

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE MANUTENÇÃO INDUSTRIAL
CURSO DE TECNOLOGIA EM MANUTENÇÃO INDUSTRIAL**

**LUCAS ANDREY BERTONI
NADILSON BORTOLUZZI**

**AUTOMAÇÃO RESIDENCIAL EM ESCALA REDUZIDA UTILIZANDO
SISTEMA MICROCONTROLADO**

TRABALHO DE CONCLUSÃO DE CURSO

**MEDIANEIRA
2016**

LUCAS ANDREY BERTONI
NADILSON BORTOLUZZI

**AUTOMAÇÃO RESIDENCIAL EM ESCALA REDUZIDA UTILIZANDO
SISTEMA MICROCONTROLADO**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Diplomação, do Curso de Tecnologia em Manutenção Industrial da Universidade Tecnológica Federal do Paraná, Câmpus Medianeira, como requisito parcial à obtenção do título de Tecnólogo em Manutenção Industrial.

Orientador: Prof. Me. Alex Lemes Guedes
Coorientador: Prof. Me. Filipe Marangoni

MEDIANEIRA
2016



TERMO DE APROVAÇÃO

AUTOMAÇÃO RESIDENCIAL EM ESCALA REDUZIDA UTILIZANDO SISTEMA MICROCONTROLADO

Por:
Lucas Andrey Bertoni
Nadilson Bortoluzzi

Este Trabalho de Conclusão de Curso (TCC) foi apresentado às 20:30 h do dia 28 de novembro de 2016 como requisito parcial para a obtenção do título de Tecnólogo no Curso Superior de Tecnologia em Manutenção Industrial, da Universidade Tecnológica Federal do Paraná, câmpus Medianeira. Os acadêmicos foram arguidos pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Me. Alex Lemes Guedes
UTFPR – Câmpus Medianeira
(Orientador)

Prof. Me. Filipe Marangoni
UTFPR – Câmpus Medianeira
(Coorientador)

Prof. Me. João Felipe Montemezzo
UTFPR – Câmpus Medianeira
(Convidado)

Prof. Me. Yuri Ferruzzi
UTFPR – Câmpus Medianeira
(Convidado)

Prof. Me. Paulo Job Brenneisen
UTFPR – Câmpus Medianeira
(Responsável pelas atividades de TCC)

A Folha de Aprovação assinada encontra-se na coordenação do Curso de Tecnologia em Manutenção Industrial.

AGRADECIMENTO

Gostaríamos de agradecer todos os nossos familiares, amigos e colegas pela ajuda, paciência e compreensão não só durante todo o período de realização deste projeto, como ao longo do curso e de nossas vidas.

RESUMO

BERTONI, Lucas Andrey; BORTOLUZZI, Nadilson. **Automação residencial em escala reduzida utilizando sistema microcontrolado**. 2016. 44 f. Trabalho de Conclusão de Curso (Tecnólogo em Manutenção Industrial) – Universidade Tecnológica Federal do Paraná. Medianeira, 2016.

O objetivo deste trabalho é desenvolver uma maquete para demonstrar a construção e o funcionamento um sistema de automação residencial utilizando o Arduino Mega 2560 e outros componentes eletrônicos. O sistema todo pode ser configurado de acordo com as necessidades de cada projeto, que neste caso foram o controle de iluminação, o sistema de alarme, monitoramento da temperatura da água da piscina e abertura e fechamento do portão. A interface utilizada para o controle pode ser uma página da web ou o aplicativo que é executado em sistema operacional Android além do controle de alguns processos que pode ser feito também fisicamente. O desenvolvimento do aplicativo demanda conhecimento específico em programação e noções básicas em linguagem C e C++ utilizando a aplicação APP INVERTOR 2, que é um programa totalmente gratuito desenvolvido pela empresa Google. A maquete da residência foi construída em MDF com uma gaveta na parte inferior para ser armazenada toda a parte elétrica. O protótipo funcionou adequadamente de acordo com o projeto.

Palavras-chave: Domótica. Arduino. Ethernet Shield. Android. App Inventor.

ABSTRACT

BERTONI, Lucas Andrey; BORTOLUZZI, Nadilson. **Small-scale residential automation by using microcontrolled system**. 2016. 44 f. Trabalho de Conclusão de Curso (Tecnólogo em Manutenção Industrial) – Universidade Tecnológica Federal do Paraná. Medianeira, 2016.

The purpose of this work is to develop model to demonstrate the construction and operation of a residential automation system using the Arduino Mega 2560 and other electronic components. The system can be configured according to the needs of each project, which in this case was control of lighting, alarm system, monitoring of the temperature of the pool water and opening and closing of the gate. An interface used for the control can be a web page or application that runs on an Android operating system beyond the control of some processes that can be done to physically. App development requires specific knowledge of programming and C and C++ language fundamentals. The model of the residence was built in MDF with a lower part to be stored the entire electrical part. The prototype worked properly according to the project.

Keywords: Home Automation. Arduino. Ethernet Shield. Android. App Inventor.

LISTA DE FIGURAS

| | |
|---|----|
| Figura 1 – Arduino Mega 2560 R3..... | 13 |
| Figura 2 – Interface de Designer (a) e edição de Blocos (b) | 14 |
| Figura 3 – Ethernet Shield..... | 15 |
| Figura 4 – IDE Arduino | 16 |
| Figura 5 – Sensor digital ds18b20..... | 17 |
| Figura 6 – Sensor de corrente SCT013-030A. | 17 |
| Figura 7 – Sensor PIR | 18 |
| Figura 8 – Esquema de funcionamento de uma ponte H (a) e Módulo I9110 (b) | 18 |
| Figura 9 – Parte da planta baixa da casa | 20 |
| Figura 10 – Planta da casa (a) e maquete montada com as paredes (b)..... | 21 |
| Figura 11 – Botões utilizados no projeto..... | 22 |
| Figura 12 – Montagem da maquete com a localização dos botões..... | 22 |
| Figura 13 – Recipiente que representa a piscina | 23 |
| Figura 14 – Botões para simular as portas e janelas | 24 |
| Figura 15 – Sistema do portão | 25 |
| Figura 16 – Sensor de corrente conectado à alimentação de toda a maquete | 26 |
| Figura 17 – Gaveta com a parte eletrônica | 27 |
| Figura 18 – Sketch com a programação do Arduino | 29 |
| Figura 19 – Criação do aplicativo | 30 |
| Figura 20 – Portão aberto..... | 33 |
| Figura 21 – Modulo com os reles para a iluminação | 34 |
| Figura 22 – Fonte de 5V para alimentar o módulo com os reles..... | 34 |
| Figura 23 – Sensores para portas e janelas (a) e sensor de presença (b)..... | 35 |
| Figura 24 – Portão fechado (a) e portão aberto (b) | 35 |
| Figura 25 – Teste de acionamento de lâmpadas pelo aplicativo | 36 |
| Figura 26 – Teste de acionamento de uma lâmpada pela página web..... | 36 |
| Figura 27 – Funcionamento do LEDs nos testes..... | 37 |

LISTA DE QUADROS E TABELAS

| | |
|---|----|
| Quadro 1 – Portas de Entradas e Saídas (E/S) utilizadas no Arduino..... | 28 |
| Quadro 2 – Lista de componentes utilizados e preços..... | 31 |
| Quadro 3 – Custo total do projeto | 32 |

LISTA DE SIGLAS E ABREVIATURAS

| | |
|-----------------|---|
| °C | Graus Celsius |
| CA | Corrente Alternada |
| CC | Corrente Contínua |
| E/S | Entrada/Saída |
| GND | <i>Ground</i> |
| HTML | <i>HyperText Markup Language</i> |
| IDE | <i>Integrated Development Environment</i> |
| IP | <i>Internet Protocol</i> |
| kB | Quilo byte |
| LED | <i>Ligth Emitting Diode</i> |
| mA | Mili amperes |
| MDF | <i>Medium-Density Fiberboard</i> |
| MHz | Mega Hertz |
| USB | <i>Universal Serial Bus</i> |
| V | Volt |
| V _{CA} | Tensão Corrente Alternada |
| V _{CC} | Tensão Corrente Contínua |
| W | Watts |

SUMÁRIO

| | |
|--|-----------|
| 1 INTRODUÇÃO..... | 10 |
| 1.1 OBJETIVOS..... | 10 |
| 2 EMBASAMENTO TEÓRICO | 12 |
| 2.1 ARDUINO | 12 |
| 2.2 APP INVENTOR 2 | 13 |
| 2.3 ETHERNET SHIELD..... | 14 |
| 2.4 AMBIENTE DE DESENVOLVIMENTO INTEGRADO..... | 15 |
| 2.5 SENSOR DIGITAL DE TEMPERATURA DS18B20..... | 16 |
| 2.6 SENSOR DE CORRENTE..... | 17 |
| 2.7 SENSOR DE PRESENÇA PIR | 18 |
| 2.8 PONTE H PARA O MOTOR DO PORTÃO | 18 |
| 2.9 LINGUAGEM DE PROGRAMAÇÃO | 19 |
| 2.10 LINGUAGEM HTML | 19 |
| 3 PROJETO E EXECUÇÃO | 20 |
| 3.1 CONSTRUÇÃO DA MAQUETE E MONTAGEM ELÉTRICA | 20 |
| 3.1.1 Construção da Maquete em MDF | 21 |
| 3.1.2 Sistema de iluminação | 22 |
| 3.1.3 Controle da Temperatura e Monitoramento da Piscina | 23 |
| 3.1.4 Sistemas de Segurança | 23 |
| 3.1.5 Portão Eletrônico..... | 24 |
| 3.1.6 Medição do Consumo de Energia | 25 |
| 3.1.7 Gaveta para a Parte Elétrica | 26 |
| 3.2 PROGRAMAÇÃO DO MICROCONTROLADOR | 27 |
| 3.3 CRIAÇÃO DO APLICATIVO..... | 29 |
| 3.4 CRIAÇÃO DA PÁGINA DA WEB..... | 30 |
| 3.5 CUSTOS..... | 31 |
| 4 TESTES E RESULTADOS..... | 33 |
| 4.1 ILUMINAÇÃO..... | 33 |
| 4.2 SISTEMA DE SEGURANÇA..... | 35 |
| 4.3 CONTROLE DO PORTÃO ELETRÔNICO | 35 |
| 4.4 TESTE DO PROGRAMA FINAL | 36 |
| 5 CONCLUSÃO | 38 |
| REFERÊNCIAS..... | 39 |
| APÊNDICE A – PLANTA BAIXA DA MAQUETE..... | 41 |
| APÊNDICE B – ESQUEMA ELÉTRICO DO PROJETO..... | 42 |

1 INTRODUÇÃO

Com o avanço da tecnologia, o homem começou a criar e desenvolver ferramentas e dispositivos que lhe proporcionassem uma melhoria em seu nível de qualidade de vida, trazendo para suas casas todo o conforto e comodidade que a tecnologia pode oferecer. A automação residencial tem mostrado que a integração de dispositivos eletroeletrônicos e eletromecânicos aumenta consideravelmente os benefícios se comparados com sistemas isolados (BOLZANI, 2010).

A automação residencial não deve ser considerada somente uma forma de luxo ou excentricidade, disponível somente às pessoas com mais recursos financeiros. A automação residencial, inicialmente, é percebida como um símbolo de status e modernidade, contudo, o resultado é um ambiente prático, confortável, agradável, mais bonito, valorizado e seguro, de acordo com a automação e o interesse do usuário. O conforto e a conveniência por ela proporcionados passam a ser decisivos. E, por fim, ela se tornará uma necessidade e um fator de economia.

Com proposta de melhorar a qualidade de vida, aumentar a segurança, o bem-estar, à redução nos afazeres domésticos, bem como a diminuição de custos, surgiu a domótica, termo que é uma fusão da palavra latina *domus* (casa) e da robótica. A domótica, que também pode ser referenciada por expressões como *smart building*, *intelligent building*, edifícios inteligentes, é um novo domínio de aplicação tecnológica, tendo como objetivo básico melhorar a qualidade de vida, aumentando o bem-estar e a segurança de seus habitantes.

A domótica também visa uma utilização racional e planejada dos diversos meios de consumo, além de procurar uma melhor integração através da automatização nas áreas de segurança, de comunicação, controle e gestão de fluidos (VECCHI E OGATA, 1999).

1.1 OBJETIVOS

O objetivo principal deste trabalho consiste em desenvolver uma maquete de uma casa (com redução de escala de 1:20) tendo os sistemas de iluminação, contro-

le de temperatura, sistema de alarme e ambientes sonoros controlados por um microcontrolador e com um sistema supervisório onde será possível realizar todo o controle da casa via internet ou smartphone.

Para que seja atingido o objetivo principal do trabalho foram traçados os seguintes objetivos específicos:

- Construir uma maquete em escala reduzida de 1:20 para demonstração;
- Programar uma página web para acionar e verificar componentes do sistema;
- Criar um aplicativo android para controle e acionamento do portão, lâmpadas, temperatura da piscina e verificação de sensores e alarmes;
- Desenvolver um supervisório para verificar o estado de sensores e alarmes;
- Implementar funcionalidades de comunicação entre o computador o aplicativo e o microcontrolador;

2 EMBASAMENTO TEÓRICO

Serão apresentados a seguir os principais elementos utilizados na realização desse projeto, bem como detalhes técnicos sobre seu funcionamento de forma a tornar melhor o entendimento do projeto.

2.1 ARDUINO

O Arduino é um projeto que surgiu na Itália em 2005 com o objetivo de tornar projetos escolares mais baratos que outros sistemas já existentes. Possui hardware e software mais simples que podem ser usados por qualquer pessoa com pouco conhecimento em programação.

Os programas (softwares) para o Arduino são chamados *sketches* (esboços) e são criados no computador com a utilização de um Ambiente de desenvolvimento (IDE). O IDE permite escrever e editar códigos e converter esses códigos em instruções que o hardware do Arduino irá entender (interpretar). O IDE Também transfere essas instruções para a placa do Arduino, um processo chamado upload (MICHAEL MARGOLIS, ARDUINO COOKBOOK).

A linguagem principal usada no Arduino é a linguagem de programação em C desenvolvida pela primeira vez pelo Instituto de Laboratórios de Bell no início de 1970 para ser usado no sistema operacional UNIX. A linguagem C usa uma sintaxe de linguagem procedural que precisa ser processada por um compilador para mapear o código legível para o computador (BRIAN EVANS, ARDUINO PROGRAMING).

O Arduino Mega2560, utilizado neste projeto, é baseado no processador ATmega2560 que possui 256 KB de memória *flash*, é dotado de 54 entradas/saídas digitais, 16 entradas analógicas, um oscilador de cristal de 16 MHz, conexão USB, fonte de alimentação e botão *reset*. Pode tanto ser alimentado pela conexão USB como através de alimentação externa, de 6 V a 20 V, com uma faixa de tensão recomendada entre 7 e 12 V, ligada aos pinos V_{IN} e GND (tensão de entrada e referência) (MULTILÓGICA SHOP, 2016).

As 54 portas digitais são numeradas de 0 a 53, operam em 5 V e podem ser usados como entrada ou saída de acordo com a programação do software, utilizando as funções `pinMode()`, `digitalWrite()`, e `digitalRead()`. Os pinos podem fornecer ou receber uma corrente de até 40 mA cada, e alguns podem ter funções especiais como por exemplo, controle PWM (pinos 0 a 13) e transmissão de dados seriais. As 16 entradas analógicas são numeradas de A0 a A15 (MULTILÓGICA SHOP, 2016).

Na Figura 1 é possível visualizar um Arduino Mega 2560 R3, onde o barramento na parte superior e na direita possuem as portas digitais, e na parte inferior (com a letra A antes da sequência numérica) estão as entradas analógicas.

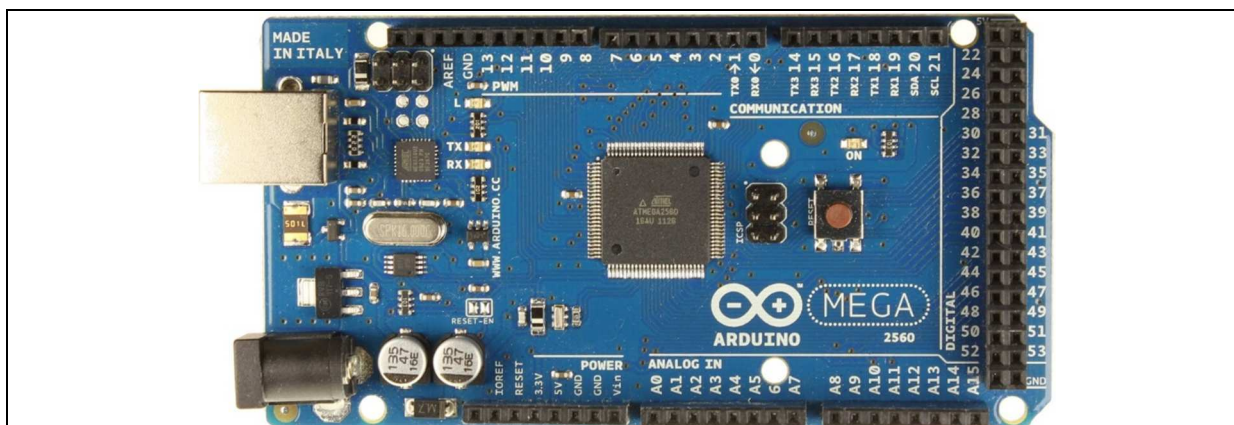


Figura 1 – Arduino Mega 2560 R3
Fonte: Scion Eletronics (2016).

2.2 APP INVENTOR 2

O App Inventor é uma plataforma para criação de aplicativos Android baseada na web, que provê uma interface visual com o objetivo de permitir que qualquer pessoa, mesmo sem grande conhecimento de codificação, possa construir aplicativos Android. O App Inventor é um exemplo do conceito PaaS (Platform as a Service/Plataforma como serviço), pois a plataforma é oferecida como um serviço e está disponível para uso, bastando apenas que o usuário tenha um computador conectado à internet e um browser (navegador). A plataforma é dividida em duas partes: App Inventor Designer, para a construção da interface gráfica da aplicação, e o App Inventor Block Editor, para associar ações aos componentes da interface (GALENO E GONÇALVES, 2013).

Na Figura 2(a) pode ser observada parte da interface de design do software App Inventor, onde no centro observa-se uma imagem que representa a tela de um smartphone, e na Figura 2(b) a interface para a edição de blocos.

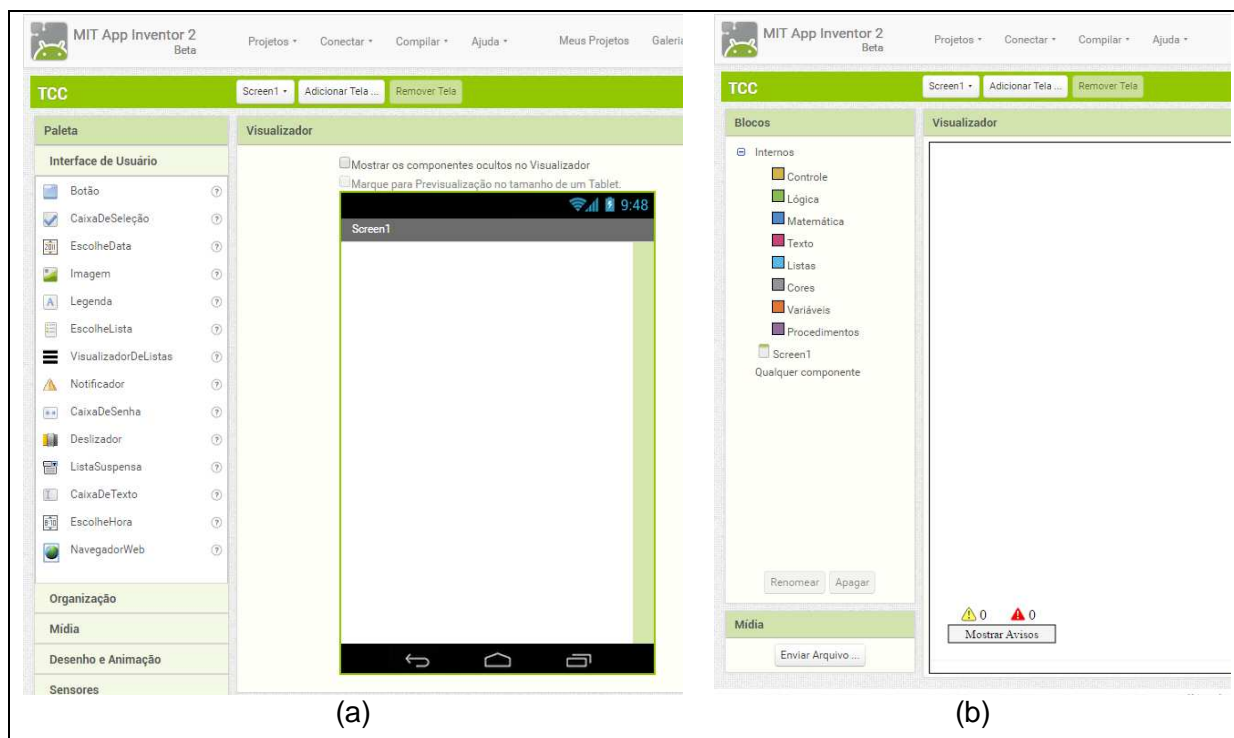


Figura 2 – Interface de Designer (a) e edição de Blocos (b)
Fonte: App Inventor (2016)

2.3 ETHERNET SHIELD

O *Ethernet Shield* é uma placa que permite a comunicação do Arduino Mega 2560 com a internet e também com uma rede local e fornece uma rede (IP) capaz de se conectar à internet ou rede local.

Na parte superior do *shield* estão alguns LEDs que indicam a energização do *Ethernet Shield*, transmissão e recepção dos sinais e conexão com a rede (ARDUINO E CIA, 2016).

Para utilizá-lo basta que o *Shield* seja encaixado sobre o Arduino utilizando os pinos 10, 11, 12 e 13 e que seja conectado um cabo do tipo RJ-45 na entrada do *shield*. A Figura 3 mostra a imagem do Ethernet Shield.

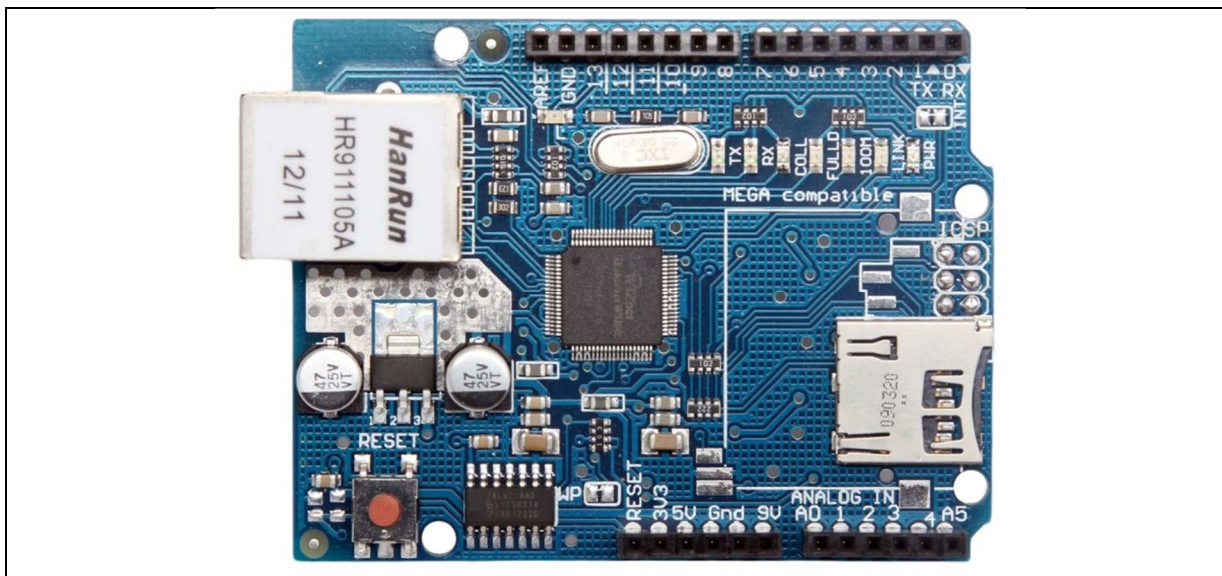


Figura 3 – Ethernet Shield
Fonte: Nexus Cyber (2016).

Após a conexão do Ethernet Shield sobre o Arduino é necessário incluir a biblioteca do *shield* no IDE do Arduino e configurar o IP que a placa terá na rede.

2.4 AMBIENTE DE DESENVOLVIMENTO INTEGRADO

O Ambiente de Desenvolvimento Integrado Arduino, ou Arduino Software (IDE), contém um editor de texto para escrever o código, uma área de mensagem, um console de texto, uma barra de ferramentas com botões para funções comuns e uma série de menus. Ele se conecta ao hardware do Arduino para carregar programas e realizar a comunicação com eles (ARDUINO.CC, 2016).

Os Programas escritos usando o Arduino Software (IDE) são chamados de *sketches*. Estas *sketches* são escritas no editor de texto e são salvos com a extensão de arquivo “.ino” (ARDUINO.CC, 2016).

A área de mensagem fornece um *feedback* ao salvar projetos e após a exportação pode exibir a ocorrência de eventuais erros. O canto inferior direito da janela exibe a placa configurada e porta serial.

Os botões da barra de ferramentas permitem que você possa verificar e carregar programa, criar, abrir e salvar desenhos e abra o monitor serial. Como mostra a Figura 4 (ARDUINO.CC, 2016).

Na Figura 4 pode ser observado um exemplo de *sketch* escrito no Ambiente de Desenvolvimento Integrado Arduino.

The image shows a screenshot of the Arduino IDE interface. The title bar reads "Blink | Arduino 1.0". The main editor window contains the following code:

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // set the LED off
  delay(1000);           // wait for a second
}
```

The status bar at the bottom indicates "1" and "Arduino Uno on /dev/tty.usbmodemfd131".

Figura 4 – IDE Arduino
Fonte: Arduino (2016).

2.5 SENSOR DIGITAL DE TEMPERATURA DS18B20

O sensor digital de temperatura DS18B20 trabalha em uma faixa de temperatura de $-55\text{ }^{\circ}\text{C}$ até $150\text{ }^{\circ}\text{C}$, tem precisão de $\pm 0,5\text{ }^{\circ}\text{C}$, possui alimentação V_{CC} e possui apenas um cabo para comunicação com o microcontrolador. Também pode ser ligado eliminando-se a fonte de alimentação externa, com a utilização do terminal de linha de dados como fonte de energia (ADD THERM, 2016).

Cada sensor possui um código 64-bit único, o que permite que diversos sensores possam ser utilizados em série na mesma entrada/saída do microcontrolador que será capaz de identificar cada sensor (ADD THERM, 2016).

A Figura 5 mostra o sensor do modelo ds18b20, usado neste projeto bem como a designação de seus terminais.

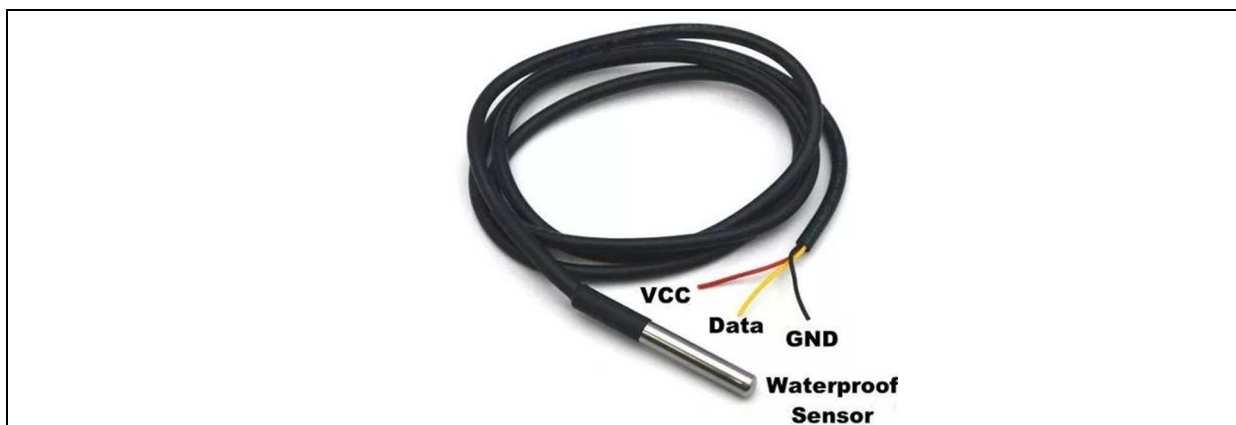


Figura 5 – Sensor digital ds18b20
Fonte: Filipeflop (2016).

2.6 SENSOR DE CORRENTE

De acordo com Carvalho (2016), quando ocorre a passagem de corrente por meio de portadores de carga sem a aproximação de um campo magnético, a corrente flui normalmente, de maneira uniforme entre os terminais do semicondutor e nenhuma corrente é detectada no sentido paralelo à passagem da corrente. Quando ocorre a aproximação de um campo magnético uma força perpendicular ao deslocamento das cargas faz com que essas se desviem da trajetória normal, se acumulando em uma das faces laterais, gerando a detecção de uma pequena tensão entre as faces paralelas do semicondutor (CARVALHO, 2016).

Na Figura 6 pode ser observado o sensor de corrente SCT013-030A que foi utilizado neste projeto.



Figura 6 – Sensor de corrente SCT013-030A.
Fonte: Filipeflop (2016)

2.7 SENSOR DE PRESENÇA PIR

O sensor utilizado para este projeto foi o sensor de presença PIR (Passive Infrared). O sensor PIR tem um circuito eletrônico capaz de amplificar os sinais e assim pode modular um sinal de saída em nível digital. São circuitos formados com o componente BISS0001 que em sua saída geram um sinal "on" ou "off" (ligado ou desligado) para acionar ou desligar um circuito externo (SILVA, 2016)



Figura 7 – Sensor PIR
Fonte: Autor (2016).

2.8 PONTE H PARA O MOTOR DO PORTÃO

Para controlar o sentido do giro de motores de corrente contínua pode ser utilizada uma ponte H. Na Figura 8(a) pode-se observar que as duas chaves fechadas fazem com que o sentido da corrente seja da esquerda para a direita, portanto, para inverter o sentido da rotação devem-se ligar as outras chaves. No projeto foi utilizado um módulo de ponte H I9110 observado na Figura 8(b).

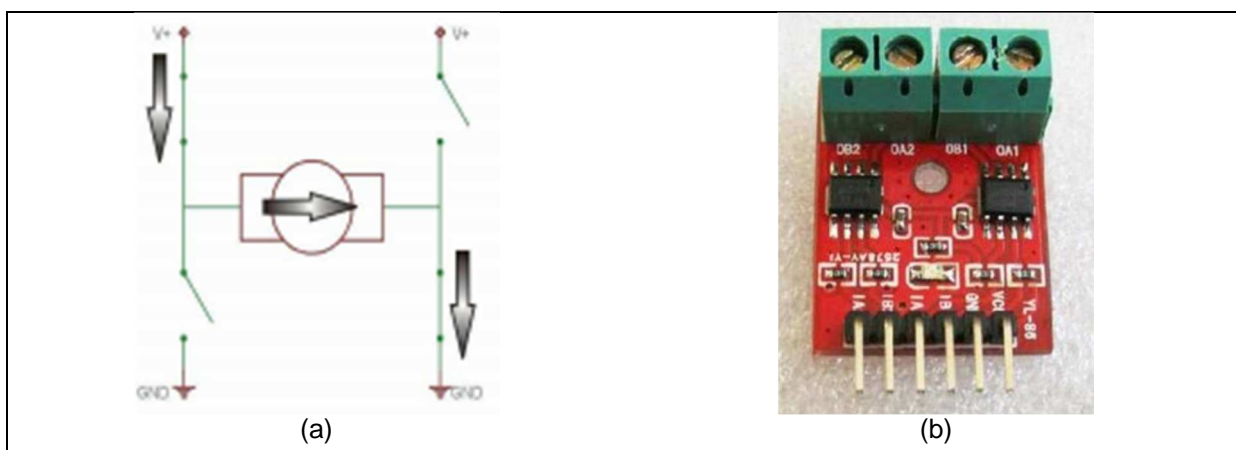


Figura 8 – Esquema de funcionamento de uma ponte H (a) e Módulo I9110 (b)
Fonte: PATSKO (2006) e Ardu (2016).

2.9 LINGUAGEM DE PROGRAMAÇÃO

Existem dois tipos de linguagem de programação: as de baixo nível e as de alto nível. As linguagens de baixo nível são interpretadas diretamente pelo computador, tornando-se uma maneira rápida de programar, porém é muito difícil de se trabalhar com ela (CRIAR WEB, 2016).

As linguagens de alto nível são aquelas onde são criadas condições e ações descritas através de um arquivo de texto (chamado de código fonte), e as palavras dentro deste código recebem o nome de instruções. Após completar o código fonte, é necessário utilizar um compilador para transformá-lo em códigos binários, já que este tipo de linguagem não é compreendido diretamente pelos computadores. O compilador transformara o texto do código fonte em um segundo arquivo que chamamos de programa objeto, e este arquivo é interpretado diretamente pelo computador (CRIAR WEB, 2016).

A linguagem de programação utilizada no Arduino é a C++ (com pequenas alterações) e com a utilização adequada de instruções e algoritmos é possível fornecer as instruções que o Arduino deverá seguir. No Arduino estes algoritmos são chamados de *sketches*.

2.10 LINGUAGEM HTML

O HTML é uma linguagem de programação, bastante simples, utilizada na criação de estruturas de página da web que utiliza *tags* (marcadores) para representar elementos de uma página como imagens e links. Cada *tag* possui um nome e é representada entre os sinais < e >. O navegador interpreta o código e carrega a página com os botões e demais elementos (DEV MEDIA, 2016).

A programação em HTML pode ser realizada tanto nos editores de texto dos sistemas operacionais quanto em editores mais complexos que possuem mais recursos, tornando a programação mais fácil (DEV MEDIA, 2016).

3 PROJETO E EXECUÇÃO

Na sequência serão apresentadas as etapas de construção da maquete e a instalação de todos os sensores e atuadores, necessários para a automação residencial, bem como informações sobre a programação do microcontrolador.

3.1 CONSTRUÇÃO DA MAQUETE E MONTAGEM ELÉTRICA

Foi projetada a planta de uma casa (em escala de 1:20) de aproximadamente 50 m² com seis cômodos, possuindo sete pontos de iluminação internos além de dois externos, piscina e portão eletrônico.

Parte da planta baixa pode ser visualizada na Figura 9, sendo que o projeto completo (com as áreas externas) pode ser encontrado no Apêndice A.

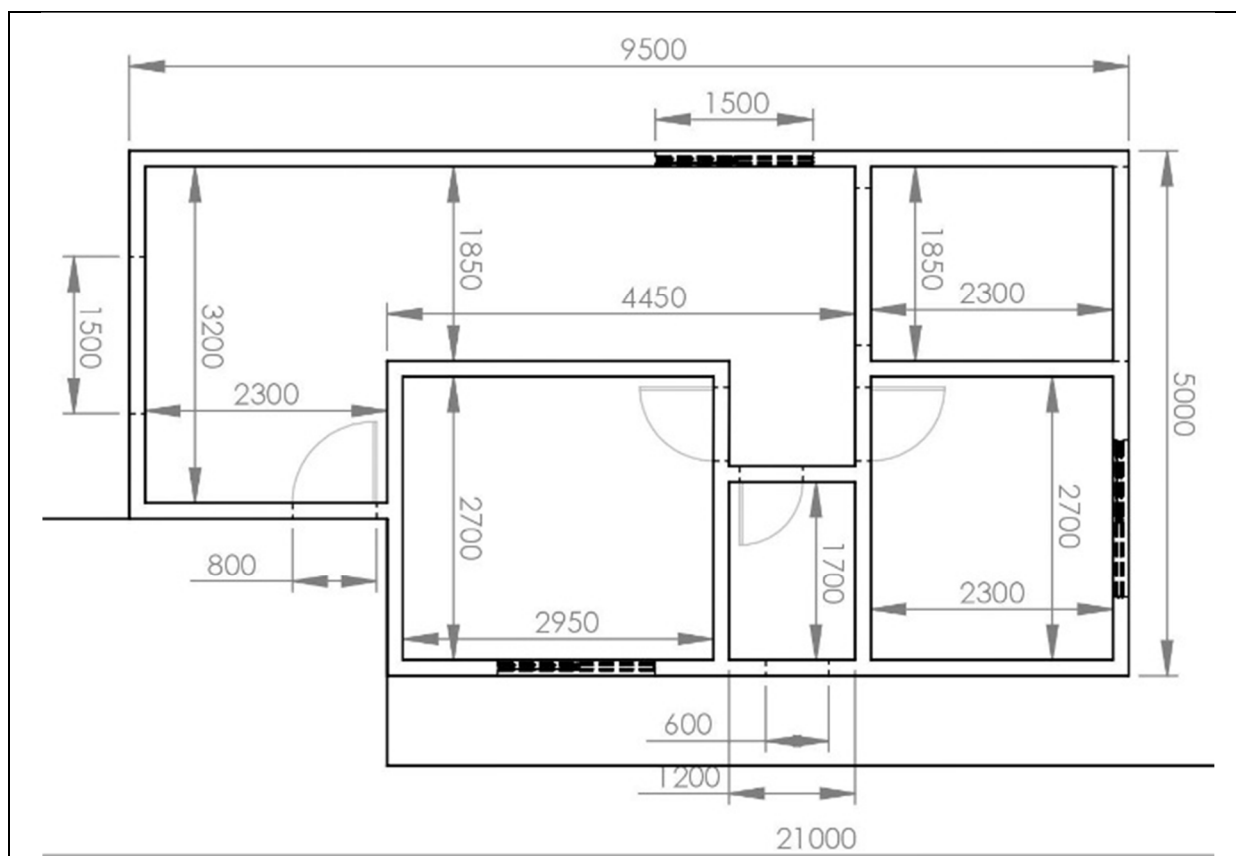


Figura 9 – Parte da planta baixa da casa
Fonte: Autores.

3.1.1 Construção da Maquete em MDF

Para a realização do projeto foi construída uma maquete usando madeira MDF e vidro. Foram montadas as chapas de madeira MDF de maneira a formar uma caixa na parte inferior, então foram feitos furos na chapa superior para passagem dos botões.

A planta baixa da residência, com cores e texturas para facilitar a identificação dos ambientes, foi impressa e colada sobre a placa de madeira MDF o que permite uma boa visualização de cada cômodo da casa, parte externa e piscina conforme pode ser observado na Figura 10(a).

Na Figura 10(b) pode ser observada a maquete com o muro externo, feito de vidro, e já com as paredes da residência devidamente posicionadas.

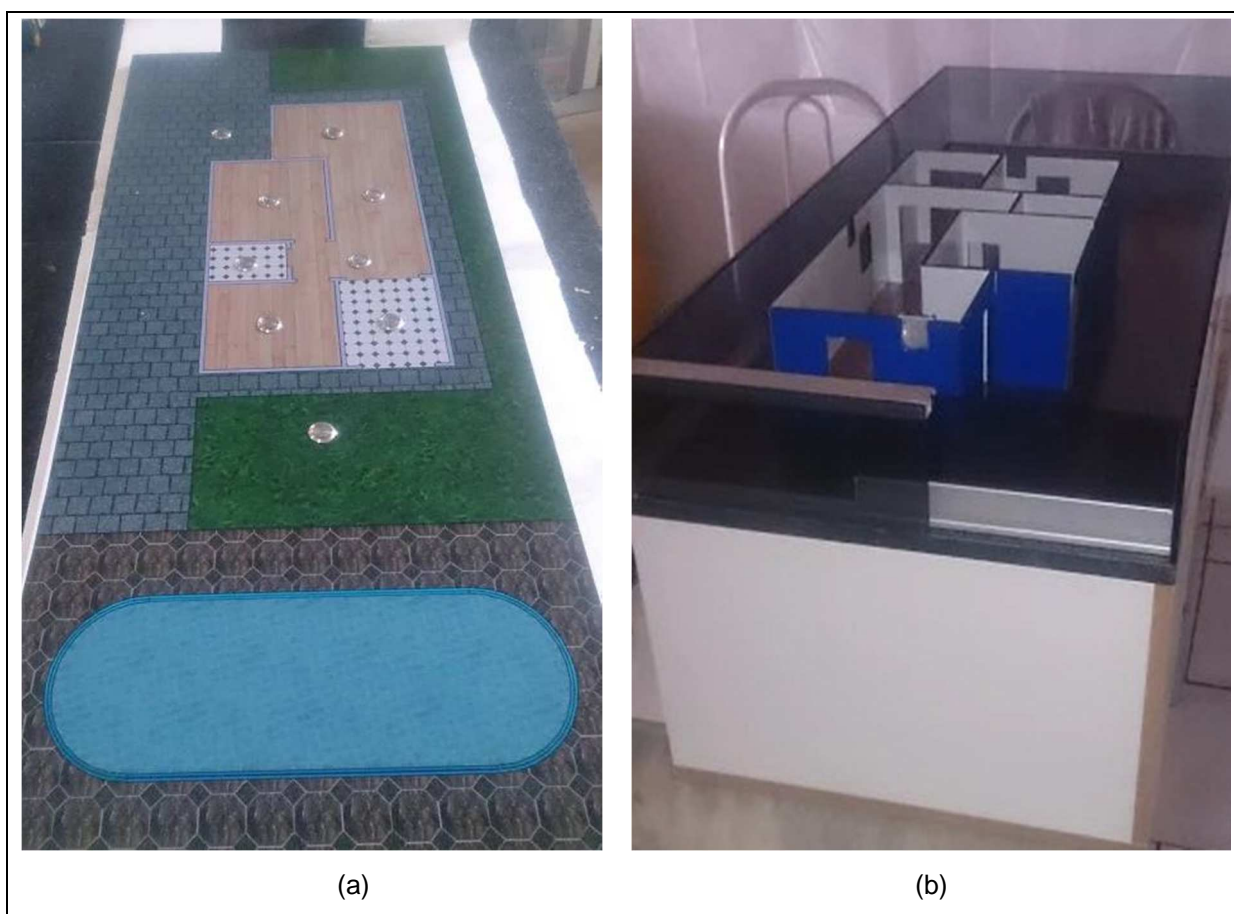


Figura 10 – Planta da casa (a) e maquete montada com as paredes (b)
Fonte: Autores.

3.1.2 Sistema de iluminação

Botões de pulso com LED que representam o sistema de iluminação são botões metálicos iluminados de alta resistência, que podem ser vistos na Figura 11. A tensão do LED é de $24 V_{CC}$ ou V_{CA} , e a capacidade dos contatos é de 5A para $48 V_{CC}$ ou para $250 V_{CA}$.



Figura 11 – Botões utilizados no projeto
Fonte: Autores.

Os botões representam interruptores e luminárias no interior de cada cômodo e na parte externa da residência, como pode ser observado na Figura 12.



Figura 12 – Montagem da maquete com a localização dos botões
Fonte: Autores.

Os LEDs podem ser acionados tanto apertando os botões, quanto por meio do aplicativo Android ou pelo computador. Quando o acionamento acontece fisicamente ocorre uma mudança no status no aplicativo e na página da *web*, permitindo ao usuário saber quais pontos de iluminação estão acionados. O acionamento dos LEDs ocorre por meio de relés dos módulos que recebem um sinal das portas de entrada/saída do Arduino e realiza a alimentação dos LEDs em $24 V_{cc}$.

3.1.3 Controle da Temperatura e Monitoramento da Piscina

Para o controle da temperatura foi utilizado um modulo de sensor DS18B20 a prova d'água e para o aquecimento uma resistência de 12 Volts CC e 40 W de potência, encontrada em impressoras 3D.

Na Figura 13 pode ser observado um recipiente com água (que representa a piscina), o sensor de temperatura e a resistência elétrica.



Figura 13 – Recipiente que representa a piscina
Fonte: Autores.

O sensor junto à resistência na piscina permite o controle e monitoramento da temperatura da água de acordo com a temperatura definida pelo usuário do aplicativo ou computador. A resistência consome 40 W, é ligada a fonte chaveada de 12 V_{CC} e também é controlada mediante acionamento de um relé dos módulos de relés.

A leitura da temperatura é feita através de um sensor digital à prova d'água capaz de realizar medidas em uma temperatura de até 125 °C com o terminal de dados ligado a uma entrada do microcontrolador.

3.1.4 Sistemas de Segurança

Os botões com trava funcionam com sensores magnéticos, por exemplo, para monitoramento das portas e janelas da residência. Cada botão representa uma porta ou janela e está ligado as portas de entrada/saída do Arduino. Quando acionados, ocorre uma mudança no status do aplicativo indicando qual porta ou janela foi

aberta, ao mesmo tempo em que uma sirene dispara. Cada botão direciona uma tensão de 5 V, quando pressionado a uma porta do microcontrolador.

Este sistema, para simular o monitoramento das portas e janelas, foi construído em um *protoboard* e pode ser visualizado na Figura 14.

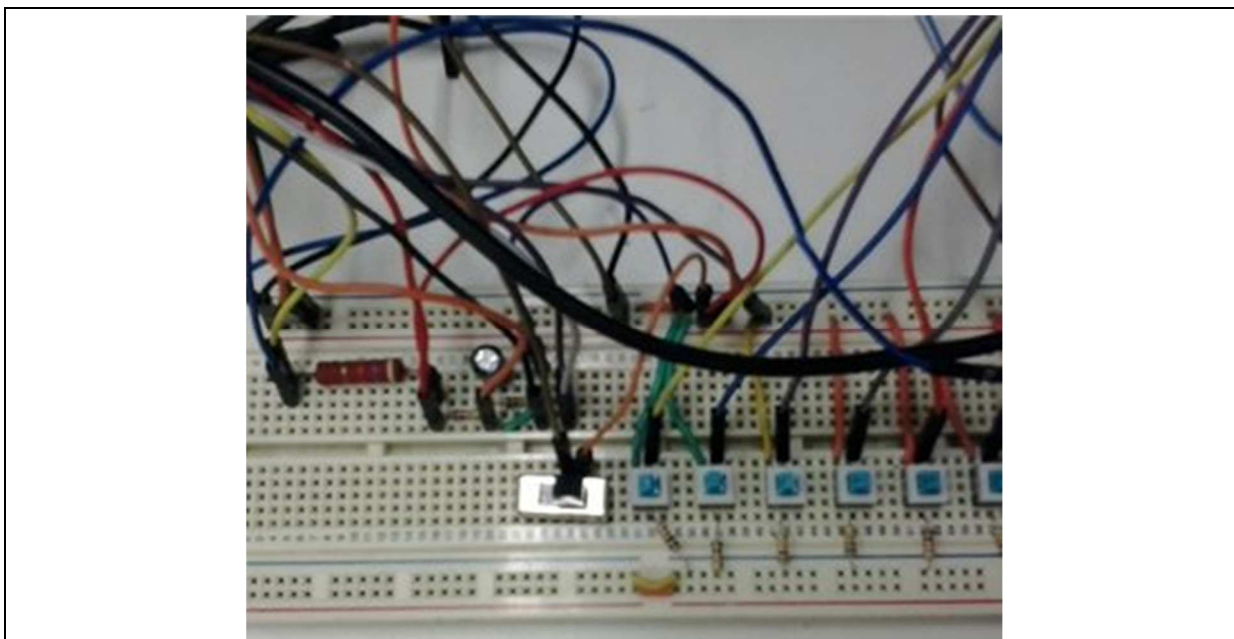


Figura 14 – Botões para simular as portas e janelas
Fonte: Autores.

Também foram utilizados quatro sensores de presença, um em cada um dos dois quartos, um para a cozinha e um para a sala, que quando acionados habilitam o disparo do alarme.

Esse tipo de sensor de presença é capaz de realizar a detecção de corpos em até 7 metros de distância (ajustável) e com um ângulo de até 100°, possui ajuste na sua sensibilidade e para o seu tempo de retardo e necessita de uma tensão de alimentação entre 5 V e 20 V.

3.1.5 Portão Eletrônico

Para a construção do portão na maquete foi utilizada uma chapa de vidro que desliza lateralmente sobre trilhos movida por um motor juntamente com um motorreductor e uma cremalheira.

Para o funcionamento do portão foi utilizado um motor de corrente contínua que funciona com alimentação de 3 V a 6 V, alimentado por uma fonte chaveada de 5 V_{CC} e uma ponte H para controle do sentido de rotação. A ponte utilizada pode controlar até dois motores (dois canais) de até 12 V e possui capacidade de corrente de até 800 mA em cada canal.

Para limitar o movimento do portão foi utilizado um fim de curso micro *switch* para informar ao microcontrolador quando o portão fecha e outro quando o portão está totalmente aberto e deve-se desligar o motor. O acionamento do motor pode ser realizado tanto por aplicativo quanto pelo computador.

Na Figura 15 pode ser observada a montagem do portão, e também podem ser visualizados à esquerda e à direita os sensores fim de curso, conectados com condutores na cor vermelha.

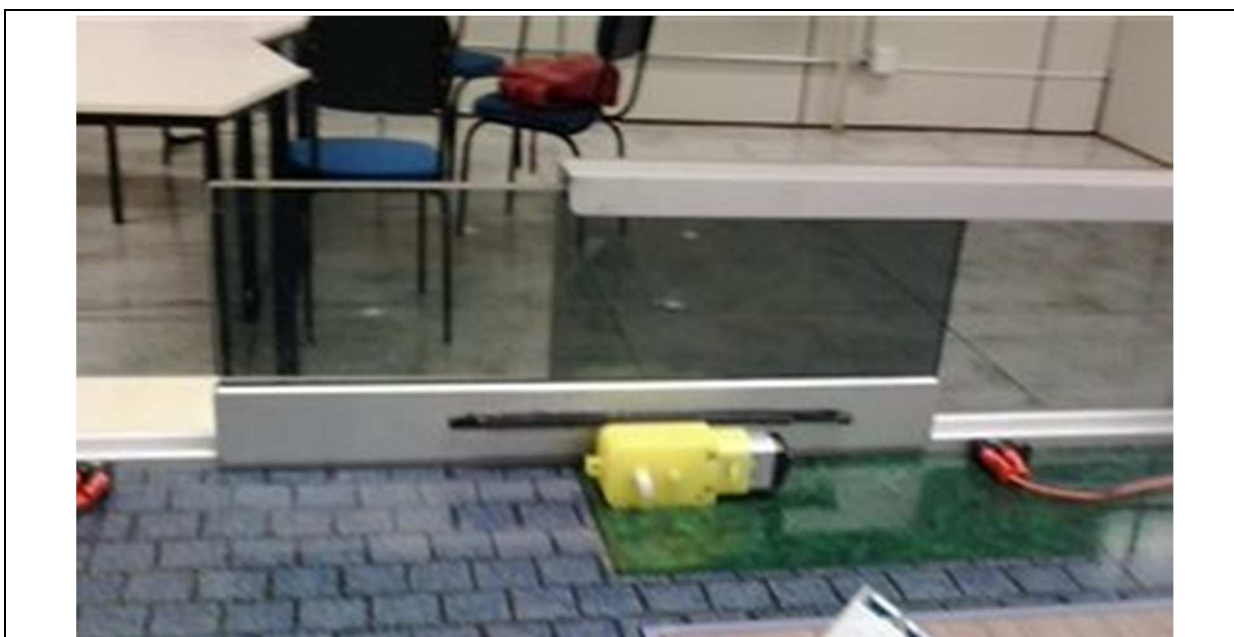


Figura 15 – Sistema do portão
Fonte: Autores.

3.1.6 Medição do Consumo de Energia

Para a medição do consumo de energia elétrica de todo o sistema foi utilizado um sensor de corrente elétrica que funciona por Efeito Hall e pode realizar medidas tanto em CC quanto em CA de até 30 A.

Como trata-se de um sensor não invasivo, não foi preciso a interrupção do circuito para instalação do mesmo, tornando seu uso muito prático, bastando apenas colocar sua garra ao redor do condutor de alimentação do sistema.

O modelo utilizado para o projeto foi o sensor SCT-013-030A, apresentado na Figura 16, sua construção é basicamente formada por um transformador de corrente e um resistor de referência. O modelo utilizado tem como saída os valores de tensão com variação de 0 a 1 V,

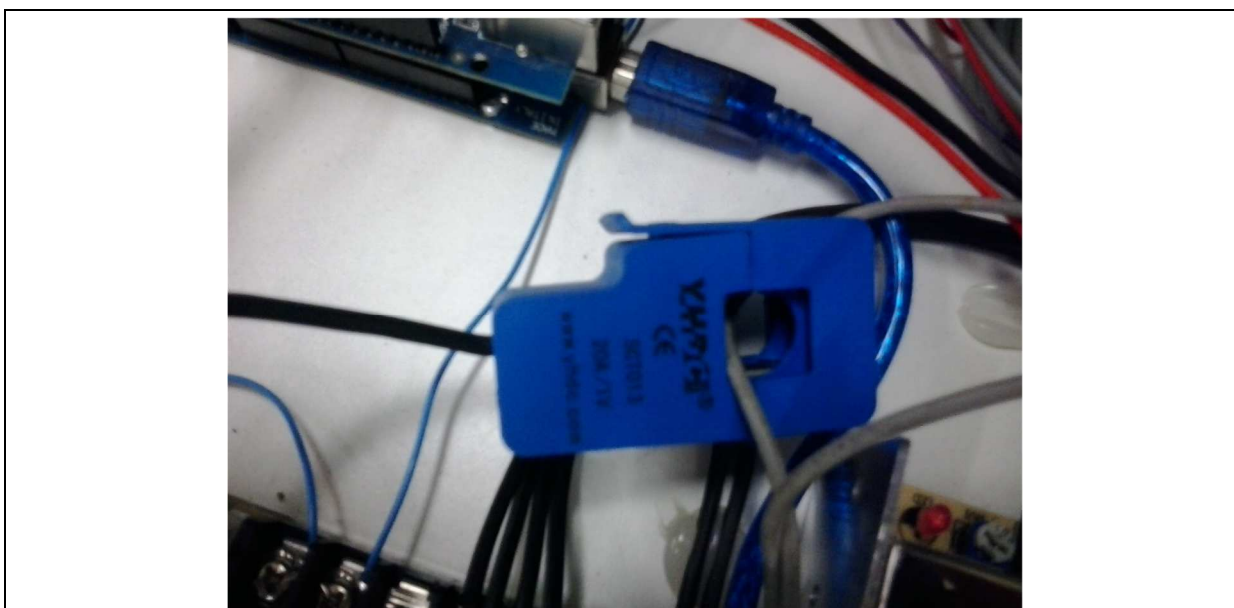


Figura 16 – Sensor de corrente conectado à alimentação de toda a maquete
Fonte: Autores.

O Arduino realiza o cálculo do consumo de energia elétrica de acordo com as informações recebidas pelo módulo do sensor de corrente e transmite as informações via Ethernet Shield que estarão disponíveis no aplicativo e na *internet*.

3.1.7 Gaveta para a Parte Elétrica

Toda parte elétrica fica sob a chapa de madeira MDF com a planta impressa e colada de forma que apenas os botões de pulso com LED ficam aparentes, em um sistema de gaveta, para que possa ser movimentado para se ter melhor acesso em casos de eventual necessidade de manutenção ou alterações no sistema como mostra a Figura 17.

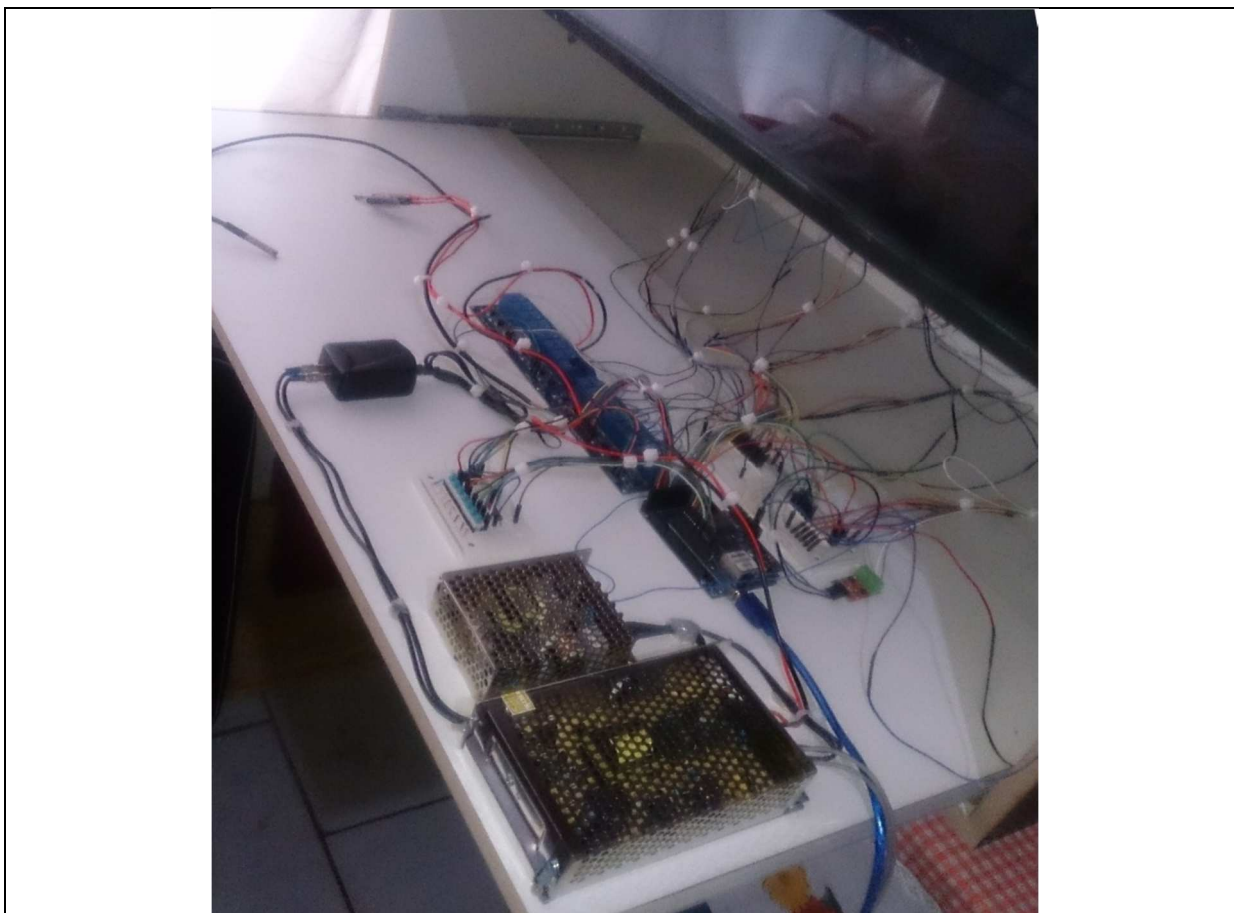


Figura 17 – Gaveta com a parte eletrônica
Fonte: Autores.

A alimentação é feita na rede de $127 V_{AC}$, onde foram ligadas as fontes chaveadas que transformam a tensão para $24 V_{CC}$, $12V_{CC}$ ou $5 V_{CC}$ conforme necessário para a alimentação de cada componente.

3.2 PROGRAMAÇÃO DO MICROCONTROLADOR

Após a montagem da maquete e a ligação física de todos os elementos elétricos e eletrônicos foi necessário realizar a programação do controlador. Inicialmente foram definidas quais portas se comportariam como entradas e quais seriam saídas de acordo com a lógica necessária para o funcionamento de cada elemento a ser controlado na residência.

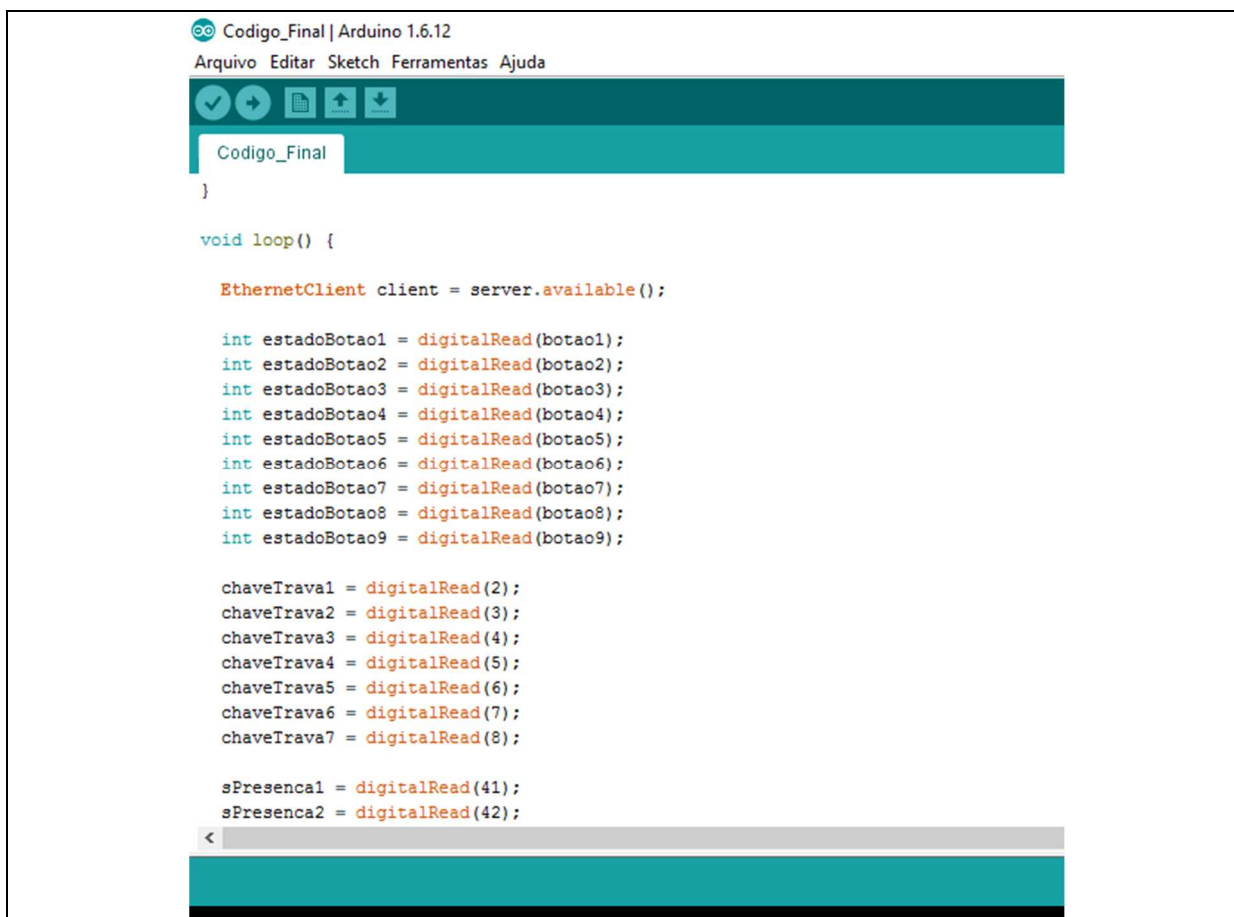
O Quadro 1 apresenta a relação das entradas/saídas do microcontrolador com os elementos a serem controlados.

| Porta | Item | Local/Descrição |
|--------------|--------------------------|----------------------------|
| 2 | Botão de pulso com trava | Porta da frente |
| 3 | Botão de pulso com trava | Janela da sala |
| 4 | Botão de pulso com trava | Janela da cozinha |
| 5 | Botão de pulso com trava | Janela do quarto da frente |
| 6 | Botão de pulso com trava | Janela do banheiro |
| 7 | Botão de pulso com trava | Janela quarto dos fundos |
| 8 | Botão de pulso com trava | Porta dos fundos |
| 9 | Sensor de temperatura | Piscina |
| 10 | Sensor de corrente | |
| 22 | Relé 1 / IN8 | L1-externa frente |
| 23 | Relé 1 / IN7 | L2-sala |
| 24 | Relé 1 / IN6 | L3-sala/cozinha |
| 25 | Relé 1 / IN5 | L4-quarto frente |
| 26 | Relé 1 / IN4 | L5-cozinha |
| 27 | Relé 1 / IN3 | L6-banheiro |
| 28 | Relé 1 / IN2 | L7-lavanderia |
| 29 | Relé 1 / IN1 | L8-quarto dos fundos |
| 30 | Relé 2 / IN7 | L9-externa fundos |
| 31 | Relé 2 / IN6 | Resistência piscina |
| 32 | Botão de pulso com LED | L1-externa frente |
| 33 | Botão de pulso com LED | L2-sala |
| 34 | Botão de pulso com LED | L3-sala/cozinha |
| 35 | Botão de pulso com LED | L4-quarto frente |
| 36 | Botão de pulso com LED | L5-cozinha |
| 37 | Botão de pulso com LED | L6-banheiro |
| 38 | Botão de pulso com LED | L7-lavanderia |
| 39 | Botão de pulso com LED | L8-quarto dos fundos |
| 40 | Botão de pulso com LED | L9-externa fundos |
| 41 | Sensor de presença pir | Quarto dos fundos |
| 42 | Sensor de presença pir | Quarto da frente |
| 43 | Sensor de presença pir | Cozinha |
| 44 | Sensor de presença pir | Sala |
| 45 | Ponte H | Motor sentido horário |
| 46 | Ponte H | Motor sentido anti-horário |
| 47 | Fim de curso | Portão fechado |
| 48 | Fim de curso | Portão aberto |

Quadro 1 – Portas de Entradas e Saídas (E/S) utilizadas no Arduino

Fonte: Autores.

Após a compilação com correção de eventuais erros cometidos na sua criação o *sketch* pode ser finalmente transferido para o microcontrolador. Na Figura 18 pode ser observado o *sketch* usado no projeto.



```
Codigo_Final | Arduino 1.6.12
Arquivo  Editar  Sketch  Ferramentas  Ajuda

Codigo_Final
}

void loop() {

  EthernetClient client = server.available();

  int estadoBotao1 = digitalRead(botao1);
  int estadoBotao2 = digitalRead(botao2);
  int estadoBotao3 = digitalRead(botao3);
  int estadoBotao4 = digitalRead(botao4);
  int estadoBotao5 = digitalRead(botao5);
  int estadoBotao6 = digitalRead(botao6);
  int estadoBotao7 = digitalRead(botao7);
  int estadoBotao8 = digitalRead(botao8);
  int estadoBotao9 = digitalRead(botao9);

  chaveTrava1 = digitalRead(2);
  chaveTrava2 = digitalRead(3);
  chaveTrava3 = digitalRead(4);
  chaveTrava4 = digitalRead(5);
  chaveTrava5 = digitalRead(6);
  chaveTrava6 = digitalRead(7);
  chaveTrava7 = digitalRead(8);

  sPresenca1 = digitalRead(41);
  sPresenca2 = digitalRead(42);
}
```

Figura 18 – Sketch com a programação do Arduino
Fonte: Autores.

3.3 CRIAÇÃO DO APLICATIVO

Na sequência foi necessária a criação do aplicativo Android a ser instalado em um smartphone, para que através da internet pudessem ser enviadas as informações para o Ethernet Shield, que as interpretaria, transformando em sinais capazes de serem lidos pelo Arduino. Foi criada uma interface simples e intuitiva utilizando-se o App Inventor Designer e a seguir a programação necessária para seu funcionamento de acordo com a lógica de programação em blocos no App Inventor Block Editor.

Na Figura 19 pode ser observada parte da criação do aplicativo com o App Inventor Block Editor.

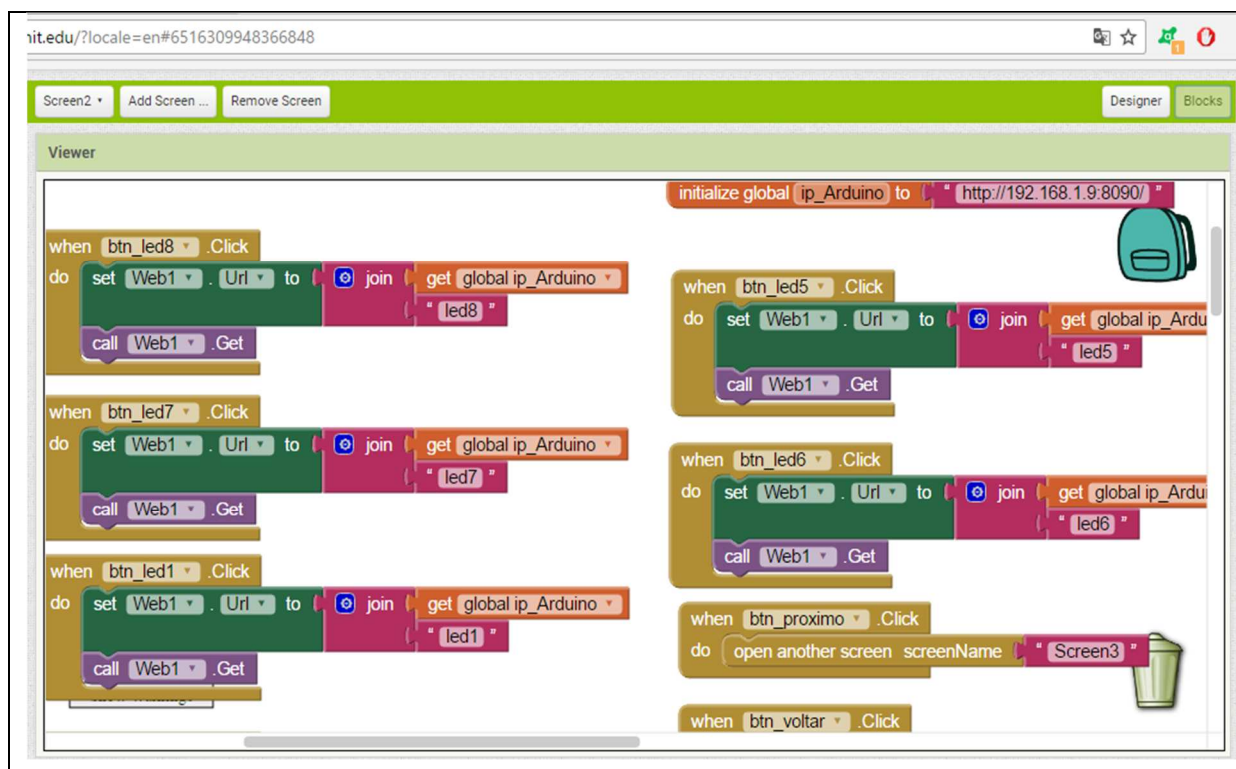


Figura 19 – Criação do aplicativo
Fonte: Autores.

3.4 CRIAÇÃO DA PÁGINA DA WEB

Também se fez necessária à criação de uma página da web, em linguagem HTML, que funciona de forma análoga ao programa projetado para Android, para controle dos dispositivos e acesso as informações do sistema pelo computador de modo que o controle possa ser feito de qualquer local e qualquer computador com acesso à internet, sem a necessidade de instalação de um programa adicional como no caso do aplicativo.

Como os autores não possuem conhecimento em programação HTML a programação necessária para criação da página foi realizada por um terceiro, cabendo aos autores apenas sua utilização para controle dos processos do Arduino, desta forma, não são apresentados maiores detalhes sobre a programação.

3.5 CUSTOS

Para a realização do projeto foi necessária à compra de vários itens, no entanto, como a ideia inicial era limitar os gastos em um sistema modesto, porém, eficiente, alguns itens que precisariam ser instalados em um projeto real de uma residência foram substituídos por outros mais baratos e simples, para a aplicação na maquete. Por exemplo, no lugar de serem utilizados sensores nas entradas das portas e janelas, foram utilizados botões de retenção, o que garante também uma maior facilidade na hora de se testar o funcionamento do programa.

Também foram utilizados materiais de fácil acesso e baixo custo, como o vidro e madeira MDF para construção da maquete. O Quadro 2 traz a relação dos itens envolvidos no projeto, bem como os valores gastos.

| Componente | Quantidade | Preço |
|---|------------|-------------------|
| Arduino Mega 2560 R3 | 1 | R\$ 90,00 |
| Módulo relé 8 canais | 2 | R\$ 76,80 |
| Sensor de temperatura DS18B20 | 1 | R\$ 16,00 |
| Ethernet Shield W5100 | 1 | R\$ 48,00 |
| Fim de curso micro switch | 2 | R\$ 6,40 |
| Motor de corrente contínua 3-6V | 1 | R\$ 9,60 |
| Botão de pulso com trava | 7 | R\$ 4,48 |
| Módulo ponte H | 1 | R\$ 16,00 |
| Botão de pulso inox 24V 1NA 1NF com LED | 9 | R\$ 119,23 |
| Módulo sensor de corrente 30A | 1 | R\$ 44,80 |
| Fonte chaveada 5V 2A com plug p4 | 1 | R\$ 34,90 |
| Fonte chaveada 24V 11A 25W | 1 | R\$ 39,00 |
| Fonte chaveada 12V 5A 60W | 1 | R\$ 15,77 |
| Resistência 12V 40W | 1 | R\$ 38,00 |
| Total | | R\$ 622,98 |

Quadro 2 – Lista de componentes utilizados e preços

Fonte: Autores.

O valor gasto na compra dos componentes eletrônicos foi de aproximadamente R\$ 620,00. Todos os elementos podem ser facilmente encontrados em lojas na internet ou em eletrônicas.

O Quadro 3 mostra os valores gastos com outros materiais do projeto, que juntamente com os componentes eletrônicos, irão compor o custo total.

| Material | Preço |
|-------------------------|-------------------|
| Madeira MDF | R\$ 100,00 |
| Vidro | R\$ 50,00 |
| Impressão da planta | R\$ 30,00 |
| Fios | R\$ 30,00 |
| Componentes eletrônicos | R\$ 622,98 |
| Total | R\$ 832,98 |

Quadro 3 – Custo total do projeto

Fonte: Autores.

Portanto, o valor total gasto para a construção da maquete e a realização da automação residencial, entre compra de componentes e materiais foi de aproximadamente R\$ 833,00.

4 TESTES E RESULTADOS

Durante o desenvolvimento do trabalho, alguns problemas precisaram ser resolvidos, e neste capítulo serão apresentados os testes realizados e os problemas identificados.

4.1 ILUMINAÇÃO

Para o sistema de iluminação foram utilizados botões de pulso com LED. Os botões representam interruptores e luminárias no interior de cada cômodo e na parte externa da residência. O problema encontrado foi que o Arduino não reconhecia quando o botão estava ou não pressionado, ele verificava somente se a porta está ligada (HIGH) ou se a porta está desligada (LOW).

Tendo em vista que podem ocorrer variações de tensão nas portas do Arduino constatou-se a necessidade de se conectar um resistor *Pull-up* em serie com o botão para que essas variações fossem isoladas e não indicassem falsos pressionamentos.

Na Figura 20 pode ser observado o esquema de ligação que foi empregado, sendo implementado com a utilização de um protoboard.

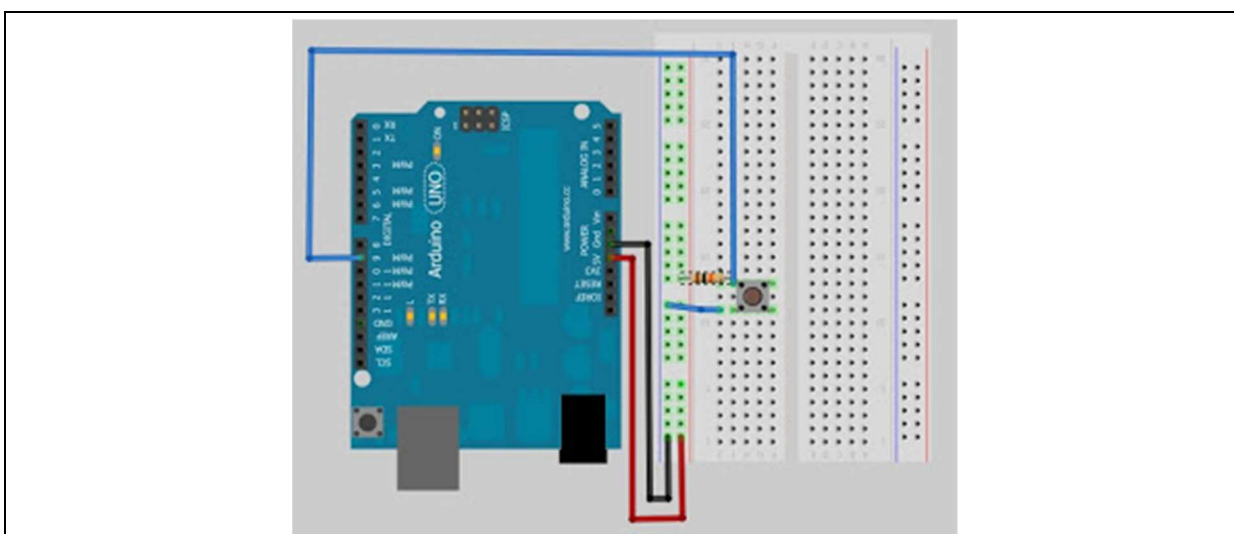


Figura 20 – Portão aberto
Fonte: Autores.

Após a ligação de todas as lâmpadas no módulo de relés (que pode ser observado na Figura 21) foram iniciados os testes e houve uma queda de tensão no sistema. Essa queda de tensão ocorreu porque o Arduino estava sendo alimentado por uma porta USB que disponibilizava 5V com uma corrente máxima de 500 mA.

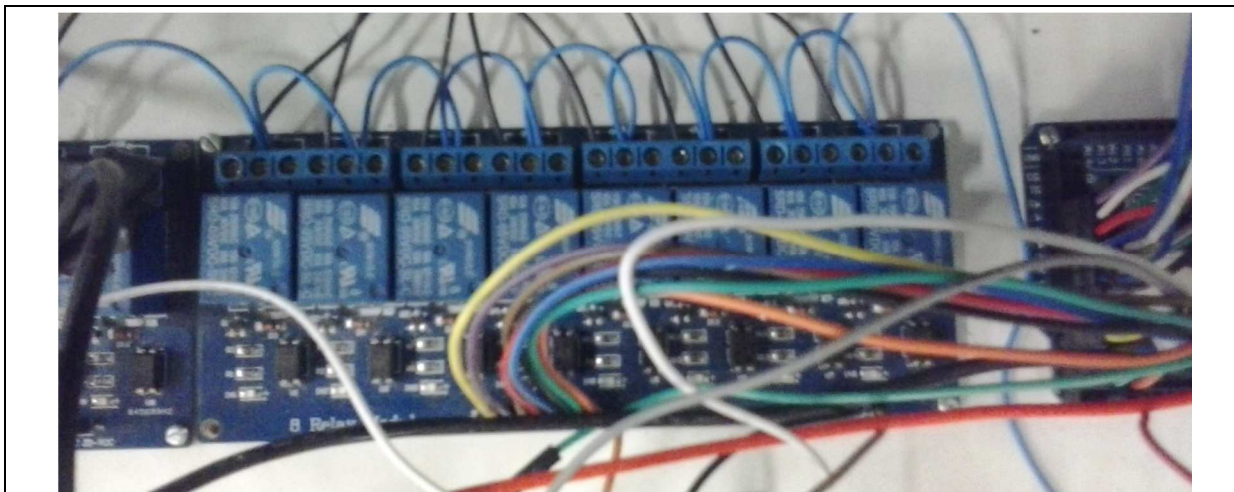


Figura 21 – Módulo com os relés para a iluminação
Fonte: Autores.

Como o circuito todo consumia uma corrente de maior valor, o dispositivo de segurança do Arduino (fusível resetável) reiniciava o sistema sempre que todas as lâmpadas eram ligadas, esse fusível serve como proteção para a porta USB do computador.

Para solucionar este problema foi utilizada uma fonte de tensão de 5 V com corrente máxima de 1 A para realizar a alimentação dos módulos de relés, que pode ser observada na Figura 22.



Figura 22 – Fonte de 5V para alimentar o módulo com os relés
Fonte: Autores.

4.2 SISTEMA DE SEGURANÇA

A simulação dos sensores magnéticos em portas e janelas não era possível devido ao tamanho dos mesmos, e para a simulação foram utilizados botões de trava, que podem ser observados na Figura 23(a). Foram utilizados sensores PIR na área externa da casa, mostrado na Figura 23(b).

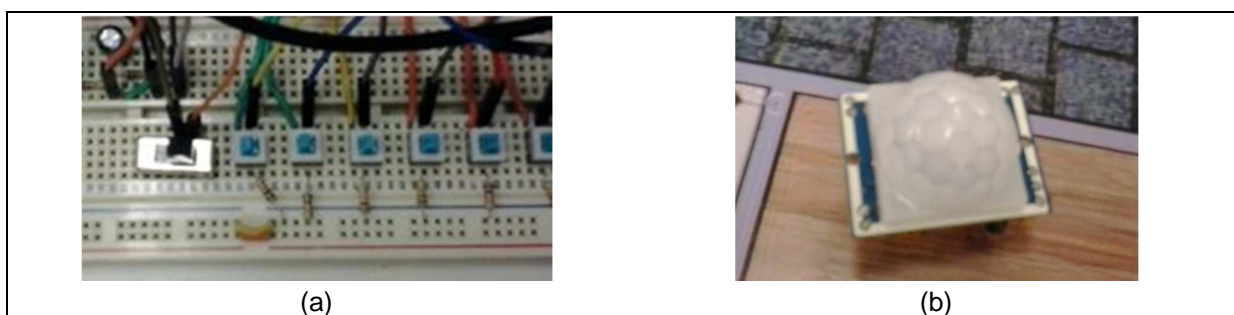


Figura 23 – Sensores para portas e janelas (a) e sensor de presença (b)
Fonte: Autores.

4.3 CONTROLE DO PORTÃO ELETRÔNICO

Ao serem acionados os sensores fim de curso o portão ainda levava um tempo até parar. Após a realização de testes foram definidas as melhores posições para os sensores. Na Figura 24 pode ser observado o funcionamento do portão.



Figura 24 – Portão fechado (a) e portão aberto (b)
Fonte: Autores.

4.4 TESTE DO PROGRAMA FINAL

Após a montagem da maquete e da parte elétrica, cada sistema (iluminação, controle de temperatura, abertura/fechamento do portão e alarme) foi testado individualmente nos primeiros testes.

Na sequência foi realizado o teste do aplicativo desenvolvido. Na figura 25 pode ser observado, à direita, a tela de um smartphone com o aplicativo instalado, indicando que 2 (duas) lâmpadas foram acionadas. À esquerda da imagem podem ser observados 2 LEDs azuis acessos na maquete.

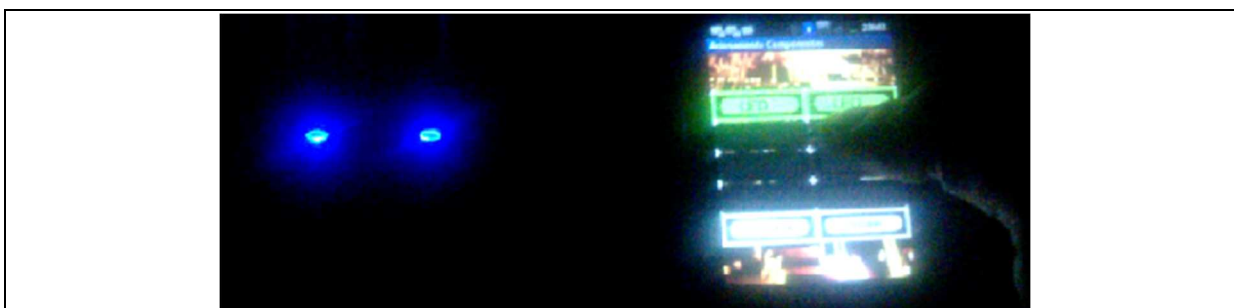


Figura 25 – Teste de acionamento de lâmpadas pelo aplicativo
Fonte: Autores.

Na Figura 26 pode ser observada a página web para se realizar o controle e verificação do acionamento das lâmpadas e o status dos sensores.



Figura 26 – Teste de acionamento de uma lâmpada pela página web
Fonte: Autores.

A seguir foram realizados os testes da comunicação entre a internet, *Ethernet Shield* e o Arduino via computador, para verificar se o *shield* estava corretamente configurado.

Finalmente após o *upload* do programa no microcontrolador, foram realizados os testes de comando para todos os sistemas presentes na maquete, tanto pelo aplicativo quanto pela página na *web*.

Na Figura 28 podem ser observados todos os LEDs acesos, funcionando tanto com acionamentos diretos (pressionando os botões), quanto utilizando o aplicativo do celular ou a página web.

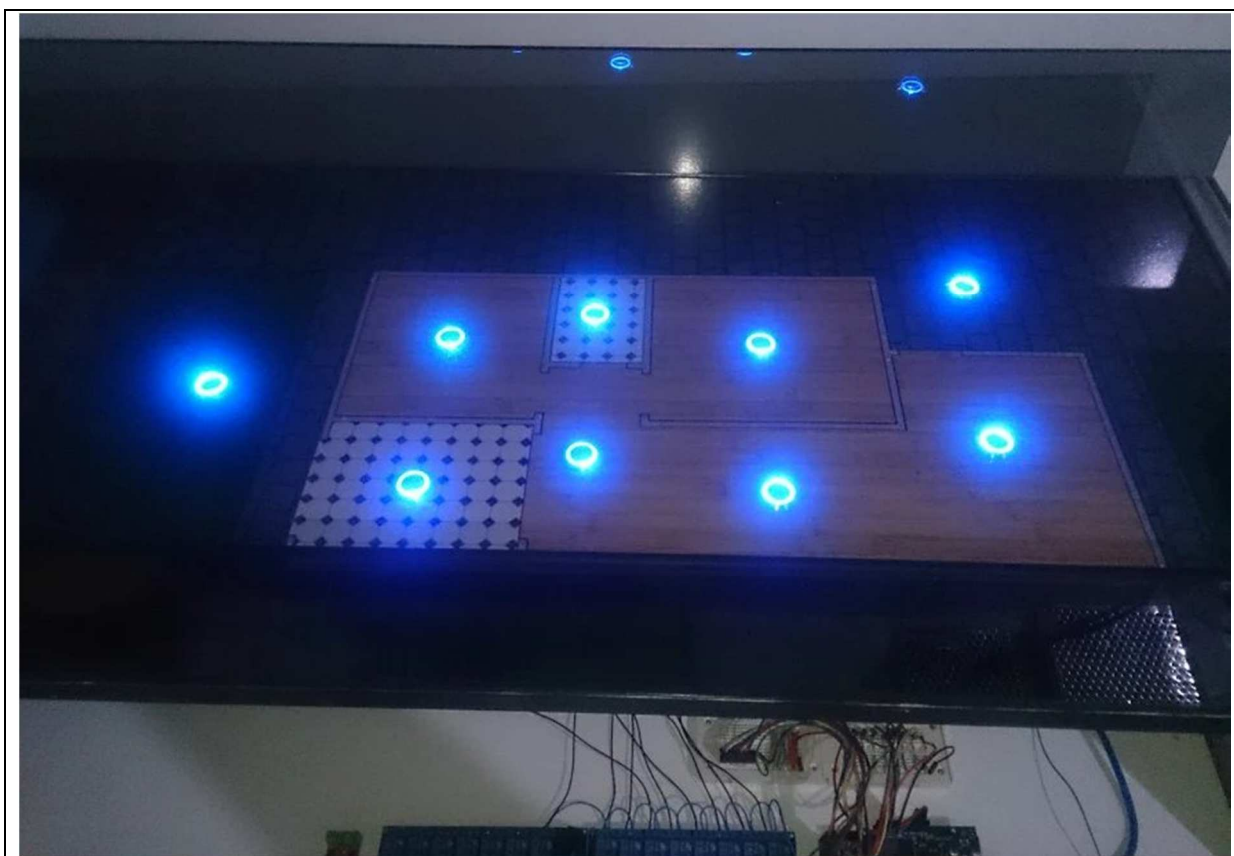


Figura 27 – Funcionamento do LEDs nos testes
Fonte: Autores.

5 CONCLUSÃO

Com a finalização do projeto, pode-se concluir que é possível realizar o controle de processos de uma residência utilizando um sistema microcontrolado, com baixo custo e de fácil acesso no mercado, capaz de transmitir e receber informações e ordens ao sistema por meio de um aplicativo de smartphone ou computador com acesso à internet.

A maior parte dos custos foi com componentes eletrônicos, sendo que os componentes de maior custo total foram os botões de pulso com LED, ou seja, elementos que não são instalados em uma residência em sistema de iluminação, onde são utilizados interruptores e lâmpadas, portanto, além da instalação do sistema microcontrolado não é necessário realizar alterações nos elementos normalmente instalados em uma casa como lâmpadas ou motor do portão.

As ferramentas encontradas para criação do aplicativo e da página na web também podem ser facilmente encontradas sem custo algum, ajudando a manter uma boa relação de custo benefício do projeto.

O sistema também permite saber os gastos com o consumo de energia elétrica, tornando melhor o controle de gastos em caso de instalação de um projeto em uma casa real.

O sistema de alarme se mostrou prático e bastante funcional em relação a sua simplicidade cumprindo a função a que foi projetado. O sistema de controle de temperatura também teve um funcionamento satisfatório tendo uma boa leitura de temperatura e um bom aquecimento da água.

REFERÊNCIAS

ADD THERM – **Sensor Digital de Temperatura**. Disponível em: <http://www.addtherm.com.br/wp-content/uploads/2015/03/A84_SensorDigital_ADD.pdf>. Acesso em: 01 nov. 2016.

ARDUINO. **IDE Arduino**. Disponível em: <<https://www.arduino.cc/en/Guide/Environment>>. Acesso em: 03 nov. 2016.

ARDUINO E CIA. **Ethernet shield Wiznet W5100 - Parte 1**. Disponível em: <<http://www.arduinoecia.com.br/2013/06/ethernet-shield-wiznet-w5100-parte-1.html>>. Acesso em: 02 nov. 2016.

CINESTEC. Disponível em: <http://www.cinestec.com.br/product_info.php?products_id=21736>. Acesso em: 05 nov. 2016.

CRIAR WEB. **Tipos de linguagens de programação** Disponível em: <<http://www.criarweb.com/artigos/685.php>>. Acesso em: 05 nov. 2016.

DEV MEDIA – **Estrutura Básica de uma Página HTML**. Disponível em: <<http://www.devmedia.com.br/html-basico-codigos-html/16596>>. Acesso em: 03 nov. 2016.

FAZEDORES. **Arduino: Conheça essa Plataforma de Hardware Livre e suas Aplicações**. Disponível em: <<http://blog.fazedores.com/arduino-conheca-esta-plataforma-de-hardware-livre-e-suas-aplicacoes/>>. Acesso em: 02 nov. 2016.

GALENO, Arthur; GONÇALVES, Tayná. **Tutorial App Inventor**. 2013 Disponível em: <http://www.pedrocalestini.net/3Vence/AppInventor_3.pdf>. Acesso em: 05 nov. 2016.

LAB DE GARAGEM. **Arduino**. Disponível em: <<http://arduino.labdegaragem.com/>>. Acesso em: 05 nov. 2016.

NEXUS CYBER. **Ethernet Shield**. Disponível em: <<http://www.nexuscyber.com/ethernet-shield-with-micro-sd-slot-for-arduino-uno>>. Acesso em: 05 nov. 2016.

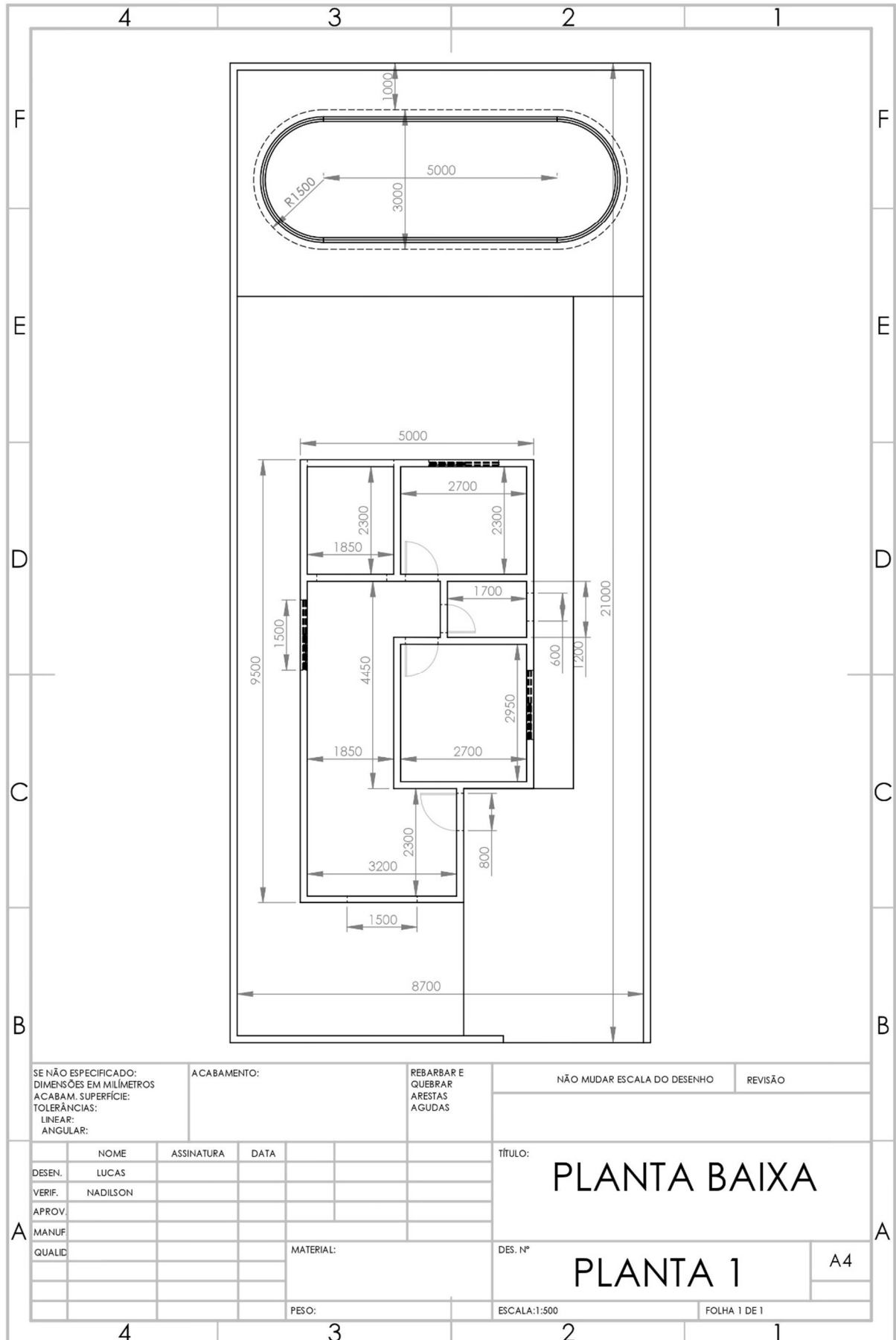
PATSKO, L. F. **Tutorial Montagem da Ponte H**. 2006. Disponível em: <[http://www.maxwellbohr.com.br/downloads/Tutorial Eletronica - Montagem de uma PonteH.pdf](http://www.maxwellbohr.com.br/downloads/Tutorial%20Eletronica%20-%20Montagem%20de%20uma%20PonteH.pdf)>. Acesso em 05 de novembro de 2016.

SCION ELETRONICS. **Arduino Mega2560 R3**. Disponível em: <<http://scionelectronics.com/product/arduino-mega-2560-r3/>>. Acesso em: 05 nov. 2016.

SILVA, Armando Vitor de Oliveira. **Alarme com ativação por sensor presencial e alerta via SMS**. Trabalho de Conclusão de Curso de Graduação. Centro Universitário de Brasília. 2013. 66 páginas.

VECCHI, Hermes F.; OGATA, Reinaldo Jiunji. **Edifícios Inteligentes**. 4ed. Rio de Janeiro, Abril, 1999. Disponível em: <<http://www.din.uem.br/ia/intelige/domotica/int.htm#introducao>>. Acesso em: 28 out. 2016.

APÊNDICE A – PLANTA BAIXA DA MAQUETE



SE NÃO ESPECIFICADO:
DIMENSÕES EM MILÍMETROS
ACABAM. SUPERFÍCIE:
TOLERÂNCIAS:
LINEAR:
ANGULAR:

ACABAMENTO:

REBARBAR E
QUEBRAR
ARESTAS
AGUDAS

NÃO MUDAR ESCALA DO DESENHO

REVISÃO

| | NOME | ASSINATURA | DATA |
|---------|----------|------------|------|
| DESEN. | LUCAS | | |
| VERIF. | NADILSON | | |
| APROV. | | | |
| MANUF. | | | |
| QUALID. | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

TÍTULO:

PLANTA BAIXA

DES. Nº

PLANTA 1

A4

PESO:

ESCALA:1:500

FOLHA 1 DE 1

APÊNDICE B – ESQUEMA ELÉTRICO DO PROJETO

