

UNIVERSIDADE FEDERAL DE VIÇOSA  
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

MARCOS AURELIO FREIRE FERRAZ PASSOS

**CONTROLE DOMÓTICO VIA SMARTPHONE E ARDUINO**

VIÇOSA  
2016

MARCOS AURELIO FREIRE FERRAZ PASSOS

## **CONTROLE DOMÓTICO VIA SMARTPHONE E ARDUINO**

Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 490 – Monografia e Seminário e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.  
Orientador: Prof. Dr. Denilson Eduardo Rodrigues.

VIÇOSA  
2016

Nesta página será inserida a ficha catalográfica correspondente à sua Monografia. Ela será elaborada pelo pessoal da Biblioteca Central da UFV.  
A priori deixa a página em branco.

MARCOS AURELIO FREIRE FERRAZ PASSOS

## **CONTROLE DOMÓTICO VIA SMARTPHONE E ARDUINO**

Monografia apresentada ao Departamento de Engenharia Elétrica do Centro de Ciências Exatas e Tecnológicas da Universidade Federal de Viçosa, para a obtenção dos créditos da disciplina ELT 490 – Monografia e Seminário e cumprimento do requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Aprovada em 13 de Dezembro de 2016.

### **COMISSÃO EXAMINADORA**

---

**Prof. Dr. Denílson Rodrigues Eduardo - Orientador**  
**Universidade Federal de Viçosa**

---

**Prof. Dra. Kétia Soares Moreira**  
**Universidade Federal de Viçosa**

---

**Engenheiro Samuel Chaves**  
**Universidade Federal de Viçosa**

---

**Engenheiro Luis Otávio Maciel de Oliveira**  
**Universidade Federal de Viçosa**

*“The key for everything is a positive attitude”*

*(Kyle Larson)*

*Dedico este trabalho primeiramente a Deus pelo dom da vida e pela força até então dada e posteriormente aos meus amigos e família da Bahia, Minas Gerais e demais regiões .*

## *Agradecimentos*

Agradecimentos são sentimentos de retribuição pela gratidão que sentimos pelas pessoas que nos ajuda, nos zela, nos cuida. E é impossível não lembrar daqueles que estiveram e estão próximos de mim, me motivando e me fazendo não desistir do que acredito que seja o melhor.

Eu gostaria de agradecer, primeiramente a Deus, por entender que Ele cuida de mim nas mínimas coisas com o pão nosso de cada dia que é aquilo que é e será necessário na minha jornada. Agradecer a minha família, pais, irmãos, primos, tios e tias por nunca me desampararem diante de tantas dificuldades. Agradecer, muitíssimo, a minha tia Ana Virgínia por me gerir suporte financeiro e conselhos, sem os quais, nunca chegaria neste momento. Agradecer a meus avôs, principalmente a minha avó Rosália que sempre me acalmou em momentos difíceis e de angústia. Agradecer aos verdadeiros amigos e amizades por sempre estarem acreditando no meu potencial ou até mesmo me fazendo rir em momentos tristes. Agradecer pela alegria de todos, pelos colegas de sala, pelos colegas de vida, pelos colegas de república, tanto do 1611 quanto da república SodaBahia. Agradecer pela família Pimentel por me abrigar tantas vezes em sua casa aqui nesta cidade. Agradecer a orientação do Professor Denilson com este trabalho realizado. Agradecer pelas pessoas de igreja, amigos e parentes que me mostraram o verdadeiro caminho que é Deus.

O meu sincero muito obrigado a todos que de alguma forma contribuíram para a pessoa quem sou hoje.

## ***Resumo***

Com a tendência a automatizar diferentes áreas de aplicações, a domótica surge com o intuito de trazer simplificação para a vida das pessoas e atender as necessidades de comunicação, conforto e segurança. Portanto, essa área contempla o conjunto de ferramentas e funções que tornam o ambiente residencial mais seguro, acessível e cômodo.

Para este trabalho executou-se em uma maquete-casa de madeira, a fim de simular a rotina diária de uma residência, três tarefas para melhorar a qualidade de vida do usuário. Os sistemas se dividem em sistema anti-chuva, sistema de iluminação e o sistema de ventilação. Todos estes sistemas são monitorados, e em sua maior parte, controlados via *smartphone*.

Para automatizar o processo utilizou-se uma placa microcontrolada *arduino UNO*, a placa *Ethernet Shield W5100*, um roteador como rede local e comum para transmissão e recepção de dados e um *smartphone* programado para o monitoramento e controle dos processos. Para os sistemas utilizou-se um sensor de chuva e motores DC 5V para acionamento da janela da casa. Também foi utilizado diodo de emissão de luz para simular a iluminação de algum cômodo da maquete-casa. Por fim, utilizou-se um outro motor 5V de corrente contínua para acionar um ventilador, simbolizando o sistema de refrigeração. Para estes três ambientes automatizados utilizou-se um *Smartphone* para atuar e monitorar a planta de acordo com a necessidade do usuário.

Os resultados mostram a aplicabilidade e conforto que estes sistemas automatizados trazem ao usuário. Além disso, é notório perceber a acessibilidade que um sistema automatizado por *arduino* via *smartphone* possui, pois é de fácil manuseio e entendimento.

**Palavras-chaves:** Domótica, Arduíno, *Smartphone*, Sensor de Chuva, *Ethernet*, Maquete.



# *Sumário*

1	Introdução.....	12
1.1	Objetivo Geral.....	13
1.1.1	Objetivos Específicos.....	13
2	Revisão Bibliográfica.....	14
2.1	Automação Residencial.....	14
2.2	Arduino.....	13
2.2.1	Arduino Uno.....	13
2.2.2	Ambiente de Desenvolvimento Integrado para Arduino.....	17
2.3	Instrumentação.....	18
2.3.1	Sensor de Chuva YL-83.....	18
2.3.2	Motores DC 5V.....	19
2.3.3	Diodos Emissores de Luz (LED's).....	19
2.3.4	Ponte H L293D.....	20
2.4	Comunicação dos Dispositivos.....	22
2.4.1	Tecnologia Ethernet/Wireless.....	22
2.4.2	SPI(Interface Periférica Serial).....	23
2.4.3	Ethernet Shield W5100.....	24
2.5	Uso das Ferramentas para Controle de Processos.....	25
2.5.1	Smartphones.....	25
2.5.2	MIT App Inventor.....	25
3	Material e Métodos.....	27
3.1	Instrumentação e Implementação do Protótipo Residencial.....	28
3.1	Configuração do Arduino para as Funções da Maquete.....	30
3.3	Configuração da Rede.....	30
3.4	O aplicativo para Automação Residencial.....	31
3.5	Comunicação entre Aplicativo e Arduino.....	33
4	Resultados.....	34
5	Conclusões.....	39
	Referências Bibliográficas.....	41

## *Lista de Figuras*

Figura 1 – Arduino Uno .....	16
Figura 2 – Componentes básicos do arduino Uno .....	16
Figura 3 – Aparência do programa IDE para Arduino .....	17
Figura 4 - Da esquerda para direita: fios conectores, sensor de chuva (YL-83) e módulo de controle.....	18
Figura 5 – Módulo de controle para o sensor de chuva.....	19
Figura 6 – Motores DC 5V.....	19
Figura 7 – Configuração física do diodo emissor de luz. ....	20
Figura 8 – Esquemático da Ponte L293D.....	21
Figura 9 – Diagrama de blocos do L293D. ....	21
Figura 10 – Ligação dos pinos entre o dispositivo mestre e os escravos. ....	23
Figura 11 – Ethernet Shield w5100. ....	24
Figura 12 – Ethernet Shield w5100 sobreposto ao arduino UNO.....	24
Figura 13 – Janela de designer do MIT inventor.....	26
Figura 14 – Janela de blocks do MIT inventor.....	27
Figura 15 – Movimentos possíveis do motor 5V. ....	29
Figura 16 – Maquete com os circuitos automatizados. ....	29
Figura 17 – Leitor QR code para baixar aplicativo.....	31
Figura 18 – MIT inventor para o circuito da janela.....	33
Figura 19 – Programação do circuito da janela. ....	34
Figura 20 – Página atualizada com o IP do computador. ....	35
Figura 21 – Tela do smartphone.....	35
Figura 22 – Tela login.....	36
Figura 23 – Login requisitado incorretamente. ....	37
Figura 24 – Tela inicial.....	37
Figura 25 – Tela dos circuitos. ....	38
Figura 26 – Aplicativos com perda de informação. ....	38

## *Lista de Tabelas*

Tabela 1 – Na ordem: enable, input1, input2 e a função saída.....	22
Tabela 2 – Telas dos aplicativos, itens e suas funcionalidades. ....	32

# 1 *Introdução*

A palavra Domótica é o neologismo entre o radical latim domus que remete a palavra casa e robótica que representa qualquer operação automatizada (Ferreira, V. Z. G, 2010). A Domótica iniciou se na década de 70 e é a área que preocupa se em prover ao proprietário de uma residência o máximo conforto possível (Chamusca, A. 2006).

A Domótica é dividida em dois tipos de arquiteturas: ABA (Arquitetura Baseada em Automação) e a ABC (Arquitetura Baseada em Comportamento), ou também, comumente, chamadas de Domótica estática e Domótica inteligente, respectivamente. A primeira trata a automação de residências a partir de controles remotos, sensores de movimento, dispositivos biométricos, todos esses ajustados de modo que os habitantes não precisem se adaptar ou até mesmo configurá-lo fazendo com que o mesmo funcione de acordo com as necessidades dos usuários. A segunda é entendida pelo processo de automação que incorpora algum mecanismo automático de tomada de decisão baseada em técnicas de Inteligência Artificial (Michell, 1997). Sendo assim, a domótica inteligente é um sistema que vai além da automação residencial.

De acordo com (Vivas, 2015) a domótica preza por praticidade, acessibilidade, economia, comunicação, segurança e comodidade. A aplicação dessa área beneficia vários públicos, por exemplo, deficientes físicos, gestantes e idosos em relação ao fácil manuseio. Outra vantagem é a redução de custos, pois é possível economizar energia, ao desligar sistemas remotamente, água, em sistemas de reutilização, ou até mesmo temporizar alguns dispositivos dependendo da necessidade.

Com o advento dos *smartphones* e a sua integralização na automatização, a domótica se tornou um processo acessível e prático. Então os processos automatizados podem ser monitorados pelo cliente via *smartphone* ou por qualquer outro dispositivo que permita interatividade através de uma interface homem-máquina.

Portanto, este trabalho se baseia em sistemas automatizados em uma maquete-casa via *smartphone* para evidenciar alguns benefícios que a domótica traz consigo como: segurança, acessibilidade, economia e praticidade.

## **1.1 Objetivo Geral**

Com o intuito de simular um ambiente automatizado, o objetivo central deste trabalho será controlar e monitorar uma maquete de madeira via *smartphone* utilizando um controlador.

### **1.1.1 Objetivos Específicos**

Ainda serão exigidos objetivos específicos como:

- Instrumentação para a maquete de madeira.
- Programação do *arduino* em linguagem C.
- Configuração e habilitação de uma rede local e *wireless* a partir do conjunto *arduino* e *ethernet shield*.
- Criação do aplicativo para *smartphone* através da ferramenta *MIT Inventor* para atuação na maquete.

## **2. Revisão Bibliográfica**

### **2.1 Automação Residencial**

Ao se construir uma residência, seja ela casa ou apartamento, raramente é lembrado pelo engenheiro ou arquiteto da obra sobre itens específicos na casa a automatizar. Isto acontece seja por redução de custos, às vezes por esquecimento ou até mesmo por não ciência.

Com o avanço da tecnologia e suas diversas áreas de atuação a automação residencial entrou nos recintos familiares com o objetivo de trazer, com a modernidade, conforto em um contexto amplo (Lecheta, 2006). Portanto, esse conforto pode vir por sistemas de segurança, sistemas inteligentes de reconhecimento de usuário, sistemas de recuperação de água ou detecção de presença para acionamento de alarmes, sistema de iluminação acionados por *tablets*, *smartphones* e afins. Tudo isso envolve a automatização de uma casa. Então pode se perceber que a utilização simultânea entre a eletricidade, eletrônica, informática e arquitetura contribuem para uma casa confortável, econômica, segura e extremamente interativa a partir de uma interface homem-máquina. Por exemplo, após o proprietário de um apartamento chegar de um dia estressante do trabalho esquece suas chaves da casa dentro do carro. O que normalmente aconteceria é à volta ao seu carro com o intuito de recuperar as chaves. Mas o sistema biométrico, sistema de reconhecimento facial ou de voz evita essa “dor de cabeça”. O proprietário consegue entrar em sua casa lhe poupando transtornos e até energia elétrica das lâmpadas que seriam gastas caso a casa não fosse automatizada.

Em vista disso, percebe-se então a relevância que a automatização possui em uma casa, um apartamento, ou qualquer ambiente residencial, que são locais propícios para a robótica, informática e a eletrônica atuarem para tornar a domótica presente.

## 2.2 Arduino

O *arduino* é um componente eletrônico que pode ser enunciado desta maneira:

O *arduino* pode ser utilizado para desenvolver objetos interativos independentes, ou pode ser conectado a um computador, a uma rede, ou até mesmo à Internet para recuperar e enviar dados do *arduino* e atuar sobre eles. Em outras palavras, ele pode enviar um conjunto de dados recebidos de alguns sensores para um site, dados estes que poderão, assim, ser exibidos na forma de um gráfico (Michael McRoberts, 2011).

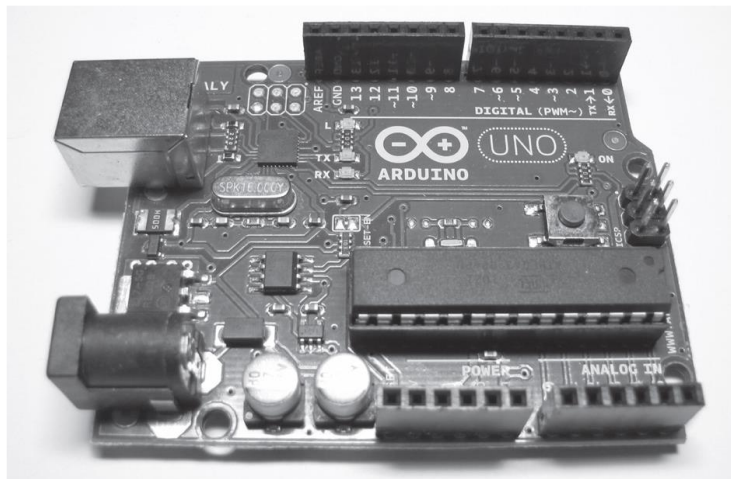
O *arduino* é uma placa microcontrolada *open source* iniciada em 2005 com ascensão no mercado até os dias atuais. Isso decorre do seu fácil manuseio e sua alta flexibilidade. Pessoas de pouco conhecimento técnico conseguem manusear este dispositivo.

Os tipos de *arduino* são muitos, mas os que merecem mais atenção são o *arduino UNO* e *arduino MEGA* devido a sua maior utilização. *Arduinos* conseguem ter várias aplicações e, portanto, podem ser conectados a *LEDs*, *displays*, botões, interruptores, motores, sensores de temperatura, de pressão, de distância, receptores *GPS*, módulos *Ethernet* ou qualquer outro dispositivo que possa ser controlado. Geralmente, um dispositivo *arduino* possui microprocessador *Atmel AVR*, um cristal ou oscilador. A placa, também, possui pinos de entrada/saída do microcontrolador, a fim de fazer ligações com outros circuitos. Com esse tipo de circuito de acordo com (Massimo Banzi, 2008) um dos fundadores do *arduino*, este dispositivo, além das aplicações habituais, ainda consegue dar uso novamente às máquinas antigas esquecidas e maior animação aos brinquedos das crianças.

### 2.2.1 Arduino Uno

Existem alguns *arduin*os no mercado que já ultrapassaram em termos de tecnologia o *arduino UNO* que era a versão mais recente dos *arduin*os até a década passada (Michael McRoberts, 2011). Entretanto o *arduino UNO* pode ser encontrado na maioria dos circuitos automatizados pelo fato de atender as exigências dos sistemas a serem automatizados através da sua configuração básica. Sendo assim, o *arduino UNO* é considerado o mais versátil e, portanto, o mais popular dos *arduin*os disponíveis no mercado. Observe a figura de um *arduino UNO* a seguir:

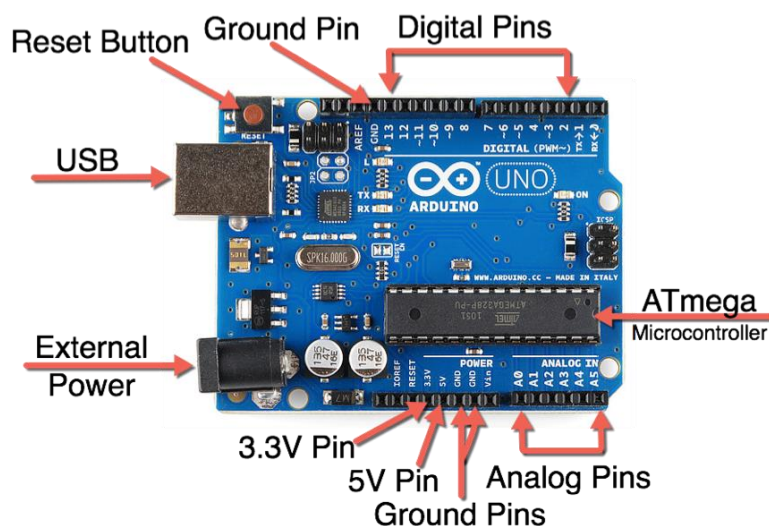
Figura 1 - Arduino UNO



Fonte:MCROBERTS, M. Arduino Básico. 1º Edição em Português. Novatec Editora, Setembro/2011

O *arduino UNO*, portanto, é uma placa microcontrolada baseada no *chip ATmega328*, com 14 pinos digitais de entrada/saída. Sua tensão de operação é de 5 volts sendo a tensão de entrada recomendada é de 7-12V e a tensão de entrada limite é de 6-20V. Possui também seis pinos analógicos, memória *flash ATmega328* e a frequência do seu *clock* de 16MHz. A figura abaixo ilustra, sucintamente, os componentes físicos de maior relevância em um *arduino UNO*.

Figura 2 - Componentes básicos do arduino UNO.



Fonte: <https://www.ardusat.com/lessons/121>.

O *arduino UNO*, como pode ser visto na figura 2, possui várias entradas e saídas como os pinos de alimentação  $V_{in}$ , 3,3V, 5V e  $GND$ , os pinos analógicos usados na

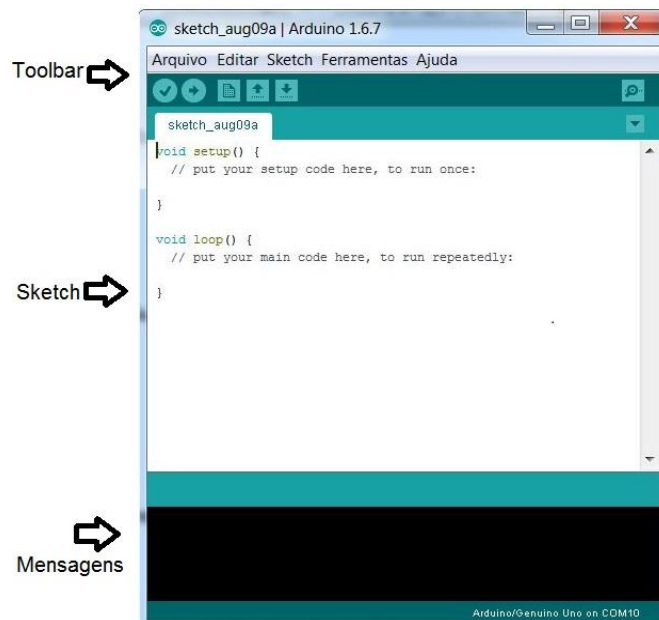


leitura de sinais de sensores, os digitais usados para detecção ou transmissão de controles digitais e o portal *USB* usado na comunicação serial com um computador.

## 2.2.2 Ambiente de Desenvolvimento Integrado

O ambiente de desenvolvimento integrado do *arduino* ou *arduino software* (*IDE-Integrated Development Environment*) é o programa destinado à criação de esboços para a placa *arduino*. Com utilização da linguagem C este programa dá “vida” ao *arduino* para diversas tarefas. O *IDE* é dividido em três partes: a *toolbar*, o código ou *Sketch* no centro, e por último, a janela de mensagens. Na *toolbar* é fornecida as bibliotecas necessárias para uso de diferentes periféricos ou recursos do *arduino*, exemplos de circuitos programados que ajudem no entendimento e na fixação, ferramenta de ajuda, rastreamento do programa a ser gravado no microcontrolador. O *Sketch* é o espaço destinado ao código em programação C que vai dar animação ao sistema a ser automatizado e por fim a janela de mensagens que indica ao programador o status do código, erros, compilações mal ou bem sucedidas, *status* de gravação ao microcontrolador. A Figura 3 facilita compreensão:

Figura 3 - Aparência do programa IDE para Arduino.



## 2.3 Instrumentação

Com o intuito de automatizar um ambiente, seja ele industrial, residencial, hospitalar ou até agrícola, é necessário um aparato de instrumentos com finalidade de medição, captação de dados e atuação na planta desejada. São exemplos de instrumentos: sensores, atuadores, relés térmicos, contadores, entre outros. Para os materiais de maior relevância deste trabalho, a seguir estão citadas suas características principais.

### 2.3.1 Sensor de Chuva YL-83

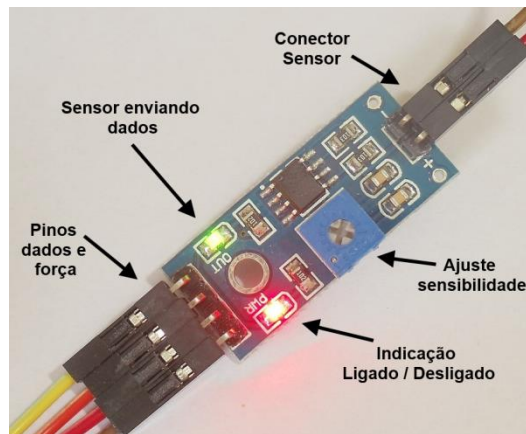
Em sua essência, o sensor YL-83 é formado por uma placa com trilhas em ambos os lados e material condutor, que se encarrega de detectar o nível de chuva/umidade do ambiente. Toda vez que é detectada umidade suficiente, as trilhas do sensor YL-83 entram em curto, enviando um sinal elétrico para o módulo de controle que é conectado a placa sensor por dois pinos. Após, identificado o sinal pelo módulo de controle é possível atribuir funções de acordo com o projeto desejado. Pode se identificar níveis de umidade pela saída analógica, A0, ou acionar motores para fechar janelas através da saída digital, D0. Os pinos de GND e  $V_{CC}$  totalizam as quatro conexões do módulo de controle para ligação com o controlador principal, neste caso, o *arduino*. Os valores de alimentação variam de 3.3 a 5 volts. Figura 4 e 5 facilitam a compreensão do sensor em questão.

Figura 4 - Conjunto de conexão. Da esquerda para direita: fios conectores, sensor de chuva (YL-83) e módulo de controle.



Fonte: <http://blog.filipeflop.com/sensores/sensor-de-chuva-yl-83.html>

Figura 5 - Módulo de controle.



Fonte: <http://blog.filipeflop.com/sensores/sensor-de-chuva-yl-83.html>

### 2.3.2 Motores DC

De acordo com (Aguirre, H. A.) motores de corrente contínua são muito utilizados em diversas aplicações devido à relativa facilidade de atribuir funções de controle de velocidade e posição. Como o *arduino* não consegue fornecer uma função de saída de tensão contínua variável, é utilizada a função *PWM* para simular uma fonte de tensão contínua variável. A partir da alimentação de zero a cinco volts em seus terminais metálicos o motor de corrente contínua gira seu eixo na saída. Figura 6 ilustra sua configuração física.

Figura 6 - Motor DC 5V.



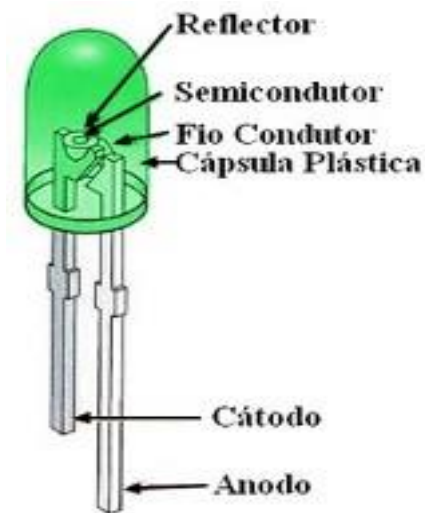
Fonte: <http://www.eng.utah.edu/~cs5789/2009/slides/motors.pdf>

### 2.3.3 Diodos Emissores de Luz (LED's)

O *LED* (Light Emitter Diode) é um semicondutor diodo que produz luz monocromática no espectro visível. Constituído por uma junção PN de material semicondutor e por dois terminais, o Ânodo(A) e o Cátodo(K), o diodo emissor de luz

sofre um deslocamento interno de elétrons livres quando se aplica uma diferença de potencial negativa do lado N e positiva do lado P. A energia dos elétrons livres ao passar pelo componente é liberada na forma de energia. Figura 7 representa um esboço do diodo emissor de luz.

Figura 7 - Configuração física do diodo emissor de luz.

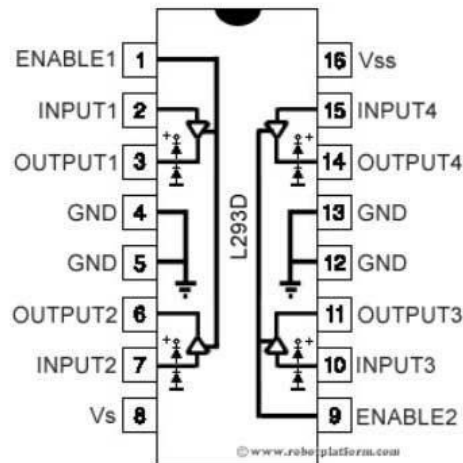


Fonte: <http://www.electronica-pt.com/led>.

### 2.3.4 Ponte H L293D

A ponte H é um módulo usado, principalmente, em robótica para a locomoção de robôs e movimentação de braços mecânicos. Este módulo, com quatro entradas (*input*), quatro pinos para *GND*, dois pinos de alimentação,  $V_s$ , e dois pinos *enable*, este dispositivo possui quatro saídas (*output*) que são canais que aceitam cargas indutivas como solenóides, motores de corrente contínua e de passo. Este dispositivo é aplicado em situações que demandam sinais com frequências de até 5kHz. Figura 8 a identificação dos seus pinos.

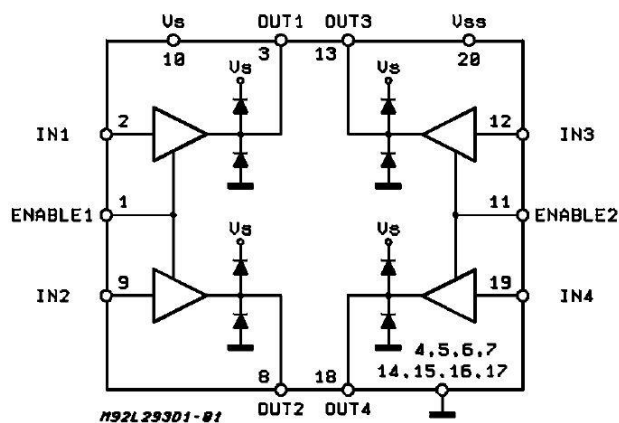
Figura 8 - Esquemático da Ponte H L293D.



Fonte: [http://www.robotplatform.com/howto/L293/motor\\_driver\\_1.html](http://www.robotplatform.com/howto/L293/motor_driver_1.html)

Outra vantagem deste dispositivo é a possibilidade de inversão de tensão sobre o componente conectado aos seus terminais de saída. Para utilizar a inversão de tensão são necessárias apenas duas entradas e duas saídas. As entradas (*input1* e *input2*) e duas saídas (*output1* e *output2*) são suficientes para gerar dois sentidos de corrente. Através da Figura 9 percebe-se que a entrada 1 (*input1*) está ligada diretamente a saída 1 (*output1*) assim como a entrada 2 está para a saída 2.

Figura 9 - Diagrama de blocos do L293D.



Fonte: <http://www.st.com/en/motor-drivers/l293d.html>.

Por meio de diferentes combinações de entrada (Pinos 1, 2 e 3), é então possível estabelecer quatro saídas (Funções) diferentes. A tabela 1 apresenta as configurações possíveis.

Tabela 1 –Estados lógicos de entrada e saída da ponte H L293D.

Pino 1	Pino 2	Pino 3	Função
Alto	Baixo	Alto	Horário
Alto	Alto	Baixo	Anti-horário
Alto	Baixo	Baixo	Parado
Alto	Alto	Alto	Parado
Baixo	Não aplicável	Não aplicável	Parado

Assim, para motores de corrente contínua ou motores de passo é possível configurar o sentido do movimento a partir da excitação nas entradas.

## **2.4 Comunicação dos dispositivos**

### **2.4.1 Tecnologia Ethernet/Wireless**

Em uma comunicação de rede existe um protocolo numérico de endereços análogo ao código endereçamento postal (*CEP*), o mesmo usado pelos correios para direcionar as correspondências. Este protocolo é conhecido pela sigla IP (*Internet Protocol*) e é dessa forma que é possível identificar os dispositivos em uma rede.

Originalmente criada com uma taxa de 2,94Mbps, o padrão ethernet consiste de uma rede local (*Local Area Network*), ou seja, dispositivos que estão conectados por uma rede física onde as estações/dispositivos possuem entre si uma proximidade do ponto de vista geográfico. Para sua padronização homologou-se ao IEEE – *Institute of Electrical and Electronic Engineers*, em 1980, a regulamentação do padrão Ethernet. De acordo com (Pinheiro, J. M. S.) este padrão possui uma subcamada de Controle de Acesso ao Meio (*MAC*) que controla a transmissão, a recepção e atua diretamente com o meio físico.

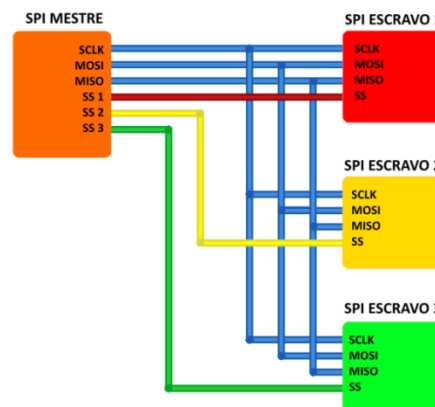
Muitos sistemas de comunicação utilizam fios, cabos ou, fibras ópticas para transmitir os dados. Entretanto, outros são implementados sem a utilização destes. Essas redes são chamadas *Wireless* e utilizam o ar como meio físico para transmitir as suas informações. São utilizadas ondas eletromagnéticas para enviar e receber as

informações ao invés de cabos e, dependendo da sua abrangência podem ser classificadas como WPAN(*Wireless Personal Area Network*), WLAN(*Wireless Local Area Network*) e WWAN(*Wireless Wide Area Network*).

## 2.4.2 SPI (Serial Peripheral Interface)

*Serial Peripheral Interface* é um protocolo de transferência de dados seriais síncronos, usado por controladores e dispositivos mestres, para troca de informações para dispositivos escravos, de forma rápida em pequenas distâncias. A Figura 10 apresenta uma conexão *SPI* entre um dispositivo mestre e três dispositivos escravos.

Figura 10 - Ligação dos pinos entre o dispositivo mestre e os escravos.



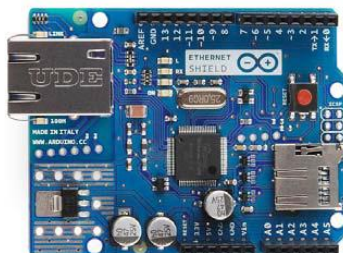
Fonte: <http://buildbot.com.br/blog/interfaces-de-comunicacao-onewire-spi-i2c/>

O que acontece nessa comunicação é o envio (*MOSI - Master Output Slave Input*) e recebimento (*MISO - Master Input Slave Output*) de dados do dispositivo mestre para os dispositivos escravos a partir do sinal de clock (*SCLK - Serial Clock*). O último pino (SS) é responsável pela seleção do escravo (*Slave Select*). O *Arduino* dedica quatro pinos para a comunicação desse tipo de protocolo, sendo o mesmo o mestre, através dos pinos 10, 11, 12 e 13, enquanto, o *Ethernet Shield* é o dispositivo escravo.

## 2.4.3 Ethernet Shield

Este módulo, *Ethernet Shield W5100*, permite compartilhar as informações processadas pelo *arduino UNO* e disponibiliza-las para uma rede comum. A Figura 11 apresenta tal dispositivo:

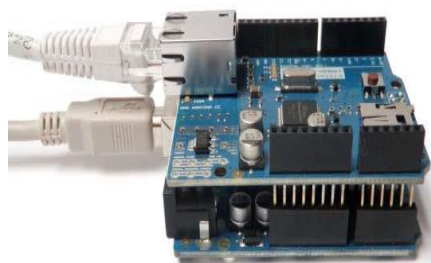
Figura 11 - Ethernet Shield W5100.



Fonte: XXI Simpósio Nacional de Ensino Física – SNEF 2015.

Com a sobreposição do *Ethernet Shield* ao *Arduino* é possível transmitir dados através da rede escolhida, seja esta uma rede local (*LAN*) ou uma rede *Wireless*. Além disso, este dispositivo permite expandir as funções do *arduino UNO* ou de qualquer outro componente que possa ser acoplado. Entretanto, a quantidade de saídas digitais, no caso do *arduino UNO*, é então reduzida de 14 (visto no item 2.2.1) para 10 portas digitais. A Figura 12 apresenta a disposição física do acoplamento do *Ethernet Shield W5100* (placa superior) ao *Arduino UNO* (placa inferior). Importante observar, também, que o cabo de rede deve ser conectado ao *Shield*, enquanto, que o *Arduino UNO*, neste caso esta energizado pelo cabo *USB*.

Figura 12 - Ethernet Shield W5100 sobreposto ao arduino UNO.



Fonte: XXI Simpósio Nacional de Ensino Física – SNEF 2015.



A comunicação entre o *Ethernet Shield* e o *Arduino UNO* é feita através do barramento SPI (*Serial Peripheral Interface*), através dos pinos 10, 11, 12 e 13.

## **2.5 Uso das Ferramentas para Controle de Processos**

### **2.5.1 Smartphones**

O primeiro *smartphone* segundo (McCarty, 2011) foi desenvolvido, em 1992, pela *IBM (International Business Machines)*. Este dispositivo possuía uma tela *touchscreen*, com capacidade de envio e recebimento de mensagens de fax e e-mails. Entretanto, apenas em 2000 que o termo *smartphone* foi utilizado pela primeira vez de acordo com (Coutinho, G. L., 2014).

Segundo (Morimoto, C. E, 2009) não existe um consenso na indústria sobre quais exigências são necessárias para se definir um *smartphone*. Entretanto os “telefones inteligentes” ou *smartphones* atuais possuem algumas características comuns, como por exemplo: câmeras acopladas, teclados virtuais, telas *touchscreen*, comunicação por *GPS, Bluetooth, Wifi e mobile data*.

A partir de todos esses recursos disponíveis em *smartphones* e sua ampla adesão pelos consumidores, estes dispositivos utilizam aplicativos como ferramentas de controle de processos em diversas áreas: industriais, residências, prediais e comerciais.

### **2.5.2 MIT App Inventor**

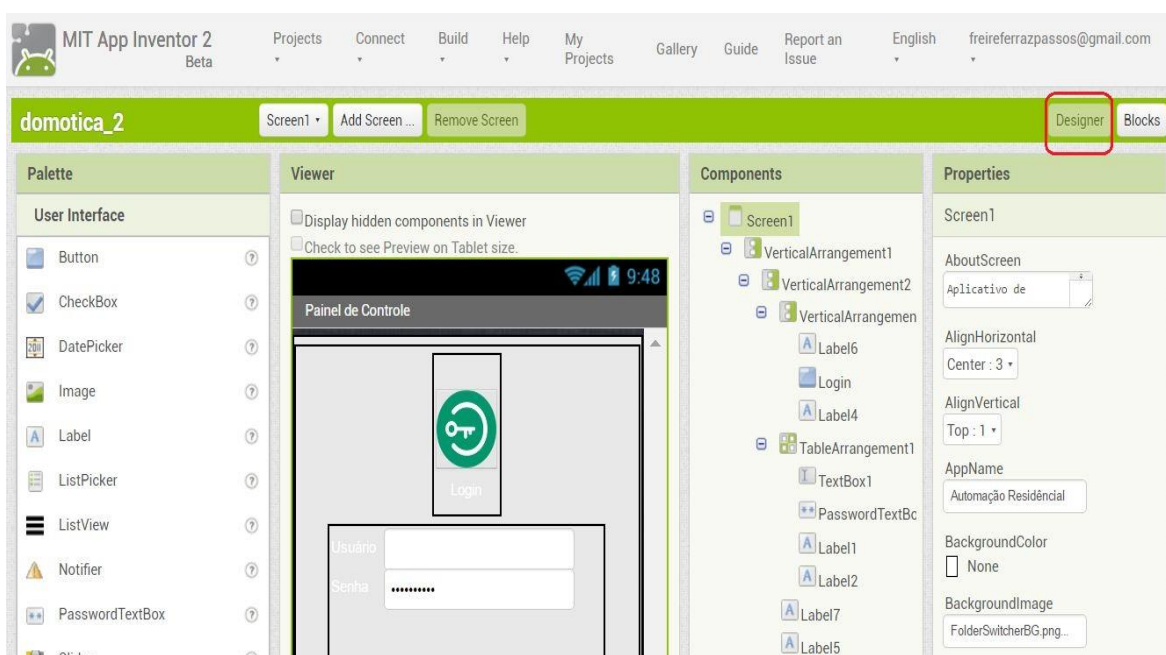
Originalmente disponibilizada pela *Google* e repassada ao MIT (*Massachusetts Institute of Technology*), o *App Inventor* é uma plataforma de criação de aplicativos para dispositivos com sistema operacional *Android*.

A interface gráfica de programação desta ferramenta utiliza a biblioteca *Java* de código aberto *Open Blocks* para criação de um ambiente visual de programação, semelhante a um diagrama de blocos. Segundo (Wolber, 2011) o objetivo da plataforma é habilitar o usuário a “unir” esses blocos em uma espécie de “quebra cabeças”, para a

finalidade da sua aplicação. Apesar da interface intuitiva, a ferramenta requer conhecimentos lógicos de programação e um estudo detalhado dos blocos disponíveis no sistema.

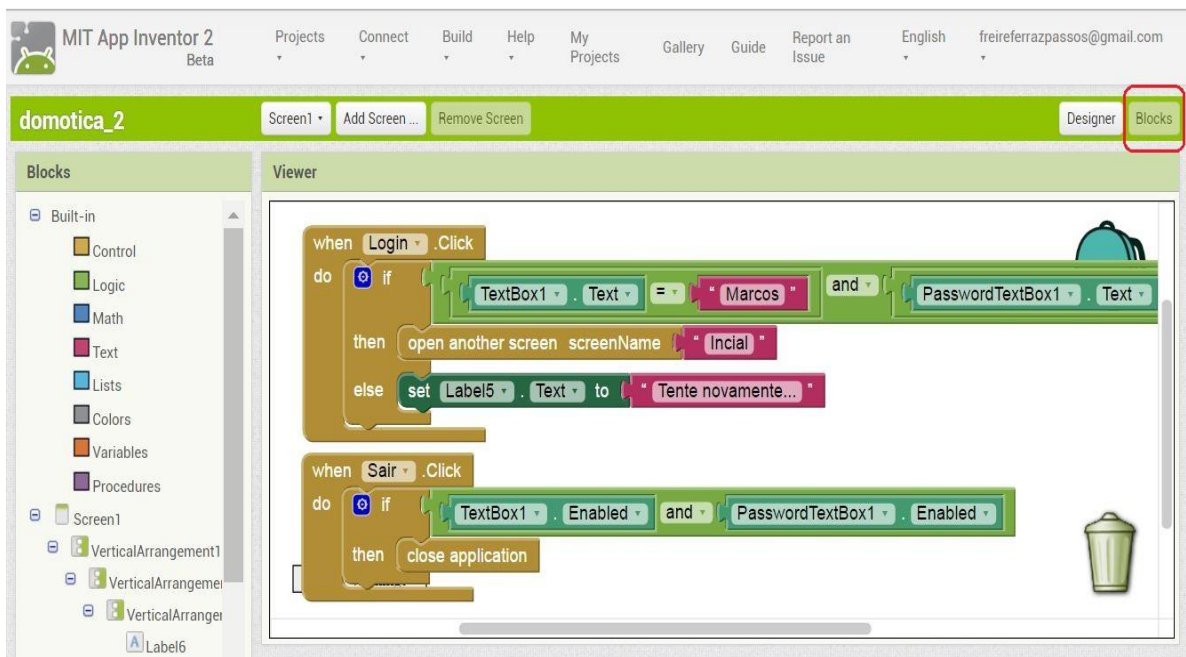
Para acessar a plataforma e iniciar algum projeto é necessário possuir uma conta de email da *Google*. Após criado um projeto pode se iniciar sua construção e então configurar o aplicativo da maneira que se desejar. Como podem ser visualizadas nas Figuras 13 e 14, existem duas janelas, uma para o desenho da tela (*designer*) e a outra destinada à programação em blocos (*blocks*). Estas devem concordar, ao menos, em lógica para que o aplicativo funcione dentro das normalidades. Para a janela de *designer* existe uma aba de ferramentas que podem ser inseridos sensores, *buttons*, *sliders*, *textbox*, *labels*, *speechrecognizers*, *layers* entre outros. É neste espaço que todo layer da tela será montado.

Figura 13 - Janela de designer do MIT Inventor.



Para a segunda janela (Figura 14), é possível construir o aplicativo em programação de blocos. Existem, na aba a esquerda, algumas funções que possibilitam construir o programa a partir de controle, funções matemáticas, funções lógicas como portas lógicas *AND* e *OR*.

Figura 14 - Janela de blocks do MIT Inventor.



### 3 *Materiais e Métodos*

Para automatizar a maquete neste trabalho dividiu se a mesma em três sistemas: a maquete da casa que envolve a construção com base em madeira, os sensores, os motores, o *arduino*, o *ethernet shield* e o resto dos componentes eletrônicos para o sistema físico; o roteador, cabo de rede, o servidor, a rede wireless e local (*WLAN*) e o programa feito no ambiente *IDE arduino* para o sistema de comunicação; e, por fim, o sistema de controle e monitoramento composto pelo aplicativo criado para possibilitar a atuação na planta por intermédio de um celular. Todos esses sistemas não trabalham de maneira independente, pois cada um deve fornecer uma resposta a partir de estímulos advindos de outros sistemas. Esses sistemas serão vistos com maior detalhamento nos tópicos subsequentes.

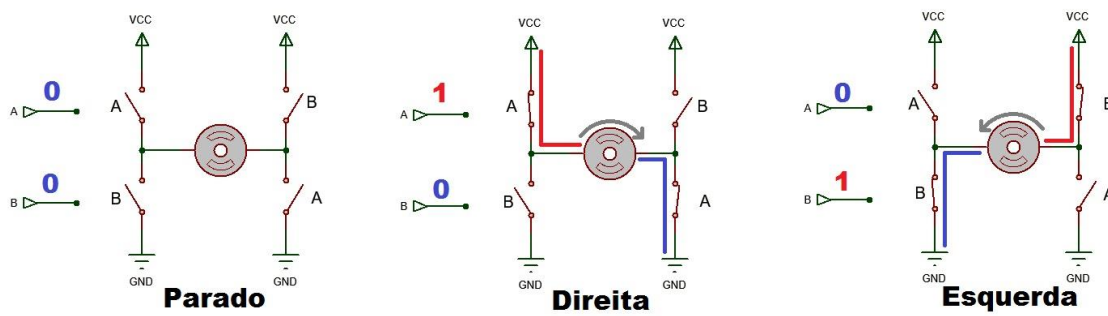
### 3.1 Instrumentação e Implementação do Protótipo Residencial

Para a confecção e funcionamento da maquete foram utilizados os seguintes materiais:

- Uma base de madeira de 650 x 490  $mm^2$
- Pallets de madeira para as paredes da maquete
- Placa microcontrolada *Arduino UNO*
- Módulo *Arduino Ethernet Shield W5100*
- Sensor de Chuva e Umidade YI-83
- Led
- Resistores
- Quatro baterias de 1,5V cada
- Três motores cc 5V
- Dois redutores de velocidade
- Duas roldanas
- Uma correia para conectar as roldanas
- Ponte H L293D
- Jumpers
- Uma cola madeira e uma cola instantânea especial
- Pregos e parafusos
- Cabo de rede
- Cabo de alimentação para o *arduino*
- Um roteador
- Arames de ferro

O sensor de chuva foi utilizado para deixar ciente o usuário sobre a ocorrência de chuva. Toda vez que o sensor for molhado as partículas de água farão com que o sensor entre em curto circuito, ou seja, farão a ponte de ligação entre as trilhas, e assim, o sensor identificará a manifestação da chuva. O módulo de controle identifica o nível de chuva, de forma analógica, através do pino A0 do *arduino*. Quando esta informação da ocorrência da chuva for sinalizada pelo sensor, deixará ciente o usuário através da tela destinada para o circuito da chuva e, então, este poderá acionar o motor de 5V pelo *smartphone* para fechar a janela. É possível também abrir a janela ou pará-la, se assim o usuário desejar. A partir das ligações dos pinos 6, 7 e 9 do *arduino* para as entradas da ponte H L293D (visto no item 2.3.4 ) o motor consegue inverter o sentido da corrente. Os pinos 6 e 7 são para inverter a corrente, enquanto, o pino 9 é para habilitar a ponte. A figura 15 ilustra como é possível inverter o sentido do motor com duas entradas de estado lógico que neste caso seriam as saídas da ponte H L293D para o motor. Para o funcionamento da janela foi utilizado dois redutores de velocidade, duas roldanas, uma correia e quatro baterias de 1,5V cada.

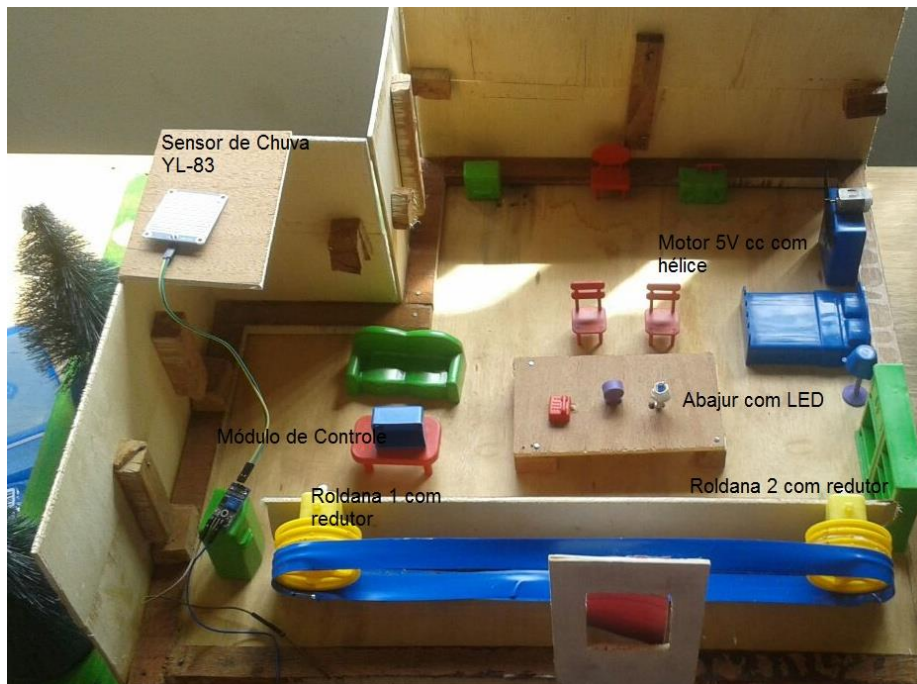
Figura 15 - Movimentos possíveis do motor de 5V.



Fonte: <http://projetomagiar.blogspot.com.br/>

Para o sistema de ventilação da casa utilizou-se um motor de 5V de corrente contínua e na sua ponta uma ventoinha acoplada para ventilar o cômodo da casa. Para este circuito deve-se ter cuidado, pois como o motor possui uma corrente alta, isto poderia danificar o *arduino* ou o *ethernet shield*. Então foi utilizada a ponte H L293D para preservar a integridade física do *arduino* e com a mesma acionar o motor. Para o sistema de luminosidade utilizou-se um *LED* ligado, juntamente, com um resistor de 10K no pino 5 do *arduino*. O acionamento destes três circuitos acontece com saídas digitais, 5V ou 0V. A Figura 16 apresenta os três circuitos ligados na maquete.

Figura 16 - Maquete com os circuitos automatizados.



## **3.2 Configuração do Arduino para as Funções da Maquete**

Para a programação no ambiente *IDE arduino* utilizou-se as bibliotecas <SPI.h> e <String.h> para a comunicação serial periférica e a leitura de strings enviadas e recebidas pelos *smartphones*, servidor *web* e o microcontrolador *arduino* acionarem os sistemas automatizados na maquete, respectivamente. A biblioteca <Ethernet.h> foi usada para disponibilizar a rede como canal de comunicação entre as diferentes partes do protótipo residencial.

O uso da comunicação *SPI* utiliza o *arduino* como mestre enquanto o *ethernet shield* como escravo. Desta forma, essa relação torna o *arduino* a parte ativa do processo, enquanto o *ethernet shield* a parte passiva, obedecendo sem contestar as tarefas incumbidas pelo primeiro.

Sobre a leitura de strings o processo se caracteriza por ser bidirecional, portanto, o envio de strings para URL que é a página habilitada como servidora acontece nos dois sentidos. Além disso, foi criado um cliente para que, a partir da interface gráfica do *smartphone*, o usuário possa ser a parte ativa do processo e com isso comandar a sua escolha os processos automatizados.

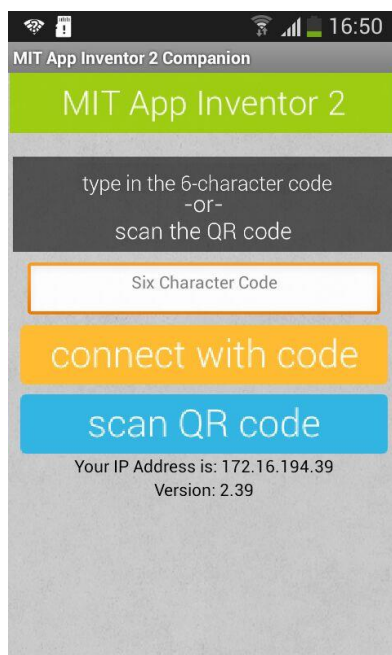
## **3.3 Configuração da Rede**

Para configurar a rede é necessário ter alguns valores, como por exemplo, os valores de *IP* que representam o endereço lógico de cada equipamento que será identificado na rede. A máscara de rede comporta-se como o endereçamento que identifica quais *IPs* pertencem aquela rede. E por fim, o *gateway* que representa o caminho de saída para outra rede. Portanto foi necessário configurar esses três valores para que os dispositivos estivessem na mesma rede e o modem provesse a comunicação.

### 3.4 O aplicativo para Automação Residencial

Utilizou-se a plataforma *MIT Inventor 2* para a criação do aplicativo da automação residencial. Terminado o programa é possível salvá-lo como “.aia” e o aplicativo pode ser baixado pelo próprio *smartphone* através do programa *MIT AI2 Companion* como arquivo “.apk”. Observe a Figura 17.

Figura 17 - Leitor QR code para baixar aplicativo.



O aplicativo possui cinco telas, sendo a primeira para login onde o usuário deve digitar corretamente usuário e senha para ter acesso as outras telas. Caso o usuário digite incorretamente será avisado para que haja uma nova tentativa através do label. Para encerrar a tela foi criado um botão para sair do aplicativo. A segunda tela é a inicial onde possui os botões que direcionam para as outras três telas: sistema de luminosidade, sistema de ventilação e sistema contra chuva. Cada uma dessas três telas tem funcionalidade no circuito. Estas possibilitam o usuário acionar os circuitos através dos botões e também monitorá-los a partir dos labels que retornam os status de cada variável. Por exemplo, ao acionar o LED pelo botão a página URL criada recebe essa informação e retornar para o smartphone a situação à qual o LED se encontra e para este caso seria “Ligado”.

Tabela 2 - Telas dos aplicativos, itens e suas funcionalidades.

<b>Telas</b>	<b>Layout</b>	<b>Função</b>
Login	Botão Login	Faz login do aplicativo
	Caixa de texto	Digitar usuário
	Password texto	Digitar senha
	Label 1	Avaliar senha e usuário
	Botão Sair	Sair do aplicativo
Inicial	Botão sistema de luminosidade	Entra na tela de controle de luminosidade
	Botão sistema de ventilação	Entra na tela de controle de ventilação
	Botão antichuva	Entra na tela de sistema antichuva
	Botão de logout	Retornar a tela de login
Sistema de Luminosidade	Botão Ligar	Liga o led
	Botão Desligar	Desliga o led
	Label	Avalia o status do led
Sistema de ventilação	Botão Ligar	Liga Ventilador
	Botão Desligar	Desliga Ventilador
	Label	Avalia status do ventilador
Sistema antichuva	Botão Parar	Para o motor
	Botão Abrir	Liga o motor em uma direção
	Botão Fechar	Liga o motor em outra direção
	Label	Avalia situação da chuva

As telas foram criadas de acordo com as opções disponíveis no aplicativo através das janelas de *designer* e *blocks* e suas funcionalidades citadas no item 2.5.2. Para que o aplicativo consiga se conectar com a rede foi adicionado o bloco de conectividade *Web* que neste caso é uma rede local.



### 3.5 Comunicação entre Aplicativo e Arduino

Com a relação cliente servidor é possível receber e enviar dados pela página criada, a *URL*, e com isso acionar dispositivos como também monitorar-los a partir de informações que advém da maquete a partir de um *IP*. Será este *IP* o destinatário das informações, comandos e leituras da maquete. Portanto a partir de comandos configurados, em comum, tanto no ambiente *IDE arduino* como no *MIT inventor* é possível criar o elo de comunicação entre as partes utilizando sempre a página *URL* criada. No ambiente *IDE* foi criada a leitura de *strings* que são coincidentes com as *strings* do aplicativo. Como pode ser visto na figura 18 cada botão se encarrega de enviar para a página criada o *IP* (“http://192.168.0.88”) com a porta (“:80”) e um tipo de informação associada: “FecharJanela”, “AbrirJanela” e “PararJanela”. Quando a página receber essa informação, esta é reenviada para o *arduino* e como pode ser visto na figura 19, os comandos relacionados a estas *strings* são realizados, finalizando o ciclo quando é atuado na planta.

Figura 18 - MIT Inventor blocks para o circuito da janela.

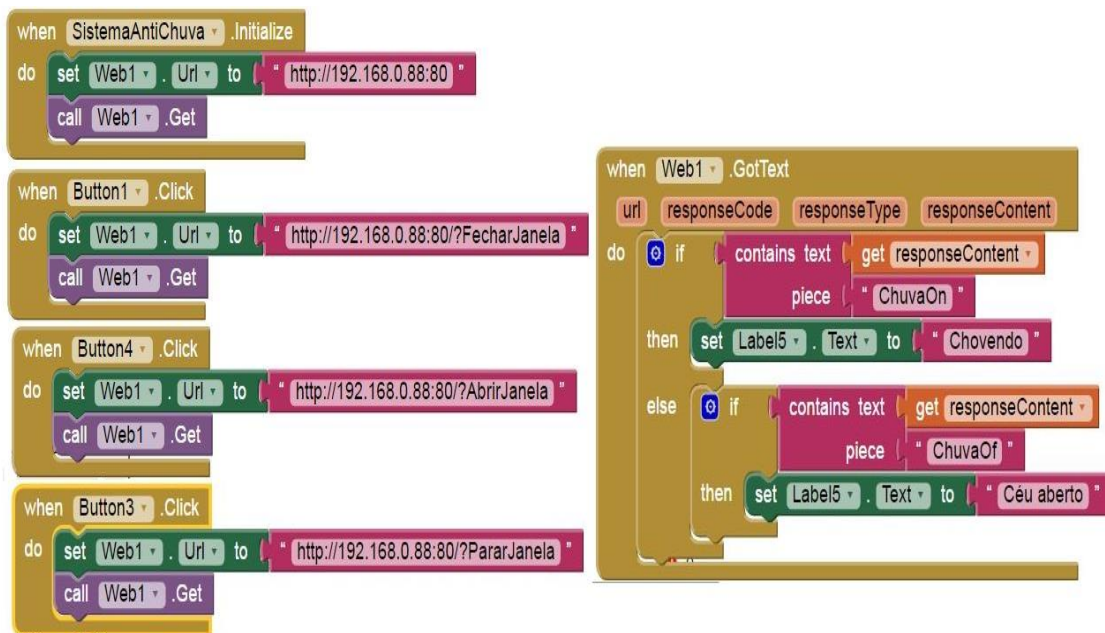


Figura 19 - Programação do circuito da janela.

The image shows a screenshot of the Arduino IDE interface. The title bar reads "programa\_final\_janela | Arduino 1.6.7". The menu bar includes "Arquivo", "Editar Sketch", "Ferramentas", and "Ajuda". Below the menu bar is a toolbar with icons for "Carregar" (Upload), "Verificar" (Verify), and "Enviar" (Send). The main text area contains the following C++ code:

```
if(readString.indexOf("PararJanela")>=0)
{
  digitalWrite(input1, LOW);
  delay(200);
  digitalWrite(input2, LOW);
  delay(200);
  statusMotor = true;
}
if(readString.indexOf("AbrirJanela")>=0)
{
  // o motor vai ser setado na direção oposta
  digitalWrite(input1, HIGH);
  digitalWrite(input2, LOW);
  statusright = true;
}
if(readString.indexOf("FecharJanela")>=0)
{
  // o motor vai ser setado na direção oposta
  digitalWrite(input2, HIGH);
  digitalWrite(input1, LOW);
  statusleft = true;
}
```

## 4 Resultados

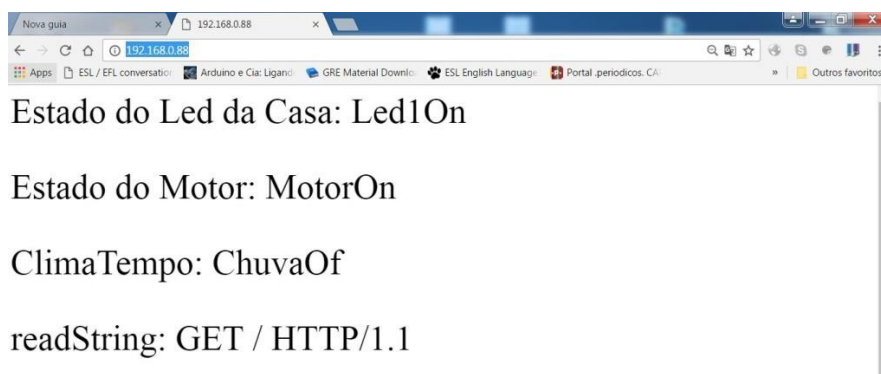
Obteve se resultados satisfatórios dos acionamentos dos três circuitos, somente, quando foi estabelecido um status para a ação de cada tarefa. Como exemplo, o acionamento do *LED* aconteceu, somente, após associar um *status* à situação do *LED*. Então, com a leitura da *string* “LigarLed1” junto com a atualização do *LED* do estado baixo para alto foi possível ligá-lo. Para o restante dos circuitos foi utilizada essa lógica, contudo, para o circuito da janela foi utilizado três status, já que para este circuito não são necessários, somente, uma lógica binária, mas sim mais de duas (“abrir”, “fechar” e “parar”). A Figura 19 ilustra a programação para este artifício.

Também pôde se comprovar a partir de ligações nas portas 10, 11, 12 e 13 que estas são reservadas para a comunicação entre o *ethernet shield* e o *arduino*. Esta é uma configuração padrão, portanto não é necessário programar esses pinos pelo *IDE arduino*. Entretanto, como feito na maquete, inicialmente, utilizou se esses pinos para acionamento dos motores e obteve se resultados não esperados. Como esses pinos são

destinados para a comunicação serial periférica os mesmos não devem ser utilizados. Após substituir os pinos por outros, as ligações dos circuitos funcionaram como esperado, o que comprova que os pinos 10, 11, 12 e 13 são para uso exclusivo do *ethernet shield* e que estes não devem ser utilizados enquanto existir comunicação *SPI*.

A resposta para a criação do *IP* e da porta no ambiente *IDE* foi à hospedagem desta página na rede local. Com isso, existe agora uma página onde é encarregada de receber as informações enviadas pelo *ethernet shield*. A Figura 20 mostra a página servidora que mostra algumas funções usadas no programa.

Figura 20 - Página atualizada com IP do computador.



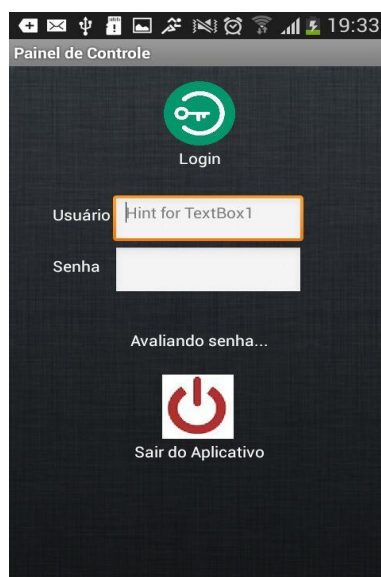
Na Figura 21 pode se visualizar o aplicativo programado chamado “Automação Residencial” e o *MIT AI2 companion* que é o programa responsável por baixa o aplicativo já criado.

Figura 21 - Tela do Smartphone.



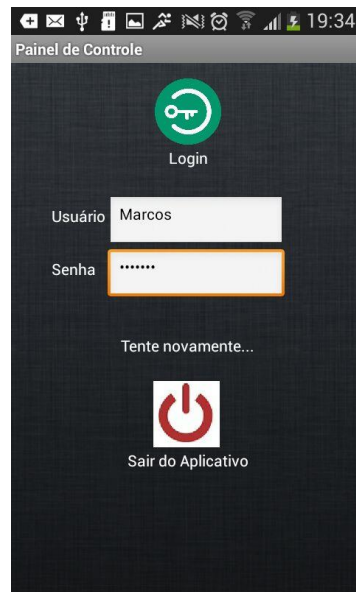
Após inicializar o aplicativo “Automação Residencial”, o programa irá entrar na tela *login*. Nesta tela existe o botão de *login* que entra na tela inicial, desde que, o usuário e senha sejam digitados corretamente. Foi escolhido usuário “Marcos” e senha “Domótica”, porém no *MIT inventor* é possível escolher uma senha e usuário alfanuméricos a sua escolha, basta apenas, programar na aba *blocks* os nomes desejados. Observe a Figura 22.

Figura 22 - Tela login.



Caso o usuário não digite os caracteres corretamente a tela *login* irá pedir para o usuário “tentar novamente” pelo *label* que está localizado abaixo do espaço destinado a senha. Com esta página o usuário restringe quem poderá atuar na planta e assim terá mais segurança. Observe a Figura 23 a seguir.

Figura 23 - Login requisitado incorretamente.



Existe também um botão para sair do aplicativo como pode ser visualizado nas duas últimas figuras mostradas.

Após efetuar o *login* corretamente o aplicativo irá para tela inicial que está ilustrada na próxima Figura 24.

Figura 24 - Tela inicial



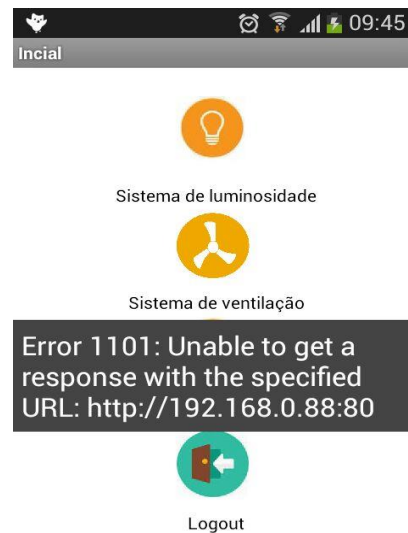
O botão *logout* retorna a página de *login* enquanto todos os outros botões levam para o controle e monitoramento dos circuitos de iluminação, ventilação e antichuva. As telas de cada circuito já citadas podem ser visualizadas na Figura 25.

Figura 25 - Telas dos circuitos.



Se a utilização do aplicativo acontecer sem a conexão da rede o *smartphone* acusará perda de resposta, assim como na Figura 26.

Figura 26 - Aplicativo com perda de informação.



## 5 *Conclusão*

Com o desenvolvimento de novas tecnologias a partir de interfaces amigáveis e de fácil manuseio e o aumento do acesso aos *smartphones* tem se uma maior aceitação por parte da população a doméstica. Portanto, a partir de todas essas ferramentas torna se acessível controlar e monitorar processos automatizados em uma maquete.

Os equipamentos como sensor e os circuitos atuadores funcionaram como esperado e suas respostas podem ser estendidas para circuitos reais, basta o projetista se atinar para a utilização de relés para a proteção dos dispositivos de baixa potência como: *arduino*, *ethernet shield* e outros.

As ferramentas que o ambiente *IDE* possui para *arduino*, como as bibliotecas, auxiliam as diferentes aplicações nas quais o usuário se interessa. Portanto o *arduino* consegue ser implementado em diversos circuitos.

A utilização de uma rede em processos automatizados abre possibilidade de adicionar mais dispositivos à mesma, aumentando a malha de equipamentos relacionados e a forma como trafegar as informações.

O aplicativo criado se mostrou satisfatório apesar de identificado alguns problemas no *App Inventor* por não responder alguns comandos de programação. Mesmo assim, essa ferramenta online se mostrou eficaz e consegue abordar uma grande possibilidade de circuitos. Este trabalho também alcançou teor de segurança, pois a página de *login* restringe a utilização apenas àqueles que detêm as informações para usuário e senha.

Outro ponto importante que vale ser citado é em relação à aplicação deste tipo de trabalho. Existem vários espaços que admitem automatização. Algumas empresas de aviação já possuem pretensão de automatizar persianas, iluminação das cabines, solicitar atendentes e até inclinar poltronas de aeronaves. Pode se considerar este trabalho, portanto, prático, porque o ambiente automatizado pode ser aquele ao qual o usuário desejar, sendo assim, terá várias aplicações em diversos meios.

Em trabalhos futuros pode se configurar o *Ethernet Shield* para a rede de dimensões maiores, ao invés, de uma rede *LAN* para uma rede *WAN*. Desse modo, o usuário poderá fazer uso do seu ambiente automatizado, remotamente, e assim trará mais conforto ao usuário quando ausente. Outro circuito que poderia ser automatizado é o circuito por comando de voz sendo outra ferramenta de fácil e cômodo acesso aos usuários, limitando se apenas a pessoas mudas. Em contrapartida seriam úteis para deficientes motores, como pessoas manetas.



## ***Referência Bibliográfica***

FERREIRA, V. Z. G. *A Domótica como instrumento para a melhoria da qualidade de vida dos portadores de deficiência*. 2010. Tese (Trabalho de Conclusão de Curso) – Curso Superior de Tecnologia em Automação Industrial do Instituto Federal de Educação, Ciência e Tecnológica da Paraíba.

CHAMUSCA, A. *Domótica & Segurança eletrônica – a inteligência que se instala*, Lisboa, Ordem dos Engenheiros/ Ingenium Edições, Lda. 2006.

MICHELL, T. *Machine Learning*. McGraw Hill, 1997. 414 p.

VIVAS, R. N., *Controle e Dispositivos em Domótica via Smartphone*, Viçosa, 2015.

LECHETA, L. P., *Sistemas de Iluminação Residencial: uma Análise sobre Alternativas para redução do consumo de Energia*. Cascavel, 2006.

MICROBERTS, M. *Arduino Básico*. 1º Edição em Português. Novatec Editora, Setembro/2011, 452 páginas.

BANZI, M. *Getting Started with Arduino*. 2nd edition. September 2011. 115 pages.

AGUIRRE, H. de A. R. *Implementação de controle digital de posição do motor CC em tempo real utilizando MATLAB e SIMULINK*. São Carlos, 2012.

PINHEIRO, J.M.S.. *Guia Completo de Cabeamento de Redes*. 1 ed. Rio de Janeiro: Campus, 2003.

MCCARTY, Brad. *The History of the Smartphone*. 2011. Disponível em: . Acesso em: 23 Nov. 2016.

COUTINHO, G. L. *A Era dos Smartphones: Um estudo Explanatório sobre o uso dos Smartphones no Brasil*. Brasília DF, 2014.

MORIMOTO, C. E. *Smartphone Guia Prático*. In: MORIMOTO, C. E. *Smartphone Guia Prático*. 1ª. ed. Porto Alegre: Meridional Ltda., 2009. p. 432.

WOLBER, D., ABELSON H., SPERTUS, E. & LOONEY, L. *App Inventor Create Your Own Android Apps*. O'Reilly Media. April 2011.