



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO  
DEPARTAMENTO DE CIÊNCIAS EXATAS E NATURAIS  
CURSO DE CIÊNCIA DA COMPUTAÇÃO

ALEXANDRO LIMA DAMASCENO

**DESENVOLVIMENTO DE UM SISTEMA DE AUTOMAÇÃO PARA  
ACIONAMENTO DE EQUIPAMENTOS ELETRÔNICOS E POSICIONAMENTO DE  
CÂMERAS DE SEGURANÇA**

MOSSORÓ-RN  
2014

ALEXANDRO LIMA DAMASCENO

**DESENVOLVIMENTO DE UM SISTEMA DE AUTOMAÇÃO PARA  
ACIONAMENTO DE EQUIPAMENTOS ELETRÔNICOS E POSICIONAMENTO DE  
CÂMERAS DE SEGURANÇA**

Monografia apresentada à Universidade Federal Rural do Semi-Árido – UFERSA, Departamento de Ciências Exatas e Naturais para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Msc. Marcelo Roberto Bastos Guerra Vale –UFERSA.

MOSSORÓ-RN

2014

**O Conteúdo desta obra é de inteira responsabilidade de seus autores**

**Dados Internacionais de Catalogação na Publicação (CIP)**

**Biblioteca Central Orlando Teixeira (BCOT)**

**Setor de Informação e Referência**

D155d Damasceno, Alexandro Lima.

Desenvolvimento de um sistema de automação para acionamento de equipamentos eletrônicos e posicionamento de câmeras de segurança./ Alexandro Lima Damasceno -- Mossoró, 2014.

47f.: il.

Orientador: Prof. Msc. Marcelo Roberto Bastos Guerra Vale

Monografia (Graduação em Ciência da Computação) – Universidade Federal Rural do Semi-Árido. Pró-Reitoria de Graduação.

1. Automação. 2. Arduino. 3. Domótica. I. Título.

RN/UFERSA/BCOT /031-14

CDD: 629.8

Bibliotecária: Vanessa Christiane Alves de Souza Borba

ALEXANDRO LIMA DAMASCENO

**DESENVOLVIMENTO DE UM SISTEMA DE AUTOMAÇÃO PARA  
ACIONAMENTO DE EQUIPAMENTOS ELETRÔNICOS E  
POSICIONAMENTO DE CÂMERAS DE SEGURANÇA**

Monografia apresentada à Universidade  
Federal Rural do Semi-Árido –  
UFERSA, Departamento de Ciências  
Exatas e Naturais para a obtenção do  
título de Bacharel em Ciência da  
Computação.

APROVADO EM: 26/02/2014

BANCA EXAMINADORA

\_\_\_\_\_  
Prof. Mcs. ~~Marcelo~~ Roberto Bastos Guerra Vale  
Presidente – Orientador

✓ \_\_\_\_\_  
Prof. Mcs. Daniel Faustino Lacerda de Souza  
Primeiro Membro

\_\_\_\_\_  
Prof. D.Sc. Angélica Félix de Castro  
Segundo Membro

\_\_\_\_\_  
Prof. D.Sc. Leonardo Augusto Casillo  
Terceiro Membro

## **AGRADECIMENTOS**

A todos os professores de Ciências da Computação pela contribuição acadêmica. Em especial ao meu orientador, Marcelo Guerra, pela orientação, paciência, sugestões e críticas construtivas.

Aos meus pais, Aurissandro e Irani, por sempre acreditarem em mim e pelo apoio e incentivos dados.

Aos meus familiares e amigos que se alegram a cada conquista minha e torcem pelo meu próximo passo desta caminhada.

Aos meus amigos, em especial a Kleber Lopes e Lucas Vinicius que me acompanharam durante todo o curso e com quem sempre pude contar.

Aos amigos Dênis Freire e Alexandre Barros que, muito me ajudaram, tornando possível a implementação desse trabalho.

E por último, mas não menos importante, à Larissa Barbosa por todo o amor, carinho, atenção e compreensão dedicados nesse momento tão impar.

## RESUMO

Esse trabalho consiste no desenvolvimento de um projeto de automação com acionamentos de lâmpadas e posicionamento de câmeras de segurança. A necessidade deste trabalho surgiu do projeto de pesquisa "UFERSA virtual" (projeto que mapeia a UFERSA em uma maquete virtual 3D). A idéia principal é representar em uma maquete física todas as ações (acionamentos de lâmpadas, equipamento e posicionamento de câmeras) que serão reproduzidas na maquete virtual 3D. Para isso, é necessário a interação dessa maquete virtual com a maquete real, onde ao acendermos luzes na virtual, uma lâmpada 220V correspondente seja ativada na maquete real. A interação é feita utilizando a plataforma de prototipagem ARDUINO com uma *SHIELD ETHERNET* que possibilitará a conexão com a Internet. Dessa forma qualquer periférico (computador, notebook, smartphone etc) com acesso à rede poderá acionar as lâmpadas da maquete virtual e real. O estado de cada lâmpada da maquete real será monitorado na maquete virtual em tempo real. Para tornar possível o acionamento de lâmpadas convencionais (Ligar/Desligar) a partir de um interruptor físico ou virtual, foi necessário a implementação de um circuito utilizando acionamentos de relés. Para isso foi necessário utilizar um interruptor em paralelo com o relé, o que são comumente conhecidos de interruptores paralelos (*Three Way*). O acionamento do relé é controlado a partir de uma plataforma de prototipagem ARDUINO MEGA 2560 com um *Shield* de extensão *Ethernet*. O conjunto (Arduino + *Shield Ethernet*) permitirá que o arduino possa se conectar à uma rede doméstica ou até mesmo à Internet, dessa forma, conseguimos programar o ARDUINO para operar como um pequeno servidor HTTP. Esse servidor recebe requisições HTTP do ambiente virtual 3D/2D e com base nelas permite o acionamento de equipamentos conectados ao circuito. Informações sobre o estado (ativo/inativo) de algum equipamento ligado ao circuito também podem ser enviados aos clientes conectados ao servidor. Para retornar informações do estado das lâmpadas (ligada/desligada) do circuito foi utilizado um sensor de luminosidade LDR para cada lâmpada, onde de acordo com a variação de luminosidade o arduino irá definir variáveis internas do servidor como "*on*" ou "*off*". Com essas informações o ambiente virtual 3D/2D pode exibir lâmpadas simulando o estado da lâmpada real.

**Palavras-chave:** Automação, Arduino, Domótica.

## **ABSTRACT**

This work presents the development of an automation project for UFERSA Virtual, a virtual representation of Universidade Federal Rural do Semi-Árido (UFERSA). The main idea is to map and control the real representation of lamps, cameras and other structures of University through of a 2D/3D virtual reality model. The interaction real/virtual is provided by platform ARDUINO integrated with a ETHERNET SHIELD to provide communication. The status of each lamp model is monitored virtually in real time. To make possible the activation of conventional lamps (on / off) from a physical or virtual switch, it was necessary to implement a circuit using relays drives. This required using a switch in parallel with the relay, which are commonly known parallel switches (Three Way). Operation of the relay is controlled from the platform ARDUINO MEGA 2560. The group (Arduino+Ethernet Shield) will allow the arduino to connect to a home network or even the Internet, thus we could program the ARDUINO to operate as a small HTTP server. This server receives HTTP requests from 2D/3D virtual environment based to reach information about the status (active / inactive) of any equipment connected to the circuit. To return the state information of lamp a light sensor LDR was connect to each lamp. According to brightness level the Arduino will set internal server status as "on" or "off". With this information the virtual environment can safely display lamps status, simulating the real state of the lamp.

Keywords: Automation, Arduino, Domotic.

## LISTA DE FIGURAS

|                                                     |    |
|-----------------------------------------------------|----|
| Figura 1. Modelos de arduinos.....                  | 16 |
| Figura 2. IDE de programação .....                  | 18 |
| Figura 3. Shield relé.....                          | 19 |
| Figura 4. Shield ethernet.....                      | 20 |
| Figura 5. Servo motor.....                          | 21 |
| Figura 6. Esquema do circuito .....                 | 23 |
| Figura 7. Câmera e servo motor.....                 | 24 |
| Figura 8. Maquete física .....                      | 25 |
| Figura 9. Esquema three way .....                   | 26 |
| Figura 10. Esquema LED .....                        | 27 |
| Figura 11. Interface do ambiente virtual .....      | 29 |
| Figura 12. Comunicação entre o arduino e o AV ..... | 30 |
| Figura 13. Resultados dos primeiros testes .....    | 36 |
| Figura 14. Interface do active webcam .....         | 39 |
| Figura 16. Resultado final AV 3D.....               | 40 |
| Figura 15. Resultado final AV 2D.....               | 40 |

## **LISTA DE TABELAS**

|                                                                    |    |
|--------------------------------------------------------------------|----|
| Tabela 1. COMPARATIVO ENTRE OS MODELOS NANO, UNO E MEGA 2560 ..... | 17 |
| Tabela 2. ESPECIFICAÇÃO SHIELD ETHERNET .....                      | 20 |
| Tabela 3. TABELA DE CUSTOS.....                                    | 41 |

## LISTA DE QUADROS

|                                                                          |    |
|--------------------------------------------------------------------------|----|
| Quadro 1. Código da estrutura object para câmera.....                    | 28 |
| Quadro 2. Função de acionamento do LED implementada no AV.....           | 31 |
| Quadro 3. Função no arduino.....                                         | 32 |
| Quadro 4. Código de controle do servo motor implementado no arduino..... | 33 |
| Quadro 5. Leitura dos sensores implementada no arduino.....              | 34 |
| Quadro 6. AV verificando estado dos LEDs.....                            | 35 |

## **LISTA DE ABREVIATURAS E SIGLAS**

AV– Ambiente Virtual

AF – Ambiente Físico

LDR – Light Dependent Resistor (Resistor Dependente de Luz)

LED – Light-emitting Diode (Diodo emissor de Luz)

## SUMÁRIO

|                                                            |           |
|------------------------------------------------------------|-----------|
| <b>1 INTRODUÇÃO.....</b>                                   | <b>11</b> |
| <b>2 AUTOMAÇÃO .....</b>                                   | <b>13</b> |
| 2.1 AUTOMAÇÃO INDUSTRIAL .....                             | 14        |
| 2.2 AUTOMAÇÃO RESIDENCIAL.....                             | 15        |
| 2.3 ARDUINO .....                                          | 16        |
| <b>2.3.1 Software de programação do arduino .....</b>      | <b>18</b> |
| <b>2.3.2 Shields e sensores .....</b>                      | <b>19</b> |
| <b>2.3.2.1 Shield relé.....</b>                            | <b>19</b> |
| <b>2.3.2.2 Shield ethernet .....</b>                       | <b>20</b> |
| <b>2.3.2.3 Servo motor .....</b>                           | <b>21</b> |
| <b>3 DESENVOLVIMENTO .....</b>                             | <b>22</b> |
| 3.1 O AMBIENTE FÍSICO.....                                 | 22        |
| <b>3.1.1 A maquete física e a utilização de LEDs .....</b> | <b>24</b> |
| <b>3.1.1.1 Circuito equivalente da maquete .....</b>       | <b>25</b> |
| 3.2 O AMBIENTE VIRTUAL .....                               | 27        |
| 3.3 A COMUNICAÇÃO ENTRE OS SERVIDORES.....                 | 29        |
| <b>4 TESTES E RESULTADOS .....</b>                         | <b>36</b> |
| <b>5 CONCLUSÕES .....</b>                                  | <b>42</b> |
| <b>REFERÊNCIAS.....</b>                                    | <b>44</b> |

## 1 INTRODUÇÃO

Automatizar um sistema ou projeto consiste em torná-lo autônomo, ou seja, operar sem a atuação humana. Essa definição surgiu no passado com a ascensão do setor industrial e a busca pelo aumento da produção e o barateamento da mão-de-obra utilizada. Nos dias atuais esse conceito de automação vem se redefinindo. Anteriormente a automação unificou linhas de produção padronizando setores, hoje ela flexibiliza essa produção permitindo a criação de perfis diferentes para diferentes usuários.

O conceito de automação surgiu na indústria com o objetivo de substituir a mão-de-obra humana por máquinas e sistemas de controle. O papel dessas máquinas e sistemas era supervisionar e otimizar o controle dos processos, visando aumentar a produtividade e a qualidade dos produtos.(MONTEBELLER, 2011).

Com a automação predial surgiu a idéia de edifício inteligente. De acordo com o *Intelligent Buildings Institute* (IBI), um edifício inteligente é aquele que oferece um ambiente produtivo e que é economicamente racional, através de otimização dos seus quatro elementos básicos – estrutura, sistemas, serviços e gestão – e das relações entre eles. Os edifícios inteligentes ajudam seus proprietários, gestores e ocupantes a atingir seus objetivos sob as perspectivas de custo, conforto, adequação, segurança, flexibilidade e valor comercial. (MARTE, 1995).

O objetivo desse trabalho é desenvolver um sistema de automação predial/residencial utilizando tecnologias atuais e de fácil acesso, buscando implementar nesses sistemas funcionalidades básicas, mas de alto impacto na sua utilização. Para desenvolver a automação predial utilizou-se a plataforma ARDUINO. Nos dias atuais o ARDUINO vem se difundindo muito entre os desenvolvedores, tudo isso graças às inúmeras utilidades que a placa de prototipagem disponibiliza. Fatores que são facilmente levados em consideração e que implicam diretamente nessa preferência de seu uso são os seguintes: baixo custo, software livre, linguagem simples e software para várias plataformas. Além dessas vantagens, ainda existe uma comunidade ativa de usuários, fazendo com que uma grande quantidade de pessoas compartilhem dúvidas e explicações.

Outra tendência nos dias atuais é a construção e utilização para vários fins dos chamados Ambientes Virtuais (AVs). Esses ambientes são desenvolvidos geralmente para

serem implantados na WEB e muito utilizados no meio educacional. AVs podem ser tridimensionais (3D) ou bidimensionais (2D), dependendo geralmente da necessidade de realidade que se deseja transparecer ao usuário. Ultimamente estão sendo inseridos no mercado da automação industrial e predial, como ferramentas de monitoramento dos ambientes reais aos quais estão vinculados. Dessa forma, através de acesso remoto é possível que um administrador e/ou usuário tenha um retorno do que acontece no ambiente real.

Criar um sistema de automação com acesso remoto eficaz vem sendo um desafio cada vez mais requisitado. Diariamente desenvolvedores se empenham em criar ou recriar seus próprios sistemas adequados a cada tipo de necessidade. Na literatura é possível encontrar trabalhos que tem foco na automação predial ou industrial, como por exemplo DIAS (2004) que em seu trabalho discute sobre os benefícios e a aplicabilidade da automação no ambiente residencial levando em consideração a necessidade atual das pessoas em obter conforto e praticidade. NEVES (2007) que em seu trabalho, faz um apanhado geral sobre os principais problemas enfrentados pela automação em suas diferentes áreas de atuação.

Para atingir o objetivo do trabalho foi desenvolvido um ambiente virtual 2D para monitorar e controlar um ambiente real automatizado com o auxílio do ARDUINO. Nesse ambiente é possível controlar equipamentos eletrônicos e posicionar câmeras de segurança tendo retorno em tempo real do estado de cada lâmpada ou equipamento.

Esse trabalho está organizado da seguinte forma: O capítulo 2 descreve as ferramentas utilizadas para o desenvolvimento da automação, como por exemplo a plataforma de prototipagem ARDUINO e suas Shields de expansão. Em seguida, no capítulo 3, será apresentada uma breve descrição de todas as etapas do desenvolvimento do sistema. No capítulo 4 os resultados obtidos serão mostrados, e, por fim, o capítulo 5 irá abordar as conclusões oriundas do trabalho.

## 2 AUTOMAÇÃO

No contexto atual, a automação não define apenas um sistema autônomo, mas também um sistema que possibilita a facilidade do controle humano sobre determinadas tecnologias. Foi a partir desse novo raciocínio que se passou a utilizar a automação também em ambientes residenciais, fundamentando-se assim o termo domótica que se refere à integração de tecnologias diversas no ambiente residencial, promovendo comodidade, economia e segurança.

Na literatura podemos encontrar outras definições para o termo domótica e seu surgimento, como (DIAS, 2004) que afirma que fatores como expectativa de vida, redução do número de filhos, aumento do número de famílias não convencionais e o aumento da violência favorecem o isolamento das pessoas em suas residências. Esses fatores refletem diretamente nas novas necessidades das moradias das famílias, destacando as aplicações da domótica como forma de suprir tais necessidades.

Atualmente a automação se destaca com a utilização de tecnologias para aumentar o conforto e a segurança dos usuários, tornando-se uma espécie de fronteira do passado com o futuro. Pelo grande impacto que a ela vem dando no mercado, conseguimos encontrar alguns trabalhos que desenvolveram algum tipo de automação, seja residencial ou industrial. Como por exemplo: (ALIANÇA NETO, 2011) que em seu trabalho, Projeto de Um Sistema de Automação Residencial, desenvolve um sistema de automação de baixo custo, controlado por um computador pessoal e utilizando como tecnologia base a plataforma ZigBee. Ou ainda (LEE, 2007) que em seu trabalho, Monitoração Remota Via IP de Sensores Industriais, explora a ascensão dos sensores nas indústrias e desenvolve um hardware com software embarcado para aquisição e transmissão via IP de dados de sensores analógicos.

A exigência do mercado por tecnologia fez crescer o uso de redes com protocolos consagrados e a automação caminha para a integração dos sistemas. Desta forma, prédios com sensores isoladamente automatizados passaram a ser integrados em um sistema onde uma central de controle pode monitorar, controlar e gerenciar todos os sensores e equipamentos.(MONTEBELLER, 2011).

Com a tecnologia wireless e a utilização da Internet, é possível controlar aparelhos eletrônicos, lâmpadas, portões, entre outros, de qualquer lugar com acesso à Rede. Dessa forma, verificar se portas e portões estão abertos, lâmpadas estão acesas, ou até mesmo

garantir a segurança de um ambiente em tempo real com a utilização de um circuito fechado de câmeras pode ser executado há distância.

A automação possui dois ramos principais, os quais são abordados nas próximas seções.

## 2.1 AUTOMAÇÃO INDUSTRIAL

Nos últimos anos, houve uma grande transformação no setor industrial. Segundo ROSARIO (2009) foi necessária a modernização dos parques industriais, visando à competitividade dos produtos, por meio do aumento da qualidade, e redução de custos, possibilitando preços mais acessíveis, dando origem a um conjunto de técnicas e procedimentos designados AUTOMAÇÃO INDUSTRIAL.

As primeiras formas de automação tinham propósitos bem técnicos, como desenvolver equipamentos de controle e de medição elétrica e pneumática, aplicados basicamente na indústria.

De acordo com ROSARIO (2009), a automação industrial pode ser dividida em três classes: a rígida, a flexível e a programável.

- Rígida: é utilizado quando o volume de produção é elevado; nesta forma, a linha de produção é composta de diversas máquinas de Comando Numérico, chamadas estações de trabalho. Nessas estações é realizado um conjunto de operações e, à medida que são terminadas, as peças são transferidas para outras estações. Dessa forma a produção possui uma linha de produção fixa, voltada apenas para a concepção de um determinado tipo de produto.
- Flexível: é utilizado para um volume de produção média, decorrente da maior interação da área de engenharia mecânica com tecnologias eletrônicas embarcadas e sistemas de informação. Nesta forma, a automação, aliada à flexibilidade, possibilita que sejam fabricados diversos produtos ao mesmo tempo, utilizando-se o mesmo sistema de produção.
- Programável: é utilizada para um volume de produção relativamente baixo e diversificado, ou seja, a produção é efetuada em pequenos lotes. Nesta forma de produção os equipamentos devem ser reprogramados a cada novo lote.

## 2.2 AUTOMAÇÃO RESIDENCIAL

A automação residencial nada mais é que a adequação de toda a tecnologia existente nos ambientes industriais às necessidades dos ambientes residenciais, para garantir maior conforto e segurança, além de outros fatores de igual relevância.

Segundo a *Domotics Integration Project - DIP* Domótica ou tecnologia da casa inteligente é a integração de serviços e tecnologias, aplicados a residências, flats, apartamentos, casas e pequenas construções, com o propósito de automatizá-los e obter um aumento em relação a segurança e proteção, conforto, comunicação e gerenciamento técnico.(DIAS; PIZZOLATO, 2004).

A automação difundiu-se nos diversos ramos industriais, chegando até as edificações corporativas. Segundo DIAS e PIZZOLATO (2004) a cada dia, novos componentes que agregam tecnologias relacionadas à automação são aplicados às instalações prediais gerando benefícios como: gerenciamento técnico, conforto, economia, prevenção de acidentes, falhas de equipamentos, e também segurança aos usuários.

Atualmente no mercado existe uma série de equipamentos utilizados para o desenvolvimento e a instalação da automação residencial, esses equipamentos são fechados em pacotes (kits). Porém todos esses kits são de uso restrito de seus fabricantes e distribuidores, além de ter um custo de aquisição e instalação muito elevado. Por outro lado esses kits têm sempre algo em comum, todos eles possuem sensores (presença, luminosidade, etc.), relés de controle, uma central de monitoramento e um processador. Geralmente esses equipamentos que formam os kits são desenvolvidos pelos próprios distribuidores e em alguns casos existem módulos de processamento individualmente disponibilizados no mercado.

Diante da disponibilidade de equipamentos como sensores e relés, que podem ser facilmente adquiridos no mercado e da possibilidade de utilizar uma nova tecnologia que está se difundindo entre os jovens desenvolvedores, o Arduino, pode-se construir um sistema de automação residencial utilizando como central de processamento a plataforma Arduino, suas *Shields* e sensores.

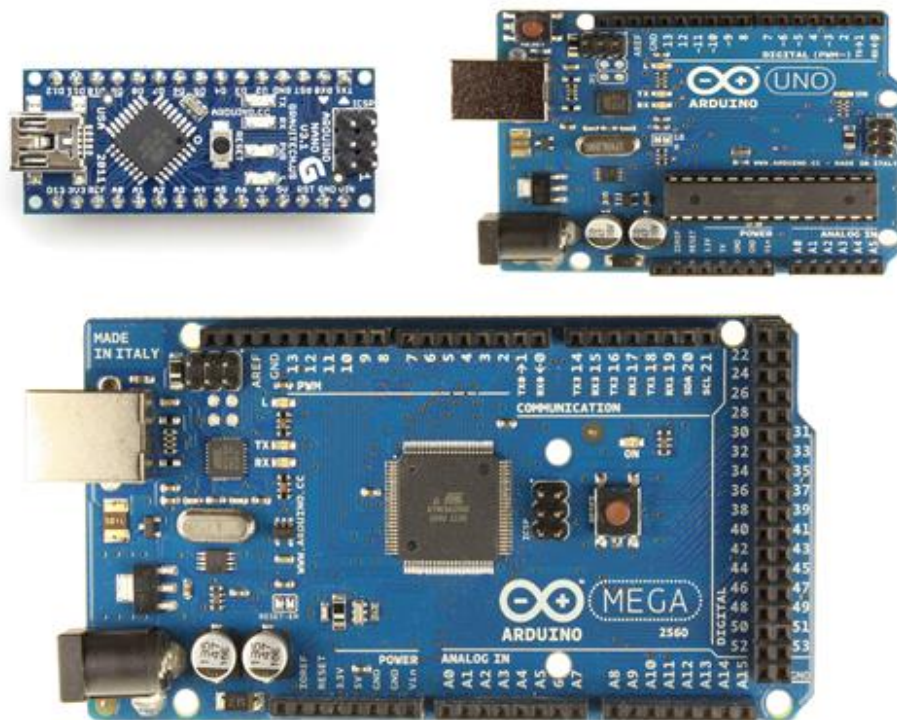
Na subseção 2.3 é apresentada uma discussão teórica sobre a plataforma de prototipagem Arduino e suas *shields*.

### 2.3 ARDUINO

Arduino é uma placa de controle de entrada de dados (IN), como sensores, e saída de dados (OUT), como motores e leds, com cristal oscilador de 16 Mhz, um regulador de tensão de 5 V, botão de reset, plugue de alimentação, pinos conectores, e alguns LEDs para facilitar a verificação do funcionamento. A porta USB já fornece alimentação enquanto estiver conectado ao computador, e a tensão de alimentação quando desconectado pode variar de 7 V a 12 V, graças ao regulador presente na placa. No Arduino, informações ou ordens são transmitidas de um computador para a placa através de Bluetooth, wireless, USB, infravermelho, etc. Essas informações devem ser traduzidas utilizando a linguagem Wiring baseada em C/C++.(DESTACOM – GRUPO DE ROBÓTICA UFMS, 2012).

Existem vários tipos de Arduinos, dentre eles destacam-se os modelos apresentados na Figura 1.

**Figura 1. Modelos de arduinos**



**FONTE: AUTORIA PRÓPRIA**

Na Figura 1, são apresentados três modelos de Arduino: Nano, Uno e Mega 2560. Na Tabela 1 é exibido um comparativo entre os três modelos.

Tabela 1. COMPARATIVO ENTRE OS MODELOS NANO, UNO E MEGA 2560

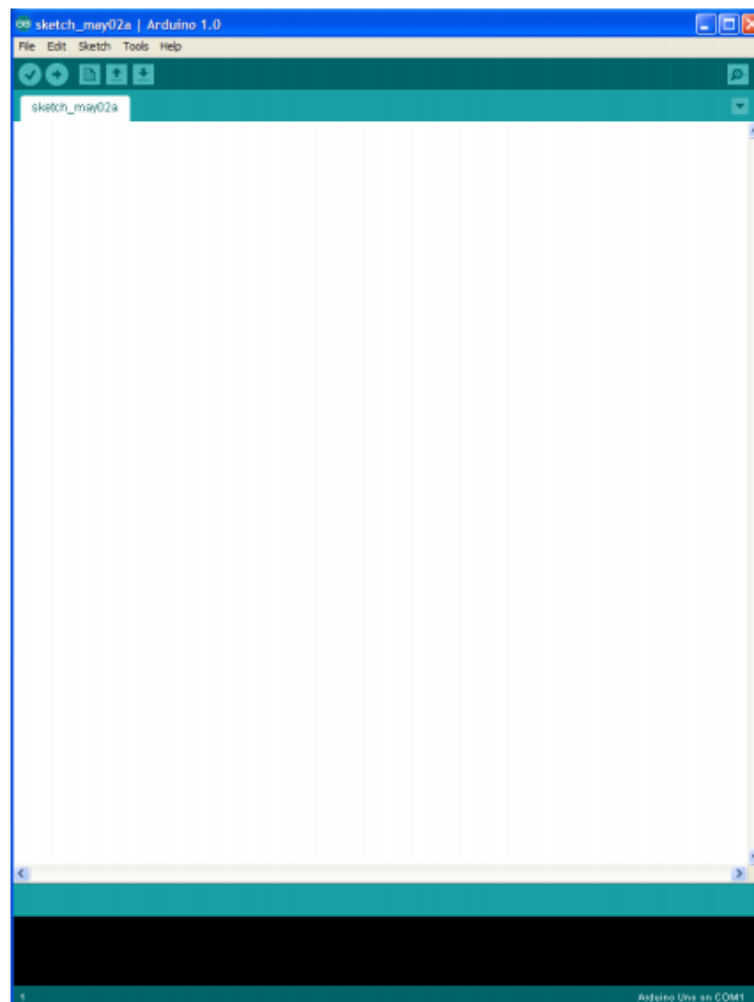
|                                                | <b>NANO</b>                          | <b>UNO</b>                           | <b>MEGA 2560</b>                      |
|------------------------------------------------|--------------------------------------|--------------------------------------|---------------------------------------|
| <b>MICROCONTROLADOR</b>                        | Atmel ATmega168 ou ATmega328         | ATmega328                            | ATmega2560                            |
| <b>TENSÃO DE OPERAÇÃO</b>                      | 5V                                   | 5V                                   | 5V                                    |
| <b>TENSÃO DE ENTRADA (recomendada)</b>         | 7-12V                                | 7-12V                                | 7-12V                                 |
| <b>TENSÃO DE ENTRADA (limite)</b>              | 6-20V                                | 6-20V                                | 6-20V                                 |
| <b>PINOS DE ENTRADA E SAÍDA (I/O) DIGITAIS</b> | 14 (dos quais 6 podem ser saída PWM) | 14 (dos quais 6 podem ser saída PWM) | 54 (dos quais 15 podem ser saída PWM) |
| <b>PINOS DE ENTRADAS ANALÓGICAS</b>            | 8                                    | 6                                    | 16                                    |
| <b>CORRENTE DC POR PINO I/O</b>                | 40mA                                 | 40mA                                 | 40mA                                  |
| <b>CORRENTE DC PARA PINO DE 3,3V</b>           | 50mA                                 | 50mA                                 | 50mA                                  |
| <b>MEMÓRIA FLASH</b>                           | 16KB                                 | 32KB                                 | 256KB                                 |
| <b>SRAM</b>                                    | 1KB                                  | 2KB                                  | 8KB                                   |
| <b>EEPROM</b>                                  | 512MB                                | 1KB                                  | 4KB                                   |
| <b>VELOCIDADE DE CLOCK</b>                     | 16MHz                                | 16MHz                                | 16MHz                                 |

Diante do comparativo realizado, optou-se por utilizar o modelo Mega 2560 pela maior quantidade de pinos digitais e analógicos possibilitando assim o controle de um maior número de equipamentos.

### 2.3.1 Software de programação do arduino

O Arduino possui um software de programação chamado **Arduino IDE** que é uma aplicação multiplataforma escrita em Java derivada dos projetos Processing e Wiring. É esquematizado para introduzir a programação a pessoas não familiarizadas com o desenvolvimento de software. Inclui um editor de código com recursos de realce de sintaxe, parênteses correspondentes e indentação automática, sendo capaz de compilar e carregar programas para a placa com um único clique. A Figura 2 representa a janela principal dessa IDE.

**Figura 2. IDE de programação**



**FONTE:(MARGOLIS, 2012)**

Ele possui a capacidade de programar em C/C++. Isto permite criar com facilidade muitas operações de entrada e saída, tendo que definir apenas duas funções para fazer um programa funcional:

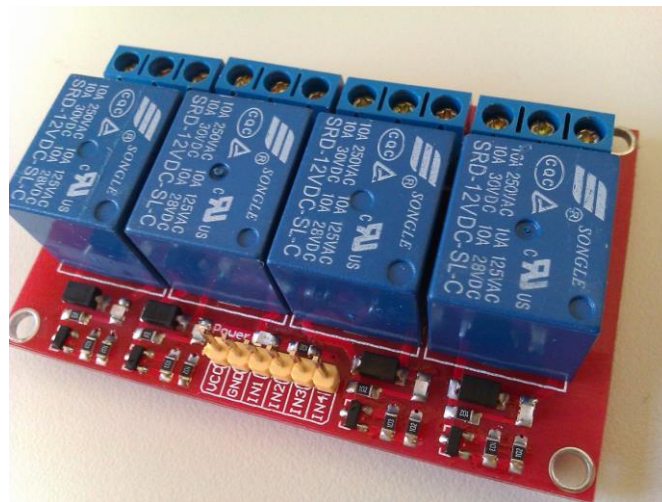
- *setup()* – Inserida no início, na qual pode ser usada para inicializar configuração, e
- *loop()* – Chamada para repetir um bloco de comandos ou esperar até que seja desligada.

### 2.3.2 Shields e sensores

Existe uma série de componentes que amplificam o poder da plataforma ARDUINO, esses componentes são chamados de Shields e permitem, entre outras coisas, que a plataforma possa ser conectada a uma rede doméstica ou a grande rede Internet, conexão via Bluetooth, acionamento de equipamentos de alta tensão, captação e processamento de áudio, etc. Nas próximas subseções são listados os Shields utilizados nesse trabalho.

#### 2.3.2.1 Shield relé

**Figura 3. Shield relé**

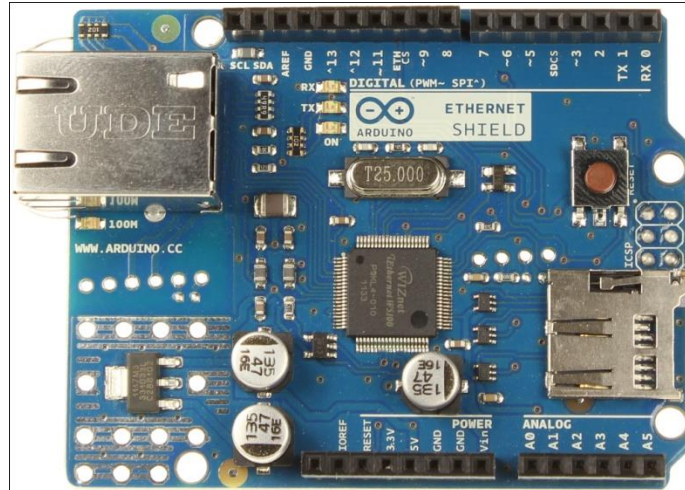


**FONTE: (ARDUINO, 2013)**

A Shield relé exibida na Figura 3 possibilita o acionamento de equipamentos elétricos de alta tensão, como lâmpadas, motores, entre outros. Geralmente ela é composta não só por relés (podem ser mais de um relé por shield), mas também por resistores e transistores, bastando conectá-la ao arduino para que ele possa controlar os equipamentos.

### 2.3.2.2 Shield ethernet

**Figura 4. Shield ethernet**



**FONTE: (ARDUINO, 2013)**

O Shield Ethernet exibida na Figura 4 permite conectar o Arduino à internet ou à uma rede local. Basta ligar este módulo à placa Arduino, conectá-lo à rede com um cabo RJ45 e fazer configurações como a definição de um endereço IP local para começar a controlar seus projetos através da internet. A biblioteca EthernetShield possui alguns exemplos predefinidos que podem ser utilizados como referencia na criação de *web servers* e *web clients*.

**Tabela 2. ESPECIFICAÇÃO SHIELD ETHERNET**

|                              |                                   |
|------------------------------|-----------------------------------|
| <b>TENSÃO DE OPERAÇÃO</b>    | 5V (fornecida pela placa Arduino) |
| <b>CONTROLADOR ETHERNET</b>  | W5100 com buffer interno 16K      |
| <b>VELOCIDADE DE CONEXÃO</b> | 10/100Mb                          |

### 2.3.2.3 Servo motor

**Figura 5. Servo motor**



**FONTE:(ARDUINO,2013)**

Servo motores, como o exibido na Figura 5, são motores de posição frequentemente usados em automação e robótica. Um servo motor tipo RC consiste essencialmente em um motor de corrente contínua com um circuito de controle de posição acoplado. Eles não dão uma volta completa em seu eixo, eles possuem uma faixa ou de 90 ou 180 graus em seu eixo, por isso, são simplesmente utilizados para posicionar objetos. Nesse trabalho utiliza-se um servo motor para posicionar uma câmera de segurança.

### 3 DESENVOLVIMENTO

A necessidade deste trabalho surgiu do projeto de pesquisa "UFERSA virtual". Um projeto que mapeia a Universidade Federal Rural do Semiárido em um ambiente virtual 3D e tem por finalidade promover aos alunos, professores, funcionários e visitantes a sensação de caminhar e explorar o espaço físico da universidade de qualquer lugar, utilizando um computador com acesso à Internet.

Após o mapeamento 3D de alguns prédios da instituição, resolveu-se desenvolver uma comunicação desse ambiente virtual com o ambiente real, para assim gerar uma maior sensação de realidade ao usuário. Já que quando luzes e outros equipamentos estivessem ligados no ambiente real, os equipamentos associados a eles também estariam ligados no ambiente virtual, assim representariam o estado real de cada lâmpada ou equipamento eletrônico da UFERSA na UFERSA virtual.

Como não seria possível a imediata instalação do sistema nos prédios da universidade, desenvolveu-se uma réplica de um dos prédios para a partir dele serem realizados testes e validações. Portanto neste trabalho foi desenvolvida uma maquete física para a instalação do circuito de automação de lâmpadas e posicionamento de câmeras de segurança. Na maquete todas as lâmpadas são representadas por LEDs, utilizados por causa do seu baixo custo de aquisição e do tamanho reduzido da maquete. Porém também foi desenvolvido um outro circuito que possui uma lâmpada 220V que é controlada a partir do mesmo sistema.

Desenvolveu-se um ambiente virtual 2D, onde todas as salas são dispostas com seus respectivos estados (LED aceso/ LED apagado) e pode-se observar em tempo real as imagens capturadas pela câmera de segurança. Esse ambiente foi construído para a partir dele realizar a conexão ("Linkagem") com o ambiente físico e enfim poder controlar e monitorar câmeras e LEDs.

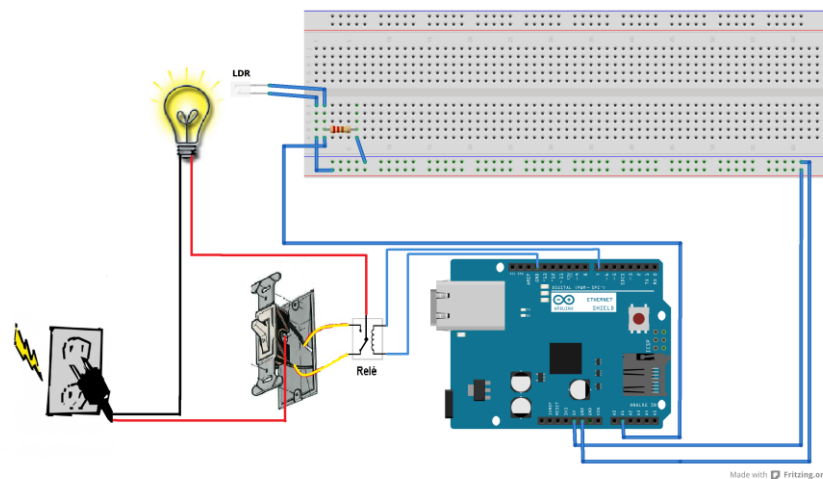
Para realizar essa conexão, utilizou-se a plataforma Arduino e seus Shields e também foi construído um circuito de controle (Ambiente Físico) que é descrito na próxima subseção

#### 3.1 O AMBIENTE FÍSICO

Para desenvolver o sistema de automação criou-se um circuito equivalente a Figura 6 que irá representar o ambiente físico, onde será possível acionar uma lâmpada convencional

(Ligar/Desligar) a partir de um interruptor físico ou virtual. Para isso foi necessário a implementação de um circuito com acionamentos de relés. Utilizou-se um interruptor em paralelo com o relé, para simular o sistema completo (físico + virtual), o que são comumente conhecidos como interruptores paralelos (*Three Way*).

**Figura 6. Esquema do circuito**



**FONTE: AUTORIA PRÓPRIA**

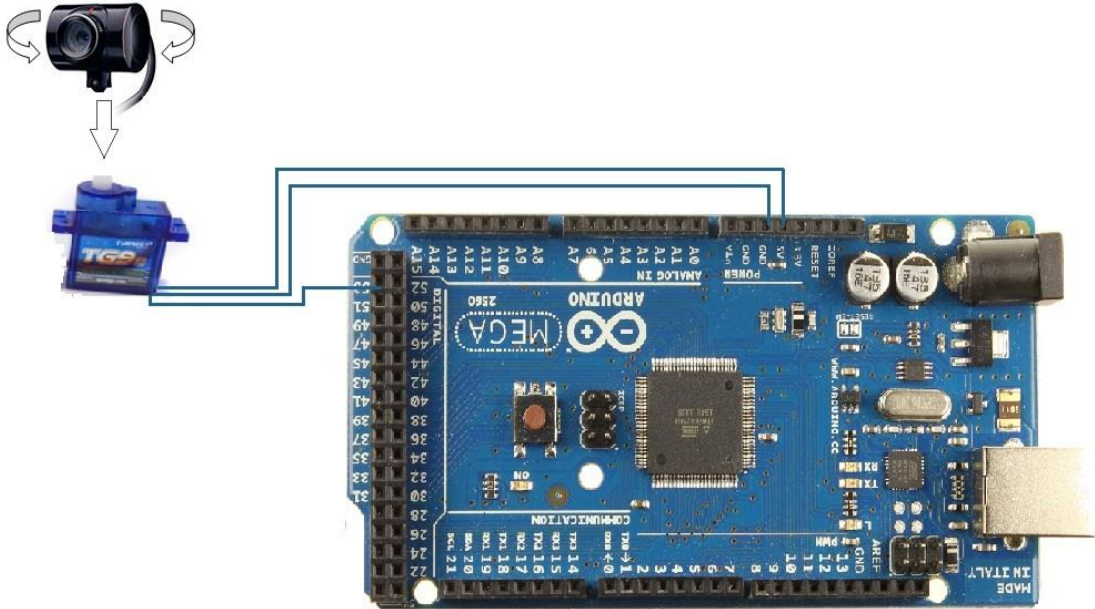
O acionamento do relé é controlado a partir de uma plataforma de prototipagem ARDUINO MEGA 2560 com um *Shield* de extensão *Ethernet*. O conjunto (Arduino + *Shield Ethernet*) permitirá que o arduino possa se conectar à uma rede doméstica ou até mesmo à Internet, dessa forma, conseguimos programar o ARDUINO para operar como um pequeno servidor HTTP. Esse servidor recebe requisições HTTP do ambiente virtual 2D e com base nelas permite o acionamento de equipamentos conectados ao circuito.

Informações sobre o estado (ligado/desligado) de algum equipamento conectado ao circuito também podem ser enviadas aos clientes conectados ao servidor. Para retornar informações do estado das lâmpadas (ligada/desligada) do circuito foi utilizado um sensor de luminosidade LDR para cada lâmpada, onde de acordo com a variação de luminosidade o arduino irá definir variáveis internas do servidor como “on” ou “off”. Com essas informações o ambiente virtual 3D/2D pode exibir lâmpadas simulando o estado da lâmpada real.

Outro serviço implementado ao sistema de automação é o posicionamento e monitoramento de câmeras de segurança. Para desenvolver esse serviço utilizou-se um servo motor, onde a câmera foi acoplada para ser rotacionada como exibido na Figura 7. O motor é

controlado a partir do ambiente virtual 2D/3D onde pode-se rotacionar para direita ou esquerda e dessa forma acompanhar as imagens geradas em tempo real pela câmera.

**Figura 7. Câmera e servo motor**



**FONTE: AUTORIA PRÓPRIA**

### 3.1.1 A maquete física e a utilização de LEDs

Para melhor visualização das ações do sistema de automação desenvolvido, construiu-se uma maquete física em escala reduzida do prédio central de aulas do departamento de ciências exatas e naturais. A maquete (Figura 8) é constituída por 6 (seis) salas de aula, um corredor de acesso a estas salas e um pátio.

**Figura 8. Maquete física**



**FONTE: AUTORIA PRÓPRIA**

Devido ao tamanho da maquete, concluiu-se que seria melhor a utilização de LEDs para a representação das lâmpadas. No entanto, o acionamento originalmente desenvolvido para lâmpadas 220V através de relés só poderia ser feito com os LEDs se para isso o trabalho dispusesse de relés de tensão contínua (5V para 5V), como não tínhamos esse componente (que resultaria na melhor solução de implementação do sistema), desenvolveu-se então uma solução alternativa.

#### 3.1.1.1 Circuito equivalente da maquete

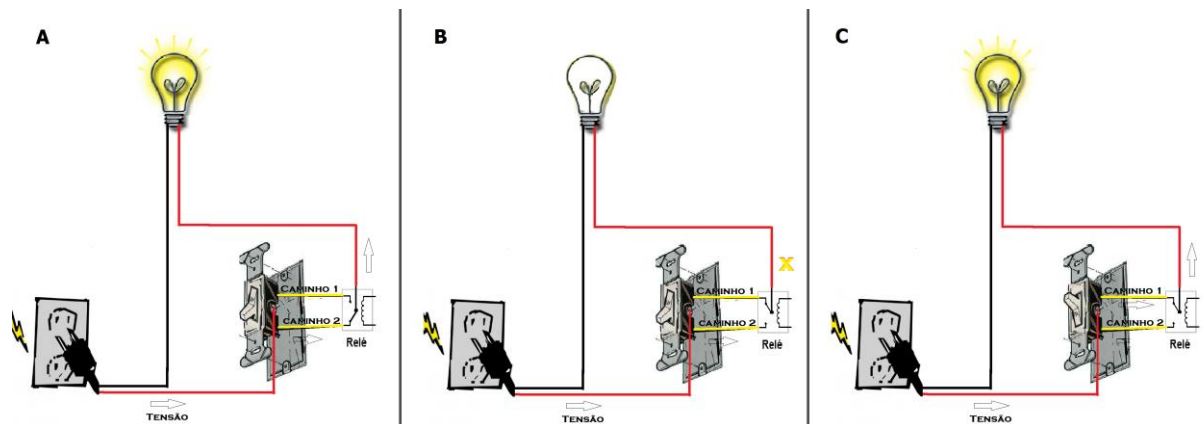
Antes de explicar o circuito equivalente da maquete será explicado primeiramente o funcionamento do circuito com uma lâmpada 220V. Neste caso, um relé com 3 caminhos, um normalmente aberto, outro normalmente fechado e um último caminho que fecha o circuito.

A corrente 220V pode passar por dois caminhos, isso é definido pelo interruptor, que pode deixar passar a corrente pelo seu pino esquerdo ou deixar que a corrente passe pelo seu pino direito. A partir de então cabe ao relé (controlado pelo arduino) decidir qual caminho vai fechar o circuito, se for o caminho por onde está passando corrente a lâmpada acende, senão a lâmpada apaga.

A Figura 9 exibe o esquema anteriormente explicado, onde em Figura 9(A) a lâmpada está acesa. Observa-se que o interruptor permite a passagem de tensão pelo caminho 2 (seu pino direito) e o relé está permitindo a passagem de tensão pelo caminho normalmente fechado (que está associado ao pino direito do interruptor), dessa forma a lâmpada acende. Já em Figura 9(B), o relé é acionado invertendo as passagens de tensão, como o interruptor

continua mandando corrente pelo caminho 2, o ocorre uma interrupção no circuito e por isso a lâmpada apaga. Na Figura 9(C) o interruptor é modificado permitindo a passagem de corrente pelo caminho 1(seu pino esquerdo), dessa forma o circuito é novamente conectado e a corrente 220V chega até a lâmpada que acende.

**Figura 9. Esquema three way**

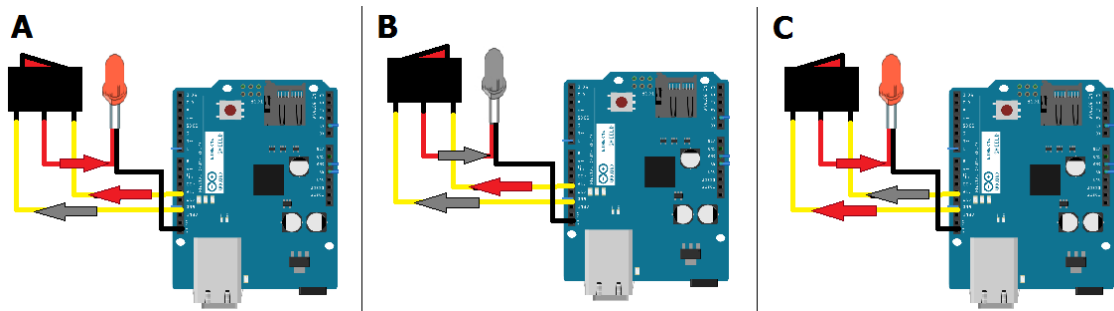


**FONTE: AUTORIA PRÓPRIA**

A solução para o LED segue esse raciocínio. Em um momento a corrente passa pelo pino direito do interruptor, e em outro momento ela passa pelo pino esquerdo. Porém, nessa solução, o LED está ligado no seu pino central do interruptor e o arduino controla as duas cargas ligadas aos pinos extremos do interruptor, como exibido na Figura 10. Essas cargas são inversas, ou seja, em um momento a carga do pino esquerdo será 1(5V) e consequentemente a carga do pino direito será 0(0V), quando o ambiente virtual envia uma requisição de acionamento do LED, as cargas são invertidas, assim, a carga do pino esquerdo será 0 e consequentemente a carga do pino direito será 1.

Levando em consideração o caso da Figura 10(A) pode-se observar que o arduino utiliza dois pinos digitais para enviar duas cargas para os pinos extremos do interruptor, essas cargas são representadas pelas setas vermelhas(5V) e pretas(0V), o interruptor permite a passagem de corrente do pino esquerdo, essa corrente está energizada com 5V e por isso o LED acende. No caso da Figura 10(B) ocorre uma mudança no interruptor, e este passa fechar o circuito com o pino da direita, que (como a seta preta indica) está energizado com 0V e por isso o LED apaga. Já na Figura 10(C) ocorre uma mudança a partir do arduino que inverte as cargas, como o interruptor permite a passagem pelo caminho energizado com 5V o LED volta a acender. Com esse esquema é possível acionar os LEDs tanto fisicamente como virtualmente.

**Figura 10. Esquema LED**



**FONTE: AUTORIA PRÓPRIA**

Depois de implementar o ambiente físico, a segunda parte do trabalho consistiu em desenvolver o ambiente virtual que será a interface de controle desse sistema. A próxima seção mostra como ele foi feito.

### 3.2 O AMBIENTE VIRTUAL

Desenvolveu-se um ambiente virtual 2D que serve de interface ao usuário e a partir dele pode-se controlar os LEDs e a câmera de segurança do sistema. Esse ambiente retrata seis (6) salas de aula assim como no ambiente físico, todas as salas são dispostas com seus respectivos estados (LED aceso/ LED apagado) e pode-se observar em tempo real as imagens capturadas pela câmera de segurança.

O ambiente foi implementado em HTML que é uma linguagem interpretada pelos navegadores web e utilizada no desenvolvimento de sites. Com HTML pode-se construir a estrutura básica de um hipertexto contendo links, botões, tabelas, textos, scripts, imagens, entre outros. Nesse trabalho muitas de suas estruturas foram utilizadas.

Também se utilizou o JavaScript, linguagem interpretada comumente utilizada para a implementação de ações através de funções e nesse trabalho foi incorporado ao HTML para atribuição de funções aos botões.

Com o HTML criou-se uma tabela com três (3) colunas e três (3) linhas, sendo que a segunda linha teve as colunas mescladas para representar o corredor. As demais células da tabela representam as salas de aula. Cada célula tem uma imagem de uma sala de aula com uma lâmpada(essa lâmpada pode estar acesa ou apagada, depende do estado atual do LED) e

um botão com o identificador da sala (Ex.: Sala 1), a cada um desses botões foi atribuída uma função JavaScript diferente, responsável por acionar o LED correspondente à cada sala.

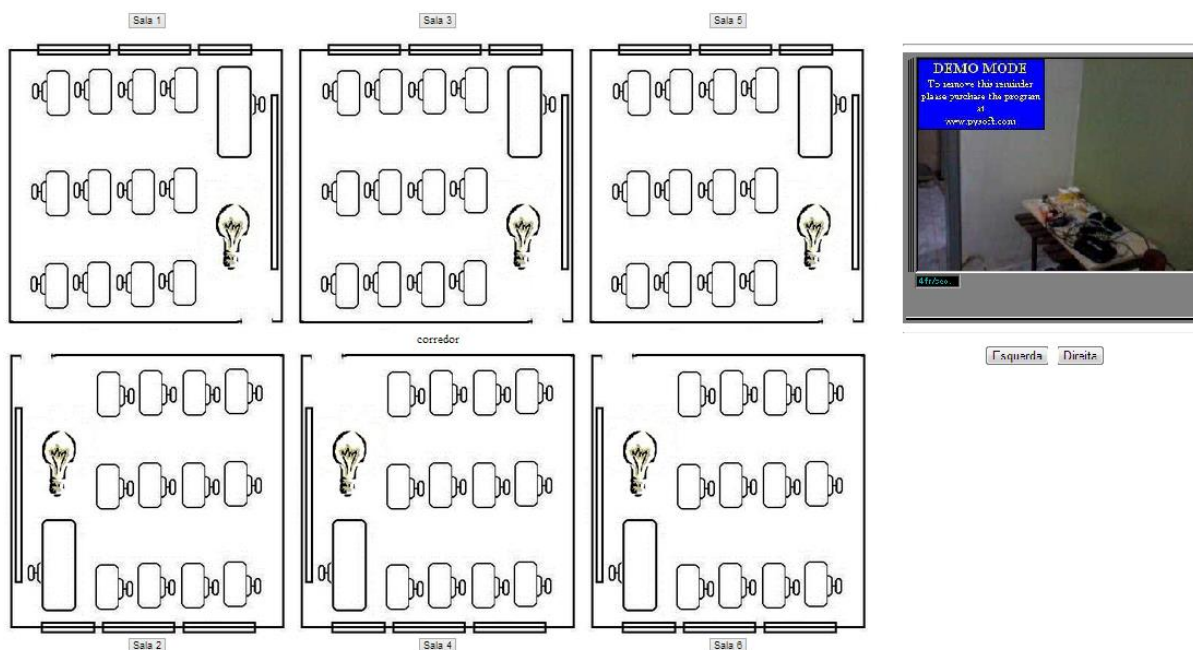
Para exibir as imagens da câmera de segurança utilizou-se uma estrutura HTML chamada *object*. Essa estrutura permite que uma página HTML exiba outra página dentro de uma região pré-estabelecida pelo *object*. A página exibida dentro do *object* é a página da câmera de segurança. No Quadro 1 pode-se ver o trecho de código responsável pela criação e exibição da estrutura *object*.

**Quadro 1. Código da estrutura *object* para câmera**

```
<input type="button" value="Esquerda"
      onclick="Func1c()" ">
<input type="button" value="Direita"
      onclick="Func1d()" ">
<!-- [if IE]>
<object classid="clsid:25336920-03F9-11CF-8FD0-00AA00686F13 "
data="http://www.maujor.com/">
<p>conteúdo alternativo para tecnologias que não suportam
OBJECT</p>
</object>
<![endif]-->
<!-- [if !IE]> <-->
<object type="text/html" width="400" height="400 "
data="http://10.0.0.100:8082">
<p>conteúdo alternativo para tecnologias que não suportam
OBJECT</p>
</object>
<!--> <![endif]-->
```

O ambiente virtual desenvolvido é exibido na Figura 12 a seguir, onde pode-se observar as 6(seis) salas com suas lâmpadas, assim como os botões de acionamento dos LEDs e a imagem da câmera de segurança.

**Figura 11. Interface do ambiente virtual**



**FONTE: AUTORIA PRÓPRIA**

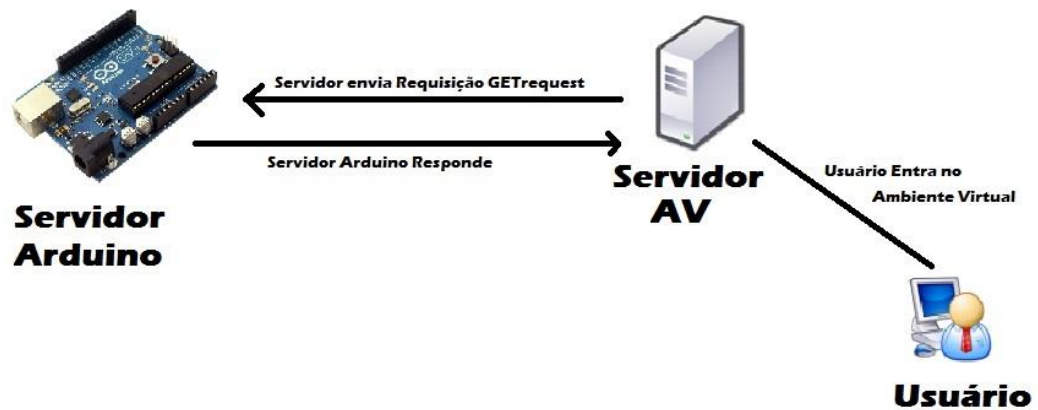
Para melhor contextualização do conteúdo, as funções JavaScript utilizadas no AV são descritas na sessão 3.3, que aborda assuntos referentes a conexão/comunicação entre os dois ambientes.

### 3.3 A COMUNICAÇÃO ENTRE OS SERVIDORES

Como relatado anteriormente, o arduino é programado para funcionar como um pequeno servidor HTTP onde recebe requisições de ações e estados do servidor do ambiente virtual 2D/3D, essa seção explica como a comunicação entre esses dois servidores acontece.

Para acessar o sistema, o usuário deve se conectar ao servidor do ambiente virtual (AV) como representado na Figura 12, após isso ele terá acesso a todos os controles de salas e câmeras. Navegando pelo AV ele decide clicar em um botão que ativa/desativa uma lâmpada. Nesse momento o servidor do AV percebe essa ação e envia uma requisição HTTP com o sinal referente à lâmpada que o usuário quer ativar/desativar ao servidor do arduino, este por sua vez, verifica o estado da lâmpada (on/off) e simplesmente inverte-o. Na resposta HTTP o estado atualizado da lâmpada é passado para o servidor do AV que modifica a cor do objeto lâmpada selecionado pelo usuário de acordo com seu novo estado (on/off).

Figura 12. Comunicação entre o arduino e o AV



FONTE: AUTORIA PRÓPRIA

A função JavaScript exibida no Quadro 2 é utilizada pelo servidor do AV para enviar a requisição HTTP para o servidor do ARDUINO, observa-se o parâmetro “lamp1” que indica qual lâmpada o usuário deseja ativar/desativar. A variável “a”, parâmetro da função *Func2(a)* é uma string utilizada para o retorno do estado da lâmpada.

**Quadro 2. Função de acionamento do LED implementada no AV**

```

function Func2(a)
{
    var http = new XMLHttpRequest();
    var url = "http://192.168.0.177/buzz";

    var params = "lamp1=0";

    http.open("POST", url, true);

    http.setRequestHeader("Content-type", "application/x-www-
form-urlencoded");
    http.setRequestHeader("Content-length", params.length);
    http.setRequestHeader("Connection", "close");

    http.onreadystatechange = function()
    {
        //Call a function when the state changes.
        if(http.readyState == 4 && http.status == 200) {}
    }

    http.send(params);
}

```

A função do Quadro 3 é utilizada pelo servidor do ARDUINO. Após verificar se ocorre uma conexão entre o servidor do AV, verifica-se também se essa conexão é do tipo POST, em caso positivo o parâmetro enviado na requisição POST pelo AV é verificado, nesse trecho de código é observado o parâmetro “lamp1” que implica no acionamento da lâmpada da sala 1. Logo em seguida o valor da variável “tensão” é invertido, o que garante que se a lâmpada estiver acesa, após essa requisição seria apagada ou vice-versa.

Todas as outras requisições, seja para acender qualquer uma das lâmpadas ou para rotacionar o servo motor, ocorrem da mesma forma. A única alteração ocorre no parâmetro enviado pelo servidor do AV, que será referente ao equipamento que deseja-se controlar (lâmpada 1, lâmpada 2, rotacionar servo motor para esquerda ou direita), e para cada parâmetro o arduino executará uma ação diferente.

### Quadro 3. Função no arduino

```

if (type == WebServer::POST)
{
    bool repeat;
    char name[16], value[16];
    do
    {
        repeat = server.readPOSTparam(name, 16, value, 16);

        if (strcmp(name, "lamp1") == 0)
        {
            if(tensao==1) tensao=0;
            else tensao=1;
            digitalWrite(BUZZER_PIN, tensao);
        }

        .
        .
        .

    }while (repeat);

```

No Quadro 4 podemos ver como o ARDUINO controla o servo motor a partir das requisições enviadas pelo AV. Observe que se o parâmetro esquerdo for recebido, o servo motor é reposicionado na posição atual adicionada 20 graus, isso consiste em rotacioná-lo para a esquerda até que atinja a rotação máxima de 180 graus. No caso de receber o parâmetro direita, o servo motor é reposicionado na posição atual decrementada 20 graus, que consiste em rotacioná-lo para a direita até atingir a rotação mínima de 0 graus.

**Quadro 4. Código de controle do servo motor implementado no arduino**

```

if (strcmp(name, "esquerda") == 0)
{
    controlservo +=20;
    if(controlservo <=180) servol.write(controlservo);
else {
    servol.write(180);
    controlservo=180;
}
}

if (strcmp(name, "direita") == 0)
{
    controlservo -=20;
    if(controlservo >=0) servol.write(controlservo);
else {
    servol.write(0);
    controlservo=0;
}
}
}

```

Na outra via de comunicação, representada na Figura 12 como a seta de retorno do servidor arduino para o servidor AV, está o retorno do estado das lâmpadas (ligada/desligada). Esses estados são verificados a todo instante pelo arduino e atualizados, podendo dessa forma garantir o estado real da lâmpada levando em consideração inclusive fatores externos e falhas das lâmpadas (caso de lâmpada queimada).

Como explicado anteriormente, foi utilizado um sensor de luminosidade LDR, esse sensor é conectado ao arduino a partir de um pino analógico. O valor coletado pelo arduino referente ao LDR varia de 0 a 1024(dependendo da quantidade de luz percebida pelo sensor), a partir de então foi determinado um limiar com valor 250, ou seja, se o valor captado pelo

sensor LDR for superior a 250 o LED é considerado aceso, caso contrário o LED é considerado apagado.

O trecho de código do Quadro 5 mostra como o Servidor ARDUINO interpreta essas informações e define as variáveis internas.

**Quadro 5. Leitura dos sensores implementada no arduino**

```
if (type == WebServer::HEAD)

    return;
if(valor_sensor >250)
{
    estado = 1;
    server.print("On");
}
else
{
    estado = 0;
    server.print("Of");
}
```

O valor do sensor é verificado constantemente na função *Loop()* do arduino, o trecho de código do Quadro 5, verifica se esse valor é maior que o limiar definido (esse limiar foi definido através de testes onde esse valor mostrou-se eficiente na divisão dos estados ligada e desligada), após interpretado como ligada o estado é atualizado com o valor “1” e é enviada uma mensagem ao servidor com o texto “On”. No caso de desligada a variável estado é atualizada com o valor “0” e a mensagem enviada tem texto “Of”. Agora cabe ao servidor do AV interpretar as informações de estado das lâmpadas e executar as devidas ações. O trecho de código a seguir mostra como esse passo é realizado.

A função *Func\_loop()* exposta pelo Quadro 6 verifica os campos da variável “a” a cada 0,5 segundos, os campos a[0] e a[1] contém o estado da lâmpada 1, os campos a[2] e a[3] contém o estado da lâmpada 2 e assim sucessivamente.

#### Quadro 6. AV verificando estado dos LEDs

```
function Func_loop(){

    var http = new XMLHttpRequest();
    var url = "http://192.168.0.177/buzz";
    http.open("GET", url, true);

    var a;

    http.onreadystatechange = function() { //Call a function
when the state changes.
        if(http.readyState == 4 && http.status == 200) {
            a=http.responseText;

            if(a[0] == 'O' && a[1]=='n'){

                document.getElementById('im2').setAttribute('src',
'sala_acesa.gif' );
            }
            else{
                document.getElementById('im2').setAttribute('src',
'sala_apagada.gif' );
            }

            http.send(null);
        }
    }
}
```

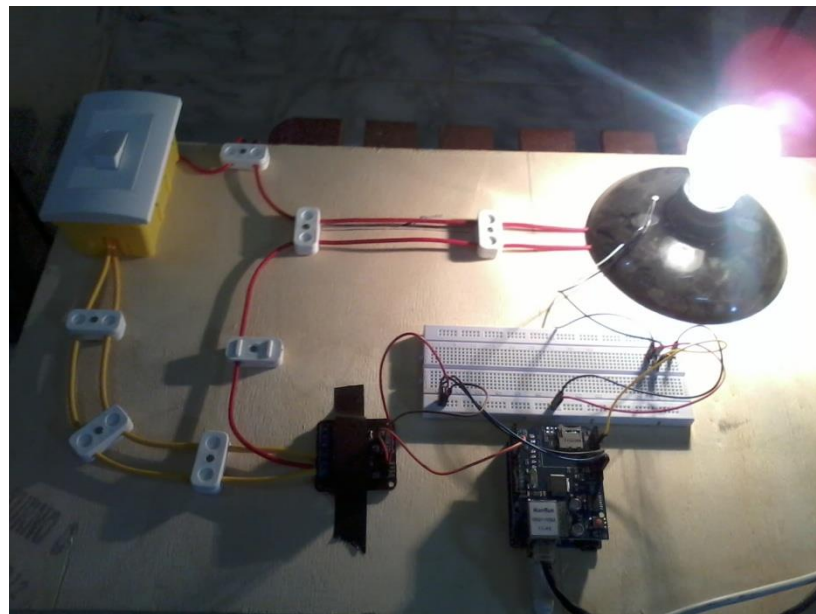
Após a verificação, é constatado se a lâmpada está acesa ou apagada. De acordo com o estado encontrado, o servidor AV executa a ação de substituir a imagem atual que contém a lâmpada com o seu antigo estado (acesa ou apagada) por uma imagem que contém o seu novo estado. A simples modificação de imagens possibilita ao usuário saber o real estado da lâmpada.

## 4 TESTES E RESULTADOS

Iniciou-se o trabalho apenas com a idéia de utilizar o Arduino e construir um sistema de comunicação com um ambiente virtual, simulando assim todas as ações em um ambiente físico.

O primeiro passo foi desenvolver um sistema que utilizasse a internet ou uma rede doméstica para controlar o arduino. Para isso adquiriu-se a plataforma arduino uno, uma *shield ethernet* e um roteador (para a construção da rede local). Resolveu-se então controlar uma lâmpada 220V, e para tanto foi preciso utilizar também uma *shield* relé. Os testes foram feitos e verificou-se que era possível acionar a lâmpada a partir do sistema virtual construído (mas apenas por ele), a partir de então instalou-se um interruptor em paralelo com o relé para que a lâmpada também pudesse ser controlada no “ambiente físico” e um sensor de luminosidade(LDR) para retornar o real estado da lâmpada. Novamente alguns testes foram realizados e os resultados foram satisfatórios. Na Figura 13 podemos ver o circuito construído para a validação dos primeiros testes.

**Figura 13. Resultados dos primeiros testes**



**FONTE: AUTORIA PRÓPRIA**

Com o resultado em mãos compreendeu-se que era possível acionar quantas lâmpadas e/ou equipamentos eletrônicos desejasse e a plataforma suportasse (quantidade de pinos de entrada e saída disponíveis). Dessa forma poderia ser desenvolvido um Ambiente Físico (AF)

que representasse as várias lâmpadas do ambiente virtual. O próximo passo foi construir um ambiente físico que tivesse a aparência do ambiente virtual. Por isso desenvolveu-se a Maquete exibida na Figura 8 na subseção 3.1.1.

Com a maquete pronta, desenvolveu-se um trabalho de instalação de LEDs, sensores de luminosidade, botões e fios para que cada sala da maquete tivesse um LED para ser controlado a partir do ambiente virtual. Após a realização desse último passo outros testes foram feitos e um problema foi diagnosticado. O Arduino UNO, que já não possui tantos pinos de entrada/saída, tinha alguns pinos digitais sendo utilizados pela Shield Ethernet, dessa forma tais pinos sofriam alteração de valores (1 para 0 ou 0 para 1) continuamente de acordo com a troca de dados feita pela rede local. Esse fato impedia que tais pinos fossem utilizados pelo trabalho e diminuiu ainda mais a quantidade de pinos que poderiam ser utilizados, inviabilizando totalmente a utilização da plataforma UNO para o desenvolvimento do trabalho.

Nesse ponto buscou-se uma nova plataforma mais robusta e com mais pinos. Passou-se a utilizar então a plataforma Arduino MEGA 2560 com 50 pinos digitais e 10 pinos analógicos a mais que o Arduino UNO. Novos testes foram feitos e dessa vez todo o sistema funcionou corretamente.

As principais dificuldades encontradas no desenvolvimento desse trabalho foram, configurar o pequeno servidor web no arduino, pois mesmo conseguindo a biblioteca Webduino que oferece tal suporte, surgiram dificuldades de como entender seu funcionamento, adequá-la ao sistema entre outros fatores. Depois desse fato encontrou-se a dificuldade de controlar os LEDs tanto virtual como fisicamente sem a utilização de um relé, a solução para esse problema foi descrita na subseção 3.1.1 A MAQUETE FÍSICA E A UTILIZAÇÃO DE LEDs.

No desenvolvimento da comunicação entre os dois ambientes (virtual e físico) surgiram novos desafios. O mais notável foi como fazer para o AV detectar uma alteração do estado das lâmpadas feito a partir do AF. Várias soluções foram cogitadas, a mais interessante seria a utilização de *sockets* que o servidor web do arduino utilizaria para informar o AV que ocorreu uma alteração no estado de alguma lâmpada no momento em que essa alteração ocorre. Porém existe uma limitação do arduino que não permite a fácil utilização de tal estrutura (existem até alguns tutoriais que fazem vários arranjos para simular tal tecnologia, mas tornaria a implementação muito complexa e custosa). Diante dessa situação encontrou-se

uma solução alternativa. A função “*Func\_loop()*” do ambiente virtual exibida no Quadro 6 é executada a cada 0,5 segundos, dessa forma ela fica praticamente a todo instante verificando o estado de cada LED e atualizando no AV. Essa solução alternativa adotada pode ser considerada uma desvantagem do sistema, pois requisições para saber qual o estado atual do LED são feitas praticamente a todo instante e em casos onde não ocorre nenhuma modificação essa requisição seria desnecessária.

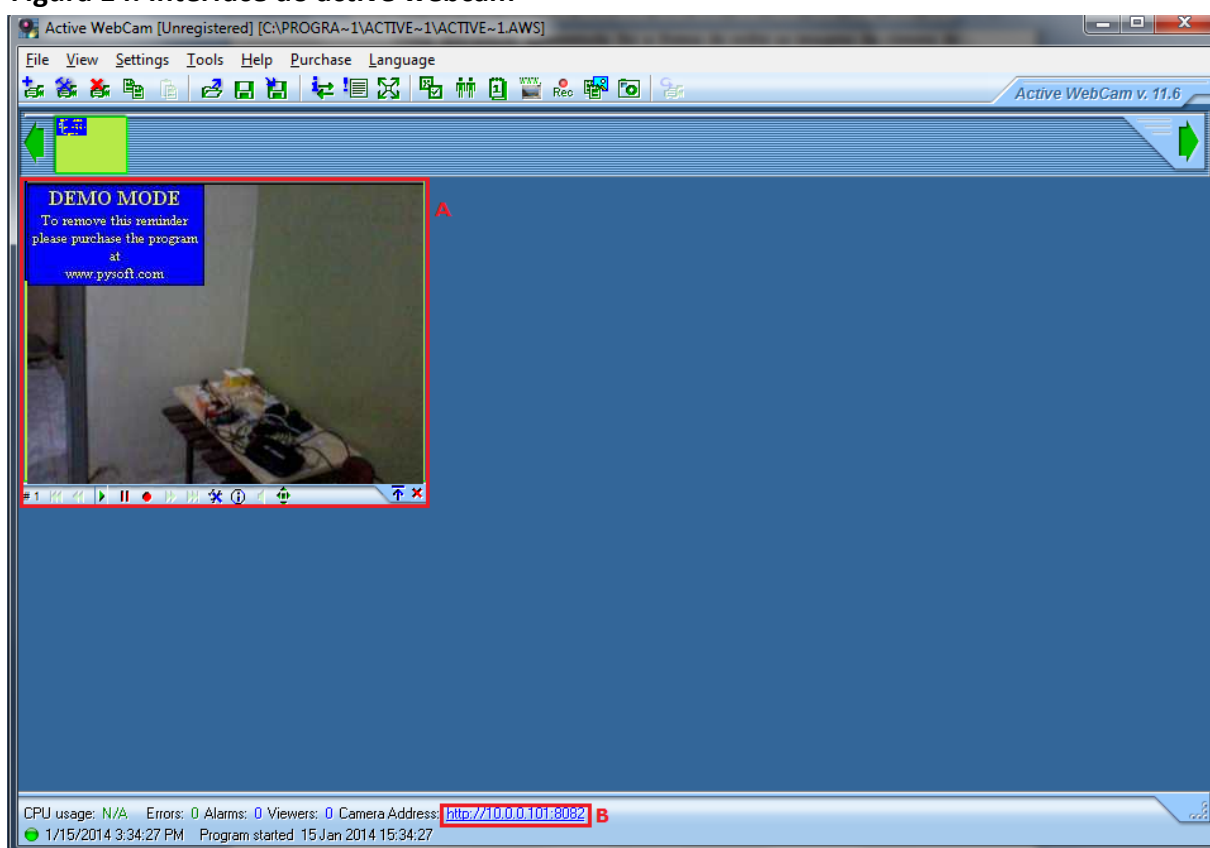
Outra dificuldade apresentada foi a forma de exibir as imagens da câmera de segurança na interface do AV. Ao iniciar a implementação do monitoramento de imagens de uma câmera de segurança em uma aplicação WEB a primeira solução cogitada foi a utilização de uma Câmera IP. Uma câmera IP consiste em um equipamento de captação de vídeo e em alguns casos até mesmo áudio, que é conectada à rede local ou à própria Internet, e disponibiliza a visualização dessas imagens em uma página WEB acessada através de um endereço IP próprio atribuído por essa rede. Utilizando um equipamento como este, bastava apenas introduzir na interface do AV um link para essa página, ou ainda, como supracitado na Seção 3.2, utilizar uma estrutura que permitisse abrir uma página WEB dentro de outra.

Porém a aquisição de uma câmera com tal tecnologia não foi possível, diante de tal fato logo se optou por uma solução alternativa. Essa solução baseia-se na utilização de uma simples Câmera WEBCAM e das possibilidades oferecidas pelo Software Active WebCam. Esse software possibilita a transmissão de imagens capturadas por câmeras analógicas, webcams, câmeras IP, entre outras diretamente para uma página web alocada local ou remotamente, ou ainda possibilita fazer *upload* dessas imagens em um servidor FTP.

De um modo simples a solução encontrada consiste em instalar uma Webcam em um computador que possui o Software Active WebCam, e associar a câmera ao software. Feito isso, é possível criar uma página web HTML que exibe as imagens capturadas pela câmera. Essa página HTML possui um endereço IP local, portanto pode-se abrir essa página dentro da página Web que serve como interface para o AV como explicado na Seção 3.2 no trecho de código exibido pelo Quadro 1. Dessa forma fica mais usual monitorar as imagens geradas por uma simples Webcam diretamente da interface do AV.

Na Figura 14, pode-se observar a interface de utilização do software Active WebCam. Nas caixas vermelhas são destacados em A: imagem da webcam associada ao software. Em B: o endereço IP da página web onde as imagens são disponibilizadas.

**Figura 14. Interface do active webcam**



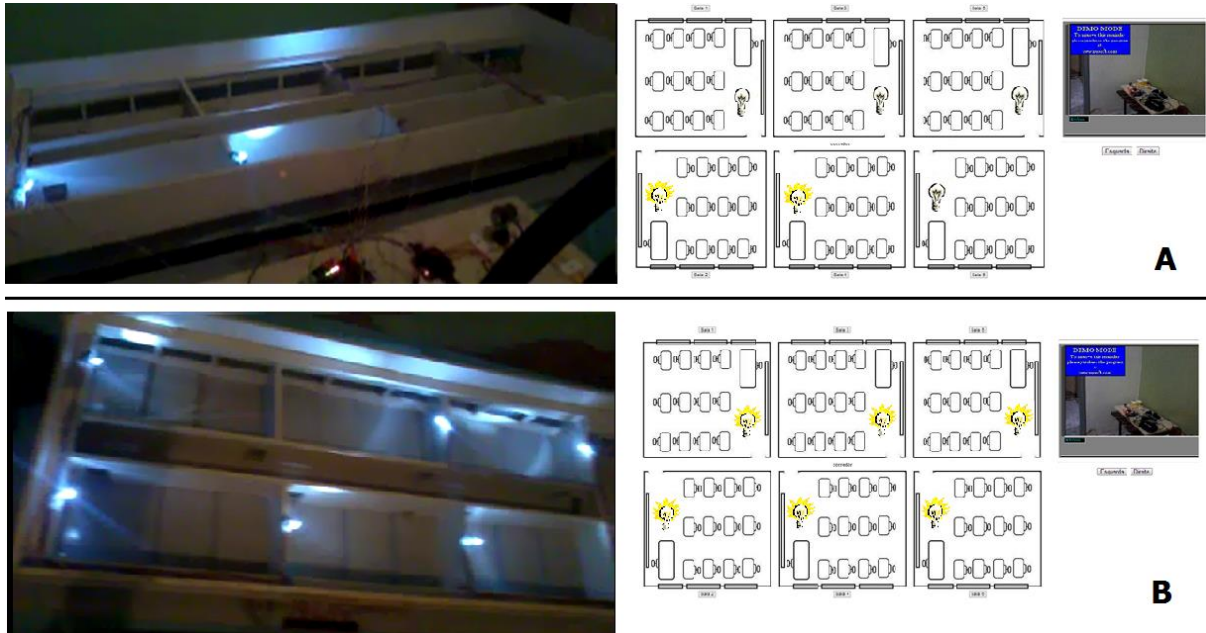
**FONTE: AUTORIA PRÓPRIA**

Após essa implementação, vários testes foram desenvolvidos buscando validar essa solução alternativa que se mostrou eficiente. É importante mostrar que no caso, se utilizar outro tipo de câmera, inclusive câmeras IP, esse software mostra-se muito eficiente, pois além de possibilitar a gravação das imagens permite também o monitoramento e gravação de várias câmeras ao mesmo tempo.

Em caso de aquisição de uma câmera IP e da oportunidade de implantação dessa câmera no sistema, pode-se rapidamente substituir a solução alternativa adotada, necessitando-se apenas modificar o endereço IP exibido pela estrutura object implementada na página HTML do ambiente virtual para o endereço IP da câmera IP.

A Figura 15 exibe o resultado final do trabalho onde utiliza-se o AV 2D para acionar os LEDs da maquete, onde em Figura 15(A) apenas as salas 2 e 4 estão acesas e em Figura 15(B) todas as salas estão acesas.

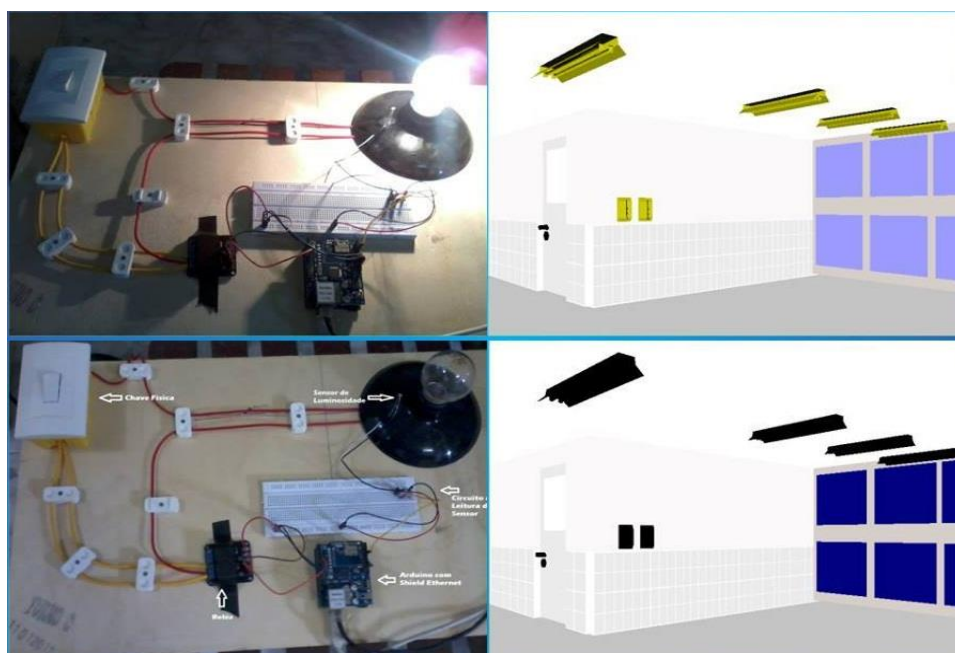
**Figura 15. Resultado final AV 2D**



FONTE: AUTORIA PRÓPRIA

A Figura 16 exibe o resultado obtido acionando uma lâmpada 220V a partir do AV 3D. Observa-se mudanças nas cores das lâmpadas virtuais e nas cores das janelas.

**Figura 16. Resultado final AV 3D**



FONTE:(MAIA,2013)

Ao final do desenvolvimento buscou-se fazer um levantamento de custo de todos os elementos utilizados. Os resultados desse levantamento podem ser observados na Tabela 4.

**Tabela 3. TABELA DE CUSTOS**

| ELEMENTO                                 | CUSTO UNITÁRIO(R\$) | QUANTIDADE | CUSTO TOTAL(R\$) |
|------------------------------------------|---------------------|------------|------------------|
| Arduino UNO                              | 69,00               | 1          | 69,00            |
| Arduino MEGA 2560                        | 179,00              | 1          | 179,00           |
| Shield Ethernet                          | 59,00               | 1          | 59,00            |
| Shield Relé                              | 25,00               | 1          | 25,00            |
| Sensor de Luminosidade LDR               | 3,00                | 7          | 21,00            |
| Servo Motor                              | 55,00               | 1          | 55,00            |
| Interruptor de três caminhos (Three Way) | 13,00               | 1          | 13,00            |
| Lâmpada 220V                             | 7,00                | 1          | 7,00             |
| Bico Base para Lâmpada                   | 5,00                | 1          | 5,00             |
| Câmera WebCam                            | 70,00               | 1          | 70,00            |
| LED                                      | 2,00                | 6          | 12,00            |
| Roteador                                 | 50,00               | 1          | 50,00            |
| Resistores                               | 1,00                | 7          | 7,00             |
| Mini-Interruptor de três caminhos        | 3,00                | 6          | 18,00            |
| Fios                                     |                     | -          | 25,00            |
| Maquete                                  | 350,00              | 1          | 350,00           |
| <b>TOTAL</b>                             |                     |            | <b>965,00</b>    |

Diante dos testes apresentados e dos seus respectivos resultados e validações geradas, pode-se afirmar que trata-se de um sistema simples e funcional, que apesar de apresentar algumas limitações, ainda assim pode ser implantado no ambiente real e garantir uma ação satisfatória.

## 5 CONCLUSÕES

Neste trabalho foram apresentados a implementação e testes de um sistema de automação predial com monitoramento de câmeras de segurança, onde para isso, utilizamos a tecnologia da plataforma de prototipagem arduino.

Como supracitado, esse trabalho surgiu a partir da necessidade do projeto UFERSA virtual, com a finalidade de ampliar a sensação de realidade do ambiente virtual 3D, fazendo com que este atualize informações de acordo com o estado atual do ambiente real. Além da necessidade de apresentar informações atualizadas do ambiente real no ambiente virtual, também se desenvolveu uma segunda via de comunicação entre os dois ambientes, onde o estado do ambiente real pode ser modificado a partir do ambiente virtual.

A eficácia do sistema pode ser percebida através dos bons resultados exibidos. Apesar das dificuldades encontradas pelo caminho, soluções alternativas foram bem aplicadas permitindo dessa forma a boa funcionalidade do sistema e garantindo a conclusão desse trabalho.

A principal vantagem desse trabalho, entre outras, é a utilização da plataforma de prototipagem ARDUINO que possui uma forma de programação simples, baixo custo, e uma série de componentes (*shields*) que ampliam seu poder de processamento. Dessa forma o desenvolvimento foi facilitado, o custo foi barateado e as possibilidades de implementar novas funcionalidades ao sistema explorando outras *shields* disponíveis são muito grandes. Além do mais, quando o trabalho se deparou com o problema da necessidade de mais portas de entrada para controle de mais equipamentos, a migração do sistema entre as plataformas foi trivial, facilitando assim o desenvolvimento de sistemas similares em diferentes modelos de ARDUINO, utilizando o mais adequado para cada aplicação.

A infinidade de possibilidades acarretadas pelo uso do arduino e seus *shields* proporciona uma série de aprimoramentos ao sistema e cria uma leque de outros desenvolvimentos que podem ser feitos utilizando a mesma tecnologia, dessa forma esse trabalho pode servir de porta de entrada para outros projetos de automação na Universidade.

Para tentar eliminar algumas limitações do ARDUINO, como a dificuldade da utilização de *sockets* ou a quantidade de portas de entrada e/ou saída para controle, pretende-se em desenvolver aprimoramentos como o desenvolvimento de uma biblioteca que

possibilite a fácil utilização de *sockets*. Dessa forma o sistema economizaria banda da rede, pois apenas enviaria sinais de que ocorreu uma mudança no estado de um equipamento quando essa, de fato, ocorresse. Outra melhoria seria estudar a forma de interligar vários ARDUINOS para que um complementasse o outro, assim suprimindo a necessidade de mais portas de controle e poder de processamento.

Por fim, é possível ver como um sistema de baixo custo e mesmo com algumas limitações pode se tornar funcional e interessante para algumas aplicações, inicialmente de pequeno porte (pequenos prédios e casas), mas se aprimorado com algumas das sugestões supracitadas pode ser transformado em um sistema de grande porte e introduzido em grandes prédios.

## REFERÊNCIAS

ALIANÇA NETO, A. S. Projeto de Um Sistema de Automação Residencial, 2011. Monografia (Graduação em Ciência da Computação) – Universidade Rural do Semiárido, 2011.

Arduino, Disponível em < <http://arduino.cc/>> Acesso em: Outubro de 2013.

CABRAL, M. M. A.; CAMPOS, A. L. P. S. Sistema de Automação Residencial de Baixo Custo: Uma Realidade Possível. Holos, Ano 24, v. 3, 2008.

DESTACOM– GRUPO DE ROBÓTICA UFMS. Introdução ao Arduino, 2012. 3-6p.

DIAS, C. L. A.; PIZZOLATO, N. D. Domótica: Aplicabilidade e Sistemas de Automação Residencial. Vértices, v. 6, n. 3, set./dez. 2004. 10p.

LEE, J. J. Monitoração Remota via IP de Sensores Industriais, Monografia (Graduação em Engenharia de Computação com Ênfase em Sistemas Embarcados) – Universidade São Paulo, Escola de Engenharia de São Carlos, 2007.

MARGOLIS, M. Arduino CookBook, 2nd Edition, Editora O'Reilly Media, 2012.

MARTE, C. L. Automação Predial: A Inteligência distribuída nas edificações. São Paulo: Carthago e Forte, 1995.

MOTEBELLER, S. J. Sensores sem fios – Avaliação e Emprego na Automação de Sistemas Prediais, 1ª edição, Editora Biblioteca 24 horas, 2011.

NEVES, C.; DUARTE, L.; VIANA, N.; LUCENA Jr, V. F. Os dez maiores desafios da automação industrial: As perspectivas para o futuro. II Congresso de Pesquisa e Inovação da Rede Norte e Nordeste de Educação Tecnológica, João Pessoa – PB, 2007.

ROSÁRIO, J. M. Automação Industrial. São Paulo: Baraúna, 2009. 15-19p.

WebDuino, Disponível em <<https://github.com/sirleech/Webduino>> Acesso em: Julho de 2013.