

**PROGRAMA INSTITUCIONAL DE BOLSAS DE  
INICIAÇÃO CIENTÍFICA E TECNOLÓGICA  
UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE  
DARCY RIBEIRO**

CENTRO CCT  
LABORATÓRIO LCMAT

Relatório do período: 10/2017 a 08/2018

**RELATÓRIO ANUAL**

**Título do Projeto: Estudo e Implementação de Tecnologias  
Assistivas: Desenvolvimento de dispositivos micro  
controlados para portadores de deficiências.**

**Título do Plano de Trabalho: Desenvolvimento de um  
protótipo Arduino para ampliar habilidades funcionais de  
pessoas com deficiências**

Nome do Bolsista/Voluntário: Gabriela Peixoto de Souza  
Curso e N° Matrícula: Ciência da Computação / 00112110945  
Orientador: Fermin Alfredo Tang Montané

Fonte financiadora da Bolsa: PIBIC / FAPERJ

## 1. ETAPAS PROPOSTAS NO PLANO DE TRABALHO

O plano de trabalho proposto compreende as seguintes etapas:

a) Revisão bibliográfica sobre os diferentes tipos de deficiência e suas necessidades, assim como das tecnologias assistivas, envolvendo as diversas áreas de aplicação.

b) Estudo da Plataforma Arduino Uno e afins. Com ênfase nas portas digitais e analógicas e nas possíveis fontes de alimentação. Diagramação de projetos usando ferramentas visuais tais como Fritzing e VBB.

c) Escolha de um projeto específico de desenvolvimento de dispositivo controlado por micro controlador Arduino. Descrição do projeto escolhido, objetivo, alternativas de desenvolvimento e detalhamento das componentes necessárias.

d) Projeto do protótipo eletrônico com base no Arduino Uno. Elaboração do diagrama de circuito usando ferramenta visual. Prototipação usando breadboard. Definição das funcionalidades e programação do dispositivo. Teste funcional.

e) Projeto da estrutura física de suporte para o protótipo eletrônico. Transferência do circuito para uma placa definitiva. Teste funcional.

f) Desenvolvimento de um aplicativo de controle do dispositivo. Escolha da linguagem de programação mais adequada. Definição das funcionalidade e programação do aplicativo. Teste funcional.

g) Realização de experimentos de avaliação e desempenho do dispositivo. Análise sobre possíveis aprimoramentos ou reformulações das etapas d) a f).

h) Elaboração de relatório técnico.

Todas as etapas propostas no plano de trabalho foram realizadas de maneira integral com exceção do item f) que foi abordado de maneira parcial, embora funcional, mas com deficiências na parte estética e acusando falta de memória no dispositivo, sendo necessário mais tempo de pesquisa e outras placas para conexão à internet que foram adquiridas recentemente. Foram desenvolvidos dois tipos de dispositivos baseados no Arduino. Um jogo de memória baseado em cores e sons usado na fase de treinamento com Arduino e um protótipo de acionamento de lâmpadas e ventiladores testado em duas modalidades, pelo acionamento do sensor de presença PIR e por um temporizador. Além disso, desenvolveu-se uma maquete para simular a automação residencial incorporando o controle de lâmpadas.

## 2. INTRODUÇÃO

De acordo com a Organização Mundial da Saúde (OMS), com dados de 2011, cerca de 1 bilhão de pessoas no mundo vivem com algum tipo de deficiência, isso significa uma a cada sete pessoas e a falta de estatísticas sobre elas contribui para sua invisibilidade. Sendo assim, isso representa um obstáculo na hora do planejamento e implementações de políticas de desenvolvimento que melhoram as vidas dessas pessoas.

No Brasil, cerca de 23,9% da população possui pelo menos um tipo de deficiência. Além disso, ter alguma deficiência aumenta o custo de vida em cerca de um terço da renda, em média. Sendo assim, é importante estudar e desenvolver Tecnologias Assistivas de baixo custo como uma forma de proporcionar a essas pessoas uma maior independência, qualidade de vida e inclusão social. E, independentemente do tipo de vulnerabilidade, todos possuem direito de ter uma qualidade de vida melhor.

Pessoas com necessidades especiais possuem limitações motoras, impossibilitando a utilização de meios de comunicação. Através das tecnologias, as pessoas portadoras de deficiência, poderão participar de atividades envolvendo trabalho, estudo e lazer. Portanto, desenvolver novas técnicas é uma maneira de amenizar os problemas causados pela deficiência colocando essas pessoas em contato com o mundo e dando a elas novas condições de vida, tornando a vivência mais fácil e prazerosa.

Desde a década de 60 alguns países discutem maneiras de reduzir as barreiras que alguns ambientes de convivência impõem às pessoas com algum tipo de necessidade especial. Para isso, existe um conceito chamado *Universal Design* com o intuito de projetar ambientes que possam ser utilizados tanto por pessoas comuns quanto por pessoas com necessidades especiais sem que isso exija adaptações significativas para estes últimos. A evolução do Universal Design começou na década de 1950 com uma nova atenção ao design para pessoas com deficiências.

### 3. OBJETIVOS

O presente trabalho tem como objetivo desenvolver a capacidade de construir um dispositivo micro controlado com base na plataforma Arduino para que se possa ampliar as habilidades funcionais dos portadores de deficiências. Para isso deverá ser realizado um estudo dos diferentes tipos de deficiências assim como das tecnologias assistivas disponíveis, fazer um levantamento de projetos com base neste tipo de tecnologia escolhendo-se um que se adeque aos recursos disponíveis e as necessidades dos portadores de deficiências, além de trabalhar todas as etapas de construção de um dispositivo micro controlado, desde o desenvolvimento do circuito eletrônico, construção da estrutura física de suporte e programação do micro controlador.

### 4. METODOLOGIA

A metodologia do projeto gira em torno do estudo sobre os diferentes tipos de deficiência e suas necessidades, assim como das tecnologias assistivas, da plataforma Arduino e a construção de dispositivos eletrônicos aplicando conceitos de eletrônica básica assim como princípios de programação em uma linguagem baseada em C/C++ específica para a plataforma Arduino.

De forma a cumprir os objetivos apontados, a pesquisa se deu início com consultas em livros e artigos disponibilizados pelo orientador em relação a plataforma Arduino e aos tipos de deficiências existentes para poder decidir qual tipo seria abordada inicialmente, sendo escolhida a deficiência física, e seguiu com a execução de práticas simples, para posteriormente realizar outras mais elaboradas como a construção do Jogo de Memória Genius com capacidade de reproduzir sons, acionamento de uma lâmpada e um ventilador por meio de um sensor de presença e um módulo relé e o acionamento de um conjunto de lâmpadas via internet utilizando uma placa *ethernet*.

Como início de trabalho foram realizadas pesquisas sobre projetos existentes que contém componentes micro controlados que auxiliam o portador de deficiência a partir da plataforma Arduino de forma a ser encontrada ideias que poderiam servir como auxílio na construção do protótipo. O primeiro projeto encontrado consiste em uma bengala eletrônica feita para deficientes visuais com o intuito de facilitar a locomoção deles. Esse projeto funciona a partir de um sensor ultrassônico que envia um sinal em forma de onda e quando encontra um objeto em sua direção o sinal é rebatido e retorna ao receptor, de forma a percorrer duas vezes a distância procurada. O segundo projeto é bem parecido com o anterior, porém a detecção se dá por meio de uma luva que irá fazer a medição da distância com o sensor ultrassônico e iria traduzir isso em forma de pressão no pulso do deficiente, ou seja, quanto menor a distância de um objeto, maior será a pressão sofrida em seu pulso.

A deficiência física pode ter causa hereditária (quando resulta de doenças transmitidas por genes, podendo manifestar-se desde o nascimento, ou aparecer posteriormente), congênita (quando existe no indivíduo ao nascer e, mais

comumente, antes de nascer, isto é, durante a fase intrauterina) e adquirida (quando ocorre depois do nascimento, em virtude de infecções, traumatismos, intoxicações).

Para a pessoa com deficiência, ainda há muitas iniciativas que precisam ser tomadas no que se refere, sobretudo ao seu deslocamento e acesso através de portas, rampas e caminhos. Essas demandas podem ser atendidas por meio de uma arquitetura inclusiva que leve em conta as inúmeras situações envolvidas. A partir desse pensamento, veio a ideia de fazer com que as pessoas com algum tipo de deficiência sintam-se à vontade dentro de sua própria residência. E como forma inicial está sendo abordada a deficiência física primeiro para que possa ser atendidas aos poucos as necessidades de cada um.

A adaptação de uma residência é muito mais do que tarefas que precisam de cálculos e trabalho manual, é utilizar o cérebro e a criatividade para pensar em melhorias e inovações. Além disso, é uma forma de padronizar residências de forma a ser utilizada por qualquer pessoa, já que uma deficiência pode ser adquirida ao longo da vida e adaptações inesperadas não serão necessárias.

As práticas realizadas contêm componentes que serão utilizados no projeto de pesquisa, de forma a conhecê-los separadamente em outras atividades.

O primeiro dispositivo realizado para estudo teve como proposta iniciar o uso do Arduino de forma simples e interativa conhecendo alguns componentes bastante utilizados em diversos projetos e consistiu no Genius, um jogo de memória com o intuito de estimular a memorização de cores e sons.

#### 4.1. PLATAFORMA ARDUINO

O Arduino foi criado no ano de 2005, na Itália, por Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis. As principais exigências eram que fosse um dispositivo barato e também uma plataforma que pudesse ser utilizada por qualquer pessoa. De acordo com [1], sua popularidade cresceu rapidamente quando o público percebeu que o Arduino era um sistema de fácil utilização, de baixo custo e que poderia ser usado em seus próprios projetos, assim como, era uma excelente introdução para programação de micro controladores. A Figura 1 mostra o layout da placa do Arduino Uno.



Figura 1 – Placa Arduino Uno – Fonte [2]

Abaixo podemos observar as especificações técnicas do Arduino Uno:

- Micro controlador: ATmega328, Tensão de Operação: 5V, Tensão de Entrada: 7-12V, Portas Digitais: 14, Portas Analógicas: 6, Corrente Pinos I/O: 40mA, Corrente Pinos 3,3V: 50mA, Memória Flash: 32KB.

O Arduino possui diversos pinos, dentre eles temos uma de tensão de 5V e outra de 3,3V que é utilizado para conectar os componentes externos e um pino para o terra que chamamos de GND (Graduated Neutral Density Filter), "Filtro Graduado de Densidade Neutra", que é utilizado para fechar a corrente dos circuitos e também conhecido como *ground*.

Para ser feita a utilização do Arduino é preciso conectar a placa ao computador por meio de um cabo USB e utilizar um ambiente de programação chamado Ambiente Integrado de Desenvolvimento (IDE), onde é possível digitar o programa, fazer testes para encontrar eventuais erros e transferir para o Arduino. Na figura 2 temos a IDE, mostrando sua tela inicial.



Figura 2 – IDE do Arduino – Fonte: Própria

Quando se começa um novo programa sua estrutura básica é composta por duas partes, o **setup()** e o **loop()**. A IDE traz um monitor serial que auxilia no recebimento e envio de dados para a placa sem a necessidade de recorrer a uma ferramenta externa. A Figura 3 mostra a janela apresentada:

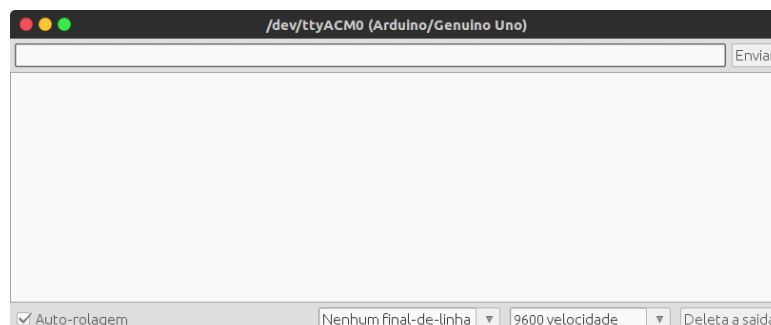


Figura 3 – Monitor Serial da IDE – Fonte: Própria

## 4.2. COMPONENTES BÁSICOS

Nesta seção são apresentados componentes básicos utilizados para a construção de dispositivos desenvolvidos para a plataforma Arduino. O

conhecimento das características dos componentes permite que eles possam ser utilizados de maneira adequada, evitando possíveis danos a outros dispositivos.

### Protoboard

A protoboard, também conhecida como matriz de contatos, é uma placa com furos e conexões condutoras para montagens provisórias e testes de projetos e possui uma facilidade na hora de inserir e retirar seus componentes, já que não requer soldagem. Ela é constituída por uma base plástica que contém inúmeros orifícios destinados à inserção dos componentes eletrônicos. Dentro da protoboard existem determinadas ligações que interconectam esses orifícios, permitindo a montagem de circuitos eletrônicos. A Figura 4 mostra uma protoboard de 830 pontos, bastante comum no meio eletrônico.

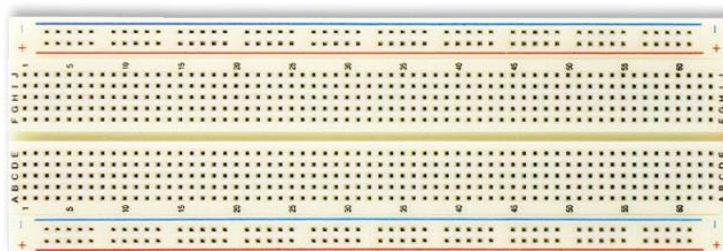


Figura 4 – Protoboard – Fonte [3]

### LEDs

O LED (Diodo Emissor de Luz) é utilizado para emitir luz e em componentes onde se torna mais conveniente a sua utilização do que a de uma lâmpada. O diodo é um componente eletrônico que faz com que a corrente elétrica circule em um único sentido. Na Figura 5 encontra-se um diodo emissor de luz.



Figura 5 – LEDs – Fonte [4]

### Resistores

Os resistores são componentes eletrônicos bem comuns na eletrônica. Eles não são polarizados, ou seja, não possuem polo positivo e negativo. O resistor tem como função limitar o fluxo de corrente elétrica que passa por ele, essa limitação é chamada de resistência e é medida em *ohms*. Portanto, ela define qual a facilidade ou dificuldade que os elétrons terão ao passar pelo resistor. Quanto maior o valor da resistência mais difícil será para os elétrons passarem pelo resistor e quanto menor o valor da resistência mais fácil será para os elétrons passarem.

Existem diversos resistores com diferentes valores em *ohms* e ele é escolhido para fazer parte do circuito de acordo com a tensão que queremos utilizar, além disso, podemos encontrar resistores de 4 ou 5 faixas de cores. Na Figura 6 temos um resistor de 330 ohms.



Figura 6 – Resistor de 330 ohms – Fonte [5]

Já a Figura 7, mostra a tabela com o código de cores utilizado para rotular os resistores. Para determinar o valor de um resistor, basta observar as cores das faixas sobre ele e consultar a tabela, segundo as seguintes orientações:

- 1ª Faixa de cor: Mostra o primeiro algarismo do valor da resistência
- 2ª Faixa de cor: Mostra o segundo algarismo do valor da resistência
- 3ª Faixa de cor: Mostra o multiplicador para o número formado pelas faixas 1 e 2.

Cor	1ª Faixa	2ª Faixa	Nº de zeros/multiplicador	Tolerância
Preto	0	0	0	
Marrom	1	1	1	
Vermelho	2	2	2	
Laranja	3	3	3	
Amarelo	4	4	4	
Verde	5	5	5	
Azul	6	6	6	
Violeta	7	7	7	
Cinza	8	8	8	
Branco	9	9	9	
Dourado			x0,1	
Prata			x0,01	
Sem cor				± 20%

Figura 7 – Tabela de cores de um resistor – Fonte [6]

Por exemplo, o resistor de 330 *ohm* mostrado na Figura 4, tem as cores laranja, preto e marrom na 1ª, 2ª e 3ª faixas.

Por outro lado, dada a voltagem de uma certa componente, para descobrir qual resistor usar, precisamos utilizar a seguinte fórmula:

$$R = (V - V_c) / L$$

Onde:

R: valor da resistência (em *ohms*) adequado para o componente pesquisado;

V: tensão (em *volts*) da fonte de energia que alimenta o componente;

V<sub>c</sub>: tensão (em *volts*) suportada pelo componente;

L: corrente do componente (em amperes).

### Chave tátil

Este botão é um tipo de interruptor pulsador, pois conduz corrente somente quando está pressionado. A Figura 8 mostra o modelo de chave tátil.



Figura 8 – Chave Tátil – Fonte [7]



Ele é comumente empregado como um botão eletrônico que possui a funcionalidade de um interruptor elétrico. Com dois contatos em seu interior, possui a forma de uma chave tátil comumente aberta, portanto, sempre que pressionado o botão o circuito se fecha.

### Jumper

O Jumper é utilizado para fazer uma ligação entre dois pontos de um circuito eletrônico. Ele é muito utilizado em protótipos de projetos que utilizam a protoboard e pode assumir vários formatos. A Figura 9 ilustra vários jumpers.



Figura 9 – Jumper – Fonte [8]

### Piezo

É um sensor que possui a capacidade de fazer a detecção de toques e vibrações, utilizando as informações como tensão de saída. Podem funcionar também em sentido inverso, utilizando uma tensão de entrada para reproduzir sons mediante vibrações. Ele é encontrado em formato de disco com dois fios. Um vermelho, o positivo, e um preto, o negativo. A Figura 10 mostra um piezo elétrico.



Figura 10 – Piezo elétrico – Fonte [9]

## 4.2. PROJETO DO JOGO GENIUS

O jogo da memória genius era um brinquedo muito popular na década de 1980. O jogo desenvolvido com a plataforma Arduino contempla o aspecto de cores e sons. Sendo composto por alguns botões e leds coloridos.

O jogo é formado por 4 leds um de cada cor (vermelho, amarelo, verde e branco) e 4 botões para acender cada led e um piezo para emitir sons associados, tendo como objetivo repetir corretamente um sequência de luzes gerada de maneira aleatória, para estimular a memória.

O processo é repetido incrementando-se o nível de dificuldade (número de leds na sequência a ser memorizada) e termina quando o jogador errar a sequência ou conseguir reproduzir corretamente a maior sequência programada no jogo.

### Componentes utilizados

Foram utilizados os seguintes componentes, descritos na seção anterior:

- 1 Arduino Uno com cabo USB
- 1 Protoboard e Jumpers variados
- 4 Leds (Vermelho, Amarelo, Verde e Branco)
- 4 Resistores de 330 ohms
- 4 Chaves tátil
- 1 Piezo

### Diagrama de Conexão

A Figura 11 ilustra o diagrama de conexão entre a placa Arduino e as componentes que fazem parte do Genius. Basicamente, temos 4 pinos digitais da placa Arduino (parte superior direita) sendo usados como OUTPUT (saída) para alimentar cada um dos 4 leds. Os 4 resistores são usados para garantir a voltagem certa em cada led, pois a voltagem de saída padrão (5V) supera a voltagem suportada pelos leds. Outros 4 pinos digitais (parte superior esquerda) são usados como INPUT (entrada) para registrar a pulsação de cada um dos 4 botões. Na parte inferior da placa, encontra-se a linha do terra (preta).

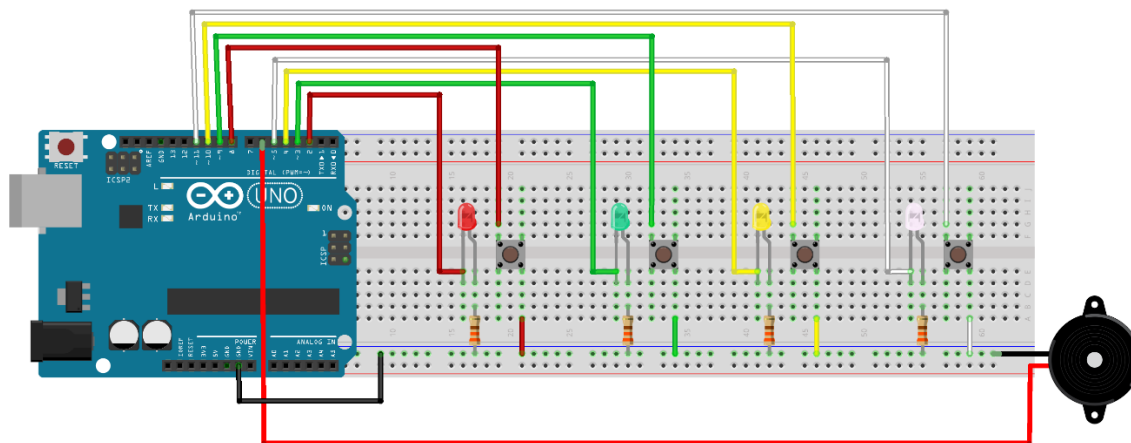


Figura 11 – Diagrama do Jogo Genius – Fonte: Própria

### Código de Controle para o Arduino

O código de controle do dispositivo é mostrado a seguir. No início o programa faz a inicialização dos leds, dos botões e do piezo. Define-se o tamanho máximo da sequência como 4, um conjunto de quatro estados possíveis para o jogo, um vetor para armazenar a sequência gerada, duas opções de tempo e o número de rodada assim como o número de cores respondidas.

```
#define LED_VERMELHO 8
#define LED_VERDE 9
#define LED_AMARELO 10
#define LED_BRANCO 11
#define INDEFINIDO -1
#define PIEZO 6
#define BOTAO_VERMELHO 2
#define BOTAO_VERDE 3
#define BOTAO_AMARELO 4
#define BOTAO_BRANCO 5
#define UM_SEGUNDO 1000
#define MEIO_SEGUNDO 500
#define TAMANHO_SEQ 5
enum Estados {PRONTO_PARA_PROXIMA_RODADA, USUARIO_RESPONDENDO,
JOGO_FINALIZADO_SUCESSO, JOGO_FINALIZADO_FALHA};
int sequenciaLuzes[TAMANHO_SEQ];
int rodada = 0;
int ledsRespondidos = 0;
```

Na parte de inicialização (Setup), inicializam-se: o monitor serial, os pinos conectados e inicia-se o jogo, que consiste em gerar e armazenar uma sequência aleatória de 5 números entre 2 e 5.

```
void setup() {
    Serial.begin(9600);
    iniciaPortas();    iniciaJogo();
}
void iniciaPortas() {
    pinMode(LED_VERMELHO, OUTPUT);    pinMode(LED_VERDE, OUTPUT);
    pinMode(LED_AMARELO, OUTPUT);    pinMode(LED_BRANCO, OUTPUT);
    pinMode(BOTAO_VERMELHO, INPUT_PULLUP);
    pinMode(BOTAO_VERDE, INPUT_PULLUP);
    pinMode(BOTAO_AMARELO, INPUT_PULLUP);
    pinMode(BOTAO_BRANCO, INPUT_PULLUP);
    pinMode(PIEZO, OUTPUT);
}
void iniciaJogo() {
    int jogo = analogRead(0);
    randomSeed(jogo);
    for (int i = 0; i < TAMANHO_SEQ; i++) {
        sequenciaLuzes[i] = sorteiaCor();
    }
}
int sorteiaCor() { return random(LED_VERMELHO, LED_BRANCO + 1); }
```

Na parte de repetição (Loop), verifica-se a ocorrência de quatro casos possíveis para o usuário, então usa-se a função estadoAtual(). Temos os seguintes casos: i) completou a rodada atual, preparaNovaRodada(), ii) ainda está respondendo a rodada atual, processaRespostaUsuario(); iii) concluiu a sequência completa de cores sem errar, jogoFinalizadoSucesso() e iv) errou antes de concluir a sequência de cores completa, jogoFinalizadoFalha().

```

void loop() {
    switch (estadoAtual()) {
        case PRONTO_PARA_PROXIMA_RODADA:
            Serial.println("Pronto para a proxima rodada"); preparaNovaRodada();
            break;
        case USUARIO_RESPONDENDO:
            Serial.println("Usuario respondendo"); processaRespostaUsuario();
            break;
        case JOGO_FINALIZADO_SUCESSO:
            Serial.println("Jogo finalizado com sucesso"); jogoFinalizadoSucesso();
            break;
        case JOGO_FINALIZADO_FALHA:
            Serial.println("Jogo finalizado com falha"); jogoFinalizadoFalha();
            break;
    }
    delay(MEIO_SEGUNDO);
}

```

Na função `estadoAtual()`, se a rodada atual é menor ou igual que o número de cores na sequência, verifica-se se todas as cores dessa rodada já foram respondidas corretamente. Se for verdade prepara-se a próxima rodada. Caso contrário, processa a próxima resposta do usuário. Já se a rodada atual for maior que o número de cores na sequência em uma unidade, considera-se que todas as cores foram acertadas e o jogo termina. Enquanto que, se a rodada atual for maior em duas unidades que o número de cores significa que foi identificado um erro na cor da última jogada.

```

int estadoAtual() {
    if (rodada <= TAMANHO_SEQ) {
        if (ledsRespondidos == rodada) { return PRONTO_PARA_PROXIMA_RODADA; }
        else { return USUARIO_RESPONDENDO; }
    }
    else if (rodada == TAMANHO_SEQ + 1) {
        return JOGO_FINALIZADO_SUCESSO;
    }
    else { return JOGO_FINALIZADO_FALHA; }
}

```

A função `preparaNovaRodada()` incrementa um o número de cores a ser reproduzido e zera o contador de respostas corretas do jogador. A função `processaRespostaUsuario()` verifica o estado dos botões, ou seja, se a cor pressionada é igual à da sequência nessa rodada. Em caso de acerto incrementa o contador de cores acertadas, caso contrário, define a condição de erro.

```

void preparaNovaRodada() {
    rodada++; ledsRespondidos = 0;
    if (rodada <= TAMANHO_SEQ) { tocaLedsRodada(); }
}
void processaRespostaUsuario() {
    int resposta = checaRespostaJogador();
    if (resposta == INDEFINIDO) { return; }
    if (resposta == sequenciaLuzes[ledsRespondidos]) { ledsRespondidos++; }
    else {
        Serial.println("Sequencia errada"); rodada = TAMANHO_SEQ + 2;
    }
}
int checaRespostaJogador() {
    if (digitalRead(BOTAO_VERMELHO) == LOW) {return piscaLed(LED_VERMELHO); }
    if (digitalRead(BOTAO_VERDE) == LOW) { return piscaLed(LED_VERDE); }
}

```

```

    if (digitalRead(BOTAO_AMARELO) == LOW) {return piscaLed(LED_AMARELO); }
    if (digitalRead(BOTAO_BRANCO) == LOW) { return piscaLed(LED_BRANCO); }
    return INDEFINIDO;
}

```

As funções `tocaLedsRodada()` e `piscaLed()` acendem leds durante um segundo, um de cada vez. Enquanto, a primeira função acende uma sequência de *leds*, a segunda acende um *led* individual. As funções `tocaSom()` e `verificaSomDoLed()` geram um som por vez na saída do piezo durante meio segundo. Enquanto, a primeira função gera um tom de áudio no piezo, a segunda envia um som com diferentes frequências de acordo com cada led individual para o piezo reproduzir um por vez.

```

void tocaLedsRodada() {
    for (int i = 0; i < rodada; i++) { piscaLed(sequenciaLuzes[i]); }
}
void tocaSom(int frequencia) { tone(PIEZO,frequencia, 500); }
int piscaLed(int portaLed) {
    verificaSomDoLed(portaLed);
    digitalWrite(portaLed,HIGH); delay(UM_SEGUNDO);
    digitalWrite(portaLed,LOW); delay(MEIO_SEGUNDO);
    return portaLed;
}

void verificaSomDoLed(int portaLed) {
    switch (portaLed) {
        case LED_VERMELHO:
            tocaSom(2400); break;
        case LED_VERDE:
            tocaSom(2000); break;
        case LED_AMARELO:
            tocaSom(2500); break;
        case LED_BRANCO:
            tocaSom(2200); break;
    }
}

```

Por fim, as funções `jogoFinalizadoSucesso()` e `jogoFinalizadoFalha()` emitem sinais luminosos para indicar o fim do jogo com sucesso e com erro. No primeiro caso, as quatro cores são acendidas em sequência. No segundo, as quatro cores são acendidas ao mesmo tempo.

```

void jogoFinalizadoSucesso() {
    piscaLed(LED_VERMELHO);    piscaLed(LED_VERDE);
    piscaLed(LED_AZUL);        piscaLed(LED_AMARELO);
    delay(MEIO_SEGUNDO);
}
void jogoFinalizadoFalha() {
    tocaSom(300);
    digitalWrite(LED_VERMELHO,HIGH);    digitalWrite(LED_VERDE,HIGH);
    digitalWrite(LED_AZUL,HIGH);        digitalWrite(LED_AMARELO,HIGH);
    delay(UM_SEGUNDO);
    digitalWrite(LED_VERMELHO,LOW);    digitalWrite(LED_VERDE,LOW);
    digitalWrite(LED_AZUL,LOW);        digitalWrite(LED_AMARELO,LOW);
    delay(MEIO_SEGUNDO);
}

```

### Protótipo do dispositivo

A Figura 12 ilustra o protótipo do Jogo Genius finalizado.

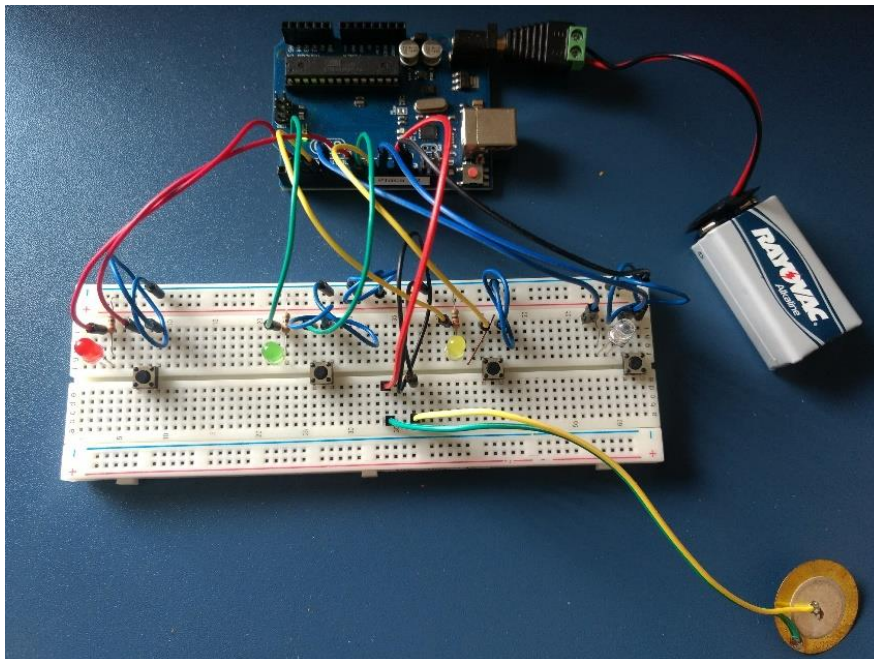


Figura 12 – Protótipo do Jogo Genius – Fonte: Própria

## 5. RESULTADOS E DISCUSSÃO

Como resultado desta pesquisa além dos vários estudos referentes à plataforma Arduino e em relação aos tipos de deficiência, com foco na deficiência física, foram desenvolvidos três dispositivos micro controlados, todos eles escolhidos em consenso com o orientador, de acordo com a disponibilidade de componentes eletrônicos e viabilidade financeira. O primeiro dispositivo que faz parte do início do projeto central, consistiu na montagem de um sensor de presença a partir de um módulo sensor de movimento PIR para acender uma lâmpada e um ventilador através de um módulo relé. O segundo dispositivo consistiu na montagem de um circuito com acionamento temporizado de 2 lâmpadas e a partir dele foi inserido uma placa *ethernet* fazendo com que esse acionamento seja feito a partir de uma página web. Os experimentos realizados neste projeto mostraram a viabilidade do controle de lâmpadas através da internet, mas evidenciou alguns problemas como falta de memória no micro controlador e a necessidade de adquirir maior conhecimentos sobre outras alternativas de conexão com a internet, assim como as componentes eletrônicas para essa finalidade. O terceiro dispositivo consiste no projeto principal, no qual foi feita a construção de uma maquete de uma residência que incluirá diversos ambientes de forma a auxiliar as principais necessidades de um deficiente físico a partir da automação residencial.

A necessidade de automação residencial tem aumentado nos últimos anos devido à crescente demanda por conectividade entre dispositivos móveis, como tablets, smartphones, e outros eletrônicos. Trata-se da aplicação de sistemas de controle baseados na automação para todas as funções encontradas no ambiente, integrando seus acionamentos e visando sempre a praticidade, simplicidade e objetividade de comandos.

Nesta seção serão apresentados os detalhes sobre cada um dos três projetos, contemplando: i) Componentes utilizadas, ii) Diagrama do dispositivo e iii) Código em linguagem C para o micro controlador e iv) Protótipo do dispositivo.

### 5.1. AUTOMAÇÃO RESIDENCIAL: ACIONAMENTO DE LÂMPADA E UM VENTILADOR COM SENSOR PIR

Neste projeto foi feita a montagem de um sensor de presença a partir de um módulo sensor de movimento PIR com acionamento de uma lâmpada fluorescente e um ventilador por meio de um relé. Quando o sensor de presença PIR detecta algum movimento a lâmpada e o ventilador são ligados automaticamente sem nenhuma interferência humana, e ficam ligados por cerca de 2min, podendo esse tempo ser alterado no sensor.

A automatização de um ambiente é uma forma de auxiliar deficientes físicos, por exemplo, a se locomoverem com mais facilidade em sua residência. Além disso, a prática tem como objetivo iniciar o estudo de sensores e ver se algum irá se adequar melhor ao projeto de pesquisa.

### Componentes utilizados

Foram utilizados os seguintes componentes:

- 1 Arduino Uno com cabo USB
- 1 Protoboard e Jumpers variados
- 1 Módulo relé Arduino de 1 canal
- 1 Módulo sensor de movimento PIR
- Cabo elétrico flexível
- 1 Lâmpada fluorescente
- 1 Plugue de tomada
- 1 Bocal
- 1 Ventilador

#### i) Módulo relé Arduino

O Módulo Relé é ideal para acionar um componente que exija até no máximo 10A contínuos utilizando o Arduino e ele funciona como um interruptor.

Na Figura 13 pode ser observado os pinos do módulo relé Arduino. Do lado esquerdo temos os pinos Power LED (relé ligado), Vcc (fonte 5V), GND (terra), IN1 (aciona o relé) e o Relay LED (status do relé acionado). Ao lado direito, os contatos NC (Normal Fechado), C (Comum), e NO (Normal Aberto).



Figura 13 – Módulo relé de 1 canal e sua pinagem – Fonte [10]

Abaixo podemos observar as especificações técnicas do Módulo Relé:

- Tensão de operação: 5V, Tensão máxima de carga: 240VAC, Corrente máxima de carga: 10A, Dimensões 34 x 27 x 17mm.

#### ii) Módulo sensor de movimento PIR

O módulo sensor de movimento PIR é um módulo robusto que contém sensor infravermelho e ajustes de sensibilidade e tempo de acionamento. Na Figura 14 podemos observar um modelo do sensor de movimento PIR.



Figura 14 – Módulo sensor de movimento PIR – Fonte [11]

Abaixo podemos observar as especificações técnicas do sensor PIR:

- Modelo: DYP-ME003, Sensor Infravermelho com controle na placa, Sensibilidade e tempo ajustável, Tensão de Operação: 4,5-20V, Tensão Dados: 3,3V (Alto) – 0V (Baixo), Distância detectável: 3-7m (Ajustável), Tempo de Delay: 5-200seg (Default: 5seg), Tempo de Bloqueio: 2,5seg, Temperatura de Trabalho: -20 ~ +80°C, Dimensões: 3,2 x 2,4 x 1,8cm, Peso: 7g.



Este componente permite o ajuste da sensibilidade de seu sensor, ou seja, qual será a distância em que um objeto será detectado, podendo ela variar de 3 a 7 metros, e também o tempo de *delay*, ou seja, o tempo em que sua saída permanecerá acionada, podendo variar de 5 a 200 segundos. Os ajustes são feitos a partir dos 2 potenciômetros soldados à placa. Observando a figura 4.13 é possível ver os potenciômetros na cor laranja, o da esquerda ajusta o tempo e o da direita ajusta a sensibilidade.

A ligação com o Arduino utiliza apenas um pino, que apresenta o estado HIGH (alto), ao detectar um movimento, e LOW (baixo) quando não há movimento próximo ao sensor.

#### Diagrama de Conexão

A Figura 15 ilustra o diagrama de conexão entre a placa Arduino e as componentes que fazem parte do projeto. Basicamente, temos 2 pinos digitais da placa Arduino (parte superior direita) sendo usados como OUTPUT (saída) para o módulo relé e INPUT (entrada) para o sensor PIR. A lâmpada, que está conectada em série com o ventilador, é ligada no contato NO de um lado e o ventilador é ligado ao terra da rede elétrica de 110V. O Contato comum do relé também é ligado à rede elétrica de 110V. Na parte inferior da placa, encontram-se as linhas de alimentação de energia contínua de 5V e o terra (vermelha e preta) ligados ao sensor e ao módulo relé.

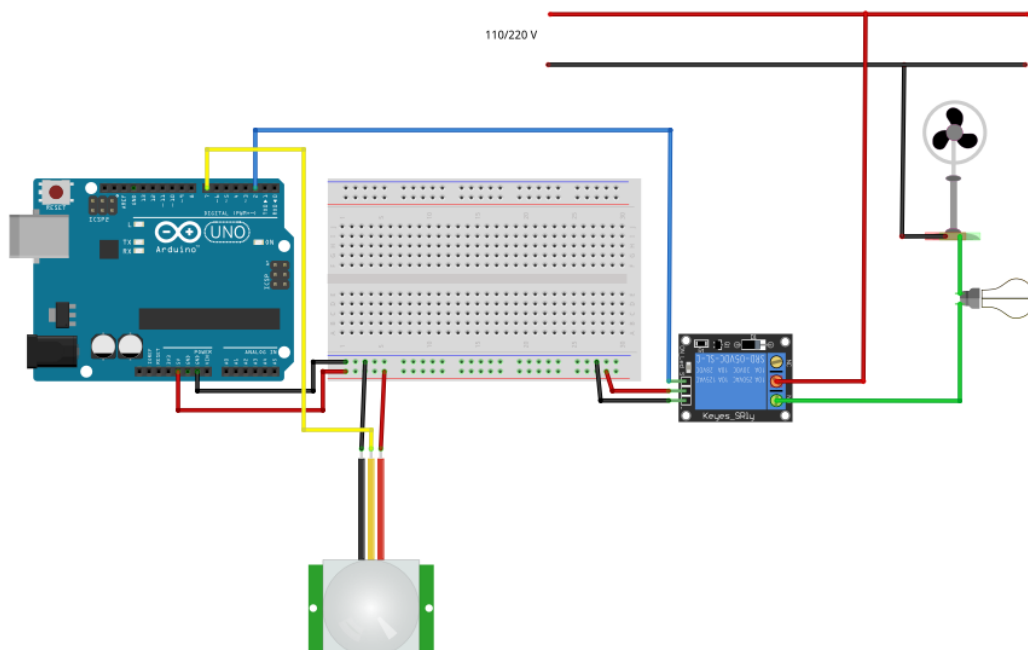


Figura 15 – Diagrama do acionamento de uma lâmpada e um ventilador com sensor de presença PIR – Fonte: Própria

#### Código de Controle para o Arduino

O código para o acionamento basicamente, verifica se a entrada do sensor de movimento foi acionada, ou seja, se está em estado alto (HIGH), e então aciona a saída para o relé, que por sua vez aciona o relé e acende a lâmpada e liga o ventilador.

```

int pinorele = 2; //Pino ligado ao rele
int pinopir = 7; //Pino ligado ao sensor PIR
int acionamento; //Variavel para guardar valor do sensor
void setup() {
    pinMode(pinorele, OUTPUT); //Define pino rele como saída
    pinMode(pinopir, INPUT); //Define pino sensor como entrada
}
void loop(){
    acionamento = digitalRead(pinopir); //Le o valor do sensor PIR
    if (acionamento == LOW) { //Sem movimento, mantém rele desligado
        digitalWrite(pinorele, LOW);
    }
    else { //Caso seja detectado um movimento, aciona o rele
        digitalWrite(pinorele, HIGH);
    }
}
}

```

### Protótipo do dispositivo

A Figura 16 ilustra o protótipo do acionamento de uma lâmpada e um ventilador por meio de um relé finalizado. A alimentação da placa Arduino é feita por meio de uma bateria de 9V e a lâmpada e o ventilador são alimentados por uma tomada da rede elétrica.

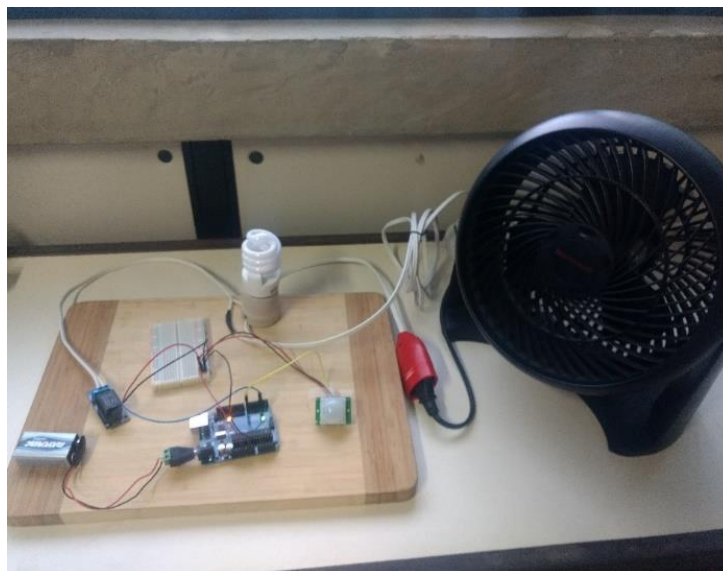


Figura 16 – Protótipo do acionamento de uma lâmpada e um ventilador com sensor de presença PIR – Fonte: Própria

## 5.2. AUTOMAÇÃO RESIDENCIAL: ACIONAMENTO DE LÂMPADA ATRAVÉS DA INTERNET

O projeto tem como foco adaptar um dispositivo de controle de lâmpadas de iluminação doméstica através da incorporação de um controle através da *internet*. Para isso, foi realizado acionamento remoto de duas lâmpadas por meio

de uma página *web* a partir de dois botões LIGA e DESLIGA através de um módulo relé e uma placa *Ethernet Shield*. Antes de realizar a conexão com a *internet* foi feito inicialmente o acionamento automático alternado com ambas as lâmpadas para verificar o funcionamento do circuito. O trabalho teve como foco testes de conectividade e a criação de uma página *web* com dois botões para acender cada lâmpada individualmente. Os testes não foram totalmente satisfatórios, embora fosse comprovada a conexão à *internet*, a página proposta apresentou problemas por falta de memória no micro-controlador.

#### Componentes utilizados

Foram utilizados os seguintes componentes:

- 1 Arduino Uno com cabo USB
- 1 Protoboard e Jumpers variados
- 1 Módulo Relé Arduino de 2 canais
- 2 Lâmpadas fluorescentes
- 2 Bocais
- 1 Plugue de tomada
- Cabo elétrico flexível
- 1 placa Ethernet Shield W5100

#### i) *Ethernet Shield* W5100

É uma placa prática quando se trata de tornar o Arduino on-line. É baseada no chip WIZnet *ethernet* W5100, que fornece acesso à rede nos protocolos TCP ou UDP. Duas bibliotecas nativas são usadas para poder usar o módulo facilmente: *Ethernet Library* e *SD Library*. É uma placa compatível com o Arduino e possui um slot para cartão micro SD. Na Figura 17 podemos observar a placa Ethernet Shield.

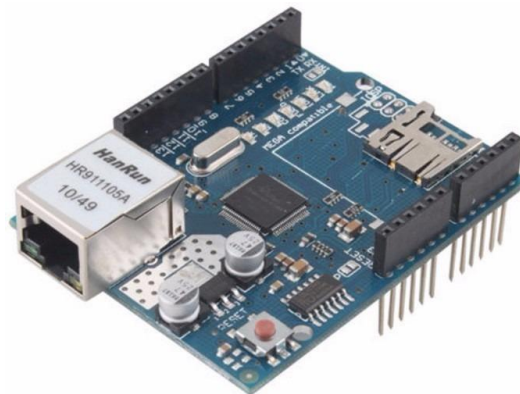


Figura 17 – Placa *Ethernet Shield* – Fonte [12]

Abaixo podemos observar as especificações técnicas da placa Ethernet:

- Suporta 4 conexões independentes simultaneamente, Memória Interna 16Kbytes para Tx / Rx Buffers, Suporte Interface Serial, Saídas função Multi, O W5100 Wiznet fornece uma rede (IP) capaz de TCP e UDP.

#### Diagrama de Conexão

A Figura 18 ilustra o diagrama de conexão das lâmpadas ao módulo relé e a *Ethernet Shield* conectada na parte superior da placa Arduino. O módulo relé

funciona basicamente como um interruptor que pode ser acionado pela placa principal. Para isso, utilizam-se 2 portas digitais para cada uma das lâmpadas, e quanto a alimentação do módulo relê é feita pelo pino 5V. O módulo relê está ligado no contato NO junto a lâmpada que está conectado ao terra da rede elétrica de 110V em sua outra extremidade. O contato comum do relê está ligado à rede elétrica de 110V.

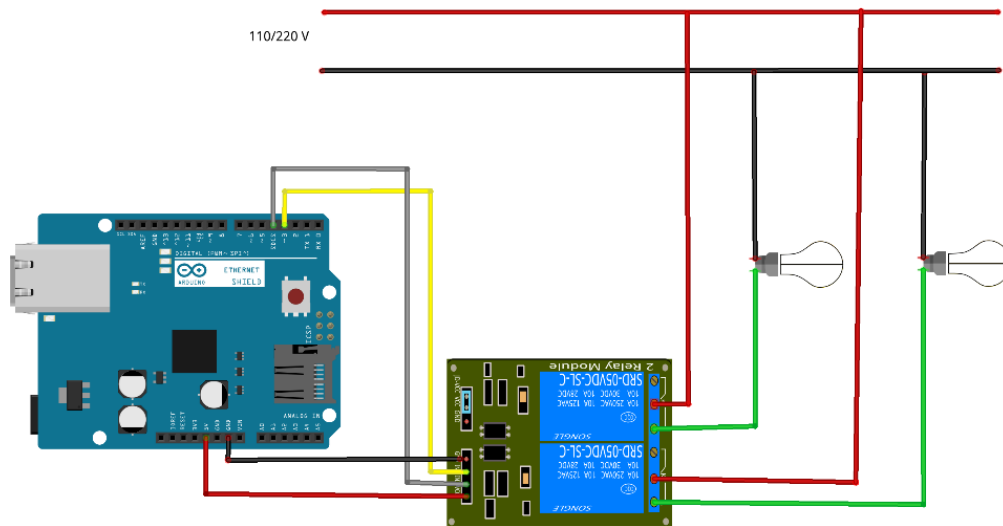


Figura 18 – Diagrama do acionamento de 2 lâmpadas via Internet – Fonte: Própria

### Código de Controle para o Arduino

Antes de rodar o código principal foi preciso descobrir o número do IP usado na placa *Ethernet Shield* e, para isso, foi utilizado um código auxiliar que retornou o IP utilizado pela placa, já que esse número é necessário para realizar o acesso a página de controle carregada pelo Arduino.

Foi utilizado como base um código do site FilipeFlop que posteriormente foi modificado. O código proposto faz o carregamento de uma página *web* usando as linguagens *html* e *css* e, utiliza também a linguagem *JavaScript* para fazer o controle das lâmpadas. Os testes realizados com este código permitiram acesso via navegador a uma página *web* semelhante à mostrada na Figura 19.



Figura 19 – Página web do controle das lâmpadas – Fonte: Própria

Ao fazer modificações na página com adaptações próprias em sua aparência, o Arduino acusou instabilidade durante a execução devido à falta de memória, o que impossibilitou o acesso a nova página web. Foram realizados novos testes utilizando um cartão SD, porém o problema persistiu e o acesso só passou a ser possível sem utilizar o CSS na página. As funções *JavaScript* continuaram funcionando, porém, toda a estética ficou inativa. Assim, embora tenha sido comprovada a viabilidade da aplicação, a página não possuía mais seu atrativo visual. O código produzido para a página foi o seguinte:

```
#include <SPI.h>
#include <Ethernet.h>

String readString;
int pino_rele1 = 3;
int pino_rele2 = 4;
boolean ligado = true;
boolean ligado_2 = true;

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
EthernetServer server(80);

void setup() {
  Serial.begin(9600);
  pinMode(pino_rele1, OUTPUT);
  pinMode(pino_rele2, OUTPUT);

  //Inicializa Ethernet Shield
  Ethernet.begin(mac);
  Serial.println("Ethernet Configured!");
  server.begin();
  Serial.print("Server Started.\nLocal IP: ");
  Serial.println(Ethernet.localIP());

  //Desliga os dois reles.n
  digitalWrite(pino_rele1, HIGH);
  digitalWrite(pino_rele2, HIGH);
}

void loop() {
  EthernetClient client = server.available();
  if (client) {
    while (client.connected()){
      if (client.available()) {
```

```

char c = client.read();
if (readString.length() < 100) {
    readString += c;
}
if (c == 'n') {
    Serial.println(readString);
    //Liga o Relé 1
    if (readString.indexOf("?ligar") > 0) {
        digitalWrite(pino_rele1, LOW);
        Serial.println("Relé 1 Ligado");
        ligado = false;
    }
    else {
        //Desliga o Relé 1
        if (readString.indexOf("?desligar") > 0) {
            digitalWrite(pino_rele1, HIGH);
            Serial.println("Relé 1 Desligado");
            Serial.println(readString.indexOf("?desligar"));
            ligado = true;
        }
    }
    Serial.println(readString);
    //Liga o Relé 2
    if (readString.indexOf("?2_ligar") > 0) {
        digitalWrite(pino_rele2, LOW);
        Serial.println("Relé 2 Ligado");
        Serial.println(readString.indexOf("?2_ligar"));
        ligado_2 = false;
    }
    else {
        //Desliga o Relé 2
        if (readString.indexOf("?2_desligar") > 0) {
            digitalWrite(pino_rele2, HIGH);
            Serial.println("Relé 2 Desligado");
            ligado_2 = true;
        }
    }
}
readString = "";
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println();
client.println("<html>");
client.println("<head>");
client.println("<title>Projeto Ethernet -UENF</title>");
client.println("<link rel='stylesheet' type='text/css' href='style.css' />");
client.println("<script type='text/javascript'");
src="https://img.filipeflop.com/files/download/automacao/automacao_residencial.js"></script>");
client.println("</head>");
client.println("<body>");
client.println("<header>");
client.println("<img id='uenf' src='https://i.imgur.com/TXAHHSw.png'>");
client.println("<img id='arduino' src='https://i.imgur.com/pyoPHnm.png'>");
client.println("</header>");
client.println("<section>");
client.println("<h1>Automação Residencial com Arduino</h1>");
client.println("<div id='funcionalidade'>");
client.println("<div id='div1'><h2>Relé 1</h2></div>");
client.println("<div id='botao'></div>");
client.print("<div id='rele'></div><div id='estado' style='visibility: hidden;'>");
client.print(ligado);
client.println("</div>");
client.println("<div id='div2'><h2>Relé 2</h2></div>");
client.println("<div id='botao_2'></div>");
client.print("<div id='rele_2'></div><div id='estado_2' style='visibility: hidden;'>");
client.print(ligado_2);
client.println("</div>");
client.println("<script>AlteraRelé1()</script>");
client.println("<script>AlteraRelé2()</script>");
client.println("</div>");

```

```

client.println("</section>");
client.println("<footer>");
client.println("<div id='alunas'>");
client.println("<h2>Projeto de Iniciação Científica desenvolvido pelas <br>alunas  
Isabela Correia e Gabriela Peixoto </h2>");
client.println("<h2>Orientador: Fermín Alfredo Tang Montané</h2>");
client.println("</div>");
client.println("<div id='curso'>");
client.println("<h2>Ciência da Computação</h2>");
client.println("<h2>LCMAT -CCT</h2> ");
client.println("</div>");
client.println("</footer>");*/
client.println("</body>");
client.println("</head>");
delay(1);
client.stop();
}
}
}
}
}

```

### Protótipo do dispositivo

A Figura 20 ilustra o protótipo do acionamento de duas lâmpadas via página web finalizado. A alimentação da placa Arduino é feita por meio de uma bateria de 9V e está conectado junto a placa *Ethernet Shield* e as lâmpadas são alimentadas mediante uma tomada conectada à rede elétrica.

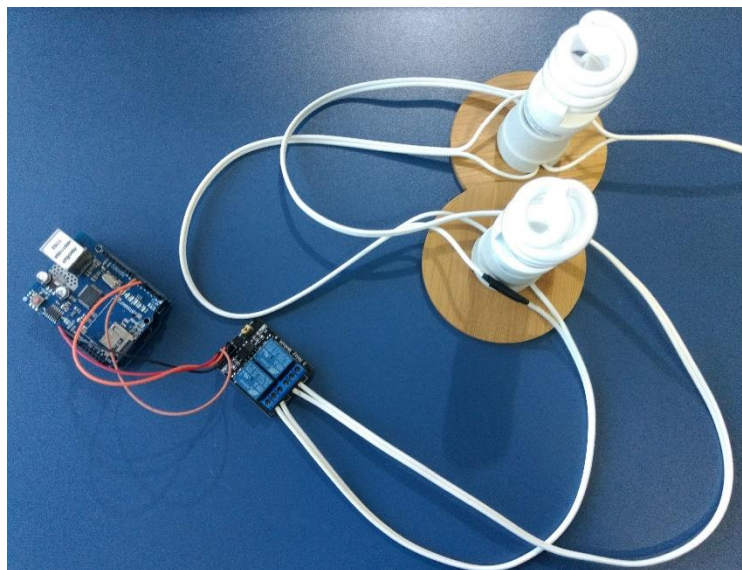


Figura 20 – Acionamento de lâmpadas via internet – Fonte: Própria

### 5.3. AUTOMAÇÃO RESIDENCIAL: MAQUETE DE UMA RESIDÊNCIA

Este protótipo consiste na aplicação do projeto principal e é baseado na construção de uma maquete de uma residência com a inclusão de quatro ambientes de uma casa com o intuito de observar de perto quais poderiam ser as maiores dificuldades que um portador de deficiência física enfrenta dentro de seu próprio lar.



Os projetos anteriores serviram como base para demonstrar em tamanho real como seria o funcionamento de cada dispositivo e para decidir quais se sairiam melhor no meio doméstico e que ao mesmo tempo tivesse uma maior aceitação e facilidade para o portador de deficiência física em sua residência.

A maquete da casa está sendo montada aos poucos e estudos estão sendo feitos para melhor adaptá-la para um ambiente automatizado. O gerenciamento dos dispositivos dentro da residência será feito a partir de um *software* que poderá ser acessado via internet pelo usuário que o retornará em forma de lista os ambientes cadastrados, sendo possível selecionar o ambiente a ser administrado e seus respectivos dispositivos.

#### Componentes utilizados

Foram utilizados os seguintes componentes:

- 2 Caixas simples com tampa em MDF
- 1 Arduino Uno com cabo USB
- 1 Protoboard e Jumpers variados
- 1 Módulo Relé Arduino de 2 canais
- 2 Módulos sensor de movimento PIR
- 2 Lâmpadas pequenas
- 2 Bocais
- 1 Plugue de tomada
- Cabo elétrico flexível

#### Protótipo do dispositivo

A Figura 21 ilustra o protótipo da maquete e ela já consta com o acionamento de 2 lâmpadas por meio de um sensor de presença com o intuito de iniciar os testes. O funcionamento é o mesmo do primeiro protótipo das lâmpadas, porém, com a montagem diferente por se tratar de dois dispositivos que são ativados de forma individual, cada um pelo seu próprio sensor, além disso, conta com uma observação em escala em um ambiente já montado. A alimentação da placa Arduino é feita por meio de uma bateria de 9V e as lâmpadas são alimentadas através da rede elétrica.

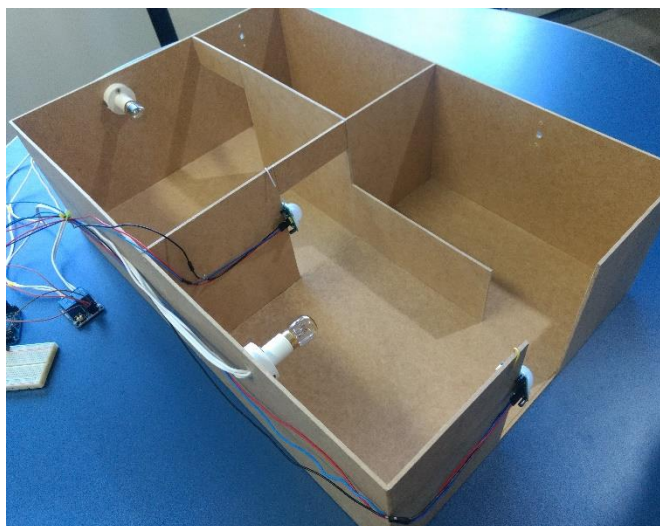


Figura 21 – Maquete de uma residência com sensores de presença – Fonte: Própria



## 6. CONCLUSÃO

Pela observação dos aspectos analisados no plano de trabalho, a primeira etapa consistiu em reunir e estudar materiais relacionados aos diferentes tipos de deficiência e suas principais necessidades, além da plataforma Arduino. O estudo das deficiências concluiu o tipo de deficiência que seria abordada no projeto, sendo escolhida a física. O treinamento com a plataforma Arduino se deu a partir de pequenos projetos disponíveis em livros e sites disponibilizados pelo coordenador. Por fim, o jogo Genius foi utilizado como forma de colocar em prática essa primeira etapa do estudo da plataforma Arduino.

Antes do início do projeto principal foram realizados alguns protótipos para escolher o que melhor se enquadraria na maquete residencial a partir dos estudos em relação a quais seriam as maiores dificuldades para um deficiente dentro de suas casas. A maquete conta atualmente com dois sensores que são responsáveis pelo acionamento de duas lâmpadas elétricas em escala pequena em dois cômodos distintos. A expectativa é que essa maquete seja integrada com um sistema de acionamento remoto para que o portador de deficiência física possa ligar algum dispositivo elétrico/doméstico dentro de qualquer cômodo de seu lar sem que haja dificuldade para ele.

As etapas do plano de trabalho referentes à construção dos protótipos e da maquete, foram em sua grande maioria cumpridas de maneira satisfatória. O projeto da estrutura física demandou tempo para a sua construção e escolha do material. Foram necessárias duas caixas em MDF que foram desmontadas, serradas e adaptadas de forma a se construir cômodos. Além disso, foram feitos furos de forma a se passar as lâmpadas e toda a fiação de energia.

Por fim, apesar do aplicativo de controle não ter sido efetuado diretamente na maquete, esse controle foi implementado no projeto de controle de lâmpadas através de uma página *web*. Na continuação da pesquisa será necessário um estudo mais a fundo sobre redes, servidores *web* e linguagens de programação adicionais para a implementação de um software de controle remoto.

## 7. REFERÊNCIAS

- [1] EVANS, Martin; NOBLE, Joshua; HOCHENBAUM, Jordan. Arduino em Ação. 1ª. ed. São Paulo, SP: Novatec, 2013. 424 p.
- [2] THOMSEN, Adilson. O que é Arduino?. 2014. Disponível em: <<https://www.filipeflop.com/blog/o-que-e-arduino/>>. Acesso em: 23 fev. 2018.
- [3] PROTOBOARD. Disponível em: <<http://www.eletronicadidatica.com.br/protoboard.html>>. Acesso em: 24 fev. 2018.
- [4] MORENO, Laila Canut. Laia: tecnologia. 2017. Disponível em: <<http://laiacmtecnologiaseda.blogspot.com.br/2017/03/>>. Acesso em: 24 fev. 2018.
- [5] DIGILENT. A National Instruments Company. Debouncing via Software: Correcting Bounce with Software. Disponível em: <<https://learn.digilentinc.com/Module/102?position=0>>. Acesso em: 25 fev. 2018.
- [6] CÓDIGO de cores de resistores. Disponível em: <<https://www.mundodaeletrica.com.br/codigo-de-cores-de-resistores/>>. Acesso em: 09 maio 2018.
- [7] OLIVEIRA, Euler. Arduino – Utilizando o Push Button. Disponível em: <<http://blogmasterwalkershop.com.br/arduino/arduino-utilizando-o-push-button/>>. Acesso em: 25 fev. 2018.
- [8] EMMANUEL, Fábio. Jumper. 2012. Disponível em: <<http://www.roboliv.re/conteudo/jumper>>. Acesso em: 26 fev. 2018.
- [9] GUEDES, Mariana. Como usar o Sensor Piezo no Arduino. 2017. Disponível em: <<https://www.arduinoportugal.pt/usando-buzzer-arduino-sensor-piezo-eletrico/>>. Acesso em: 17 abr. 2018.
- [10] ARDUINO E ELETRÔNICA. (João Pessoa). Módulo Relé de 1 Canal. Disponível em: <<https://www.arduinoeeletronica.com.br/?product=modulo-rele-de-1-canal>>. Acesso em: 26 fev. 2018.
- [11] THOMSEN, Adilson. Acendendo uma lâmpada com sensor de presença. 2013. Disponível em: <<https://www.filipeflop.com/blog/acendendo-uma-lampada-com-sensor-de-presenca/>>. Acesso em: 26 fev. 2018.
- [12] VIDAL, Vitor. Ethernet Shield W5100 com Arduino. Disponível em: <<http://blog.eletrogate.com/ethernet-shield-w5100-com-arduino/>>. Acesso em: 21 jul. 2018.

## **8. PERSPECTIVAS DE CONTINUIDADE OU DESDOBRAMENTO DO TRABALHO**

Na continuação da pesquisa, conforme descrito no plano de trabalho proposto na solicitação de renovação para o segundo ano, propõe-se dar continuidade ao projeto de automação residencial assistiva elaborado durante o primeiro ano, mediante o desenvolvimento de um software de controle dos dispositivos conectado através da internet. Os primeiros testes de conectividade com a internet realizados no primeiro mostraram a viabilidade da proposta, mas revelaram alguns problemas com as tecnologias utilizadas. Com isso é necessário maior estudo sobre redes em geral e formas de conexão com a internet através de dispositivos micro-controlados. Foram adquiridas novas componentes de para conexão com a internet: placas Esp8266-01, e NodeMCU. Assim, o plano de trabalho para o segundo ano propõe dedicar um tempo ao estudo e familiarização com estas componentes e também ao estudo de linguagens de programação necessárias para o desenvolvimento da aplicação de controle que poderá ser um website ou através de dispositivo móvel.

## **9. PARTICIPAÇÃO EM CONGRESSOS E TRABALHOS PUBLICADOS OU SUBMETIDOS E OUTRAS ATIVIDADES ACADÊMICAS E DE PESQUISA**

Os protótipos desenvolvidos nesta pesquisa, o Jogo Genius Estendido e o dispositivos de controle de lâmpadas e ventilador controlados mediante sensor de presença, foram apresentados durante a VII Semana de Computação e Tecnologia da Informação da UENF, que aconteceu de 06 a 11 de Novembro de 2017, como parte de um minicurso ministrado pela autora desta pesquisa. O Certificado deste evento encontra-se no **Anexo I**.

## **10. DATA E ASSINATURA DO BOLSISTA**

30 de Agosto de 2018,



## **11. DATA E ASSINATURA DO ORIENTADOR**

30 de Agosto de 2018,



## CERTIFICADO

Certificamos que Gabriela Peixoto de Souza ministrou o Minicurso: **Arduino: do zero ao jogo da Semana Acadêmica de Computação e Tecnologia da Informação**, realizada na Universidade Estadual do Norte Fluminense Darcy Ribeiro – UENF, no período de 06 a 11 de novembro de 2017, com carga horária de 04 horas.

Campos dos Goytacazes, 11 de novembro de 2017.  
Registrado sob o nº 21312 no livro 01 folha 80.



Isabela Correia Pereira  
Presidente da Comissão Organizadora



Marina Satika Suzuki  
Pró-Reitora de Graduação