UNIVERSIDADE ESTADUAL DO MARANHÃO
CENTRO DE CIÊNCIAS TECNOLÓGICAS
CURSO DE ENGENHARIA DE COMPUTAÇÃO

AUTOMAÇÃO RESIDENCIAL INTEGRANDO ARDUINO, ANDROID E COMUNICAÇÃO SEM FIO

AUTOMAÇÃO RESIDENCIAL

por

ERYCK DE ARAUJO OLIVEIRA

Professor Dr. Carlos Henrique Rodrigues de Oliveira
Orientador

São Luís (MA), 02 de dezembro de 2016

ERYCK DE ARAUJO OLIVEIRA

AUTOMAÇÃO RESIDENCIAL INTEGRANDO ARDUINO, ANDROID E COMUNICAÇÃO SEM FIO

AUTOMAÇÃO RESIDENCIAL

Trabalho de Conclusão de Curso apresentado à Banca Examinadora do Curso de Engenharia de Computação, da Universidade Estadual do Maranhão, como pré-requisito para a obtenção de título de Bacharel em Engenharia de Computação.

Orientador: Professor Dr. Carlos Henrique Rodrigues de Oliveira

São Luís (MA), 02 de dezembro de 2016

Oliveira, Eryck de Araújo.
Automação residencial integrando arduino, android e comunicação sem fio / Eryck de Araújo Oliveira. – São Luís, 2016.
54 f.

Monografia (Graduação) – Curso de Engenharia de Computação, Universidade Estadual do Maranhão, 2016.

Orientador: Prof. Carlos Henrique Rodrigues de Oliveira.

1. Automação residencial. 2. Arduino. 3. Redes sem fio. 4. Android
I. Título.

CDU 004.42: 681.5

ERYCK DE ARAUJO OLIVEIRA

AUTOMAÇÃO RESIDENCIAL INTEGRANDO ARDUINO, ANDROID E COMUNICAÇÃO SEM FIO

Trabalho de Conclusão de Curso apresentado à Banca Examinadora do Curso de Engenharia de Computação, da Universidade Estadual do Maranhão, como pré-requisito para a obtenção de título de Bacharel em Engenharia de Computação.

BANCA EXAMIDADORA:

_____

Prof. Dr. Carlos Henrique Rodrigues de Oliveira

Orientador

Universidade Estadual do Maranhão

_____

Prof. Dr. Leonardo Henrique Gonsioroski Furtado da Silva

Universidade Estadual do Maranhão

_____

Prof. Dr. Rogerio Moreira Lima Silva

Universidade Estadual do Maranhão

São Luís (MA), 02 de dezembro de 2016

AGRADECIMENTOS

Queria agradecer primeiramente a Deus pelo dom da vida, pela saúde diária e por suas bênçãos durante toda essa jornada, por sempre me dar forças nos momentos difíceis não deixando desistir jamais e por conduzir minha vida aos melhores caminhos.

Sou imensamente grato aos meus pais Antônio Cavalcante de Oliveira e Maria Edileusa de Araujo Oliveira pelo imenso esforço e sacrifício para nos dar oportunidade de poder tentar uma vida melhor longe de casa, me possibilitando a educação e os ensinamentos passados que me tornaram uma pessoa melhor, pelo amor e carinho demonstrados durante toda a vida e por mostrar os caminhos da verdade, honestidade e humildade.

Aos meus irmãos Emerson Antônio de Araujo Oliveira e Elysson José Araujo de Oliveira, que sempre estiveram comigo durante essa jornada no meio acadêmico, se tornando peças essenciais nos momentos difíceis, me dando apoio e sempre disponíveis dando conselhos, mostrando companheirismo e afeto.

Agradeço aos meus avós Raimundo Silva (In Memória) e Maria Silva, que sempre me incentivaram a continuar no meio acadêmico, pelo exemplo de vida e de pessoas que são, por todo o amor e dedicação característicos durante toda a vida, essa conquista dedico a vocês.

Agradeço a todos os amigos e colegas de curso conquistados durante essa jornada em especial à turma de 2011.1. Agradeço imensamente aos amigos Robson Monteiro, Rayanne Salles, Rodrigo Oliveira, Richardson, Magno e Luiz Ricardo que durante toda essa jornada acadêmica estiveram do meu lado passando pelas maiores dificuldades, ajudando sempre a vencer, compartilhando conhecimento e sendo companheiros.

Por fim agradeço à UEMA e ao corpo docente do curso de Engenharia de Computação, pela oportunidade de aqui me desenvolver pessoal e profissionalmente, agradeço pela educação, conselhos passados e pela dedicação durante esses anos. Em especial ao meu orientador Professor Carlos Henrique Rodrigues pelo empenho em tornar este projeto possível, pela dedicação, pela cobrança de excelência em tudo que faz, e pelos conselhos que levarei para a vida.

“Que os vossos esforços desafiem as impossibilidades, lembrai-vos de que as grandes coisas do homem foram conquistadas do que parecia impossível.”

(Charles Chaplin)

LISTA DE ACRÔNIMOS

ADC	Conversor Analógico/digital
DAC	Conversor Digital/Analógico
EEPROM	*Electrically-Erasable Programmable Read-Only Memory*
HTTP	Protocolo de Transferência de Hipertexto (*Hypertext Transfer Protocol*)
IDE	Ambiente de Desenvolvimento Integrado (*Integrated Development Environment*)
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos (*Institute of Electrical and Electronics Engineer*)
IOT	Internet das Coisas (*Internet of Things*)
LAN	Rede Local (*Local Area Network*)
LED	Diodo Emissor de Luz (*Light Emitting Diode*)
MAN	Rede Metropolitana (*Metropolitan Area Network*)
PAN	Rede Pessoal (*Personal Area Network*)
SO	Sistema Operacional
TCP	Protocolo de Controle de Transmissão (*Transmission Control Protocol*)
UDP	Protocolo de Datagrama do Usuário (*User Datagram Protoco*)
UEMA	Universidade Estadual do Maranhão
USB	*Universal Serial Bus*
WAN	Rede Geograficamente distribuída (*Wide Area Network*)

LISTA DE FIGURAS

LISTA DE TABELAS

RESUMO

O conforto e a facilidade na execução de tarefas simples do cotidiano está sendo cada vez mais necessário a qualquer sistema, muito pela relação "custo x tempo" exigida hoje em dia, e como forma de prover acessibilidade e segurança para as pessoas independente de suas limitações. O emprego de ferramentas que facilitam a interação dos sistemas com o usuário se torna um dos fatores primordiais para sua aplicação e elaboração. Assim a proposta deste trabalho é apresentar uma aplicação de Automação Residencial utilizando a plataforma de prototipagem eletrônica Arduino através do uso de redes de comunicação sem fio para acesso remoto, integrados ao aplicativo para *Smartphones* com Android desenvolvido em linguagem JAVA para a interface com o usuário, sendo este de fácil manipulação e familiaridade. Como resultados, o protótipo será capaz de fazer o acionamento de várias aplicações de uma residência por meio de comandos feitos por um dispositivo móvel remotamente, acionando o controlador Arduino que por meio dos sensores e atuadores realiza as tarefas desejadas. O produto final deste trabalho será uma tecnologia de baixo custo, capaz de ampliações posteriores simples por ser modular, oferecendo segurança, comodidade, facilidade de acesso e economia para seus usuários.

Palavras-chave: Automação Residencial. Arduino. Redes Sem Fio. Android. Acessibilidade.

ABSTRACT

The comfort and ease in performing simple daily tasks is increasingly necessary in any system, much the ratio "cost-time" required today and as a way to provide access and security for the people regardless of their limitations. The use of tools that facilitate the interaction of the system with the user becomes one of the key factors for its application and development. Therefore, the purpose of this paper is to present an application of Home Automation using electronic prototyping Arduino platform using wireless communication networks for remote access, integrated into Smartphones Android application developed in Java language for the user interface, which is easy handling and familiarity. As a result, the prototype will be able to drive the various applications in a residence via commands made by a remotely mobile, triggering the Arduino controller by means of sensors and actuators performs the desired tasks. The final product of this work will be a low-cost technology capable of simple later expansion for its modularity, offering security, convenience, ease of access and savings for its users.

Keywords: *Home Automation. Arduino. Wireless Networks. Android. Accessibility.*

SUMÁRIO

1. INTRODUÇÃO

Com a necessidade de se obter sempre a melhor forma de lidar com todas as tarefas e trabalhos do dia-a-dia, e para uma melhor forma de interagir com o constante desenvolvimento eletrônico para o seu conforto e comodidade, o ser humano vem se aprimorando nos métodos e práticas relacionados ao bem-estar pessoal e à acessibilidade nos ambientes. Dentro desse escopo podemos citar a Automação Residencial que visa sempre trazer o conforto, comodidade, acessibilidade e segurança para o usuário permitindo controlar a residência remotamente, poupando tempo com tarefas repetitivas, economizando recursos (MOREIRA, 2013). Tendo atualmente uma posição de destaque no mercado mundial.

Partindo da necessidade de melhorar a qualidade de vida, utilização mais eficiente dos recursos energéticos, conforto e segurança dos usuários, podemos fazer o uso de sistemas microcontrolados, mais especificamente de sistemas conhecidos como sistemas embarcados, para tal finalidade. A grande vantagem dos sistemas embarcados é que são sistemas desenvolvidos para uma aplicação específica e que ao serem projetados podem atuar sem a interferência de qualquer controle externo, destacam-se pelo baixo custo, pequena manutenção e robustez. Outra característica que facilita o desenvolvimento e integração desses sistemas é o seu uso associado aos dispositivos móveis, fato que possibilita o uso dos *softwares* livres que trazem consigo o dinamismo no desenvolvimento e também se tornam uma ferramenta de custo acessível. Assim o desenvolvimento de um protótipo de controle dos sistemas internos em uma residencial se torna algo trivial e de fácil aplicação.

Assim neste trabalho apresenta-se a construção de um protótipo de baixo custo para a aplicação dos conceitos de Automação Residencial com o objetivo de controlar e monitorar as variáveis de iluminação de uma residência, além dos módulos de temperatura, acionamento de portões e controle de tomadas da mesma.

Buscando-se de uma forma simples fazer a interação de sensores e atuadores instalados no protótipo, com o uso de um sistema de controle utilizando uma lógica sequencial, e com uma comunicação serial entre a plataforma Arduino e o dispositivo de monitoramento (Computador, celular, por exemplo). Para que essa interação seja possível, há a produção dos *softwares* necessários ao projeto, o primeiro estabelece uma interface entre o usuário e os outros dois *softwares* (O Aplicativo "App" instalado no celular do usuário) programado em Linguagem *JAVA* e hospedado no Sistema Operacional Android, amplamente difundido

atualmente entre os usuários. O segundo *software* permite que o comando enviado pelo celular, por meio da tecnologia Wi-fi de redes sem fio ou rede de dados móveis, seja transmitido através de requisições socket para o *software* instalado no Arduino estes comandos são enviados pela porta *ethernet* do roteador ao módulo *ethernet shield* integrado no Arduino, que recebe esses dados e faz parte do processamento como programado previamente. O terceiro *software*, hospedado no Arduino por meio da porta serial de programação no Ambiente de Desenvolvimento Integrado (IDE) próprio, recebe os dados emitidos pela porta *ethernet* faz a leitura e processamento de qual atuador acionar e executa a ação de acordo com o dado recebido. Neste momento os sensores e atuadores aplicam os comandos.

1.1. OBJETIVOS

1.1.1. Objetivo Geral

- Apresentar a construção de um protótipo de sistema de automação residencial com controle de dispositivos de um ambiente usando a plataforma de prototipagem eletrônica Arduino em conjunto com um aplicativo Android em um *Smartphone* capaz de comunicar-se remotamente com os atuadores responsáveis pelo controle dos dispositivos através de redes de comunicação sem fio local ou remotamente;

1.1.2. Objetivos Específicos

- Realizar uma revisão bibliográfica de conceitos sobre a Automação Residencial em conjunto com as ferramentas necessárias ao cumprimento do projeto;

- Prover a construção de um firmware que possibilite à placa de prototipagem Arduino controlar os módulos dos dispositivos e comunicar-se com o dispositivo móvel utilizando uma rede de comunicação sem fio pela placa *ethernet* e rede Wi-fi;

- Desenvolver os *softwares* necessários à aplicação Android no smartphone, bem como os *softwares* de controle de rede sem fio e à integração dos dispositivos do sistema;

- Avaliar a relação custo benefício envolvida com a instalação do projeto.

1.2. METODOLOGIA

O presente trabalho está dividido em duas etapas principais, que são:

- 1ª etapa: pesquisa e fundamentação teórica para desenvolvimento do sistema;
- 2ª etapa: montagem, teste e validação do protótipo contendo a maquete física de representação do local, integrados com o *software* Android no Smartphone e com os sensores e atuadores controlados pelo Arduino.

Nesta primeira etapa realizou-se uma abordagem do problema, justificando-se a necessidade do desenvolvimento do protótipo proposto, pois gerar uma solução de baixo custo em beneficio ao usuário se torna essencial nesse mercado. Fez-se uma revisão de conceitos pertinentes à resolução do problema, e apresentando-se algumas técnicas já existentes para a solução desse projeto. Esta etapa abrange pontos como conceitos de automação e pré-automação residencial, bem como as suas atuais situações em termos de custo no Brasil para que, a partir do conhecimento destes tópicos iniciais o protótipo possa ser desenvolvido de maneira viável e eficiente. Assim as melhores técnicas foram selecionadas e aplicadas ao escopo do projeto.

Buscou-se fundamentação e aprofundamento sobre as tecnologias envolvidas em projetos de automação residencial, como a integração de microcontroladores com os sensores e atuadores ideias para a solução apresentada, também sobre os *softwares* utilizados no projeto para a comunicação dos mesmos.

Na segunda etapa temos o protótipo do sistema de automação propriamente dito, compreendendo três etapas básicas: comunicação, componentes físicos e *Softwares*. A comunicação se dá através de redes de comunicação sem fio Wi-fi e rede de dados pelo celular bem como a comunicação USB para o Arduino com o computador que se hospeda o *software*. A parte dos componentes físicos pode ser dividida em partes menores, sejam elas: lâmpadas, relês, diodos, resistores, LED's, microcontrolador, notebook e *Smartphone*. A parte de *software* será subdividida em dois ambientes de desenvolvimento, sendo eles: *software* Arduino e *software* Android Studio, onde no *software* Arduino desenvolveu-se toda a sistemática para a aplicação de controle e automação dos dispositivos através do IDE próprio da plataforma, já o *software* Android foi desenvolvido em linguagem *JAVA* por meio do IDE Android *Studio 2.0*.

Na etapa seguinte, mostram-se os testes no protótipo de forma a garantir o correto funcionamento, a confiabilidade e também a durabilidade do sistema de automação.

Finalmente, são apresentados dados tais como: facilidade da instalação, custos e viabilidade econômica, manutenção, durabilidade e benefícios ao usuário.

1.3. ESTRUTURA DO TRABALHO

O trabalho está estruturado, em seu primeiro capítulo, com uma introdução sobre o tema, seguido do segundo capítulo com uma fundamentação sobre microcontroladores, automação residencial: soluções e padrões já estabelecidos, bem como conceitos pertinentes à tecnologia de redes sem fio essenciais ao escopo do trabalho.

Em seu terceiro capítulo, é feita a descrição detalhada do modelo proposto para solução final do projeto. Feito isso, é apresentada a descrição de todos os componentes necessários ao protótipo e como se dá seu devido funcionamento.

No quarto capítulo, são descritas as constatações pertinentes aos ensaios, montagem e funcionamento do protótipo bem como os códigos desenvolvidos para as aplicações no Arduino e no Aplicativo para os Smartphones. Esse capítulo inclui dados, tabelas, fotos e demais materiais obtidos no ato do desenvolvimento das aplicações.

No penúltimo capítulo, são analisados, testados e obtidos os resultados pertinentes à aplicação do protótipo, incluindo dados, tabelas e fotos dos testes realizados e seus resultados.

Ao último capítulo, cabe apresentar as conclusões e as soluções obtidas para os objetivos traçados com o funcionamento do sistema de automação residencial. É constatada a viabilidade ou não do sistema proposto.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados conceitos teóricos pertinentes ao andamento do projeto, tais como conceitos de Automação, Automação Residencial, e também explorando a versatilidade dos Microcontroladores em geral essenciais aos projetos de protótipos de automação, bem como redes de computadores que fazem parte do sistema de comunicação do projeto.

2.1. AUTOMAÇÃO RESIDENCIAL

Segundo (PEREIRA, 2013), "a automação trata de sistemas desenvolvidos para execução automática de atividades repetitivas ou quando da ocorrência de determinados eventos". Já segundo (BOLZANI, 2004) é o conceito de tornar atividades repetitivas em automáticas, utilizando dispositivos que coletem dados e atuem nos processos, diminuindo a necessidade da interferência humana, resultando em maior velocidade nas operações, redução de erros e fidelidade de informações, que são elementos cruciais para melhor administração.

Seguindo esses conceitos pode-se afirmar que a automação residencial tem o foco centralizado em automatizar tarefas domésticas em função do conforto, comodidade e segurança de seu usuário, provendo sistemas automatizados por vezes inteligentes e pré-programados para facilitar a aplicação dessas tarefas.

Outro ponto importante dessa tecnologia é a integração facilitada do usuário com esses sistemas, através de comandos simples por uma central de comando, por botões de acionamento ou mesmo por dispositivos móveis dependendo do nível da aplicação dessa automação, pois pode-se obter essa automação por sistemas autônomos (sem ligação e comunicação entre si), por meio de sistemas integrados (utilizando um controlador central, ainda assim sistemas funcionando unicamente) e por sistemas complexos (sistema gerenciador ao invés de só controlador, torna-se uma comunicação dupla entre sistema e usuário, compartilhando-se ações e dados, sistema personalizável) (ALVAREZ; ANTUNES, 2015).

Atualmente, a automação residencial mostra-se extremamente útil quando se refere à necessidade de segurança, conforto, praticidade, economia e valorização do imóvel. Com a evolução dos sistemas e com o advento dos microcontroladores e sistemas embarcados a automação residencial vem se desenvolvendo de modo que os custos com a instalação dos

módulos (como são chamados os dispositivos responsáveis por automatizarem os sistemas residenciais) vem decrescendo a cada dia com as tecnologias descentralizadas, de mais fácil operação e com isso a acessibilidade dos módulos de automação vem se popularizando. Tais fatores são determinantes para a maior difusão desta tecnologia, pois facilitam ao usuário adquirir os módulos sem necessitar de mão de obra especializada para garantir o funcionamento do sistema de automação.

A Automação Residencial é um mercado em crescente expansão onde esse mercado relacionado a Internet das Coisas (Iot – *Internet of Things*) chega a movimentar cerca de US$ 4,1 bilhões este ano segundo a IDC (IDC BRASIL, 2016). A Associação Brasileira de Automação Residencial (Aureside) mostra em levantamento que cerca de 2 milhões de residências brasileiras já teriam potencial para utilizar sistemas automatizados, mas o número de usuários de fato não chega a 20% desse total (AURESIDE, 2016).

Os conceitos de automação residencial, automação predial e domótica estão muito ligados entre si, gerando uma certa confusão em sua delimitação individual. Apesar da semelhança a diferença entre elas situa-se no foco e aplicação dos sistemas, ou seja, a automação residencial é aplicada a uma só residência e automação predial aplicada a espaços comuns como condomínios e prédios. A automação residencial e predial constitui-se por um ou mais dispositivos atuando singularmente sem qualquer comunicação entre os mesmos, já a domótica descreve a integração entre todos os dispositivos fazendo com que eles atuem em conjunto para uma determinada função especificada no projeto (GOMES, 2010, p.5).

2.2. MICROCONTROLADORES

Os microcontroladores surgiram de uma evolução natural dos circuitos digitais e do meio tecnológico em si, pois a demanda de complexidade das aplicações especificas unida com a necessidade de supressão de espaços físicos das instalações elétricas que levou ao desenvolvimento de circuitos digitais com uma grande gama de transistores em um único chip, levando a uma economia e facilidade na instalação desses circuitos nos produtos dando início aos primeiros microcontroladores. Com as aplicações demandando cálculos mais complexos e funções com padrões específicos, eles se inseriram rapidamente no mercado e tiveram seu aperfeiçoamento ao longo dos anos, pois permitem a elaboração relativamente rápida e fácil de novos equipamentos embarcados, devido a sua facilidade de uso em ampla faixa de aplicações e a sua compatibilidade com os componentes necessários aos projetos.

Eles incorporam em um único encapsulamento um microprocessador com função específica de acordo com o projeto requerido, periféricos (i.e. temporizadores, comunicação serial, conversores analógico/digital e moduladores de largura de pulso PWM) e memória de programa onde se definem os objetivos do sistema por meio da programação do dispositivo e dados que ficam alocados no periférico e são usados durante o processamento.

Sendo eles microprocessadores que podem ser programados para funções específicas, em contraste com outros microprocessadores de propósito geral (como os utilizados nos PCs), os microcontroladores, além dos componentes lógicos e aritméticos usuais de um microprocessador de uso geral, integram elementos adicionais em sua estrutura interna, como memória de leitura e escrita para armazenamento de dados, memória somente de leitura para armazenamento de programas, EEPROM para armazenamento permanente de dados, dispositivos periféricos como conversores analógico/digitais (ADC), conversores digitais/analógicos (DAC) e interfaces de entrada e saída de dados (I/O) (USATEGUI & MARTÍNEZ, 1999).

Outra característica marcante e importante dos microcontroladores é sua economia de recursos em aplicações específicas como automação residencial, predial e industrial. São amplamente difundidos nesses ambientes, pois com sua aplicação em sistemas embarcados que demandam alta complexidade e interação constante com uma variedade enorme de periféricos contando com o menor uso possível dos recursos disponíveis, estes são muito exigidos e correspondem com seus *softwares* programados especificamente para a aplicação desejada.

Dentre os dispositivos destaques no mercado atual pode-se citar o Arduino placa de prototipagem de código aberto, conta com o microcontrolador Atmel AVR, entradas e saídas digitais e analógicas, tem uma gama enorme de sensores, Shields (placas independentes que fornecem funções diferentes e usuais para os projetos mais complexos), atuadores. Conta com uma programação de fácil entendimento e intuitiva. Podendo atender às mais complexas requisições de *software* e *hardware* atendendo as necessidades de projetos, além de manter os projetos com baixo custo, acessíveis e flexíveis a mudanças.

2.3. REDES DE COMPUTADORES

As redes de comunicação e troca de dados existem para que dados possam ser enviados de um lado ao outro sem que se tenha perda e que garantam a rapidez de envio e recebimento

desse dado. A tempos atrás esses conceitos eram pensamentos utópicos e difíceis de serem alcançados, com o desenvolvimento e aperfeiçoamento das tecnologias, hoje os dados são enviados e trocados de forma extremamente rápida e dentre locais imensamente distantes, facilitando e embasando negócios, trazendo mais e melhores informações de forma instantânea para a população, fortalecendo empresas, etc.

Para que haja essa comunicação dos dados estes precisam estar navegando em uma rede de computadores, que é segundo (FOROUZAN, 2006) um conjunto de dispositivos conectados por meio de *links* de comunicação podendo enviar e/ou receber dados gerados por outros nós dentro da rede, trocando informações de diversos tipos.

As redes de computadores surgem então por volta de 1950, na segunda geração dos sistemas computacionais caracterizadas pelo uso dos terminais "burros" e de lenta transmissão de dados, daí então as redes foram ganhando poder computacional juntamente com o desenvolvimento dos computadores, surgindo novos protocolos, tecnologias de transmissão avançadas e criptografia de dados.

Dentre os itens em que se podem classificar uma rede de computadores destacam-se duas dimensões principais: o tipo de tecnologia de transmissão de dados e a escala utilizada. Segundo (TANENBAUM, 2003) há dois tipos de tecnologias de transmissão de dados em uso disseminados: redes de difusão (*broadcast*) e redes ponto-a-ponto. Em redes de difusão todas as máquinas compartilham um único canal de comunicação por meio de pacotes endereçados, ou seja, o pacote é enviado por uma máquina e recebido por todas as máquinas na rede, mas somente é interpretado por aquela em que foi endereçado. Já nas redes ponto-a-ponto as máquinas são conectadas por pares individuais com outras máquinas, assim o pacote enviado será transmitido para o receptor final obedecendo a um caminho previamente estabelecido, necessitando do roteamento na rede para que possa atingir seu destino final, nestes casos, o primeiro receptor de uma mensagem pode não ser o seu destinatário, mas apenas um ponto de passagem da mensagem.

Em relação a escala, as redes são classificadas de acordo com a Tabela 1. Onde as redes pessoais (PAN – *Personal Area Network*) são redes de uso pessoal, normalmente utilizadas por uma pessoa só, por exemplo no uso de dispositivos móveis conectados por *bluetooth*, mouses e teclados sem fio, etc.

Tabela 1 - Classificação de redes em escala

Fonte: (TANENBAUM, 2003)

Distância entre processadores	Processadores no mesmo local	Exemplo
1 m	Metro quadrado	Rede Pessoal (PAN)
10 m	Sala	Rede Local (LAN)
100 m	Edifício	
1 km	Campus	
10 km	Cidade	Rede Metropolitana (MAN)
100 km	País	Rede geograficamente distribuída (WAN)
1.000 km	Continente	
10.000 km	Planeta	A Internet

As redes locais (LAN – *Local Area Network*) são as mais populares em uso atualmente, são redes privadas normalmente ocupando um edifício ou uma área de um campus com poucos quilômetros de extensão, normalmente são redes cabeadas para o uso dos computadores e também redes mistas (sem fio e cabeada) com o advento dos dispositivos móveis de acesso a rede difundidos popularmente no cenário atual. Nas LAN's os padrões mais utilizados são os IEEE 802.3 e o 802.11 que correspondem ao padrão *ethernet* e ao padrão Wi-fi respectivamente.

As redes metropolitanas (MAN – *Metropolitan Area Network*) são redes de maior alcance e usam dos preceitos básicos das LAN's, conectam várias redes em distâncias de dezenas a centenas de quilômetros. E as redes de longo alcance (WAN – *Wide Area Network*) atingem distâncias de continentes, nelas competem os maiores fluxos de dados e são normalmente de propriedade pública ou de operadoras de telecomunicações. Os usuários finais ficam conectados dentro de uma sub-rede de comunicação comandados pelos provedores de acesso à rede.

2.3.1. Camadas de rede

Para reduzir a complexidade do projeto, a maioria das redes é organizada como uma pilha de camadas ou níveis, colocadas umas sobre as outras diferenciando-se ou não de uma rede para a outra (TANENBAUM, 2003). Cada camada oferece serviços distintos à camada superior dentro de sua classificação inicial para haver a comunicação. A **camada n** em uma máquina, para desempenhar suas funções estabelece uma conversação com a **camada n** em

outra máquina. As regras e convenções usadas na comunicação entre camadas de mesmo nível são conhecidas como protocolo, são eles que estabelecem a correta comunicação entre as camadas.

Quando se agrupam um conjunto de camadas e protocolos de uma rede, pode-se dizer que temos ali uma arquitetura de rede. Assim ao longo da história das redes dois modelos de referência para as arquiteturas foram destaque: o modelo OSI e o modelo TCP/IP.

2.3.2. Modelo ISO RM-OSI

Esse modelo se baseia em uma proposta desenvolvida pela *International Standards Organization* (ISO) como um primeiro passo em direção à padronização internacional dos protocolos empregados nas diversas camadas (TANENBAUM, 2003). Chamado de Modelo de Referência para Interconexão de Sistemas Abertos (RM-OSI – *Reference Model for Open Systems Interconnection*), trata da interconexão de sistemas abertos à comunicação, não pode ser considerado uma arquitetura de rede, pois ele não define exatamente os serviços e protocolos usados em cada camada, somente informa o que cada camada deve fazer e não como fazer.

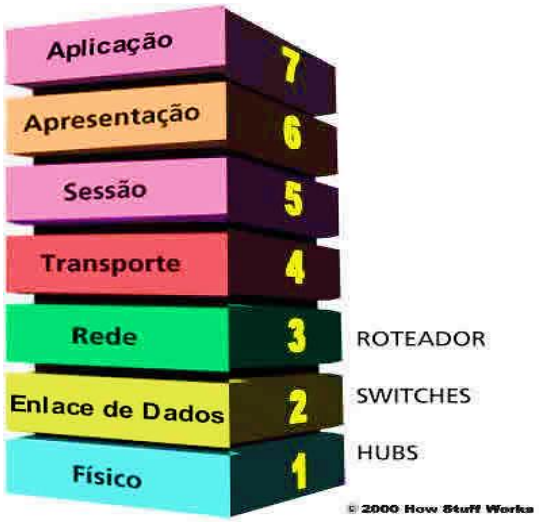

Figura 1 - Modelo OSI
Fonte: (ALENCAR, 2010)

O modelo foi dividido em sete camadas distintas como na Figura 1, que operam na forma de encapsulamento dos dados, ou seja, em cada camada o dado será acrescido de um cabeçalho, posteriormente envelopados, assim as informações passam para a camada superior onde será acrescido de um novo cabeçalho e novamente encapsulado, assim sucessivamente.

A Figura 2 mostra um resumo das tarefas realizadas por cada uma das camadas do modelo OSI, mostrando sua versatilidade e a importância que este modelo trouxe para a padronização e o correto uso das redes de computadores, já que as camadas são bem definidas e os possíveis serviços relacionados a elas bem expostos.

Camada	Descrição
Físico	Esta camada pega os quadros enviados pela camada de enlace e os transforma em sinais compatíveis com o meio por onde os dados deverão ser transmitidos.
Enlace de Dados	A camada de enlace pega os pacotes de dados recebidos da camada de rede e os transforma em quadros que trafegarão pela rede, adicionando informações como o endereço da placa de rede de origem, o endereço da placa de rede de destino, os dados de controle, os dados em si e a checagem de redundância cíclica (CRC).
Rede	É responsável pelo endereçamento dos pacotes, convertendo endereços lógicos em endereços físicos, de forma que os pacotes consigam chegar corretamente ao destino.
Transporte	Esta camada é responsável por pegar os dados enviados pela camada de sessão e dividi-los em pacotes que serão transmitidos à camada de rede.
Sessão	A camada de sessão permite que duas aplicações em computadores diferentes estabeleçam uma sessão de comunicação.
Apresentação	A camada de apresentação converte o formato do dado recebido pela camada de aplicação em um formato comum a ser usado na transmissão desse dado.
Aplicação	A camada de aplicação faz a interface entre o protocolo de comunicação e o aplicativo que pediu ou receberá a informação através da rede.

Figura 2 - Resumo das camadas do modelo OSI

Fonte: (ALENCAR, 2010)

2.3.3. Modelo TCP/IP

Criada pelo Pentágono, a ARPANET foi a primeira rede de computadores. Era uma rede de pesquisa em que pouco a pouco, centenas de universidades e repartições públicas foram conectadas, usando linhas telefônicas dedicadas. Quando se adicionaram as redes de rádio e satélite, começaram a surgir problemas com os protocolos existentes, o que forçou a criação de uma nova arquitetura de referência (TANENBAUM, 2003).

Essa arquitetura chamada de TCP/IP foi patrocinado pela *Defense Advanced Research Projects Agency* (DARPA), agência norte americana de pesquisas avançadas em defesa para poder conectar os computadores dentro da ARPANET. A arquitetura baseia-se principalmente em: um serviço de transporte orientado à conexão, fornecido pelo *Transmission Control Protocol* (TCP), e em um serviço de rede não-orientado à conexão (datagrama não confiável), fornecido pelo *Internet Protocol* (IP).

O modelo TCP/IP é composto por quatro camadas: enlace, inter-rede (internet), transporte e aplicação. Suprimindo as camadas de Sessão e Apresentação presentes no modelo OSI, isso porque o TCP/IP tem uma camada de sessão relativamente leve, competindo a ela abrir e fechar conexões sobre TCP, RTP e fornecer diferentes números de portas para diferentes aplicações sobre TCP e UDP. Se necessário, essas funções podem ser aumentadas por aplicações individuais (ou bibliotecas usadas por essas aplicações).

O quadro resumo contido na Figura 3 mostra as funcionalidades das camadas do modelo TCP/IP:

Camada	Descrição
Interface de rede (acesso à rede)	Esta camada, de acesso à rede, é a primeira do modelo TCP/IP, sua função é dar suporte à camada de rede, através dos serviços de acesso físico e lógico ao meio físico.
Inter-rede (Internet)	O nível inter-rede (Internet) é o responsável pelo envio dos datagramas de um computador qualquer para o outro computador, independente de suas localizações na rede.
Transporte	A camada de transporte é responsável por prover suporte à camada de aplicação de maneira confiável (ou não), independente dos serviços oferecidos pelas camadas de interface de rede e inter-rede.
Aplicação	A quarta camada do modelo TCP/IP é denominada de camada de aplicação. Nesta camada, estão os protocolos que dão suporte às aplicações dos usuários.

Figura 3 - Camadas do modelo TCP/IP

Fonte: (ALENCAR, 2010)

Na camada inter-rede todo o trabalho de se conectar redes de tecnologias distintas foi resolvido por meio do protocolo IP que oferece serviços de envio e recebimento de datagramas. O endereçamento do IP facilita a distribuição de informações entre os hosts em qualquer lugar do mundo.

Na camada de aplicação tem-se destaques os protocolos de nível mais alto dentro do modelo TCP/IP oferecendo serviços de transferência de arquivos (FTP), serviços de correio eletrônico (SMTP), o serviço de domínio de nomes na internet (DNS) e além do HTTP, usado para buscar páginas na *World Wide Web*. Na camada de transporte destacam-se os serviços de transmissão confiável entre as aplicações, orientado a conexão e assegurando que o pacote seja recebido (TCP) e o serviço de datagramas não orientado a conexão, que não oferece a entrega confiável do pacote, onde o importante é a velocidade dos pacotes (UDP).

3. MODELO PROPOSTO

Para se elaborar um sistema de automação residencial, alguns fatores devem ser considerados como a infraestrutura necessária à aplicação dos módulos do sistema requeridos pelo cliente, custos relativos ao projeto, normas e padrões que irão assegurar a qualidade, usabilidade, segurança e desempenho das instalações. Portanto a proposta de modelo neste projeto apresenta o desenvolvimento de um sistema integrando componentes de *hardware* e *software* que interagem por meio de comandos pré-programados na plataforma de prototipagem eletrônica Arduino Mega 2560 juntamente com o módulo *Ethernet* Shield, componente necessário para a comunicação dos sensores com a web por meio da rede local ou pela internet móvel, enquanto o Arduino Mega irá centralizar todas as regras do sistema e processar todos os comandos relativos aos módulos instalados na residência, um smartphone por meio de um aplicativo Android enviará tais comandos para os atuadores controlados.

Na primeira etapa do projeto foi elaborado o planejamento do protótipo, incluindo a perspectiva do projeto, em suma, quais os módulos pertinentes à aplicação do projeto e quais os métodos requeridos para a instalação destes módulos, as atividades seguiram com a construção física do ambiente protótipo a ser controlado, o algoritmo básico para a implantação dos módulos e a estimativa de quais componentes e materiais seriam utilizados ao longo do desenvolvimento da aplicação.

No projeto utilizou-se o sistema Android para o aplicativo smartphone por ser uma plataforma aberta, de fácil desenvolvimento e intuitiva, também foi utilizado o Arduino por ser uma plataforma microcontrolada, de licença aberta, para desenvolvimento de protótipos. O projeto será desenvolvido em um Ambiente de Desenvolvimento Integrado (IDE) próprio do Arduino utilizando a linguagem *Wiring* baseada em C/C++, em paralelo com o desenvolvimento de um aplicativo Android para controle da solução. O uso do *Ethernet Shield* no Arduino, simplifica o projeto de maneira que o usuário possa estar dentro da residência ou não e mesmo assim poder controlar os sistemas internos da mesma. Estes componentes levam em consideração o baixo custo de aquisição do sistema para residências de renda mais baixa.

3.1. DESCRIÇÃO DOS COMPONENTES DO SISTEMA

3.1.1. Arduino Mega 2560

O Arduino Mega é uma placa com microcontrolador ATMEGA2560 apresentado na Figura 4. Ele é ideal para projetos mais robustos que necessitam de um número maior de portas lógicas, melhor distribuição da alimentação e o maior uso de componentes no sistema, pois ele possui 54 pinos de entradas/saídas digitais, 16 entradas analógicas, 4 UARTs (portas seriais de *hardware*), um oscilador de cristal de 16 MHz, uma conexão USB, uma entrada de alimentação e um botão de *reset*. Ele contém tudo o que é necessário para dar suporte ao microcontrolador, basta conectar a um computador com um cabo USB ou a uma fonte de alimentação e já está pronto para começar (ARAÚJO, 2012).

Alimentação: A placa pode operar com alimentação externa entre 6 e 20 volts. No entanto, se menos de 7 volts forem fornecidos, o pino de 5V pode fornecer menos de 5 volts e a placa pode ficar instável. Com mais de 12V o regulador de voltagem pode superaquecer e danificar a placa, ou seja, a faixa recomendável é de 7 a 12 volts.

Memória: O ATmega2560 tem 256kB de memória flash para armazenamento de código (dos quais 8kB é usado para o bootloader), 8kB de SRAM e 4kB de EEPROM (que pode ser lido e escrito com a biblioteca EEPROM). (ARDUINO MEGA 2560, 2016)

Entrada e Saída: Cada um dos 54 pinos digitais do Mega pode ser usado como entrada ou saída, usando as funções de pinMode(), digitalWrite() e digitalRead(). Eles operam a 5 volts. Cada pino pode fornecer ou receber um máximo de 40mA e possui um resistor interno (desconectado por default) de 20 a 50kΩ. Em adição alguns pinos possuem funções especializadas (LIMA; NOBRE, 2015).

Comunicação: O Arduino Mega possui várias facilidades para se comunicar com um computador, com outro Arduino ou outro microcontrolador. O ATMEGA2560 fornece quatro portas de comunicação serial UARTs para TTL (5V). Um chip FTDI FT232RL direciona uma destas portas para a conexão USB e os drivers FTDI (que acompanham o *software* do Arduino) fornecem uma porta COM virtual para *softwares* no computador. O *software* do Arduino inclui um monitor serial que permite que dados simples de texto sejam enviados para a placa Arduino. Os LEDs RX e RT piscarão enquanto dados estiverem sendo transmitidos pelo chip FTDI e pela conexão USB ao computador (mas não para comunicação serial nos pinos 0 e 1) (ARDUINO MEGA 2560, 2016).

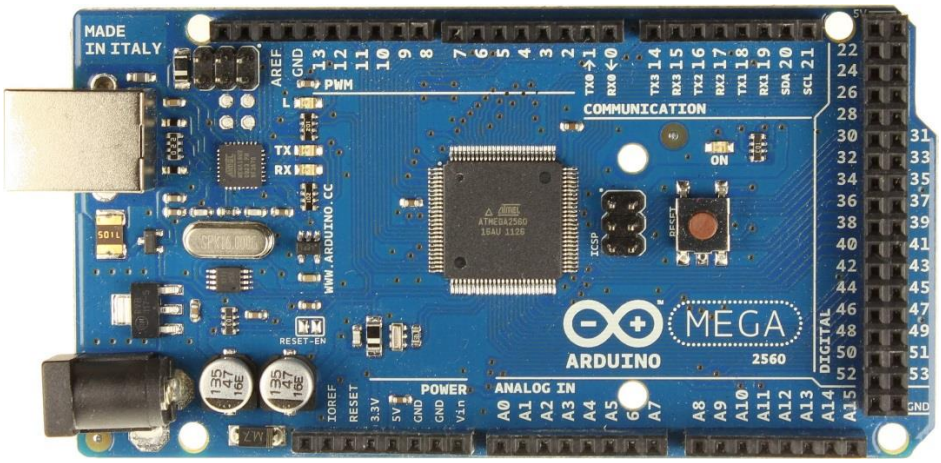

Figura 4 - Vista superior da placa Arduino Mega 2560

Fonte: (ARDUINO MEGA 2560, 2016)

3.1.2. Ethernet Shield

O Arduino *Ethernet Shield* na Figura 5 permite que um Arduino seja conectado à Internet de modo fácil e rápido, basta ligar este módulo em sua placa Arduino, conectá-lo à sua rede com um cabo RJ45 e seguir algumas instruções simples para começar a controlar suas funções através da internet. Ele é baseado no chip Wiznet W5100 que fornece uma biblioteca de rede (IP) que suporta tanto TCP como UDP. Ele suporta até quatro conexões de soquetes simultâneos. A biblioteca *Ethernet* permite escrever esboços de programas que se conectam à internet através do shield (DEVAL, 2015).

A biblioteca *Ethernet* serve para escrever programas que se conectem a internet através deste shield, ele se conecta ao Arduino por barras de pinos empilháveis, mantendo o layout e permitindo que outro shield se encaixe por cima. Há também um slot para cartões micro-SD que pode ser utilizado para armazenar arquivos que estejam disponíveis na rede, observado na Figura 5.

Com sua programação sempre disponível no meio eletrônico, pois trata-se de uma placa de código aberto, os programas relativos à controles das mais variadas funções do Arduino são de fácil associação e desenvolvimento. É compatível com o Arduino Uno e com o Mega. O Arduino comunica tanto com o W5100 quanto com o cartão SD utilizando o barramento SPI (através do cabeçalho ICSP). O compartilhamento de W5100 e SD card do barramento SPI não são de uso simultâneo, ou seja, o comando relativo ao uso do cartão SD

funcionará enquanto não houver solicitações do W5100 em relação a solicitações de IP, servidor, etc. (ARDUINO, 2016)

Figura 5 - Arduino *Ethernet Shield*
Fonte: (ARDUINO, 2016)

3.1.3. Sensores e Atuadores

A seguir, estão expostos os sensores e atuadores utilizados no projeto:

Relé Shield – O relé shield proporciona de maneira simples o controle de dispositivos de alta tensão, como se fosse um interruptor. Ele aciona as cargas a ele acopladas, dando segurança aos projetos, protegendo componentes sensíveis a alteração de tensão na rede. O relé é um dispositivo eletromecânico ou não (relés térmicos), com inúmeras aplicações possíveis em comutação de contatos elétricos. Servindo para ligar ou desligar dispositivos, em algumas aplicações serve como dispositivo de segurança para os dispositivos elétricos.

No caso do Relé eletromecânico, a comutação é realizada alimentando-se a bobina do mesmo. Quando uma corrente originada no primeiro circuito passa pela bobina, um campo eletromagnético é gerado, acionando o relé, possibilitando o funcionamento do segundo circuito. No caso do relé shield compatível com o Arduino, ele funciona com uma tensão de operação de 5V DC normalmente acoplado à uma das saídas digitais para seu controle, permite controlar cargas de 220V AC, tem uma corrente típica de operação de 15 a 20mA ideal para projetos com muitos componentes pois utiliza pouca carga e ainda possui um LED indicador de status. Conforme indica a Figura 6.

Figura 6 - Relé Shield
Fonte: (USINAINFO, 2016)

Micro Servo 9g SG90 - O **Micro Servo Motor SG90** é um motor muito utilizado em aplicações para robótica, nos sistemas microcontroladores como, por exemplo, Arduino, PIC e AVR. Ele é um módulo que apresenta movimentos proporcionais aos comandos indicados, controlando o giro e a posição, diferente da maioria dos motores. Servo motores são dispositivos de malha fechada, seu funcionamento dá-se por meio do recebimento de um sinal de controle, onde passam a verificar a posição atual de acordo com a programação feita e atuam no sistema indo para a posição desejada que vai de 0º a 180º. Opera em uma tensão de 3 a 6 V.

O Micro Servo Motor 9g é um motor compacto, pesa apenas 9g e oferece um torque máximo de ~1,6 kg, mostrando-se ideal para as mais diversas aplicações em projetos robóticos e eletrônicos em geral. A alimentação do micro é bem simples ele possui três fios de interface, onde dois servem para alimentação (Vcc e Gnd) e o outro para recebimento de sinais do controle. Exemplificado na Figura 7.

Figura 7 - Micro servo 9g SG – 90
Fonte: (USINAINFO, 2016)

Sensor LDR - O Resistor Dependente de Luz ou foto resistência (LDR) é um Componente eletrônico passivo do tipo resistor variável, ou seja, sua resistência varia conforme a intensidade da luz que incide sobre ele. Na prática, o LDR converte a luminosidade em valor de resistência, quanto maior a luminosidade menor a resistência e quanto menor a luminosidade, maior a resistência. Este sensor é comumente utilizado nas câmeras, alarmes de segurança, iluminação residencial ou até mesmo iluminação pública, apresentado na Figura 8. (USINAINFO, 2016)

Figura 8 - Sensor de luminosidade LDR
Fonte: (USINAINFO, 2016)

Sensor DHT11 – O Sensor DHT11 é definido como sendo um sensor de temperatura e umidade que permite medir temperaturas de 0 a 50 graus Celsius, e umidade na faixa de 20 a 90%. Mantendo sua faixa de precisão para temperatura em 2 graus e 5% para umidade. O Sensor DHT11 apresentado na Figura 9, na prática detecta a umidade e a temperatura, enviando estas informações para a placa microcontroladora, que deve estar programada para realizar alguma ação quando atingida determinada temperatura ou umidade. Opera com uma tensão de alimentação de 3 a 5,5V DC, com uma faixa de Corrente entre 0,5 a 2,5mA.

Um exemplo a ser destacado de sua utilização do Sensor de Umidade e Temperatura DHT11 é por meio da placa Arduino, onde é possível programar a placa microcontroladora para ligar, por exemplo, o ar-condicionado quando o ambiente atingir determinada temperatura, ou ligar a função desumidificar quando atingir determinada umidade. Possui ainda um elemento resistivo do tipo NTC para realizar a medição da temperatura. (USINAINFO, 2016)

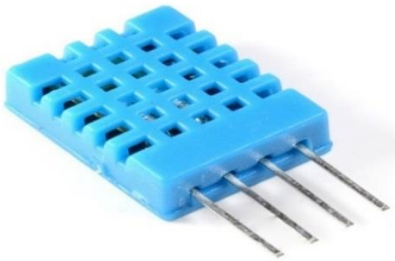

Figura 9 - Sensor de umidade e temperatura DHT11
Fonte: (USINAINFO, 2016)

Sensor PIR – Os sensores de presença mais comuns usam sensores PIR (*Passive Infrared Sensor*, ou Sensor Infravermelho Passivo) como detector de movimentos. No Arduino, temos o **módulo PIR DYP-ME003** apresentado na Figura 10, que une numa mesma estrutura o sensor PIR e também os circuitos necessários para ajuste e controle do sinal de saída.

O módulo contém o sensor PIR propriamente dito, composto internamente por duas faixas com material sensível ao infravermelho. Na parte externa, uma espécie de capa/tampa que na verdade é uma lente fresnel. Quando há variação na detecção do sinal infravermelho entre essas duas faixas de material sensível, a saída é acionada por um determinado tempo, ou seja, ele faz a detecção da variação de radiação infravermelha imposta pelo movimento do corpo humano.

A lente fresnel tem a função de "ampliar" o campo de visão do sensor, condensando a luz em um único ponto. É muito importante referir também que o Sensor PIR ou Sensor de Movimento para Arduino detecta somente movimento e não presença, deste modo, se algo permanecer parado frente a ele, o sensor PIR não irá detectar. (USINAINFO, 2016)

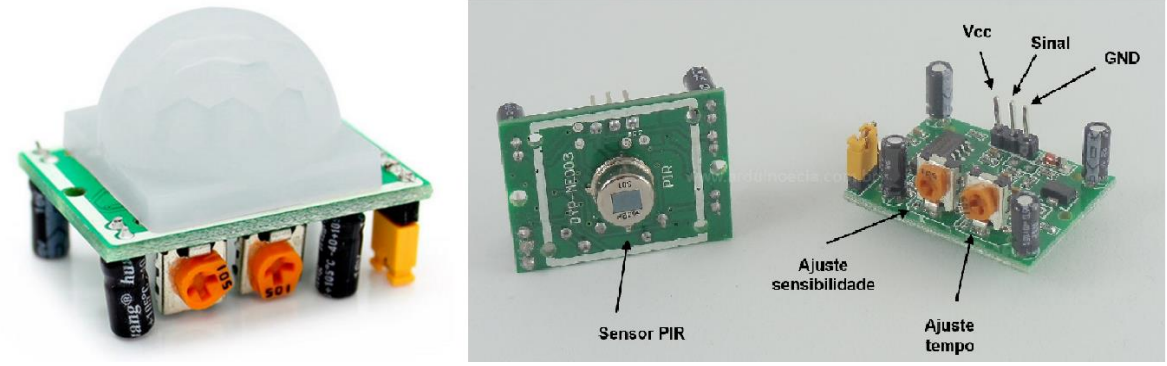

Figura 10 - Sensor de presença PIR DYP-ME003
Fonte: (USINAINFO, 2016)

LED – O principal atuador do projeto será o LED (do inglês *Light Emitting Diode*), onde este será distribuído sobre o protótipo a fim de simular a presença de lâmpadas reais numa residência. O Diodo Emissor de Luz (LED) é um diodo semicondutor (junção P-N) que quando é energizado emite luz visível – por isso LED. A luz não é monocromática (como em um laser), mas consiste de uma banda espectral relativamente estreita e é produzida pelas interações energéticas do elétron. O processo de emissão de luz pela aplicação de uma fonte elétrica de energia é chamado eletroluminescência. Conforme Figura 11.

Figura 11 - Diodo emissor de luz (LED)
Fonte: (USINAINFO, 2016)

4. DESENVOLVIMENTO DO MODELO PROPOSTO

4.1. CONSTRUÇÃO DA MAQUETE

Primeiramente para a realização do protótipo foi necessária a construção de uma maquete residencial com medidas em escala reduzida a fim de representar uma casa e seus respectivos cômodos que terão os dispositivos internos automatizados. A maquete é composta pelos seguintes cômodos: área externa, garagem, sala de estar, copa/cozinha, quarto 1, banheiro e quarto 2. A Figura 12 ilustra a disposição dos cômodos do protótipo.

Para a confecção da maquete foi utilizada uma folha de isopor para fazer a fixação dos cômodos. Já a parte superior da maquete foi confeccionada em folhas de papel Paraná. O teto serve de apoio para a fixação da placa Arduino e como melhor local para a disposição dos sensores e atuadores dentro da maquete.

Figura 12 - Visão aérea da maquete da residência a ser automatizada

Fonte: (Autor)

4.2. LIGAÇÕES FÍSICAS

Após a conclusão da maquete iniciaram os trabalhos de prototipagem e ligação física dos componentes nos respectivos cômodos, e logo após começaram os testes com códigos isolados de testes de cada componente em atuação com o Arduino. Cada LED deve ser ligado a um resistor para que a corrente emitida pelo Arduino não possa queimá-los e a um determinado pino no Arduino para que o controle de cada LED ocorra individualmente em cada cômodo, seguindo com a montagem, a disposição ficou como mostrado na Figura 13.

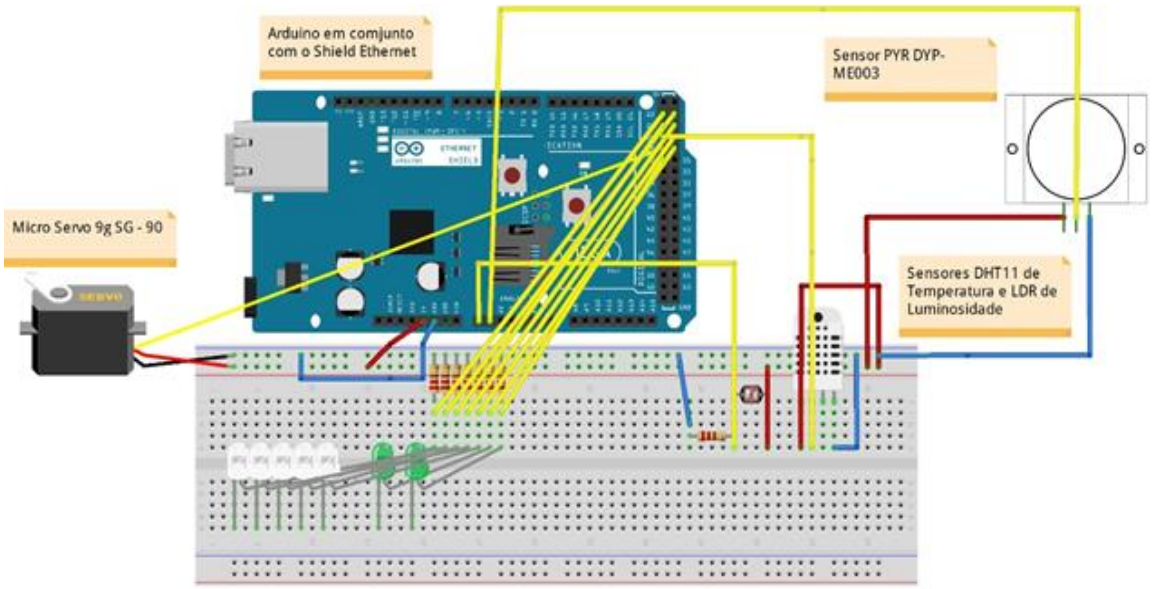

Figura 13 - Disposição dos leds e sensores sobre a protoboard com as ligações necessárias

Fonte: (Autor)

Cada cômodo tem seu controle independente através de um determinado pino no Arduino conforme a Tabela 2:

Tabela 2 - Disposição de pinos para o funcionamento do protótipo

Fonte: (Autor)

CÔMODO	PINO	SENSOR	PINO
SALA	22	UMIDADE E TEMPERATURA	26
COPA/COZINHA	23	PRESENÇA	A0

BANHEIRO	24	LUMINOSIDADE	A1
QUARTO 1	25		
QUARTO 2	27		
PORTÃO	28		
GARAGEM	29		
JARDIM	30		

4.3. DESENVOLVIMENTO DO SOFTWARE ARDUINO

Com o ambiente Arduino open-source torna se fácil escrever o código e enviá-lo à placa. Ele pode ser compilado em Windows, Mac OS X e Linux. O ambiente é escrito em Java e baseado em na linguagem Processing semelhante ao C/C++, avr-gcc e outros *softwares* de código aberto. Dando embasamento a quem já é habituado com ambientes de programação e facilitando a escrita de códigos. Conforme a Figura 14 observa-se a facilidade com qual o desenvolvedor pode operar o programa.

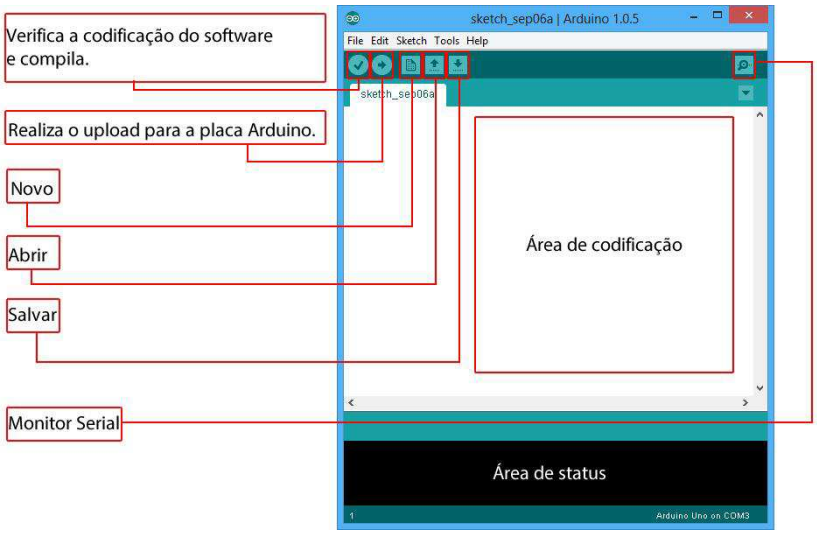

Figura 14 - Ambiente de desenvolvimento integrado arduino (IDE)

Fonte: (Autor)

Através do *bootloader* dispensa-se o uso de programadores para o chip no caso a família AVR do fabricante (ATMEL) facilitando ainda mais o seu uso uma vez que não exige compiladores ou *hardware* adicional (SOUZA, 2011). Após iniciar a IDE, é preciso definir qual a placa que está sendo usada, para que a IDE reconheça o *bootloader* e consiga enviar o seu código corretamente. Depois, apontar em que porta serial (virtual) está conectada o Arduino. Normalmente se configura a instalação com a porta **COM 3**, mas dependendo do console a porta pode ter um número superior, lembrando que essas configurações só serão possíveis com a placa conectada pelo USB ao computador (Figura 15).

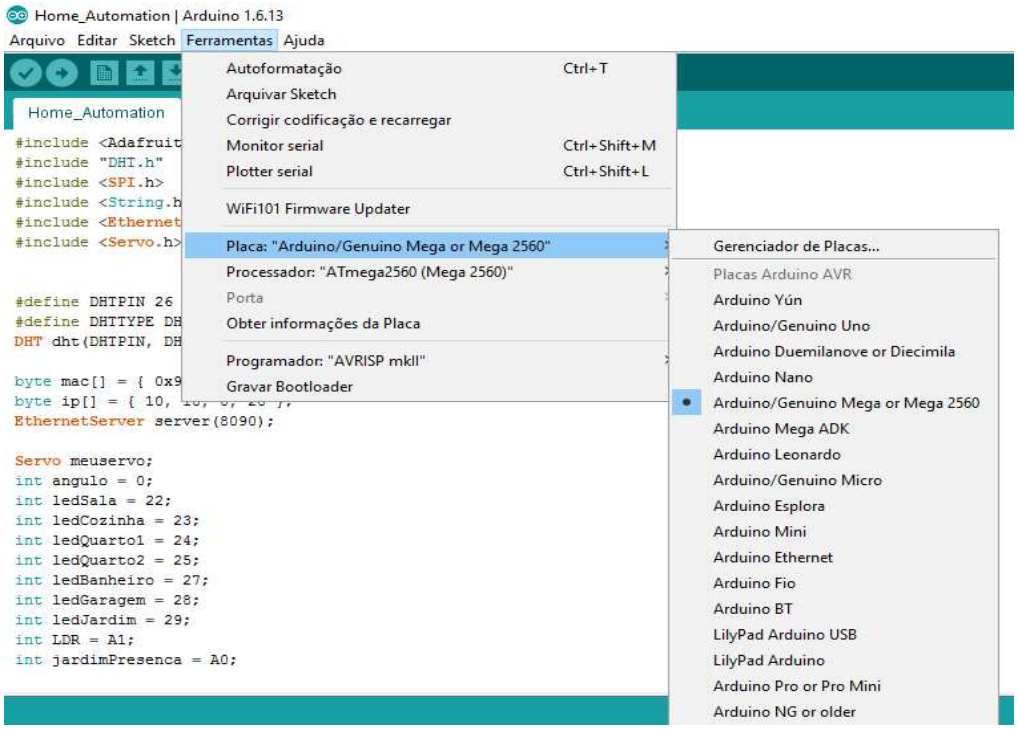

Figura 15 - Seleção da placa Arduino referente ao projeto

Fonte: (Autor)

Feito a instalação correta do IDE, faz-se a seleção da placa Arduino Mega 2560 e a seleção da Porta COM utilizada pelo mesmo. Foi iniciado o código de implementação do controle de iluminação dos cômodos, sendo que na etapa inicial foram definidas as variáveis e seus respectivos pinos e logo após, exportadas as bibliotecas necessárias aos sensores.

Junto com a definição dos pinos necessários aos comandos dos dispositivos, foi definido a interface de comunicação Arduino com a internet, com os comandos padrões do *ethernet shield* referentes ao endereço de IP necessário à aplicação, definição da porta do servidor em que o sistema irá se comunicar remotamente pela internet, observados na Figura 16. O IP pode

ser requerido de acordo com o local em que a aplicação esteja conectada, esse parâmetro é tratado no aplicativo do celular.

Figura 16 - Definição das variáveis dos cômodos e do servidor de internet utilizado

Fonte: (Autor)

Após a definição das variáveis foram escritas as funções que comandam os valores dos LEDs, sendo que cada pino trabalha separadamente com seu comando necessário e iniciada a leitura da porta serial do Arduino.

E com isso foi definido por meio de instruções em linguagem HTML um servidor interno de tratamento de requisições HTTP para que os comandos enviados pelo aplicativo possam ser tratados pelo *ethernet shield*, onde está criado o servidor web e a lógica de tratamento dos comandos, assim repassados os comandos ao Arduino pode-se visualizar a execução das ações dos sensores e atuadores. Assim o servidor recebe o comando enviado pelo celular e através de método *get* o servidor trata esse comando e atualiza o status dos dispositivos. A Figura 17 traz parte do código do servidor interno.

Assim após essa fase começou a fase de leitura dos sensores e aprimoramento dos mesmo de acordo com os valores lidos no monitor serial do Arduino (Calibração). Assim

fazendo os ajustes necessários às condições do ambiente o código de leitura dos sensores de temperatura e umidade (DHT11), presença (PIR DYP) e luminosidade (LDR) foram ajustados, conforme Figura 18.

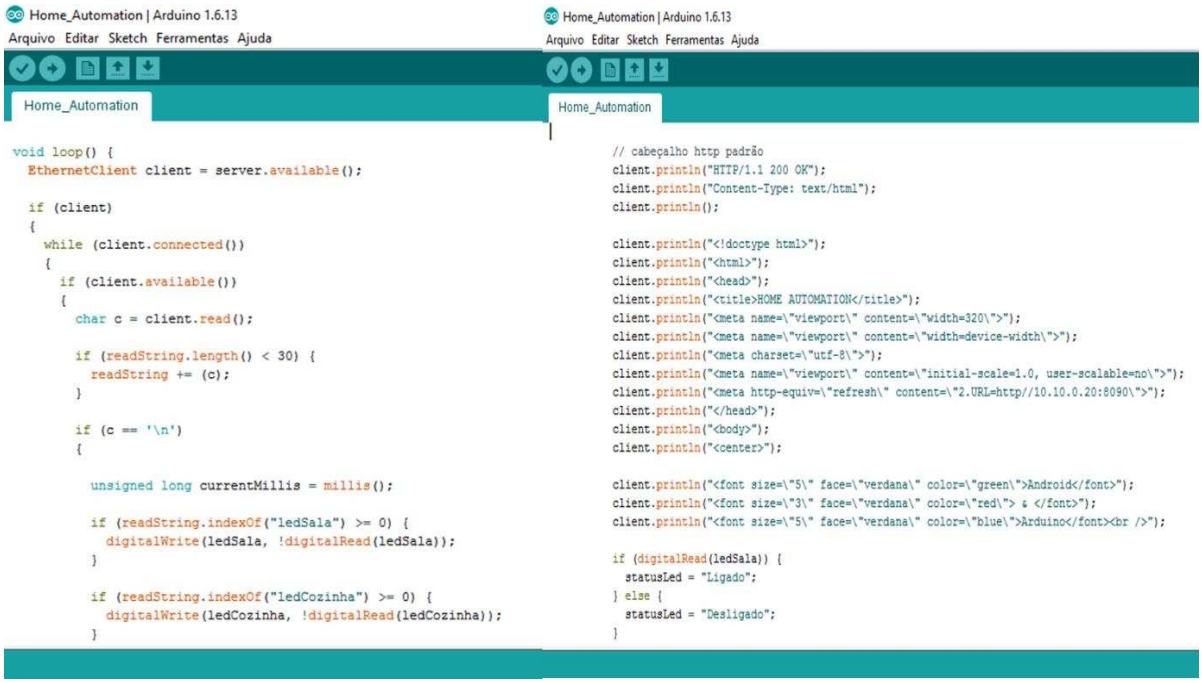

Figura 17 - Definição da página do servidor interno para a aplicação no Arduino

Fonte: (Autor)

Figura 18 - Lógica de programação do sensor de temperatura

Fonte: (Autor)

Depois de iniciar a leitura dos sensores foi confeccionada a lógica de instrução do sensor de presença a fim de prover um item de segurança ao protótipo, funcionando a partir do momento em que o ambiente estivesse em uma situação noturna e que alguma pessoa e/ou objeto ativasse o módulo de presença instalado na frente da residência. A Figura 19 mostra a função controladora do sensor. E também concluiu-se a lógica de controle do micro servo para a abertura e fechamento do portão da residência, com uma lógica simples após receber o comando "Abrir" pelo aplicativo um loop *for* é acionado fazendo com que o ângulo de rotação do micro servo seja incrementado a cada rotação, fazendo a abertura do portão, voltando a ser fechado assim que o comando "Fechar" é recebido pelo Arduino.

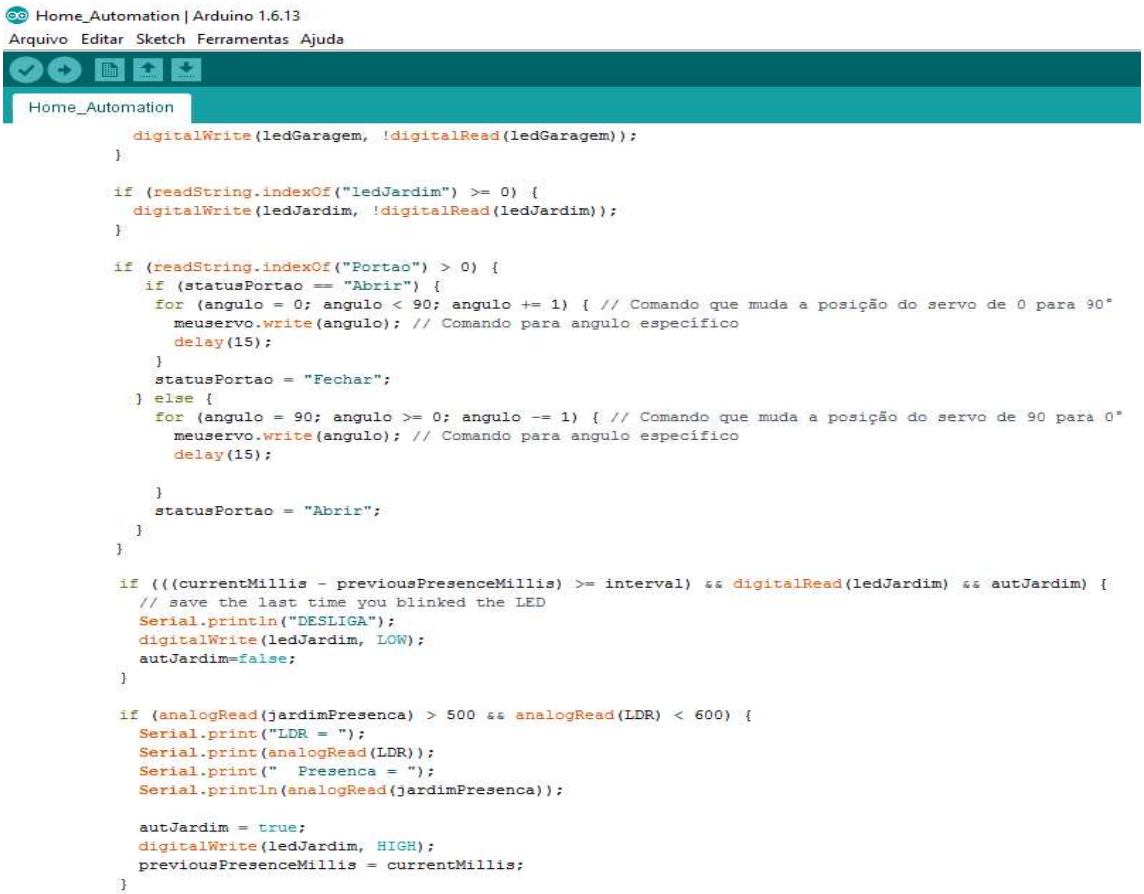

Figura 19 - Lógica dos sensores de presença, luminosidade e controle do portão

Fonte: (Autor)

4.4. DESENVOLVIMENTO SOFTWARE ANDROID

O Android é uma plataforma desenvolvida pela Google voltada para dispositivos móveis, totalmente escrita e desenvolvida em código aberto. Desde sua aplicação em 2007, sua

constante evolução e o constante aumento de dispositivos usuários do sistema, as contribuições para auxílio às aplicações se tornam um fator principal à escolha do sistema.

O Android SDK é uma ferramenta de desenvolvimento que disponibiliza um conjunto de APIs necessárias para desenvolver aplicações para a plataforma Android, utilizando a linguagem Java em conjunto com o Android Studio que fornece assistentes como a Máquina Virtual Java (JVM) voltada para simulação dos dispositivos móveis e suporte multimídia, além de modelos que verificam os requisitos do sistema como o *Java Development Kit* (JDK), memória RAM disponível e exemplos de aplicações integradas ao IDE. Há um editor de código inteligente capaz de realizar, refatorar e analisar códigos avançados. O poderoso editor de código auxilia na maior produtividade do desenvolvedor de aplicativos Android.

No núcleo da plataforma Android há o *Kernel Linux* para os serviços centrais do sistema tais como segurança, gestão de memória, gestão de processos, etc. O *kernel* também atua como uma camada de abstração entre o *hardware* e o resto do *software*. Assim a arquitetura da plataforma nos retorna um sistema robusto, capaz de tratar das mais variadas necessidades dos usuários presentes nos smartphones distribuídos atualmente visualizados na Figura 20.

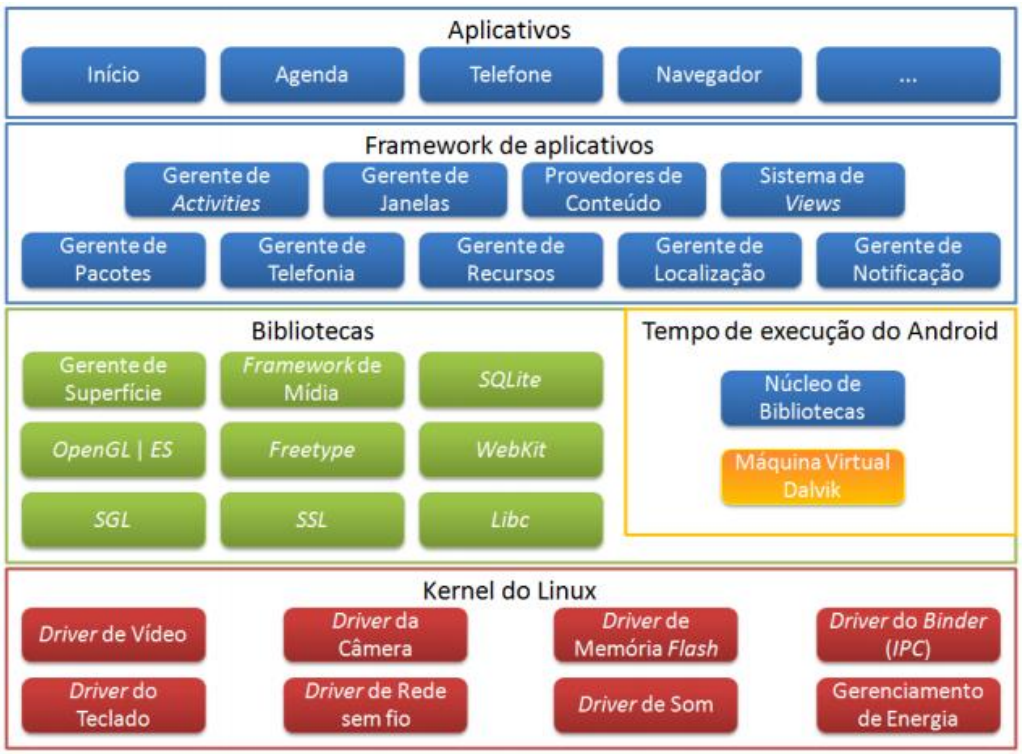

Figura 20 - Arquitetura geral da plataforma Android

Fonte: (ANDROID, 2016)

O *software* do aplicativo foi desenvolvido sob o Android Studio, apresentado na Figura 21, o IDE oficial para desenvolvimento de aplicações aos sistemas Android, baseado na linguagem de programação JAVA integrado ao SDK Android e alinhado ao sistema de criação de layouts em linguagem XML, o que nos retorna uma facilidade enorme ao desenvolvimento de aplicações.

Um aplicativo Android tem **04 pilares principais**, que fazem com que ele funcione como você o vê no seu celular, incluindo todas as suas funcionalidades, são elas as *Activitys*, os *Services*, os *Content Providers* e os *Broadcast Providers*. (ANDROID,2016)

As *Activitys* representam uma tela única com uma interface do usuário. Em cada tela do aplicativo que funcionam as mais variadas funções do aplicativo, no projeto as *activitys* representam as telas de login do usuário, a tela principal de controle da residência e a tela de configuração onde é inserido o endereço do servidor atualmente utilizado pela aplicação. No Apêndice A, são encontradas as lógicas do sistema implementado.

Os *Services* são componentes que são executados em segundo plano para executar operações de longa duração, ou seja, você não consegue visualizar a sua execução. São, por exemplo, eventos de música que pode tocar em segundo plano transparente ao usuário. Podem ser iniciados pelas *activitys* e continuar a funcionar independente. Na aplicação, o serviço de conexão com a internet é um exemplo de serviço.

Os *Content Providers* gerenciam um conjunto compartilhado de dados do aplicativo. Você pode armazenar os dados no sistema de arquivos, banco de dados ou na web. Através deles, outros aplicativos podem consultar ou mesmo modificar os dados da sua aplicação, caso o *Content Provider* permita essas ações incialmente. Os provedores de conteúdo são úteis para ler e gravar dados privados no aplicativo e não compartilhados

Um *Broadcast Receiver* é um componente que responde aos estímulos do sistema Android. Embora os receptores de transmissão não exibam nenhuma interface do usuário, eles podem criar uma notificação na barra de status para alertar ao usuário quando ocorre uma transmissão como, por exemplo, notificação de bateria fraca. Mais comumente, no entanto, um receptor de transmissão é somente um "portal" para outros componentes e realiza uma quantidade mínima de trabalho.

Para embasar todo o trabalho embutido nas *activitys* o Android Studio oferece um editor de layout avançado que permite arrastar e soltar funções diversas como botões, imagens e visualizadores de texto no layout e visualiza-los ao editar o XML. Assim o XML permite todo o tratamento da parte gráfica do aplicativo facilitando a codificação, indicando modelos e layouts para melhor fundamentar sua aplicação. Visualizado na Figura 21.

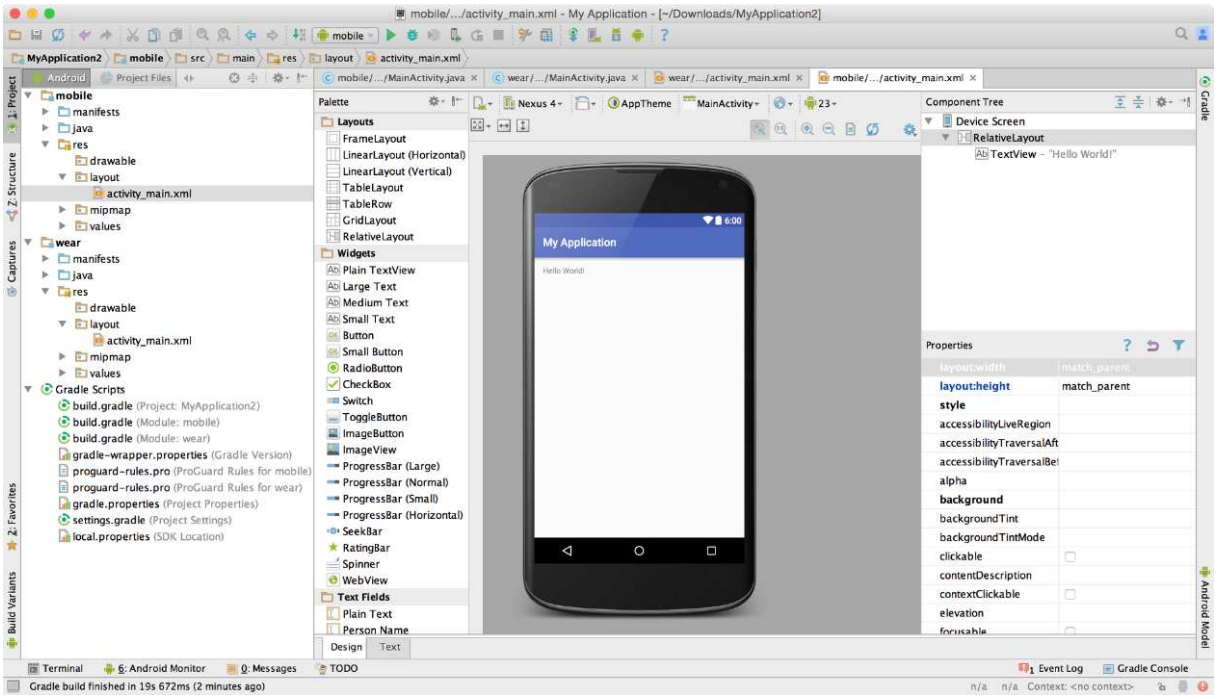

Figura 21 - Ambiente de desenvolvimento (IDE) Android Studio
Fonte: (ANDROID, 2016)

Após a familiarização com o ambiente de desenvolvimento, começou-se o desenvolvimento do App pertinente ao controle do sistema de automação residencial. O código fonte da aplicação pode ser visualizado perante o Apêndice A deste trabalho. Tendo como resultado final as telas do sistema e toda a conexão aplicada com o *ethernet shield* do Arduino, resultando assim no modelo da Figura 22.

Dentro da classe *Login()* (Figura 22a) foram estabelecidos os parâmetros de conexão com o servidor, como a definição do endereço de IP e qual porta de acesso ao servidor, após a conexão estabelecida por meio da classe *SolicitaDados()* os dados do usuário são requeridos por meio do método *Get* e se o usuário está adicionado ao banco de dados do servidor a aplicação segue normalmente. Caso o usuário não esteja cadastrado, ele pode ser adicionado ao banco de dados do servidor por meio da tela de cadastro (Figura 22b).

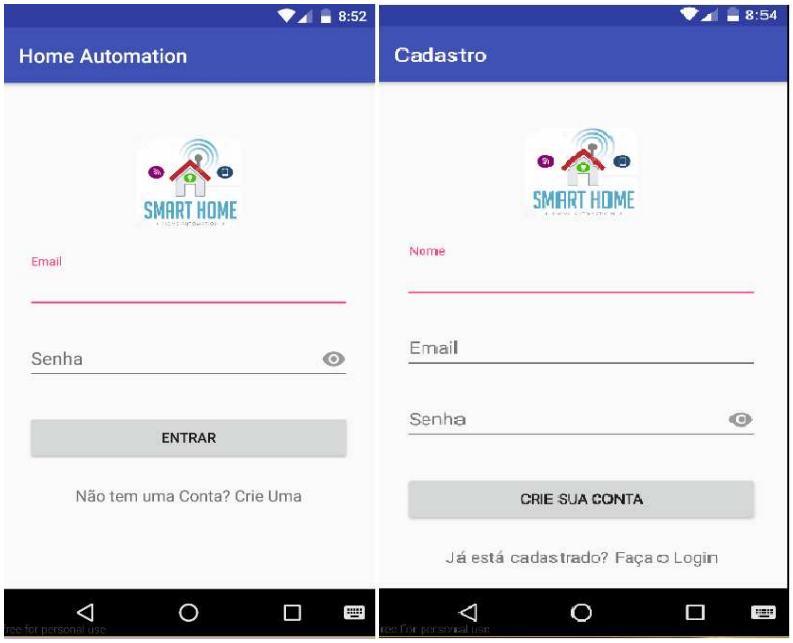

Figura 22 - a) Tela de login b) Tela de cadastro

Fonte: (Autor)

Onde nessas classes foram desenvolvidos os parâmetros necessários à conexão do aplicativo com um servidor web por meio de requisições HTTP.

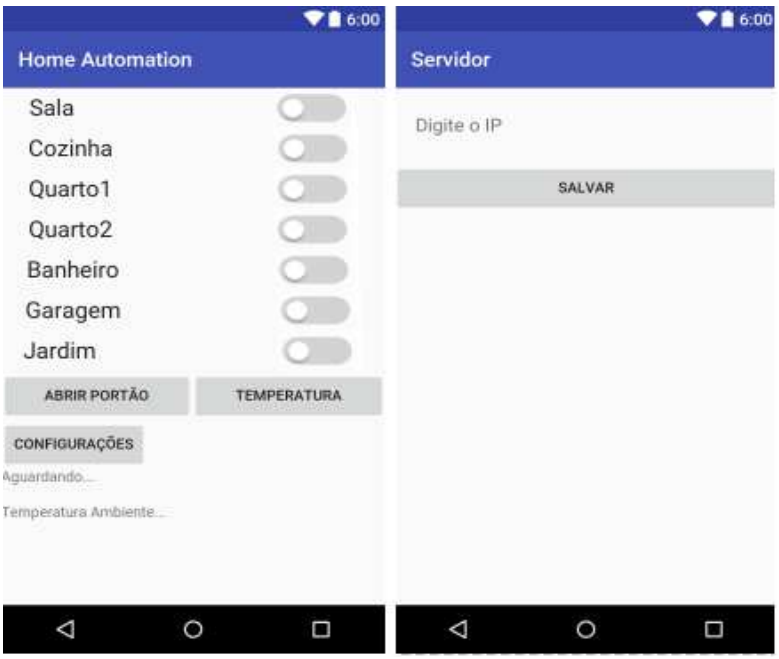

Figura 23 - a) Tela principal do aplicativo Home Automation b) Tela de configuração

Fonte: (Autor)

Após o *login* do usuário no sistema, o mesmo é levado à tela principal da aplicação onde se encontram os botões de controle da iluminação da residência, o botão de abertura e

fechamento do portão da residência e um botão para a solicitação e atualização da temperatura ambiente visualizados na Figura 23a. Além de um botão de configurações onde o usuário pode adicionar o servidor necessário para a conexão entre o aplicativo e o Arduino na Figura 23b.

Nessa parte da aplicação se encontra toda a programação da comunicação entre o aplicativo e o servidor controlado por meio do Arduino, caracterizando o modelo cliente/servidor para a comunicação utilizando os protocolos TCP e HTTP por meio das portas de comunicação externa (EUZEBIO; MELO, 2012).

Assim na *activity* principal do aplicativo temos o uso das *Tasks* por meio da classe *AsyncTask*, que permite fazer operações em *background* e enviar os resultados das requisições para a tela sem a necessidade de ter que implementar *threads* ou *handlers* (SILVA, 2014). Nessa classe foram implementados os métodos:

- *doInBackground()*: responsável por iniciar a conexão via sockets e mantê-la durante a aplicação;
- *Runnable()*: responsável por atualizar o status da aplicação depois de qualquer ação executada tanto pelo aplicativo quanto pelo servidor;
- *onPostExecute()*: responsável por receber e enviar por meio de *Strings* de dados as ações do sistema;
- *Solicita()*: responsável pela conexão com o servidor e tratamento dos dados após a conexão estabelecida.

Assim, com a implementação de toda a interface e comunicação do aplicativo com o servidor Arduino, o usuário tem toda a facilidade dos comandos por meio do toque na tela do aplicativo e tendo a rápida resposta do controlador no acionamento dos dispositivos e nas requisições dos dados dos sensores.

5. TESTES E RESULTADOS DO PROTÓTIPO

Alguns experimentos foram realizados para testar o funcionamento do protótipo. Em primeiro lugar, o *hardware* foi testado sem conexão com a internet, somente com os comandos sendo enviados diretamente pelo notebook e gravados no Arduino por meio da porta serial. Assim com esses comandos o Arduino aplicava nos módulos finais os comandos necessários, levando ao acionamento dos leds e demais componentes do sistema. Na fase seguinte, testou-se todo o sistema, incluindo o aplicativo Android para enviar os comandos aos módulos internos do sistema por meio do servidor de internet. Conforme Figura 24.

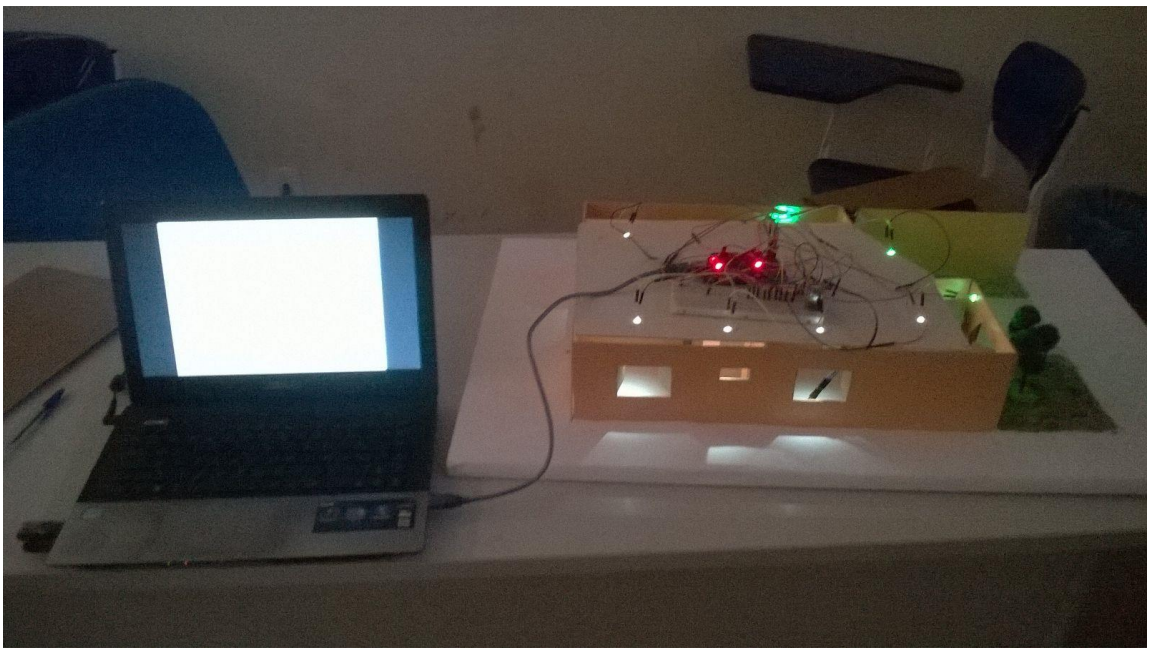

Figura 24 - Testes com comandos pelo notebook ao Arduino

Fonte: (Autor)

5.1. TESTES DE HARDWARE

Os testes do *hardware* objetivaram verificar:

- A correta comunicação entre o notebook e a placa de prototipagem eletrônica Arduino Mega 2560;
- O correto funcionamento dos comandos enviados para os respectivos dispositivos internos do sistema de automação.

Para a verificação do correto funcionamento, pode ser efetuado o envio de comandos através da funcionalidade *Serial Monitor*, disponível no *Software* IDE de compilação do Arduino. Ao enviar o comando “Ligar” deverá ser acionado a luz do cômodo sala, ao enviar “Desligar” deve ocorrer o desligamento. A Figura 25 demonstra alguns testes realizados com o Arduino:

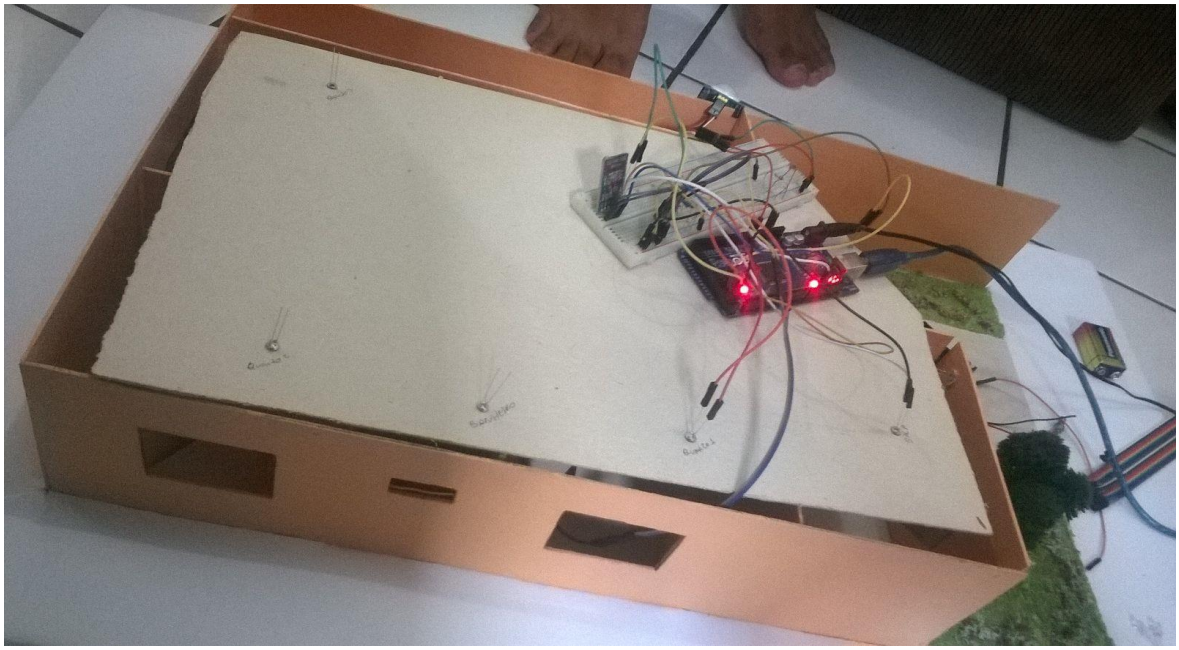

Figura 25 - Testando LEDs referentes à sala e quarto 1 com comando serial Arduino

Fonte: (Autor)

5.2. TESTES FINAIS DO SISTEMA

Uma vez verificada a correta comunicação entre o notebook e o Arduino, efetuou-se o teste da plataforma completa. O Smartphone possui o aplicativo *Home Automantion* previamente instalado, por meio deste aplicativo que são enviados os comandos que acionam as lâmpadas e os dispositivos do sistema. Para que ocorra essa interligação do Smartphone com o Arduino é preciso estabelecer a conexão com o servidor indicado pelo aplicativo. Assim quando o aplicativo for iniciado o sistema de segurança do mesmo solicitará o *login* do usuário, tendo o *login* efetuado com sucesso a parte principal da aplicação é acionada e tendo a aplicação um servidor válido cadastrado e conectado, ela ficará disponível aos comandos do usuário. Caso não haja servidor conectado a aplicação solicita o cadastro do servidor para a continuação da aplicação.

Após a correta conexão os testes de fato foram efetuados. Ao pressionar o botão denominado “Sala” na tela do Smartphone, a luz correspondente deverá acender na maquete.

Isso ocorre pelo fato de que no momento em que o botão é pressionado o aplicativo dispara o envio de um pacote com a *string* “LedSala”, assim o Arduino ao receber essa *string* devolve ao aplicativo o status do led correspondente à solicitação, a *string* é capturada e o *Processing* faz a interpretação. Após a interpretação, o aplicativo muda o status da variável com o valor contrário ao valor recebido, ou seja, se o Led estava ligado o aplicativo envia o socket para desligá-lo e vice-versa. Assim feitas todas as requisições pelo aplicativo o correto funcionamento dos LEDs de todos os compartimentos pode ser constatado como na Figura 26.

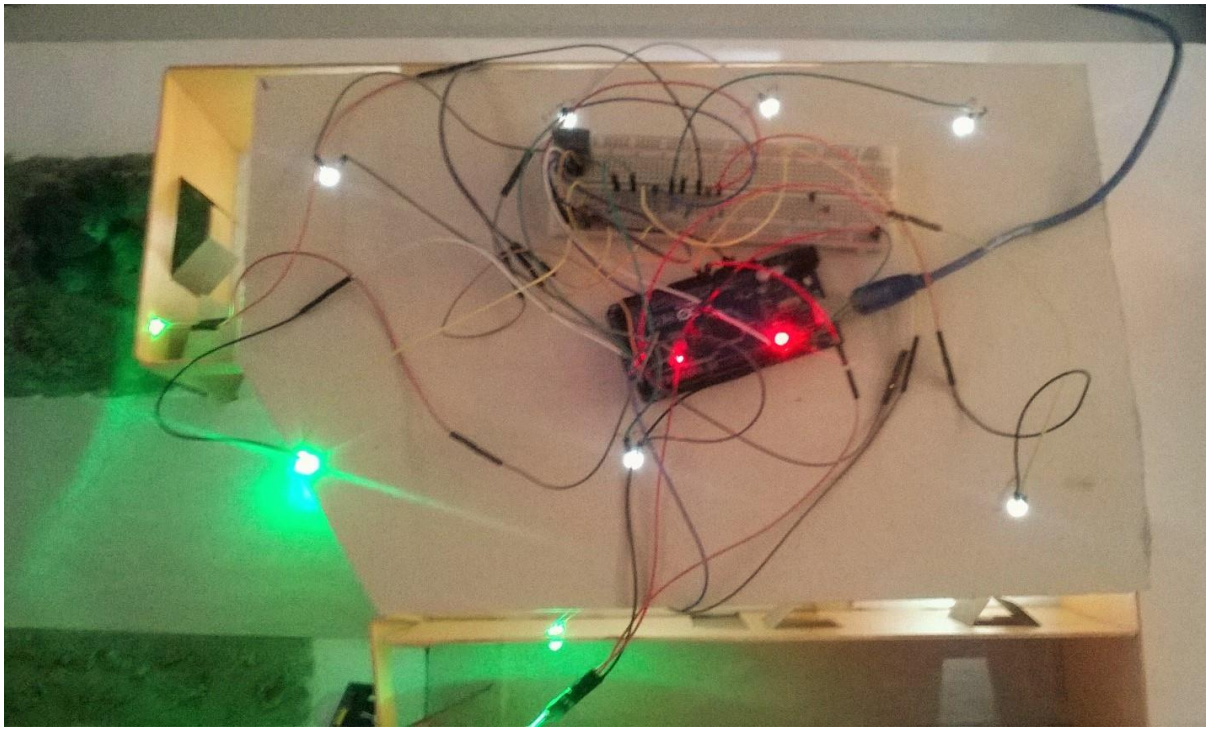

Figura 26 - Testes do sistema aplicativo com comandos em touch.

Fonte: (Autor)

Constatando também que no final todos os sensores atenderam aos pré-requisitos estabelecidos funcionando corretamente em todas as fases de testes realizados no protótipo. No teste de solicitação da temperatura ambiente o resultado foi satisfatório, o tempo de resposta foi de aproximadamente 0,5 s entre o clique no aplicativo até o momento de exibição na tela do celular mantendo assim a leitura constante de 5 em 5 s. Já o sensor de presença nos deu um resultado bem satisfatório percebendo a presença de pessoas a mais de 5 m de distância, o que seria de grande valia para uma instalação real.

6. CONCLUSÕES E CONSIDERAÇÕES FINAIS

A automação residencial se apresenta como uma revolução nos ambientes domésticos por incorporar esse novo conceito de integração entre os diversos equipamentos e dispositivos de uma casa em uma conexão remota de comandos. Apesar do ceticismo que ainda existe por parte da população advindo do pensamento sobre ao alto ônus empregado em uma aplicação como esta, percebe-se que cada vez mais a sociedade e usuários demandam por soluções de automação em suas residências, principalmente quando o propósito está voltado à acessibilidade, conforto, segurança, entretenimento e economia de recursos.

De acordo com os tópicos apresentados neste trabalho, pode-se constatar que a criação de um sistema de automação residencial modular com a atuação de sensores e atuadores no processo final de execução da tarefa requerida, contando com a viabilidade tecnológica da utilização de um Smartphone para o controle no ambiente residencial, proposto neste trabalho, foi validada por meio de uma arquitetura adequada e da integração entre os códigos desenvolvidos, ressaltando o fato de que essa tecnologia para ser empregada não necessita de um orçamento alto e pode ser implantada de modo fácil e intuitivo, sendo um sistema confiável com baixo custo/benefício comparado com o grau de proteção que proporciona.

No presente trabalho conseguiu-se atender aos objetivos propostos e atingir os resultados esperados, constatando-se que é possível a construção de um sistema de automação residencial por meio de um aplicativo para funcionar em Smartphones com o sistema Android enviando comandos remotamente por meio da internet para os dispositivos responsáveis pelo controle da iluminação e demais tarefas em um ambiente residencial. Sendo implementados a programação dos módulos e seus respectivos blocos para o controle da automação em uma placa microcontroladora Arduino, para a comunicação entre este e o dispositivo móvel.

Assim a interação final de todos os módulos atendeu aos requisitos impostos no início do trabalho.

SUGESTÕES PARA TRABALHOS FUTUROS

- Uma das possibilidades de melhoria no sistema seria uma sessão de configuração de perfis de acordo com cada usuário do sistema, assim o sistema possibilitaria a criação e o gerenciamento dos perfis do usuário. Esses perfis seriam formados por um grupo de dispositivos que seriam ligados/desligados automaticamente em determinadas horas do dia;

- Melhorias na interface do aplicativo Android pensando na facilidade de acesso e na melhor interação do usuário com o sistema, podendo adicionar sistemas de acessibilidade a pessoas com deficiência e outros módulos de controle;

- Possibilidade de adicionar o registro das atividades e informações pertinentes ao usuário em nuvem por meio de um banco de dados, possibilitando a checagem da quantidade de tempo que cada dispositivo ficou ligado, horário em que foi ligado e desligado, entre outros, provendo mais segurança ao usuário;

- Além disso, podem ser realizadas melhorias no *software* para o Android pensando em um controle mais eficaz da luminosidade do ambiente. Uma vez que certos aparelhos possuem um sensor de luminosidade integrado, pode-se desenvolver um aplicativo que utilize esse sensor para realizar a dimerização das lâmpadas.

REFERÊNCIAS

[1] ALENCAR, Márcio A. S. **Fundamentos de Redes de Computadores**. Manaus: Universidade Federal do Amazonas, CETAM, 2010. 47 p.

[2] ALVAREZ, Daniel F. S.; ANTUNES, F. I. **Automação residencial utilizando bluetooth, ethernet e smartphone**. Trabalho de Conclusão de Curso – (Graduação em Tecnologia em Mecatrônica Industrial), UTFPR. Curitiba. 2015.

[3] ANDROID. **Fundamentos de aplicativos**. Disponível em: < https://developer.android.com/guide/components/fundamentals.html>. Acesso em: 20 out. 2016.

[4] ARAÚJO, I. B. Q. et al. **Desenvolvimento de um protótipo de automação predial/residencial utilizando a plataforma de prototipagem eletrônica arduino.** João Pessoa. Setembro. 2012.

[5] ARDUINO. **Arduino Ethernet Shield**. Disponível em: <https://www.arduino.cc/en/Main/ArduinoEthernetShield>. Acesso em: 12 out. 2016.

[6] ARDUINO. **Arduino Mega 2560**. Disponível em: <http://arduino.cc/en/Main/ArduinoBoardMega 2560>. Acesso em: 19 out. 2016.

[7] AURESIDE. **Potencial do mercado de automação residencial**. Disponível em: <http://aureside.org.br/noticias/automacao-residencial--demanda-na-construcao-civil>. Acesso em: 28 out. 2016.

[8] BOLZANI, Caio A. M. **Residências inteligentes.** 1. ed. São Paulo: Editora Livraria da Física, 2004.

[9] DEVAL, Felipe A. **Automação residencial de baixo custo utilizando tecnologias open source**. Trabalho de Conclusão de Curso – (Graduação em Engenharia de Computação), do Centro Universitário de Araraquara-UNIARA. Araraquara. 2015.

[10] EUZEBIO, Michel V. de M.; MELLO, Emerson R. **DroidLar - Automação residencial através de um celular Android**. IFSC. Santa Catarina. 2012.

[11] FOROUZAN, Behrouz A. **Comunicação de Dados e Redes de Computadores**. 3ª edição. Porto Alegre: Bookman, 2006.

[12] GOMES, Victor Z. Ferreira. **A Domótica como instrumento para a melhoria da qualidade de vida dos portadores de deficiência.** Trabalho de Conclusão de Curso – (Graduação em Tecnologia em Automação Industrial), IFPB. João Pessoa, 2010.

[13] IDC BRASIL, **Previsão da IDC para o mercado de TIC no Brasil em 2016 aponta crescimento de 2,6%**. Disponível em: **<**http://br.idclatin.com/releases/**>**. Acesso em: 25 out. 2016.

[14] LIMA, Emanuel M. S.; NOBRE, Antonio Y. M. **Automação residencial de baixo custo com arduino mega e ethernet shield**. Centro Universitário Estácio do Ceará. Fortaleza. 2015.

[15] MOREIRA, Jonathan Rosa. **AutoControl: Uma proposta de acessibilidade e segurança residencial com apoio da plataforma Arduino**; Trabalho de Conclusão de Curso; (Graduação em Sistema de Informação), Faculdade Projeção. São Paulo, 2013.

[16] PEREIRA, Wellington A. et. al. **Desenvolvimento e análise de um sistema de automação predial utilizando uma central de controle via rede externa.** Trabalho de Conclusão de Curso – (Graduação em Engenharia de Computação), Curitiba. 2013.

[17] SILVA, Bruna R. S. **Sistema de automação residencial de baixo custo para redes sem fio**. Trabalho de Conclusão de Curso – (Graduação em Engenharia de Computação), UFRGS. Porto Alegre. 2014.

[18] TANENBAUM, Andrew S. **Redes de Computadores**. Rio de Janeiro, 2003.

[19] USATEGUI, José M. A.; MARTÍNEZ, Inácio A. **MICROCONTROLADORES PIC: DISEÑO PRAC. DE APLICACIONES 16F84 PA**. 3. ed. Madri: Mcgraw-hill, 1999.

[20] USINAINFO. **Sensores para Arduino**. Disponível em: <http://www.usinainfo.com.br/sensores-para-arduino>. Acesso em: 10 out. 2016.

APÊNDICE A - Código Fonte Arduino e Android

Para facilitar o acesso aos códigos fonte da aplicação uma alternativa encontrada foi alocar a mesma em um site de repositórios de projetos, onde no mesmo se encontra o projeto contendo a programação necessária para um sistema de automação residencial com o aplicativo dispondo de uma interface de *login* e todos os comandos necessários ao sistema de automação. Junto com o código do servidor Arduíno. Sendo que a parte de *hardware* do sistema foi explicada durante o projeto sendo ela configurada e controlada pela parte de *software*.

Essa parte do sistema foi dividida em duas frentes onde o aplicativo *Home Automation* ficou responsável pelo controle de todos os dispositivos do sistema e o *software* do controlador ficou responsável por hospedar o servidor necessário à aplicação e por aplicar as ações em conjunto com os sensores e atuadores.

Assim para maiores esclarecimentos sobre o sistema basta acessar os códigos do mesmo no endereço abaixo, tendo as classes todas comentadas para a fácil compreensão dos códigos:

https://github.com/eryckaraujo/Home_Automation.

Onde encontram-se os arquivos:

Conexao.java, Conexao_Login.java, LoginActivity.java, MainActivity.java, MainConfig.java, TelaCadastro.java, arquivos referentes à conexão do cliente Android via sockets tanto com o servidor quanto com o Arduino.

As classes **activity_login.xml, activity_main.xml, main_config.xml, tela_cadastro.xml** são responsáveis pela programação completa das telas do sistema, cada arquivo corresponde a uma das telas do sistema.

E o **AndroidManifest.xml** é responsável pela chamada das classes no Android e as informações passadas ao sistema.