conversão dos idiomas. É importante observar que o título e texto NÃO DEVEM estar em itálico.Ouvir

Keywords: Recomenda-se que o autor traduza para o inglês as Palavras-Chave em português e faça, se necessário, os ajustes referentes à conversão dos idiomas.

# SUMÁRIO

# 1- INTRODUÇÃO 00

## 2- REVISÃO DE LITERATURA 00

### 2.1- Android 00

### 2.2- Arduino 00

### 2.3- Bluetooth 00

### 2.4- Tipos de Automação Residencial 00

### 2.4.1- Automação Via Bluetooth 00

### 2.4.2- Automação Via Zigbee e Servidor 00

### 2.4.3- Automação Via Circuito Lógico Programável (CLP) 00

### 2.4.4- Automação Via Ethernet 00

## 3- MATERIAIS E MÉTODOS 00

### 3.1- IDE Eclipse e IDE Arduino 00

## 3.2- Android SDK para Eclipse 00

### 3.3- O Microcontrolador Arduino e suas Vantagens 00

### 3.4- Módulo Atuadores do Sistema – Bluetooth e Relé 00

### 3.5- Comunicações Bluetooth entre Android e Arduino 00

### 3.6- SQLite 00

### 3.7- Estrutura do Projeto 00 **3.7.1**- Protótipo

### 3.7.2- Algoritmo Utilizado Para Envio de Informações 00

### 3.7.3- Diagrama de Caso de Uso 00

### 3.7.4- Diagrama de Sequência 00

### 3.7.5- Diagrama UML de Classe 00

### 3.8- Estrutura Final do Sistema de Controle 00

### 3.8.1- Interface com Usuário 00

### 3.8.2.1- Tela Inicial 00

### 3.8.2.2- Login 00

### 3.8.2.3- Menu Principal 00

### 3.8.2.4- Controle 00

### 3.8.2.5- Usuários 00

### 3.8.2.6- Ferramentas 00

### **3.8.2.7-** Log 00

### **3.8.2.8-** Sobre 00

### 3.8.2- Estrutura do Banco de Dados 00

## 4- RESULTADOS E DISCUSSÕES 00

## 5- CONCLUSÃO 00

**1- INTRODUÇÃO**

A área de Tecnologia da Informação (TI) tem transformado significativamente a maneira como vivemos. De fato, a integração da TI com distintos setores da indústria vêm proporcionando muitos benefícios para as áreas econômicas, políticas, culturais e de entretenimento (BEAL, 2001).

Uma revolução significativa que pode ser verificada é a popularização dos smartphones e dos tablets, que possibilitaram uma mudança radical no comportamento das classes sociais A, B e C. Essa última, constituída por um grande volume de pessoas que, de acordo com ABRANET (2011), representam 47% da população brasileira e passou a ser considerada prioridade pelo setor de marketing de muitas empresas para a venda de jogos eletrônicos, música, serviços Web.

Segundo Millennial Media (2012), os smartphones e os tablets são aparelhos eletro portáteis contendo um sistema operacional de alto desempenho que tem como os mais conhecidos o Android e IOS. O T-Mobile fabricado pela HTC foi primeiro Smartphone com Android (LECHETA, 2010). Eles também podem se conectar com outros dispositivos de diversas maneiras como via USB, Bluetooth, Wireless.

Os Smartphones reúnem diversas capacidades como: de um celular, Personal Digital Assistant – PDA (também conhecido como Palmtop) e de computador, já os tablets não tem a capacidade de realizar ligações. De acordo com Lecheta (2010), atualmente esses equipamentos estão realizando vários tipos de tarefas como: jogos, GPS, acesso a internet, e-mail, musicas, Bluetooth e uma ótima interface visual, ou seja, esses aparelhos estão cada vez mais presentes na vida das pessoas. Estudo divulgado pela empresa de consultoria Strategy Analytics (2012), aponta que até o terceiro trimestre de 2012 o total de smartphones superou um bilhão de aparelhos no mundo.

Para adicionar uma maior funcionalidade a esses dispositivos existem os micro controladores que cria uma interface que permite aos dispositivos smartphone e tablets controlarem equipamentos físicos tais como, lâmpadas, exaustores e aquecedores segundo (OXER E BLEMINGS, 2009).

O micro controlador com um maior destaque no mercado atualmente é o Arduino que tem como objetivo segundo Oxer e Blemings (2009), criar um equipamento open-source, acessível, de baixo custo, flexível e fácil de ser utilizado. Pode-se conectar esse dispositivo diretamente para controlar um equipamento, mas também pode conecta-lo em um computador, que através das portas de entrada e de saída fornece uma interface serial ou USB para tal conexão, toda codificação é realizada utilizando um computador e gravada no Arduino através dessas entradas (OLIVEIRA E SILVA, 2012).

De acordo com Bolzani (2004), ambientes inteligentes são aqueles que aperfeiçoam suas funções referentes à operação e a administração de uma residência. Sendo uma novidade, a automatização residencial de inicio é apenas símbolo de status, mas que logo se torna um meio de conforto e que por fim fornece um fator de economia se tornando essencial. (AURESIDE, 2013)

Nesse contexto o objetivo desse trabalho é desenvolver um sistema computacional que através de um micro controlador e um dispositivo com Sistema Operacional Android, ambos com comunicação Bluetooth, controlar determinadas funções de uma residência, como (Luz, TV, Fechadura das Portas entre outros) de uma maneira mais rápida e eficaz, auxiliando assim as pessoas nessas simples tarefas, de forma que melhore a qualidade, o conforto e a maneira com que as pessoas realizam essas atividades.

É importante ressaltar que as funções de controle serão enviadas do dispositivo Android para o Arduino através de uma conexão Bluetooth, fazendo então, que um equipamento elétrico ou eletrônico seja capaz de realizar tarefas pré-programadas e ordenadas pelo Android, também podendo ser ativadas tarefas, como a função Casa Em Modo Viagem, que fará com que a residência se controle automaticamente de tal maneira simulando assim a presença de pessoas.

Os equipamentos que serão utilizados neste trabalho são: Smartphone LG Optimus L5 (E612F) com Android 4.0.3, um micro controlador Arduino Uno R3, 14 Pinos de Entrada, Memória Flash de 32KB, a conexão entre os dois dispositivos será realizada através de um Módulo Bluetooth Master e Slave, Frequência 2.4 GHz com alcance máximo de 10 metros que se conecta ao Arduino, e que através de um código de verificação validará os dispositivos Android permitidos a utilizar o programa de controle residencial.

Com isso, no capítulo 2 terá uma leve abordagem sobre os atuais sistemas de automatização residencial de maneira a entender como funcionam, quais equipamentos utilizam e qual é a solução adotada atualmente.

**2 – REVISÃO DE LITERATURA**

Este capítulo irá mostrar fundamentos sobre a automação residencial via Bluetooth e quais as soluções implementadas atualmente para esse tipo de tecnologia.

**2.1 – Android**

De acordo com Korjenioski (2011), a Google em agosto de [2005](http://pt.wikipedia.org/wiki/2005) adquiriu a Android Inc., uma pequena empresa de Palo Alto (Califórnia - EUA) que desenvolvia uma plataforma para celulares, baseada em [Linux](http://pt.wikipedia.org/wiki/Linux_(n%C3%BAcleo)), com o objetivo de ser flexível, aberta e de fácil migração para os fabricantes (HASHIMI E KOMATINEMI, 2009).

Atualmente essa plataforma segundo Rabello (2007), é mantida pelo Open Handset Alliance, que é um grupo constituído por mais de 30 empresas, entre elas estão gigantes como Sony, Samsung, HTC entre outras. Essas se uniram para inovar e acelerar o desenvolvimento de aplicações, serviços e tecnologias, trazendo aos usuários uma experiência rica em recursos e baratas em termos financeiros.

O Android foi desenvolvido baseado no sistema operacional Linux. Sendo assim todas as propriedades essenciais desse sistema foram incorporadas, como: Sistema de arquivos, Kernel, servidores de terminais, entre outras (AN, 2011).

**2.2 – Arduino**

É um projeto Open-Source baseado em uma placa microcontroladora simples, e um ambiente de desenvolvimento para escrever os códigos que serão gravados na placa (ARDUINO, 2013).

Pode-se utilizar o Arduino para criar objetos interativos, recebendo entradas a partir de diversos tipos de sensores e controlando equipamentos de diferentes maneiras como: acendendo e apagando luzes, ligando e desligando motores elétricos além de outras saídas físicas (ARDUINO, 2013).

**2.3 – Bluetooth**

O Bluetooth foi criado em torno de muitas empresas de grande porte como Motorola, Dell, Samsung, Siemens entre outras, com um mesmo interesse, criar um tecnologia que troque informações a partir de um meio sem fio (BLUEMEL, 2009).

De acordo com Kobayashi (2004), o Bluetooth é uma tecnologia *wireless* com pouco consumo de energia e baixo custo financeiro, seu principal objetivo é conectar diferentes dispositivos de distintos fabricantes tornando possível a troca de informações, arquivos por esses equipamentos.

Segundo Silva e Oliveira (2012), Essa comunicação é feita através de radiofrequência e no caso do Bluetooth é dividido em três classes:

1. Classe A com alcance máximo de 100 metros,
2. Classe B com raio de 10 metros e
3. Classe C com somente 1 metro de alcance entre os módulos.

O sistema Bluetooth tem como unidade fundamental o piconet, que é um nó mestre e até sete nós escravos ativos dentro de uma distância. É possível utilizar diversas piconets em um mesmo ambiente e essas podem ser conectadas umas as outras através de um nó ponte, e quando se há diversas redes piconets conectadas essa passa a ser chamada de scatternet (TANENBAUM, 2003).

**2.4 – Tipos de Automação Residencial**

Atualmente no mercado de automação residencial, apresentam-se alguns tipos de tecnologias, dos mais simples e baratos aos mais complexos e caros, podendo ser utilizando tecnologias como Bluetooth e Zigbee, CLP ou via servidor.

**2.4.1 – Automação via Bluetooth**

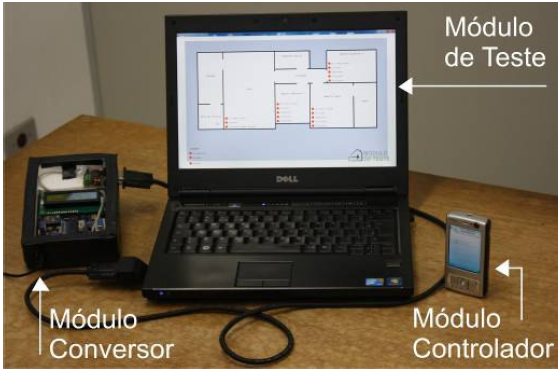
Abreu e Valim (2011), desenvolveram um modelo de automação utilizando um aparelho celular Nokia N95 com sistema operacional Symbian que tem como linguagem nativa o C++, a IDE NetBeans 6.9.1, instalada em um computador com sistema operacional Windows 7.

Após a conexão Bluetooth ser estabelecida é apresentado no aparelho celular uma interface exibindo os cômodos da residência, após o usuário escolher um cômodo o aplicativo dispõe os objetos que são automatizados, sendo disponível liga-lo, desliga-lo ou alterar sua intensidade.

Alguns dos componentes utilizados para o desenvolvimento desse modelo de automação foram: Um conector DB9 (fêmea), que realiza a conexão física do cabo serial ao módulo Bluetooth, um CI MAX232 que converte o nível de tensão RS232 para TTL, um led para sinalizar quando o módulo Bluetooth recebe conexões e um cristal que exibe a frequência de funcionamento do micro controlador, um display LCD de 16x2, onde são informados os dados recebidos pelo módulo controlador PIC18F252.

Para realizar as validações foi desenvolvido um módulo de testes que recebe os dados através de uma porta serial e simula os cômodos e os objetos a serem controlados. Toda validação do projeto foi utilizando interface de log onde se visualiza o endereço MAC do dispositivo local e remoto, a URL de conexão e os comandos enviados.

O aplicativo foi testado nos aparelhos celular Nokia N95, Nokia N98, Nokia 6822 e Motorola Defy MB525 e foi observado que o sinal Bluetooth sob interferências de sinais externos como roteadores WI-FI e telefones sem fio, consomem mais energia, pois tem que realizar mais saltos entre as frequências possíveis.

Figura 1: Ilustração do sistema em módulo de teste

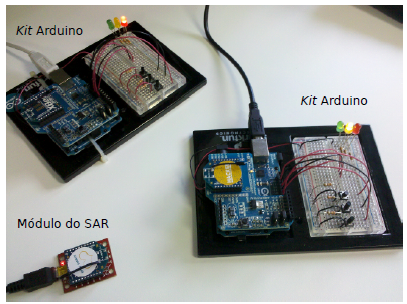
Fonte: Abreu e Valim (2011)

**2.4.2 – Automação via Zigbee e Servidor**

Outro modelo de automação residencial é a via tecnologia Zigbee, que é um protocolo criado para sistemas sem fio que não necessitam de altas taxas de transmissão (ZIGBEE ALLIANCE, 2005).

No modelo desenvolvido por Euzébio (2011) utilizou-se uma arquitetura centralizada, através de um servidor central responsável por comunicar-se com as interfaces e os dispositivos controladores. Sendo o dispositivo Android a interface de controle, e o Arduino o controlador de dispositivo, ambos utilizando o Zigbee para comunicar-se com o servidor.

Para simular esse modelo foram utilizados led’s, um Kit Arduino, um módulo Xbee. A figura 1 mostra a estruturação do projeto.

Figura 2: Estruturação do projeto

Fonte: Euzébio (2011)

Os led’s fazem o papel das lâmpadas e os botões conectados fazem o papel dos interruptores das lâmpadas. O módulo SAR (Servidor de Automação Residencial), tanto se comunica pelo protocolo Zigbee quanto pelo TCP/IP, esse deve permanecer constantemente conectado, pois deve gerenciar todas as mensagens trocadas entre os dispositivos da rede, o servidor foi desenvolvido em forma de um software executado em um computador pessoal.

Nesse trabalho também foi testado aplicações web utilizando serviços web, que tem como principal característica operar em sistemas distintos com uma comunicação multiplataforma. O que faz essa funcionalidade possível é a utilização de arquivos XML para realizar a comunicação entre os dispositivos. Para o desenvolvimento do SAR foram utilizado a linguagem Java para ter acessos diretos as requisições HTTP recebidas pelo servidor web e também servidor de aplicativo como o Glassfish.

Figura 3: Ilustração da interface do sistema DroidlarFonte: Euzébio (2011).

**2.4.3 – Automação via Circuito Lógico Programável (CLP)**

No trabalho feito por Alievi (2008) exibe o projeto de automação *SmartHouse* criado em Washington E.U.A., e que não disponibiliza o controle dos equipamentos para um dispositivo móvel, mas sim, uma automação inteligente o suficiente para tomar as decisões de acordo com valores lidos por diversos sensores distribuídos no ambiente que se encontram instalados.

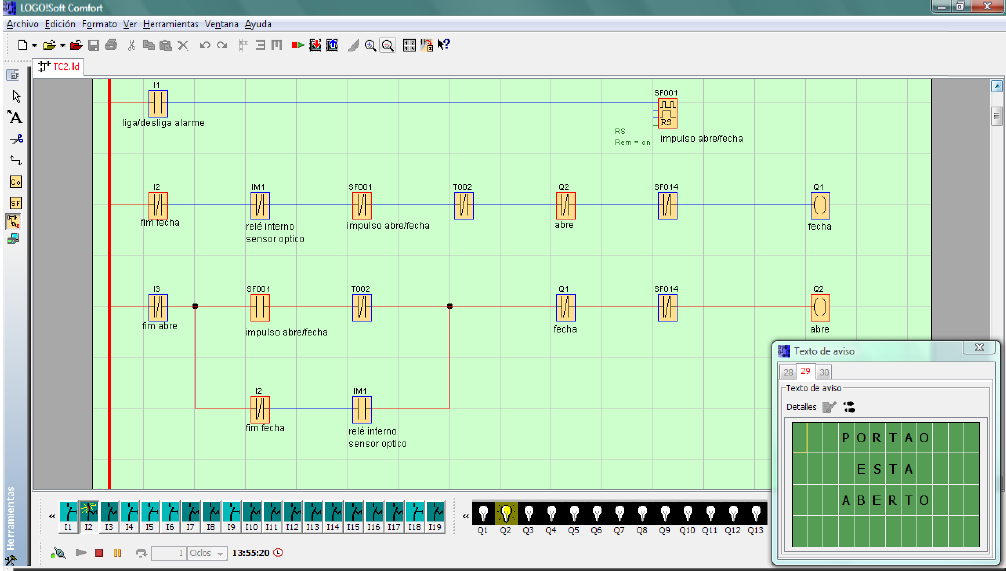
A instalação dos equipamentos é unificada e com inteligência distribuída que toma conta da distribuição de energia, do controle das comunicações e do sistema de telefonia. Quando um aparelho é acionado ele envia um sinal para o computador central para fazer a validação se o sinal está de acordo com as especificações só então o sistema libera energia para seu funcionamento.

Essas funções também são utilizadas no sistema de equipamentos que funcionam a gás, reduzindo assim riscos de choques e vazamentos, pois logo que o servidor percebe que algum equipamento parou de enviar o sinal de identificação, ele corta a energia disponível para esse equipamento.

Todo aparelho a ser utilizado deve seguir o protocolo de comunicação programado no sistema central, caso algum equipamento não seja compatível com o meio de conexão, é utilizado adaptadores de entrada e saída para comunicação tanto para dispositivos elétricos quanto para equipamentos a gás.

Nesse projeto foram realizados estes tipos de automação: Sistema de automação de portão de contra peso, sistema de alarme de segurança patrimonial, sistema de iluminação externa, sistema de iluminação interna, sistema de *stand-by* da sala de estar, sistema de irrigação de jardim e sistema de segurança residencial contra incêndio.

Uma vantagem desse modelo de automação, é que ele dispensa o uso permanentemente de um computador conectado ao microcontrolador, a utilização de interface homem/máquina muito elaboradas, controles remotos muito complexos.

Figura 4: Ilustração do ambiente de desenvolvimento do sistema controle do portão.

Fonte: Alievi, (2008).

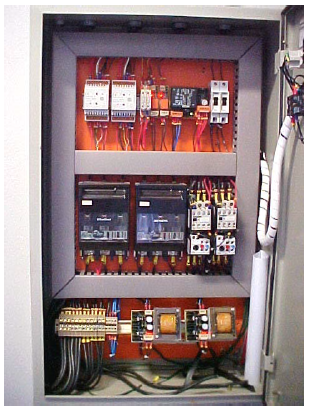
**2.4.4 – Automação via Ethernet**

Sena (2005) descreve um modelo de automação via ethernet criada pela indústria Européia de microprocessadores, criando assim uma tecnologia barata para aplicação da domótica no mercado residencial, para esse modelo deu-se o nome de European Home System (EHS) e esse se baseia no modelo OSI (Open Standard Interconnection).

Nesse projeto Sena (2005) divide em dois principais setores, são esses: Pavimento superior e térreo. No pavimento superior foi implantado circuito de TV para monitoramento interno através das câmeras instaladas na residência, também se podem controlar os portões por meio dos porteiros eletrônicos, a iluminação desse setor é reduzida pela metade a partir das 23 horas. Também foi instalados sistemas de controle nas cortinas que são acionadas por controle remoto.

No térreo existem algumas câmeras que fazem o monitoramento 24 horas por dia, distribuídas nos muros, sala de espera e na garagem da residência. Todas as câmeras enviam a imagem para um servidor localizado em uma sala técnica que as digitaliza e envia para uma central de conectividade que podem ser observadas por um canal de TV ou via internet. Nas entradas existem fechaduras biométricas que quando ativadas envia uma mensagem para a central de controle que interpreta a digital e libera a abertura da porta.

O cabeamento utilizado para transmissão de dados e voz são do tipo RG-6, cabo múltiplo composto de dois cabos UTP, dois cabos coaxiais categoria 5 e um cabo de fibra ótica. Na sala técnica existe um quadro de conectividade que comporta os módulos de controle. Um das vantagens desses módulos é que quando for necessário realizar a expansão da automação na residência, basta acrescentar mais módulos controladores ao quadro de conexão.

Figura 5: Exemplo de quadro de conectividade.  
Fonte: Sena (2005).

**3- MATERIAIS E MÉTODOS**

**3.1- IDE Eclipse e IDE Arduino**

Figura 6: IDE (*Integrated Development Environment*) do Eclipse



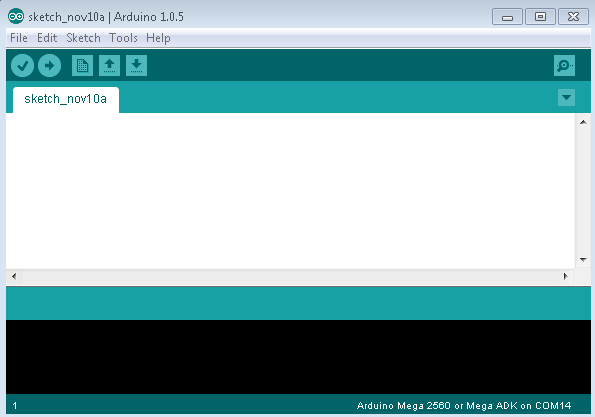
Fonte: Eclipse, (2013).

De acordo com Burnett, 2006, o Eclipse é uma IDE “Para qualquer coisa e nada mais”, ou seja, pode ser utilizado para desenvolver softwares em diversas linguagens e não se bastando apenas ao Java, como é mais conhecido. No inicio, o Eclipse era apenas um substituto do Visual Age For Java, da IBM, mas após ter o código-fonte aberto (*open source*) em 2001 tornou-se um dos principais IDE’s de desenvolvimento tendo desde 2001 mais de 50 milhões de downloads. Segundo Burnett, 2006, o Eclipse é mantido pelo Eclipse Foundation, uma organização sem fins lucrativo e independente desde 2001.

A escolha dessa plataforma se deu por algumas características, dentre elas destacam-se:

* Software Livre;
* Auxilia no desenvolvimento e construção de aplicações;
* Depuração integrada em tempo real, ajudando a encontrar os erros de compilação;
* Ambiente amigável;
* Plataforma portável e extensível, o qual permite adicionar outras ferramentas.

A IDE do Arduino é de código aberto assim como o próprio Arduino, e pode ser baixada através do site: http://arduino.cc/en/Main/Software, nela pode-se desenvolver programas que utilizam diversos componentes lógicos e eletrônicos, sendo possível gravar e executar o código no microcontrolador Arduino(ARDUINO, 2013). Na figura 7 mostra a interface da IDE do Arduino.

Figura 7: IDE do Arduino

Fonte: JAYCONSYSTEMS, (2013).

**3.2- Android SDK para Eclipse**

Segundo Lecheta (2010), o Android SDK (Software Development Kit) é o software utilizado para desenvolver aplicações no Android. Ele conta com um emulador para simular o smartphone, ferramentas utilitárias e uma API completa para a linguagem Java, incluindo todas as classes necessárias. Com o plug-in do Android SDK para Eclipse é possível executar o emulador diretamente do Eclipse sendo a aplicação instalada automaticamente tanto no emulador quando em um smartphone real conectado ao computador pela porta USB.

De acordo com Lecheta, 2010, O plug-in suporta tais sistemas operacionais:

* Windows XP (32 bits), Vista (32 e 64 bits) e Windows 7 (32 e 64 bits);
* Mac OS X 10.5.8 ou posterior (Somente x86);
* Linux (apenas Ubuntu).

**3.3- Microcontrolador Arduino e suas Vantagens**

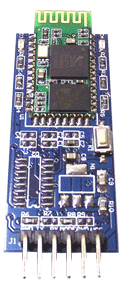
De acordo com ARDUINO (2013), o Arduino é uma ferramenta para tornar os computadores capazes de detectar e controlar mais do mundo físico. É um projeto Open-Source baseada em uma placa micro controladora simples, e um ambiente de desenvolvimento para escrever os códigos que serão gravados na placa.

Pode-se utilizar o Arduino para criar objetos interativos, recebendo entradas a partir de diversos sensores e controlando equipamentos de diversas maneiras como: acendendo e apagando luzes, ligando e desligando motores elétricos e muitas outras saídas físicas. O Arduino pode ficar tanto stand-alone quanto se comunicar com software rodando em outro computador, segundo (ARDUINO, 2013).

As placas podem ser montadas por qualquer pessoa ou comprada pronta, sua IDE é de código aberto e pode ser baixada gratuitamente. Sua principal linguagem de programação é o wiring, baseada no C e no C++ de acordo com (ARDUINO, 2013).

Existem outros micros controladores e plataformas disponíveis tais como: Parallax Basic Stamp, Netmedia do BX-24, Phidgets, Handyboard do MIT, e diversas mais oferecem funcionalidades semelhantes. Porém essas ferramentas contêm detalhes muito confusos da programação de microcontroladores ao contrario do Arduino que simplifica todo o processo de se trabalhar com micro controlador, o que é uma grande vantagem para ser utilizado por professores, estudantes e amadores interessados no assunto, de acordo com (ARDUINO, 2013).

Um dos motivos de se uso, são suas vantagens como:

* Barato: O Arduino é relativamente barato se comparado a outros micro controladores. Em sua versão mais simples o módulo pode ser montado à mão e ainda é possível encontrar a placa montada por mais ou menos R$60,00.
* Multiplataforma: Seu software funciona tanto em Macintosh OSX, Linux e no Windows, sendo que a maioria dos outros softwares de micro controladores somente é executada no sistema operacional da Microsoft.
* Programação simples e clara: O ambiente de programação do Arduino é fácil para iniciantes, mas também é flexível permitindo que usuários mais experientes também possam utilizar e aproveitar o seu melhor.
* Código aberto.
*  Software e Hardware extensível: Sua linguagem pode ser expandida através de bibliotecas C++.

**3.4- Módulos Atuadores do Sistema – Bluetooth e Relé**

Figura 8: Módulo Bluetooth

Fonte: Web Trônico, (2013).

De acordo com Guia de Usuário do Módulo Bluetooth (2010), esse dispositivo contém dois modos de operação, que são:

* Ligação Automática: Modo normal de funcionamento do dispositivo.
* Modo de Comando.
* Para variar entre os modos, deve-se alterar o valor do pino 11 entre baixo e alto, sendo que estando como baixo, o funcionamento do dispositivo está em modo normal, já no estado de alto ele opera em modo de comando.

Após ter comunicação estabelecida, pode-se enviar e receber dados através da interface UART (Universal Asynchronous Receiver-Transmitter), tudo de maneira sem fio. O modo de comando é a forma avançada do módulo Bluetooth, onde o usuário pode alterar as configurações padrão, como nome do dispositivo, senha e até mesmo taxa de transmissão (GUIA DE USUÁRIO DO MÓDULO BLUETOOTH, 2010).

O módulo Relê para acionamento de cargas através de micro controladores como o Arduino, ocorre através de um sinal lógico de 5 volts. Com esse módulo é possível acionar cargas de até 5500 watts com total segurança para o micro controlador, que fica protegido por opto acopladores contra surtos de até 5 mil volts (SERIAL LINK, 2011).

As Características do módulo relê são:

* Saídas com opção Normalmente Aberto (NO) e Normalmente Fechado (NC)
* Saídas de 5500 Watts por contato (10 Ampère em 110 Volts ou 7 Ampère em 220 Volts)
* Proteção por opto acopladores contra surtos de até 5 mil volts
* Led indicador de acionamento da Saída
* Conectores parafusáveis para garantir uma conexão segura das cargas
* Acionamento através de sinal lógico de 5 Volts
* Regulador de Tensão de para alimentação externa de 12 Volts
* Duas formas de alimentação
  + Através dos 5 Volts do Arduino, ou;
  + Através dos 12 Volts de fonte de alimentação externa
* Com alimentação externa de 12 Volts, o módulo pode alimentar o Arduino com 5 Volts.

Figura 9: Módulo Relê

  
Fonte: ROBOCORE, (2013).

**3.5- Comunicações Bluetooth entre Android e Arduino**

De acordo com Android Developer (2013), para que ocorra a comunicação entre o Android e o Arduino, é necessário utilizar algumas técnicas como:

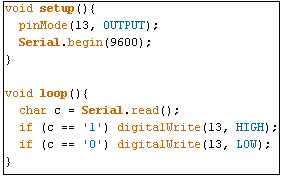
* [***BluetoothAdapter***](http://developer.android.com/reference/android/bluetooth/BluetoothAdapter.html)**:** Representa o adaptador *Bluetooth* local (radio *Bluetooth*). O [*BluetoothAdapter*](http://developer.android.com/reference/android/bluetooth/BluetoothAdapter.html) é o ponto de entrada para todas as interações *Bluetooth*. Através desse método pode-se encontrar outros dispositivos *Bluetooth*, consultar uma lista de dispositivos emparelhados, instanciar um [*BluetoothDevice*](http://developer.android.com/reference/android/bluetooth/BluetoothDevice.html) usando um endereço MAC conhecido, e criar um [*BluetoothServerSocket*](http://developer.android.com/reference/android/bluetooth/BluetoothServerSocket.html) para escutar por comunicações de outros dispositivos.
* [***BluetoothDevice***](http://developer.android.com/reference/android/bluetooth/BluetoothDevice.html)**:** Simula um dispositivo remoto *Bluetooth*. Utilizado para requerer uma conexão com um dispositivo remoto através de um [*BluetoothSocket*](http://developer.android.com/reference/android/bluetooth/BluetoothSocket.html) ou consultar informações sobre o dispositivo, como seu nome, endereço, e estado da ligação.
* [***BluetoothSocket***](http://developer.android.com/reference/android/bluetooth/BluetoothSocket.html)**:** Representa a interface para um *socket* *Bluetooth*. Ele é um ponto de conexão que permite a troca de dados entre dispositivos *Bluetooth* através do *InputStream* e *OutputStream*.
* [***BluetoothServerSocket***](http://developer.android.com/reference/android/bluetooth/BluetoothServerSocket.html)**:** Simula um *socket* de servidor que escuta por requisições que chegam ao dispositivo onde a aplicação roda. De forma a conectar dois dispositivos Android, um dos dispositivos precisa abrir um *socket* servidor com essa classe. Quando um dispositivo *Bluetooth* faz uma requisição de conexão a esse dispositivo, o [*BluetoothServerSocket*](http://developer.android.com/reference/android/bluetooth/BluetoothServerSocket.html) retornará um [*BluetoothSocket*](http://developer.android.com/reference/android/bluetooth/BluetoothSocket.html)  conectado quando a conexão é aceita.

Também é preciso configurar o arquivo Android *Manifest* para permitir a utilização do *Bluetooth* do smartphone pela aplicação, para isso é necessário utilizar a seguinte permissão:

**<uses-permission android:name="android.permission.BLUETOOTH"/>**

Para que o Arduino receba os comandos enviados pelo Android, é necessário iniciar a porta serial 9600 e a partir dela verificar o valor recebido, como pode ser visto na imagem a seguir.

Figura 10: Leitura da porta serial do Arduino

****Fonte: Do Autor, (2013).

**3.6- SQLite**

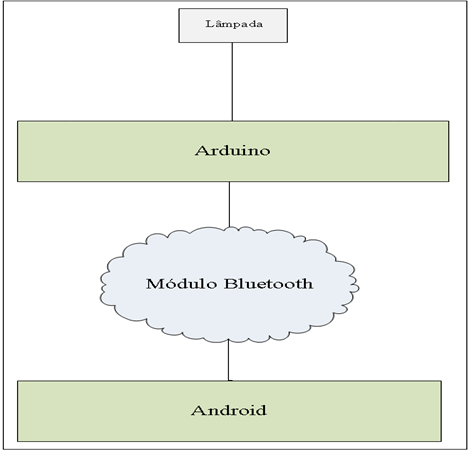
De acordo com SQLite (2013), o SQLite é um banco de dados transacional, não necessita de configuração, não precisa de servidor separado, seu código-fonte é de domínio público e atualmente encontra-se em milhares de aplicativos. Nele podem-se ter tabelas, campos, triggers, índices, views é um banco de dados completo, ele pode ser copiado entre sistemas operacionais 32 e 64 bits.

Segundo Lecheta (2010), o Android tem suporte ao banco de dados SQLite, sendo que cada aplicação pode criar um ou vários bancos de dados. No Android o banco de dados somente é visível para a aplicação que a criou e existem várias formas de criar um banco de dados, são estas:

* Via código utilizando a API do Android para SQLite.
* Através de um cliente do SQLite como o SQLite Expert Personal disponível para download em: http://www.softsland.com/sqlite\_expert\_personal.html.
* Ou utilizando o aplicativo Sqlite3 pelo console do emulador.

**3.7 – Estrutura do Projeto**

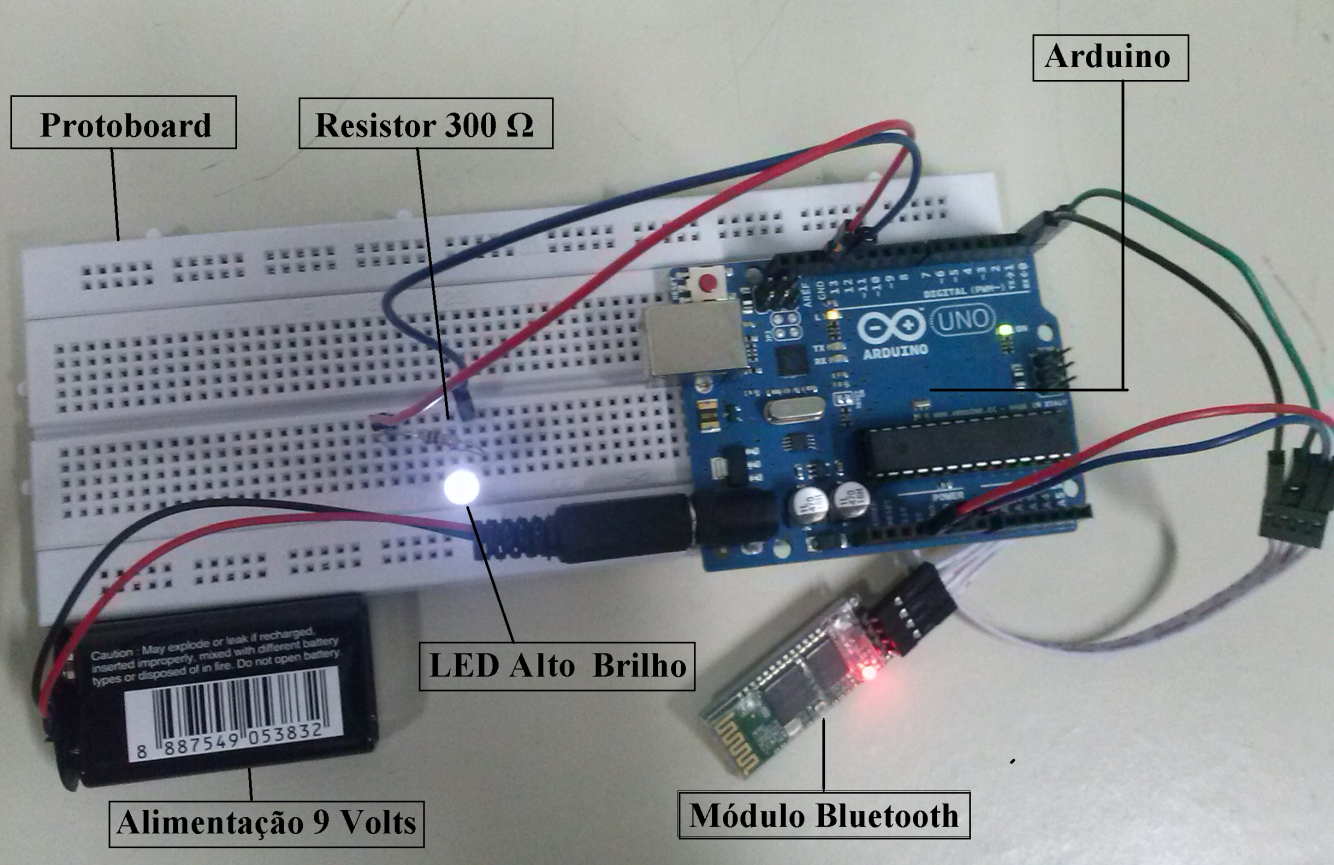
A estrutura do projeto segue os seguintes passos: Primeiro, é necessário ter um dispositivo com Sistema Operacional Android, segundo, é necessário um micro controlador Arduino, terceiro, é preciso utilizar um módulo Bluetooth conectado fisicamente ao Arduino para realizar a comunicação sem fio entre o Android e o Arduino. Por último os comandos enviados do Android para o Arduino via Bluetooth, serão traduzidos no micro controlador e repassados para os dispositivos conectados ao Arduino, conforme pode ser verificado observando a disposição dos componentes na figura 11.

Figura 11: Estrutura básica do projeto  
****

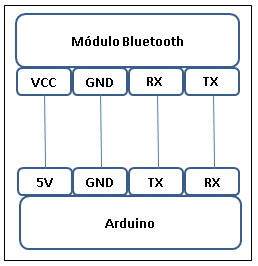
Fonte: Do Autor, (2013).

**3.7.1– Protótipo**

Para demostrar a conectividade entre o Android e o Arduino via Bluetooth, foi criado um protótipo aplicando os códigos acima descritos, que realiza a função de acender e apagar uma lâmpada, sendo simulado por um LED de alto brilho. O tipo de alimentação utilizado foi uma bateria de 9 Volts ligada através de um adaptador para a transferência de energia até o Arduino.

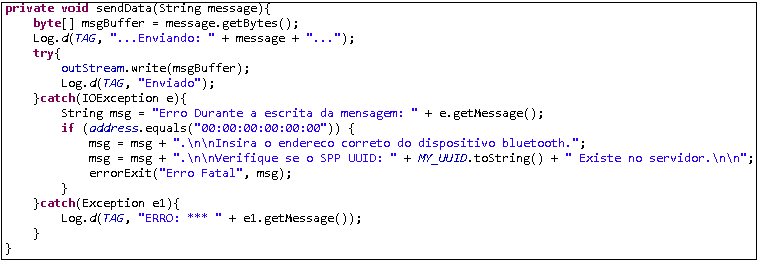
Figura 12: Protótipo  
****  
Fonte: Do Autor, (2013).

A conexão entre o módulo Bluetooth e o Arduino é necessária para que o Arduino possa receber e enviar os dados via tecnologia *wireless*, aumentando ainda mais sua utilização e viabilidade. Para realizar essa conexão basta conectar a entrada de energia do módulo Bluetooth (VCC) na porta de 5volts ou 3.3volts do Arduino, conectar o fio terra entre os dois dispositivos, e ligar os canais de comunicação entre eles, que no caso são as entradas RX e TX ligados inversamente, como demonstrado na figura 13.

Figura 13: Ligação do módulo Bluetooth no Arduino  
  
Fonte: Do Autor, (2013).

**3.7.2- Algoritmo Utilizado Para Envio de Informações**

Para o envio das informações para o Arduino é necessário criar um OutputStream e configurá-lo com o módulo Bluetooth, através de um socket. Para enviar essas informações via Bluetooth usa-se o método *Write* do OutputStream e deve-se converter a String recebida para bytes e a partir disso enviar a mensagem para o Arduino. Como pode ser visualizado na figura 14.

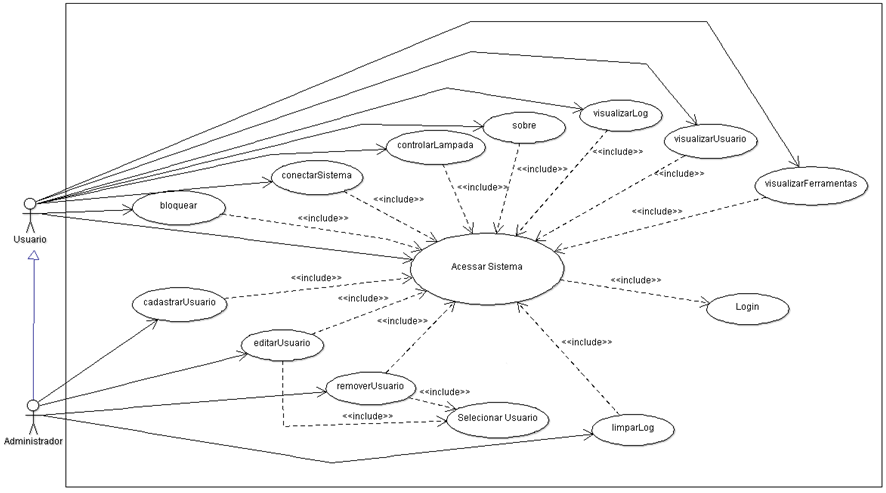
****Figura 14: Método para enviar as informações para o Arduino via Bluetooth

Fonte: Do Autor, 2013.

**3.7.3- Diagrama de Caso de Uso**

O diagrama de caso de uso conforme a figura 15, demostra que, existem dois tipos de atores no aplicativo, que são: O ator Usuário, não tem permissão de acesso a diversas áreas do aplicativo onde se devem tomar decisões como, por exemplo, não ter acesso à função de cadastrar, alterar e excluir um usuário. Já o ator Administrador dentro do aplicativo tem permissão total de utilização, onde o esse pode realizar qualquer tipo de ação dentro do sistema, como por exemplo, limpar todos os registros de log ou criar um novo usuário.

Figura 15: Diagrama de Caso de Uso

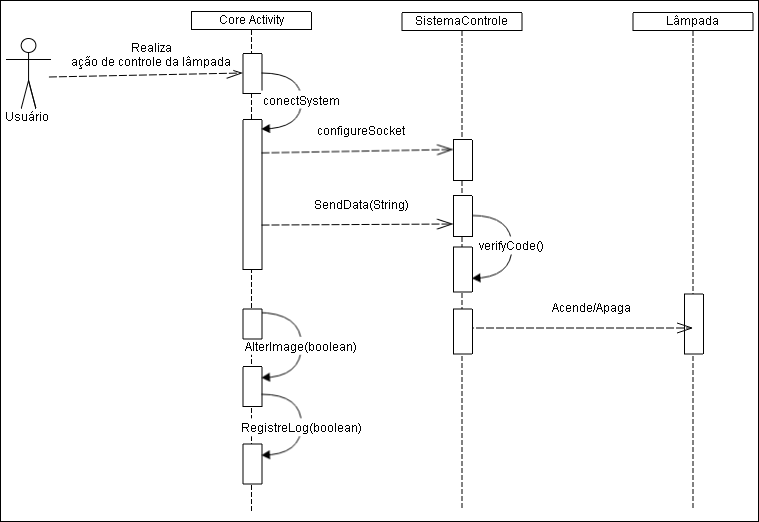


Fonte: Do Autor, (2013).

**3.7.4- Diagrama de sequência**

A principal função do aplicativo é controlar o acender e apagar de uma lâmpada, por esse motivo essa etapa está descrita na forma de um diagrama de sequência, onde dá-se da seguinte maneira, quando o usuário tocar no ícone da lâmpada conforme mostrado na figura 24, a classe responsável verificará se o aplicativo já está conectado com o sistema de controle, caso não esteja ela tentará realizar a conexão, caso consiga se conectar, será enviado uma mensagem via bluetooth para o sistema de controle onde esse verificará se o código é válido, sendo válido ele enviará o sinal para o módulo relé que realizará a ação de acender ou apagar a lâmpada. Após esses passos a classe Core irá alterar a imagem da lâmpada e a ação será registrada no log. Conforme pode ser verificado na figura 16.

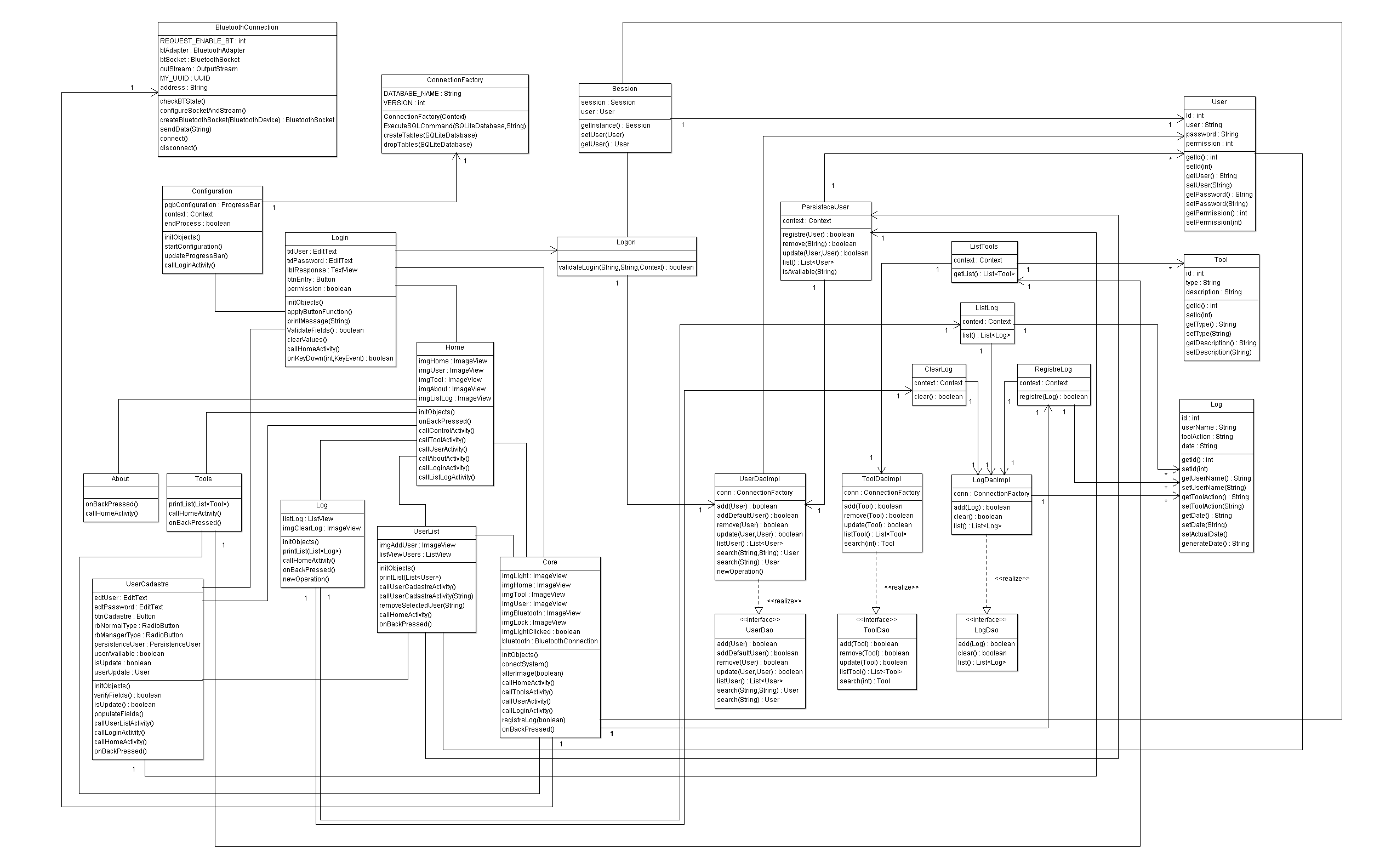
Figura 16: Diagrama de Sequencia da principal atividade do sistema

****

Fonte: Do Autor, (2013).

**3.7.4- Diagrama UML de Classe (Vide Apêndice A)**

Figura 17: Diagrama de Sequencia da principal atividade do sistema

****

Fonte: Do Autor, 2013.

**3.8- Estrutura Final do Sistema de Controle**

Após a criação de um protótipo com o led, foi desenvolvido um sistema mais próximo da realidade, utilizando uma lâmpada 220volts, um módulo relé, um módulo Bluetooth e um micro controlador Arduino, montados dentro de uma estrutura plástica, que pode ser visualizada na figura 18.

Também foi utilizado um led vermelho na parte externa da estrutura para e exibição de um status de recebimento de informações, ou seja, quando o Arduino recebe um dado, o led acende e apaga no intervalo de 200 milissegundos, essa funcionalidade foi acrescentada para a verificação de futuros erros, pois caso o dispositivo a ser controlado não realize a função desejada, através desse led pode-se filtrar o local do problema.

Figura 18: Estrutura Final do Sistema de Controle

Fonte: Do Autor, (2013).

**3.8.1- Interface com o Usuário**

Toda a interação com o usuário foi desenvolvida para ser o mais intuitivo possível, para facilitar a utilização do aplicativo e para que se tenha uma navegação mais agradável. Para a construção das telas foi utilizado a IDE Eclipse, junto com o SDK do Android para o Eclipse, pois com isso pode-se desenvolver de maneira rápida, estruturada e limpa, os códigos e as telas para o dispositivo Android.

**3.8.1.1- Tela Inicial**

A figura 19 mostra a primeira tela que é aberta ao se iniciar o aplicativo, nesta tela é feita a criação do banco de dados, das tabelas e dos registros padrão, tanto o usuário padrão do sistema, que futuramente poderá ser substituído, quanto à funcionalidade padrão, que é a de acender e apagar a lâmpada, porém isso só é realmente realizado quando não existir algum dos itens citados.

Figura 19: Tela inicial do sistema

****

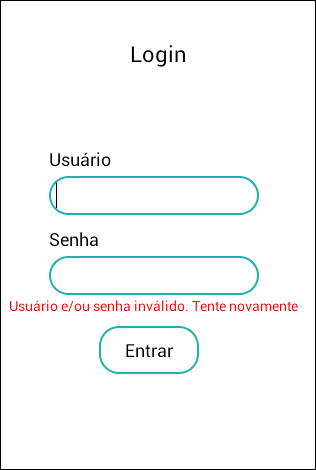
**3.8.1.2- Login**

A figura 20 exibe a tela de login, onde o usuário deverá preencher dois campos. O nome de usuário e senha, se os dados estiverem corretos, ele será direcionado para a tela de menu principal (figura 22), caso contrário o usuário permanecerá na tela de login sendo notificado sobre o erro nas informações digitadas.

Figura 20: Tela de login



A figura 21 mostra a mensagem que é informada ao usuário quando esse tenta entrar no sistema com informações incorretas, ou seja, se digitar um nome de usuário que não exista no sistema ou informar uma senha que não pertença ao nome de usuário digitado, o acesso ao sistema será negado e a mensagem “Usuário e/ou senha inválido. Tente novamente” será exibida.

****Figura 21: Tela de login inválido

**3.8.1.3- Menu Principal**

Na figura 22 vê-se a tela de menu principal, nessa parte do aplicativo o usuário determina o que quer fazer, entre cinco opções de navegação, as opções existentes no menu são: Abrir a tela de controle de iluminação (figura 23 e figura 24), visualizar os usuários atualmente cadastrados conforme a figura 25, listar as ferramentas do sistema (Figura 30), visualizar informações sobre o aplicativo (Figura 32) e também verificar o log das ações do software conforme figura 31.

Figura 22: Tela de Menu Principal

****

**3.8.1.4- Controle**

A figura 23 mostra a inicialização da tela de controle. Quando essa tela é iniciada, tenta-se realizar conexão com o sistema de controle, caso o aplicativo consiga se conectar então o ícone do Bluetooth ficará ativo e o usuário consegue tanto acender quanto apagar lâmpada acoplada ao sistema de controle, caso contrário o ícone permanece inativo e o usuário não pode realizar o controle do dispositivo.

Figura 23: Conexão ao sistema de controle

****

A figura 24 exibe a tela de controle da lâmpada. Nessa tela o usuário tem várias opções de navegação tais como, ao clicar no desenho do cadeado bloqueia-se a aplicação fazendo com que retorne para a tela de login. Caso selecione a figura do Bluetooth, o aplicativo se conecta ou desconecta do sistema de controle. Tocando na lâmpada será enviado sinal para o sistema de controle para realizar as ações de acender ou apagar a lâmpada. Ainda nesta tela, existe um resumo do menu principal de fácil acesso na parte inferior, onde o usuário pode navegar até o menu principal, ferramentas ou a visualização dos usuários.

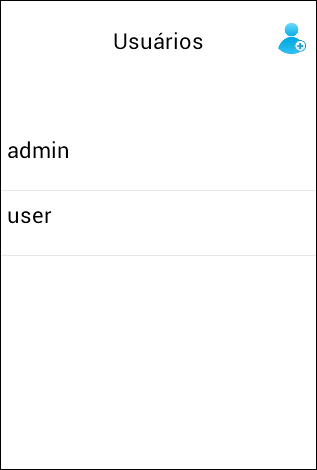
Figura 24: Opções de controle

****

**3.8.1.5- Usuários**

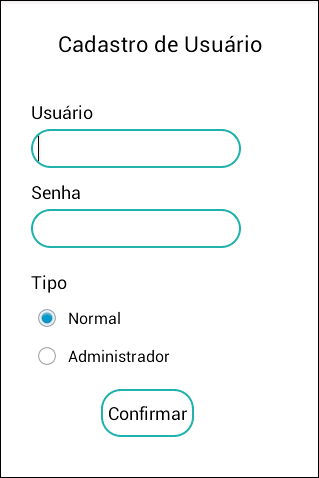
Na figura 25 visualiza-se a listagem de todos os usuários cadastrados atualmente no sistema, e caso a permissão de quem estiver logado seja de Administrador, poderá tanto adicionar novos usuários, quanto alterar e excluir o usuário selecionado.

Figura 25: Listagem dos usuários

****

A figura 26 exibe a tela de cadastro de usuário onde que para se cadastrar um usuário deve informar um nome que ainda não esteja cadastrado no sistema, uma senha e também selecionar um tipo de conta, que pode ser tanto Normal com limitação e diversas partes do aplicativo, quanto Administrador, que tem acesso total ao sistema.

Figura 26: Cadastro de usuário

****

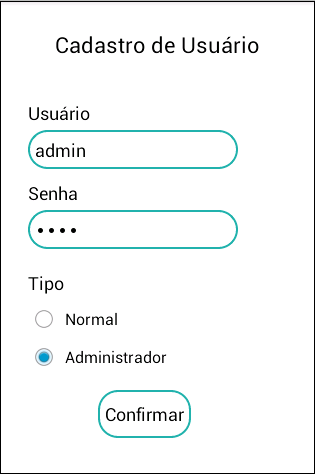
Na figura 27 é exibida a mensagem que aparece quando um usuário do tipo administrador mantem um toque pressionado sobre o nome de um usuário da lista. Nessa mensagem o usuário determina o que quer fazer, ou seja, ele poderá escolher entre alterar os dados ou excluir quem foi selecionado.

Figura 27: Opções da seleção



Ao selecionar a opção de alterar o cadastro de usuário, a tela de cadastro aparecerá preenchida com os dados, conforme a figura 28, a partir disso, basta alterar os dados e pressionar o botão confirmar. Caso o usuário alterado seja o mesmo que está logado no sistema, o sistema automaticamente se redirecionará para a tela de login. Caso contrário, voltará para a tela de visualização dos usuários.

Figura 28: Alteração de usuário

****

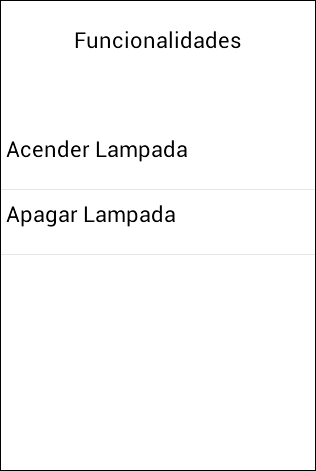
Ao escolher a opção de excluir o usuário, o aplicativo exibirá uma tela de confirmação, para que não ocorra exclusão indevida de acordo com a figura 29. Caso o usuário excluído seja o que está logado no sistema, o sistema automaticamente se redirecionará para a tela de login. Caso contrário, voltará para a tela de visualização dos usuários.

Figura 29: Confirmação de Exclusão  
****

**3.8.1.6- Ferramentas**

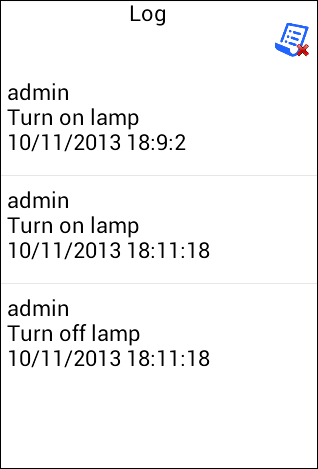
A figura 30 exibe a lista de todas as funcionalidades atualmente disponíveis no sistema, o que para esse trabalho se resume a acender e apagar lâmpada.

Figura 30: Lista das funcionalidades

****

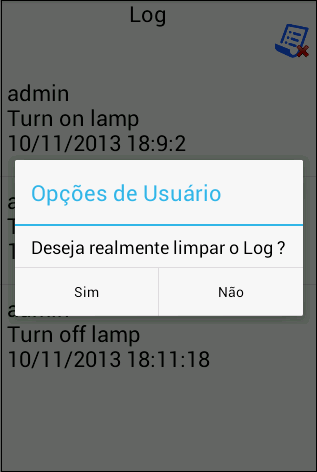
**3.8.1.7- Log**

A figura 31 exibe a lista de todas as atividades realizadas no aplicativo, que envolva as requisições feitas para o sistema de controle, através da tela de controle conforme visto na figura 24.

Figura 31: Visualização das atividades do sistema  
****

Na tela de visualização dos logs conforme exibido na figura 31, se tem a opção de excluir todo o log do sistema, simplesmente tocando na imagem exibida na tela. Ao pressionar o botão de exclusão do log, aparecerá a confirmação de exclusão para que não ocorra uma limpeza indesejada de acordo com a figura 32, após a confirmação a limpeza é executada e totalmente removida do banco de dados da aplicação.

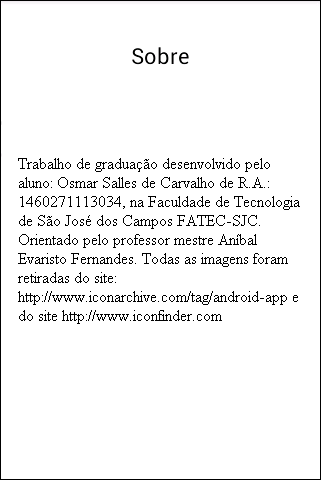
Figura 32: Exclusão do Log

****

**3.8.1.8- Sobre**

A figura 33 mostra a tela sobre do aplicativo, onde se pode visualizar quem é o autor do trabalho, quem é o orientador e de onde foram retiradas as imagens utilizadas para compor o aplicativo.

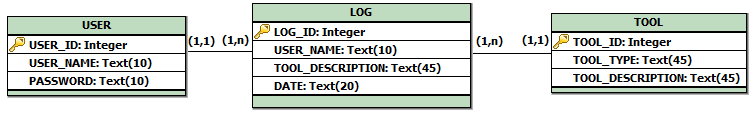
Figura 33: Informação sobre o aplicativo

****

**3.8.2-** Estrutura do Banco de Dados

**- M.E.R. (Modelo Entidade Relacionamento)**

Figura 33: Modelo Entidade Relacionamento do Aplicativo

****

Fonte: Do Autor, (2013).

**4- Resultados e Discussões**

**5- Conclusão**

A computação móvel é ainda uma área com uma capacidade enorme de aplicações...

Esse trabalho teve por objetivo realizar integração entre um Smartphone e um micro processador denominado Arduino...

Como recomendação futura....

**Referências**

ABRANET. Associação Brasileira de Internet. **Classe média digital**. Disponível em: http://www.abranet.org.br/index.php/noticias/108-classe-media-digital. Acesso em: 30/03/2013.

ABREU, E. R., VALIM, P. R. O. **Domótica: Controle de Automação Residencial Utilizando Celulares com Bluetooth.** Disponível em: [http://www.aedb.br/seget/ artigos11/16014124.pdf](http://www.aedb.br/seget/%20artigos11/16014124.pdf). Acesso em: 14/02/2013.

ALIEVI, C. A. **Automação Residencial Com Utilização de Controlador Lógico Programável.** Disponível em: [http://www.ebah.com.br/content/ABAAAAsK4AB/ automacao-residencial-com-utilizacao-controlador-logico-programavel](http://www.ebah.com.br/content/ABAAAAsK4AB/%20automacao-residencial-com-utilizacao-controlador-logico-programavel). Acesso em: 15/04/2013.

AN, L. **Android application controlled surge protector.** Disponível em: http://digitalcommons.calpoly.edu/eesp/113/. Acesso em: 15/03/2013.

ANDROID DEVELOPER. **Bluetooth**. Disponível em: http://developer.android.com/guide/ topics/connectivity/bluetooth.html. Acesso em: 15/05/2013.

ARDUINO. **What is Arduino?** Disponível em: http://www.arduino.cc/en/guide/introduction. Acesso em: 15/03/2013.

AURESIDE. Associação Brasileira de Automação Residencial. **Temas técnicos: Conceitos Básicos, Benefícios da Automação.** Disponível em: http://www.aureside. org.br/temastec/default.asp?file=concbasicos.asp. Acesso em: 28/02/2013.

BEAL, A. **Introdução à gestão de tecnologia da informação.** Disponível em: <http://www.atarp.com.br/novo/tiplanning/ti.pdf>. Acesso em: 30/03/2013.

BOLZANI, C. A.M. **Residências Inteligentes: Um curso de domótica**. Primeira Edição, São Paulo: Editora Livraria da Física, 2004.

BLUEMEL, P. C. P. **Concepção de interfaces de utilizador genéricas recorrendo ao uso da tecnologia rádio Bluetooth, 2009.** Disponível em: http://paginas.fe.up.pt/~ee82014/Dissertav2.pdf. Acesso em: 30/03/2013.

BURNETT, Ed. **Eclipse IDE Guia de bolso.** Edição traduzida, Porto Alegre: Editora O’Reilly, 2006. Disponível em: http://books.google.com.br/books?hl=pt-BR&lr=&id=J-pBNeOT\_W0C&oi=fnd&pg=PA22&dq=IDE+ECLIPSE&ots=LckQZg\_QPz&sig=SJBc7XC rnu2VuBdyfpflx71ookw#v=onepage&q&f=false. Acesse em: 10/06/2013.

EUZÉBIO, M. V. M. **Droidlar - Automação Residencial Através de Um Celular Android.** Disponível em: http://wiki.sj.ifsc.edu.br/wiki/images/b/b7/TCC\_Michel EusebioMello.pdf. Acesso em: 15/04/2013.

FUTURE ELECTRONICS. **What is Light Dependent Sensor (LDR) Tutorial**. Disponível em: http://www.fut-electronics.com/wp-content/plugins/fe\_downloads/Uploads/what%20isLi ght%20Dependent%20sensor%20(LDR)%20&%20Tutorial.pdf. Acesso em: 15/06/2013.

GUIA DE USUÁRIO DO MÓDULO BLUETOOTH. **BTM-5 Bluetooth Wireless TTL Master/Slave transceiver Module AT Command User Guide,** 2010. Disponível em: http://www.webtronico.com/documentos/BTM5\_AT\_COMMAND\_User\_Guide.pdf. Acesso em: 16/06/2013.

HASHIMI, S. Y e KOMATINEMI, S. **Pro Android**. Estados Unidos da America: Editora Apress, 2009.

JAYCONSYSTEMS. **Run a Sketch in Arduino IDE**. Disponível em: http://www.jayconsystems.com/tutorial/sketchardide. Acesso em: 15/10/2013.

KOBAYASHI, C. Y. **A tecnologia Bluetooth e aplicações, 2004.** Disponível em: http://grenoble.ime.usp.br/movel/monografia\_bluetooth.pdf. Acesso em: 30/03/2013.

KORJENIOSKI, M. **Desenvolvimento de jogos 2D com Android**, Disponível em: http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/490/1/CT\_JAVA\_VI\_2010\_14.PDF. Acesso em: 30/03/2012.

LECHETA, R.R.**Google** **Android aprenda a criar aplicações para dispositivos móveis com Android SDK.** Segunda edição, São Paulo: Novatec Editora, 2010.

MILLENNIAL MEDIA. **2012 Year in review: a comprehensive look back at mobile device trends**. Disponível em: http://www.millennialmedia.com/mobile-intelligence/mobile-mix/. Acesso em: 30/03/2013.

OLIVEIRA, A. C. S., SILVA, G. A. C. **Aplicabilidade dos sistemas de informação na relação de interação dos sistemas Android e Arduino.** Trabalho de conclusão de curso (ciência da computação), Centro Universitário Salesiano de São Paulo - U.E. Lorena, 2012.

OXER, J. e BLEMINGS, H. **Practical Arduino: Cool Projects for Open Source Hardware.** Estados Unidos da America: Editora Apress, 2009.

RABELLO R. R. **Android: Um novo paradigma de desenvolvimento móvel.** Disponível em: http://www.cesar.org.br/site/files/file/WM18\_Android.pdf. Acesso em: 28/03/2013.

SENA, D. C. S. **Automação Residencial.** Disponível em: http://www2.ele.ufes.br/~ projgrad/documentos/PG2005\_2/dianecristinasouzasena.pdf Acesso em: 15/03/2013.

SERIAL LINK. **Manual Técnico Módulo de Relê 4 Canais.** Disponível em: http://www.seriallink.com.br/lab/Arduino/Datasheet\_Modulo\_Rele\_4\_Canais\_SerialLink.pdf. Acesso em: 15/06/2013.

SQLITE. **About SQLite.** Disponível em: http://www.sqlite.org/about.html. Acesso em: 10/06/2013.

## STRATEGY ANALYTICS. Handset Sales Forecast by Type: Phablet, Superphone, Smartphone, Feature Phone & Basic Phone. Disponível em: <http://www.strategyanalytics.com/default.aspx?mod=reportabstractviewer&a0=7836.Acesso> em: 30/03/2013.

TANENBAUM, A. S. **Redes de Computadores**.Rio de Janeiro: Elsevier Editora Ltda., 2003.

ZIGBEE ALLIANCE. **Zigbee specification**. Disponível em: http://www3.nd. edu/~mhaenggi/ee67011/zigbee.pdf. Acesso em: 14/04/2013.